

มิเตอร์อัจฉริยะเฟสเดียวที่สอบเทียบตัวเองที่สอดคล้องกับโปรโตคอล OpenADR



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2563
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

A SINGLE-PHASE SMART METER WITH AUTOCALIBRATION CONFORMED TO OPENADR
PROTOCOL



A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Electrical Engineering

Department of Electrical Engineering

FACULTY OF ENGINEERING

Chulalongkorn University

Academic Year 2020

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	มิเตอร์อัจฉริยะเฟสเดียวที่สอบเทียบตัวเองที่สอดคล้องกับ โพรโทคอล OpenADR
โดย	นายณพเมธ ปวิธพาณิชย์
สาขาวิชา	วิศวกรรมไฟฟ้า
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.วันเฉลิม โปรา

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่ง
ของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

.....	คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)	
คณะกรรมการสอบวิทยานิพนธ์	
.....	ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.สุรียะ พุ่มรินทร์)	
.....	อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์ ดร.วันเฉลิม โปรา)	
.....	กรรมการภายนอกมหาวิทยาลัย
(รองศาสตราจารย์ ดร.เอกชัย ลีลารัมย์)	

นพเมธ ปวีรพานิชย์ : มิเตอร์อัจฉริยะเฟสเดียวที่สอบเทียบตัวเองที่สอดคล้องกับโปรโตคอล OpenADR. (A SINGLE-PHASE SMART METER WITH AUTOCALIBRATION CONFORMED TO OPENADR PROTOCOL) อ.ที่ปรึกษาหลัก : รศ. ดร.วันเฉลิม โปธา

ในปัจจุบันมิเตอร์อัจฉริยะถือได้ว่าเป็นอุปกรณ์ที่มีความสำคัญต่อระบบโครงข่ายไฟฟ้า โดยจะทำหน้าที่เป็นตัวกลางสื่อสารระหว่างผู้ใช้ไฟฟ้าและผู้ให้บริการไฟฟ้าในการส่งข้อมูลพฤติกรรมการใช้ไฟ มิเตอร์อัจฉริยะนั้นเป็นอุปกรณ์มาตรวัดที่จำเป็นจะต้องทำการสอบเทียบอย่างสม่ำเสมอ เพื่อให้การวัดค่าพลังงานทางไฟฟ้าเกิดความคลาดเคลื่อนน้อยที่สุดเท่าที่เป็นไปได้ ในวิทยานิพนธ์ฉบับนี้นำเสนอการพัฒนา มิเตอร์อัจฉริยะเฟสเดียวที่สามารถสอบเทียบตัวเองได้โดยไม่ต้องใช้แหล่งจ่ายอ้างอิงหรือมิเตอร์อ้างอิงที่มีความแม่นยำสูงในการสอบเทียบ ทำให้การสอบเทียบของมิเตอร์อัจฉริยะนี้เป็นไปอย่างมีประสิทธิภาพ สามารถรักษาความแม่นยำที่ได้อยู่เสมอ ช่วยลดต้นทุนจากการทำการสอบเทียบแบบดั้งเดิม อีกทั้งยังสะดวกต่อการใช้งาน นอกจากนี้มิเตอร์อัจฉริยะในงานวิจัยฉบับนี้ยังมีความสามารถในการตอบสนองด้านโหลดอัตโนมัติที่สอดคล้องกับมาตรฐานโปรโตคอล OpenADR ซึ่งจะสามารถดำเนินการตอบสนองด้านโหลดเพื่อช่วยในการลดช่องว่างระหว่างความต้องการใช้ไฟฟ้าและกำลังการผลิตโดยเฉพาะในช่วงเวลาที่มีความต้องการใช้ไฟฟ้าสูง (Peak) ช่วยให้โครงข่ายไฟฟ้าสามารถรักษาเสถียรภาพ ป้องกันการเกิดไฟดับเป็นวงกว้าง หรือลดความผิดปกติที่ทำให้ความน่าเชื่อถือของระบบโครงข่ายไฟฟ้าต่ำลงได้ ดังนั้นการตอบสนองด้านโหลดจึงเป็นวิธีการสำคัญที่จะช่วยในการจัดการสภาวะวิกฤตด้านพลังงานไฟฟ้า สร้างสมดุลระหว่างความต้องการใช้พลังงานกับกำลังการผลิต และช่วยเสริมความมั่นคงในด้านพลังงานไฟฟ้าทั้งในระยะสั้นและระยะยาว

สาขาวิชา วิศวกรรมไฟฟ้า

ปีการศึกษา 2563

ลายมือชื่อนิสิต

ลายมือชื่อ อ.ที่ปรึกษาหลัก

6270128621 : MAJOR ELECTRICAL ENGINEERING

KEYWORD: Smart Meter, Autocalibration, OpenADR, Automated Demand Response

Noppameth Pawitpanich : A SINGLE-PHASE SMART METER WITH AUTOCALIBRATION CONFORMED TO OPENADR PROTOCOL. Advisor: Assoc. Prof. WANCHALERM PORA, Ph.D.

Nowadays, Smart meters are essential components in smart grids as they provide a connection between utilities and their customers. All smart meters must be calibrated before they can be installed. Moreover, they should be calibrated every few years. In this research describes a single-phase smart meter with an autocalibration feature. The accurate voltage and current sources are in the same chip as the metering part. This makes the calibration process unnecessary. So, cost from laboring and calibration equipment can be saved. Apart from automatic billing, a smart meter can help implementation of Demand Response (DR), which tries to balance the power required by the demand side and power capability of the supply side. DR is one of the most important functions in smart grid operation. However, no commercial smart meters in the market have yet to provide this DR function. Besides autocalibration, this prototype can communicate with the utility according to the OpenADR standard.

Field of Study: Electrical Engineering

Student's Signature

Academic Year: 2020

Advisor's Signature

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สามารถสำเร็จลุล่วงไปด้วยดี เนื่องจากได้รับความช่วยเหลือเป็นอย่างสูงจาก รศ. ดร. วันเฉลิมโปธา ซึ่งเป็นอาจารย์ที่ปรึกษาวิทยานิพนธ์ ข้าพเจ้าขอขอบพระคุณเป็นอย่างสูงที่กรุณาให้ความช่วยเหลือด้วยความเอาใจใส่อย่างดียิ่ง คอยเป็นแรงผลักดัน สนับสนุน รวมถึงให้คำแนะนำ ข้อคิดเห็น ทั้งในด้านวิชาการ งานวิจัย ความรู้ การเขียนวิทยานิพนธ์และอื่น ๆ อีกมาก รวมถึงโอกาสที่ดี อีกมากมายที่ท่านเคยได้มอบให้แก่ข้าพเจ้า

ขอขอบพระคุณ ผศ. ดร. สุรีย์ พุ่มรินทร์ และ รศ. ดร. เอกชัย สีสารัมภ์ เป็นอย่างสูงที่ได้ให้เกียรติมาเป็นกรรมการวิทยานิพนธ์ รวมถึงการแสดงความคิดเห็น ตรวจสอบ และแก้ไขให้วิทยานิพนธ์ฉบับนี้ให้สำเร็จลุล่วงไปด้วยดี

ขอขอบพระคุณเหล่าคณาจารย์ ทั้งจากภาควิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมเครื่องกล และภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัยทุกท่าน ที่เป็นผู้ประสพวิชา ให้ความรู้ คำแนะนำ ให้การสนับสนุนแก่ข้าพเจ้าได้พัฒนาศักยภาพตนเองในหลาย ๆ ด้าน ขอขอบพระคุณบุคลากรและเจ้าหน้าที่ประจำภาควิชาวิศวกรรมไฟฟ้าทุกท่านที่คอยให้ความช่วยเหลือที่ดีตลอดมา และขอขอบคุณเหล่านิสิต เพื่อนพี่น้องในห้องปฏิบัติการ Embedded System and IC Design Research Laboratory (ESID) ทุกคนที่คอยสนับสนุน ให้ความช่วยเหลือ มุมมอง ความรู้ใหม่ ๆ รวมถึงแลกเปลี่ยนประสบการณ์ที่ดีต่อกัน

สุดท้ายนี้ ข้าพเจ้าขอกราบขอบพระคุณบิดา มารดา ครอบครัว เป็นอย่างสูง ที่คอยให้ความช่วยเหลือคำจุน สนับสนุนส่งเสริม เข้าอกเข้าใจ และเป็นกำลังใจที่ดีแก่ข้าพเจ้าตลอดมาจนวิทยานิพนธ์สำเร็จลุล่วงไปได้ด้วยดี

นพเมธ ปวิธพาณิชย์

สารบัญ

	หน้า
.....	ค
บทคัดย่อภาษาไทย.....	ค
.....	ง
บทคัดย่อภาษาอังกฤษ.....	ง
กิตติกรรมประกาศ.....	จ
สารบัญ.....	ฉ
สารบัญตาราง.....	ญ
สารบัญรูปภาพ.....	ฎ
บทที่ 1 บทนำ.....	1
1.1 ที่มาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของวิทยานิพนธ์.....	2
1.3 ขอบเขตของวิทยานิพนธ์.....	3
1.3.1 ภาคฮาร์ดแวร์.....	3
1.3.2 ภาคซอฟต์แวร์.....	3
1.3.3 การทดสอบไมโครอัจฉริยะ.....	4
1.4 ขั้นตอนการดำเนินงานวิทยานิพนธ์.....	4
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	4
1.6 โครงสร้างของวิทยานิพนธ์.....	5
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	6
2.1 โครงสร้างพื้นฐานระบบไมโครcontrollerชั้นสูงในโครงข่ายไฟฟ้าอัจฉริยะ.....	6
2.1.1 ไมโครอัจฉริยะ.....	9

2.1.2	โครงข่ายการสื่อสาร.....	11
2.1.3	ระบบบริหารจัดการข้อมูล.....	12
2.2	การตอบสนองด้านโหลด	13
2.2.1	รูปแบบของการดำเนินการตอบสนองด้านโหลด	14
2.2.2	ประเภทของการตอบสนองด้านโหลดและความยืดหยุ่นในการจัดการระบบไฟฟ้า	15
2.2.2.1	มาตรการการตอบสนองด้านโหลดต่อความน่าเชื่อถือของระบบ (Reliability-based Options)	15
2.2.2.2	มาตรการการตอบสนองด้านโหลดต่อกลไกราคา (Price-based Options).....	16
2.2.3	ประโยชน์ของการตอบสนองด้านโหลด	18
2.2.4	ความซับซ้อนการดำเนินการตอบสนองด้านโหลดแบบควบคุมด้วยเจ้าหน้าที่ กึ่งอัตโนมัติ และอัตโนมัติสมบูรณ์.....	19
2.3	ทฤษฎีซอฟต์แวร์ที่เกี่ยวข้อง.....	21
2.3.1	ซอฟต์แวร์การตอบสนองด้านโหลดแบบอัตโนมัติ OpenADR.....	21
2.3.1.1	การปรับใช้รูปแบบตามสถานการณ์ (Deployment Scenarios)	22
2.3.1.2	ประเภทของโปรแกรมการตอบสนองด้านโหลดที่รองรับ (Demand Response Program Type).....	29
2.3.1.3	ความสัมพันธ์ระหว่างสถานการณ์การใช้งานโปรแกรมกับประเภทของการ ตอบสนองด้านโหลด.....	30
2.3.2	ซอฟต์แวร์ไอโอทีคลาวด์แพลตฟอร์ม (NETPIE)	31
2.3.3	ซอฟต์แวร์ฐานข้อมูลเอสคิวแอลไลต์ (SQLite).....	33
2.4	หลักการคำนวณกำลังไฟฟ้าพื้นฐานทั่วไป	34
2.4.1	ค่ากำลังไฟฟ้าจริง	35
2.4.2	ค่ากำลังไฟฟ้ารีแอกทีฟ.....	36
2.4.3	ค่ากำลังไฟฟ้าปรากฏ.....	36
2.4.4	ค่าตัวประกอบกำลังไฟฟ้า.....	36

บทที่ 3 การออกแบบฮาร์ดแวร์ของมิเตอร์อัจฉริยะ.....	38
3.1 รายละเอียดอุปกรณ์ที่ใช้ในมิเตอร์อัจฉริยะ	39
3.1.1 ไมโครคอนโทรลเลอร์.....	39
3.1.2 ไอซีวัดค่าพลังงานทางไฟฟ้า	41
3.1.2.1 หลักการวัดกระแสของไอซี	42
3.1.2.2 หลักการวัดแรงดันของไอซี	44
3.1.2.3 หลักการสอบเทียบตัวเองของไอซี.....	45
3.1.3 บอร์ดราสเบอร์รี่พาย	50
3.1.4 หน่วยความจำภายนอก	51
3.2 ฮาร์ดแวร์มิเตอร์อัจฉริยะต้นแบบที่พัฒนาขึ้น.....	52
บทที่ 4 การออกแบบและพัฒนาซอฟต์แวร์ของมิเตอร์อัจฉริยะ.....	54
4.1 ซอฟต์แวร์ที่พัฒนาสำหรับมิเตอร์อัจฉริยะ.....	54
4.1.1 การกักการวัดค่าพลังงานทางไฟฟ้า (Energy Metering).....	56
4.1.2 การกักการสอบเทียบตัวเอง (Autocalibration).....	56
4.1.3 การกักส่งข้อมูลขึ้นคลาวด์ (NETPIE Dashboard).....	57
4.1.4 การกักเชื่อมต่ออินเทอร์เน็ตแบบไร้สาย (Wi-Fi).....	58
4.1.5 การกักจัดเก็บข้อมูลลงฐานข้อมูล (SQLite Database).....	58
4.2 ซอฟต์แวร์ไอโอทีคลาวด์แพลตฟอร์ม (NETPIE).....	59
4.3 ซอฟต์แวร์การตอบสนองด้านโหลด OpenADR.....	61
4.3.1 โครงสร้างสถาปัตยกรรมพื้นฐานของซอฟต์แวร์ OpenADR	61
4.3.2 กระบวนการทำงานของโปรโตคอล OpenADR.....	62
4.3.2.1 บริการการลงทะเบียน (Registration Service).....	63
4.3.2.2 บริการเหตุการณ์ (Event Service).....	64
4.3.2.3 บริการรายงาน (Report Service)	65

4.3.2.4 บริการตรวจสอบสถานะ (OpenADR Poll).....	65
4.3.2.5 บริการเข้าร่วมมาตรการ (Opt Service).....	66
4.3.3 การพัฒนาซอฟต์แวร์ OpenADR.....	68
4.3.3.1 การพัฒนาซอฟต์แวร์โหนดต้นเสมือน.....	69
4.3.3.2 การพัฒนาซอฟต์แวร์โหนดปลายเสมือน.....	70
บทที่ 5 การทดลองและผลการทดลอง.....	75
5.1 ทดสอบวัดค่าพลังงานทางไฟฟ้าจากการใช้งานจริง.....	75
5.2 ทดสอบความผิดพลาดการวัดค่าพลังงานทางไฟฟ้าของมิเตอร์.....	77
5.2.1 ผลการทดสอบความผิดพลาดการวัดค่าพลังงานไฟฟ้าของมิเตอร์.....	78
5.3 ทดสอบการส่งข้อมูลพลังงานทางไฟฟ้าไปยัง NETPIE Dashboard.....	83
5.4 ทดสอบการสอบเทียบตัวเองของมิเตอร์อัจฉริยะต้นแบบ.....	84
5.5 ทดสอบการจัดเก็บข้อมูลลงฐานข้อมูลด้วย SQLite.....	84
5.6 ทดสอบซอฟต์แวร์การตอบสนองด้านโหลด OpenADR 2.0.....	85
5.6.1 ทดสอบการลงทะเบียนโหนดปลายเสมือนเข้าระบบ.....	85
5.6.2 ทดสอบการสร้างเหตุการณ์ด้วยโหนดต้นเสมือน.....	89
5.6.3 ทดสอบการโต้ตอบระหว่างโหนดต้นเสมือนและโหนดปลายเสมือน.....	92
5.6.4 ทดสอบการทำงานตอบสนองด้านโหลดด้วยซอฟต์แวร์ OpenADR ที่พัฒนาขึ้น.....	97
บทที่ 6 สรุปผลงานวิจัยและข้อเสนอแนะ.....	100
6.1 สรุปผลงานวิจัย.....	100
6.2 ข้อเสนอแนะและแนวทางในการพัฒนาต่อ.....	101
บรรณานุกรม.....	102
ประวัติผู้เขียน.....	105

สารบัญตาราง

หน้า

ตารางที่ 1 ความสัมพันธ์ของการใช้งานโปรแกรมกับประเภทของโปรแกรมการตอบสนองด้านโหลด	30
ตารางที่ 2 รายละเอียดเบื้องต้นของโหนดต้นเสมือนและโหนดปลายเสมือน	68
ตารางที่ 3 ข้อกำหนดหลักเกณฑ์การทดสอบตามมาตรฐาน IEC 62053-21 Class 1	78



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สารบัญรูปภาพ

	หน้า
รูปที่ 1 ภาพรวมของระบบมิเตอร์ขั้นสูง.....	8
รูปที่ 2 ตัวอย่างมิเตอร์แบบต่าง ๆ	9
รูปที่ 3 ตัวอย่างรูปแบบการตอบสนองด้านโหลดแบบต่าง ๆ [7].....	15
รูปที่ 4 โปรแกรมมาตรการตอบสนองด้านโหลดที่เกี่ยวข้องต่อการบริหารจัดการไฟฟ้า (Demand Response Spectrum) [7-9].....	17
รูปที่ 5 ตัวอย่างประโยชน์ของการตอบสนองด้านโหลด [5]	18
รูปที่ 6 การดำเนินการตอบสนองด้านโหลดแบบควบคุมด้วยเจ้าหน้าที่ แบบกึ่งอัตโนมัติ และแบบอัตโนมัติโดยสมบูรณ์ ตามลำดับ	20
รูปที่ 7 ภาพรวมการทำงานของซอฟต์แวร์ OpenADR	21
รูปที่ 8 โครงสร้างรูปแบบตรงแบบที่ 1 (Direct 1).....	23
รูปที่ 9 โครงสร้างรูปแบบตรงแบบที่ 2 (Direct 2).....	24
รูปที่ 10 โครงสร้างรูปแบบตรงแบบที่ 3 (Direct 3).....	25
รูปที่ 11 โครงสร้างรูปแบบตรงแบบที่ 4 (Direct 4).....	26
รูปที่ 12 โครงสร้างรูปแบบผู้ให้บริการ (Facilitator).....	27
รูปที่ 13 โครงสร้างรูปแบบผู้รวบรวมข้อมูล (Aggregator).....	28
รูปที่ 14 การประยุกต์ใช้บริการ NETPIE เข้ากับอินเทอร์เน็ตและสรรพสิ่งรูปแบบต่าง ๆ.....	31
รูปที่ 15 โมเดลการสื่อสารของโพรโทคอล MQTT	32
รูปที่ 16 ค่าความสัมพันธ์ของกำลังไฟฟ้าในระบบไฟฟ้ากระแสสลับ	36
รูปที่ 17 แผนภาพบล็อกฮาร์ดแวร์ของมิเตอร์อัจฉริยะต้นแบบ	38
รูปที่ 18 โมดูล ESP32-WROOM-32	39
รูปที่ 19 ไอซีวัดค่าพลังงานทางไฟฟ้า ADE9153A ที่มีฟังก์ชันสอบเทียบตัวเอง	41

รูปที่ 20 ตัวอย่างบอร์ดประเมินของไอซีวัดค่าพลังงานทางไฟฟ้า ADE9153A (EV-ADE9153ASHIELDZ)	42
รูปที่ 21 แสดงแผนภาพบล็อกเส้นทางข้อมูลกระแสของช่องสัญญาณที่หนึ่ง (Current Channel A Datapath).....	42
รูปที่ 22 แสดงแผนภาพบล็อกเส้นทางข้อมูลกระแสของช่องสัญญาณที่สอง (Current Channel B Datapath).....	43
รูปที่ 23 แสดงแผนภาพบล็อกเส้นทางข้อมูลแรงดัน (Voltage Channel Datapath).....	44
รูปที่ 24 โหมดพลังงานของการสอบเทียบตัวเอง (mSure Autocalibration Power Modes to Same Certainty)	46
รูปที่ 25 ความเร็วในการลู่เข้าของการสอบเทียบตัวเอง (ช่องสัญญาณกระแสในโหมดพลังงานต่ำ) ขึ้นอยู่กับค่าตัวต้านทานชนิด	46
รูปที่ 26 ความเร็วในการลู่เข้าของการสอบเทียบตัวเอง (ช่องสัญญาณกระแสในโหมดปกติ) ขึ้นอยู่กับค่าตัวต้านทานชนิด	46
รูปที่ 27 ความเร็วในการลู่เข้าของการสอบเทียบตัวเอง (ช่องสัญญาณแรงดัน) ขึ้นอยู่กับอัตราส่วนของวงจรแบ่งแรงดัน	47
รูปที่ 28 ความสัมพันธ์ระหว่างค่าคงตัวการแปลงผันที่วัดได้กับค่าคงตัวการแปลงผันเป้าหมาย	49
รูปที่ 29 บอร์ดทรานสเบอร์รี่พาย 4	50
รูปที่ 30 ตัวอย่างโมดูลอ่าน/เขียน Micro SD Card.....	51
รูปที่ 31 ฮาร์ดแวร์ต้นแบบของไมโครคอนโทรลเลอร์อะดัจฉริยะเฟสเดียวที่มีฟังก์ชันสอบเทียบตัวเอง.....	52
รูปที่ 32 ส่วนประกอบฮาร์ดแวร์ต้นแบบของไมโครคอนโทรลเลอร์อะดัจฉริยะ	53
รูปที่ 33 แผนภาพเค้าร่างของวงจร ADE9153A และอุปกรณ์รอบข้าง.....	53
รูปที่ 34 ลักษณะการทำงานของการทำงานของไมโครคอนโทรลเลอร์อะดัจฉริยะที่สอบเทียบตัวเอง.....	55
รูปที่ 35 แผนภูมิสายงานภารกิจวัดค่าพลังงานทางไฟฟ้า.....	56
รูปที่ 36 แผนภูมิสายงานภารกิจสอบเทียบตัวเอง	57
รูปที่ 37 แผนภูมิสายงานภารกิจส่งข้อมูลขึ้นคลาวด์	57
รูปที่ 38 แผนภูมิสายงานภารกิจเชื่อมต่ออินเทอร์เน็ตแบบไร้สาย.....	58

รูปที่ 39 แผนภูมิสายงานภารกิจจัดเก็บข้อมูลลงฐานข้อมูล	59
รูปที่ 40 ตัวอย่าง Device Schema ในรูปแบบ JSON.....	60
รูปที่ 41 โครงสร้างสถาปัตยกรรมการสื่อสารโพรโทคอล OpenADR	62
รูปที่ 42 ตัวอย่างการดำเนินการบริการด้วยวิธี PUSH.....	62
รูปที่ 43 ตัวอย่างการดำเนินการบริการด้วยวิธี PULL.....	63
รูปที่ 44 การดำเนินการบริการลงทะเบียนระหว่างโหนดปลายเสมือนและโหนดต้นเสมือน	64
รูปที่ 45 การดำเนินการบริการรายงานระหว่างโหนดต้นทางและโหนดปลายทาง.....	65
รูปที่ 46 การดำเนินการบริการตรวจสอบสถานะระหว่างโหนดปลายเสมือนและโหนดต้นเสมือน	66
รูปที่ 47 การดำเนินการบริการเข้าร่วมมาตรการระหว่างโหนดปลายเสมือนและโหนดต้นเสมือน.....	67
รูปที่ 48 แสดงความสัมพันธ์ของคลาส VENManager EventManager และ IEventService	71
รูปที่ 49 แสดงฟังก์ชันที่เกี่ยวข้องในไฟล์หลัก (main).....	73
รูปที่ 50 แสดงกระบวนการสร้างคลังโปรแกรมโหนดปลายเสมือน	74
รูปที่ 51 การทดสอบวัดค่าพลังงานทางไฟฟ้าของมิเตอร์อัจฉริยะต้นแบบกับโหลด	75
รูปที่ 52 การทดสอบวัดค่าพลังงานทางไฟฟ้าของมิเตอร์อัจฉริยะต้นแบบเมื่อ “ปิด” การใช้งานโหลด	76
รูปที่ 53 การทดสอบวัดค่าพลังงานทางไฟฟ้าของมิเตอร์อัจฉริยะต้นแบบเมื่อ “เปิด” การใช้งานโหลด	76
รูปที่ 54 เครื่องมือที่ใช้ในการทดสอบความความผิดพลาดการวัดของมิเตอร์อัจฉริยะ	77
รูปที่ 55 ผลการทดสอบที่ค่ากระแสไฟฟ้า $0.05I_n \leq I \leq 0.1I_n$ ตัวประกอบกำลังเท่ากับ 1	78
รูปที่ 56 ผลการทดสอบที่ค่ากระแสไฟฟ้า $0.1I_n \leq I \leq I_{max}$ ตัวประกอบกำลังเท่ากับ 1	79
รูปที่ 57 ผลการทดสอบที่ค่ากระแสไฟฟ้า $0.1I_n \leq I \leq 0.2I_n$ ตัวประกอบกำลังเท่ากับ 0.5 ล้าหลัง...79	
รูปที่ 58 ผลการทดสอบที่ค่ากระแสไฟฟ้า $0.1I_n \leq I \leq 0.2I_n$ ตัวประกอบกำลังเท่ากับ 0.8 นำหน้า..80	
รูปที่ 59 ผลการทดสอบที่ค่ากระแสไฟฟ้า $0.2I_n \leq I \leq I_{max}$ ตัวประกอบกำลังเท่ากับ 0.5 ล้าหลัง.....80	
รูปที่ 60 ผลการทดสอบที่ค่ากระแสไฟฟ้า $0.2I_n \leq I \leq I_{max}$ ตัวประกอบกำลังเท่ากับ 0.8 นำหน้า....81	
รูปที่ 61 ผลการทดสอบที่ค่ากระแสไฟฟ้า $0.2I_n \leq I \leq I_n$ ตัวประกอบกำลังเท่ากับ 0.25 ล้าหลัง.....81	

รูปที่ 62 ผลการทดสอบที่ค่ากระแสไฟฟ้า $0.2I_n \leq I \leq I_n$ ตัวประกอบกำลังเท่ากับ 0.5 หน้า82	82
รูปที่ 63 การทดสอบส่งข้อมูลค่าพลังงานทางไฟฟ้าไปยัง NETPIE Dashboard เพื่อแสดงผล83	83
รูปที่ 64 การทดสอบการสอบเทียบตัวเองของมิเตอร์อัจฉริยะต้นแบบที่ถูกพัฒนาขึ้น84	84
รูปที่ 65 ตัวอย่างการจัดเก็บข้อมูลลงในฐานข้อมูลด้วย SQLite85	85
รูปที่ 66 หน้าต่างเข้าสู่ระบบโหนดต้นเสมือน86	86
รูปที่ 67 สร้างโหนดปลายเสมือนเพื่อลงทะเบียนเข้าสู่ระบบ86	86
รูปที่ 68 ตั้งชื่อโหนดปลายเสมือนเพื่อลงทะเบียนเข้าสู่ระบบ87	87
รูปที่ 69 ตั้งค่ารูปแบบโปรไฟล์ OpenADR 2.0b Specification สำหรับโหนดปลายเสมือน87	87
รูปที่ 70 รายชื่อและสถานะของโหนดปลายเสมือนที่ได้ลงทะเบียนเข้าสู่ระบบเรียบร้อยแล้ว88	88
รูปที่ 71 สร้างเหตุการณ์สำหรับการตอบสนองด้านโหลด89	89
รูปที่ 72 กรอกรายละเอียดเพื่อสร้างเหตุการณ์สำหรับการตอบสนองด้านโหลด90	90
รูปที่ 73 เลือกเป้าหมายของเหตุการณ์สำหรับการตอบสนองด้านโหลด90	90
รูปที่ 74 เผยแพร่เหตุการณ์ไปยังโหนดปลายเสมือน91	91
รูปที่ 75 แสดงสถานะเบื้องต้นของเหตุการณ์91	91
รูปที่ 76 ผลลัพธ์ที่คาดหวังจากโปรแกรมทดสอบการโต้ตอบ testoadr92	92
รูปที่ 77 แสดงการโต้ตอบของชุดข้อมูล oadrCreatePartyRegistration93	93
รูปที่ 78 แสดงการโต้ตอบของชุดข้อมูล oadrCreatedPartyRegistration93	93
รูปที่ 79 แสดงการโต้ตอบของชุดข้อมูล oadrRegisterReport94	94
รูปที่ 80 แสดงการโต้ตอบของชุดข้อมูล oadrRegisteredReport94	94
รูปที่ 81 แสดงการโต้ตอบของชุดข้อมูล oadrRequestEvent95	95
รูปที่ 82 แสดงการโต้ตอบของชุดข้อมูล oadrDistributeEvent95	95
รูปที่ 83 แสดงการโต้ตอบของชุดข้อมูล oadrCreatedEvent96	96
รูปที่ 84 แสดงการโต้ตอบของชุดข้อมูล oadrResponse96	96
รูปที่ 85 แสดงการโต้ตอบของชุดข้อมูล oadrPoll97	97

รูปที่ 86 แสดงการโต้ตอบเมื่อได้รับเหตุการณ์จากโหนดต้นเสมือนด้วย oadrDistributeEvent.....98

รูปที่ 87 แสดงการโต้ตอบเมื่อตอบกลับการเข้าร่วมมาตรการด้วย oadrCreatedEvent98

รูปที่ 88 แสดงการทำงานเมื่อถึงเวลาของเหตุการณ์การตอบสนองด้านโพลิตกรณีเข้าร่วมมาตรการ 99

รูปที่ 89 แสดงการทำงานเมื่อสิ้นสุดเหตุการณ์การตอบสนองด้านโพลิตกรณี99

รูปที่ 90 แสดงการทำงานเมื่อถึงเวลาของเหตุการณ์การตอบสนองด้านโพลิตกรณีไม่เข้าร่วมมาตรการ
.....99



บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

มิเตอร์อัจฉริยะ (Smart Meter; SM) เป็นอุปกรณ์ที่มีบทบาทสำคัญเป็นอย่างมากในปัจจุบัน เนื่องจากโครงข่ายไฟฟ้าอัจฉริยะ (Smart Grid; SG) นั้นมีการเติบโตและพัฒนาอยู่ตลอดเวลา มิเตอร์อัจฉริยะจึงเข้ามามีบทบาทแทนที่มิเตอร์แบบจานหมุนมากขึ้น จากในอดีตโครงข่ายไฟฟ้าอัจฉริยะใช้ระบบอ่านหน่วยมิเตอร์ไฟฟ้าแบบอัตโนมัติ (Automatic Meter Reading; AMR) ซึ่งเป็นระบบสำหรับอ่านค่าพลังงานทางไฟฟ้าแบบอัตโนมัติจากมิเตอร์อิเล็กทรอนิกส์ทั้งแบบสายสัญญาณและแบบไร้สาย ข้อมูลที่อ่านได้นั้นจะถูกส่งต่อไปยังระบบฐานข้อมูลกลาง (AMR Data Center) เพื่อจัดเก็บข้อมูลการใช้ไฟฟ้าของผู้ใช้ไฟฟ้า รวมถึงการจัดทำใบแจ้งค่าไฟฟ้าแบบเวลาจริง (Real time Billing) นอกจากนี้ระบบ AMR สามารถพัฒนาระบบบริหารจัดการข้อมูล (Meter Data Management System; MDMS) ซึ่งมีหน้าที่หลักคือนำข้อมูลที่ถูกบันทึกอยู่ในระบบฐานข้อมูลกลางมาบริหารจัดการอย่างเป็นสัดส่วน เพื่อนำไปใช้ในการวิเคราะห์ต่าง ๆ ได้อย่างมีประสิทธิภาพ

โครงสร้างพื้นฐานระบบมิเตอร์ขั้นสูง (Advanced Metering Infrastructure; AMI) คือระบบที่พัฒนาต่อยอดมาจากระบบ AMR โดยมีมิเตอร์อัจฉริยะเป็นองค์ประกอบหลัก ไม่เพียงแค่อ่านหน่วยไฟฟ้าอัตโนมัติได้เท่านั้น โครงสร้างพื้นฐานระบบมิเตอร์ขั้นสูงนี้ยังมีความสามารถในการติดต่อสื่อสารสองทาง (Two-way communication) ระหว่างผู้ให้บริการไฟฟ้ากับผู้ไฟฟ้า ทำให้สามารถส่งสัญญาณการควบคุมไฟฟ้าไปยังผู้ไฟฟ้าผ่านสัญญาณทางไกลเพื่อให้เกิดการลดการใช้พลังงาน อีกทั้งยังสามารถตรวจจับความผิดปกติที่เกิดขึ้นภายในโครงข่ายไฟฟ้าเพื่อป้องกันไม่ให้เกิดไฟฟ้าดับ (Outage) รวมถึงยังสามารถส่งสัญญาณบอกผู้ไฟฟ้าในการปรับเปลี่ยนพฤติกรรมการใช้ไฟให้สอดคล้องกับการตอบสนองด้านโหลด (Demand Response; DR) เพื่อเพิ่มศักยภาพในการลดความต้องการไฟฟ้าสูงสุด (Peak Demand) ซึ่งจะเป็นประโยชน์ต่อการบริหารจัดการสภาวะวิกฤตด้านพลังงานไฟฟ้า ทำให้การกระจายพลังงานไฟฟ้าเป็นไปได้อย่างมีประสิทธิภาพและมีความมั่นคงต่อเนื่อง โดยภาพรวมแล้วโครงสร้างพื้นฐานระบบมิเตอร์ขั้นสูงนี้จะช่วยให้โครงข่ายไฟฟ้าอัจฉริยะนั้นมีความชาญฉลาดมากขึ้น เสถียรภาพที่ดีขึ้น อีกทั้งยังเพิ่มความน่าเชื่อถือให้กับโครงข่ายไฟฟ้ามากขึ้นอีกด้วย

แม้ว่ามีมิเตอร์อัจฉริยะที่ใช้งานในปัจจุบันจะรองรับโครงข่ายไฟฟ้าอัจฉริยะซึ่งทำให้มิเตอร์สามารถในการสื่อสารกับศูนย์ควบคุม หรือความสามารถในการเก็บข้อมูลและการแจ้งเตือนไปยังศูนย์ควบคุมเมื่อเกิดเหตุการณ์ผิดปกติโดยอัตโนมัติ แต่หากมิเตอร์อัจฉริยะเหล่านี้ถูกใช้งานไประยะเวลาหนึ่ง ค่าพลังงานทางไฟฟ้าต่าง ๆ ที่วัดได้จากมิเตอร์อัจฉริยะอาจมีความคลาดเคลื่อนไปจากความจริงไม่มากนักน้อย อาจส่งผลให้ผู้ให้บริการไฟฟ้าวิเคราะห์การใช้พลังงานไฟฟ้าได้ไม่ถูกต้อง ทำให้ผู้ให้บริการไฟฟ้าสูญเสียรายได้ที่ควรจะเป็น หรืออาจก่อให้เกิดความผิดพลาดที่ตามมาในหลาย ๆ ด้าน วิธีการแก้ไขปัญหาดังกล่าวนี้ผู้ให้บริการมีความจำเป็นที่จะต้องส่งเจ้าหน้าที่เพื่อเข้าไปทำการตรวจสอบซ่อมบำรุง พร้อมทั้งทำการสอบเทียบมาตรฐาน (Calibration) ใหม่เพื่อให้มิเตอร์อัจฉริยะมีความแม่นยำดังเดิม ซึ่งทำให้เกิดภาระค่าใช้จ่ายที่ตามมา อาทิ การว่าจ้างเจ้าหน้าที่ประจำตำแหน่งสำหรับดูแลรักษาและสอบเทียบมาตรฐานมิเตอร์อัจฉริยะ ค่าเดินทางสำหรับการส่งเจ้าหน้าที่เพื่อเข้าไปตรวจสอบมิเตอร์อัจฉริยะในพื้นที่ ๆ กำหนด

ดังนั้นผู้วิจัยจึงเล็งเห็นและสนใจถึงปัญหาข้างต้นที่กล่าวไป ผู้วิจัยจึงเริ่มศึกษาค้นคว้าเกี่ยวกับมิเตอร์อัจฉริยะแบบต่าง ๆ ที่ผ่านมาในอดีต รวมถึงไอซีมิเตอร์ที่มีความสามารถพิเศษอย่างการสอบเทียบตัวเอง (Autocalibration) อีกทั้งยังศึกษามาตรฐานการสื่อสารรูปแบบต่าง ๆ ที่เหมาะสมกับมิเตอร์อัจฉริยะในปัจจุบัน ผู้วิจัยจึงสนใจที่จะพัฒนามิเตอร์อัจฉริยะพร้อมฟังก์ชันในการสื่อสารโพรโทคอลที่เกี่ยวข้องกับการตอบสนองด้านโหลด เพื่อที่จะช่วยพัฒนาโครงข่ายไฟฟ้าอัจฉริยะให้มีเสถียรภาพที่ดีขึ้น นอกจากนี้จะสามารถช่วยปรับเปลี่ยนพฤติกรรมการใช้ไฟฟ้าในช่วงพีคด้วยการตอบสนองด้านโหลดแล้ว ผู้ให้บริการไฟฟ้าจะสามารถทำงานได้อย่างมีประสิทธิภาพที่ดีมากขึ้น เนื่องจากผู้ให้บริการสามารถทำการสอบเทียบตัวเองของมิเตอร์อัจฉริยะได้อย่างอัตโนมัติ ทำให้ลดความจำเป็นในการส่งเจ้าหน้าที่เข้าไปยังพื้นที่เพื่อทำการสอบเทียบ ซึ่งจะช่วยประหยัดเวลาและลดภาระค่าใช้จ่ายในการเดินทางได้ กล่าวโดยรวมแล้วผู้วิจัยจึงมีความตั้งใจที่จะพัฒนาอุปกรณ์ต้นแบบมิเตอร์อัจฉริยะที่มีฟังก์ชันสอบเทียบตัวเองและสามารถตอบสนองด้านโหลดได้อย่างอัตโนมัติ ซึ่งจะอธิบายรายละเอียดในวิทยานิพนธ์ฉบับนี้ต่อไป

1.2 วัตถุประสงค์ของวิทยานิพนธ์

- 1.2.1 พัฒนามิเตอร์อัจฉริยะที่มีฟังก์ชันสอบเทียบตัวเอง (Autocalibration)
- 1.2.2 พัฒนาซอฟต์แวร์มิเตอร์อัจฉริยะที่มีความสามารถในการตอบสนองด้านโหลดอัตโนมัติ โดยสื่อสารตามโพรโทคอล Open Automated Demand Response (OpenADR)

1.3 ขอบเขตของวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้สามารถแบ่งออกได้เป็นสองส่วนหลักคือทางภาคฮาร์ดแวร์ และภาคซอฟต์แวร์

1.3.1 ภาคฮาร์ดแวร์

- อุปกรณ์รวบรวมข้อมูลวัดค่าทางไฟฟ้าโดยใช้ไมโครคอนโทรลเลอร์เป็นหน่วยประมวลผลหลัก
- มิเตอร์อัจฉริยะต้นแบบโดยใช้ไอซีสำเร็จรูป ADE9153A ซึ่งสามารถอ่านค่าพลังงานทางไฟฟ้าเฟสเดียว ได้ดังนี้
 - ค่าแรงดันไฟฟ้า (Voltage)
 - ค่ากระแสไฟฟ้า (Current)
 - ค่ากำลังไฟฟ้าจริง (Active Power)
 - ค่ากำลังไฟฟารีแอกทีฟ (Reactive Power)
 - ค่ากำลังไฟฟ้าปรากฏ (Apparent Power)
 - ค่าตัวประกอบกำลังไฟฟ้า (Power Factor) และ
 - ค่าความถี่ทางไฟฟ้า (Frequency)
- มิเตอร์อัจฉริยะรองรับแรงดันพิกัด 180 V ถึง 265 V และกระแสสูงสุดที่ 10 A ที่ความถี่ 50 Hz
- มีความแม่นยำในการวัดพลังงานไฟฟ้าทำงานตามมาตรฐาน IEC 62053-21
- สามารถจัดเก็บฐานข้อมูลภายในตัวเองได้ทุก ๆ 1 - 15 นาที สูงสุด 1,000 ระเบียบ
- สามารถสื่อสารผ่าน Wi-Fi

1.3.2 ภาคซอฟต์แวร์

- มิเตอร์อัจฉริยะสามารถสอบเทียบตัวเองได้โดยไม่ต้องใช้แหล่งจ่ายอ้างอิงหรือมิเตอร์อ้างอิงในการสอบเทียบ
- มิเตอร์อัจฉริยะสามารถส่งข้อมูลค่าพลังงานทางไฟฟ้าต่าง ๆ ให้แสดงผลข้อมูลบนแผงหน้าปัดเน็ตพาย (NETPIE Dashboard)
- มิเตอร์อัจฉริยะสามารถสื่อสารกับตัวรวบรวมโหลด (Load Aggregator) ด้วยโพรโทคอล OpenADR
- มิเตอร์อัจฉริยะสามารถควบคุมการทำงานของเครื่องใช้ไฟฟ้า ให้สอดคล้องกับสถานะทางไฟฟ้าภายในโครงข่ายไฟฟ้าอัจฉริยะในช่วงที่มีการตอบสนองด้านโหลดได้

1.3.3 การทดสอบมิเตอร์อัจฉริยะ

- ทดสอบความผิดพลาดการวัดค่าพลังงานทางไฟฟ้าของมิเตอร์อัจฉริยะ
- ทดสอบการส่งข้อมูลพลังงานทางไฟฟ้าไปยัง NETPIE Dashboard
- ทดสอบการสอบเทียบตัวเองของมิเตอร์อัจฉริยะ
- ทดสอบการจัดเก็บข้อมูลลงในฐานข้อมูล
- ทดสอบซอฟต์แวร์การตอบสนองด้านโหลด OpenADR

1.4 ขั้นตอนการดำเนินงานวิทยานิพนธ์

1.4.1 ศึกษาปัญหาของมิเตอร์อัจฉริยะที่มีใช้อยู่ในปัจจุบัน

1.4.2 ศึกษาอุปกรณ์และซอฟต์แวร์ที่เกี่ยวข้องสำหรับพัฒนามิเตอร์อัจฉริยะต้นแบบ รวมถึงรูปแบบการสื่อสารแบบต่าง ๆ ได้แก่

- การใช้วงจรรวมสำหรับการวัดค่าพลังงานทางไฟฟ้า
- การสอบเทียบตัวเอง
- การรับส่งข้อมูลของบริการ NETPIE
- การเก็บข้อมูลฐานข้อมูลด้วย SQLite
- โพรโทคอลการตอบสนองด้านโหลด OpenADR
- การควบคุมโหลด

1.4.3 ศึกษางานวิจัยและมาตรฐานที่เกี่ยวข้อง

1.4.4 ทดลองประกอบวงจรที่มาจากบอร์ดประเมิน (Evaluation Board) เป็นส่วนใหญ่

1.4.5 พัฒนาซอฟต์แวร์ให้มิเตอร์อัจฉริยะทำตามขอบเขตที่กำหนดไว้

1.4.6 ทดสอบการทำงานของมิเตอร์อัจฉริยะ

1.4.7 สรุปผลการทดลอง

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 กระบวนการออกแบบมิเตอร์อัจฉริยะเฟสเดียวที่มีฟังก์ชันสอบเทียบตัวเอง

1.5.2 มิเตอร์อัจฉริยะที่มีฟังก์ชันสอบเทียบตัวเองที่มีความแม่นยำเป็นไปตามมาตรฐาน IEC 62053-21 Class 1

1.5.3 กระบวนการติดตั้งซอฟต์แวร์ OpenADR ทั้งในส่วนของโหนดต้นเสมือน (VTN) และโหนดปลายเสมือน (VEN) พร้อมทั้งวิธีการตั้งค่าเพื่อใช้งาน

1.5.4 ซอฟต์แวร์การตอบสนองด้านโหลด OpenADR ที่พร้อมประยุกต์ใช้งาน

1.6 โครงสร้างของวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้จะแบ่งเนื้อหาออกเป็น 6 ส่วนโดย

บทที่ 1 จะกล่าวถึงที่มาและความสำคัญของปัญหา วัตถุประสงค์ ขอบเขตของวิทยานิพนธ์ ขั้นตอนการดำเนินงานวิทยานิพนธ์ ประโยชน์ที่คาดว่าจะได้รับ รวมไปถึงโครงสร้างของวิทยานิพนธ์

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง จะกล่าวถึงทฤษฎีที่เกี่ยวข้องกับวิทยานิพนธ์โดยรวม ได้แก่ โครงสร้างพื้นฐานระบบมิเตอร์ขั้นสูงในโครงข่ายไฟฟ้าอัจฉริยะ การตอบสนองด้านโหลด ทฤษฎีซอฟต์แวร์ที่เกี่ยวข้องต่าง ๆ และหลักการคำนวณกำลังไฟฟ้าพื้นฐาน

บทที่ 3 การออกแบบฮาร์ดแวร์ของมิเตอร์อัจฉริยะ จะกล่าวถึงรายละเอียดอุปกรณ์ต่าง ๆ ที่ใช้ในมิเตอร์อัจฉริยะ รวมถึงฮาร์ดแวร์มิเตอร์อัจฉริยะต้นแบบที่ถูกพัฒนาขึ้น

บทที่ 4 การออกแบบและพัฒนาซอฟต์แวร์ของมิเตอร์อัจฉริยะ จะกล่าวถึงซอฟต์แวร์ที่พัฒนาสำหรับมิเตอร์อัจฉริยะ ซอฟต์แวร์ไอโอทีคลาวด์แพลตฟอร์ม รวมถึงซอฟต์แวร์การตอบสนองด้านโหลด OpenADR

บทที่ 5 การทดลองและผลการทดลอง จะกล่าวถึงการทดสอบวัดค่าพลังงานทางไฟฟ้าจากการใช้งานจริง การทดสอบความผิดพลาดการวัดค่าพลังงานทางไฟฟ้าของมิเตอร์ การทดสอบการส่งข้อมูลพลังงานทางไฟฟ้าไปยัง NETPIE Dashboard การทดสอบการสอบเทียบตัวเองของมิเตอร์อัจฉริยะต้นแบบ การทดสอบการจัดเก็บข้อมูลลงฐานข้อมูลด้วย SQLite ไปจนถึงการทดสอบซอฟต์แวร์การตอบสนองด้านโหลด OpenADR ซึ่งจะสามารถแบ่งได้เป็น 2 ส่วนคือ การทดสอบโหนดต้นเสมือน และการทดสอบโหนดปลายเสมือน

บทที่ 6 สรุปผลงานวิจัยและข้อเสนอแนะ จะกล่าวถึงการสรุปผลงานวิจัย รวมถึงข้อเสนอแนะและแนวทางในการพัฒนาต่อของงานวิจัยฉบับนี้

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 โครงสร้างพื้นฐานระบบมิเตอร์ขั้นสูงในโครงข่ายไฟฟ้าอัจฉริยะ

ในยุคเริ่มต้นของโครงข่ายไฟฟ้าอัจฉริยะนั้น เทคโนโลยีอ่านหน่วยมิเตอร์ไฟฟ้าแบบอัตโนมัติ (AMR) ถูกพัฒนาขึ้น มีจุดประสงค์มุ่งเน้นในการทดแทนการจดหน่วยแบบเดิม ทำให้สามารถเพิ่มความแม่นยำในการจดหน่วยด้วยการอ่านหน่วยมิเตอร์ไฟฟ้า เพิ่มความเร็วและง่ายต่อการเข้าถึงข้อมูลมิเตอร์จากระยะไกลผ่านโครงข่ายการสื่อสาร อีกทั้งยังสามารถลดต้นทุนที่เกี่ยวข้องกับการส่งเจ้าหน้าที่เข้าไปยังพื้นที่เพื่อจดหน่วยมิเตอร์ได้ ระบบอ่านหน่วยมิเตอร์ไฟฟ้าแบบอัตโนมัติถือเป็นจุดเริ่มต้นที่ริเริ่มในการนำมิเตอร์อิเล็กทรอนิกส์ (Electronic Meter) เข้ามาใช้แทนที่มีเตอร์แบบจานหมุน (Watt-Hour Meter) ซึ่งมีมิเตอร์แบบจานหมุนนั้นมีเพียงหน้าที่เดียวคือวัดหน่วยการใช้ไฟฟ้า โดยหลักการของมิเตอร์ชนิดนี้คือเมื่อมีการใช้พลังงานไฟฟ้า ขดลวดกระแสและขดลวดแรงดันจะมีการเหนี่ยวนำทำให้เกิดสนามแม่เหล็กขึ้น ส่งผลให้เกิดกระแสไหลวน (Eddy Current) ภายในจานอะลูมิเนียมจนทำให้เกิดแรงบิดและเคลื่อนที่ไปตามทิศทางของแรงบิดที่เกิดขึ้น โดยแกนกลางของจานอะลูมิเนียมจะมีฟันเฟืองที่ทำหน้าที่ส่งแรงไปขับตัวเลขที่หน้าปัดของเครื่องวัดทำให้มาตรวัดมีการเปลี่ยนแปลงค่า ซึ่งแรงขับที่เกิดขึ้นจะเป็นสัดส่วนระหว่างความเข้มของสนามแม่เหล็กที่เกิดขึ้นระหว่างขดลวดกระแสและขดลวดแรงดัน จะเห็นได้ว่ามิเตอร์แบบจานหมุนนี้มีกลไกการทำงานเชิงกลเมื่อถูกใช้งานไประยะเวลาหนึ่งอาจทำให้เกิดความคลาดเคลื่อนในการวัดค่าพลังงานไฟฟ้า นอกจากนี้ยังมีข้อเสียด้านอื่น ๆ อีกเช่น มิเตอร์แบบจานหมุนนี้ไม่สามารถตรวจจับการรบกวนมิเตอร์ทำให้เกิดการโกงค่าไฟได้ง่าย หรือความผิดพลาดในการจดหน่วยไฟฟ้าของเจ้าหน้าที่จดหน่วยเอง เป็นต้น

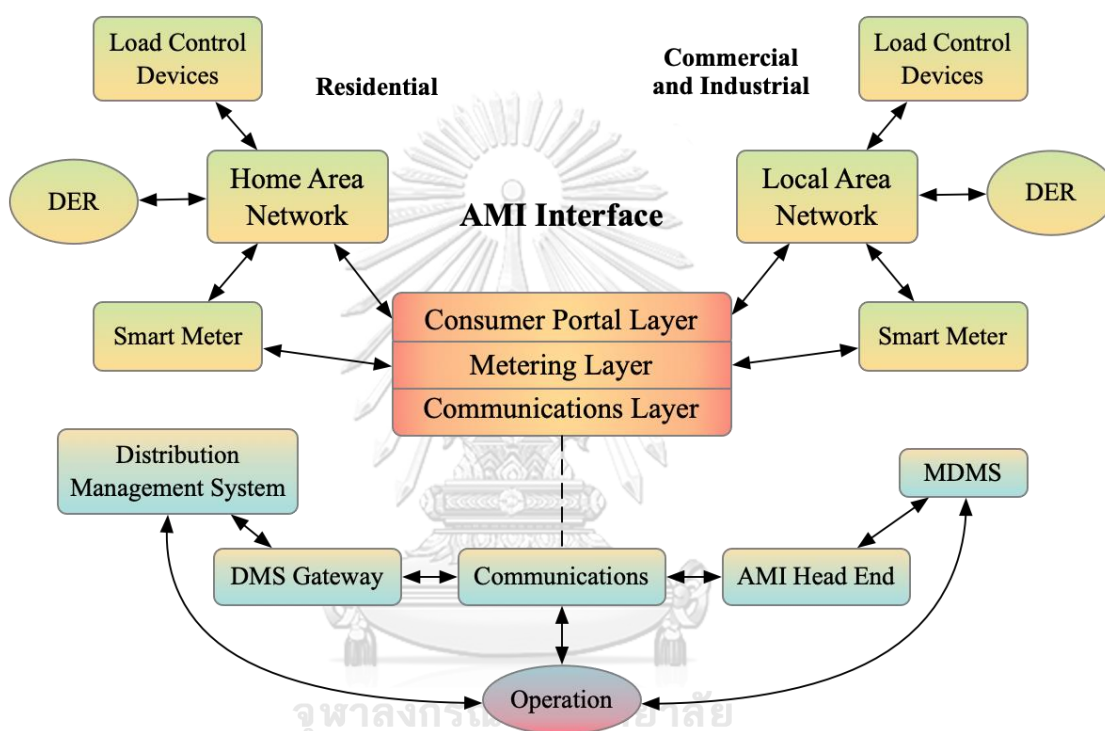
เมื่อยุคสมัยเปลี่ยนไป เทคโนโลยีมีการพัฒนาให้มีความทันสมัยมากขึ้น มิเตอร์แบบอิเล็กทรอนิกส์จึงถูกนำมาใช้อย่างแพร่หลายทั่วโลก แน่แน่นอนว่ามิเตอร์แบบอิเล็กทรอนิกส์มีศักยภาพที่ดีกว่ามิเตอร์แบบจานหมุนในหลาย ๆ ด้านด้วยกัน มิเตอร์อิเล็กทรอนิกส์นั้นมีหลักการวิเคราะห์การใช้พลังงานไฟฟ้าที่แตกต่างจากมิเตอร์แบบจานหมุนซึ่งจะไม่มีส่วนประกอบที่เป็นกลไกเชิงกลแล้ว แต่จะเปลี่ยนมาใช้วงจรรวม (Integrated Circuit; IC) ในการประมวลผลค่าทางไฟฟ้าแทน โดยอินพุตที่รับสัญญาณทางไฟฟ้าเข้ามานั้นจะรับค่ากระแสและแรงดันเข้าไปยังวงจรมิเตอร์ผ่านการตรวจวัดจากตัวรับรู้ (Sensor) ในรูปแบบสัญญาณแอนะล็อก (Analog Signal) เช่น หม้อแปลงกระแสไฟฟ้า (Current Transformer; CT) หม้อแปลงแรงดัน (Potential Transformer; PT) ตัวต้านทานชนิด

(Shunt Resistor; R-Shunt) เป็นต้น จากนั้นจะส่งค่าทางไฟฟ้าที่วัดได้ไปยังวงจรรวมแปลงค่าจากสัญญาณแอนะล็อกไปเป็นสัญญาณดิจิทัล (Digital Signal) เพื่อเตรียมความพร้อมให้กับชุดข้อมูลทางไฟฟ้า ข้อมูลที่ได้จะถูกส่งไปยังวงจรรวมคำนวณเพื่อคำนวณค่าพลังงานทางไฟฟ้า อาทิ ค่าแรงดันอาร์เอ็มเอส (RMS Voltage) ค่ากระแสอาร์เอ็มเอส (RMS Current) ค่ากำลังไฟฟ้าจริง (Active Power) ค่ากำลังไฟฟ้ารีแอคทีฟ (Reactive Power) ค่ากำลังไฟฟ้าปรากฏ (Apparent Power) ค่าตัวประกอบกำลังไฟฟ้า (Power Factor) และอื่น ๆ หลังจากที่ได้รับชุดข้อมูลพลังงานทางไฟฟ้าถูกคำนวณเรียบร้อยแล้ว ชุดข้อมูลที่ได้ก็จะพร้อมใช้งานโดยผู้ให้บริการไฟฟ้าสามารถเรียกดูข้อมูลต่าง ๆ ผ่านโครงการสื่อสารเพื่อนำไปใช้ประโยชน์ในการวิเคราะห์ค่าทางไฟฟ้าได้ต่อไป

ระบบอ่านหน่วยมิเตอร์ไฟฟ้าแบบอัตโนมัติ (AMR) นั้นเป็นระบบที่ทำหน้าที่อ่านค่าพลังงานทางไฟฟ้า ซึ่งมีลักษณะเป็นการสื่อสารเพียงทางเดียว (One-way communication) เท่านั้น หมายความว่าผู้ให้บริการไฟฟ้าจะสามารถเข้าถึงข้อมูลการใช้พลังงานไฟฟ้าของผู้ใช้ไฟฟ้าได้เพียงอย่างเดียว ไม่สามารถส่งข้อความเพื่อสื่อสารกับผู้ให้บริการไฟฟ้า สั่งการ หรือควบคุมโดยตรงได้ โดยลักษณะการทำงานของระบบอ่านหน่วยมิเตอร์ไฟฟ้าอัตโนมัตินั้นมิเตอร์อิเล็กทรอนิกส์ที่เชื่อมต่อกับระบบทุกตัวจะทำการอ่านข้อมูลพลังงานทางไฟฟ้าต่าง ๆ แบบอัตโนมัติและจะทำการส่งข้อมูลผ่านระบบการสื่อสารไปยังระบบฐานข้อมูลกลาง (AMR data center) เพื่อรวบรวมและจัดเก็บข้อมูลของผู้ใช้ไฟฟ้าแต่ละราย โดยจะมีมาตรฐานความละเอียดในการเก็บข้อมูลทุก ๆ 15 นาที อาจมากหรือน้อยกว่านั้นขึ้นอยู่กับความเหมาะสมของระบบ จากนั้นระบบจะประมวลผลข้อมูลการใช้ไฟฟ้าของมิเตอร์แต่ละเครื่องและส่งข้อมูลดังกล่าวไปยังหน่วยงานที่เกี่ยวข้องสามารถจัดพิมพ์ใบแจ้งค่าไฟฟ้าได้ เป้าหมายหลักอีกประการนอกจากสามารถทำใบแจ้งค่าไฟได้อย่างรวดเร็วแล้ว นั่นคือผู้ให้บริการไฟฟ้าสามารถให้ข้อมูลการใช้ไฟฟ้าของผู้ใช้ไฟฟ้าแต่ละรายผ่านการแสดงผลในรูปแบบเว็บไซต์ โดยจะสามารถตรวจสอบข้อมูลสรุปเป็นรายวัน รายสัปดาห์ รายเดือนและรายปี หรือเปรียบเทียบข้อมูลการใช้ไฟฟ้า ณ ระยะเวลาใดเวลาหนึ่งได้ เพื่อให้ผู้ใช้ไฟฟ้านั้นสามารถนำข้อมูลปริมาณการใช้ไฟ (Load Profile) ไปบริหารจัดการไฟฟ้า (Demand-Side Management) ของตนให้เกิดประสิทธิภาพและเกิดความคุ้มค่าสูงสุด

โครงสร้างพื้นฐานระบบมิเตอร์ขั้นสูงหรือ Advanced Metering Infrastructure (AMI) นั้นเป็นโครงสร้างพื้นฐานแบบใหม่ที่ถูกพัฒนามาจากโครงสร้างเดิมของระบบอ่านหน่วยมิเตอร์ไฟฟ้าแบบอัตโนมัติ ซึ่งเป็นองค์ประกอบสำคัญที่จะช่วยพัฒนาศักยภาพให้โครงข่ายไฟฟ้าอัจฉริยะมีความชาญฉลาดมากขึ้น ช่วยให้ผู้ให้บริการไฟฟ้าและผู้ใช้ไฟฟ้าสามารถเข้าถึงกันได้ง่ายขึ้นด้วยการสื่อสารแบบ

สองทิศทาง (Two-way communication) ทำให้ผู้ให้บริการไฟฟ้าสามารถบริหารจัดการและควบคุมไฟฟ้า รวมทั้งสังเกตการณ์หรือตัดต่อวงจรจากระยะไกลผ่านเครือข่ายสื่อสารเพื่อรักษาความเสถียรภาพให้โครงข่ายไฟฟ้าอัจฉริยะได้อยู่เสมอ ในด้านผู้ใช้บริการไฟฟ้าเองก็ยังสามารถรับรู้ข้อมูลต่าง ๆ เช่น ข้อมูลการใช้พลังงานไฟฟ้าของตนเองผ่านจอแสดงผลภายในบ้าน (In-Home Displays; IHD) ตามเวลาจริง ณ ขณะนั้น ซึ่งจะช่วยให้ผู้ใช้ไฟฟ้านั้นสามารถตัดสินใจและควบคุมการใช้ไฟของตนให้เกิดความคุ้มค่าที่สุดได้เช่นกัน



รูปที่ 1 ภาพรวมของระบบมิเตอร์ขั้นสูง

จากงานวิจัย [1, 2] กล่าวว่าความชาญฉลาดของโครงข่ายไฟฟ้าอัจฉริยะจะไม่เกิดขึ้นเลยหากไม่มีระบบย่อย (Sub-system) หลาย ๆ ส่วนประกอบการทำงานเข้าด้วยกัน โดยระบบย่อยแต่ละระบบจะมีการทำงานเป็นลำดับชั้น (Layer) ในแต่ละลำดับชั้นจะสร้างข้อมูลส่งออก (Output) และจะทำการส่งผ่าน (Transmit) เพื่อเป็นข้อมูลนำเข้า (Input) สำหรับลำดับชั้นถัดไปดังแสดงในรูปที่ 1 ซึ่งการวางระบบย่อยที่มีความเสถียรภาพและมั่นคงจะเป็นกลไกสำคัญที่จะทำให้โครงข่ายไฟฟ้าอัจฉริยะมีประสิทธิภาพที่ดีขึ้น โครงสร้างพื้นฐานระบบมิเตอร์ขั้นสูงนี้มีโครงสร้างพื้นฐานภายในที่ประกอบไปด้วยระบบย่อยสำคัญได้แก่ มิเตอร์อัจฉริยะ โครงข่ายการสื่อสาร และระบบบริหารจัดการข้อมูล ซึ่งแต่ละระบบนั้นมีหน้าที่สำคัญที่แตกต่างกันออกไปดังนี้

2.1.1 มิเตอร์อัจฉริยะ

จากบทความ [3] กล่าวว่ามิเตอร์อัจฉริยะ (Smart Meter) เป็นอุปกรณ์อิเล็กทรอนิกส์ที่ใช้สำหรับวัดค่าพลังงานทางไฟฟ้าและสามารถเก็บบันทึกข้อมูลค่าที่วัดได้ไว้ในหน่วยความจำภายใน โดยมิเตอร์อัจฉริยะนี้ถูกออกแบบมาเพื่อทดแทนมิเตอร์แบบจานหมุนหรือมิเตอร์อิเล็กทรอนิกส์ดั้งเดิมที่มีความสามารถจำกัด นอกจากจะสามารถวัดค่าพลังงานไฟฟ้าที่ใช้ไปในช่วงเวลาหนึ่งแล้ว มิเตอร์อัจฉริยะยังสามารถติดต่อสื่อสารหรือส่งสัญญาณแจ้งเตือนไปยังศูนย์ควบคุมกลางได้อีกด้วย มิเตอร์อัจฉริยะนั้นสามารถติดต่อสื่อสารโดยตรงจากผู้ให้บริการไฟฟ้าได้บ่อยเท่าที่ต้องการ โดยไม่จำเป็นต้องส่งเจ้าหน้าที่ไปยังพื้นที่ต่าง ๆ เพื่อจดหน่วยไฟฟ้าในทุก ๆ เดือนอีกต่อไป อีกทั้งการซ่อมบำรุงก็สามารถทำได้โดยง่ายเนื่องจากชิ้นส่วนภายในที่เป็นเชิงกลมีน้อยลงกว่าเดิมมาก หรือในอีกมุมหนึ่งคือช่วยให้ผู้ให้บริการไฟฟ้าเข้าใจถึงสถานการณ์การใช้พลังงานไฟฟ้า ณ ช่วงเวลานั้น ๆ เพื่อที่จะได้ควบคุมปริมาณไฟฟ้าได้อย่างถูกต้องและทันเวลาในกรณีที่ผู้ให้บริการพบว่าการผลิตไฟฟ้าอาจไม่เพียงพอต่อการตอบสนองการใช้พลังงานที่กำลังเพิ่มขึ้น ผู้ให้บริการไฟฟ้าก็จะส่งข้อมูลผ่านมิเตอร์อัจฉริยะไปยังผู้ใช้ไฟฟ้าเพื่อให้ทราบถึงสถานะดังกล่าวและขอความร่วมมือในการลดการใช้ปริมาณไฟฟ้า หรือลดการใช้พลังงานในช่วงเวลาที่มีการใช้พลังงานขึ้นสูงสุด (Peak) โดยข้อมูลที่ผู้ให้บริการไฟฟ้าจะส่งไปให้ผู้ใช้ไฟฟ้านั้นอาจมีการแจ้งค่าใช้ไฟฟ้าในช่วงเวลาพีคดังกล่าวซึ่งอาจมีราคาที่สูงขึ้นเป็นเท่าตัวหรือมากกว่านั้น ซึ่งเป็นตัวแปรสำคัญที่จะช่วยให้ผู้ใช้ไฟฟ้าตัดสินใจได้ว่า จะลดการใช้พลังงานอย่างไร ตัวอย่างเช่นการเปลี่ยนจากการใช้เครื่องปรับอากาศเป็นพัดลมหรือเปิดประตูหน้าต่างเพื่อช่วยในการถ่ายเทอากาศเพื่อลดการใช้พลังงาน เป็นต้น [4]



(ก.)

(ข.)

(ค.)

รูปที่ 2 ตัวอย่างมิเตอร์แบบต่าง ๆ

(ก.) มิเตอร์แบบจานหมุน (ข.) มิเตอร์อิเล็กทรอนิกส์ (ค.) มิเตอร์อัจฉริยะ

โดยทั่วไปแล้วมิเตอร์อัจฉริยะนั้นมีสองหน้าที่หลักคือการวัดค่าพลังงานทางไฟฟ้าและการสื่อสาร โดยการวัดค่าทางไฟฟ้านั้นในความเป็นจริงแล้วมีความแตกต่างกันออกไปขึ้นอยู่กับปัจจัยหลายประการ อาทิ ภูมิประเทศ สภาวะทางไฟฟ้าของพื้นที่นั้น ๆ ความแม่นยำที่เหมาะสม โปรแกรมประยุกต์ (Application) ที่จะนำไปใช้งาน หรือในส่วนของ การสื่อสารเองก็มีปัจจัยที่สำคัญมากมาย โดยเฉพาะความปลอดภัยและการเข้ารหัสของข้อมูล ซึ่งเราจะต้องเลือกอุปกรณ์ที่มีความเหมาะสมให้เข้ากับปัจจัยที่กล่าวมาข้างต้น ดังนั้นมิเตอร์อัจฉริยะที่ดีควรมีคุณสมบัติพื้นฐานที่สำคัญดังต่อไปนี้ [2]

- การวัดเชิงปริมาณ (Quantitative Measurement) : มิเตอร์ควรจะวัดค่าพลังงานทางไฟฟ้าโดยใช้หลักการ รูปแบบ หรือวิธีการทางไฟฟ้าแบบต่าง ๆ ได้อย่างแม่นยำ
- การควบคุมและการสอบเทียบ (Control and Calibration) : โดยทั่วไปแล้วมิเตอร์ควรจะสามารถชดเชยค่าที่มีความคลาดเคลื่อนไปเล็กน้อยได้
- การสื่อสาร (Communication) : มิเตอร์ควรจะส่งข้อมูลที่เก็บไว้ในหน่วยความจำ หรือรับคำสั่งจากศูนย์สั่งการ หรือแม้กระทั่งความสามารถในการยกระดับเฟิร์มแวร์ (Upgrade firmware) ได้เป็นอย่างดี
- การบริหารจัดการกำลังงาน (Power Management) : ในกรณีที่แหล่งกำเนิดไฟฟ้าหลักดับ ระบบควรจะสามารถรักษาสถานะการทำงานไว้ได้
- การแสดงผล (Display) : ผู้ใช้บริการไฟฟ้าควรจะเห็นข้อมูลมิเตอร์แบบเวลาจริงของตนได้ เนื่องจากข้อมูลดังกล่าวเป็นฐานข้อมูลในการเรียกเก็บเงิน ทำให้ผู้ให้บริการไฟฟ้าสามารถบริหารจัดการการใช้พลังงานไฟฟ้าได้อย่างเหมาะสม
- การประสานเวลา (Synchronization) : มิเตอร์ที่ดีจะต้องทำงานการประสานเวลาได้ เนื่องจากการทำงานประสานเวลาเป็นสิ่งสำคัญในการส่งข้อมูลต่าง ๆ ไปยังศูนย์สั่งการ หรือระบบอื่น ๆ ที่เกี่ยวข้องเพื่อที่จะสามารถวิเคราะห์ข้อมูลได้แบบเวลาจริง (Real-time)

จากคุณสมบัติพื้นฐานของมิเตอร์อัจฉริยะข้างต้น ทำให้สามารถสรุปหน้าที่หลักของมิเตอร์อัจฉริยะได้ดังนี้

- การคิดเงินค่าไฟโดยใช้เวลาเป็นฐานราคา (Time-based pricing)
- ให้บริการข้อมูลการบริโภคพลังงาน (Providing consumption data)
- ระบบหักลบกลบหน่วยอัตโนมัติจากไฟฟ้าที่ผลิตใช้เอง (Net metering)
- ระบบแจ้งเตือนเมื่อเกิดความผิดปกติหรือไฟดับ (Failure and outage notification)
- ควบคุมการทำงานจากระยะไกล (Remote command operations)

- ควบคุมโหลดเพื่อให้สอดคล้องกับการตอบสนองด้านโหลด (Load limiting for demand response purposes)
- สังเกตการณ์คุณภาพทางไฟฟ้า (Power quality monitoring)
- ตรวจจับการรัดแงะและการขโมยไฟฟ้า (Tamper and energy theft detection)
- สื่อสารกับอุปกรณ์อัจฉริยะอื่น ๆ (Communications with other intelligent devices) และ
- เป็นมิตรต่อสิ่งแวดล้อมโดยการลดการปล่อยแก๊ส (Improving environmental conditions by reducing emission)

2.1.2 โครงข่ายการสื่อสาร

โครงสร้างพื้นฐานระบบมิเตอร์ขั้นสูงที่ติดตั้งจะต้องมีสิ่งสำคัญอีกประการหนึ่งคือตัวกลางที่ใช้ในการรับส่งข้อมูลที่มีประสิทธิภาพ สามารถสื่อสารได้รวดเร็วและแม่นยำเพื่อให้สอดคล้องกับการสื่อสารแบบเวลาจริง (Real-Time Communication)

โครงข่ายระบบสื่อสารเป็นส่วนที่มีความสำคัญต่อ AMI เป็นอย่างยิ่ง จะต้องมีความน่าเชื่อถือสูง สามารถรองรับข้อมูลปริมาณมากได้ รับส่งข้อมูลและคำสั่งควบคุมได้รวดเร็วทันเวลา ต้องมีความมั่นคงจากภัยพิบัติ มีความปลอดภัยต่อการโจมตีทางกายภาพและการโจมตีทางไซเบอร์ ไม่มีข้อจำกัดในด้านรูปแบบการเชื่อมต่อหรือสถาปัตยกรรม โดยพื้นฐานของระบบสื่อสารนี้สามารถใช้ตัวกลางการสื่อสารข้อมูลแบบมีสาย เช่น PLC, Broadband over Power Lines (BPL), Optical Fiber หรือใช้การสื่อสารแบบไร้สาย เช่น RF, cellular 3G/4G/5G, Narrow Band (NB) ก็ได้เช่นกัน

มาตรฐานและโพรโทคอลที่เกี่ยวข้องกับเทคโนโลยี AMI และโครงข่ายไฟฟ้าอัจฉริยะมีหลากหลายมาตรฐาน เช่น ANSI C12.XX, DLMS/COSEM, MODBUS RTU, DNP3, M-Bus และ IEC61850 โดยโพรโทคอลที่เป็นที่นิยมสำหรับใช้กับระบบ AMR คือ ZigBee, MODBUS, IEC62056 DLMS/COSEM, IEC61107 และ ANSI C.12.18 ความท้าทายด้านความปลอดภัยของระบบ AMI เป็นอีกหนึ่งเรื่องสำคัญที่ต้องคำนึงถึงทั้งเครือข่ายสื่อสารและระบบข้อมูล ซึ่งได้แก่ ความปลอดภัยในด้านข้อมูลผู้บริโภค เช่น ข้อมูลส่วนตัว พฤติกรรมการบริโภค หรือวิถีชีวิต เป็นต้น ความปลอดภัยจากการโจมตีทางกายภาพและภัยธรรมชาติ และที่สำคัญมากในปัจจุบันคือความปลอดภัยต่อการโจมตีทางไซเบอร์ ดังนั้น AMI ต้องสามารถให้การรับรองความปลอดภัยในด้านการรักษาความลับข้อมูล, ความถูกต้องของข้อมูล, ความพร้อมใช้งานของระบบ และความรับผิดชอบของระบบ ตามมาตรฐาน ISO/IEC 27000 หรือ Information Security Management Systems (ISMS) Standards

นอกจากนี้ AMI สามารถช่วยในการตรวจวัดพลังงานไฟฟ้าที่สูญเสียในสายส่งและตรวจจัดการขโมยใช้ไฟฟ้า เพื่อช่วยให้การปฏิบัติการและการบริหารจัดการโครงข่ายไฟฟ้ามีประสิทธิภาพมากที่สุดอีกด้วย

2.1.3 ระบบบริหารจัดการข้อมูล

ระบบบริหารจัดการข้อมูล (Data Management System; DMS) เป็นระบบที่ผู้ให้บริการไฟฟ้ารวบรวมข้อมูลเพื่อใช้ในการวิเคราะห์เพื่อวัตถุประสงค์ต่าง ๆ เช่นการวิเคราะห์ข้อมูลการใช้พลังงาน การใช้ข้อมูลเพื่อวิเคราะห์การตอบสนองด้านโหลด หรือการทำบิล เป็นต้น สำหรับระบบบริหารจัดการข้อมูลในโครงสร้างพื้นฐานระบบมิเตอร์ขั้นสูงนี้จะมีองค์ประกอบหลายส่วนที่ประกอบการทำงานเข้าด้วยกัน อาทิ ระบบบริหารจัดการข้อมูลมิเตอร์ผู้ใช้ไฟฟ้า (Meter Data Management System; MDMS) ระบบข้อมูลลูกค้า (Customer Information System; CIS) ระบบจัดการเมื่อกระแสไฟฟ้าขัดข้อง (Outage Management System; OMS) การบริหารทรัพยากรขององค์กร (Enterprise Resource Planning; ERP) ระบบสารสนเทศภูมิศาสตร์ (Geographic Information System; GIS) รวมไปถึงการบริหารจัดการโหลดของหม้อแปลงไฟฟ้า (Transformer Load Management; TLM)

นอกจากนี้ระบบบริหารจัดการข้อมูลของ AMI มีส่วนประกอบของโครงสร้างพื้นฐานศูนย์ข้อมูล ซึ่งจะประกอบไปด้วยคอมพิวเตอร์เซิร์ฟเวอร์ ระบบฐานข้อมูล ระบบจัดเก็บข้อมูล รวมถึงระบบคอมพิวเตอร์เสมือน โครงสร้างและระบบย่อยต่าง ๆ เหล่านี้จะเป็นส่วนสำคัญที่จะช่วยปรับปรุงและเพิ่มประสิทธิภาพการจัดการโครงข่ายไฟฟ้า ช่วยปรับปรุงและเพิ่มประสิทธิภาพการจัดการสาธารณูปโภค อีกทั้งเพิ่มการมีส่วนร่วมของผู้บริโภค อันเป็นจุดประสงค์หลักของระบบบริหารจัดการข้อมูล โดยที่ระบบบริหารจัดการข้อมูลที่มีความสำคัญต่อ AMI มากที่สุดคือ ระบบบริหารจัดการข้อมูลมิเตอร์ผู้ใช้ไฟฟ้าหรือ MDMS ซึ่งเป็นโมดูลกลางที่ทำหน้าที่สื่อสารแลกเปลี่ยนข้อมูลกับโมดูลต่าง ๆ โดย MDMS นี้จะติดตั้งอยู่ที่ศูนย์สั่งการของผู้ให้บริการไฟฟ้า ทำหน้าที่เป็นตัวกลางในการแลกเปลี่ยนข้อมูลค่าพลังงานทางไฟฟ้าจากมิเตอร์ของ AMI กับระบบต่าง ๆ นอกจากข้อมูลค่าพลังงานทางไฟฟ้าแบบคาบเวลาจากมิเตอร์แล้ว AMI ยังให้ข้อมูลเกี่ยวกับราคาไฟฟ้า เช่น อัตราค่าไฟฟ้าตามช่วงเวลาการใช้ (Time of Use Rates; TOU), อัตราค่าไฟฟ้าช่วงวิกฤต (Critical Peak Pricing; CPP) และ อัตราค่าไฟฟ้า ณ เวลาปัจจุบัน (Real Time Pricing; RTP) เป็นต้น รวมไปถึงข้อมูลเกี่ยวกับผู้ใช้ไฟฟ้า พฤติกรรมการใช้ไฟฟ้า ข้อมูลการทำธุรกรรม ประเภทธุรกิจหรือประเภทโหลดไฟฟ้า เพื่อสนับสนุนระบบทำใบแจ้งค่าไฟฟ้า และการบริหารจัดการการใช้พลังงานไฟฟ้าอย่างมีประสิทธิภาพ ซึ่งข้อมูลต่าง ๆ AMI จะใช้สำหรับคำนวณคาดการณ์ความต้องการการใช้ไฟฟ้า ตรวจสอบวิเคราะห์รูปแบบการใช้ไฟฟ้า และวางแผนการจัดหาและสำรองไฟฟ้า ทำให้ผู้ให้บริการ

ไฟฟ้าสามารถบริหารการผลิตไฟฟ้าล่วงหน้าได้อย่างเหมาะสมและมีประสิทธิภาพ สามารถใช้ข้อมูลในการคำนวณและพิมพ์ใบแจ้งค่าไฟฟ้า รวมถึงรายงานข้อมูลและแจ้งเตือนเหตุสำคัญให้กับผู้ดูแลระบบหรือผู้ใช้ไฟฟ้าได้โดยตรง ผู้ใช้ไฟฟ้าเองก็สามารถตรวจสอบข้อมูลการใช้ไฟฟ้า ค่าไฟฟ้า ค่าชดเชยจากการลดการใช้ไฟฟ้า หรือรายได้จากการขายไฟฟ้าผ่านแอปพลิเคชันบนอุปกรณ์อัจฉริยะต่าง ๆ ซึ่งจะช่วยให้ผู้ใช้ไฟฟ้าสามารถตัดสินใจเลือกใช้ไฟฟ้าได้ในราคาถูกลงอย่างเหมาะสม

2.2 การตอบสนองด้านโหลด

การตอบสนองด้านโหลด (Demand Response; DR) นั้น คือการตอบสนองการใช้พลังงานไฟฟ้าด้วยการเปลี่ยนแปลงพฤติกรรมการใช้ไฟฟ้าไปจากเดิม ถือเป็นมาตรการที่ใช้กระตุ้นผู้ใช้ไฟฟ้าให้มีการลดการใช้พลังงานไฟฟ้าในช่วงเวลาต่าง ๆ โดยเฉพาะช่วงเวลาที่มีความต้องการใช้ไฟฟ้าสูงสุด (Peak Demand) ซึ่งถือว่าเป็นช่วงเวลาที่ราคาค่าไฟฟ้าสูง หรือแม้กระทั่งช่วงเวลาที่ระบบไฟฟ้าเกิดความผิดปกติทำให้ความน่าเชื่อถือของระบบไฟฟ้าต่ำ มาตรการการตอบสนองด้านโหลดจึงเป็นแรงจูงใจที่จะทำให้ผู้ใช้ไฟฟ้าเกิดการเปลี่ยนแปลงพฤติกรรมการใช้ไฟ เพื่อตอบสนองต่อการเปลี่ยนแปลงราคาค่าไฟฟ้าในช่วงเวลาหนึ่ง หรือเพื่อผลตอบแทนจูงใจ (Incentive Payment) ที่ผู้ให้บริการไฟฟ้าออกแบบมาเพื่อใช้โน้มน้าวให้ผู้ใช้ไฟฟาลดการใช้พลังงานในช่วงเวลาที่ราคาค่าไฟฟ้าขึ้นสูง หรือเพื่อตอบสนองต่อความจำเป็นต่อการบริหารจัดการพลังงานไฟฟ้าในเรื่องของการควบคุมความต้องการใช้ไฟฟ้าให้มีความเหมาะสมกับความสามารถในการผลิตพลังงานไฟฟ้าในช่วงเวลาต่าง ๆ ได้อย่างมีประสิทธิภาพ สามารถช่วยป้องกันไม่ให้เกิดสภาวะที่กำลังการผลิตนั้นมีไม่เพียงพอต่อการใช้งานของผู้ใช้ไฟฟ้า ซึ่งอาจส่งผลให้เกิดไฟฟ้าดับเป็นวงกว้าง [5] ดังนั้นการตอบสนองด้านโหลดจึงเป็นวิธีการสำคัญที่จะช่วยในการจัดการสภาวะวิกฤตด้านพลังงานไฟฟ้า สร้างสมดุลระหว่างความต้องการใช้พลังงานกับกำลังการผลิต และเสริมความมั่นคงในด้านพลังงานไฟฟ้าทั้งในระยะสั้นและระยะยาว ตัวอย่างของระบบที่มีโครงสร้างการตอบสนองด้านโหลดได้แก่ การที่ผู้ให้บริการไฟฟ้าส่งข้อมูลอัตราค่าไฟฟ้าตามเวลาจริงไปยังมิเตอร์อัจฉริยะของผู้ใช้ไฟฟ้า เมื่อผู้ใช้ไฟฟ้าได้รับข้อมูลดังกล่าว ผู้ใช้ไฟฟ้าจะสามารถตัดสินใจและตอบสนองถึงการบริหารจัดการการใช้ไฟฟ้าของตนให้มีความเหมาะสมกับสภาวะทางไฟฟ้าของโครงข่าย อีกทั้งยังช่วยให้ผู้ใช้ไฟฟ้าสามารถประหยัดเงินจากการหลีกเลี่ยงการใช้ไฟในช่วงเวลาดังกล่าวได้อีกด้วย

การบริหารจัดการการใช้พลังงาน (Demand Side Management; DSM) ในความเป็นจริงแล้วมีความหมายคล้ายคลึงกับการตอบสนองด้านโหลด (Demand Response; DR) โดยที่การตอบสนองด้านโหลดนั้นจะมุ่งเน้นในเรื่องของการจูงใจให้ผู้ใช้ไฟฟ้ามีการเปลี่ยนแปลงพฤติกรรมการใช้ไฟในช่วงเวลาสั้น ๆ เพื่อตอบสนองต่อสภาวะทางไฟฟ้าในโครงข่าย ณ ขณะนั้นรวมถึงป้องกันไม่ให้

เกิดไฟฟ้าดับ ส่วนการบริหารจัดการการใช้พลังงานหรือ DSM นั้นจะมุ่งเน้นถึงการใช้พลังงานไฟฟ้าอย่างมีประสิทธิภาพ (Energy Efficiency) โดยการให้ผู้ใช้ไฟฟ้าเปลี่ยนพฤติกรรมหรือเปลี่ยนอุปกรณ์ไฟฟ้า ซึ่งเป็นการปรับเปลี่ยนลักษณะของความต้องการใช้พลังงานอันเป็นผลดีในระยะยาว การบริหารจัดการการใช้พลังงานและการตอบสนองด้านโหลดมักจะถูกนำมาใช้ร่วมกันเพื่อบริหารจัดการการใช้พลังงานไฟฟ้าให้เกิดประสิทธิภาพสูงสุด เพิ่มความสมดุลระหว่างการใช้พลังงานและการผลิต โดยเฉพาะอย่างยิ่งในกรณีที่ผู้ให้บริการไฟฟ้ามีโรงไฟฟ้าพลังงานหมุนเวียนขนาดใหญ่ และอาจช่วยลดการลงทุนก่อสร้างโรงไฟฟ้าที่ใช้ในการรองรับช่วงเวลาที่ระบบมีความต้องการใช้ไฟฟ้าสูงสุด (Peaking Plant) ซึ่งมีต้นทุนสูงได้ [6]

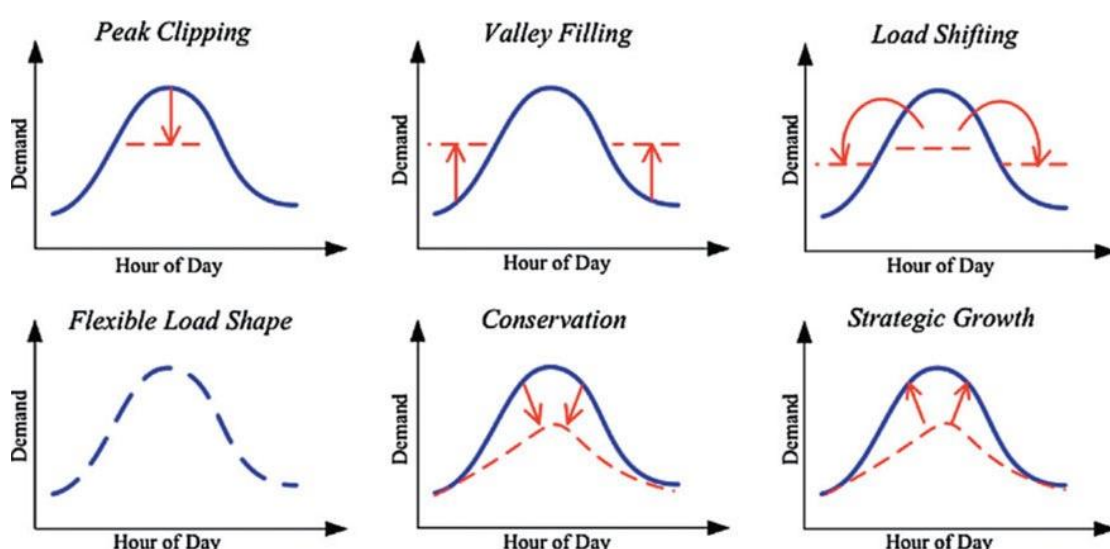
2.2.1 รูปแบบของการดำเนินการตอบสนองด้านโหลด

การดำเนินการตอบสนองด้านโหลดนั้นมีหลายรูปแบบขึ้นอยู่กับบริบทของการใช้และการผลิตพลังงานที่แตกต่างกัน ทำให้บทบาทของการตอบสนองด้านโหลดไม่ได้ขึ้นอยู่กับแค่การช่วยลดการใช้พลังงานเท่านั้น ในบางครั้งการตอบสนองด้านโหลดอาจนำไปสู่การใช้พลังงานไฟฟ้าในปริมาณเท่าเดิม หรืออาจเพิ่มการใช้พลังงานให้มากขึ้นในบางช่วงเวลาดังตัวอย่างที่แสดงให้เห็นในรูปที่ 3 ซึ่งการตอบสนองด้านโหลดที่สามารถพบเห็นได้บ่อยคือ การดำเนินการตอบสนองด้านโหลดแบบตัดลดความต้องการไฟฟ้าสูงสุด (Peak Clipping) หรือก็คือการลดปริมาณความต้องการไฟฟ้าสูงสุดด้วยวิธีการลดหรืองดการใช้พลังงานไฟฟ้าในช่วงเวลาที่มีความต้องการไฟฟ้าสูงสุด ตัวอย่างเช่นหากความต้องการไฟฟ้าสูงสุดอยู่ที่ช่วงเวลาเที่ยงวัน (12:00 น.) ผู้ให้บริการไฟฟ้าจะส่งสัญญาณไปยังผู้ใช้ไฟฟ้าเพื่อขอให้ลดการใช้พลังงานในช่วงเวลาดังกล่าว เป็นต้น

รูปแบบการตอบสนองด้านโหลดแบบที่สองคือ การดำเนินการตอบสนองด้านโหลดแบบ Valley Filling เป็นรูปแบบการตอบสนองด้านโหลดที่ไม่ได้ทำเพื่อลดการใช้พลังงานลง ในทางกลับกัน รูปแบบการตอบสนองด้านโหลดรูปแบบนี้เป็นการเพิ่มปริมาณการใช้พลังงานไฟฟ้า ทำให้ภาพรวมเกิดการใช้พลังงานที่เพิ่มขึ้น การดำเนินการรูปแบบนี้จะถูกนำมาใช้ในกรณีที่มีการผลิตไฟฟ้าส่วนเกิน หรือการผลิตไฟฟ้านั้นมีมากเกินความต้องการการใช้ไฟฟ้า ในปัจจุบันนี้เริ่มมีแหล่งผลิตไฟฟ้าพลังงานหมุนเวียนที่เชื่อมต่อกับโครงข่ายไฟฟ้ามากขึ้น โดยที่ความสามารถในการผลิตไฟฟ้าของโรงไฟฟ้าพลังงานหมุนเวียนเหล่านี้จะขึ้นอยู่กับสภาพอากาศในแต่ละช่วงเวลา ซึ่งในบางเวลาอาจมีการผลิตพลังงานไฟฟ้าส่วนเกินขึ้นมา ทำให้การตอบสนองด้านโหลดแบบ Valley Filling เข้ามามีบทบาทในการช่วยบริหารจัดการพลังงานส่วนเกินในระบบโครงข่ายไฟฟ้ามากขึ้น การดำเนินการแบบ Valley Filling นั้นผู้ให้บริการไฟฟ้าจะส่งสัญญาณบอกผู้ใช้ไฟฟ้าเพื่อขอให้เพิ่มการใช้พลังงานมากขึ้น หากในกรณีที่ผู้ใช้ไฟฟ้ามีระบบกักเก็บพลังงาน ผู้ให้บริการไฟฟ้าจะทำการร้องขอให้เปิดการใช้งาน

ระบบเพื่อกักเก็บพลังงานส่วนเกิน เพื่อให้ค่าความต้องการใช้พลังงานอยู่ในระดับที่ใกล้เคียงกับกำลังการผลิตไฟฟ้าที่ไม่สามารถควบคุมได้ ณ ขณะนั้น ทำให้ไม่จำเป็นต้องปลดโรงไฟฟ้าพลังงานหมุนเวียนออกจากระบบ อันเป็นการช่วยรักษาเสถียรภาพของโครงข่ายไฟฟ้าให้มีความมั่นคงสม่ำเสมอ

และรูปแบบสุดท้ายคือการดำเนินการตอบสนองด้านโหลดแบบการปรับเลื่อนการใช้ไฟฟ้า (Load Shifting) รูปแบบนี้ใช้วิธีการเปลี่ยนช่วงเวลาการใช้ไฟฟ้าไปเป็นช่วงเวลาอื่นเพื่อลดความต้องการการใช้ไฟฟ้าสูงสุดในช่วงเวลาหนึ่ง ๆ ซึ่งไม่ได้เป็นการลดการใช้พลังงานไฟฟ้าแต่อย่างใด ผู้ใช้ไฟฟ้ายังคงใช้พลังงานเช่นเดิมเพียงแต่เปลี่ยนไปใช้พลังงานในช่วงเวลาอื่นเท่านั้น



รูปที่ 3 ตัวอย่างรูปแบบการตอบสนองด้านโหลดแบบต่าง ๆ [7]

2.2.2 ประเภทของการตอบสนองด้านโหลดและความยืดหยุ่นในการจัดการระบบไฟฟ้า

ประเภทของการตอบสนองด้านโหลดสามารถแบ่งได้ตามลักษณะของกลไกการตอบสนองได้ 2 ลักษณะ คือ

2.2.2.1 มาตรการการตอบสนองด้านโหลดต่อความน่าเชื่อถือของระบบ (Reliability-based Options)

เป็นรูปแบบการตอบสนองด้านโหลดต่อช่วงเวลาที่มีความน่าเชื่อถือของไฟฟ้าต่ำ เกิดเหตุการณ์ผิดปกติ หรือเหตุฉุกเฉิน ซึ่งอาจมีผลกระทบต่อเสถียรภาพต่อโครงข่ายไฟฟ้า โดยอาจมีการกำหนดค่าตอบแทนที่เหมาะสมให้กับผู้ใช้ไฟฟ้าที่เข้าร่วมดำเนินการ ได้แก่

- มาตรการควบคุมโหลดโดยตรง (Direct Load Control)
- มาตรการเชื่อมต่อผ่านระบบจัดการควบคุมโหลดผ่านระบบจัดการ (AutoDR to EMS)

- มาตรการอัตราค่าไฟฟ้าที่สามารถงดจ่ายไฟฟ้าได้ (Curtailable/Interruptible Tariff)
- มาตรการตอบสนองแบบฉุกเฉิน (Emergency Demand Response Program)
- มาตรการประมูลหรือซื้อคืน (Demand Bidding/Buyback Program)
- มาตรการตลาดกำลังไฟฟ้า (Capacity Market Program)
- มาตรการเครื่องกำเนิดไฟฟ้าสำรอง (Standby Generator)
- มาตรการระบบกักเก็บพลังงาน (Energy Storage)
- มาตรการกำลังผลิตเสริมความมั่นคง (Ancillary Service)

2.2.2.2 มาตรการการตอบสนองด้านโหลดต่อกลไกราคา (Price-based Options)

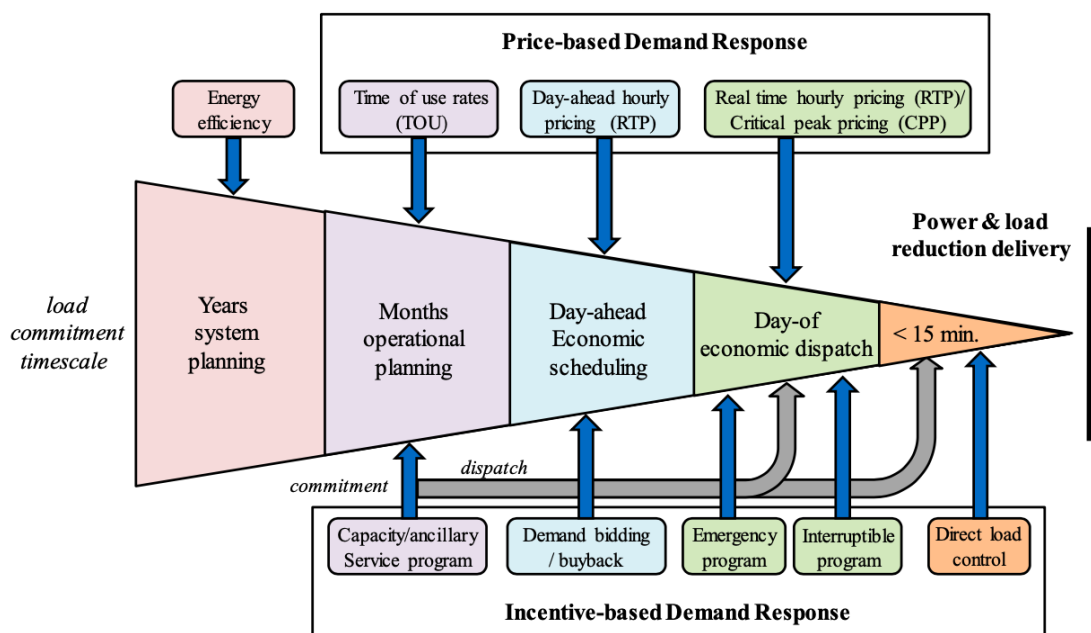
เป็นรูปแบบการตอบสนองด้านโหลดโดยใช้กลไกราคา ตั้งราคาค่าไฟฟ้าให้มีราคาสูงในช่วงที่มีความต้องการใช้ไฟฟ้าสูง หรือช่วงที่มีความเสี่ยงต่อที่จะเกิดเหตุการณ์ผิดปกติในระบบไฟฟ้า เพื่อให้ผู้ใช้ไฟฟ้าลดการใช้ไฟฟ้า หรือหลีกเลี่ยงไปใช้ไฟฟ้าในช่วงเวลาอื่นที่มีความต้องการใช้ไฟฟ้าต่ำ ซึ่งมีราคาค่าไฟฟ้าถูกกว่า ได้แก่

- มาตรการอัตราค่าไฟฟ้าตามช่วงเวลาการใช้ (Time of Use Rates)
- มาตรการอัตราค่าไฟฟ้าช่วงวิกฤต (Critical Peak Pricing)
- มาตรการอัตราค่าไฟฟ้าส่วนลดช่วงวิกฤต (Peak Time Rebate)
- มาตรการอัตราค่าไฟฟ้า ณ เวลาปัจจุบัน (Real Time Pricing)

การบริหารจัดการเพื่อเพิ่มความยืดหยุ่นให้กับระบบไฟฟ้า จำเป็นจะต้องมีการพิจารณาวางแผนและหาวิธีพัฒนากำลังการผลิตไฟฟ้าในระยะยาวเพื่อเพิ่มความยืดหยุ่นในระบบไฟฟ้าให้ครอบคลุม ควบคู่ไปกับการบริหารจัดการระบบไฟฟ้าในระยะสั้นในการส่งการเดินเครื่องโรงไฟฟ้าให้สามารถตอบสนองต่อความผันผวนตามความต้องการไฟฟ้าในแต่ละช่วงเวลาได้อย่างมีประสิทธิภาพ ปัจจุบันมีทางเลือกสำหรับเพิ่มความยืดหยุ่นให้กับระบบไฟฟ้าที่หลากหลายมากขึ้น โดยสามารถจัดลำดับตามต้นทุนเฉลี่ยในการบริหารจัดการไฟฟ้าได้ดังนี้

1. การปรับปรุงการเดินระบบไฟฟ้า (Operation Improvement)
2. การดำเนินการตอบสนองด้านโหลด (Demand Response)
3. การเพิ่มขีดความสามารถของระบบส่งและจำหน่ายไฟฟ้า (Grid Infrastructure)
4. การใช้แหล่งจ่ายไฟฟ้าที่มีการตอบสนองรวดเร็ว (Fast Ramping Supply)
5. การใช้ระบบกักเก็บพลังงาน (Energy Storage)

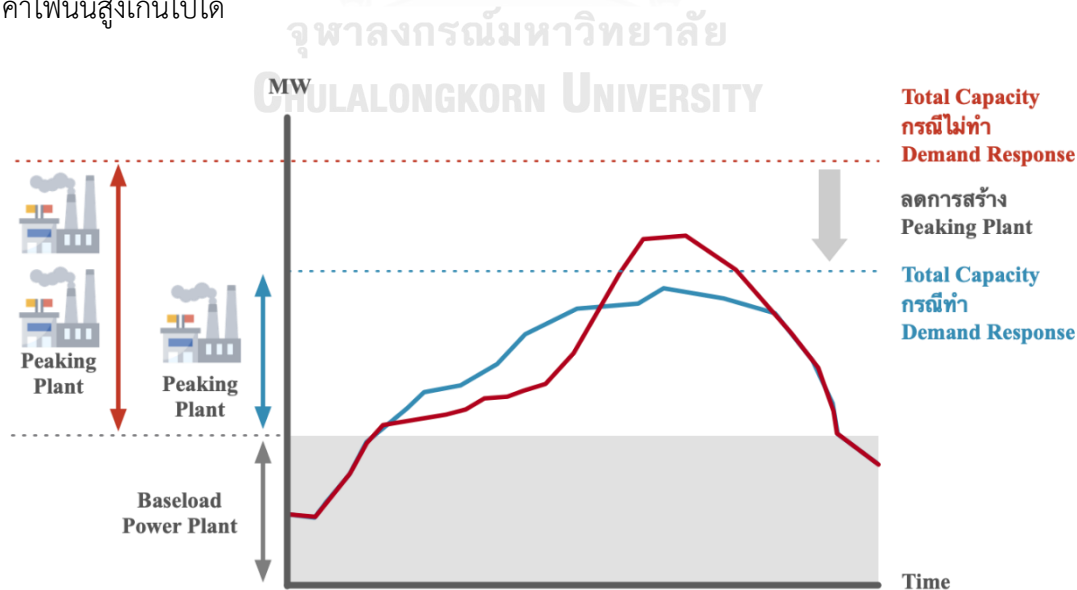
แน่นอนว่าการดำเนินการตอบสนองด้านโหลดเป็นหนึ่งในทางเลือกที่สำคัญที่จะเพิ่มความยืดหยุ่นให้กับโครงข่ายไฟฟ้า อีกทั้งยังเป็นสิ่งที่จะช่วยทั้งการวางแผนเพื่อบริหารการเดินระบบไฟฟ้าในระยะสั้น และวางแผนพัฒนาการผลิตไฟฟ้าในระยะยาว โปรแกรมการตอบสนองด้านโหลดทั้งสองรูปแบบดังแสดงให้เห็นในรูปที่ 4 นั้นก็มีข้อดีแตกต่างกัน ทั้ง Price-based demand response และ Incentive-based demand response ต่างมีคุณสมบัติที่เหมาะสมกับการบริหารจัดการระบบไฟฟ้าที่แตกต่างกันในแต่ละช่วงเวลา เริ่มตั้งแต่การวางแผนการผลิตไฟฟ้าในระดับปี (Years system planning) การกำหนดตารางการทำงานล่วงหน้าระดับเดือน (Months operational planning) หรือระดับวัน (Day-ahead Economic scheduling) การสั่งการเดินเครื่องผลิตไฟฟ้าระหว่างวัน รวมถึงการปรับสมดุลไฟฟ้าให้เป็นไปตามอุปสงค์อุปทานในระดับนาที่ ทั้งนี้การออกแบบโครงสร้างตลาด รวมถึงโครงสร้างกิจการไฟฟ้าซึ่งมีลักษณะเฉพาะตัวที่แตกต่างกันออกไป ดังนั้นการเลือกโปรแกรมการตอบสนองด้านโหลดที่เหมาะสม จึงขึ้นอยู่กับภาพรวมของตลาดหรือโครงสร้างกิจการไฟฟ้าอีกด้วย



รูปที่ 4 โปรแกรมมาตรการตอบสนองด้านโหลดที่เกี่ยวข้องต่อการบริหารจัดการไฟฟ้า (Demand Response Spectrum) [7-9]

2.2.3 ประโยชน์ของการตอบสนองด้านโหลด

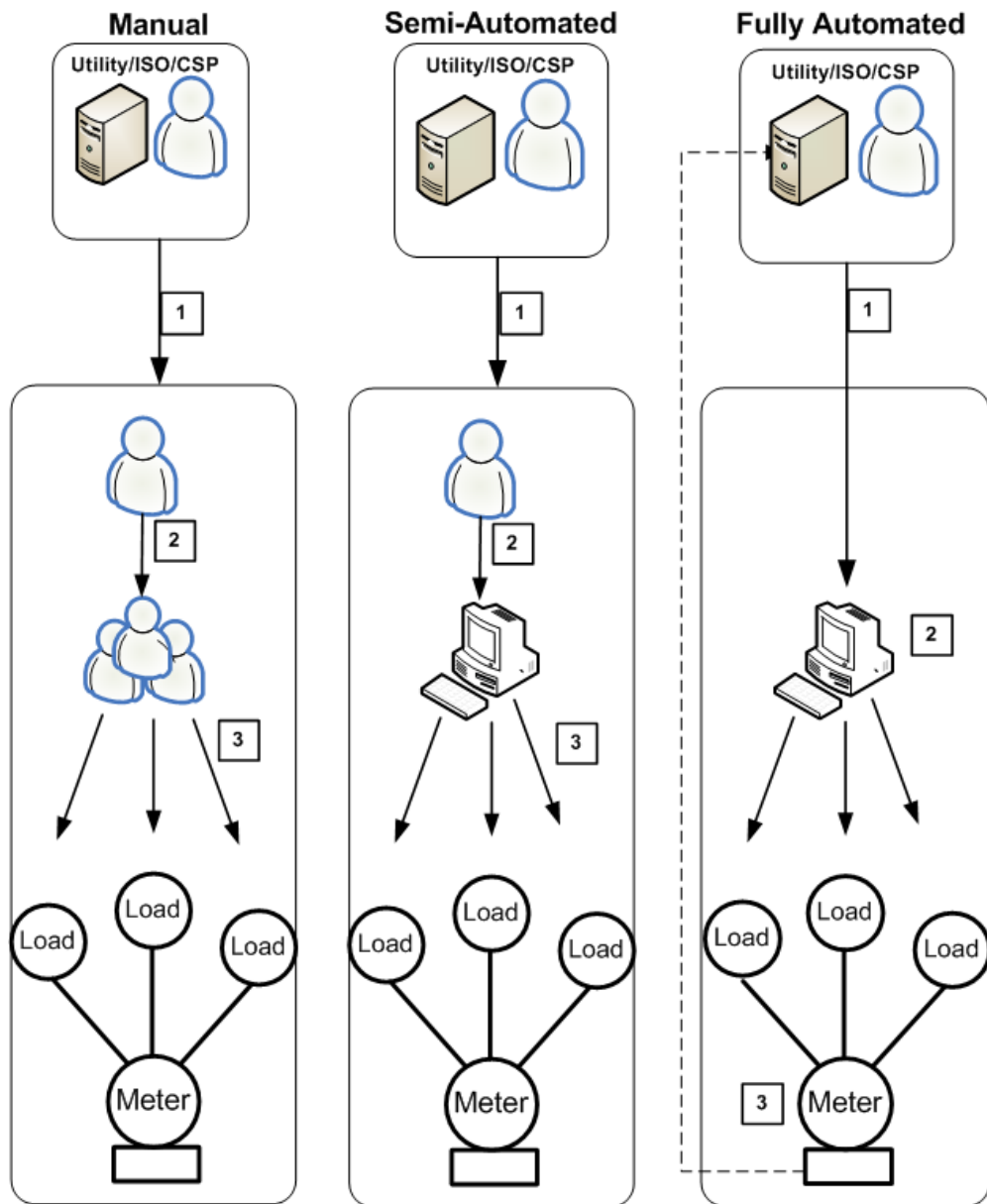
ประโยชน์ของการตอบสนองด้านโหลดที่เห็นได้ชัดเจนที่สุดคือการลดหรือชะลอการสร้างโรงไฟฟ้าที่ใช้ในการรองรับช่วงเวลาที่ระบบมีความต้องการใช้ไฟฟ้าสูงสุด (Peaking Plant) หากความต้องการใช้พลังงานไฟฟ้ามีมากขึ้นแน่นอนว่าจะต้องเพิ่มกำลังการผลิตที่สูงขึ้น จำเป็นที่จะต้องเพิ่มกำลังการผลิตไฟฟ้าสูงสุด ซึ่งในบางครั้งอาจมีกำลังการผลิตที่ไม่เพียงพอต่อความต้องการการใช้พลังงาน นำมาสู่กระบวนการสร้างโรงไฟฟ้าข้างต้นซึ่งมีค่าใช้จ่ายในการดำเนินการสูง แต่หากนำรูปแบบการตอบสนองด้านโหลดมาประยุกต์ใช้งานและดูค่าความต้องการใช้พลังงานไฟฟ้าในช่วงเวลาต่าง ๆ จะทำให้สามารถบริหารจัดการพลังงานไฟฟ้าที่มีอยู่อย่างจำกัดให้เกิดประโยชน์สูงสุดได้ ทั้งนี้หน่วยงานด้านการไฟฟ้าจำเป็นต้องวางแผนการผลิตไฟฟ้าให้เพียงพอต่อความต้องการไฟฟ้า ซึ่งสามารถแบ่งโรงไฟฟ้าตามความต้องการออกได้เป็น 2 แบบคือ โรงไฟฟ้าที่ผลิตตามความต้องการฐาน (Baseload Power Plant) และโรงไฟฟ้าที่ผลิตไฟฟ้าเพื่อรองรับช่วงเวลาที่มีความต้องการใช้ไฟฟ้าสูงสุด (Peaking Plant) โดยลักษณะของโรงไฟฟ้าที่ผลิตตามความต้องการฐานนั้นจะใช้เชื้อเพลิงราคาถูกในการผลิตไฟฟ้า มีการเดินเครื่องที่กำลังการผลิตคงที่อยู่ตลอดเวลา อาทิ โรงไฟฟ้าพลังงานความร้อนถ่านหิน ส่วนโรงไฟฟ้า Peaking Plant นั้นจะมีลักษณะการเดินเครื่องเฉพาะในเวลาที่มีความต้องการไฟฟ้าสูงสุด ตัวอย่างเช่นโรงไฟฟ้าที่ใช้ น้ำมันดีเซลเป็นเชื้อเพลิง หรือโรงไฟฟ้าพลังน้ำแบบสูบกลับ เป็นต้น โดยปกติแล้วต้นทุนที่ใช้ในการผลิตไฟฟ้าของโรงไฟฟ้า Peaking Plant นั้นสูง ซึ่งเป็นตัวแปรสำคัญที่จะส่งผลกระทบต่อราคาค่าไฟในช่วงที่มีความต้องการสูงสุดด้วยเช่นกัน เพราะฉะนั้นการลดการสร้างโรงไฟฟ้าสำหรับรองรับช่วงเวลาที่มีความต้องการการใช้ไฟฟ้าสูงสุดนั้นจะสามารถช่วยไม่ให้ราคาค่าไฟนั้นสูงเกินไปได้



รูปที่ 5 ตัวอย่างประโยชน์ของการตอบสนองด้านโหลด [5]

2.2.4 ความซับซ้อนการดำเนินการตอบสนองด้านโหลดแบบควบคุมด้วยเจ้าหน้าที่ กึ่งอัตโนมัติ และอัตโนมัติสมบูรณ์

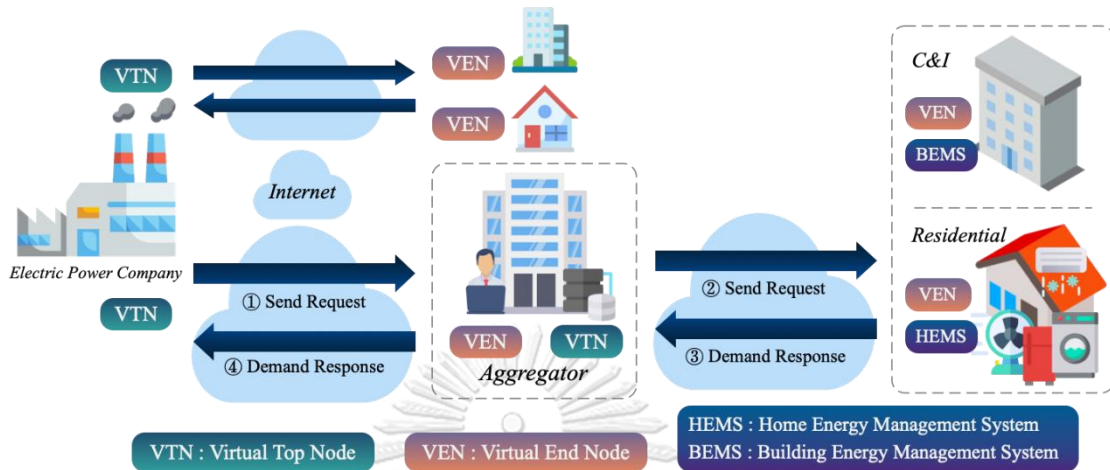
ลักษณะการดำเนินการตอบสนองด้านโหลดนั้นสามารถแบ่งประเภทตามความซับซ้อนของการทำงานได้อีก 3 แบบได้แก่ การดำเนินการตอบสนองด้านโหลดแบบควบคุมด้วยเจ้าหน้าที่ (Manual) แบบกึ่งอัตโนมัติ (Semi-Automated) และแบบอัตโนมัติโดยสมบูรณ์ (Fully Automated) โดยแบบควบคุมด้วยเจ้าหน้าที่นั้นจะเริ่มต้นลักษณะการดำเนินการด้วยการส่งข้อความผ่านอีเมลหรือโทรศัพท์ และจะถูกตัดสินใจเข้าร่วมโปรแกรมการตอบสนองด้านโหลดด้วยบุคคล จากนั้นจึงจะเริ่มดำเนินการตอบสนองด้านโหลดด้วยการลดการใช้พลังงาน หรือควบคุมอุปกรณ์เครื่องใช้ไฟฟ้าด้วยมือของบุคคล (Manually) ดังแสดงในรูปที่ 6 ซึ่งจะสามารถดำเนินการได้ทันทีโดยไม่ต้องจำเป็นจะต้องมีระบบโครงข่ายไฟฟ้าอัจฉริยะเข้ามารองรับแต่อย่างใด แตกต่างจากแบบอัตโนมัติโดยสมบูรณ์ที่จำเป็นต้องมีโครงข่ายไฟฟ้าอัจฉริยะเข้ามาเป็นตัวกลางในการสื่อสารเพื่อให้เกิดการประสานงานโดยตรงระหว่างอุปกรณ์กับอุปกรณ์ (Machine to Machine) ส่งผลให้การตอบสนองด้านโหลดนั้นเป็นไปได้อย่างรวดเร็วมากกว่า และไม่มีการใช้บุคคลเข้ามาเกี่ยวข้องกับกระบวนการทั้งหมด ตัวอย่างมาตรฐานการดำเนินการตอบสนองด้านโหลดที่มีชื่อเสียงคือ OpenADR ซึ่งถือเป็นมาตรฐานในการสื่อสารและสามารถสั่งการได้อย่างอัตโนมัติโดยสมบูรณ์โดยการสั่งการจากเซิร์ฟเวอร์ (Servers) ไปยังไคลแอนท์ (Clients) เพื่อแจ้งเหตุการณ์ (Events) สำหรับการตอบสนองด้านโหลด จากนั้นทำการจัดการปรับเปลี่ยนการใช้งานโหลดไฟฟ้า หรือปลดโหลดไฟฟ้าออกจากระบบ ทำให้อุปกรณ์ต่าง ๆ สามารถทำงานร่วมกันได้อย่างมีประสิทธิภาพตามมาตรการการตอบสนองด้านโหลด ซึ่งจะช่วยรักษาเสถียรภาพของโครงข่ายไฟฟ้าให้เป็นไปได้อย่างราบรื่นอีกด้วย



รูปที่ 6 การดำเนินการตอบสนองด้านโหลดแบบควบคุมด้วยเจ้าหน้าที่ แบบกึ่งอัตโนมัติ และแบบอัตโนมัติโดยสมบูรณ์ ตามลำดับ

2.3 ทฤษฎีซอฟต์แวร์ที่เกี่ยวข้อง

2.3.1 ซอฟต์แวร์การตอบสนองด้านโหลดแบบอัตโนมัติ OpenADR



รูปที่ 7 ภาพรวมการทำงานของซอฟต์แวร์ OpenADR

การดำเนินการตอบสนองด้านโหลดรวมถึงการจัดการไฟฟ้าบนโครงข่ายไฟฟ้าอัจฉริยะในปัจจุบันมีลักษณะความซับซ้อนในการดำเนินการทั้งแบบกึ่งอัตโนมัติและแบบอัตโนมัติ กล่าวคือ จะต้องมีการส่งการความต้องการการตอบสนองด้านโหลดจากศูนย์สั่งการ (Server) ไปยังลูกข่าย (Client) เพื่อให้เกิดการจัดการปรับเปลี่ยนการใช้งานโหลดไฟฟ้า หรือปลดโหลดไฟฟ้าออกจากระบบ โดยมาตรฐานที่ใช้กันอยู่ในระบบโครงข่ายไฟฟ้าอัจฉริยะในปัจจุบันมีอยู่หลายมาตรฐานด้วยกัน อาทิ มาตรฐาน IEC 61850 (Substation Automation) มาตรฐาน Smart Energy Protocol (SEP) ซึ่งมักใช้คู่กับมาตรฐานการสื่อสาร ZigBee หรือ *Open Automated Demand Response (OpenADR)* ที่ได้รับการสนับสนุนอย่างกว้างขวางในอุตสาหกรรมเป็นต้น

มาตรฐาน OpenADR มีถิ่นกำเนิดในมลรัฐแคลิฟอร์เนีย ประเทศสหรัฐอเมริกา ซึ่งถือเป็นต้นแบบของการดำเนินการตอบสนองด้านโหลดแบบอัตโนมัติที่มีชื่อเสียง ปัจจุบัน OpenADR นั้นได้พัฒนาจากรุ่น 1.0 มาเป็นรุ่น 2.0 แล้ว ซึ่งในอนาคตจะถูกพัฒนาไปเป็นมาตรฐาน IEC ต่อไป โดยรายละเอียดของมาตรฐานสามารถแยกออกมาได้ 2 ระดับคือ OpenADR 2.0a สำหรับอุปกรณ์ง่าย ๆ ที่ไม่ต้องการฟังก์ชันการทำงานที่ซับซ้อน ส่วน OpenADR 2.0b สำหรับระบบและอุปกรณ์ที่ต้องการฟังก์ชันการทำงานแบบเต็มรูปแบบ OpenADR 2.0a สามารถทำให้การใช้งาน OpenADR ในอุปกรณ์เล็ก ๆ เช่นเครื่องใช้ไฟฟ้าภายในบ้านนั้นสามารถทำได้โดยที่มีต้นทุนไม่สูงมาก สำหรับมาตรฐาน OpenADR จึงมีจุดเด่นที่สำคัญในการเป็นมาตรฐานของโครงข่ายไฟฟ้าอัจฉริยะในด้านการตอบสนองด้านโหลด นั่นคือมีฟังก์ชันการทำงานด้านการตอบสนองด้านโหลดอย่างครบถ้วน นอกจากนี้

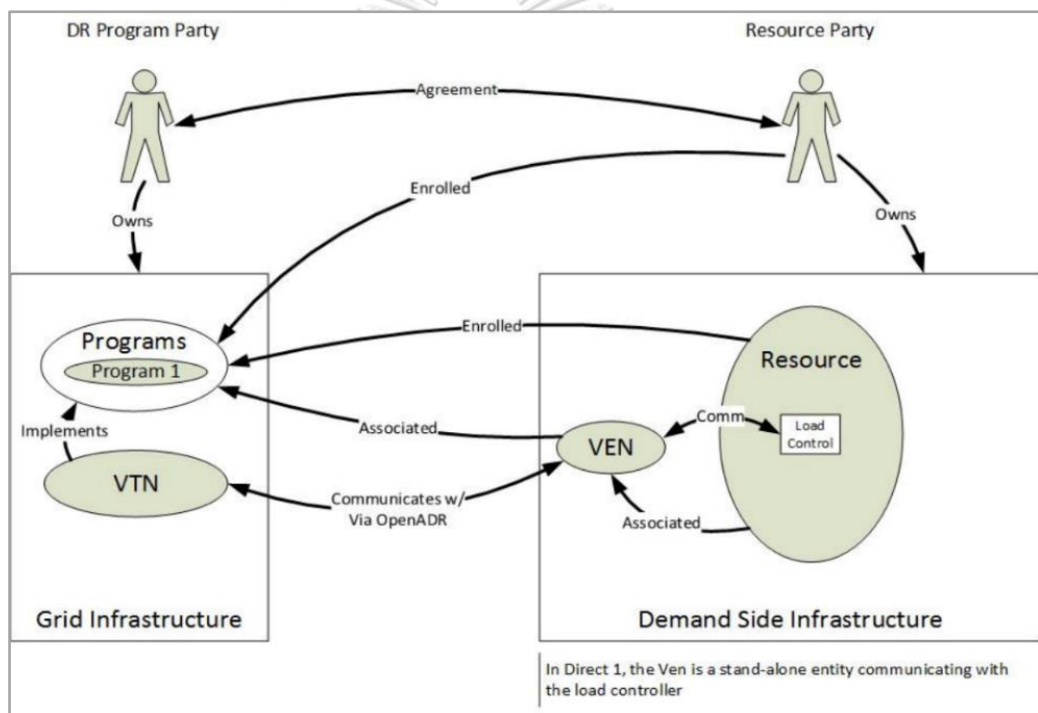
OpenADR ได้ถูกออกแบบให้สามารถรองรับระบบโครงข่ายไฟฟ้าอัจฉริยะทั้งหมด ตั้งแต่เซิร์ฟเวอร์ การบริหารจัดการพลังงาน (EMS Server) ของระบบโครงข่ายไฟฟ้าหลัก ผู้รวบรวมโหลด ระบบบริหารจัดการพลังงานในบ้านเรือน อาคาร หรือโรงงานอุตสาหกรรม (HEMS/BEMS/FEMS) มิเตอร์อัจฉริยะ รวมถึงเครื่องใช้ไฟฟ้าในบ้านเรือน มาตรฐาน OpenADR ได้ถูกทดลองใช้ในมลรัฐแคลิฟอร์เนีย โดยได้ผลลัพธ์ที่น่าพึงพอใจ ปัจจุบันได้ถูกบรรจุอยู่ในรายการมาตรฐานหลักทางด้านสมรรถนะของ NIST ประเทศสหรัฐอเมริกา นอกจากนี้ OpenADR ได้รับการยอมรับแล้วจากหลายภาคส่วน อาทิ สหภาพยุโรป เกาหลี ญี่ปุ่น ปัจจุบันได้มีจัดตั้งพันธมิตรมาตรฐาน OpenADR (OpenADR Alliance) ซึ่งประกอบด้วยสมาชิกจำนวนมาก บางส่วนถือได้ว่าเป็นผู้จำหน่ายเทคโนโลยีระดับโลก เช่น Google, Schneider, Mitsubishi, Siemens, Honeywell, Emerson, Alstom, Hitachi, Toshiba, Fuji Electric เป็นต้น

2.3.1.1 การปรับใช้รูปแบบตามสถานการณ์ (Deployment Scenarios)

สถานการณ์ที่สามารถปรับใช้มาตรฐาน OpenADR ได้นั้นสามารถเป็นไปได้หลากหลายรูปแบบ [10] ขึ้นอยู่กับส่วนประกอบต่าง ๆ ทั้งโหนดต้นเสมือน (Virtual Top Node; VTN) จำนวนโหนดปลายเสมือน (Virtual End Node; VENs) จำนวนโหลดและทรัพยากร (Resources) รวมถึงสภาพแวดล้อมอื่น ๆ โดยรอบประกอบเข้าด้วยกันจนเกิดเป็นสถานการณ์ที่มีความซับซ้อนแตกต่างกันออกไป โดยสถานการณ์ที่จะแสดงต่อไปนี้คือการปรับใช้รูปแบบการตอบสนองด้านโหลดแบบอัตโนมัติให้เข้ากับสถานการณ์ต่าง ๆ ดังนี้

1. แบบตรงที่ 1 (Direct 1)

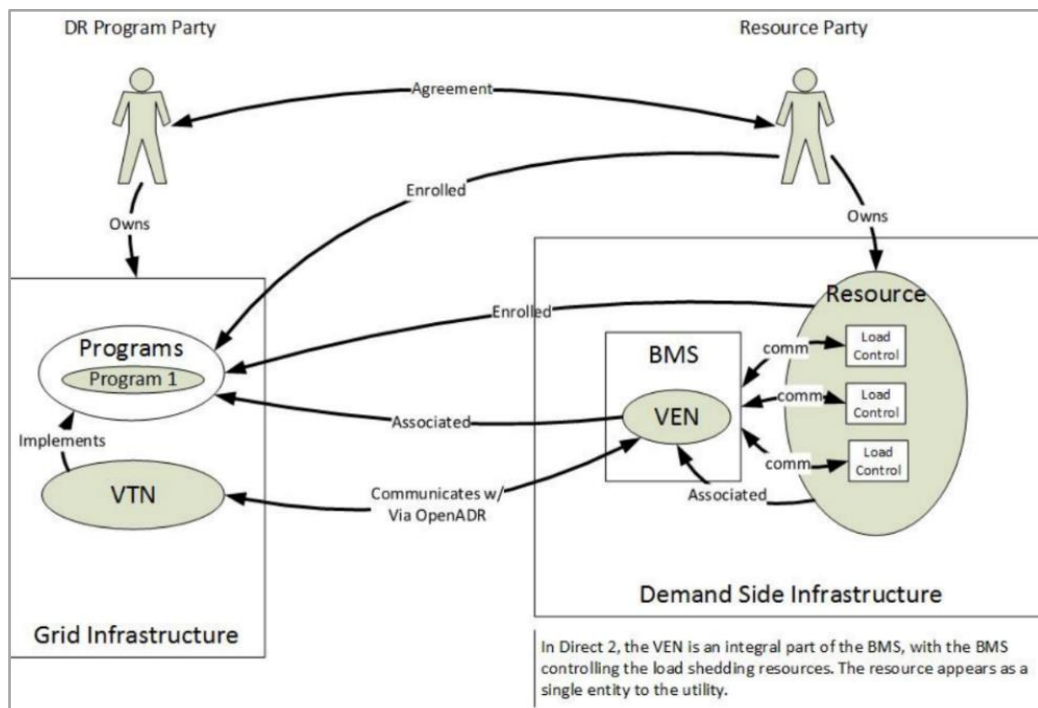
รูปแบบการสื่อสารตามสถานการณ์แบบตรงแบบที่ 1 นี้เป็นรูปแบบที่เรียบง่ายที่สุด เป็นการสื่อสารโดยตรงระหว่างฝ่ายควบคุมการตอบสนองด้านโหลดกับฝ่ายผู้ใช้ไฟฟ้า หากผู้ใช้ไฟฟ้าต้องการเข้าร่วมมาตรการดังกล่าว ผู้ใช้ไฟฟ้าจะต้องทำการลงทะเบียนอุปกรณ์โหลดของตนเข้ากับโหนดปลายเสมือนของผู้ใช้ไฟฟ้า เมื่อมีการส่งสัญญาณหรือสั่งการมาจากโหนดต้นเสมือนจากฝั่งผู้ให้บริการไฟฟ้าผ่านโพรโทคอล OpenADR แล้วนั้น โหนดปลายเสมือนจะทำการรับและแปลคำสั่ง จากนั้นจะส่งสัญญาณต่อไปยังตัวควบคุมโหลด (Load Controller) เพื่อควบคุมให้การทำงานเป็นไปตามที่ต้องการ ซึ่งโหนดปลายเสมือนและโหลดของผู้ใช้ไฟฟ้านั้นจะแยกจากกันอย่างอิสระจากโหนดปลายเสมือนและโหลดตัวอื่น ๆ ดังแสดงให้เห็นในรูปที่ 8



รูปที่ 8 โครงสร้างรูปแบบตรงแบบที่ 1 (Direct 1)

2. แบบตรงที่ 2 (Direct 2)

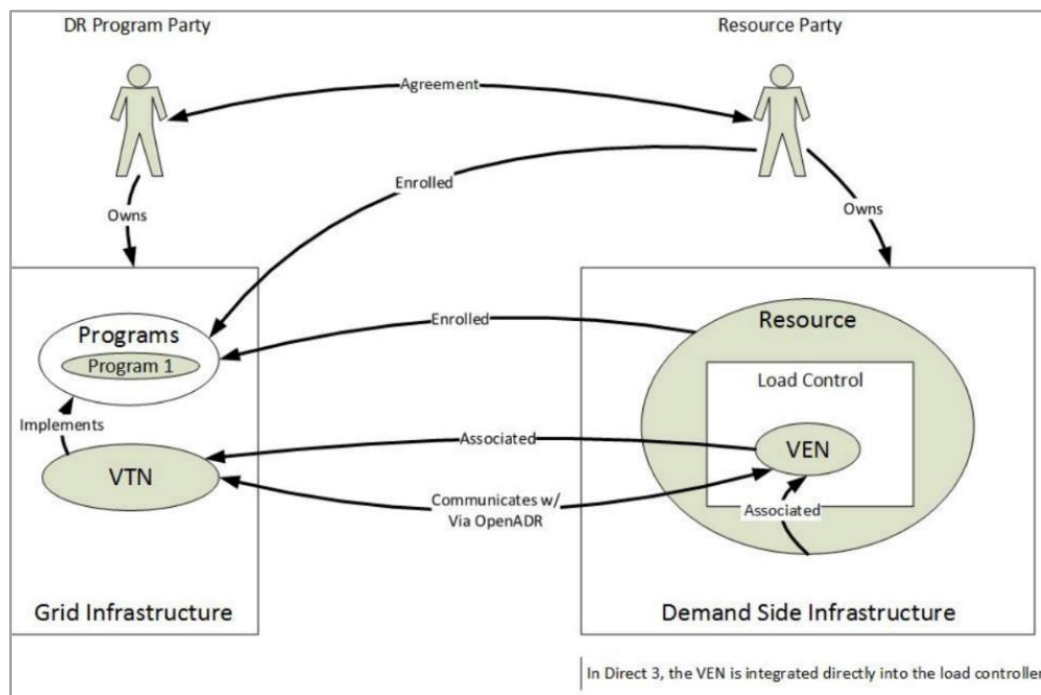
รูปแบบการสื่อสารตามสถานการณ์แบบตรงแบบที่ 2 จะมีความคล้ายคลึงกับแบบตรงแบบที่ 1 แตกต่างกันตรงที่การทำงานของโหนดปลายเสมือนจะสามารถทำหน้าที่ควบคุมอุปกรณ์ได้หลายตัว เปรียบเสมือนเป็นศูนย์กลางที่ทำหน้าที่ควบคุมการทำงานของอุปกรณ์ทุกตัว การทำงานลักษณะนี้จะเหมาะสำหรับการปรับใช้งานเข้ากับระบบบริหารจัดการภายในอาคาร (Building Management System; BMS) ที่ต้องควบคุมโหลดจำนวนหลาย ๆ ตัว ดังแสดงให้เห็นในรูปที่ 9



รูปที่ 9 โครงสร้างรูปแบบตรงแบบที่ 2 (Direct 2)

3. แบบตรงที่ 3 (Direct 3)

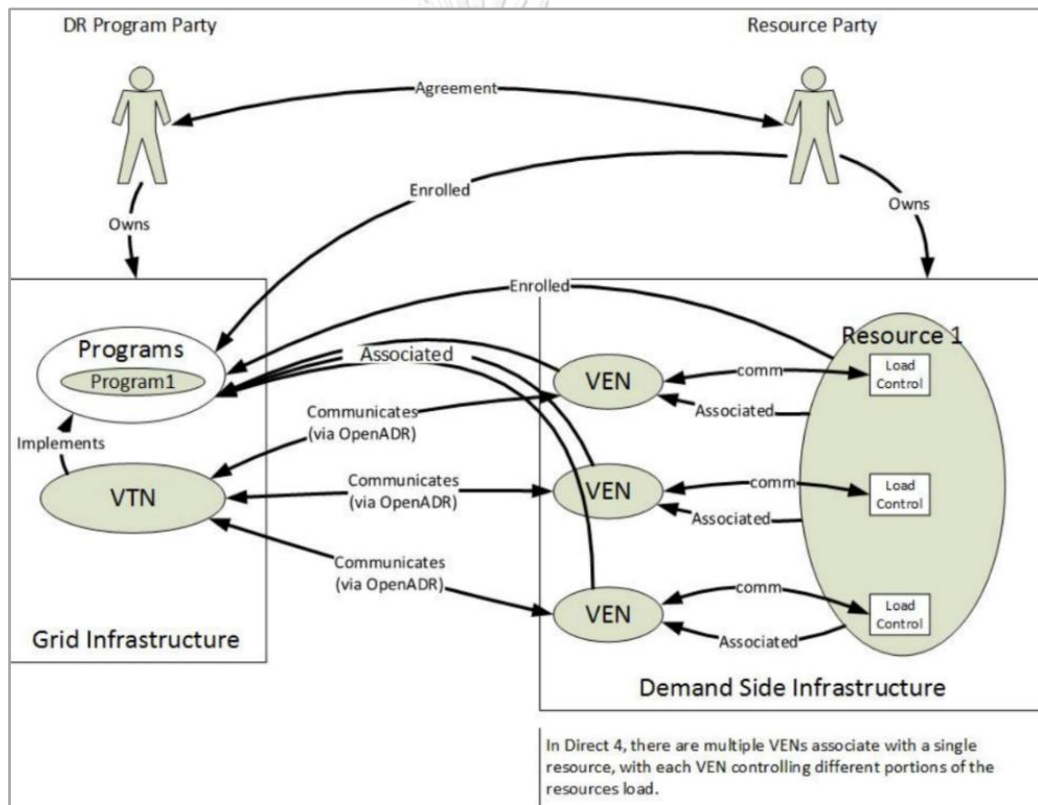
รูปแบบการสื่อสารตามสถานการณ์แบบตรงแบบที่ 3 จะมีความคล้ายคลึงกับแบบตรงแบบที่ 1 เช่นเดียวกัน จะแตกต่างที่อุปกรณ์บริเวณโหลดของผู้ใช้ไฟฟ้านั้นจะทำการฝังหนดปลายเสมือนที่สามารถสื่อสารโพรโทคอล OpenADR อยู่ในอุปกรณ์ชิ้นนั้นลงไป ทำให้ผู้บริการสามารถควบคุมโหลดได้โดยตรงดังแสดงให้เห็นในรูปที่ 10 มักใช้รูปแบบนี้กับการควบคุมโหลดชนิดระบบปรับอากาศ (Heating, Ventilation, and Air Conditioning; HVAC) เช่นตัวควบคุมอุณหภูมิ (Thermostat) เป็นต้น



รูปที่ 10 โครงสร้างรูปแบบตรงแบบที่ 3 (Direct 3)

4. แบบตรงที่ 4 (Direct 4)

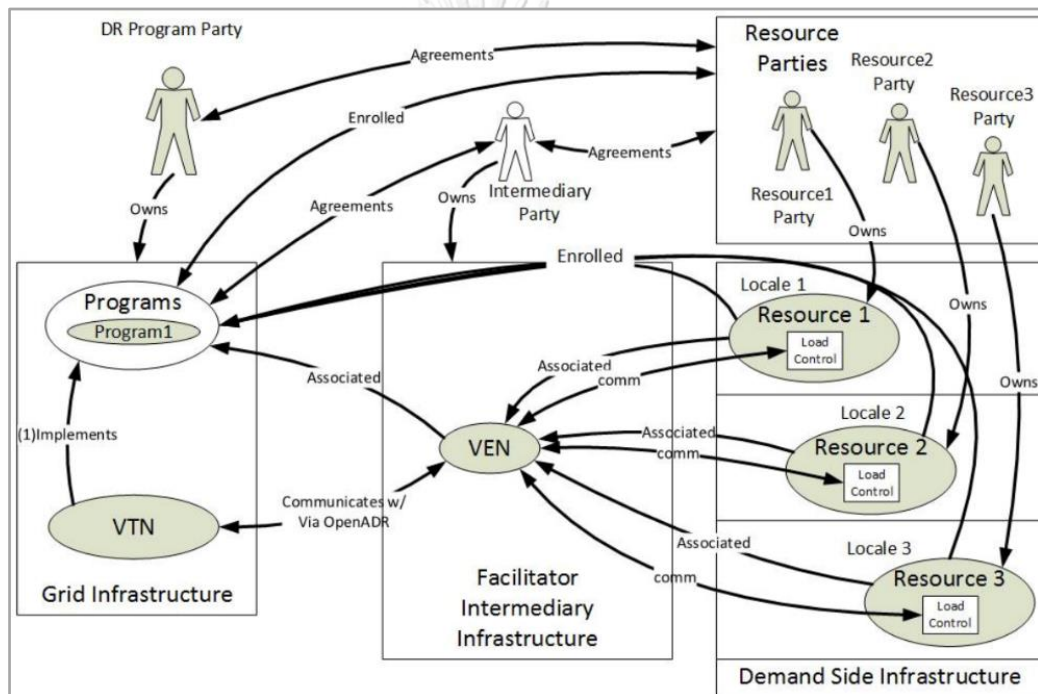
รูปแบบการสื่อสารตามสถานการณ์แบบตรงแบบที่ 4 นี้จะเป็นการผสมแบบตรงแบบที่ 1 และแบบตรงแบบที่ 2 เข้าด้วยกัน ตัวควบคุมโหลดแต่ละตัวจะมีโหนดปลายเสมือนเป็นของตัวเองซึ่งแต่ละชุดจะเป็นอิสระจากกัน แตกต่างกับแบบตรงแบบที่ 2 ที่มีโหนดปลายเสมือนตัวเดียวที่เป็นศูนย์กลางการควบคุม สัญญาณที่ถูกส่งมาจากโหนดต้นเสมือนนั้นจะถูกส่งไปยังโหนดปลายเสมือนทุกตัวในลักษณะเดียวกันทั้งหมดดังแสดงให้เห็นในรูปที่ 11 โครงสร้างการสื่อสารแบบนี้เหมาะสำหรับระบบบริหารจัดการภายในอาคารที่มีตัวควบคุมอยู่ในแต่ละชั้น โดยแต่ละชั้นจะมีโหนดปลายเสมือนเป็นของตัวเอง ไม่มีการรวมศูนย์แบบรูปแบบตรงแบบที่ 2



รูปที่ 11 โครงสร้างรูปแบบตรงแบบที่ 4 (Direct 4)

5. แบบผู้ให้บริการ (Facilitator)

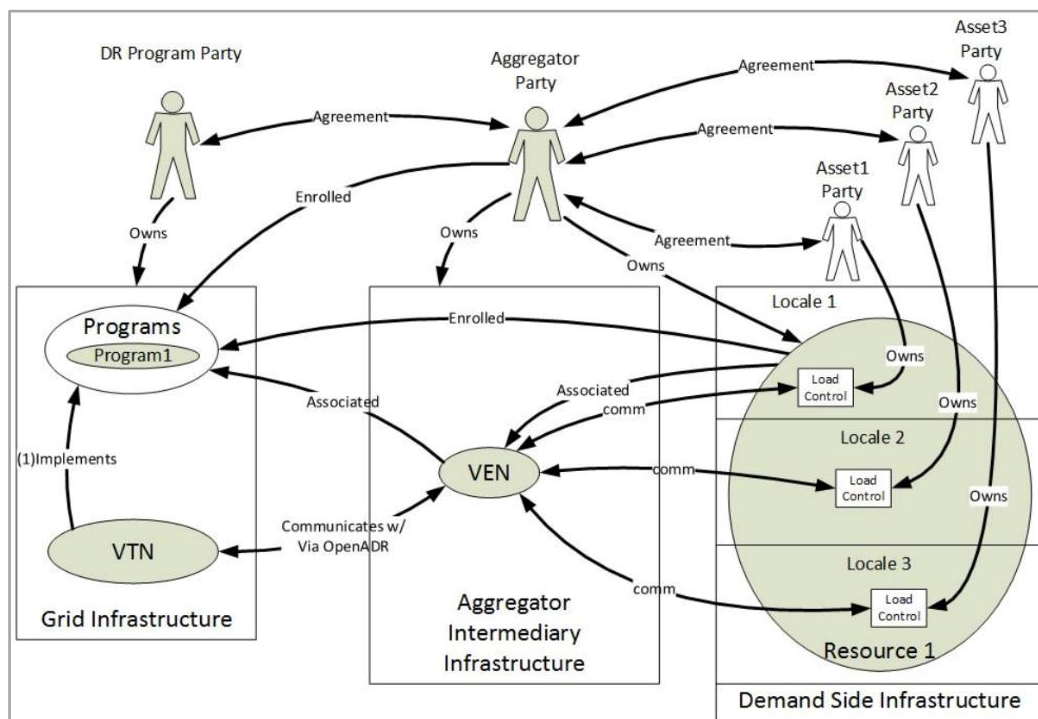
รูปแบบการสื่อสารตามสถานการณ์แบบผู้ให้บริการนี้ จะมีผู้ให้บริการคอยอำนวยความสะดวกและช่วยบริหารจัดการโหลดให้กับผู้ใช้ไฟฟ้า เปรียบเสมือนเป็นคนกลางที่มีหนอตปลายเสมือนที่คอยควบคุมสั่งการอุปกรณ์ให้กับผู้ใช้ไฟฟ้าที่เข้าร่วมมาตรการ จะแตกต่างจากรูปแบบปกติที่ผู้ใช้ไฟฟ้าจะต้องมีหนอตปลายเสมือนเป็นของตัวเองเพื่อใช้ในการสื่อสารและควบคุมอุปกรณ์ต่าง ๆ เพียงแต่ผู้ใช้ไฟฟ้ายังจำเป็นที่จะต้องลงทะเบียนในระบบด้วยตัวเองโดยตรง ทำให้ผู้ให้บริการจะมองผู้ใช้ไฟฟ้าแยกออกเป็นรายบุคคลดังแสดงให้เห็นดังรูปที่ 12 รูปแบบโครงสร้างลักษณะนี้จะเหมาะสำหรับการควบคุมอุปกรณ์ในภาคอุตสาหกรรม หรือบริษัทที่ให้บริการด้านพลังงาน เป็นต้น



รูปที่ 12 โครงสร้างรูปแบบผู้ให้บริการ (Facilitator)

6. แบบผู้รวบรวมข้อมูล (Aggregator)

รูปแบบการสื่อสารตามสถานการณ์แบบผู้รวบรวมนี้จะคล้ายกับรูปแบบผู้ให้บริการ แต่จุดที่แตกต่างกันคือผู้รวบรวมข้อมูลหรือคนกลางนี้จะเป็นคนทำหน้าที่ต่าง ๆ แทนผู้ใช้ไฟฟ้าทั้งหมด ที่ควบคุมสั่งการอุปกรณ์รวมถึงการลงทะเบียนของผู้ใช้ไฟฟ้าทุกราย ซึ่งผู้ให้บริการไฟฟ้าจะมองว่าเป็นผู้ใช้ไฟรายเดียวเนื่องจากกระบวนการทั้งหมดจะผ่านผู้รวบรวมข้อมูลเท่านั้น แตกต่างจากแบบผู้ให้บริการที่จะมองเห็นเป็นรายบุคคล แน่แน่นอนว่าผู้ให้บริการไม่สามารถรู้ข้อมูลรายบุคคลของผู้ใช้ไฟฟ้ารวมถึงลักษณะการบริหารจัดการโหลดของผู้รวบรวมอีกด้วยดังที่แสดงในรูปที่ 13



รูปที่ 13 โครงสร้างรูปแบบผู้รวบรวมข้อมูล (Aggregator)

2.3.1.2 ประเภทของโปรแกรมการตอบสนองด้านโหลดที่รองรับ (Demand Response Program Type)

จากที่ได้กล่าวถึงประเภทของการตอบสนองด้านโหลดไปในหัวข้อ 2.2.2 แล้วนั้น การสื่อสารด้วยโพรโทคอล OpenADR มีการรองรับโปรแกรมการตอบสนองด้านโหลดอยู่หลายประเภท [10] ได้แก่

- มาตรการอัตราค่าไฟฟ้าช่วงวิกฤต (Critical Peak Pricing) เป็นมาตรการสร้างแรงจูงใจจากอัตราหรือราคาค่าไฟฟ้าที่ตั้งไว้ล่วงหน้าเป็นจำนวนชั่วโมงหรือวันตามที่กำหนดไว้ เพื่อช่วยลดการใช้พลังงานในช่วงเวลาที่คาดว่าจะมีการใช้พลังงานสูงสุดของวัน โดยผู้ที่ไม่สามารถเลี่ยงการใช้ไฟในช่วงเวลาช่วงวิกฤตได้จะต้องจ่ายค่าพลังงานที่สูงกว่าปกติ
- มาตรการประมูลกำลังไฟฟ้า (Capacity Bidding Program) เป็นมาตรการที่กำหนดเงื่อนไขเพื่อให้ผู้เข้าร่วมเสนอปริมาณการลดการใช้พลังงานในราคาที่เฉพาะเจาะจง โดยมีวัตถุประสงค์หลักคือเพื่อวางแผนสำหรับลดการผลิตไฟฟ้าในช่วงเวลาที่มีต้นทุนสูง
- มาตรการควบคุมอุณหภูมิหรือโหลดโดยตรง (Thermostat Program/Direct Load Control) เป็นมาตรการที่ทำการควบคุมเครื่องใช้ไฟฟ้าระยะไกลโดยที่ไม่แจ้งให้ทราบล่วงหน้าเช่น เครื่องปรับอากาศ ซึ่งมาตรการนี้เหมาะสำหรับสถานประกอบการขนาดเล็กหรือที่พักอาศัย
- มาตรการการตอบสนองฉับพลันหรือมาตรการกำลังผลิตเสริมความมั่นคง (Fast DR Dispatch/Ancillary Services Program) เป็นมาตรการที่ช่วยรักษาเสถียรภาพของระบบด้วยการลดการใช้ไฟฟ้าในกรณีฉุกเฉินหรือเกิดสถานะที่ระบบผิดปกติ ผู้ที่เข้าร่วมจะได้รับผลตอบแทนสูงในอัตราพิเศษ โดยที่มาตรการนี้จะทำการตอบสนองด้านโหลดโดยอัตโนมัติหรือในทันทีที่เกิดเหตุการณ์ เพื่อป้องกันไม่ให้เกิดความล้มเหลวของระบบไฟฟ้าทั้งหมด
- มาตรการการตอบสนองสำหรับยานพาหนะไฟฟ้า (Electric Vehicle (EV) DR Program) เป็นมาตรการตอบสนองความต้องการในการชาร์จยานพาหนะ โดยการปรับราคาค่าไฟ เพื่อจูงใจให้ผู้ใช้บริการปรับเปลี่ยนเวลาในการชาร์จของยานพาหนะ
- มาตรการการตอบสนองสำหรับการใช้พลังงานทางเลือก (Distributed Energy Resources (DER) DR Program) เป็นมาตรการการตอบสนองด้านโหลดสำหรับผู้ที่ใช้ไฟฟ้าที่ต้องการใช้ไฟจากภายนอกที่ต่อเข้ากับโครงข่ายไฟฟ้า เป็นไปอย่างราบรื่น

2.3.1.3 ความสัมพันธ์ระหว่างสถานการณ์การใช้งานโปรแกรมกับประเภทของการตอบสนองด้านโหลด

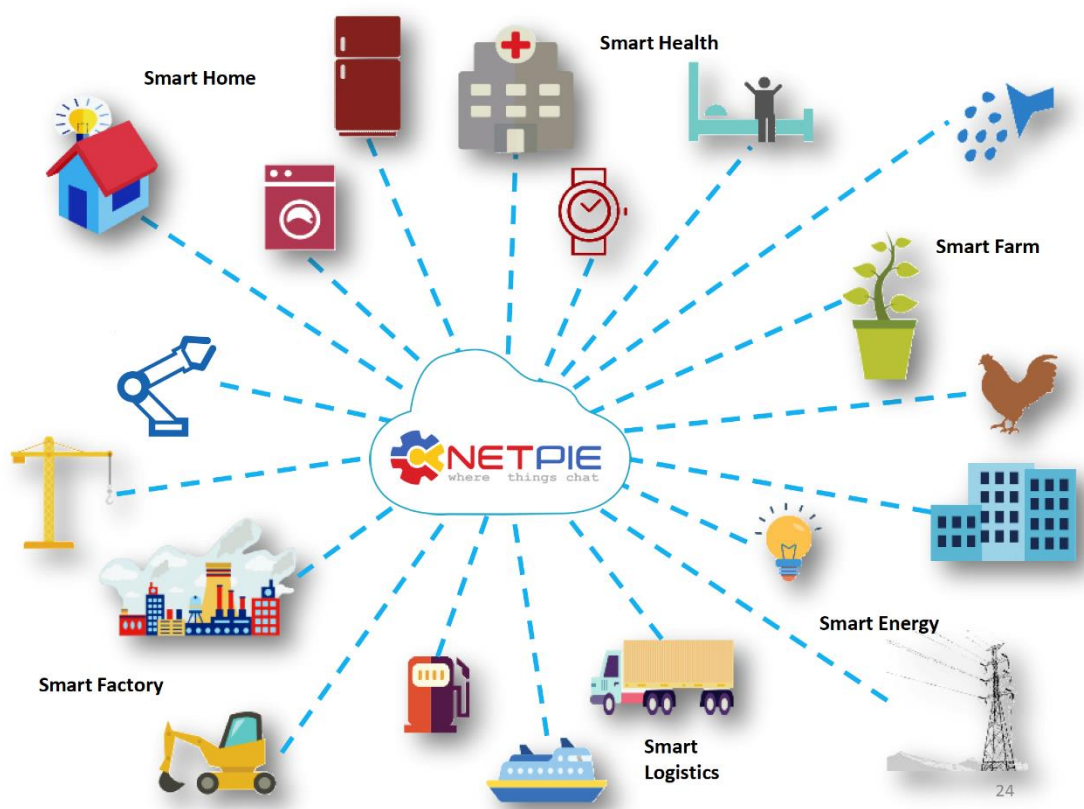
การใช้งานโปรแกรมและประเภทของการตอบสนองด้านโหลดจะมีความสัมพันธ์ดังตารางที่ 1 ซึ่งสามารถสังเกตได้ว่าการใช้งานตามสถานการณ์รูปแบบตรงและรูปแบบผู้ให้บริการจะสนับสนุนประเภทของการตอบสนองด้านโหลดที่คล้ายคลึงกัน เนื่องจากผู้ใช้งานไฟฟ้าจะต้องทำการตอบรับการตอบสนองที่ถูกส่งมาจากผู้ให้บริการก่อน จึงจะเริ่มดำเนินการตอบสนองด้านโหลดได้ แตกต่างจากรูปแบบผู้รวบรวมข้อมูลที่ได้ทำข้อตกลงไว้ล่วงหน้าแล้ว จึงสามารถรองรับประเภทการตอบสนองแบบฉับพลันได้

ตารางที่ 1 ความสัมพันธ์ของการใช้งานโปรแกรมกับประเภทของโปรแกรมการตอบสนองด้านโหลด

	Deployment Scenario		
DR Template	Direct 1, 2, 3, 4	Facilitator	Aggregator
CPP Program	●	●	
Capacity Bidding Program			●
Direct Load Control		●	
Fast DR Dispatch			●
EV DR Program	●	●	
DER DR Program	●	●	

2.3.2 ซอฟต์แวร์ไอโอทีคลาวด์แพลตฟอร์ม (NETPIE)

ในปัจจุบันเทคโนโลยีอินเทอร์เน็ตและสรรพสิ่ง (Internet Of Things; IoT) เข้ามามีบทบาทที่สำคัญในชีวิตประจำวัน สามารถเพิ่มขีดจำกัดความสามารถของอุปกรณ์ อีกทั้งประยุกต์ใช้งานได้หลากหลาย จุดเด่นของอินเทอร์เน็ตและสรรพสิ่งคือการสื่อสารและเชื่อมต่อกันได้ผ่านโครงข่ายการสื่อสารทั้งแบบมีสายและไร้สาย โดยแต่ละอุปกรณ์จะสามารถระบุตัวตน รับรู้สภาพแวดล้อม แสดงผลควบคุม และสามารถทำงานด้วยกันได้

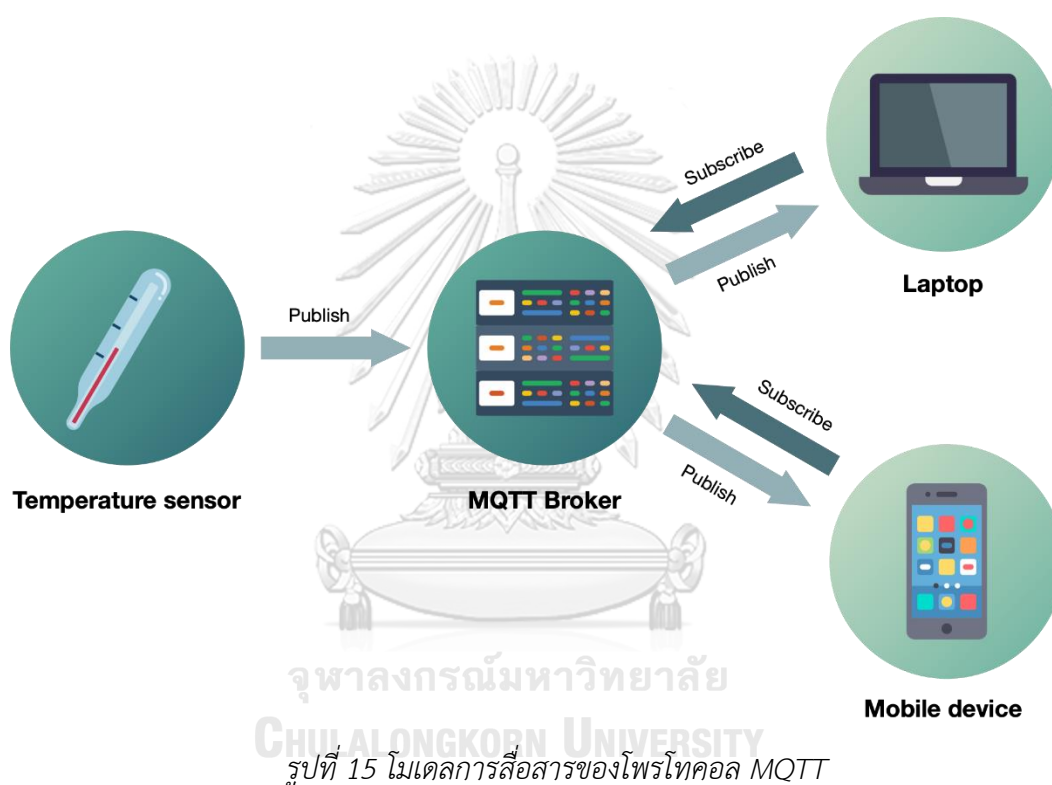


รูปที่ 14 การประยุกต์ใช้บริการ NETPIE เข้ากับอินเทอร์เน็ตและสรรพสิ่งรูปแบบต่าง ๆ

NETPIE (Network Platform for Internet of Everything) [11] คือ คลาวด์แพลตฟอร์มที่ให้บริการด้านอินเทอร์เน็ตและสรรพสิ่ง เป็นเครื่องมือที่ช่วยอำนวยความสะดวกให้กับนักพัฒนา นอกจากนี้ไม่จำเป็นต้องดูแลระบบเซิร์ฟเวอร์ หรือระบบสื่อสารใด ๆ แล้ว NETPIE คือทางเลือกหนึ่งที่ช่วยให้นักพัฒนาซอฟต์แวร์สามารถเข้าถึงอุปกรณ์อิเล็กทรอนิกส์ต่าง ๆ ได้ง่ายยิ่งขึ้น NETPIE เปิดให้บริการครั้งแรกเมื่อวันที่ 16 กันยายน พ.ศ.2558 โดยทีมวิจัยพัฒนาของเนคเทค-สวทช. โดยเป้าหมายหลักของ NETPIE คือเพื่อผลักดันและส่งเสริมให้นักพัฒนาและอุตสาหกรรมขนาดย่อมสามารถพัฒนาเทคโนโลยีที่สามารถประยุกต์ใช้ทางด้านอินเทอร์เน็ตและสรรพสิ่ง เพื่อเชื่อมต่อ

อุปกรณ์ต่าง ๆ เข้าด้วยกันโดยใช้อินเทอร์เน็ตเป็นสื่อกลาง โดย NETPIE นั้นเปรียบเสมือนมิดเดิลแวร์ที่เป็นจุดนัดพบของอุปกรณ์ต่าง ๆ (Things) ให้สามารถติดต่อสื่อสารถึงกันได้ดังแสดงในรูปที่ 14

ในส่วนของการสื่อสารระหว่างอุปกรณ์ที่เชื่อมต่อกับบริการของ NETPIE นั้นจะใช้ MQTT Protocol (Message Queuing Telemetry Transport) ผ่านวิธีการส่งข้อความแบบ Publish/Subscribe ดังแสดงให้เห็นในรูปที่ 15 MQTT นั้นถูกนำมาใช้แพร่หลายในการสื่อสารแบบ M2M (Machine to Machine) เนื่องจากเหมาะสมกับอุปกรณ์ที่มีขนาดเล็ก กินพลังงานต่ำและสามารถสื่อสารระยะไกลได้ดี



รูปที่ 15 โมเดลการสื่อสารของโพรโทคอล MQTT

ประโยชน์ที่โดดเด่นของ NETPIE ประการแรกคือช่วยให้อุปกรณ์ต่าง ๆ สามารถสื่อสารถึงกันได้โดยไม่ต้องคำนึงว่าอุปกรณ์นั้นจะอยู่ที่ใด หากอุปกรณ์นั้นเชื่อมต่ออินเทอร์เน็ตอยู่เสมอ NETPIE จะช่วยดูแลการเชื่อมต่อสื่อสารให้ทั้งหมด ทำให้ช่วยลดการใช้ทรัพยากรการเชื่อมต่อลงได้ นอกจากนี้ผู้ใช้ยังสามารถตัดปัญหาการเข้าถึงอุปกรณ์ระยะไกล อย่างการ Fixed IP address หรือแม้กระทั่งการทำ Port forwarding ซึ่งวิธีการที่กล่าวไปนั้นล้วนมีความยุ่งยากในขั้นตอนการติดตั้งหรือตั้งค่าทั้งสิ้น ประโยชน์ที่สำคัญอีกประการหนึ่งคือ NETPIE ยังช่วยเพิ่มความยืดหยุ่นต่อการขยายระบบ เนื่องจาก NETPIE นั้นใช้สถาปัตยกรรมแบบคลาวด์เซิร์ฟเวอร์โดยสมบูรณ์ ทำให้เกิดความคล่องตัวในการขยายระบบ ดังนั้นผู้พัฒนาจึงไม่ต้องกังวลกับการขยายตัวเพื่อรับโหลดที่เพิ่มขึ้นในระบบอีกต่อไป

2.3.3 ซอฟต์แวร์ฐานข้อมูลเอสคิวแอลไลต์ (SQLite)

ฐานข้อมูลเอสคิวแอลไลต์ เป็นฐานข้อมูลขนาดเล็กที่ได้รับความนิยมอย่างมาก เนื่องจากเป็นซอฟต์แวร์ประมวลผลฐานข้อมูลที่มีขนาดเล็กที่สามารถทำงานได้ด้วยตัวเอง (Standalone) มีความรวดเร็ว ใช้งานง่าย และมีความน่าเชื่อถือสูง มักถูกนำไปประยุกต์ใช้กับอุปกรณ์ขนาดเล็กอย่างอุปกรณ์ฝังตัว หรือโทรศัพท์เคลื่อนที่ ซึ่งรูปแบบของไฟล์เอสคิวแอลไลต์นั้นมีความเสถียร ทำให้สามารถทำงานข้ามแพลตฟอร์มได้ นอกจากนี้ภาษาโปรแกรมที่ใช้ในเอสคิวแอลไลต์นั้นคือภาษาเอสคิวแอล (Structured Query Language; SQL) ซึ่งเป็นภาษาที่นิยมใช้ในโปรแกรมฐานข้อมูลอื่น ๆ เช่น MySQL หรือ PostgreSQL เป็นต้น



2.4 หลักการคำนวณกำลังไฟฟ้าพื้นฐานทั่วไป

ค่ากำลังไฟฟ้าขณะหนึ่ง (Instantaneous Power) เป็นค่าที่แสดงให้เห็นถึงค่ากำลังไฟฟ้าที่ถูกใช้ไป ณ เวลาใดเวลาหนึ่ง หากค่ากำลังไฟฟ้าขณะหนึ่งนี้มีปริมาณที่สูงเกินขีดจำกัดของตัวอุปกรณ์ไฟฟ้า อาจส่งผลให้เกิดความเสียหายต่ออุปกรณ์ ดังนั้นค่ากำลังไฟฟ้าขณะหนึ่งจึงมีความสำคัญต่อการออกแบบวงจรไฟฟ้า โดยค่ากำลังขณะหนึ่งเมื่อเขียนในโดเมนเวลา (Time domain) คือ

$$p(t) = v(t)i(t) \quad (2.1)$$

โดยที่ $v(t)$ คือ ค่าแรงดันในโดเมนเวลา มีหน่วยเป็นโวลต์ (Volt; V)

$i(t)$ คือ ค่ากระแสในโดเมนเวลา มีหน่วยเป็นแอมแปร์ (Ampere; A)

ซึ่งค่าแรงดันและกระแสนั้นสามารถเขียนเป็นสมการได้ดังนี้

$$v(t) = V_m \cos(\omega t + \theta_v) \quad (2.2)$$

$$i(t) = I_m \cos(\omega t + \theta_i) \quad (2.3)$$

เมื่อนำสมการแรงดันและกระแสไปแทนในสมการแทนค่าลงในสมการที่ (2.1) จะได้

$$p(t) = V_m I_m \cos(\omega t + \theta_v) \cos(\omega t + \theta_i) \quad (2.4)$$

จากคุณสมบัติของตรีโกณมิติ

$$\cos(A) \cos(B) = \frac{1}{2} [\cos(A - B) + \cos(A + B)] \quad (2.5)$$

จะได้ว่า

$$p(t) = \frac{1}{2} V_m I_m [\cos(\theta_v - \theta_i) + \cos(2\omega t + \theta_i + \theta_v)] \quad (2.6)$$

$$p(t) = \frac{V_m I_m}{2} \cos(\theta_v - \theta_i) + \frac{V_m I_m}{2} \cos(2\omega t + \theta_i + \theta_v) \quad (2.7)$$

จากสมการจะเห็นว่าค่ากำลังไฟฟ้าขณะหนึ่งนั้นมีทั้งค่าคงที่และค่าที่แปรผันตามเวลา โดยส่วนที่เป็นค่าคงที่นั่นคือค่ากำลังไฟฟ้าที่ได้จะขึ้นอยู่กับค่าความต่างเฟสระหว่างแรงดันและกระแส ซึ่งมีขนาดสูงสุดที่ $V_m I_m / 2$ และมีค่าต่ำสุดที่ 0 ส่วนของค่าที่แปรผันตามเวลานั้น ฟังก์ชันจะมีลักษณะเป็นรูปไซน์ที่มีความถี่เป็น 2 เท่าของความถี่แรงดันหรือกระแส

2.4.1 ค่ากำลังไฟฟ้าจริง

ค่ากำลังไฟฟ้าจริง (Active Power; P) หรือค่ากำลังไฟฟ้าเฉลี่ย (Average Power) เป็นกำลังไฟฟ้าที่จริงนำไปใช้งานได้จริง หรือเรียกได้ว่าเป็นกำลังไฟฟ้าที่ทำให้เกิดงานจริง (Real work) มีหน่วยเป็นวัตต์ (Watt; W) มักจะเกิดกับอุปกรณ์จำพวกตัวต้านทาน หรือโหลดประเภทตัวต้านทาน (Resistive Load) โดยการหาค่ากำลังไฟฟ้าจริงนั้นสามารถทำได้ด้วยการหาค่าเฉลี่ยของ $p(t)$ ในช่วงเวลา 1 คาบ สามารถแสดงได้ดังสมการที่ (2.8)

$$P = \frac{1}{T} \int_0^T p(t) dt \quad (2.8)$$

โดยที่ T คือ คาบของสัญญาณ

เมื่อนำสมการที่ (2.7) มาแทนค่าลงในสมการที่ (2.8) จะได้

$$P = \frac{1}{T} \int_0^T \left(\frac{V_m I_m}{2} \cos(\theta_v - \theta_i) + \frac{V_m I_m}{2} \cos(2\omega t + \theta_i + \theta_v) \right) dt \quad (2.9)$$

$$P = \frac{1}{T} \int_0^T \frac{V_m I_m}{2} \cos(\theta_v - \theta_i) dt + \frac{1}{T} \int_0^T \frac{V_m I_m}{2} \cos(2\omega t + \theta_i + \theta_v) dt \quad (2.10)$$

$$P = \frac{1}{T} \frac{V_m I_m}{2} \cos(\theta_v - \theta_i) \int_0^T dt + \frac{1}{T} \frac{V_m I_m}{2} \sin(2\omega t + \theta_i + \theta_v) \Big|_0^T \quad (2.11)$$

แทนค่า $T = \frac{\pi}{\omega}$ ลงในสมการ จะได้

$$P = \frac{V_m I_m}{2} \cos(\theta_v - \theta_i) + \frac{1}{\pi/\omega} \frac{V_m I_m}{2} [\sin(2\pi + \theta_i + \theta_v) - \sin(\theta_i + \theta_v)] \quad (2.12)$$

$$P = \frac{V_m I_m}{2} \cos(\theta_v - \theta_i) \quad (2.13)$$

$$P = \frac{V_m}{\sqrt{2}} \frac{I_m}{\sqrt{2}} \cos(\theta_v - \theta_i) \quad (2.14)$$

$$P = V_{rms} I_{rms} \cos(\theta_v - \theta_i) \quad (2.15)$$

สามารถเขียนในรูปอย่างง่ายได้เป็น

$$P = V_{rms} I_{rms} \cos \theta \quad ; \theta = (\theta_v - \theta_i) \quad (2.16)$$

2.4.2 ค่ากำลังไฟฟ้ารีแอกทีฟ

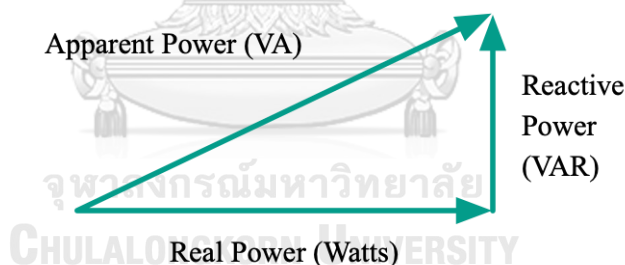
ค่ากำลังไฟฟ้ารีแอกทีฟ (Reactive Power; Q) เป็นกำลังไฟฟ้าที่เกิดจากแรงดันและกระแสไฟฟ้าในตัวเก็บประจุหรือตัวเหนี่ยวนำ ซึ่งเป็นกำลังไฟฟ้าส่วนที่ไหลรับเข้าไปและมีการจ่ายคืนสู่แหล่งจ่าย โดยกระแสไฟฟ้าในตัวเหนี่ยวนำจะล้าหลังแรงดันไฟฟ้า ส่วนกระแสไฟฟ้าในตัวเก็บประจุจะนำหน้าแรงดันไฟฟ้า ซึ่งจะมีค่าระหว่าง 1 - 90 องศา กำลังไฟฟ้าส่วนนี้เป็นส่วนกำลังไฟฟ้าที่ไม่ทำให้เกิดงาน มีหน่วยเป็นวาร์ (VAR) สามารถเขียนสมการในรูปอย่างง่ายได้ดังสมการที่ (2.17)

$$Q = V_{rms}I_{rms} \sin \theta \quad (2.17)$$

2.4.3 ค่ากำลังไฟฟ้าปรากฏ

กำลังไฟฟ้าปรากฏ (Apparent Power; S) คือค่าที่แสดงถึงกำลังไฟฟ้าโดยรวมทั้งหมดที่จ่ายให้กับวงจร มีหน่วยเป็นโวลต์แอมแปร์ (VA) โดยเป็นผลคูณระหว่างค่าที่วัดได้จากโวลต์มิเตอร์ไฟฟ้ากระแสสลับ ซึ่งวัดค่าออกมาได้เป็นค่าโวลต์อาร์เอ็มเอส และค่าที่วัดได้จากแอมมิเตอร์ไฟฟ้ากระแสสลับ ซึ่งวัดค่าออกมาได้เป็นแอมอาร์เอ็มเอส ซึ่งสามารถเขียนได้ดังสมการที่ (2.18)

$$S = V_{rms}I_{rms} = \sqrt{P^2 + Q^2} \quad (2.18)$$



รูปที่ 16 ค่าความสัมพันธ์ของกำลังไฟฟ้าในระบบไฟฟ้ากระแสสลับ

2.4.4 ค่าตัวประกอบกำลังไฟฟ้า

ค่าตัวประกอบกำลังไฟฟ้า (Power Factor; PF) คือพารามิเตอร์ที่บ่งบอกถึงประสิทธิภาพของวงจร สามารถแสดงให้เห็นถึงค่าประสิทธิภาพโดยมีค่าอยู่ระหว่าง 0 ถึง 1 โดยเมื่อพิจารณาว่าตัวประกอบกำลังไฟฟ้ายิ่งมีค่าเข้าใกล้ 1 มากเท่าไรยิ่งทำให้เครื่องใช้ไฟฟ้ามีประสิทธิภาพเท่านั้น ซึ่งสามารถหาค่าได้จากอัตราส่วนของกำลังไฟฟ้าจริงหารด้วยค่ากำลังไฟฟ้าปรากฏ ดังแสดงให้เห็นในสมการที่ (2.19)

$$PF = \cos \theta = \frac{P}{V_{rms}I_{rms}} = \frac{P}{S} \quad (2.19)$$

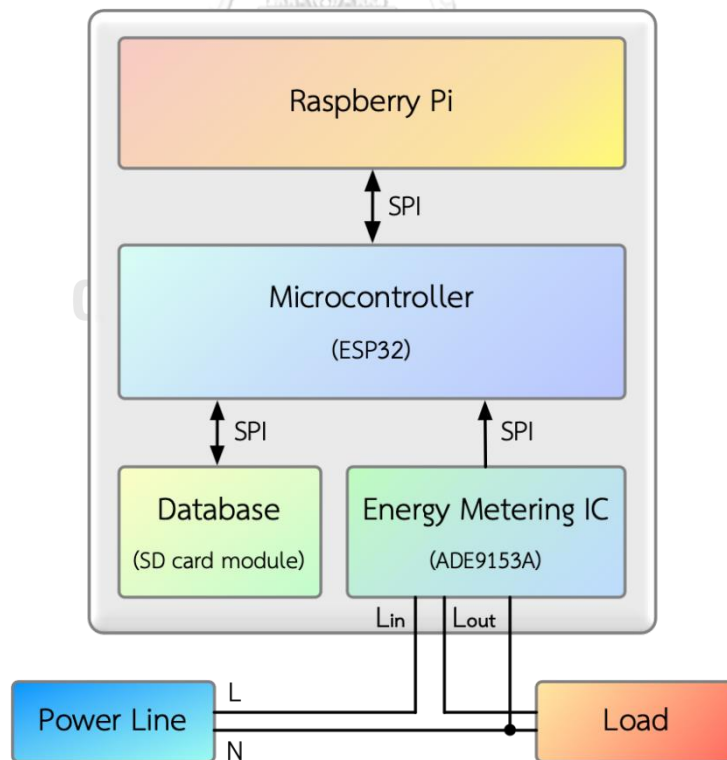
ค่าตัวประกอบกำลังไฟฟ้าในโหลดแต่ละชนิดมีความแตกต่างกัน สามารถพิจารณาโหลดออกได้เป็น 3 แบบคือ โหลดตัวต้านทาน (Resistive Load; R) โหลดตัวเหนี่ยวนำ (Inductive Load; L) และโหลดตัวเก็บประจุ (Capacitive Load; C) โดยโหลดตัวต้านทานจะมีค่าตัวประกอบกำลังไฟฟ้าเท่ากับ 1 และลักษณะความสัมพันธ์ของกระแสและแรงดันจะมีมูฟเฟสเดียวกัน แต่โหลดตัวเหนี่ยวนำและโหลดตัวเก็บประจุจะมีค่าตัวประกอบกำลังไฟฟ้าไม่เท่ากับ 1 โดยโหลดตัวเหนี่ยวนำนั้นจะมีตัวประกอบกำลังไฟฟ้าชนิดล่าช้า (Lagging Power Factor) ซึ่งเป็นภาวะที่กระแสไฟฟ้ามืดเฟสล่าช้าแรงดันไฟฟ้า ส่วนโหลดตัวเก็บประจุนั้นจะมีตัวประกอบกำลังไฟฟ้าชนิดนำหน้า (Leading Power Factor) ซึ่งเป็นภาวะที่กระแสไฟฟ้ามืดเฟสนำหน้าแรงดันไฟฟ้า



บทที่ 3

การออกแบบฮาร์ดแวร์ของมิเตอร์อัจฉริยะ

วิทยานิพนธ์ฉบับนี้จะแบ่งการออกแบบเป็นสองส่วนหลัก ๆ คือ การออกแบบฮาร์ดแวร์ และการออกแบบและพัฒนาซอฟต์แวร์ โดยในบทนี้จะนำเสนอการพัฒนาฮาร์ดแวร์มิเตอร์อัจฉริยะ ต้นแบบที่มีความสามารถในการวัดพลังงานทางไฟฟ้าเฟสเดียว (Single-Phase) ที่รองรับแรงดันพิกัดระหว่าง 180 V ถึง 265 V กระแสสูงสุดที่ 10 A ความถี่ 50 Hz โดยมีส่วนประกอบฮาร์ดแวร์ที่สำคัญหลัก ๆ ได้แก่ ไมโครคอนโทรลเลอร์ ไอซีวัดค่าพลังงานทางไฟฟ้า บอร์ดตราสเบอร์รี่พาย และหน่วยความจำภายนอก ดังแสดงให้เห็นในรูปที่ 17 ซึ่งโดยหลักการแล้ว ไอซีวัดค่าพลังงานนั้นจะทำการแปลงสัญญาณแอนะล็อกที่ได้รับจากตัวรับรู้ (Sensor) ไปเป็นสัญญาณดิจิทัล จากนั้นจะทำการประมวลผลเบื้องต้นเพื่อเตรียมความพร้อมของข้อมูลก่อนที่จะส่งไปยังไมโครคอนโทรลเลอร์เพื่อคำนวณค่าพลังงานทางไฟฟ้าต่อไป ข้อมูลที่คำนวณแล้วเสร็จจะถูกนำมาเก็บไว้ในหน่วยความจำในรูปแบบฐานข้อมูลซึ่งจะสามารถนำมาใช้ประโยชน์ได้ในภายหลัง ส่วนบอร์ดตราสเบอร์รี่พายจะคอยทำหน้าที่เป็นโหนดปลายเสมือนในการติดต่อสื่อสารด้วยโพรโทคอล OpenADR สำหรับการดำเนินการตอบสนองด้านโหลดในช่วงเวลาที่มีความต้องการการใช้ไฟฟ้าไม่สอดคล้องกับกำลังการผลิต

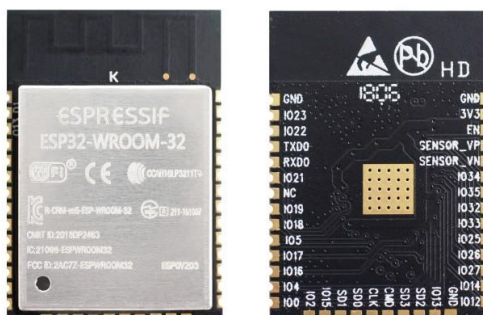


รูปที่ 17 แผนภาพบล็อกฮาร์ดแวร์ของมิเตอร์อัจฉริยะต้นแบบ

3.1 รายละเอียดอุปกรณ์ที่ใช้ในไมโครอัจฉริยะ

3.1.1 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ (Microcontroller) คืออุปกรณ์ที่ทำหน้าที่ตั้งสมองของวงจรอิเล็กทรอนิกส์ ใช้ในการสั่งการและควบคุมการทำงานของวงจรซึ่งมีความสามารถคล้ายกับคอมพิวเตอร์ขนาดเล็ก โดยภายในของไมโครคอนโทรลเลอร์นั้นจะมีส่วนประกอบหลักคือ หน่วยประมวลผลกลาง (Central Processing Unit; CPU), หน่วยความจำ (Memory), พอร์ตที่ใช้สำหรับติดต่อกับอุปกรณ์ภายนอก (Port), ช่องทางการสื่อสาร (Bus), และวงจรกำเนิดสัญญาณนาฬิกา



รูปที่ 18 โมดูล ESP32-WROOM-32

ESP32 คือโมดูลไมโครคอนโทรลเลอร์ที่ผลิตโดยบริษัท Espressif มีความสามารถในการสื่อสารผ่านโปรโตคอลไร้สายอย่าง Wi-Fi, Bluetooth และ BLE (Bluetooth Low Energy) ซึ่ง ESP32 เป็นไมโครคอนโทรลเลอร์ที่มีการพัฒนาต่อยอดมาจาก ESP8266 โดยเพิ่มขีดจำกัดความสามารถในด้านต่าง ๆ อาทิ มีแกนประมวลผล (Core) ที่เพิ่มขึ้นจากแกนประมวลผลเดี่ยว (Single-core) เป็นแกนประมวลผลคู่ (Dual-core) ความละเอียดในการแปลงค่าแอนะล็อกเป็นดิจิทัลที่เพิ่มขึ้น หรือแม้กระทั่งจำนวนขา GPIO (General-purpose input/output) ที่เพิ่มขึ้น โดยตัว ESP32 จะมีคุณสมบัติหลักดังนี้

- หน่วยประมวลผล Tensilica Xtensa 32-bit LX6 microprocessor 2 แกน ประมวลผล สัญญาณนาฬิกา 240 Mhz ความสามารถในการประมวลผลระดับ 600 DMIPS
- รองรับการเชื่อมต่อ Wi-Fi 802.11 b/g/n (802.11n @ 2.4 GHz ความเร็วสูงสุด 150 Mbit/s)
- รองรับการเชื่อมต่อ Bluetooth v4.2 BR/EDR และ Bluetooth Low Energy (BLE)

- หน่วยความจำภายใน (SRAM) ขนาด 520 Kb
- รองรับหน่วยความจำภายนอกสูงสุดที่ 16 Mb
- รองรับการเชื่อมต่อ GPIO/ADC/DAC/SPI/I2C/I2S/UART/PWM
- มาตรฐานความปลอดภัย WEP, WPA/WPA2 PSK/Enterprise
- รองรับการเข้ารหัส AES/SHA2/ECC/RNG/RSA

ในงานวิจัยฉบับนี้เลือกใช้งาน ESP32 เนื่องจากเป็นโมดูลไมโครคอนโทรลเลอร์ที่มีความสามารถหลากหลาย ทำงานได้อย่างรวดเร็ว กินพลังงานต่ำ รองรับการสื่อสารหลายรูปแบบ ทำให้สามารถประยุกต์ใช้งานได้ง่าย เหมาะที่จะนำมาทำระบบฝังตัวในระบบไมโคร โดยหน้าที่หลักของ ESP32 ในงานวิจัยฉบับนี้จะเป็นตัวที่ทำหน้าที่ควบคุมการทำงานของมิเตอร์อัจฉริยะ ซึ่งรูปแบบการทำงานของ ESP32 สำหรับมิเตอร์อัจฉริยะต้นแบบสามารถแบ่งออกได้เป็นดังนี้

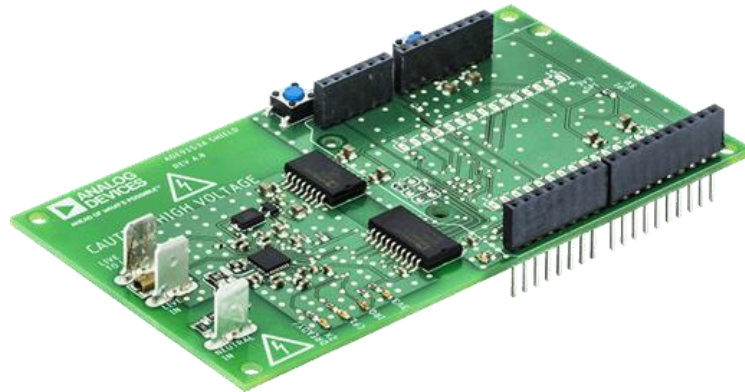
- หน่วยประมวลผลใน ESP32 นั้นเป็นแบบ 2 แกนประมวลผล (Dual-Core) ซึ่งมีความแตกต่างจากไมโครคอนโทรลเลอร์ทั่วไปที่มีแกนประมวลผลเดียว (Single-Core) เป็นจุดเด่นที่ทำให้ ESP32 นั้นสามารถทำงานได้หลายภารกิจในเวลาเดียวกัน (Multitasking) เนื่องจาก ESP32 นั้นมีระบบปฏิบัติการชนิด RTOS (Real-Time Operating System) ที่มีชื่อว่า FreeRTOS ซึ่งเป็นซอฟต์แวร์แบบโอเพนซอร์ส ทำให้สามารถแบ่งภารกิจ (Task) ออกได้เป็นหลาย ๆ ส่วน และสามารถจัดเรียงลำดับความสำคัญ (Priority) ของภารกิจได้ตามความเหมาะสม ซึ่งในมิเตอร์อัจฉริยะต้นแบบในงานวิจัยฉบับนี้มีการแบ่งภารกิจออกเป็นส่วน ๆ ตัวอย่างเช่น ภารกิจวัดค่าพลังงานทางไฟฟ้า ภารกิจการเก็บข้อมูลค่าพลังงานไฟฟ้าลงฐานข้อมูล และภารกิจการเชื่อมต่ออินเทอร์เน็ต เป็นต้น
- การวัดค่าพลังงานทางไฟฟ้านั้น ESP32 มีหน้าที่รับชุดข้อมูลที่วัดได้จากไอซีวัดค่าพลังงานทางไฟฟ้า ADE9153A โดยการส่งสัญญาณเรียกเพื่อขอรับข้อมูลผ่านการสื่อสารแบบ SPI (Serial Peripheral Interface) จากนั้นจะนำข้อมูลที่ได้อ่านมาคำนวณเพื่อหาค่าพลังงานทางไฟฟ้า ได้แก่ ค่าแรงดันไฟฟ้า ค่ากระแสไฟฟ้า ค่ากำลังไฟฟ้าจริง ค่ากำลังไฟฟ้ารีแอกทีฟ ค่ากำลังไฟฟ้าปรากฏ ค่าตัวประกอบกำลัง และค่าความถี่ทางไฟฟ้า
- นำข้อมูลที่วัดได้ไปเก็บบันทึกในฐานข้อมูลเพื่อเป็นการสำรองข้อมูลพลังงานไฟฟ้าที่วัดได้หรือนำไปใช้ประโยชน์ในการวิเคราะห์พลังงานไฟฟ้าใน
- การส่งข้อมูลขึ้นคลาวด์ผ่านรูปแบบการสื่อสารไร้สาย Wi-Fi เพื่อใช้ในการเฝ้าสังเกตการณ์ค่าพลังงานทางไฟฟ้า (Monitoring)

3.1.2 ไอซีวัดค่าพลังงานทางไฟฟ้า

จากงานวิจัยในอดีตที่ผ่านมา ชิปหรือไอซีที่นำมาใช้สำหรับการวัดค่าพลังงานทางไฟฟ้ามีให้เลือกใช้หลากหลาย มีคุณสมบัติที่แตกต่างกันออกไปขึ้นอยู่กับบริบทที่จะนำไปใช้งาน สำหรับงานวิจัยฉบับนี้ได้ทำการศึกษาค้นคว้าข้อมูลเกี่ยวกับไอซี ADE9153A [12] ซึ่งเป็นไอซีวัดค่าพลังงานทางไฟฟ้าแบบเฟสเดียวที่มีความแม่นยำสูง มาพร้อมกับฟังก์ชันสอบเทียบตัวเองโดยที่ไม่จำเป็นต้องใช้ไมโครคอนโทรลเลอร์หรือแหล่งจ่ายที่มีความแม่นยำใด ๆ ในการสอบเทียบ ไอซีนี้สามารถวัดค่าพลังงานทางไฟฟ้า ตัวอย่างเช่น ค่าแรงดันไฟฟ้า ค่ากระแสไฟฟ้า ค่ากำลังไฟฟ้าจริง ค่ากำลังไฟฟ้ารีแอกทีฟ ค่ากำลังไฟฟ้าปรากฏ ค่าตัวประกอบกำลังไฟฟ้าและ ค่าความถี่ทางไฟฟ้า เป็นต้น การวัดค่าพลังงานไฟฟ้านั้นจะรับข้อมูลที่ได้จากตัวรับรู้ (Sensor) จากภายนอก อาจเป็นวงจรแบ่งแรงดัน (Voltage divider) ตัวต้านทานชัณฑ์ (R-Shunt) หรือหม้อแปลงกระแสไฟฟ้า (CT) ต่อเข้ามายังตัวไอซี โดยมีวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล (Analog to Digital Converter; ADC) อยู่ภายใน นอกจากนี้ไอซีตัวนี้ยังรองรับการวัดค่าพลังงานที่สามารถนำไปใช้ประโยชน์ในด้านอื่น ๆ ได้อีกเช่น คุณภาพกำลังไฟฟ้า (Power Quality) การวัดมุมเฟส (Angle Measurement) หรือ แรงดันไฟฟ้าตก/เกินชั่วขณะ (Dip and Swell) เป็นต้น ซึ่งหากนำไอซีนี้มาประยุกต์ใช้ในการพัฒนามิเตอร์อัจฉริยะ จะทำให้ได้มิเตอร์อัจฉริยะรูปแบบใหม่ที่มีความสามารถพิเศษในการสอบเทียบตัวเอง อันเป็นประโยชน์ต่อผู้ใช้และผู้ให้บริการไฟฟ้าทั้งในทางตรงและทางอ้อม



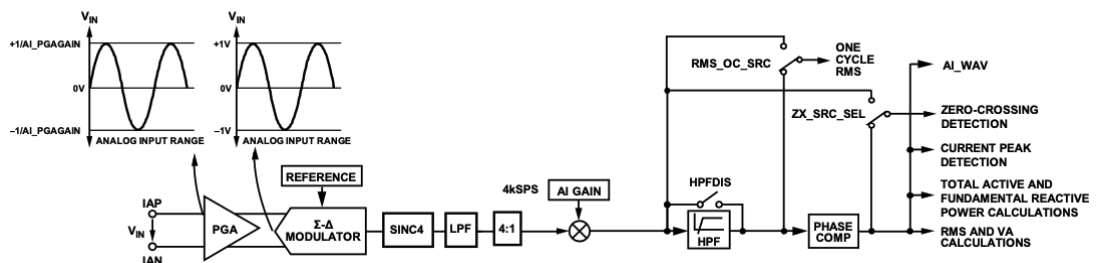
รูปที่ 19 ไอซีวัดค่าพลังงานทางไฟฟ้า ADE9153A ที่มีฟังก์ชันสอบเทียบตัวเอง



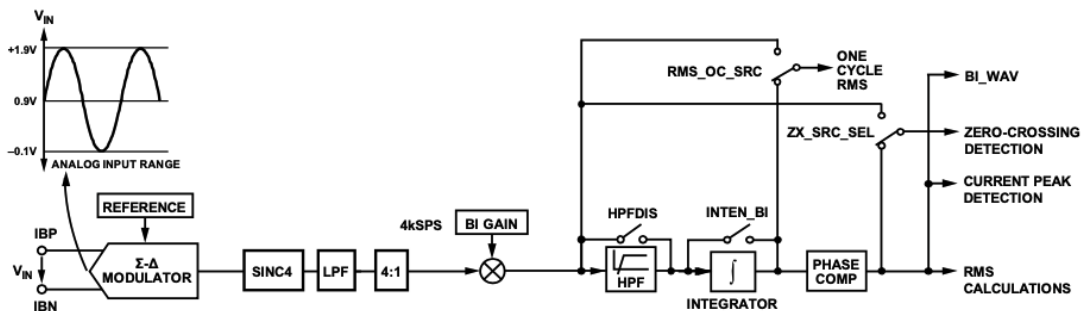
รูปที่ 20 ตัวอย่างบอร์ดประเมินของไอซีวัดค่าพลังงานทางไฟฟ้า ADE9153A
(EV-ADE9153ASHIELDZ)

3.1.2.1 หลักการวัดกระแสของไอซี

สำหรับการวัดกระแสของไอซี ADE9153A นั้นมีช่องสัญญาณกระแส (Current Channel) จำนวน 2 ช่องทาง ช่องสัญญาณแรก (Channel A) นั้นจะเหมาะสำหรับตัวรับรู้แบบตัวต้านทานชนิด ส่วนช่องสัญญาณที่สอง (Channel B) จะเหมาะสำหรับตัวรับรู้แบบหม้อแปลงกระแสไฟฟ้า โดยเส้นทางการไหลของข้อมูลทั้งสองช่องสัญญาณจะแสดงให้เห็นดังรูปที่ 21 และ 22 ตามลำดับ



รูปที่ 21 แสดงแผนภาพบล็อกเส้นทางข้อมูลกระแสของช่องสัญญาณที่หนึ่ง (Current Channel A Datapath)



รูปที่ 22 แสดงแผนภาพบล็อกเส้นทางข้อมูลกระแสของช่องสัญญาณที่สอง (Current Channel B Datapath)

ข้อมูลที่ได้จากการวัดของตัวรับรู้ต่าง ๆ นั้นจะถูกส่งต่อมายังขาเข้า (Input) ของไอซีเพื่อประมวลผลข้อมูลต่าง ๆ ต่อไป โดยสามารถเรียงลำดับได้ดังนี้

- $\Sigma - \Delta$ Modulator, SINC4, Low-pass Filter (LPF) ทำหน้าที่แปลงสัญญาณแอนะล็อกให้เป็นสัญญาณดิจิทัล (Analog to Digital Converter; ADC) เพื่อเตรียมข้อมูลให้พร้อมสำหรับการคำนวณ
- ขยายสัญญาณกระแส (Current Channel Gain) ทำหน้าที่ขยายสัญญาณ โดยแต่ละช่องสัญญาณ (AIGAIN และ BIGAIN) จะมีรีจิสเตอร์สำหรับการคำนวณอัตราขยายโดยใช้สมการ

$$\text{Current Channel Gain} = \left(1 + \frac{xIGAIN}{2^{27}}\right) \quad (3.1)$$

- กรองความถี่สูงผ่าน (High-pass Filter; HPF) ทำหน้าที่กำจัดค่าดีซีออฟเซต (DC Offset) เพื่อให้ได้ค่าอาร์เอ็มเอส (RMS) รวมถึงค่าพลังงานทางไฟฟ้าต่าง ๆ มีความแม่นยำมากขึ้น
- ดิจิทัลอินทิเกรเตอร์ (Digital Integrator) จะมีอยู่ในช่องสัญญาณที่สองเท่านั้น ทำให้ช่องสัญญาณที่สองนั้นสามารถรองรับตัวรับรู้แบบหม้อแปลงกระแสไฟฟ้า รวมถึงโรโกวสกีคอยล์ (Rogowski Coil) ได้
- ชดเชยเฟส (Phase Compensation) โดยจะสามารถชดเชยได้อยู่ในช่วง -15° ถึง $+2.25^\circ$ ที่ความถี่ 50 Hz และ -15° ถึง $+2.7^\circ$ ที่ความถี่ 60 Hz ทั้งสองช่องสัญญาณจะมีรีจิสเตอร์สำหรับการคำนวณค่าการชดเชยเฟสโดยใช้สมการ

$$xPHASECAL = \left(\frac{\sin(\varphi - \omega) + \sin \omega}{\sin(2\omega - \varphi)} \right) \times 2^{27} \quad (3.2)$$

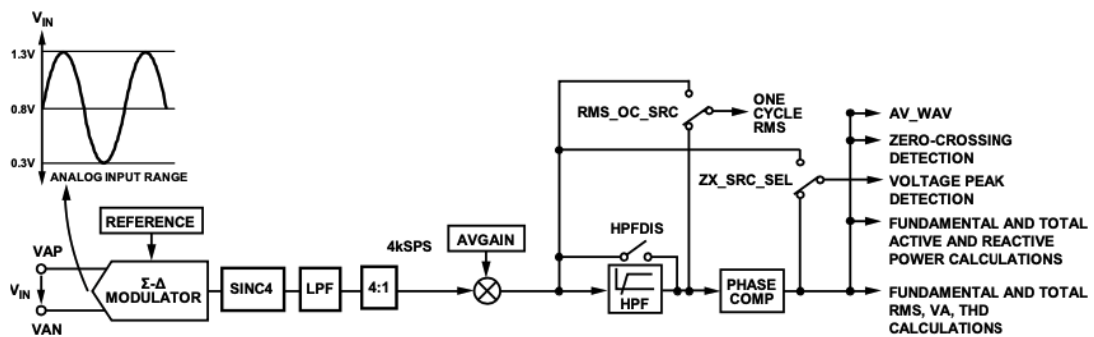
เมื่อ $\omega = 2\pi \times f_{LINE}/f_{DSP}$

โดยที่ φ คือ ค่าการแก้ไขเฟส (Phase correction)

f_{LINE} คือ ความถี่ของไฟฟ้ากระแสสลับ

f_{DSP} คือ ความถี่คิมที่มีค่าเท่ากับ 4 kHz

3.1.2.2 หลักการวัดแรงดันของไอซี



รูปที่ 23 แสดงแผนภาพบล็อกเส้นทางข้อมูลแรงดัน (Voltage Channel Datapath)

สำหรับการวัดแรงดันของไอซี ADE9153A นั้นมีช่องสัญญาณเพียงช่องทางเดียวดังแสดงให้เห็นในรูปที่ 23 ซึ่งจะมีเส้นทางไหลของข้อมูลเหมือนกันกับเส้นทางไหลของข้อมูลกระแสเพียงแต่ช่องสัญญาณแรงดัน (Voltage Channel) นั้นจะมีรีจิสเตอร์ขยายสัญญาณแรงดันโดยเฉพาะที่ชื่อว่า AVGAIN โดยสมการการคำนวณค่านั้นจะมีลักษณะเหมือนกันกับรีจิสเตอร์ของช่องสัญญาณกระแส ดังนี้

$$\text{Voltage Channel Gain} = \left(1 + \frac{AVGAIN}{2^{27}} \right) \quad (3.3)$$

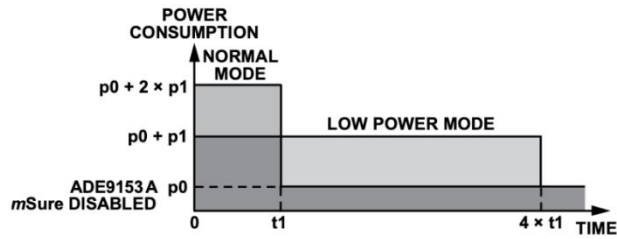
3.1.2.3 หลักการสอบเทียบตัวเองของไอซี

ไอซี ADE9153A [12] เป็นวงจรรวมสำหรับพัฒนามิเตอร์อิเล็กทรอนิกส์สมัยใหม่ที่มีความแม่นยำสูง และมีความสามารถในการสอบเทียบตัวเอง (Autocalibration) ซึ่งสามารถสอบเทียบตัวเองได้ทั้งช่องสัญญาณกระแสและแรงดัน ทำให้สะดวกต่อการสอบเทียบมาตรฐานสำหรับมิเตอร์อัจฉริยะมากขึ้นโดยไม่จำเป็นต้องใช้มิเตอร์อ้างอิงใด ๆ ในการสอบเทียบ

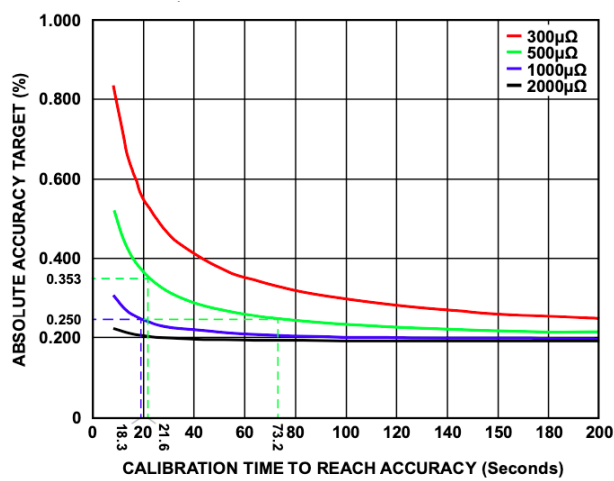
ในการสอบเทียบตัวเองของไอซีนั้นจะมีศัพท์เฉพาะด้าน (Terminology) ที่ควรทราบได้แก่

- ความแม่นยำสัมบูรณ์ (Absolute Accuracy) คือ ค่าเข้าสู่ความแม่นยำของการสอบเทียบตัวเองเมื่อเวลาผ่านไปขณะหนึ่ง ยิ่งดำเนินการสอบเทียบนานเท่าไร ยิ่งได้ความแม่นยำเท่านั้น
- ความเร็วในการลู่เข้า (Speed of Convergence) คือ เวลาที่ใช้ในการสอบเทียบตัวเองจนกว่าความแม่นยำจะถึงจุดสมดุล (Certain level of accuracy)
- ค่าคงตัวการแปลงผัน (Conversion Constant; CC) คือ ค่าที่ไอซีได้รับมาจากการประมาณค่าฟังก์ชันส่งผ่าน (Transfer function) ของตัวรับรู้และฟรอนท์เอนด์ของไอซี
- ความมั่นใจของการประมาณค่า (Certainty of Estimation; CERT) คือ ค่าความมั่นใจของการวัดค่าทางไฟฟ้า ซึ่งในอีกความหมายคือเป็นหน่วยวัดที่ใช้บอกถึงความแม่นยำของการวัดค่าพลังงานทางไฟฟ้า

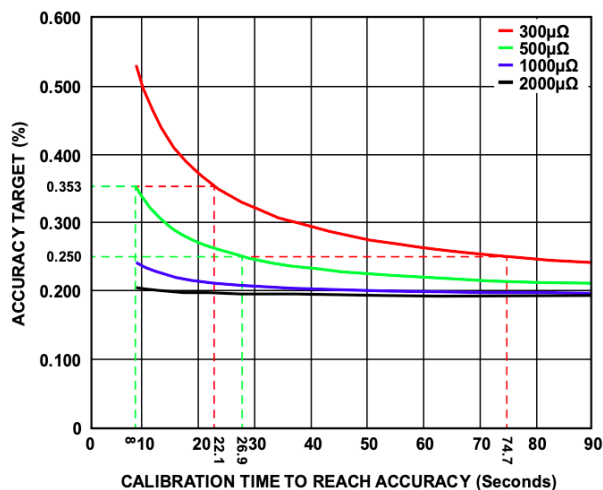
เมื่อมีการเรียกใช้งานฟังก์ชันสอบเทียบตัวเอง ช่องสัญญาณกระแสสามารถเลือกทำงานได้สองโหมดพลังงานคือโหมดปกติ และโหมดพลังงานต่ำ โดยแต่ละโหมดพลังงานนั้นจะมีผลได้เสียจากการแลกเปลี่ยนระหว่างความเร็วในการลู่เข้ากับอัตราการกินพลังงาน ในโหมดปกติจะมีความเร็วในการลู่เข้ามากกว่าโหมดพลังงานต่ำถึง 4 เท่า แต่อัตราการกินพลังงานนั้นเพิ่มขึ้นเพียง 2 เท่าเท่านั้น สามารถสรุปได้ว่าอัตราการกินพลังงานเฉลี่ยโดยรวมในโหมดปกติที่มีการกินพลังงานน้อยกว่าโหมดพลังงานต่ำ แต่หากเทียบอัตราการกินพลังงานในหนึ่งหน่วยเวลา (Instantaneous consumption) นั้น โหมดปกติจะกินพลังงานมากกว่าดังแสดงให้เห็นในรูปที่ 24



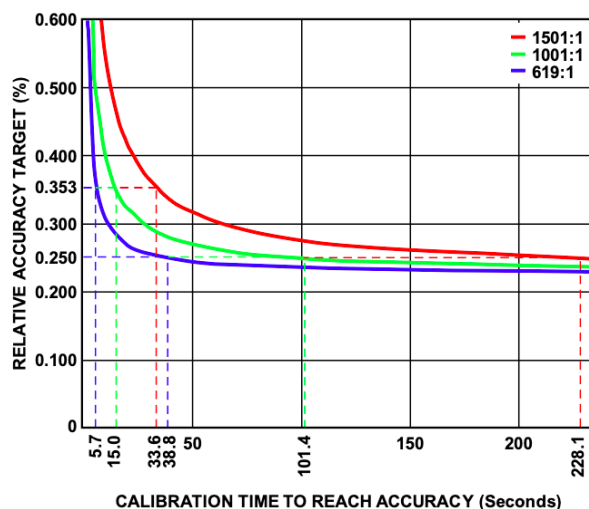
รูปที่ 24 โหมดพลังงานของการสอบเทียบตัวเอง (mSure Autocalibration Power Modes to Same Certainty)



รูปที่ 25 ความเร็วในการลู่เข้าของการสอบเทียบตัวเอง (ช่องสัญญาณกระแสในโหมดพลังงานต่ำ) ขึ้นอยู่กับค่าตัวต้านทานชนิด



รูปที่ 26 ความเร็วในการลู่เข้าของการสอบเทียบตัวเอง (ช่องสัญญาณกระแสในโหมดปกติ) ขึ้นอยู่กับค่าตัวต้านทานชนิด



รูปที่ 27 ความเร็วในการสู่เข้าของการสอบเทียบตัวเอง (ช่องสัญญาณแรงดัน)
ขึ้นอยู่กับอัตราส่วนของวงจรแบ่งแรงดัน

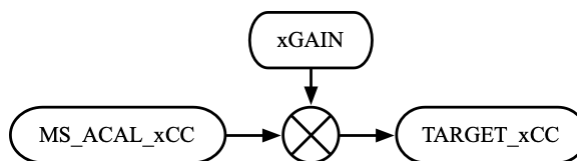
ในการทำงานของฟังก์ชันสอบเทียบตัวเองโดยที่ลดความจำเป็นในการใช้มิเตอร์อ้างอิงได้นั้น อันเนื่องมาจากไอซีมีความสามารถในการหาค่าคงตัวการแปลงผันของทั้งช่องสัญญาณกระแสและแรงดัน ซึ่งมีเพียงแค่ว่าค่าคงตัวการแปลงผันนี้เพียงอย่างเดียวก็สามารถดำเนินการสอบเทียบตัวเองได้

เมื่อไอซีได้รับพลังงานจากแหล่งจ่ายไฟ ฟังก์ชันการสอบเทียบตัวเองก็จะสามารถทำงานได้ โดยมีรีจิสเตอร์ MS_ACAL_CFG เป็นตัวกำหนดค่าอนุญาตการทำงานสอบเทียบของทั้งสองช่องสัญญาณ การสอบเทียบนั้นจะดำเนินการครั้งละหนึ่งช่องสัญญาณเท่านั้นโดยที่จะทำงานตามกำหนดระยะเวลาที่ได้ตั้งค่าไว้ หลังจากที่มีการดำเนินการสอบเทียบเสร็จสิ้น ค่าความมั่นใจของการประเมินค่าจะถูกยืนยันด้วยรีจิสเตอร์ MS_ACAL_xCERT จากนั้นรีจิสเตอร์ MS_ACAL_xCC จึงจะสามารถคำนวณค่าอัตราขยาย xGAIN เพื่อใช้ในการสอบเทียบมิเตอร์ได้

ขั้นตอนการสอบเทียบตัวเองจะสามารถอธิบายได้ตามลำดับขั้นตอนดังนี้

1. ทำการตรวจสอบความพร้อมจากบิต MS_SYSRDYP ซึ่งจะอยู่ในรีจิสเตอร์ MS_STATUS_CURRENT หากบิตนี้มีค่าเป็นบิตสูงนั้นหมายความว่าระบบพร้อมทำงาน หากมีค่าเป็นบิตต่ำหมายความว่าไม่สามารถทำงานได้
2. ทำการปรับแต่งรีจิสเตอร์ MS_ACAL_CFG ตามช่องสัญญาณที่ต้องการใช้งานรวมถึงเลือกโหมดพลังงานที่จะใช้ โดยมีขั้นตอนดังนี้

- ทำการเซ็ตบิตรีจิสเตอร์แต่ละบิตของ AUTOCAL_AI AUTOCAL_BI ไปจนถึง AUTOCAL_AV เพื่อเปิดการทำงานของช่องสัญญาณที่ต้องการ
 - จากนั้นให้เลือกโหมดพลังงานที่ต้องการใช้งานโดยเลือก 0 เมื่อต้องการใช้โหมดพลังงานต่ำ และ 1 เมื่อต้องการใช้โหมดปกติ โดยเซ็ตค่าที่รีจิสเตอร์ ACALMODE_AI และ ACALMODE_BI
 - ทำการเซ็ตบิต ACAL_RUN และ ACAL_MODE เพื่อยืนยันการตั้งค่าทั้งหมด
3. ค่าความมั่นใจของการประมาณค่านั้นจะมีหน่วยเป็น ppm ตามความไม่แน่นอนในการวัดค่าคงตัวการแปลงผัน โดยเมื่อมีการดำเนินการสอบเทียบบิต MS_READY ของรีจิสเตอร์ MS_STATUS_IRQ จะเป็นบิตที่บอกว่าค่าความมั่นใจของการประมาณค่าและค่าคงตัวการแปลงผันมีการปรับให้เป็นปัจจุบันแล้ว
 4. เมื่อเวลาหรือความมั่นใจถึงค่าที่เหมาะสม การดำเนินการสอบเทียบจะหยุดโดยทำการเซ็ตบิตไปยังรีจิสเตอร์ MS_ACAL_CFG
 5. ทำการอ่านค่าผลลัพธ์จากรีจิสเตอร์ MS_ACAL_xCC และ MS_ACAL_xCERT เพื่อรับค่าคงตัวการแปลงผันกับค่าความมั่นใจของการประมาณค่าที่ได้ของแต่ละช่องสัญญาณออกมา
 6. ทำขั้นตอนการสอบเทียบตัวเองตั้งแต่ข้อ 1 ถึง 5 อีกครั้งสำหรับช่องสัญญาณถัดไปจนครบทุกช่องสัญญาณตามที่ตั้งค่าไว้
 7. ก่อนจะทำการเริ่มดำเนินการสอบเทียบของช่องสัญญาณใด ๆ ตรวจสอบให้แน่ใจว่าบิต MS_SYSRDYP ของรีจิสเตอร์ MS_STATUS_CURRENT ถูกเซ็ตแล้ว หรือบิต MS_SYSRDY ของรีจิสเตอร์ MS_STATUS_IRQ ถูกเซ็ตเพื่อเป็นการยืนยันว่าพร้อมสำหรับการดำเนินการสอบเทียบของช่องสัญญาณถัดไป
 8. หลังจากทำการสอบเทียบตัวเองดำเนินการในช่องสัญญาณที่ต้องการเรียบร้อยแล้ว นำค่า MS_ACAL_AICC MS_ACAL_BICC และ MS_ACAL_AVCC ที่วัดได้มาใช้ในการหาค่าอัตราส่วนระหว่างค่าคงตัวการแปลงผันที่วัดได้ กับค่าคงตัวการแปลงผันเป้าหมาย จากนั้นนำค่าอัตราส่วนที่ได้ไปใส่ในรีจิสเตอร์ xGAIN ของแต่ละช่องสัญญาณ สามารถแสดงความสัมพันธ์ได้ดังรูปที่ 28



รูปที่ 28 ความสัมพันธ์ระหว่างค่าคงตัวการแปลงผันที่วัดได้กับค่าคงตัวการแปลงผันเป้าหมาย

โดยสามารถเขียนสมการความสัมพันธ์ได้ดังนี้

$$xGAIN_REGISTER_VALUE = \left(\frac{MS_ACAL_xCC}{TARGET_xCC} - 1 \right) \times 2^{27} \quad (3.4)$$

เมื่อ $TARGET_xICC = \frac{I_{MAX} \times xI_{HEADROOM}}{52,725,703}$

และ $TARGET_AVCC = \frac{V_{NOM} \times V_{HEADROOM}}{26,362,852}$

โดยที่ I_{MAX} คือ ค่ากระแสไฟฟ้าสูงสุด
 $xI_{HEADROOM}$ คือ ค่าเต็มสเกลที่ช่องสัญญาณกระแสรับได้ (Headroom)
 V_{NOM} คือ ค่าแรงดันไฟฟ้าปกติ
 $V_{HEADROOM}$ คือ ค่าเต็มสเกลที่ช่องสัญญาณแรงดันรับได้ (Headroom)

และสามารถหาค่า Headroom ของช่องสัญญาณต่าง ๆ ได้โดย

ในกรณีช่องสัญญาณกระแส A ซึ่งเหมาะสำหรับตัวรับรู้แบบตัวต้านทานชนิด จะหาได้จาก

$$AI_{HEADROOM} = \frac{\pm 1 V}{R_{SHUNT} \times I_{MAX} \times \sqrt{2}} \times \frac{AI_PGAGAIN}{16} \quad (3.5)$$

โดยที่ R_{SHUNT} คือ ค่าความต้านทานของตัวต้านทานชนิด
 $AI_PGAGAIN$ คือ ค่าของตัวขยายสัญญาณแบบโปรแกรมได้ (PGA) ของช่องสัญญาณกระแส A โดยค่าที่แนะนำสำหรับช่องสัญญาณนี้มีค่าเท่ากับ 16

ในกรณีของช่องสัญญาณกระแส B ซึ่งเหมาะสำหรับตัวรับรู้แบบหม้อแปลงกระแสไฟฟ้า

$$BI_{HEADROOM} = \frac{\pm 1 V}{R_{BURDEN} \times \frac{I_{MAX}}{CT_{RATIO}} \times \sqrt{2}} \times \frac{BI_PGAGAIN}{16} \quad (3.6)$$

โดยที่ R_{BURDEN} คือ ค่าความต้านทานของตัวต้านทานเบอร์เดน
 $BI_PGAGAIN$ คือ ค่าของตัวขยายสัญญาณแบบโปรแกรมได้ (PGA) ของช่องสัญญาณ
 กระแส B โดยค่าที่แนะนำสำหรับช่องสัญญาณนี้เท่ากับ 1
 CT_{RATIO} คือ ค่าอัตราส่วนของหม้อแปลงกระแส

และในกรณีของช่องสัญญาณแรงดัน

$$V_{HEADROOM} = \left(\frac{R_{BIG}}{R_{SMALL}} + 1 \right) \times \frac{\pm 0.5 V}{V_{NOM} \times \sqrt{2}} \quad (3.7)$$

โดยที่ R_{BIG} คือ ค่าความต้านทานของตัวต้านทานขนาดใหญ่ของวงจรแบ่งแรงดัน
 R_{SMALL} คือ ค่าความต้านทานของตัวต้านทานขนาดเล็กของวงจรแบ่งแรงดัน

3.1.3 บอร์ดราสเบอร์รี่พาย



รูปที่ 29 บอร์ดราสเบอร์รี่พาย 4
 จุฬาลงกรณ์มหาวิทยาลัย
 CHULALONGKORN UNIVERSITY

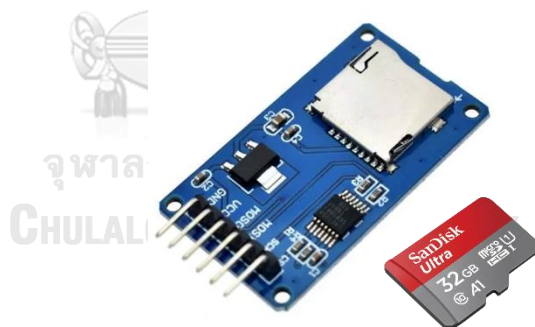
Raspberry Pi หรือบอร์ดพัฒนาราสเบอร์รี่พาย เป็นบอร์ดคอมพิวเตอร์ขนาดเล็กซึ่งอยู่ในกลุ่มคอมพิวเตอร์แบบฝังตัว (Embedded Computer) รองรับระบบปฏิบัติการลินุกซ์ได้หลายระบบ อาทิ Raspbian (Debian) Pidora (Fedora) หรือ Ubuntu เป็นต้น ในปัจจุบันเป็นที่นิยมมากขึ้นเนื่องจากราสเบอร์รี่พายสามารถนำไปใช้ประโยชน์อย่างกว้างขวางในอุตสาหกรรมต่าง ๆ อย่างในอุตสาหกรรมหุ่นยนต์ เครื่องเซิร์ฟเวอร์ หรืออุปกรณ์อินเทอร์เน็ตและสรรพสิ่ง เป็นต้น

ปัจจุบันบอร์ดราสเบอร์รี่พายมีจำหน่ายหลายรุ่นแล้ว โดยบอร์ดทุกรุ่นจะใช้ชิพประมวลผลกลาง (CPU) Broadcom ซึ่งมีหน่วยประมวลผลภาพ (GPU) รวมถึงแรมในตัวเดียวกัน บอร์ดราสเบอร์รี่พายแต่ละรุ่นก็จะมีคุณลักษณะแตกต่างกันไปเช่นชิพประมวลผลกลางมีความเร็วการทำงานตั้งแต่ 700 MHz ไปจนถึง 1.4 GHz โดยที่ในบอร์ดราสเบอร์รี่พายรุ่น 3 โมเดล B+ จะมีแรมตั้งแต่

256 MB จนถึง 1 GB นอกจากนี้มีช่องสำหรับการ์ดความจำเพื่อใช้ในการติดตั้งระบบปฏิบัติการ มีช่องสำหรับ USB ตั้งแต่ 1 ถึง 4 ช่อง มีช่องสัญญาณภาพ HDMI อีกทั้งยังมีช่องเอนกประสงค์ General Purpose Input/Output (GPIO) ที่จะใช้ในการควบคุมอุปกรณ์อิเล็กทรอนิกส์ และสำหรับในบอร์ดราสเบอร์รี่พายในรุ่นปัจจุบันยังมีความสามารถในการเชื่อมต่อสัญญาณไร้สายอย่าง Wi-Fi 802.11n และ Bluetooth ที่ติดตั้งมากับบอร์ดด้วย

สำหรับในงานวิจัยฉบับนี้จะประยุกต์ใช้ราสเบอร์รี่พายเข้ากับมิเตอร์อัจฉริยะเพื่อทำเป็นเกตเวย์ให้กับมิเตอร์ในการสื่อสารกับโพรโทคอลการตอบสนองด้านโหลด OpenADR ซึ่งจะทำตัวเป็นลูกข่ายของโพรโทคอลในการเชื่อมต่อสื่อสารเพื่อแลกเปลี่ยนข้อมูลเหตุการณ์การตอบสนองด้านโหลดกับเซิร์ฟเวอร์ที่เรียกว่าโหนดต้นเสมือน เมื่อเกิดเหตุการณ์ที่มีปริมาณการใช้ไฟฟ้าไม่สอดคล้องกับกำลังการผลิต โหนดต้นเสมือนจะดำเนินการสร้างและส่งเหตุการณ์การตอบสนองด้านโหลดไปยังลูกข่ายหรือโหนดปลายเสมือนเพื่อขอความร่วมมือในการลดการใช้พลังงานลง หากผู้ใช้ไฟฟ้าเข้าร่วมมาตรการการตอบสนองด้านโหลด ราสเบอร์รี่พายที่เป็นโหนดปลายเสมือนนี้จะทำการส่งการไปยังอุปกรณ์ไฟฟ้าที่เกี่ยวข้องเพื่อปรับลดหรือปิดเครื่อง อันเป็นการช่วยลดการใช้พลังงานตามมาตรการการตอบสนองด้านโหลดที่จะช่วยให้โครงข่ายไฟฟ้าสามารถรักษาเสถียรภาพที่ตีไว้ได้

3.1.4 หน่วยความจำภายนอก



รูปที่ 30 ตัวอย่างโมดูลอ่าน/เขียน Micro SD Card

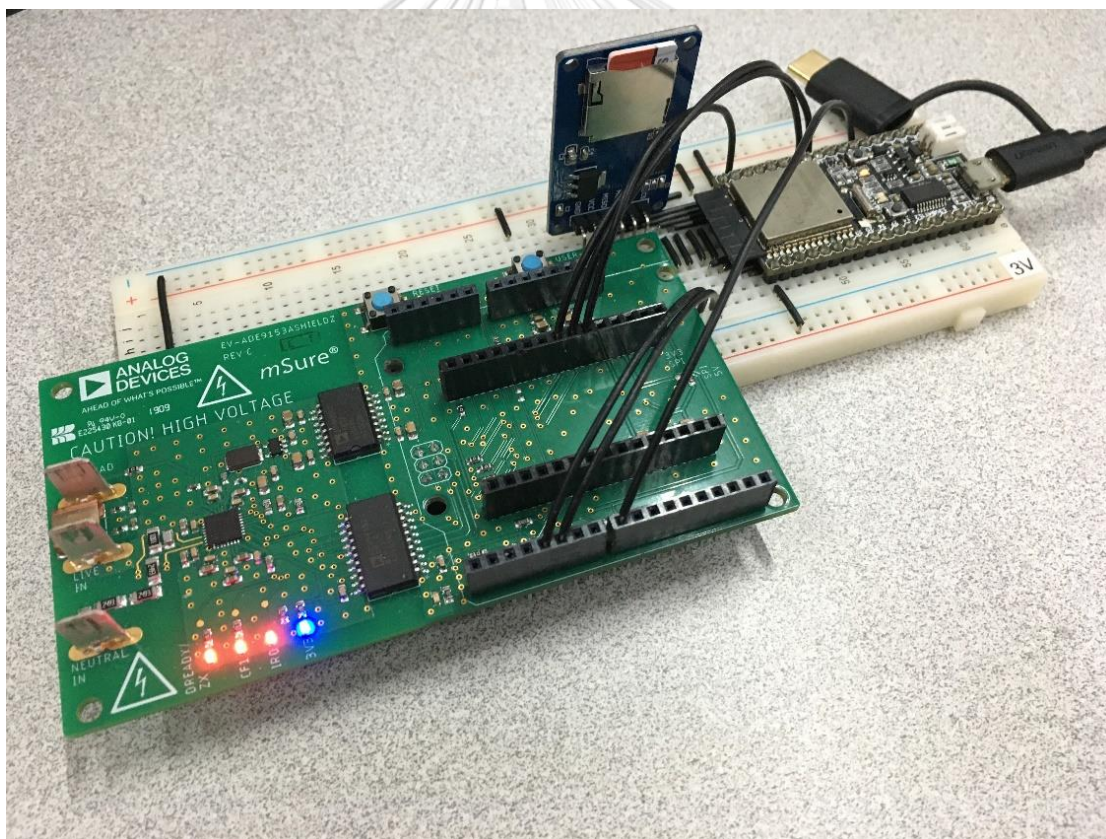
เป็นอุปกรณ์ที่ใช้เชื่อมต่อกับไมโครคอนโทรลเลอร์เพื่อบันทึกหรืออ่านข้อมูลของการ์ดความจำ โดยมีการเชื่อมต่อสื่อสารแบบอนุกรม (Serial Peripheral Interface; SPI) สามารถรองรับการ์ดความจำทั้งแบบไมโครเอสดี ไมโครเอสดีเอชซี (การ์ดความเร็วสูง) รองรับไฟเลี้ยงที่แรงดันช่วงพิกัด 3.3 - 5 Vdc โดยในงานวิจัยฉบับนี้นำมาประยุกต์ใช้เพื่อเก็บฐานข้อมูลของการวัดค่าพลังงานทางไฟฟ้าในแต่ละช่วงเวลา ซึ่งจะมีการเขียนข้อมูลลงไปในการ์ดความจำทุก ๆ 1 - 15 นาที ข้อมูลที่ถูกเก็บจะสามารถนำไปใช้ประโยชน์ในการวิเคราะห์พลังงานทางไฟฟ้าได้ต่อไป

3.2 ฮาร์ดแวร์มิเตอร์อัจฉริยะต้นแบบที่พัฒนาขึ้น

มิเตอร์อัจฉริยะต้นแบบที่พัฒนาขึ้นจะใช้บอร์ดพัฒนา ESP32 (ESP32 Development Board) ประกอบเข้ากับโมดูลหน่วยความจำภายนอกและบอร์ดประเมิน EV-ADE9153ASHIELDZ ซึ่งเป็นบอร์ดประเมินของไอซีวัดค่าพลังงาน ADE9153A

โดยบอร์ดประเมิน EV-ADE9153ASHIELDZ จะมีคุณลักษณะดังนี้

- ตัวรับรู้กระแส แบบตัวต้านทานชนิดขนาด $1\text{ m}\Omega$ ติดตั้งมาให้บนบอร์ด สามารถรับกระแสปกติที่ผ่านตัวต้านทานชนิดนี้ได้ที่พิกัด 5 A_{rms} และสูงสุดที่ 10 A_{rms}
- ตัวรับรู้แรงดัน แบบตัวต้านทานแบ่งแรงดัน (Resistor dividers) ติดตั้งมาให้บนบอร์ด สามารถรับแรงดันสูงสุดได้ที่พิกัด 240 V_{rms} โดยมีอัตราส่วนในการลดทอนแรงดันอยู่ที่ $100 : 1$



รูปที่ 31 ฮาร์ดแวร์ต้นแบบของมิเตอร์อัจฉริยะเฟสเดียวที่มีฟังก์ชันสอบเทียบตัวเอง

บทที่ 4

การออกแบบและพัฒนาซอฟต์แวร์ของมิเตอร์อัจฉริยะ

งานวิจัยฉบับนี้มีอุปกรณ์หลาย ๆ ส่วนประกอบเข้าด้วยกัน ซอฟต์แวร์ที่ใช้จึงมีความแตกต่างกันออกไปตามลักษณะของแต่ละส่วนประกอบ โดยงานวิจัยนี้สามารถแบ่งซอฟต์แวร์ออกเป็น 3 ส่วนด้วยกัน ได้แก่ ซอฟต์แวร์ที่ใช้ในการพัฒนามิเตอร์อัจฉริยะ ซอฟต์แวร์ไอโอทีคลาวด์แพลตฟอร์ม (NETPIE) และซอฟต์แวร์สำหรับการสื่อสารโพรโทคอล OpenADR

4.1 ซอฟต์แวร์ที่พัฒนาสำหรับมิเตอร์อัจฉริยะ

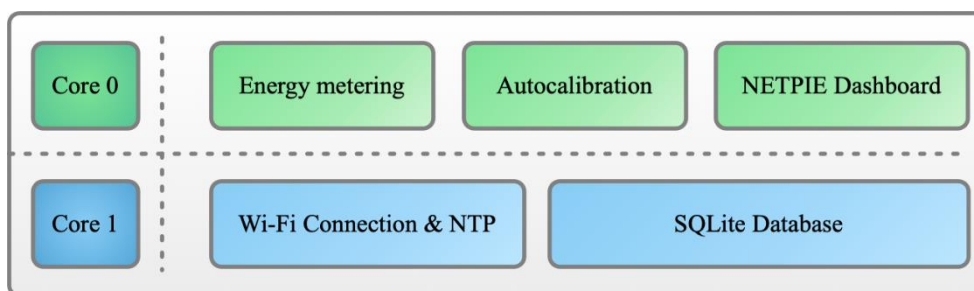
ESP32 นั้นเป็นไมโครคอนโทรลเลอร์ที่รองรับภาษาที่ใช้ในการพัฒนาได้หลากหลาย อาทิ ภาษาซี (C/C++) หรือภาษาไมโครไพธอน (MicroPython) เป็นต้น ในกระบวนการพัฒนามิเตอร์อัจฉริยะต้นแบบนี้ ผู้วิจัยเลือกใช้ภาษาซีซึ่งถือเป็นภาษาสามัญของไมโครคอนโทรลเลอร์ประเภทนี้ในการพัฒนาซอฟต์แวร์ของมิเตอร์

ซอฟต์แวร์ที่ถูกพัฒนาขึ้นสำหรับมิเตอร์อัจฉริยะจะใช้ระบบปฏิบัติการ RTOS (Real-Time Operating System) ที่มีชื่อว่า FreeRTOS ซึ่ง ESP32 มีความสามารถนำระบบปฏิบัติการนี้มาประยุกต์ใช้งานได้ โดย FreeRTOS จะมีความสามารถการทำงานหลายภารกิจ (Multitasking) ในเวลาเดียวกัน ซึ่งมิเตอร์อัจฉริยะต้นแบบนี้จะแบ่งภารกิจ (Tasks) ออกเป็น 5 ส่วนด้วยกัน ได้แก่

- ภารกิจการวัดค่าพลังงานทางไฟฟ้า (Energy Metering)
- ภารกิจการสอบเทียบตัวเอง (Autocalibration)
- ภารกิจส่งข้อมูลขึ้นคลาวด์ (NETPIE Dashboard)
- ภารกิจเชื่อมต่ออินเทอร์เน็ตแบบไร้สาย (Wi-Fi)
- ภารกิจจัดเก็บข้อมูลลงฐานข้อมูล (SQLite Database)

โดยทั่วไปภารกิจต่าง ๆ ของมิเตอร์นี้สามารถทำงานภายในแกนประมวลผล (Core) เดียวกันได้ เพียงแต่การทำงานภารกิจต่าง ๆ เหล่านี้จะต้องเรียงลำดับการทำงานให้เป็นขั้นเป็นตอนเนื่องจากไมโครคอนโทรลเลอร์ในโหมดปกติจะมีการทำงานในรูปแบบลูปเดียว (Cyclic executive) ทำให้ผู้ใช้งานจะต้องมีการบริหารจัดการการทำงานภารกิจต่าง ๆ ภายในลูปเดียวกัน ซึ่งอาจซับซ้อนและยากต่อการบริหารจัดการหากภารกิจการทำงานมีจำนวนที่มากขึ้น และยังส่งผลต่อประสิทธิภาพการทำงานในเรื่องของการหน่วงเวลา ทำให้การทำงานอาจล่าช้า (Delay) กว่าปกติที่คาดการณ์ไว้ ดังนั้นเพื่อเป็นการดึงความสามารถของ ESP32 อันเนื่องมาจากที่มีแกนประมวลผล 2 แกน ผู้วิจัยจึงได้จัด

วางภารกิจต่าง ๆ ให้กระจายอยู่ในแกนประมวลผลทั้ง 2 แกนอย่างเหมาะสมเพื่อเป็นการกระจายภาระการทำงานไม่ให้หนักไปที่เพียงแกนเดียว โดยระบบปฏิบัติการ RTOS นี้ยังสามารถเรียงลำดับความสำคัญ (Priority) เพื่อให้ภารกิจสามารถทำงานร่วมกันได้อย่างเป็นระบบ โดยการออกแบบภารกิจการทำงานของไมโครอัจฉริยะต้นแบบสามารถแสดงได้ดังรูปที่ 34



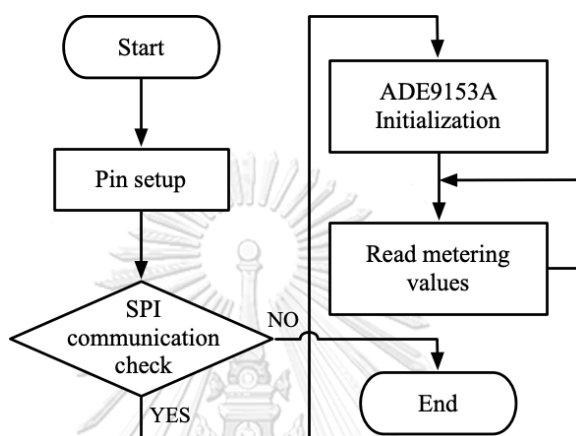
รูปที่ 34 ลักษณะภารกิจการทำงานของไมโครอัจฉริยะที่สอบเทียบตัวเอง

ภารกิจจะถูกแบ่งการทำงานออกเป็น 2 แกน โดยที่แกนประมวลผล 0 จะประกอบไปด้วยภารกิจการวัดค่าพลังงานทางไฟฟ้า ภารกิจการสอบเทียบตัวเอง และภารกิจส่งข้อมูลขึ้นคลาวด์ อยู่ในแกนเดียวกัน เนื่องจากผู้วิจัยต้องการที่จะบริหารจัดการภารกิจที่เกี่ยวข้องกับการวัดค่าพลังงานทางไฟฟ้าของไมโครอัจฉริยะให้อยู่ภายในแกนเดียวกัน โดยที่ผู้วิจัยจะจัดลำดับความสำคัญให้กับภารกิจการสอบเทียบตัวเองมีค่าเท่ากับ 2 ซึ่งเป็นความสำคัญสูงสุดในแกนประมวลผล 0 ตามด้วยภารกิจการวัดค่าพลังงานทางไฟฟ้า และภารกิจส่งข้อมูลขึ้นคลาวด์จะมีค่าความสำคัญเท่ากับ 1 เท่ากัน ซึ่งเป็นลำดับความสำคัญที่ต่ำกว่าเนื่องจากเป็นภารกิจที่ต้องทำตลอดเวลาทั้งสองภารกิจ ซึ่งหากตั้งค่าลำดับความสำคัญมีค่าสูง อาจทำให้ภารกิจการสอบเทียบตัวเองถูกรบกวนและไม่สามารถทำงานได้

สำหรับแกนประมวลผล 1 จะประกอบไปด้วยภารกิจเชื่อมต่ออินเทอร์เน็ตแบบไร้สาย และภารกิจจัดเก็บข้อมูลลงฐานข้อมูล อยู่ในแกนเดียวกัน ซึ่งผู้วิจัยจัดให้เป็นแกนประมวลผลสำหรับภารกิจการทำงานในส่วนเสริมของไมโครอัจฉริยะ โดยจะจัดลำดับความสำคัญให้กับภารกิจเชื่อมต่ออินเทอร์เน็ตแบบไร้สาย มีค่าลำดับความสำคัญเท่ากับ 2 ซึ่งเป็นความสำคัญสูงสุดในแกนประมวลผล 1 ทำให้เมื่อมีการเรียกใช้ภารกิจนี้ในการตรวจสอบสถานะของการเชื่อมต่ออินเทอร์เน็ต สามารถแทรกการทำงานภารกิจจัดเก็บข้อมูลลงฐานข้อมูลซึ่งมีลำดับความสำคัญเท่ากับ 1 ซึ่งต่ำกว่าได้ สามารถอธิบายการทำงานของภารกิจได้ดังนี้

4.1.1 การจัดการวัดค่าพลังงานทางไฟฟ้า (Energy Metering)

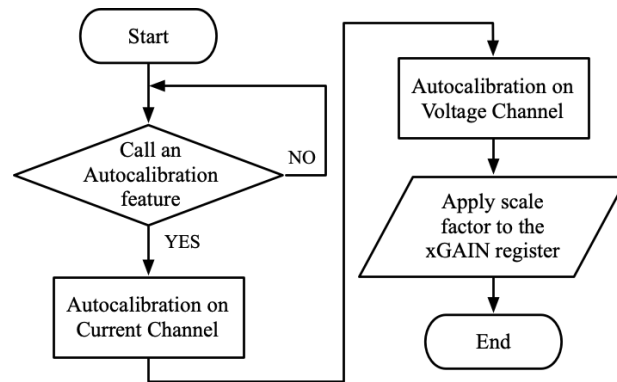
ภารกิจวัดค่าพลังงานทางไฟฟ้าจะเป็นภารกิจที่มีเตอร์อัจฉริยะใช้ในการจัดเตรียมความพร้อมสำหรับไอซีวัดค่าพลังงาน โดยจะทำการตรวจสอบไอซีในวงจรว่ามีการเชื่อมต่อการสื่อสาร SPI และพร้อมสำหรับการทำงานหรือไม่ หากการเชื่อมต่อไม่มีข้อผิดพลาดก็จะสามารถเข้าสู่กระบวนการวัดค่าพลังงานทางไฟฟ้าได้ โดยสามารถแสดงรูปแบบการทำงานได้ดังรูปที่ 35



รูปที่ 35 แผนภูมิสายงานภารกิจวัดค่าพลังงานทางไฟฟ้า

4.1.2 การจัดการสอบเทียบตัวเอง (Autocalibration)

ภารกิจนี้จะเป็นภารกิจที่ทำให้มิเตอร์อัจฉริยะมีความเที่ยงตรงและแม่นยำอยู่เสมอ เมื่อมีการเรียกใช้ภารกิจนี้ ตัวภารกิจจะทำการส่งสัญญาณขัดจังหวะ (Interrupt) ไปยังภารกิจอื่น ๆ ในแกนประมวลผลเดียวกันเพื่อขอให้หยุดการทำงานชั่วคราว และจะเข้าสู่โหมดการสอบเทียบตัวเองโดยจะสามารถทำการสอบเทียบตัวเองได้ที่ละหนึ่งช่องสัญญาณเท่านั้น เริ่มจากการสอบเทียบตัวเองของช่องสัญญาณกระแส จากนั้นทำการสอบเทียบช่องสัญญาณแรงดันตามลำดับ หลังจากทำการสอบเทียบตัวเองแล้วเสร็จ ไมโครคอนโทรลเลอร์จะทำการคำนวณค่าอัตราขยายจากอัตราส่วนค่าคงตัวการแปลงผันที่ได้จากการสอบเทียบตัวเอง เมื่อเสร็จสิ้นภารกิจจะส่งสัญญาณไปบอกภารกิจอื่น ๆ ให้สามารถทำงานต่อได้อย่างปกติ โดยจะสามารถแสดงรูปแบบการทำงานของภารกิจนี้ได้ดังรูปที่ 36

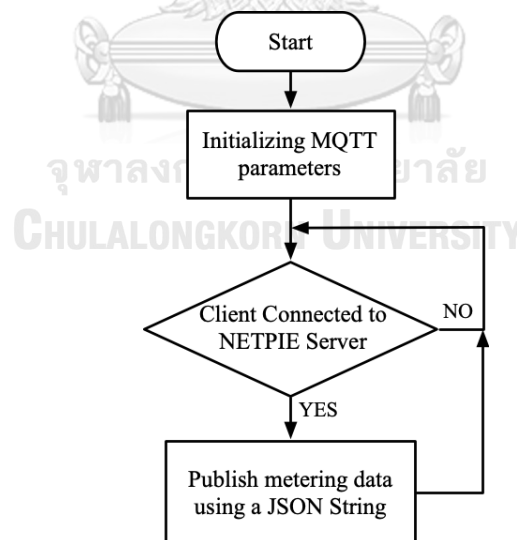


รูปที่ 36 แผนภูมิสายงานภารกิจสอบเทียบตัวเอง

4.1.3 ภารกิจส่งข้อมูลขึ้นคลาวด์ (NETPIE Dashboard)

ภารกิจนี้เป็นภารกิจสำหรับส่งข้อมูลไปยังคลาวด์แพลตฟอร์ม โดยใช้บริการ NETPIE ซึ่งเป็นคลาวด์แพลตฟอร์มที่เปิดให้บุคคลทั่วไปใช้งานโดยมีเว็บพอร์ทัลที่สามารถลงทะเบียนและจัดการตัวตนและสิทธิของแอปพลิเคชันและอุปกรณ์ได้ที่เว็บไซต์ โดยหน้าที่หลักของภารกิจนี้คือส่งข้อมูลค่าพลังงานทางไฟฟ้าต่าง ๆ ขึ้นไปเพื่อใช้ในการแสดงผลด้วย NETPIE Dashboard ทำให้สามารถเฝ้าสังเกตผ่านเว็บไซต์ได้แบบเวลาจริง (Real-time) โดยจะสามารถแสดงรูปแบบการทำงานได้ดังรูปที่

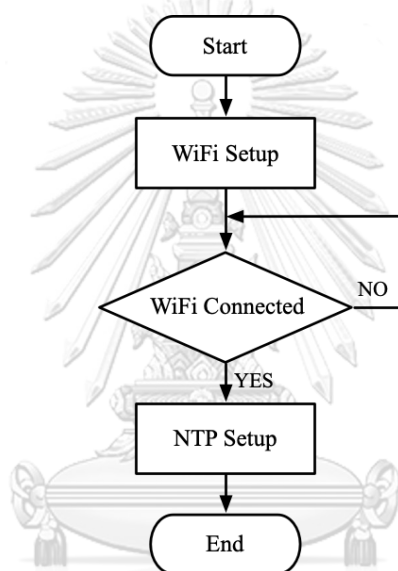
37



รูปที่ 37 แผนภูมิสายงานภารกิจส่งข้อมูลขึ้นคลาวด์

4.1.4 ภารกิจเชื่อมต่ออินเทอร์เน็ตแบบไร้สาย (Wi-Fi)

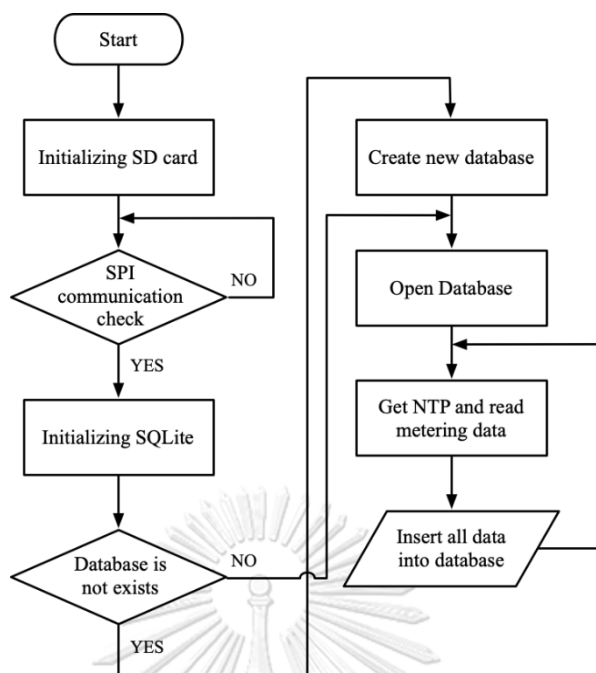
ภารกิจนี้จะเป็นตัวที่ทำหน้าที่ในการจัดเตรียมการเชื่อมต่อสัญญาณอินเทอร์เน็ตแบบไร้สาย โดยจะเริ่มเชื่อมต่อกับสัญญาณอินเทอร์เน็ตเป็นภารกิจแรกเมื่อเปิดเครื่อง เพื่อให้ ESP32 พร้อมสำหรับการส่งข้อมูลต่าง ๆ ไปยังคลาวด์ได้ อีกทั้งยังสามารถดึงเวลาจาก NTP Server (Network Time Protocol Server) ซึ่งเป็นเซิร์ฟเวอร์อ้างอิงเวลามาตรฐานสากลที่ใช้เพื่อตั้งค่าเวลาอุปกรณ์อิเล็กทรอนิกส์ให้ตรงกัน โดยหลังจากที่เข้าสู่การทำงานปกติ ภารกิจนี้จะถูกเรียกทุก ๆ 5 นาทีเพื่อทำการตรวจสอบสถานะการเชื่อมต่อว่าปกติหรือไม่ หากขาดการเชื่อมต่อจะทำการเชื่อมต่อใหม่อีกครั้ง เพื่อให้การทำงานเป็นไปได้อย่างปกติ สามารถแสดงรูปแบบการทำงานของภารกิจนี้ได้ดังรูปที่ 38



รูปที่ 38 แผนภูมิสายงานภารกิจเชื่อมต่ออินเทอร์เน็ตแบบไร้สาย

4.1.5 ภารกิจจัดเก็บข้อมูลลงฐานข้อมูล (SQLite Database)

ภารกิจการจัดเก็บฐานข้อมูลนั้นเป็นส่วนที่ทำให้ ESP32 สามารถสร้างฐานข้อมูลได้ด้วยตัวเอง โดยการเรียกใช้งานคลังโปรแกรม (Library) ฐานข้อมูลใน ESP32 ที่ชื่อ “sqlite3.h” จากนั้นจะสามารถบริหารจัดการฐานข้อมูลได้ด้วยภาษาโปรแกรมอย่าง SQL โดยสามารถแสดงลักษณะการจัดเก็บข้อมูลได้ดังรูปที่ 39



รูปที่ 39 แผนภูมิสายงานภารกิจจัดเก็บข้อมูลลงฐานข้อมูล

4.2 ซอฟต์แวร์ไอโอทีคลาวด์แพลตฟอร์ม (NETPIE)

การใช้งานบริการ NETPIE ในงานวิจัยฉบับนี้ เบื้องต้นจะใช้เป็นลักษณะของการเฝ้าสังเกตข้อมูลต่าง ๆ ของมิเตอร์อัจฉริยะผ่านเว็บพอร์ทัลของ NETPIE โดยข้อมูลการวัดค่าพลังงานทางไฟฟ้าจะถูกส่งขึ้นไปยังคลาวด์หลังจากที่ ESP32 นั้นทำการเชื่อมต่ออินเทอร์เน็ตเรียบร้อยแล้ว โดยลักษณะของการส่งข้อมูลนั้น ESP32 จะทำการสร้างชุดข้อมูลที่มีลักษณะแบบ JSON ไฟล์ (JavaScript Object Notation) จากนั้นจะเป็นหน้าที่ของบริการ NETPIE ที่จะคัดแยกข้อมูลออกเป็นส่วน ๆ เพื่อใช้ในการแสดงผล ควบคุมการทำงานของอุปกรณ์ต่าง ๆ รวมถึงการจัดเก็บลงฐานข้อมูล (Timeseries Database)

บริการ NETPIE จะมีโครงสร้างการรับข้อมูลอยู่ 4 รูปแบบคือ Device shadow, Device schema, Device trigger และ Event Hook โดยจะอธิบายการทำงานได้ดังนี้

- Device Shadow คือ ฐานข้อมูลเสมือนของอุปกรณ์ เป็นฐานข้อมูลเล็ก ๆ ที่มีคู่อยู่กับอุปกรณ์ (Device) ทุกตัว ใช้สำหรับเก็บข้อมูลต่าง ๆ เกี่ยวกับอุปกรณ์นั้น ๆ (Device Shadow Data) เช่น ข้อมูลที่เกิดจากเซนเซอร์ ข้อมูลการกำหนดองค์ประกอบต่าง ๆ (Device Configuration) เป็นต้น โดยข้อมูลที่ถูกส่งมายัง Device Shadow นั้นจะถูกเขียนทับด้วยค่าที่ถูกส่งมาล่าสุดเท่านั้น

- Device Schema คือ แผงผังข้อมูลที่กำหนดไว้เพื่อใช้กำกับ Device Shadow สำหรับอุปกรณ์ที่ต้องมีการจัดการข้อมูลเป็นลักษณะ Timeseries Database เสมือนว่าเป็นแผ่นแบบ (Template) ของข้อมูล ทำให้เซิร์ฟเวอร์สามารถทำการตรวจสอบข้อมูลแปลงข้อมูล รวมถึงการจัดเก็บข้อมูลลงในฐานข้อมูลได้อย่างถูกต้อง
- Device Trigger คือ ฟังก์ชันที่ใช้สำหรับสร้างเงื่อนไขเมื่อข้อมูลมีการเปลี่ยนแปลงไป โดยปกติจะใช้งานคู่กับ Event Hook สามารถกำหนดได้ว่าเมื่อตรงตามเงื่อนไขแล้วจะให้มีการกระทำในลักษณะใด
- Event Hook คือ ชุดการทำงานที่จะถูกเรียกใช้เมื่อเกิด Device Trigger

```

{
  "additionalProperties": false,
  "properties": {
    "temp": {
      "operation": {
        "store": {
          "ttl": "30d"
        },
        "transform": {
          "expression": "(${temp} * 1.8) + 32"
        }
      },
      "type": "number"
    },
    "place": {
      "operation": {
        "store": {
          "ttl": "7m"
        }
      },
      "type": "string"
    }
  }
}

```

รูปที่ 40 ตัวอย่าง Device Schema ในรูปแบบ JSON

4.3 ซอฟต์แวร์การตอบสนองด้านโหลด OpenADR

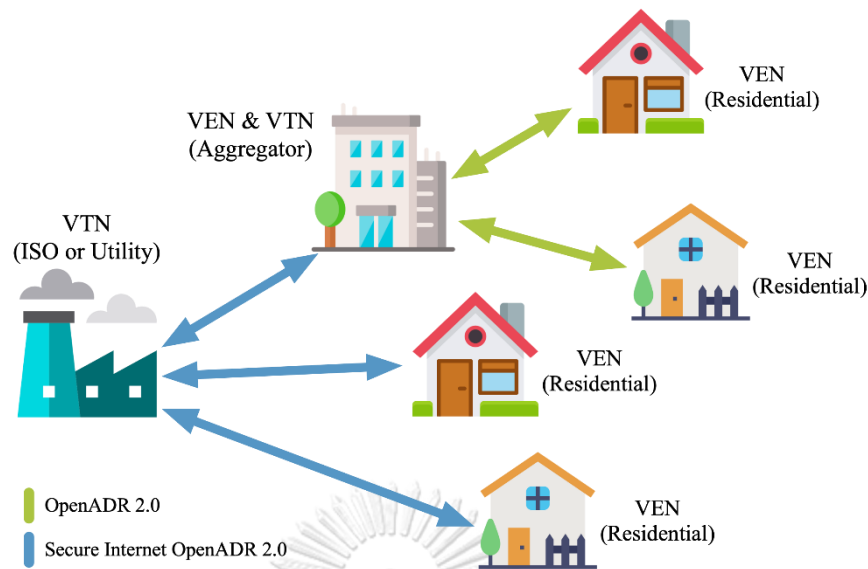
4.3.1 โครงสร้างสถาปัตยกรรมพื้นฐานของซอฟต์แวร์ OpenADR

ซอฟต์แวร์ OpenADR นั้นถูกสร้างขึ้นมาเพื่อใช้เป็นโพรโทคอลในการสื่อสารแลกเปลี่ยนข้อมูลเกี่ยวกับการตอบสนองด้านโหลด โดยโครงสร้างการสื่อสารของโพรโทคอล OpenADR นี้มีส่วนประกอบสองส่วนได้แก่ โหนดต้นเสมือน (Virtual Top Nodes; VTNs) และโหนดปลายเสมือน (Virtual End Nodes; VENs) ซึ่งในการสื่อสาร OpenADR 2.0 นั้น โหนดต้นเสมือนจะสื่อสารกับโหนดปลายเสมือนหนึ่งตัวหรือมากกว่าเท่านั้น จะไม่มีการสื่อสารระหว่างโหนดต้นเสมือนกับโหนดต้นเสมือน หรือโหนดปลายเสมือนกับโหนดปลายเสมือนด้วยกัน โดยสามารถอธิบายเอนทิตี (Entity) ทั้งสองพอสังเขปได้ดังนี้

- โหนดต้นเสมือน (VTN) คือ เอนทิตีที่มีหน้าที่รับผิดชอบระบบสื่อสารของโพรโทคอล รวมถึงการสร้างเหตุการณ์การตอบสนองด้านโหลด (Event) การปรับเปลี่ยนแก้ไขเหตุการณ์ (Modify) หรือสั่งการไปยังโหนดปลายเสมือนในการควบคุมอุปกรณ์ปลายทาง
- โหนดปลายเสมือน (VEN) คือ เอนทิตีที่มีหน้าที่สื่อสารกับโหนดต้นเสมือน รวมถึงควบคุมการทำงานหรือควบคุมความต้องการในการใช้พลังงานของอุปกรณ์ปลายทาง เพื่อให้สอดคล้องต่อสถานะของระบบไฟฟ้าให้เป็นไปตามมาตรการการตอบสนองด้านโหลด

โหนดต้นเสมือนจะเปรียบเสมือนกับระบบเซิร์ฟเวอร์ที่บริหารจัดการการตอบสนองด้านโหลด (Demand Response Management System; DRMS) ส่วนโหนดปลายเสมือนจะเปรียบได้กับลูกข่าย มักจะเป็นส่วนประกอบของระบบบริหารจัดการพลังงาน (Energy Management System; EMS) ทั้งในบ้านเรือน อาคาร หรือโรงงานอุตสาหกรรมในการควบคุมอุปกรณ์เครื่องใช้ไฟฟ้าปลายทาง

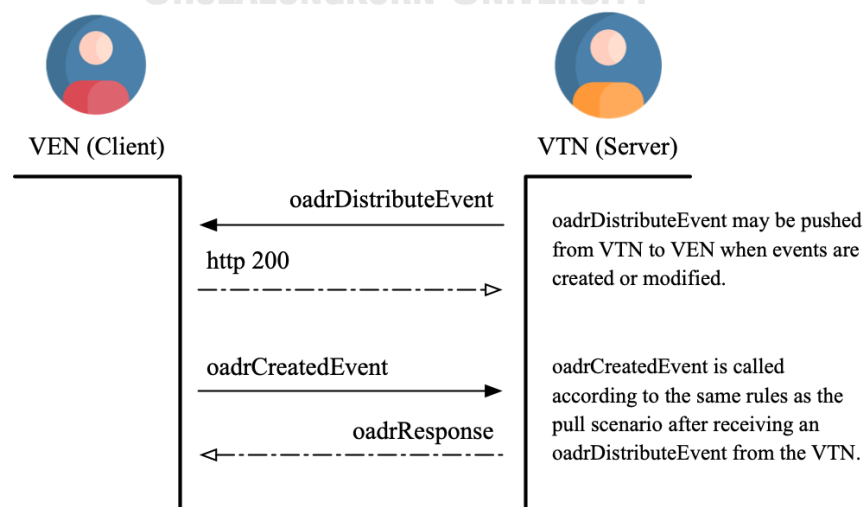
นอกจากนี้ยังมีตัวกลางอย่างผู้รวบรวม (Aggregator) ที่จะสามารถเป็นได้ทั้งโหนดต้นเสมือนและโหนดปลายเสมือนในเวลาเดียวกัน โดยจะทำหน้าที่เป็นโหนดปลายเสมือนก็ต่อเมื่อต้องมีการติดต่อกับผู้ให้บริการไฟฟ้า และจะทำหน้าที่เป็นโหนดต้นเสมือนเมื่อต้องส่งสัญญาณการตอบสนองด้านโหลดไปยังผู้ใช้ไฟฟ้า ดังแสดงให้เห็นถึงสถาปัตยกรรมการสื่อสารโพรโทคอล OpenADR ในรูปที่



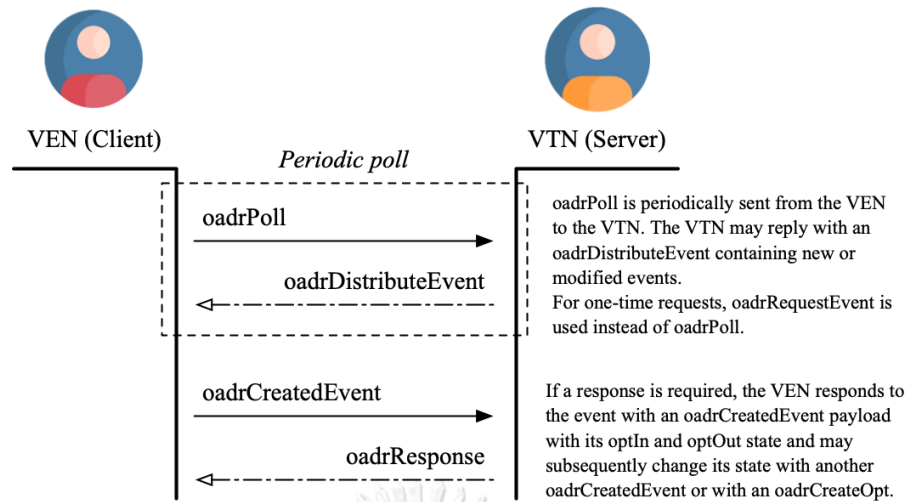
รูปที่ 41 โครงสร้างสถาปัตยกรรมการสื่อสารโพรโทคอล OpenADR

4.3.2 กระบวนการทำงานของโพรโทคอล OpenADR

บริการ (Services) ต่าง ๆ ของโพรโทคอล OpenADR [13] นั้นมีลักษณะการดำเนินการด้วยวิธี PUSH หรือ PULL ในการโต้ตอบสื่อสารกัน โดยสำหรับวิธีการ PUSH นั้น จะเป็นการส่งเหตุการณ์หรือการร้องขอจากโหนดต้นเสมือนไปยังโหนดปลายเสมือน โดยใช้ชุดข้อมูล `oadrDistributeEvent` ดังแสดงในรูปที่ 42 ส่วนในวิธีการ PULL นั้น โหนดปลายเสมือนจะทำการส่งชุดข้อมูลตรวจสอบสถานะ `oadrPoll` ไปยังโหนดต้นเสมือนเพื่อขอข้อมูลสถานะและเหตุการณ์ต่าง ๆ ที่เป็นปัจจุบัน จากนั้น `oadrDistributeEvent` จะถูกส่งในลักษณะการตอบกลับไปยังโหนดปลายเสมือนที่ได้ทำการส่งคำร้องขอมา ดังแสดงในรูปที่ 43



รูปที่ 42 ตัวอย่างการดำเนินการบริการด้วยวิธี PUSH

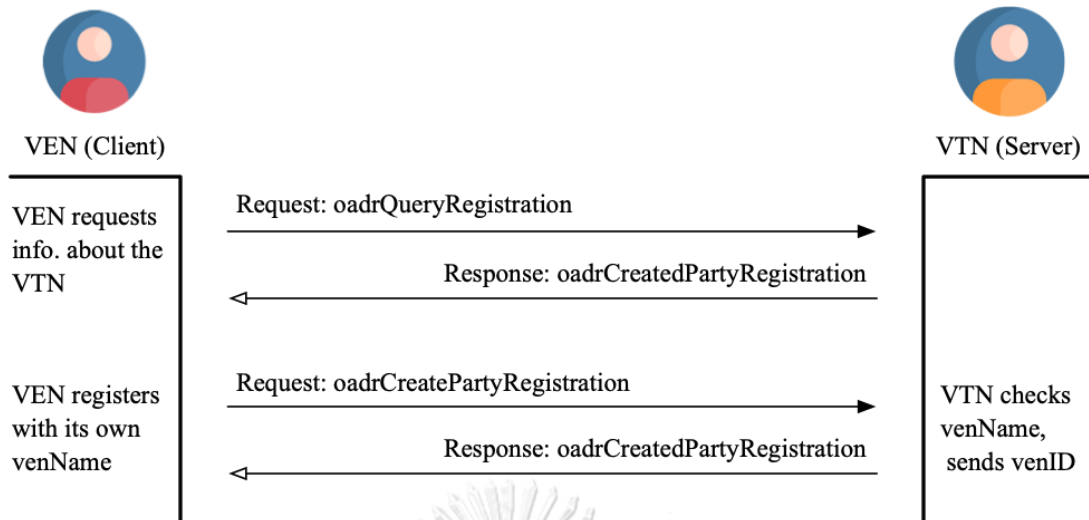


รูปที่ 43 ตัวอย่างการดำเนินการบริการด้วยวิธี PULL

4.3.2.1 บริการการลงทะเบียน (Registration Service)

การลงทะเบียนโหนดปลายเสมือนในระบบนั้น กระบวนการจะเริ่มจากที่โหนดปลายเสมือนทำการสอบถามไปยังโหนดต้นเสมือนเพื่อทำการขอข้อมูลเบื้องต้นโดยใช้ชุดข้อมูล oadrQueryRegistration ดังแสดงให้เห็นในรูปที่ 44 ซึ่งกระบวนการนี้จะเกิดขึ้นได้ก็ต่อเมื่อโหนดปลายเสมือนจะต้องรู้ที่อยู่ของโหนดต้นเสมือนเสียก่อน จากนั้นเมื่อโหนดต้นเสมือนได้รับชุดข้อมูลแล้ว จะทำการตอบกลับด้วยชุดข้อมูล oadrCreatedPartyRegistration ที่มีข้อมูลเบื้องต้นของโหนดต้นเสมือนเช่น ข้อมูลแสดงลักษณะเฉพาะ (Profile) หรือรูปแบบการสื่อสารที่โหนดต้นเสมือนรองรับตอบกลับไปยังโหนดปลายเสมือนที่ได้สอบถามมา

หลังจากนั้นโหนดปลายเสมือนจะเริ่มลงทะเบียนด้วยการส่งข้อมูลเบื้องต้นของโหนดปลายเสมือนโดยส่งชุดข้อมูล oadrCreatePartyRegistration ซึ่งมีข้อมูลแสดงลักษณะเฉพาะรวมถึงรูปแบบการสื่อสารที่โหนดปลายเสมือนเลือกใช้ ไปยังโหนดต้นเสมือน จากนั้นโหนดต้นเสมือนจะทำการตอบกลับด้วยชุดข้อมูล oadrCreatedPartyRegistration ที่มีข้อมูลการลงทะเบียนที่สำคัญต่าง ๆ เช่นข้อมูลเบื้องต้นของโหนดต้นเสมือน ไอดีการลงทะเบียน (registrationID) ไอดีของโหนดปลายเสมือน (venID) เป็นต้น



รูปที่ 44 การดำเนินการบริการลงทะเบียนระหว่างโหนดปลายเสมือนและโหนดต้นเสมือน

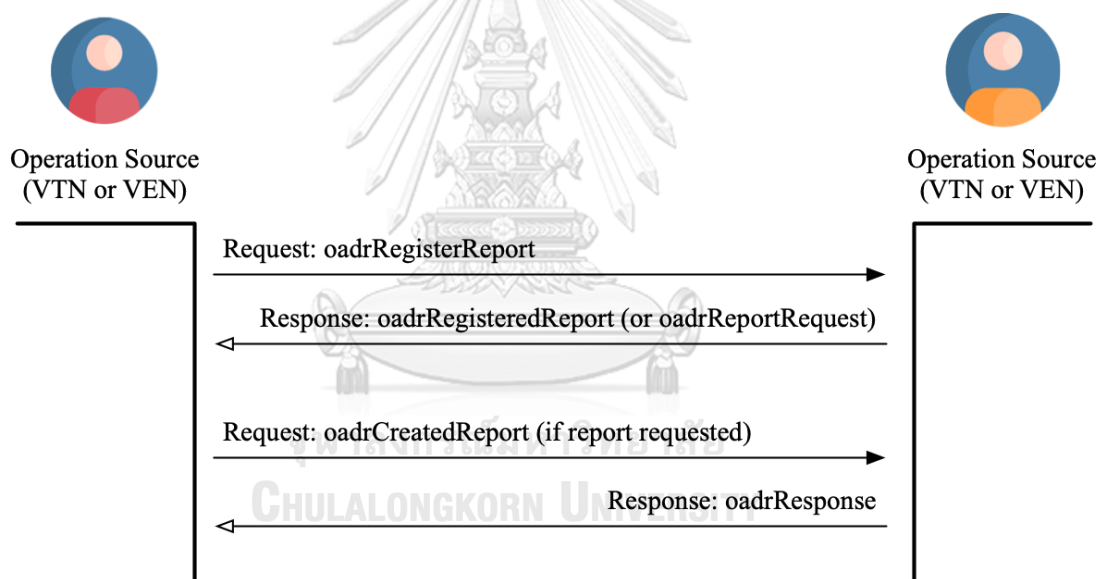
หากโหนดปลายเสมือนต้องการเปลี่ยนแปลงข้อมูลใด ๆ สามารถเรียกใช้ชุดข้อมูล oadrCreatePartyRegistration โดยแนบ registrationID เพื่อทำการลงทะเบียนใหม่ได้ทุกเมื่อ และนอกจากนี้ registrationID จะคงอยู่เสมอ จนกว่าฝ่ายใดฝ่ายหนึ่งยกเลิกการลงทะเบียนลง ในทางกลับกันหากโหนดต้นเสมือนมีการเปลี่ยนแปลงข้อมูล โหนดต้นเสมือนสามารถส่งคำร้องขอการลงทะเบียนของโหนดปลายเสมือนด้วยชุดข้อมูล oadrRequestReregistration ได้เช่นกัน

4.3.2.2 บริการเหตุการณ์ (Event Service)

เมื่อโหนดต้นเสมือนได้สร้างเหตุการณ์การตอบสนองด้านโหนดแล้วนั้น เหตุการณ์ต่าง ๆ จะถูกส่งไปยังโหนดปลายเสมือนด้วยชุดข้อมูล oadrDistributeEvent ซึ่งประกอบไปด้วยส่วนประกอบ (Element) ข้อมูลต่าง ๆ ที่สำคัญของเหตุการณ์ อาทิ ส่วนประกอบชุดข้อมูล oadrEvent ที่ระบุถึงเหตุการณ์ที่โหนดปลายเสมือนได้มีการลงทะเบียนเข้าร่วมมาตรการ อาจมีหนึ่งเหตุการณ์หรือมากกว่านั้น หรือส่วนประกอบชุดข้อมูล oadrResponseRequired ที่บอกถึงการร้องขอให้โหนดปลายเสมือนมีการตอบสนอง (Response) หรืออาจไม่ต้องการให้มีการตอบสนองขึ้นอยู่กับเหตุการณ์นั้น ๆ ตัวอย่างเช่นหากโหนดต้นเสมือนต้องการที่จะให้โหนดปลายเสมือนมีการตอบสนอง โหนดปลายเสมือนจะต้องตอบรับด้วยการระบุการเข้าร่วม (opt) ว่าเข้าร่วมหรือไม่เข้าร่วมมาตรการ และส่งตอบกลับไปด้วยชุดข้อมูล oadrCreatedEvent ซึ่งจะแนบส่วนประกอบอย่าง eventResponse ไปในแต่ละชุดข้อมูล oadrEvent ในการตอบกลับด้วย โดยในทางกลับกันหากโหนดต้นเสมือนไม่ต้องการให้มีการตอบสนอง โหนดปลายเสมือนจะไม่มีคำตอบกลับเหตุการณ์ไปยังโหนดต้นเสมือน ดังแสดงตัวอย่างของบริการเหตุการณ์ในรูปที่ 42 ก่อนหน้านี้

4.3.2.3 บริการรายงาน (Report Service)

ปกติแล้วบริการรายงานนี้จะเกิดขึ้นหลังจากที่ทำบริการลงทะเบียนเสร็จสิ้น อีกทั้งยังสามารถเริ่มได้จากทั้งโหนดต้นเสมือนหรือโหนดปลายเสมือน โดยกระบวนการของบริการรายงานจะเริ่มจากโหนดที่ต้องส่งการรายงานข้อมูลหรือโหนดต้นทาง จะทำการส่งความสามารถการรายงาน (Reporting capabilities) ไปยังโหนดปลายทางด้วยชุดข้อมูล oadrRegisterReport จากนั้นโหนดปลายทางจะตอบกลับด้วยชุดข้อมูล oadrRegisteredReport เพื่อรับทราบ หรืออาจตอบกลับด้วยชุดข้อมูล oadrRequestReport เพื่อร้องขอการรายงานที่มีข้อมูลเฉพาะที่โหนดปลายทางต้องการ โดยให้โหนดต้นทางสร้างข้อมูลนั้น ๆ และแนบตอบกลับมาในชุดข้อมูลต่อไปด้วย หากโหนดปลายทางมีการร้องขอรายงาน โหนดต้นทางจะทำการตอบกลับด้วยชุดข้อมูล oadrCreatedReport แต่หากไม่มีการร้องขอรายงานใด ๆ โหนดต้นทางจะถือว่าไม่มีข้อมูลใด ๆ ที่โหนดปลายทางต้องการ ดังนั้นจึงจะไม่มีคำตอบกลับไปยังโหนดปลายทาง โดยจะแสดงตัวอย่างให้เห็นในรูปที่ 45

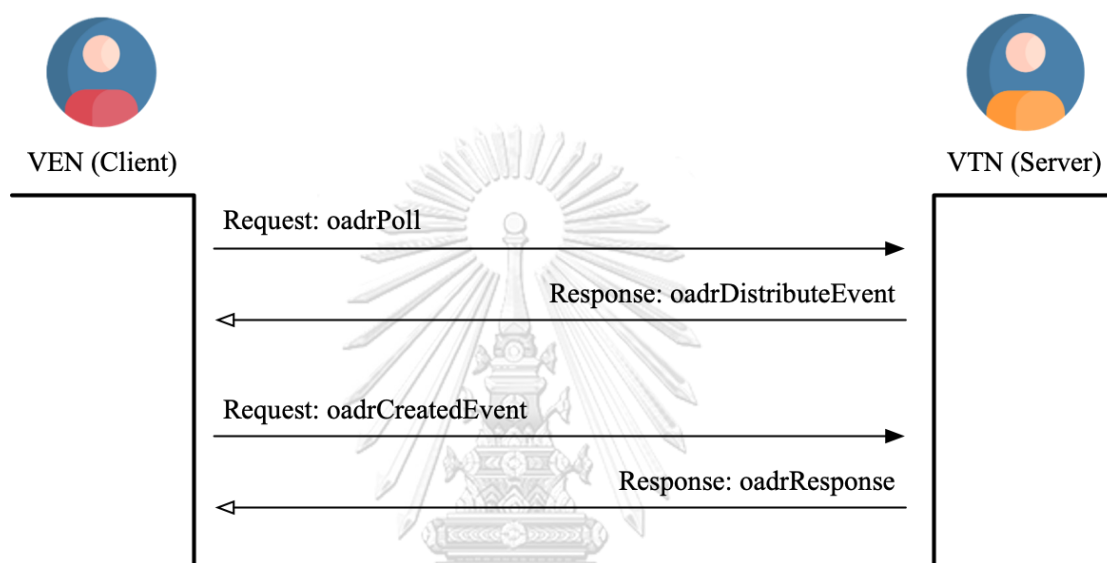


รูปที่ 45 การดำเนินการบริการรายงานระหว่างโหนดต้นทางและโหนดปลายทาง

4.3.2.4 บริการตรวจสอบสถานะ (OpenADR Poll)

การตรวจสอบสถานะ (Polling) เป็นบริการที่โหนดปลายเสมือนใช้ตรวจสอบสถานะจากโหนดต้นเสมือน เพื่อใช้ตรวจสอบสถานการณ์เชื่อมต่อ คำสั่งรายงาน หรือสำรวจเหตุการณ์เป็นระยะ ๆ โดยกระบวนการจะเริ่มจากที่โหนดปลายเสมือนส่งชุดข้อมูล oadrPoll ไปยังโหนดต้นเสมือน หากโหนดต้นเสมือนมีการสร้างเหตุการณ์สำหรับการตอบสนองด้านโหลดขึ้นมา โหนดต้นเสมือนจะทำการตอบกลับด้วยชุดข้อมูล oadrDistributeEvent ที่ประกอบไปด้วยรายละเอียดของเหตุการณ์ อาทิ อดี

เหตุการณ์ (eventID) เวลาที่สร้างเหตุการณ์ (createdDateTime) เวลาเริ่มการทำงาน (dateTimeStart) รวมถึงระยะเวลาการทำงาน (Duration) เป็นต้น เมื่อโหนดปลายเสมือนได้รับเหตุการณ์ โหนดปลายเสมือนจะทำการสร้างเหตุการณ์นั้นและจะตอบกลับไปยังโหนดด้วยชุดข้อมูล oadrCreatedEvent เพื่อตอบรับเหตุการณ์ จากนั้นโหนดต้นเสมือนจะตอบกลับด้วยชุดข้อมูล oadrResponse เพื่อรับทราบในการสร้างเหตุการณ์ของโหนดปลายเสมือน ดังแสดงให้เห็นในรูปที่ 46

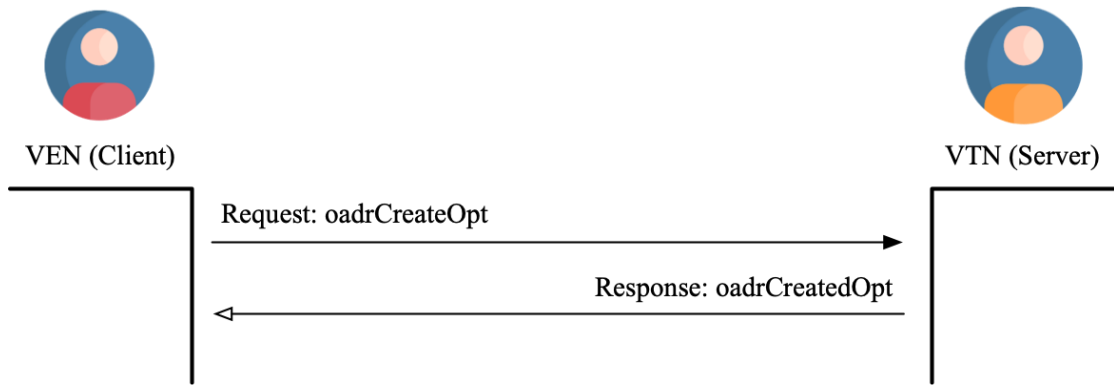


รูปที่ 46 การดำเนินการบริการตรวจสอบสถานะระหว่างโหนดปลายเสมือนและโหนดต้นเสมือน

4.3.2.5 บริการเข้าร่วมมาตรการ (Opt Service)

ในการสื่อสารของโพรโทคอล OpenADR อีกหนึ่งบริการที่สำคัญคือบริการเข้าร่วมมาตรการ ซึ่งเป็นบริการที่ผู้ใช้ไฟฟ้าหรือโหนดปลายเสมือนใช้ในการแจ้งไปยังโหนดต้นเสมือนหรือผู้ให้บริการว่าเราต้องการเข้าร่วมมาตรการหรือไม่ โดยสามารถระบุได้ว่าต้องการเข้าร่วมมาตรการ (optIn) หรือไม่เข้าร่วมมาตรการ (optOut)

โดยกระบวนการของบริการนี้สามารถดำเนินการได้หลังจากที่โหนดปลายเสมือนได้ทำการลงทะเบียน รวมถึงต้องได้รับข้อมูลเหตุการณ์จากโหนดต้นเสมือนแล้ว จากนั้นกระบวนการจะเริ่มจากที่โหนดปลายเสมือนจะทำการส่งชุดข้อมูล oadrCreateOpt ไปยังโหนดต้นเสมือนเพื่อร้องขอที่จะเข้าร่วมมาตรการโดยระบุ optID ไปกับชุดข้อมูลด้วย จากนั้นโหนดต้นเสมือนจะทำการตอบกลับด้วยชุดข้อมูล oadrCreatedOpt ที่มี optID นั้น ๆ กลับมายังโหนดปลายเสมือนเพื่อรับทราบและยืนยันการเข้าร่วม ดังแสดงให้เห็นในรูปที่ 47






รูปที่ 47 การดำเนินการบริการเข้าร่วมมาตรการระหว่างโหนดปลายเสมือนและโหนดต้นเสมือน



4.3.3 การพัฒนาซอฟต์แวร์ OpenADR

ในงานวิจัยฉบับนี้จะนำซอฟต์แวร์ OpenADR มาพัฒนาและประยุกต์ใช้งาน โดยซอฟต์แวร์ OpenADR นั้นเป็นซอฟต์แวร์แบบโอเพนซอร์ส ทำให้สามารถนำมาทดลองประยุกต์ใช้งานได้โดยไม่มีค่าใช้จ่าย ซึ่งซอฟต์แวร์แต่ละชนิดจะมีระบบปฏิบัติการที่รองรับรวมถึงภาษาโปรแกรมที่ใช้งานแตกต่างกันออกไปดังที่แสดงให้เห็นในตารางที่ 2

ตารางที่ 2 รายละเอียดเบื้องต้นของโหนดต้นเสมือนและโหนดปลายเสมือน

	 VTN	 VEN	 VEN
Role	Virtual Top Node	Virtual End Node	Virtual End Node
Designed Use	DRMS	Desktop Client	Embedded Client
License	BSD 3-Clause	BSD 3-Clause	BSD 3-Clause
Profile	2.0a and 2.0b	2.0b	2.0b
Data Models	Push/Pull (Poll)	Pull (Poll)	Pull (Poll)
Transports	HTTP, XMPP	HTTP	HTTP
Programing Language	JRuby, Java	C#	C++
Tested Operating System	Linux, MacOS	Windows	C++

ในส่วนของโหนดปลายเสมือนนั้นสามารถพัฒนาซอฟต์แวร์ได้ทั้งในภาษาโปรแกรม C# และ C++ โดยขึ้นอยู่กับอุปกรณ์ที่ต้องการใช้งานดังตารางที่ 2 หากต้องการใช้งานเพื่อทดสอบการทำงานเบื้องต้นหรือเพื่อทำความเข้าใจการสื่อสารของโพรโทคอล ซอฟต์แวร์โหนดปลายเสมือนภาษาโปรแกรม C# นั้นสามารถตอบโจทย์นี้ได้ดีเนื่องจากมีหน้าต่างส่วนต่อประสานกับผู้ใช้ (User Interface) ทำให้สามารถทำความเข้าใจได้ไม่ยาก แต่สำหรับงานวิจัยฉบับนี้ต้องการที่จะนำซอฟต์แวร์ OpenADR มาพัฒนาอยู่ในรูปแบบอุปกรณ์ฝังตัว ดังนั้นผู้วิจัยจึงเลือกใช้ซอฟต์แวร์โอเพนซอร์สโหนดปลายเสมือนภาษาโปรแกรม C++ ในการพัฒนามิเตอร์อัจฉริยะนี้ โดยจะอธิบายในหัวข้อการพัฒนาซอฟต์แวร์โหนดปลายเสมือนต่อไป

4.3.3.1 การพัฒนาซอฟต์แวร์โหนดต้นเสมือน

ในงานวิจัยฉบับนี้ ผู้วิจัยได้ทำการพัฒนาซอฟต์แวร์โหนดต้นเสมือนด้วยซอฟต์แวร์รุ่น 0.9.7 ซึ่งเป็นซอฟต์แวร์การสื่อสาร OpenADR ฟังก์ชันเฟเวอร์ (OpenADR-Virtual-Top-Node-0.9.7) โดยใช้ระบบปฏิบัติการ Ubuntu 20.04.2 LTS ซึ่งเป็นระบบปฏิบัติการลินุกซ์บนเครื่องคอมพิวเตอร์ตั้งโต๊ะ แต่เนื่องด้วยซอฟต์แวร์เป็นแบบโอเพนซอร์ส และอาจมีความไม่เข้ากันบางอย่างกับระบบปฏิบัติการ Ubuntu 20.04 LTS ทำให้ระหว่างกระบวนการการสร้างโหนดต้นเสมือนอาจเกิดข้อผิดพลาดขึ้น ผู้วิจัยจึงได้ทำการปรับเปลี่ยนแก้ไขและทำการติดตั้งแพ็คเกจซอฟต์แวร์ที่จำเป็นใหม่ โดยสามารถสรุปได้ดังนี้

- Ruby, Ruby-dev
- OpenJDK 8
- PostgreSQL 11 หรือต่ำกว่า
- Torquebox 3.0.2

หลังจากที่ทำการติดตั้งแพ็คเกจซอฟต์แวร์ข้างต้นแล้ว จะสามารถติดตั้งซอฟต์แวร์โหนดต้นเสมือนตามคู่มือการใช้งานได้ [14] และเมื่อทำการติดตั้งซอฟต์แวร์โหนดปลายเสมือนเรียบร้อยแล้ว ก่อนที่จะสามารถเริ่มดำเนินการโปรแกรมได้ จะต้องสั่งทำการเปิดใช้งานซอฟต์แวร์เสียก่อนด้วยคำสั่ง

```
~$ torquebox deploy
```

จากนั้นให้ทำการใส่ค่าที่อยู่ไอพี (IP Address) ของเครื่องโหนดต้นเสมือนลงไปคำสั่งเพื่อเริ่มการดำเนินงานโปรแกรมโหนดต้นเสมือนด้วยไอพีนั้น ๆ

```
~$ torquebox run -b 161.200.85.66
```

หลังจากที่ทำการกระบวนการทั้งหมดเสร็จสิ้น จะสามารถเรียกหน้าต่างเข้าสู่ระบบของโหนดต้นเสมือนได้ผ่านเว็บเบราว์เซอร์ โดยใส่ที่อยู่ไอพีของเครื่องโหนดต้นเสมือนตามด้วยพอร์ต 8080 ต่อท้าย เช่น <http://161.200.85.66:8080> หากสามารถขึ้นหน้าต่างเข้าสู่ระบบแบบที่แสดงในการทดสอบดังรูปที่ 66 หมายความว่าโหนดต้นเสมือนนั้นพร้อมใช้งานแล้ว เป็นอันเสร็จสิ้นกระบวนการสร้างโหนดต้นเสมือนของโพรโทคอล OpenADR

4.3.3.2 การพัฒนาซอฟต์แวร์โหนดปลายเสมือน

อย่างที่ทราบไปแล้วนั้นว่าซอฟต์แวร์โหนดปลายเสมือนรองรับทั้งภาษาโปรแกรม C# และ C++ สำหรับซอฟต์แวร์โหนดปลายเสมือนที่จะใช้ในการพัฒนาในงานวิจัยฉบับนี้ ผู้วิจัยเลือกใช้ซอฟต์แวร์ภาษาโปรแกรม C++ เนื่องจากมีความยืดหยุ่นในการทำงานได้มากกว่า สามารถนำมาประยุกต์ใช้ในอุปกรณ์ระบบฝังตัวอย่างบอร์ดราสเบอร์รี่พายได้ แต่ในทางกลับกันซอฟต์แวร์ตัวนี้มีความซับซ้อนต่อการนำมาประยุกต์ใช้งานเช่นกันเมื่อเทียบกับภาษาโปรแกรม C# โดยซอฟต์แวร์โหนดปลายเสมือนนี้มีส่วนประกอบคลังโปรแกรม (Library) ที่สำคัญ ได้แก่

1. *googletest-release-1.7.0* คือ เพิ่มข้อมูลคลังโปรแกรมทดสอบ *gtest* รุ่น 1.7.0 ซึ่งจะถูกเรียกใช้โดยโปรแกรม *oadrtest*
2. *oadr* คือ เพิ่มคลังโปรแกรม OpenADR 2.0 ที่ใช้สำหรับสร้างโหนดปลายเสมือน
3. *oadrsd* คือ เพิ่มคลังโปรแกรมสำหรับจัดเรียงอนุกรมข้อมูล (serialize or deserialize) สำหรับข้อความ XML ของโหนดปลายเสมือนที่ใช้ในการตอบโต้กับโหนดต้นเสมือน
4. *oadrtest* คือ เพิ่มชุดข้อมูลทดสอบที่จะใช้ร่วมกับคลังโปรแกรม *gtest*
5. *Samplevenmgr* คือ เพิ่มโปรแกรมตัวอย่างการใช้งานเบื้องต้น

แต่ก่อนที่จะนำซอฟต์แวร์ไปใช้งานได้นั้น ผู้พัฒนาจะต้องทำการปรับปรุงแก้ไขคลังโปรแกรมให้เหมาะสมต่อการใช้งานเสียก่อน โดยซอฟต์แวร์ OpenADR ของโหนดปลายเสมือนนี้จะมีโปรแกรมต่อประสาน (interface) ที่สำคัญที่ใช้ในการสื่อสารแลกเปลี่ยนข้อความกับโหนดต้นเสมือน โดยส่วนต่อประสานต่าง ๆ จะถูกจัดเก็บอยู่ในแฟ้ม *oadr* ซึ่งจะมีดังต่อไปนี้

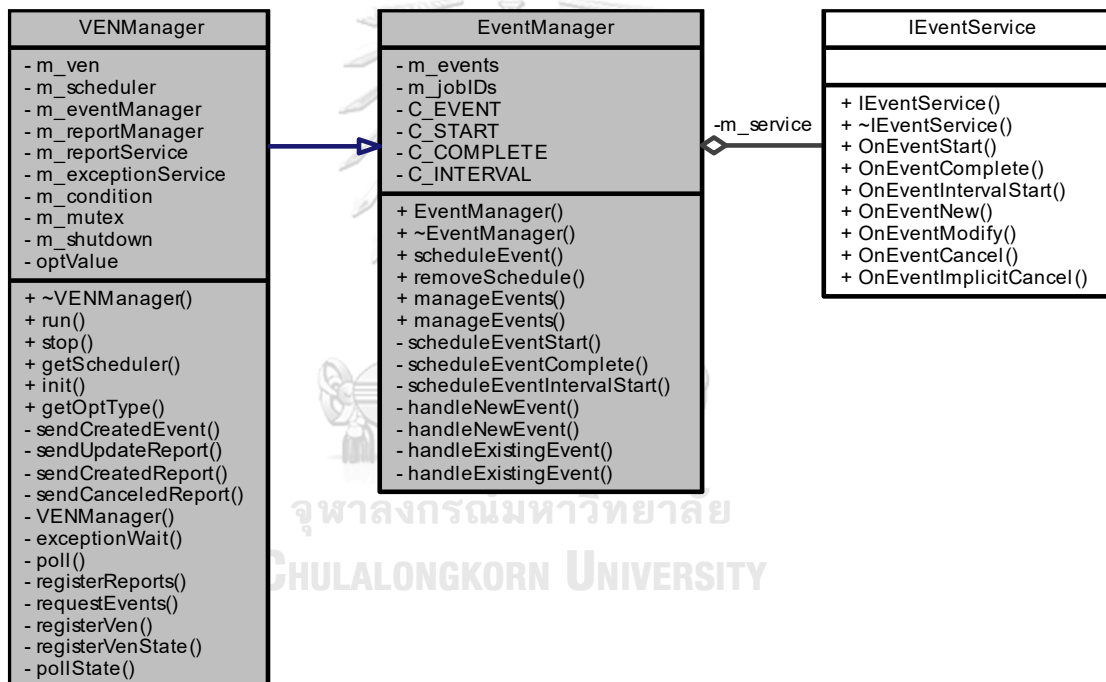
- *IEventService* เป็นส่วนต่อประสานทำหน้าที่เกี่ยวกับบริการเหตุการณ์ (Event Services)
- *IReportService* เป็นส่วนต่อประสานทำหน้าที่เกี่ยวกับบริการรายงาน (Report Services)
- *IOadrMessage* เป็นส่วนต่อประสานทำหน้าที่เกี่ยวกับรับส่งข้อความในการโต้ตอบต่าง ๆ
- *IOADRExceptionService* เป็นส่วนต่อประสานทำหน้าที่เกี่ยวกับข้อผิดพลาดในการสื่อสาร

สำหรับงานวิจัยฉบับนี้มีจุดมุ่งหมายที่จะพัฒนาซอฟต์แวร์โหนดปลายเสมือนให้มีความสามารถในการตอบรับเหตุการณ์เข้าร่วมมาตรการการตอบสนองด้านโหลด (opt) จากเหตุการณ์ที่โหนดต้นเสมือนสร้างขึ้น โดยผู้วิจัยจะทำการปรับเปลี่ยนแก้ไขฟังก์ชันที่เกี่ยวกับการตอบรับเข้าร่วมมาตรการ (optIn/optOut) โดยการส่งผ่านค่าเริ่มต้นเข้าร่วมมาตรการ (Default opt) จากภายนอกเข้าไปยังคลังโปรแกรม ซึ่งฟังก์ชันต่าง ๆ ที่กล่าวมาจะมีความเกี่ยวข้องกับส่วนต่อประสาน *IEventService* เนื่องจากเป็นส่วนต่อประสานหลักที่เกี่ยวกับบริการเหตุการณ์รูปแบบต่าง ๆ

คลาส (Class) การทำงานที่เกี่ยวข้องกับค่าเริ่มต้นเข้าร่วมมาตรการ (Default opt) ที่จะทำการปรับปรุงในงานวิจัยฉบับนี้นั้นมี 3 คลาส ได้แก่

- IEventService เป็นคลาสแม่แบบ (Template) ที่มีฟังก์ชันเท่ากับศูนย์ กล่าวคือเป็นฟังก์ชันที่ไม่ทำให้เกิดผลใด ๆ แต่จะสามารถถูกเรียกและนำไปประยุกต์ใช้งานได้
- EventManager เป็นคลาสที่นำคลาสแม่แบบ IEventService มาประยุกต์ใช้งานให้เกิดผล
- VENManager เป็นคลาสหลักของโหนดปลายเสมือนที่มีการเรียกใช้ฟังก์ชันของ EventManager มาใช้งาน โดยผ่านตัวชี้ (Pointer) เพื่อส่งผ่านค่าเริ่มต้นเข้าร่วมมาตรการไปยังฟังก์ชันของ EventManager

โดยจะแสดงความสัมพันธ์ของคลาสทั้งสามได้ดังรูปที่ 48



รูปที่ 48 แสดงความสัมพันธ์ของคลาส VENManager EventManager และ IEventService

จากรูปที่ 48 จะเห็นว่าคลาส IEventService นั้นมีฟังก์ชันสำหรับนำไปใช้งานในกรณีต่าง ๆ ของเหตุการณ์ แต่ฟังก์ชันที่เกี่ยวข้องโดยตรงกับการนำค่าเริ่มต้นเข้าร่วมมาตรการไปทำให้เกิดผลจะมีอยู่ 3 ฟังก์ชันด้วยกันได้แก่

- OnEventNew คือ ฟังก์ชันที่ทำงานเมื่อได้รับเหตุการณ์ใหม่จากโหนดต้นเสมือน
- OnEventModify คือ ฟังก์ชันที่ทำงานเมื่อได้รับเหตุการณ์ที่มีการแก้ไขจากโหนดต้นเสมือน
- OnEventCancle คือ ฟังก์ชันที่ทำงานเมื่อได้รับเหตุการณ์ที่ถูกยกเลิกจากโหนดต้นเสมือน

ฟังก์ชันทั้งสามนี้จะมีตัวแปรสำหรับรับค่าเริ่มต้นเข้าร่วมมาตรการชื่อ “optType” ซึ่งจะสามารถมีค่าสองค่าได้แก่ เข้าร่วมมาตรการ (optIn) และไม่เข้าร่วมมาตรการ (optOut) โดยสามารถอธิบายความสัมพันธ์ของฟังก์ชันและค่าเริ่มต้นเข้าร่วมมาตรการพอสังเขปได้ว่า เมื่อโหนดปลายเสมือนได้รับเหตุการณ์ใหม่ แก๊ซ หรือยกเลิกจากโหนดต้นเสมือน ค่าเริ่มต้นเข้าร่วมมาตรการจะถูกส่งผ่านไปยังฟังก์ชันทั้งสามและตอบกลับไปยังโหนดต้นเสมือน ทำให้ผู้ใช้สามารถตอบรับการเข้าร่วมมาตรการได้

กระบวนการส่งผ่านค่าเริ่มต้นเข้าร่วมมาตรการจะเริ่มจากที่คลาส EventManager มีฟังก์ชันชั้นในสุดอย่าง *handleNewEvent* และ *handleExistingEvent* ซึ่งเป็นฟังก์ชันที่นำฟังก์ชันแม่แบบของ IEventService มาประยุกต์ใช้งานเพื่อทำให้เกิดผล ผู้วิจัยจะเพิ่มตัวแปรสำหรับรับค่าเริ่มต้นเข้าร่วมมาตรการที่มาจากภายนอกคลังโปรแกรมขึ้นมาหนึ่งตัวแปรให้กับทั้งสองฟังก์ชันข้างต้นและจากนั้นจะส่งผ่านไปยัง *OnEventNew* *OnEventModify* และ *OnEventCandle* เพื่อทำให้เกิดผล โดยที่ฟังก์ชัน *handleNewEvent* จะเป็นฟังก์ชันที่เรียกใช้งานฟังก์ชัน *OnEventNew* ผ่านตัวชี้ชื่อ *m_service* ในการประยุกต์ใช้งานเมื่อได้รับเหตุการณ์ใหม่ ๆ และฟังก์ชัน *handleExistingEvent* ซึ่งเป็นฟังก์ชันที่เรียกใช้งานฟังก์ชัน *OnEventModify* กับ *OnEventCandle* ผ่านตัวชี้ *m_service* ในการประยุกต์ใช้งานเมื่อโหนดปลายเสมือนได้รับเหตุการณ์ที่มีการแก๊ซหรือเหตุการณ์ที่ถูกยกเลิก

จากนั้นฟังก์ชัน *handleNewEvent* และ *handleExistingEvent* นั้นจะถูกนำมารวมอยู่ในฟังก์ชันที่ชื่อ *manageEvents* โดยผู้วิจัยจะเพิ่มตัวแปรส่งผ่านค่าให้กับฟังก์ชันนี้เช่นเดียวกับสองฟังก์ชันก่อนหน้า โดยที่ฟังก์ชันนี้จะประกาศฟังก์ชันเป็นแบบสาธารณะ (Public) ทำให้คลาสอื่น ๆ สามารถเรียกใช้ฟังก์ชันนี้ของคลาส EventManager ได้

หลังจากนั้นฟังก์ชัน *poll* และ *requestEvents* ของคลาส VENManager จะเป็นส่วนที่นำฟังก์ชัน *manageEvents* ของคลาส EventManager มาใช้งาน โดยจะสามารถส่งผ่านค่าเริ่มต้นเข้าร่วมมาตรการผ่านตัวชี้ *m_eventManager* ไปยังฟังก์ชัน *handleNewEvent* และ *handleExistingEvent* โดยผ่านฟังก์ชัน *manageEvents* ได้

นอกจากนี้ ไฟล์หลัก (main) ของซอฟต์แวร์ก็เป็นอีกส่วนที่มีความสำคัญเช่นกัน เนื่องจากเป็นส่วนที่บริหารจัดการซอฟต์แวร์หลัก รวมถึงเป็นส่วนต่อประสานกับผู้ใช้งานโปรแกรมซึ่งจะต้องป้อนข้อมูลเฉพาะในการเปิดโปรแกรมสำหรับเริ่มการทำงานซอฟต์แวร์โหนดปลายเสมือน โดยจะมีฟังก์ชันที่ทำหน้าที่รับข้อมูลนำเข้าคือฟังก์ชัน *parseParams* ดังแสดงให้เห็นในรูปที่ 49 เมื่อผู้ใช้งานต้องการเรียกใช้งานซอฟต์แวร์ ผู้ใช้งานจำเป็นต้องใส่อาร์กิวเมนต์หรือข้อมูลให้ครบถ้วนจึงจะสามารถเปิดโปรแกรมโหนดปลายเสมือนได้ โดยจะมี 3 อาร์กิวเมนต์หลัก ๆ คือ ชื่อโปรแกรมโหนดปลาย

เสมือน ที่อยู่ไอพีหรือชื่อโดเมนของโหนดต้นเสมือนที่ต้องการเชื่อมต่อ และชื่อโหนดปลายทางเสมือนที่ลงทะเบียนไว้แล้ว ทั้งนี้ผู้วิจัยจึงทำการปรับแต่งซอฟต์แวร์ให้สามารถรับอาร์กิวเมนต์เพิ่มอีกหนึ่งตัวคือค่าเริ่มต้นเข้าร่วมมาตรการ (Default opt) ทำให้สามารถรับค่าเริ่มต้นเข้าร่วมมาตรการจากผู้ใช้งานซอฟต์แวร์ได้โดยตรง ซึ่งเมื่อผู้ใช้งานเรียกใช้งานซอฟต์แวร์จะทำการเลือกที่จะเข้าร่วมมาตรการ (optIn) หรือ ไม่เข้าร่วมมาตรการ (optOut) ได้ ซึ่งค่าดังกล่าวจะถูกส่งผ่านไปยังคลังโปรแกรมในคลาส VENManager ในการสั่งฟังก์ชันต่อ ๆ ไป



รูปที่ 49 แสดงฟังก์ชันที่เกี่ยวข้องในไฟล์หลัก (main)

หลังจากที่แก้ไขคลังโปรแกรม ซอฟต์แวร์จะถูกนำมาแปลโปรแกรม (Compile) บนบอร์ดราสเบอร์รี่พายรุ่น 4 แรม 4 Gb ที่ได้ทำการติดตั้งระบบปฏิบัติการ Ubuntu Desktop 21.04 โดยนำซอฟต์แวร์ OpenADR โหนดปลายทางเสมือนที่ใช้ภาษาโปรแกรม C++ รุ่น 0.7.0 (OpenADR-VEN-Library-0.7.0) ที่ผู้วิจัยได้ทำการปรับปรุงเรียบร้อยแล้วมาทำการแปลโปรแกรม เพื่อสร้างชุดคลังโปรแกรมสำเร็จที่จะนำไปใช้งานสำหรับบอร์ดคอมพิวเตอร์อัจฉริยะในงานวิจัยฉบับนี้ ซึ่งวิธีการสร้างคลังโปรแกรมเพื่อที่จะนำมาใช้งานนั้นเริ่มจากการติดตั้งแพ็คเกจซอฟต์แวร์ที่จำเป็นต่าง ๆ ดังนี้

- Libcurl
- OpenSSL
- Xerces C++
- Cmake
- G++

โดยสามารถติดตั้งแพ็คเกจซอฟต์แวร์ได้ด้วยคำสั่ง

```
~$ sudo apt install g++ cmake libxerces-c-dev libcurl4-openssl-dev
```

เมื่อทำการติดตั้งแพ็คเกจซอฟต์แวร์เรียบร้อยแล้ว จะสามารถสร้างคลังโปรแกรมสำเร็จได้โดยสามารถทำตามขั้นตอนในคู่มือการสร้างคลังโปรแกรมสำเร็จของโหนดปลายเสมือน [15] โดยรูปที่ 50 นี้จะแสดงถึงกระบวนการสร้างคลังโปรแกรม หากกระบวนการสร้างไม่มีข้อผิดพลาดจะทำให้ได้ไฟล์คลังโปรแกรมสำเร็จออกมา ซึ่งไฟล์ดังกล่าวนี้จะสามารถนำไปเรียกใช้งานโหนดปลายเสมือนได้ต่อไป

```
[ 83%] Building CXX object CMakeFiles/oadr.dir/oadr/oadr/scheduler/IScheduler.cpp.o
[ 83%] Building CXX object CMakeFiles/oadr.dir/oadr/oadr/scheduler/JobLambda.cpp.o
[ 83%] Building CXX object CMakeFiles/oadr.dir/oadr/oadr/scheduler/Scheduler.cpp.o
[ 84%] Building CXX object CMakeFiles/oadr.dir/oadr/oadr/scheduler/SchedulerJob.cpp.o
[ 84%] Building CXX object CMakeFiles/oadr.dir/oadr/oadr/scheduler/SchedulerJobs.cpp.o
[ 84%] Building CXX object CMakeFiles/oadr.dir/oadr/oadr/ven/IOadrMessage.cpp.o
[ 84%] Building CXX object CMakeFiles/oadr.dir/oadr/oadr/ven/VEN2b.cpp.o
[ 85%] Building CXX object CMakeFiles/oadr.dir/oadr/oadr/ven/http/CurlBuffer.cpp.o
[ 85%] Building CXX object CMakeFiles/oadr.dir/oadr/oadr/ven/http/CurlException.cpp.o
[ 85%] Building CXX object CMakeFiles/oadr.dir/oadr/oadr/ven/http/CurlReceiveBuffer.cpp.o
[ 86%] Building CXX object CMakeFiles/oadr.dir/oadr/oadr/ven/http/CurlSendBuffer.cpp.o
[ 86%] Building CXX object CMakeFiles/oadr.dir/oadr/oadr/ven/http/HttpCurl.cpp.o
[ 86%] Building CXX object CMakeFiles/oadr.dir/oadr/oadr/ven/http/IHttp.cpp.o
[ 87%] Linking CXX shared library liboadr.so
[ 87%] Built target oadr
```

รูปที่ 50 แสดงกระบวนการสร้างคลังโปรแกรมโหนดปลายเสมือน



บทที่ 5

การทดลองและผลการทดลอง

5.1 ทดสอบวัดค่าพลังงานทางไฟฟ้าจากการใช้งานจริง

ในการวัดค่าพลังงานทางไฟฟ้าของมิเตอร์อัจฉริยะต้นแบบที่ถูกพัฒนาขึ้นนั้นจะทำการทดลองกับการวัดค่าไฟฟ้ากระแสสลับแบบเฟสเดียว โดยจะเชื่อมต่อโหลดเข้ากับมิเตอร์อัจฉริยะเฟสเดียวที่สอบเทียบตัวเองผ่านรางเต้าเสียบที่ได้ทำการดัดแปลงเพื่อการทดสอบ สำหรับตัวอย่างโหลดที่นำมาทำการทดสอบคือเครื่องฟอกอากาศยี่ห้อ TOSHIBA รุ่น CAF-H50(W) และผลการทดสอบวัดค่าพลังงานจะแสดงให้เห็นในรูปที่ 51 - 53



รูปที่ 51 การทดสอบวัดค่าพลังงานทางไฟฟ้าของมิเตอร์อัจฉริยะต้นแบบกับโหลด

```

| 14 August 2021 | 19:04:13 |
SELECT rowid,* FROM ADE_dataLog ORDER BY rowid DESC LIMIT 1;
Callback function called: rowid = 4
Timestamp = 14/08/2021-19:04:13
IRMS = 0.02
VRMS = 223.88
ActPWR = 0.99
ReaPWR = -3.44
AppPWR = 3.92
PF = 0.26
Freq = 49.96
Temp = 31.6

```

รูปที่ 52 การทดสอบวัดค่าพลังงานทางไฟฟ้าของมิเตอร์อัจฉริยะต้นแบบเมื่อ “ปิด” การใช้งานโหลด

```

| 14 August 2021 | 19:07:13 |
SELECT rowid,* FROM ADE_dataLog ORDER BY rowid DESC LIMIT 1;
Callback function called: rowid = 10
Timestamp = 14/08/2021-19:07:13
IRMS = 0.43
VRMS = 223.85
ActPWR = 55.85
ReaPWR = -10.38
AppPWR = 95.97
PF = 0.58
Freq = 49.97
Temp = 31.03

```

รูปที่ 53 การทดสอบวัดค่าพลังงานทางไฟฟ้าของมิเตอร์อัจฉริยะต้นแบบเมื่อ “เปิด” การใช้งานโหลด



5.2 ทดสอบความผิดพลาดการวัดค่าพลังงานทางไฟฟ้าของมิเตอร์

การทดสอบความผิดพลาดการวัดค่าทางไฟฟ้าของมิเตอร์ภายใต้การทดสอบ (Meter Under Test; MUT) นี้จะใช้แหล่งจ่ายพลังงานอ้างอิง CALSOURCE 200 ซึ่งเป็นแหล่งจ่ายไฟฟ้าที่สามารถกำหนดขนาดรวมถึงมุมเฟสของแรงดันและกระแสไฟฟ้าได้ ร่วมกับมิเตอร์อ้างอิง (Meter Test Equipment; MTE) PRS 1.3 ซึ่งเป็นมิเตอร์ที่ใช้สำหรับทดสอบจากบริษัท MTE Meter Test Equipment AG ในการทดสอบมิเตอร์อัจฉริยะของงานวิจัยฉบับนี้ให้เป็นไปตามมาตรฐาน IEC 62053-21



รูปที่ 54 เครื่องมือที่ใช้ในการทดสอบความผิดพลาดการวัดของมิเตอร์อัจฉริยะ

วิทยานิพนธ์ฉบับนี้ได้กำหนดการทดสอบตามมาตรฐาน IEC 62053-21 [16, 17] ซึ่งเป็นการทดสอบมาตรฐานของอุปกรณ์ที่ใช้เชื่อมต่อกับโครงข่ายไฟฟ้า สำหรับระบบจำหน่ายไฟฟ้าแรงดันต่ำ (ไม่เกิน 1000 โวลต์) Class 1 โดยมีข้อกำหนดตามหลักเกณฑ์การทดสอบดังแสดงในตารางที่ 3 ซึ่งจะใช้ค่ากระแสไฟฟ้าใช้งานปกติ (I_n) 5 A ค่ากระแสไฟฟ้าสูงสุด (I_{max}) 10 A และค่าแรงดันไฟฟ้าปกติ 230 V ที่ความถี่ 50 Hz ในการทดสอบมาตรฐานนี้

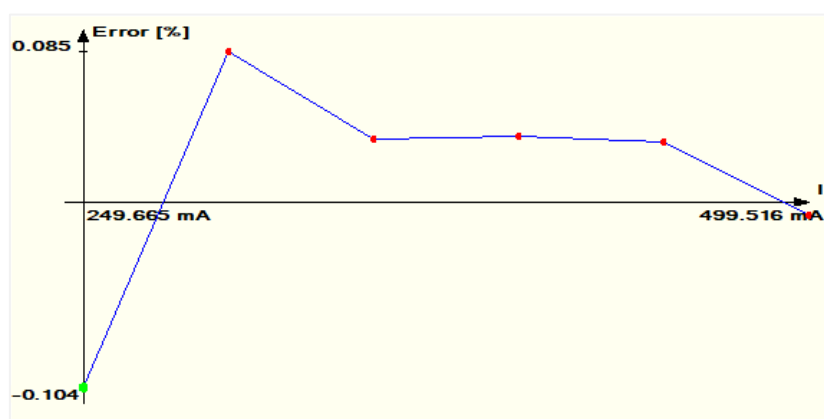
ตารางที่ 3 ข้อกำหนดหลักเกณฑ์การทดสอบตามมาตรฐาน IEC 62053-21 Class 1

Values of Current	Power Factor	Percentage Error Limits Class 1
$0.05I_n \leq I \leq 0.1I_n$	1	± 1.5
$0.1I_n \leq I \leq I_{max}$	1	± 1.0
$0.1I_n \leq I \leq 0.2I_n$	0.5 Inductive	± 1.5
	0.8 Capacitive	± 1.5
$0.2I_n \leq I \leq I_{max}$	0.5 Inductive	± 1.0
	0.8 Capacitive	± 1.0
$0.2I_n \leq I \leq I_n$	0.25 Inductive	± 3.5
	0.5 Capacitive	± 2.5

5.2.1 ผลการทดสอบความผิดพลาดการวัดค่าพลังงานไฟฟ้าของมิเตอร์

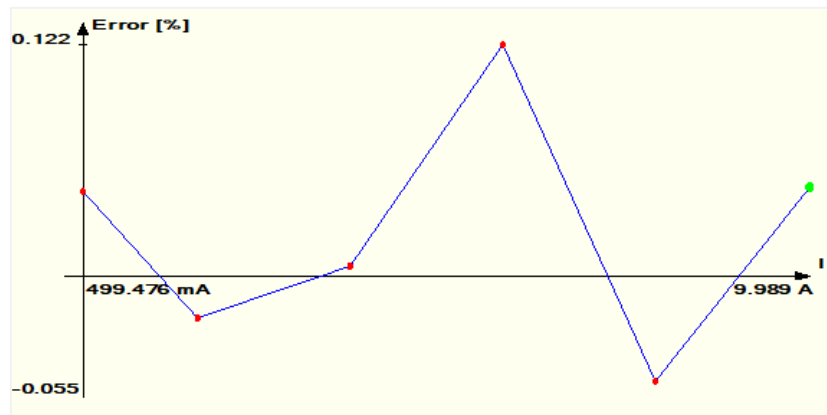
ผลการทดสอบจะแบ่งออกเป็น 8 ระยะเวลาตามข้อกำหนดหลักเกณฑ์การทดสอบมาตรฐานในตารางที่ 3 ซึ่งจะสามารถแสดงได้ตามลำดับ ดังนี้

ผลการทดสอบระยะที่ 1 เป็นการทดสอบที่ค่ากระแสไฟฟ้า $0.05I_n \leq I \leq 0.1I_n$ ที่ค่าตัวประกอบกำลังเท่ากับ 1 โดยจะมีค่าความผิดพลาดสูงสุดเท่ากับ -0.104% ซึ่งผ่านข้อกำหนดตามหลักเกณฑ์การทดสอบความแม่นยำโดยมีความผิดพลาดไม่เกิน $\pm 1.5\%$



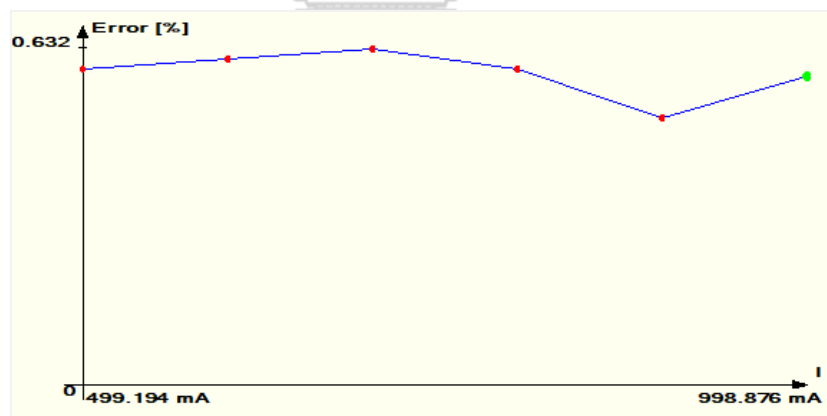
รูปที่ 55 ผลการทดสอบที่ค่ากระแสไฟฟ้า $0.05I_n \leq I \leq 0.1I_n$ ตัวประกอบกำลังเท่ากับ 1

ผลการทดสอบระยะที่ 2 เป็นการทดสอบที่ค่ากระแสไฟฟ้า $0.1I_n \leq I \leq I_{max}$ ที่ค่าตัวประกอบกำลังเท่ากับ 1 โดยจะมีค่าความผิดพลาดสูงสุดเท่ากับ **0.122%** ซึ่งผ่านข้อกำหนดตามหลักเกณฑ์การทดสอบความแม่นยำโดยมีความผิดพลาดไม่เกิน $\pm 1.0\%$



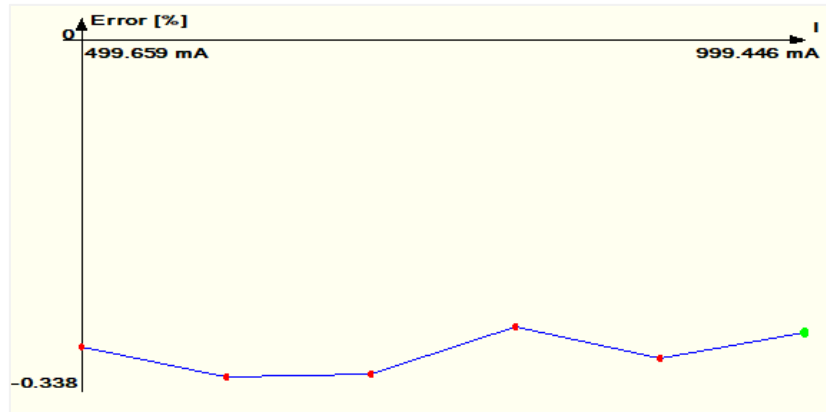
รูปที่ 56 ผลการทดสอบที่ค่ากระแสไฟฟ้า $0.1I_n \leq I \leq I_{max}$ ตัวประกอบกำลังเท่ากับ 1

ผลการทดสอบระยะที่ 3 เป็นการทดสอบที่ค่ากระแสไฟฟ้า $0.1I_n \leq I \leq 0.2I_n$ ที่ค่าตัวประกอบกำลังเท่ากับ 0.5 ล้าหลัง โดยจะมีค่าความผิดพลาดสูงสุดเท่ากับ **0.632%** ซึ่งผ่านข้อกำหนดตามหลักเกณฑ์การทดสอบความแม่นยำโดยมีความผิดพลาดไม่เกิน $\pm 1.5\%$



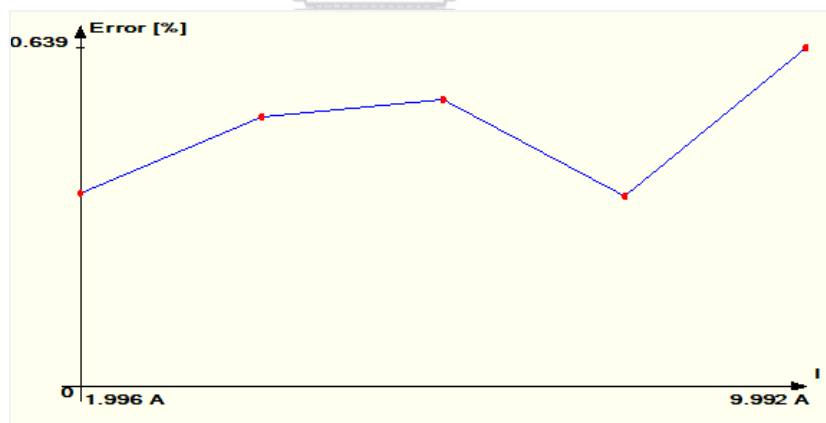
รูปที่ 57 ผลการทดสอบที่ค่ากระแสไฟฟ้า $0.1I_n \leq I \leq 0.2I_n$ ตัวประกอบกำลังเท่ากับ 0.5 ล้าหลัง

ผลการทดสอบระยะที่ 4 เป็นการทดสอบที่ค่ากระแสไฟฟ้า $0.1I_n \leq I \leq 0.2I_n$ ที่ค่าตัวประกอบกำลังเท่ากับ 0.8 นำหน้า โดยจะมีค่าความผิดพลาดสูงสุดเท่ากับ **-0.338%** ซึ่งผ่านข้อกำหนดตามหลักเกณฑ์การทดสอบความแม่นยำโดยมีความผิดพลาดไม่เกิน $\pm 1.5\%$



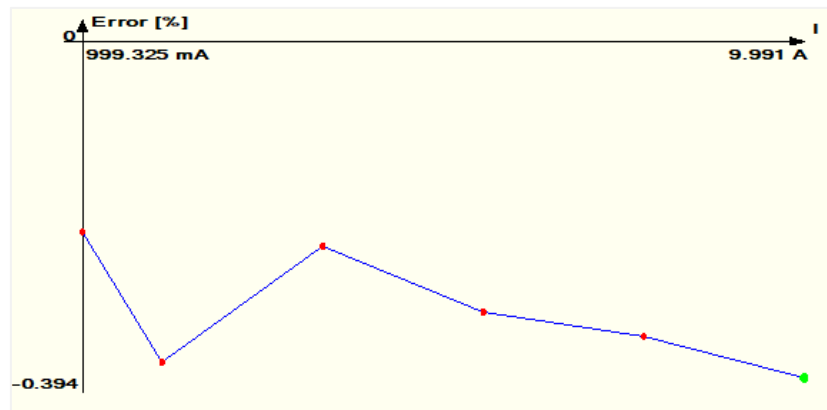
รูปที่ 58 ผลการทดสอบที่ค่ากระแสไฟฟ้า $0.1I_n \leq I \leq 0.2I_n$ ตัวประกอบกำลังเท่ากับ 0.8 นำหน้า

ผลการทดสอบระยะที่ 5 เป็นการทดสอบที่ค่ากระแสไฟฟ้า $0.2I_n \leq I \leq I_{max}$ ที่ค่าตัวประกอบกำลังเท่ากับ 0.5 ล้าหลัง โดยจะมีค่าความผิดพลาดสูงสุดเท่ากับ **0.639%** ซึ่งผ่านข้อกำหนดตามหลักเกณฑ์การทดสอบความแม่นยำโดยมีความผิดพลาดไม่เกิน $\pm 1.0\%$



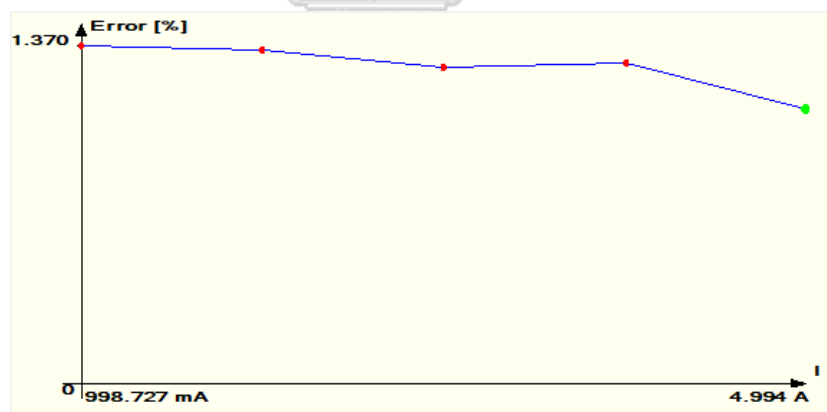
รูปที่ 59 ผลการทดสอบที่ค่ากระแสไฟฟ้า $0.2I_n \leq I \leq I_{max}$ ตัวประกอบกำลังเท่ากับ 0.5 ล้าหลัง

ผลการทดสอบระยะที่ 6 เป็นการทดสอบที่ค่ากระแสไฟฟ้า $0.2I_n \leq I \leq I_{max}$ ที่ค่าตัวประกอบกำลังเท่ากับ 0.8 นำหน้า โดยจะมีค่าความผิดพลาดสูงสุดเท่ากับ **-0.394%** ซึ่งผ่านข้อกำหนดตามหลักเกณฑ์การทดสอบความแม่นยำโดยมีความผิดพลาดไม่เกิน $\pm 1.0\%$



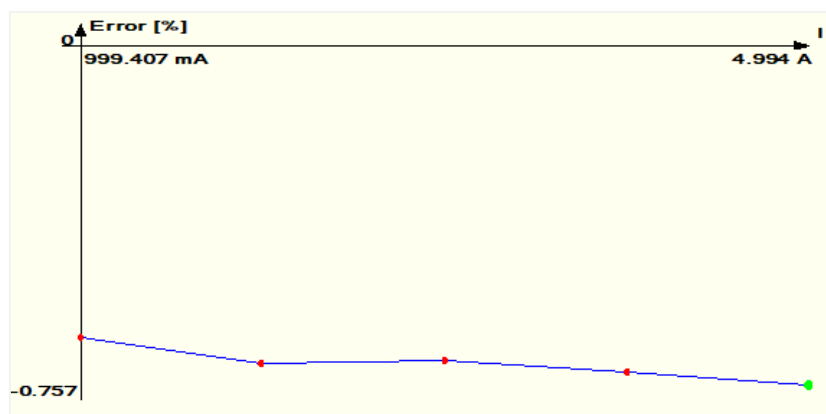
รูปที่ 60 ผลการทดสอบที่ค่ากระแสไฟฟ้า $0.2I_n \leq I \leq I_{max}$ ตัวประกอบกำลังเท่ากับ 0.8 นำหน้า

ผลการทดสอบระยะที่ 7 เป็นการทดสอบที่ค่ากระแสไฟฟ้า $0.2I_n \leq I \leq I_n$ ที่ค่าตัวประกอบกำลังเท่ากับ 0.25 ล้าหลัง โดยจะมีค่าความผิดพลาดสูงสุดเท่ากับ **1.370%** ซึ่งผ่านข้อกำหนดตามหลักเกณฑ์การทดสอบความแม่นยำโดยมีความผิดพลาดไม่เกิน $\pm 3.5\%$



รูปที่ 61 ผลการทดสอบที่ค่ากระแสไฟฟ้า $0.2I_n \leq I \leq I_n$ ตัวประกอบกำลังเท่ากับ 0.25 ล้าหลัง

ผลการทดสอบระยะที่ 8 เป็นการทดสอบที่ค่ากระแสไฟฟ้า $0.2I_n \leq I \leq I_n$ ที่ค่าตัวประกอบกำลังเท่ากับ 0.5 นำหน้า โดยจะมีค่าความผิดพลาดสูงสุดเท่ากับ **-0.757%** ซึ่งผ่านข้อกำหนดตามหลักเกณฑ์การทดสอบความแม่นยำโดยมีความผิดพลาดไม่เกิน $\pm 2.5\%$



รูปที่ 62 ผลการทดสอบที่ค่ากระแสไฟฟ้า $0.2I_n \leq I \leq I_n$ ตัวประกอบกำลังเท่ากับ 0.5 นำหน้า

5.3 ทดสอบการส่งข้อมูลพลังงานทางไฟฟ้าไปยัง NETPIE Dashboard

ข้อมูลค่าพลังงานทางไฟฟ้าจะถูกส่งจากมิเตอร์อัจฉริยะผ่านอินเทอร์เน็ตด้วยการเชื่อมต่อสัญญาณอินเทอร์เน็ตแบบไร้สาย (Wi-Fi) โดยส่งข้อมูลออกไปในรูปแบบ JSON เพื่อให้ NETPIE สามารถนำไปใช้งานในการแสดงผลบนแผงหน้าปัด (Dashboard) โดยจะสามารถแสดงค่าพลังงานทางไฟฟ้าต่าง ๆ ได้ดังตัวอย่างในรูปที่ 63



รูปที่ 63 การทดสอบส่งข้อมูลค่าพลังงานทางไฟฟ้าไปยัง NETPIE Dashboard เพื่อแสดงผล

5.4 ทดสอบการสอบเทียบตัวเองของมิเตอร์อัจฉริยะต้นแบบ

เมื่อมีการเรียกใช้งานภารกิจการสอบเทียบตัวเองของมิเตอร์ ภารกิจจะดำเนินการสอบเทียบครั้งละหนึ่งช่องสัญญาณโดยเริ่มจากช่องสัญญาณกระแส และตามด้วยช่องสัญญาณแรงดันในเวลาต่อมา หลังจากที่มิเตอร์ทำการสอบเทียบเสร็จสิ้น มิเตอร์จะแสดงผลค่าคงตัวการแปลงผัน (CC) และค่าความมั่นใจของการประมาณค่า (CERT) ที่คำนวณได้จากการสอบเทียบครั้งนี้ ซึ่งจะสามารถแสดงตัวอย่างการสอบเทียบตัวเองให้เห็นได้ดังรูปที่ 64

```

Autocalibrating Current Channel
Autocalibrating Voltage Channel
AICC: 834.44
AICERT: 2371
AVCC: 13347.51
AVCERT: 2474
Autocalibration Complete
RMS Current: 0.00 A
RMS Voltage: 226.42 V
Active Power: 0.00 W
Reactive Power: 0.00 VAR
Apparent Power: 0.12 VA
Power Factor: 0.05
Frequency: 50.02 Hz
Temperature: 31.03 degC

```

รูปที่ 64 การทดสอบการสอบเทียบตัวเองของมิเตอร์อัจฉริยะต้นแบบที่ถูกพัฒนาขึ้น

5.5 ทดสอบการจัดเก็บข้อมูลลงฐานข้อมูลด้วย SQLite

เมื่อมีการเปิดใช้งานมิเตอร์ไฟฟ้าอัจฉริยะ ค่าพลังงานต่าง ๆ ที่วัดได้จะถูกจัดเก็บด้วยซอฟต์แวร์ขนาดเล็กที่มีชื่อว่า SQLite โดยค่าจะถูกบันทึกลงในการ์ดหน่วยความจำ (SD Card) ของมิเตอร์ ด้วยภาษาโปรแกรมที่ใช้ในการจัดเก็บข้อมูลอย่าง SQL ทำให้ข้อมูลที่ถูกรวบรวมมีความเป็นระเบียบ และสามารถแยกประเภทข้อมูลที่จัดเก็บได้

ประเภทข้อมูลที่ถูกรวบรวมลงในฐานข้อมูลของมิเตอร์ต้นแบบนี้มี 2 ประเภทด้วยกันได้แก่

- ประเภทข้อความ (Text) ใช้สำหรับการจัดเก็บข้อมูลตราเวลา (Timestamp)
- ประเภทตัวเลขทศนิยม (Float) ใช้สำหรับการจัดเก็บค่าพลังงานที่วัดได้ต่าง ๆ

ข้อมูลที่ถูกจัดเก็บลงในฐานข้อมูลสามารถแสดงให้เห็นได้ในรูปที่ 65

```
SELECT * FROM ADE_datalog WHERE rowid BETWEEN 76 AND 92;
```

Output:

Timestamp	IRMS	VRMS	ActPWR	ReaPWR	AppPWR	PF	Freq	Temp
14/08/2021-19:40:13	0.0	0.01	0.0	0.0	0	1.0	50.0	31.6
14/08/2021-19:45:30	0.0	20.89	0.0	0.0	0.09	0.01	50.02	30.46
14/08/2021-19:46:00	0.0	20.89	0.0	0.0	0.01	-0.21	49.99	31.32
14/08/2021-19:46:30	0.0	20.96	0.0	0.0	0.01	-0.21	50.05	31.03
14/08/2021-19:47:00	0.0	20.97	0.0	0.0	0.01	-0.21	50.06	31.32
14/08/2021-19:47:30	0.0	20.96	0.0	0.0	0.01	-0.21	50.05	30.46
14/08/2021-19:48:00	0.0	20.95	0.0	0.0	0.01	0.0	50.04	31.6
14/08/2021-19:48:30	0.41	223.5	93.17	0.02	91.93	1.0	50.0	30.75
14/08/2021-19:49:00	0.42	223.46	54.01	-77.97	94.85	0.57	50.0	31.32
14/08/2021-19:49:30	0.42	223.39	53.93	-78.0	94.83	0.57	50.0	31.03
14/08/2021-19:50:00	0.42	223.44	53.95	-78.01	94.85	0.57	50.0	31.6
14/08/2021-19:50:30	0.42	223.38	53.93	-77.99	94.83	0.57	50.0	31.6
14/08/2021-19:51:00	0.42	223.38	53.92	-78.0	94.82	0.57	50.0	31.6
14/08/2021-19:51:30	0.42	223.42	53.93	-78.01	94.84	0.57	50.0	31.03
14/08/2021-19:52:00	0.42	223.43	53.94	-78.01	94.85	0.57	50.0	32.17
14/08/2021-19:52:30	0.42	223.43	53.95	-78.0	94.85	0.57	50.0	31.32
14/08/2021-19:53:00	0.42	223.37	53.92	-77.99	94.82	0.57	50.0	31.89

รูปที่ 65 ตัวอย่างการจัดเก็บข้อมูลลงในฐานข้อมูลด้วย SQLite

5.6 ทดสอบซอฟต์แวร์การตอบสนองด้านโหนด OpenADR 2.0

การทดสอบซอฟต์แวร์ OpenADR 2.0 นี้จะแบ่งออกเป็น 2 ส่วนหลัก ๆ ได้แก่การทดสอบ โหนดต้นเสมือน และการทดสอบโหนดปลายเสมือน โดยโหนดต้นเสมือนนั้นจะเริ่มทดสอบจากการ ลงทะเบียนโหนดปลายเสมือนเข้าระบบ และทดสอบการสร้างเหตุการณ์การตอบสนองด้านโหนด ส่วนโหนดปลายเสมือนจะเริ่มจากการทดสอบการเชื่อมต่อสื่อสารระหว่างโหนดต้นเสมือนและโหนด ปลายเสมือน ไปจนถึงลักษณะการตอบสนองด้านโหนดด้วยโปรแกรมที่ผู้วิจัยได้พัฒนาขึ้น

5.6.1 ทดสอบการลงทะเบียนโหนดปลายเสมือนเข้าระบบ

เมื่อสามารถแปลโปรแกรม (Compile) ได้สำเร็จแล้ว จะสามารถเปิดเว็บเบราว์เซอร์เพื่อเข้าสู่ ระบบในการกำหนดค่าต่าง ๆ ได้ โดยใส่ชื่อที่อยู่ไอพีของโหนดต้นเสมือนดังรูปที่ 66 จากนั้นเข้าสู่ ระบบด้วยไอดีและรหัสผ่าน โดยค่าเริ่มต้นหลังจากแปลโปรแกรมแล้วเสร็จคือ *admin* และ *testing* ตามลำดับ

EPRI | ELECTRIC POWER RESEARCH INSTITUTE | Auto DR Test Server | Log in (not signed in) | About

Please Login

User Name

Password

I agree to the terms of the EPRI software agreement.

Login

Copyright © 2021 EPRI | v0.9.7

รูปที่ 66 หน้าต่างเข้าสู่ระบบโหนดต้นเสมือน

เมื่อเข้าสู่ระบบแล้ว ให้เลือกแถบ “VENs” เพื่อลงทะเบียนโหนดปลายเสมือนเข้าสู่ระบบ จากนั้นเลือก “Create Ven” เพื่อตั้งค่าดังรูปที่ 67

EPRI | ELECTRIC POWER RESEARCH INSTITUTE | Auto DR Test Server | Log out (admin) | About

Events | **VENs** | Accounts | Status & Export | VTN Config | TOU Schedules | Download VEN

VENs

Search | Create Ven

Name: VEN_RPI2 VEN ID: aca514633830bbb88807	Data Edit offline
Name: VEN_RPI VEN ID: a3e27048d98646a75614	Data Edit offline
Name: testven VEN ID: oadrtest1	Data Edit offline

รูปที่ 67 สร้างโหนดปลายเสมือนเพื่อลงทะเบียนเข้าสู่ระบบ

จากนั้นใส่ชื่อโหนดปลายเสมือนที่ต้องการลงทะเบียนเข้าระบบดังรูปที่ 68 จากนั้นให้เลือก “Create VEN” อีกครั้งเพื่อทำการสร้างโพรไฟล์โหนดปลายเสมือน

The screenshot shows the 'Auto DR Test Server' interface. On the left is a navigation menu with options: Events, VENS, Accounts, Status & Export, VTN Config, TOU Schedules, and Download VEN. The main content area is titled 'VENs' and contains an 'Identification' section with two input fields: 'VEN Name' containing 'ESID-VEN' and 'Common Name' containing 'esid-lab-ven'. Below these fields is a prominent green 'Create VEN' button and a 'Back' button. The footer indicates 'Copyright © 2021 EPRI' and 'v0.9.7'.

รูปที่ 68 ตั้งชื่อโหนดปลายเสมือนเพื่อลงทะเบียนเข้าสู่ระบบ

หลังจากที่ลงทะเบียนสำเร็จแล้วให้เลื่อนลงมาตั้งค่ารูปแบบโพรไฟล์ OpenADR Profile ให้เป็น “2.0b” และกด Save ดังรูปที่ 69 เพื่อบันทึกค่าโหนดปลายเสมือน

The screenshot shows the 'Auto DR Test Server' interface with the 'ESID-VEN' tab selected. The 'OpenADR Profile Settings' section is visible, featuring a dropdown menu for 'OpenADR Profile' set to '2.0b', a checkbox for 'Http Push?' which is unchecked, and an empty 'Transport Address' input field. A 'Save' button is located at the bottom of this section. The navigation menu on the left remains the same as in the previous screenshot.

รูปที่ 69 ตั้งค่ารูปแบบโพรไฟล์ OpenADR 2.0b Specification สำหรับโหนดปลายเสมือน

เมื่อกลับมาที่หน้าหลักของโหนดปลายเสมือน จะเห็นรายชื่อโหนดปลายเสมือนและสถานะของโหนดปลายเสมือนที่เราได้ทำการลงทะเบียนเรียบร้อยแล้วดังรูปที่ 70 เป็นอันเสร็จสิ้นขั้นตอนการตั้งค่าเบื้องต้นสำหรับการลงทะเบียนโหนดปลายเสมือนเข้าสู่ระบบ

Name	VEN ID	Status
ESID-VEN	cc312a0bf5d25600b96d	online
VEN_RPI	a3e27048d98646a75614	offline
testven	oadrtest1	offline

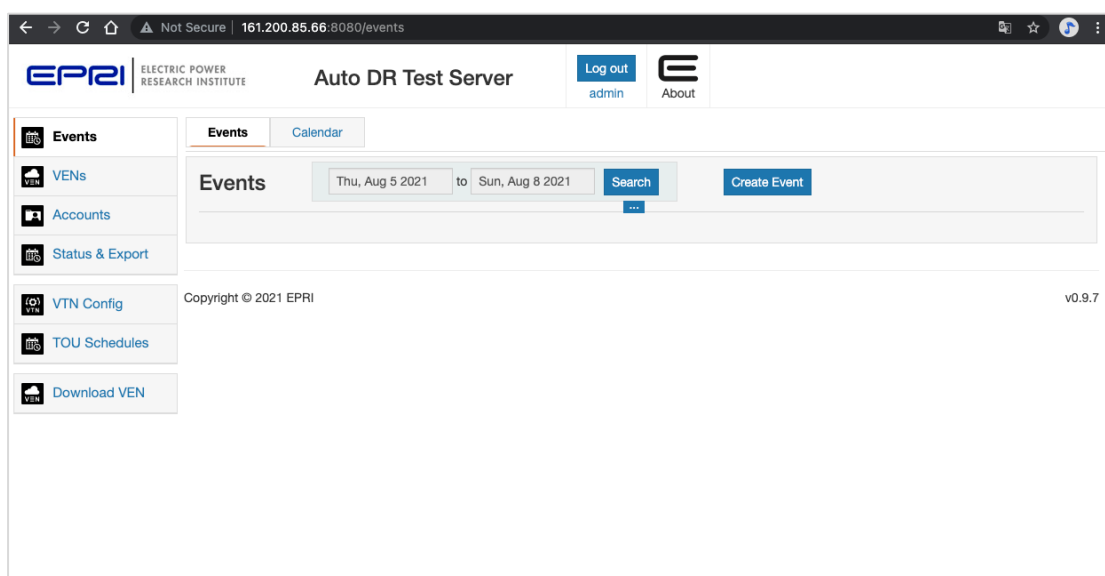
รูปที่ 70 รายชื่อและสถานะของโหนดปลายเสมือนที่ได้ลงทะเบียนเข้าสู่ระบบเรียบร้อยแล้ว



5.6.2 ทดสอบการสร้างเหตุการณ์ด้วยโหนดต้นเสมือน

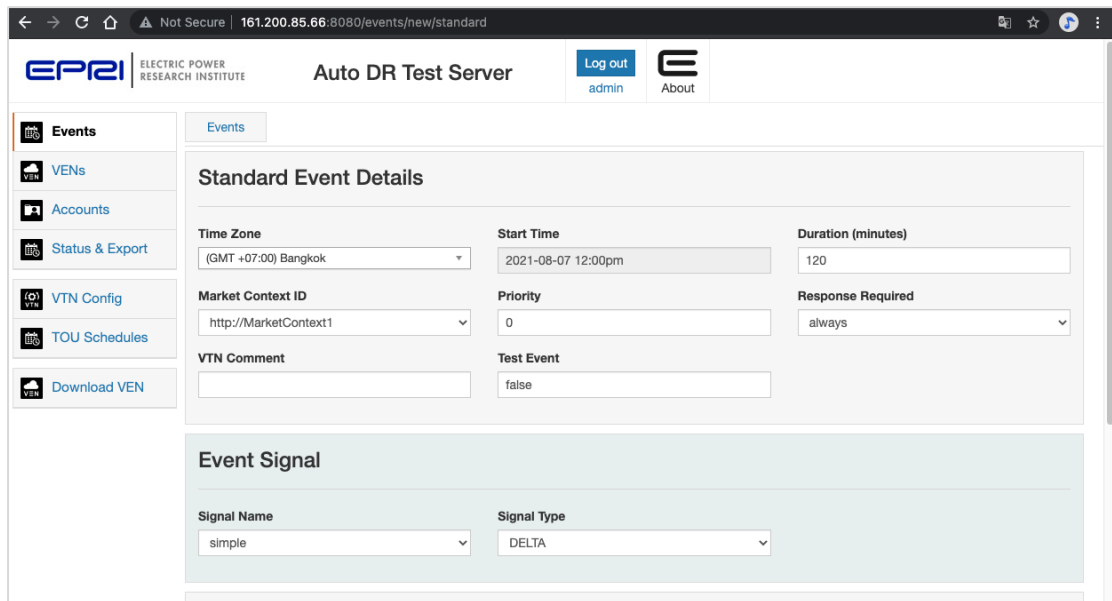
เมื่อเราได้ทำการลงทะเบียนโหนดปลายเสมือนเสร็จสิ้นแล้ว เราสามารถสร้างเหตุการณ์ (Event) สำหรับการตอบสนองด้านโหลดอัตโนมัติ เพื่อรองรับโหนดปลายเสมือนที่จะเชื่อมต่อเข้ามา รวมมาตรการการตอบสนองด้านโหลด ด้วยวิธีการดังนี้

จากรูปที่ 71 เมื่อเข้าสู่ระบบแล้ว ให้เลือกแถบ “Events” จากนั้นเลือก “Create Event” เพื่อสร้างเหตุการณ์สำหรับการตอบสนองด้านโหลด



รูปที่ 71 สร้างเหตุการณ์สำหรับการตอบสนองด้านโหลด

จากนั้นให้กรอกข้อมูลที่ต้องการจะสร้างเหตุการณ์ อาทิ ค่าเขตเวลา (Time zone) วันเวลาเริ่มดำเนินเหตุการณ์ (Start time) ระยะเวลาดำเนินการตอบสนองด้านโหลดของเหตุการณ์ (Duration) รวมถึงรูปแบบสัญญาณการตอบสนองด้านโหลด เป็นต้น ซึ่งแสดงให้เห็นในรูปที่ 72 เมื่อกรอกข้อมูลที่ต้องการครบถ้วนแล้ว สามารถเลือก “Create Event” เพื่อสร้างเหตุการณ์ได้



Standard Event Details

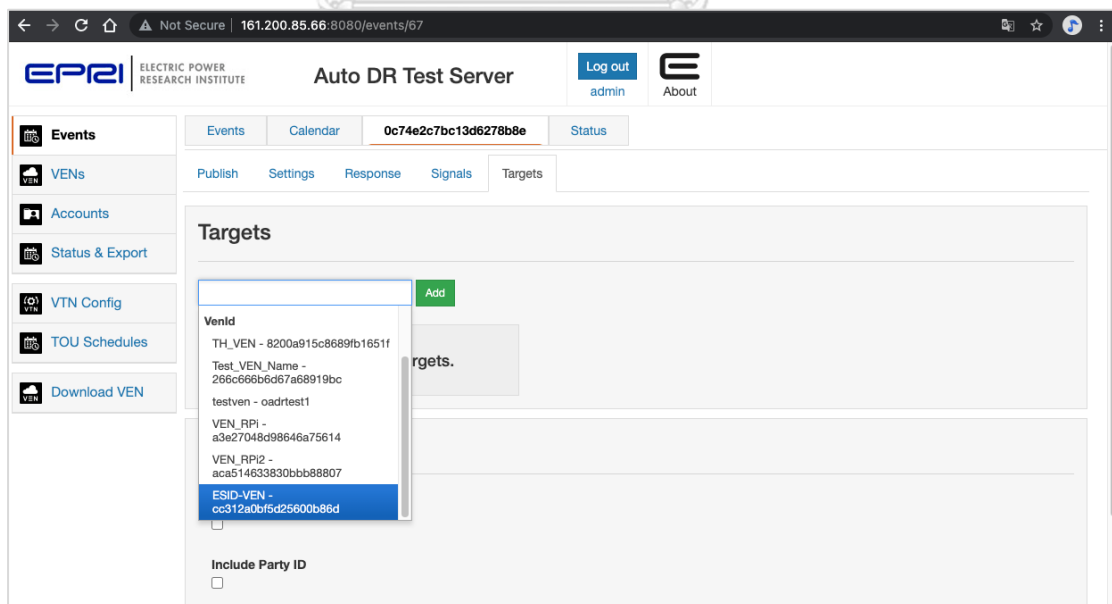
Time Zone (GMT +07:00) Bangkok	Start Time 2021-08-07 12:00pm	Duration (minutes) 120
Market Context ID http://MarketContext1	Priority 0	Response Required always
VTN Comment	Test Event false	

Event Signal

Signal Name simple	Signal Type DELTA
-----------------------	----------------------

รูปที่ 72 กรอกรายละเอียดเพื่อสร้างเหตุการณ์สำหรับการตอบสนองด้านโหลด

หลังจากที่สร้างเหตุการณ์เรียบร้อยแล้ว ให้เลือกแถบ “Targets” ที่อยู่ด้านบนเพื่อทำการเลือกเป้าหมายของเหตุการณ์ในครั้งนี โดยเลือกจากชื่อโหนดปลายเสมือนที่ได้ทำการลงทะเบียนเรียบร้อยแล้วก่อนหน้านี้ดังแสดงให้เห็นในรูปที่ 73 จากนั้นเลือก “Add” เพื่อเพิ่มรายชื่อโหนดและ “Save” เพื่อบันทึกการเปลี่ยนแปลงตามลำดับ



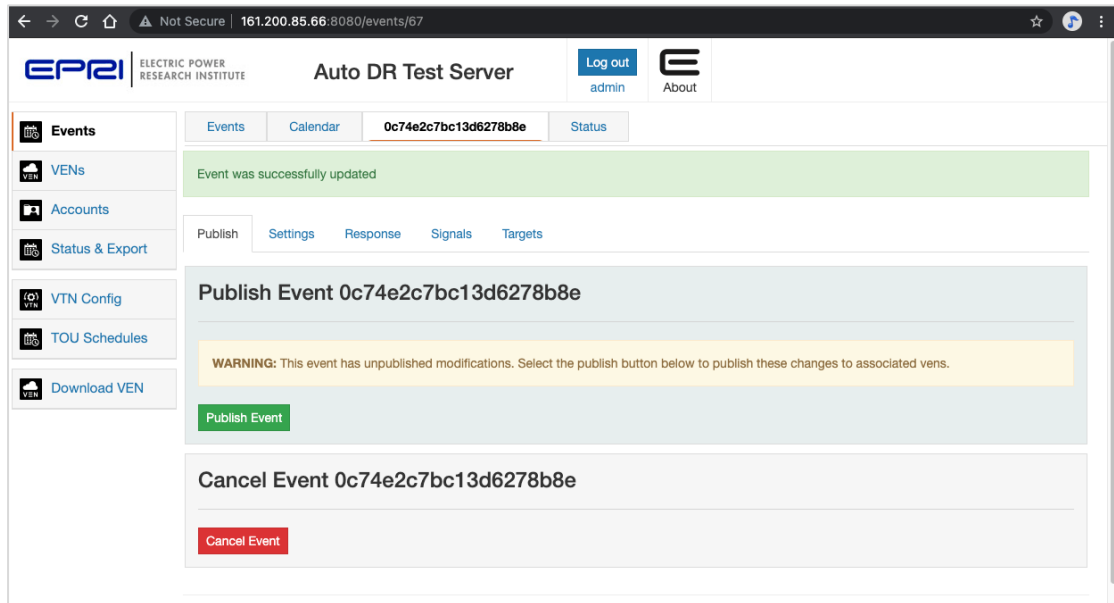
Targets

VENid	TH_VEN - 8200a915c8689fb1651f
Test_VEN_Name	266c666b6d67a68919bc
testven - oadrtest1	
VEN_RPI -	a3e27048d98646a75614
VEN_RPI2 -	aca514633830bbb88807
ESID-VEN -	cc312a0bf5d25600b88d

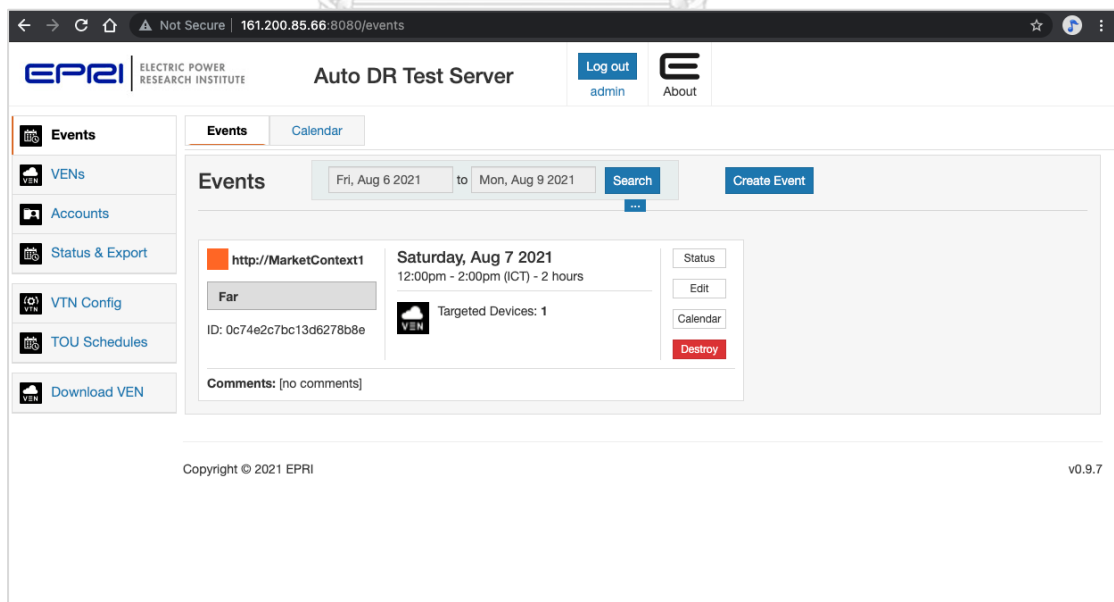
Include Party ID

รูปที่ 73 เลือกเป้าหมายของเหตุการณ์สำหรับการตอบสนองด้านโหลด

เมื่อเสร็จสิ้นกระบวนการ เหตุการณ์ที่ได้รับการตั้งค่าก็พร้อมที่จะเผยแพร่สู่ลูกค้าโหมดปลายเสมือนที่ได้เชื่อมต่อเข้ามายังโหมดต้นเสมือนนี้ โดยสามารถเลือกที่ “Publish Event” ดังรูปที่ 74 เป็นอันเสร็จสิ้นกระบวนการการสร้างเหตุการณ์การตอบสนองด้านโหลด นอกจากนี้ยังสามารถติดตามสถานะหรือแก้ไขเหตุการณ์นั้น ๆ ได้ทุกเมื่อที่หน้าหลักดังแสดงในรูปที่ 75



รูปที่ 74 เผยแพร่เหตุการณ์ไปยังโหมดปลายเสมือน



รูปที่ 75 แสดงสถานะเบื้องต้นของเหตุการณ์

5.6.3 ทดสอบการโต้ตอบระหว่างโหนดต้นเสมือนและโหนดปลายเสมือน

ในเบื้องต้น หากซอฟต์แวร์โหนดปลายเสมือนสามารถสร้าง (Build) ออกมาได้สำเร็จโดยไม่มีข้อผิดพลาดใด ๆ ซอฟต์แวร์ที่สร้างแล้วเสร็จนั้นสามารถทดสอบได้ด้วยโปรแกรมที่ถูกสร้างขึ้นมาในชื่อ `testoadr` ซึ่งเป็นไฟล์ทดสอบการสื่อสารโต้ตอบโดยการตรวจสอบเช็คความพร้อมของโปรแกรม รวมถึงตรวจสอบไฟล์ต่าง ๆ ที่มีส่วนเกี่ยวข้องกับการสื่อสาร ซึ่งเราสามารถเรียกใช้งานได้ (Executable file) โดยราสเบอร์รี่พายที่ใช้ระบบปฏิบัติการลินุกซ์นั้นสามารถเรียกใช้โปรแกรมได้โดย `./testoadr` ซึ่งผลที่ได้จะแสดงให้เห็นดังรูปที่ 76

```
[ RUN ] Scheduler.ExecuteTwoOnetimeJobs
[ OK ] Scheduler.ExecuteTwoOnetimeJobs (0 ms)
[ RUN ] Scheduler.ScheduleJobsOutOfOrder
[ OK ] Scheduler.ScheduleJobsOutOfOrder (0 ms)
[ RUN ] Scheduler.SchedulerShouldRejectJobsWithRuntimeInPast
[ OK ] Scheduler.SchedulerShouldRejectJobsWithRuntimeInPast (0 ms)
[ RUN ] Scheduler.ScheduleSinglePeriodicJob
[ OK ] Scheduler.ScheduleSinglePeriodicJob (0 ms)
[ RUN ] Scheduler.ExecuteSlowJob
[ OK ] Scheduler.ExecuteSlowJob (0 ms)
[ RUN ] Scheduler.ExecuteExceptionJob
[ OK ] Scheduler.ExecuteExceptionJob (0 ms)
[ RUN ] Scheduler.JobWPastStartTime
[ OK ] Scheduler.JobWPastStartTime (0 ms)
[-----] 11 tests from Scheduler (0 ms total)

[-----] 2 tests from VEN2bTest
[ RUN ] VEN2bTest.Registration
[ OK ] VEN2bTest.Registration (1 ms)
[ RUN ] VEN2bTest.RegistrationFail
[ OK ] VEN2bTest.RegistrationFail (1 ms)
[-----] 2 tests from VEN2bTest (2 ms total)

[-----] Global test environment tear-down
[=====] 53 tests from 14 test cases ran. (26 ms total)
[ PASSED ] 53 tests.

YOU HAVE 1 DISABLED TEST
```

รูปที่ 76 ผลลัพธ์ที่คาดหวังจากโปรแกรมทดสอบการโต้ตอบ `testoadr`

นอกจากโปรแกรมทดสอบ `testoadr` แล้ว ยังมีอีกโปรแกรมทดสอบที่ซอฟต์แวร์ได้สร้างขึ้นมานั้นคือ `samplevenmgr` โปรแกรมนี้ถือเป็นโปรแกรมทดสอบการโต้ตอบจริงกับโหนดต้นเสมือนที่ไม่ได้มีการปรับแต่งใด ๆ สามารถเรียกใช้โปรแกรมได้ด้วยคำสั่ง `./samplevenmgr` ตามด้วยพารามิเตอร์ที่สำคัญอย่างที่อยู่ของโหนดต้นเสมือน และชื่อของโหนดปลายเสมือนที่ได้ลงทะเบียนไว้แล้ว โดยตัวอย่างคำสั่งเรียกใช้โปรแกรมคือ

```
~$ ./samplevenmgr http://161.200.85.66:8080/ ESID-VEN
```

โดยที่ *samplevenmgr* คือ โปรแกรมที่เรียกใช้งานการทดสอบโต้ตอบกับโหนดต้นเสมือน
<http://161.200.865.66:8080/> คือ ที่อยู่ของโหนดต้นเสมือนที่ผู้วิจัยสร้างขึ้น
ESID-VEN คือ ชื่อของโหนดปลายเสมือนที่ได้ทำการลงทะเบียนไว้แล้ว

เมื่อโปรแกรมเริ่มทำงาน โหนดปลายเสมือนจะทำการลงทะเบียนกับระบบของโหนดต้นเสมือนผ่านชุดข้อมูล *oadrCreatePartyRegistration* ที่ประกอบไปด้วยข้อมูลเบื้องต้นต่าง ๆ ของโหนดปลายเสมือน ดังแสดงในรูปที่ 77

```
honest@honest-esid-ubuntu:~/Downloads/buildVEN/OpenADR-VEN-Library-0.7.0/build/debug$ ./samplevenmgr http://161.200.85.66:8080/ ESID-VEN
2021-08-07 23:09:36,559 INFO [default]
VEN ----> VTN
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<p1:oadrPayload xmlns:p1="http://openadr.org/oadr-2.0b/2012/07">
  <p1:oadrSignedObject>
    <p1:oadrCreatePartyRegistration xmlns:p3="http://docs.oasis-open.org/ns/energyinterop/201110" p3:schemaVersion="2.0b" xmlns:p2="http://docs.oasis-open.org/ns/energyinterop/201110/payloads">
      <p2:requestID>98AD6CD557</p2:requestID>
      <p1:oadrProfileName>2.0b</p1:oadrProfileName>
      <p1:oadrTransportName>simpleHttp</p1:oadrTransportName>
      <p1:oadrReportOnly>false</p1:oadrReportOnly>
      <p1:oadrXmlSignature>false</p1:oadrXmlSignature>
      <p1:oadrVenName>ESID-VEN</p1:oadrVenName>
      <p1:oadrHttpPullModel>true</p1:oadrHttpPullModel>
    </p1:oadrCreatePartyRegistration>
  </p1:oadrSignedObject>
</p1:oadrPayload>
=====
```

รูปที่ 77 แสดงการโต้ตอบของชุดข้อมูล *oadrCreatePartyRegistration*

หลังจากที่โหนดต้นเสมือนได้รับข้อมูลการลงทะเบียนแล้ว โหนดต้นเสมือนจะทำการตอบกลับไปด้วยชุดข้อมูล *oadrCreatedPartyRegistration* ซึ่งประกอบไปด้วยข้อมูลที่สำคัญต่าง ๆ เช่น VTN info, registrationID รวมถึง venID ดังแสดงในรูปที่ 78

```
2021-08-07 23:09:36,574 INFO [default]
VEN <---- VTN
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:oadrPayload xmlns="http://docs.oasis-open.org/ns/emix/2011/06/power" xmlns:ns2="http://docs.oasis-open.org/ns/emix/2011/06/siscale"
  xmlns:ns3="http://openadr.org/oadr-2.0b/2012/07" xmlns:ns4="http://www.w3.org/2005/Atom" xmlns:ns5="http://www.opengis.net/gml/3.2" xmlns:ns6="http://docs.oasis-open.org/ns/emix/2011/06" xmlns:ns7="http://www.w3.org/2009/xmldsig11#" xmlns:ns8="http://www.w3.org/2000/09/xmldsig#"
  xmlns:ns9="http://openadr.org/oadr-2.0b/2012/07/xmldsig-properties" xmlns:ns10="urn:ietf:params:xml:ns:icalendar-2.0" xmlns:ns11="http://docs.oasis-open.org/ns/energyinterop/201110"
  xmlns:ns12="http://docs.oasis-open.org/ns/energyinterop/201110/payloads" xmlns:ns13="urn:ietf:params:xml:ns:icalendar-2.0:stream" xmlns:ns14="urn:un:unece:uncefact:codelist:standard:5:ISO42173A:2010-04-07">
  <ns3:oadrSignedObject>
    <ns3:oadrCreatedPartyRegistration ns11:schemaVersion="2.0b">
      <ns11:eiResponse>
        <ns11:responseCode>200</ns11:responseCode>
        <ns11:responseDescription>OK</ns11:responseDescription>
        <ns12:requestID>98AD6CD557</ns12:requestID>
      </ns11:eiResponse>
      <ns11:registrationID>ab1608947262f178ae6a</ns11:registrationID>
      <ns11:venID>cc312a0bf5d25600b86d</ns11:venID>
      <ns11:vtnID>EPRI_VTN</ns11:vtnID>
    </ns3:oadrCreatedPartyRegistration>
  </ns3:oadrSignedObject>
  <ns3:oadrProfiles>
    <ns3:oadrProfile>
      <ns3:oadrProfileName>2.0a</ns3:oadrProfileName>
    </ns3:oadrProfile>
  </ns3:oadrProfiles>
</ns3:oadrPayload>
```

รูปที่ 78 แสดงการโต้ตอบของชุดข้อมูล *oadrCreatedPartyRegistration*

จากนั้นโหนดปลายเสมือนจะส่งคำร้องขอลงทะเบียนบริการรายงานไปยังโหนดต้นเสมือนด้วย oadrRegisterReport เพื่อบอกรายละเอียดความสามารถในการรายงาน (Report capabilities) ดังแสดงในรูปที่ 79

```

2021-08-07 23:09:36,575 INFO [default] register report
2021-08-07 23:09:36,576 INFO [default]
VEN ---> VTN
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<p1:oadrPayload xmlns:p1="http://openadr.org/oadr-2.0b/2012/07">
  <p1:oadrSignedObject>
    <p1:oadrRegisterReport xmlns:p3="http://docs.oasis-open.org/ns/energyinterop/201110" p3:schemaVersion="2.0b" xmlns:p2="http://docs.oasis-open.org/ns/energyinterop/201110/payloads">
      <p2:requestID>98AD6CD557</p2:requestID>
      <p1:oadrReport xmlns:p3="urn:ietf:params:xml:ns:icalendar-2.0" xmlns:p4="http://docs.oasis-open.org/ns/energyinterop/201110">
        <p3:duration>
          <p3:duration>PT120M</p3:duration>
        </p3:duration>
        <p1:oadrReportDescription xmlns:p4="http://docs.oasis-open.org/ns/energyinterop/201110" xmlns:p5="http://docs.oasis-open.org/ns/emix/2011/06/power" xmlns:p6="http://docs.oasis-open.org/ns/emix/2011/06">
          <p4:rID>rid_energy_4184bb93</p4:rID>
          <p4:reportDataSource>
            <p4:resourceID>DEVICE1</p4:resourceID>
          </p4:reportDataSource>
          <p4:reportType>reading</p4:reportType>
          <p5:energyReal xmlns:p6="http://docs.oasis-open.org/ns/emix/2011/06/siscale">
            <p5:itemDescription/>
  </p1:oadrReport>
    </p1:oadrRegisterReport>
  </p1:oadrSignedObject>
</p1:oadrPayload>

```

รูปที่ 79 แสดงการโต้ตอบของชุดข้อมูล oadrRegisterReport

จากนั้นโหนดต้นเสมือนจะตอบกลับด้วยชุดข้อมูล oadrRegisteredReport เพื่อรับทราบการลงทะเบียนบริการรายงาน ดังแสดงในรูปที่ 80

```

2021-08-07 23:09:36,648 INFO [default]
VEN <--- VTN
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:oadrPayload xmlns="http://docs.oasis-open.org/ns/emix/2011/06/power" xmlns:ns2="http://docs.oasis-open.org/ns/emix/2011/06/siscale"
  xmlns:ns3="http://openadr.org/oadr-2.0b/2012/07" xmlns:ns4="http://www.w3.org/2005/Atom" xmlns:ns5="http://www.opengis.net/gml/3.2" xmlns:ns6="http://docs.oasis-open.org/ns/emix/2011/06" xmlns:ns7="http://www.w3.org/2009/xmldsig11#" xmlns:ns8="http://www.w3.org/2000/09/xmldsig#"
  xmlns:ns9="http://openadr.org/oadr-2.0b/2012/07/xmldsig-properties" xmlns:ns10="urn:ietf:params:xml:ns:icalendar-2.0" xmlns:ns11="http://docs.oasis-open.org/ns/energyinterop/201110"
  xmlns:ns12="http://docs.oasis-open.org/ns/energyinterop/201110/payloads" xmlns:ns13="urn:ietf:params:xml:ns:icalendar-2.0:stream" xmlns:ns14="urn:un:unece:uncefact:odelist:standard:5:ISO42173A:2010-04-07">
  <ns3:oadrSignedObject>
    <ns3:oadrRegisteredReport ns11:schemaVersion="2.0b">
      <ns11:eiResponse>
        <ns11:responseCode>200</ns11:responseCode>
        <ns11:responseDescription>OK</ns11:responseDescription>
        <ns12:requestID>98AD6CD557</ns12:requestID>
      </ns11:eiResponse>
    </ns3:oadrRegisteredReport>
  </ns3:oadrSignedObject>
</ns3:oadrPayload>
=====

```

รูปที่ 80 แสดงการโต้ตอบของชุดข้อมูล oadrRegisteredReport

หลังจากลงทะเบียนบริการรายงานเสร็จ โหนดปลายเสมือนจะส่งชุดข้อมูล oadrRequestEvent พร้อมแนบ venID ไปยังโหนดต้นเสมือนเพื่อสอบถามถึงเหตุการณ์ที่เกี่ยวข้องกับโหนดปลายเสมือนนี้ ดังแสดงในรูปที่ 81

```

2021-08-07 23:09:36,699 INFO [default]
VEN ---> VTN
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<p1:oadrPayload xmlns:p1="http://openadr.org/oadr-2.0b/2012/07">
  <p1:oadrSignedObject>
    <p1:oadrRequestEvent xmlns:p3="http://docs.oasis-open.org/ns/energyinterop/201110" p3:schemaVersion="2.0b" xmlns:p2="http://docs.oasis-open.org/ns/energyinterop/201110/payloads">
      <p2:eiRequestEvent xmlns:p3="http://docs.oasis-open.org/ns/energyinterop/201110">
        <p2:requestID>98AD6CD557</p2:requestID>
        <p3:venID>cc312a0bf5d25600b86d</p3:venID>
      </p2:eiRequestEvent>
    </p1:oadrRequestEvent>
  </p1:oadrSignedObject>
</p1:oadrPayload>
=====

```

รูปที่ 81 แสดงการโต้ตอบของชุดข้อมูล oadrRequestEvent

จากรูปที่ 82 โหนดต้นเสมือนจะทำการตอบกลับด้วย oadrDistributeEvent เพื่อแสดงถึงเหตุการณ์ที่เกี่ยวข้องกับโหนดปลายเสมือนตัวนั้น ๆ เพียงแต่ในโปรแกรมตัวอย่างนี้ยังไม่มีการสร้างเหตุการณ์ใด ๆ ขึ้น โดยจะแสดงการตอบกลับนี้ในกรณีที่ได้รับเหตุการณ์ในลำดับถัดไป

```

2021-08-07 23:09:36,707 INFO [default]
VEN <--- VTN
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:oadrPayload xmlns="http://docs.oasis-open.org/ns/emix/2011/06/power" xmlns:ns2="http://docs.oasis-open.org/ns/emix/2011/06/siscale"
  xmlns:ns3="http://openadr.org/oadr-2.0b/2012/07" xmlns:ns4="http://www.w3.org/2005/Atom" xmlns:ns5="http://www.opengis.net/gml/3.2" xmlns:ns6="http://docs.oasis-open.org/ns/emix/2011/06" xmlns:ns7="http://www.w3.org/2009/xmldsig11#" xmlns:ns8="http://www.w3.org/2000/09/xmldsig#"
  xmlns:ns9="http://openadr.org/oadr-2.0b/2012/07/xmldsig-properties" xmlns:ns10="urn:ietf:params:xml:ns:icalendar-2.0" xmlns:ns11="http://docs.oasis-open.org/ns/energyinterop/201110"
  xmlns:ns12="http://docs.oasis-open.org/ns/energyinterop/201110/payloads" xmlns:ns13="urn:ietf:params:xml:ns:icalendar-2.0:stream" xmlns:ns14="urn:un:unece:uncefact:odelist:standard:5:ISO42173A:2010-04-07">
  <ns3:oadrSignedObject>
    <ns3:oadrDistributeEvent ns11:schemaVersion="2.0b">
      <ns11:eiResponse>
        <ns11:responseCode>200</ns11:responseCode>
        <ns11:responseDescription>OK</ns11:responseDescription>
        <ns12:requestID>ecad1e148eef4c13b95c</ns12:requestID>
      </ns11:eiResponse>
      <ns12:requestID>d049d61fda22304b7792</ns12:requestID>
      <ns11:vtnID>EPRI_VTN</ns11:vtnID>
    </ns3:oadrDistributeEvent>
  </ns3:oadrSignedObject>
</ns3:oadrPayload>
=====

```

รูปที่ 82 แสดงการโต้ตอบของชุดข้อมูล oadrDistributeEvent

หลังจากนั้น โหนดปลายเสมือนจะทำการตอบกลับเหตุการณ์ด้วย oadrCreatedEvent กลับไปยังโหนดต้นเสมือน ดังแสดงในรูปที่ 83 ซึ่งถ้าหากมีเหตุการณ์ที่เกี่ยวข้องกับโหนดปลายเสมือนนี้ โหนดปลายเสมือนจะแสดงความต้องการเข้าร่วมมาตรการ optIn หรือ optOut ไปกับชุดข้อมูล oadrCreatedEvent ด้วย

```

2021-08-07 23:09:36,707 INFO [default]
VEN ---> VTN
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<p1:oadrPayload xmlns:p1="http://openadr.org/oadr-2.0b/2012/07">
  <p1:oadrSignedObject>
    <p1:oadrCreatedEvent xmlns:p3="http://docs.oasis-open.org/ns/energyinterop/201110" p3:schemaVersion="2.0b" xmlns:p2="http://docs.oasis-open.org/ns/energyinterop/201110/payloads">
      <p2:eiCreatedEvent xmlns:p3="http://docs.oasis-open.org/ns/energyinterop/201110">
        <p3:eiResponse>
          <p3:responseCode>200</p3:responseCode>
          <p3:responseDescription>OK</p3:responseDescription>
          <p2:requestID>98AD6CD557</p2:requestID>
        </p3:eiResponse>
        <p3:eventResponses />
        <p3:venID>cc312a0bf5d25600b86d</p3:venID>
      </p2:eiCreatedEvent>
    </p1:oadrCreatedEvent>
  </p1:oadrSignedObject>
</p1:oadrPayload>
=====

```

รูปที่ 83 แสดงการโต้ตอบของชุดข้อมูล oadrCreatedEvent

จากรูปที่ 84 โหนดต้นเสมือนทำการตอบกลับด้วย oadrResponse กลับไปยังโหนดปลายทางเสมือนเพื่อรับทราบข้อมูล

```

2021-08-07 23:09:36,715 INFO [default]
VEN <--- VTN
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:oadrPayload xmlns="http://docs.oasis-open.org/ns/emix/2011/06/power" xmlns:ns2="http://docs.oasis-open.org/ns/emix/2011/06/siscale"
  xmlns:ns3="http://openadr.org/oadr-2.0b/2012/07" xmlns:ns4="http://www.w3.org/2005/Atom" xmlns:ns5="http://www.opengis.net/gml/3.2" xmlns:ns6="http://docs.oasis-open.org/ns/emix/2011/06" xmlns:ns7="http://www.w3.org/2009/xmlsig11#" xmlns:ns8="http://www.w3.org/2000/09/xmlsig#"
  xmlns:ns9="http://openadr.org/oadr-2.0b/2012/07/xmlsig-properties" xmlns:ns10="urn:ietf:params:xml:ns:icalendar-2.0" xmlns:ns11="http://docs.oasis-open.org/ns/energyinterop/201110"
  xmlns:ns12="http://docs.oasis-open.org/ns/energyinterop/201110/payloads" xmlns:ns13="urn:ietf:params:xml:ns:icalendar-2.0:stream" xmlns:ns14="urn:un:unece:uncefact:odelist:standard:5:ISO42173A:2010-04-07">
  <ns3:oadrSignedObject>
    <ns3:oadrResponse ns11:schemaVersion="2.0b">
      <ns11:eiResponse>
        <ns11:responseCode>200</ns11:responseCode>
        <ns11:responseDescription>OK</ns11:responseDescription>
        <ns12:requestID>98AD6CD557</ns12:requestID>
      </ns11:eiResponse>
      <ns11:venID>cc312a0bf5d25600b86d</ns11:venID>
    </ns3:oadrResponse>
  </ns3:oadrSignedObject>
</ns3:oadrPayload>
=====

```

รูปที่ 84 แสดงการโต้ตอบของชุดข้อมูล oadrResponse

โดยหลังจากนี้โหนดปลายทางเสมือนจะคอยส่งชุดคำร้อง oadrPoll มายังโหนดต้นเสมือนเพื่อคอยตรวจสอบสถานะต่าง ๆ ดังแสดงในรูปที่ 85 ส่วนโหนดต้นเสมือนก็จะตอบกลับด้วย oadrResponse เพื่อตอบรับอยู่เสมอ จนกว่าจะมีการเรียกใช้บริการอื่น หรือจนกว่าจะมีการเปลี่ยนแปลงข้อมูลใด ๆ


```

=====
2021-08-07 23:09:36,716 INFO [default]
VEN ---> VTN
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<p1:oadrPayload xmlns:p1="http://openadr.org/oadr-2.0b/2012/07">
  <p1:oadrSignedObject>
    <p1:oadrPoll xmlns:p2="http://docs.oasis-open.org/ns/energyinterop/201110" p2:schemaVersion="2.0b">
      <p2:venID>cc312a0bf5d25600b86d</p2:venID>
    </p1:oadrPoll>
  </p1:oadrSignedObject>
</p1:oadrPayload>
=====

```

รูปที่ 85 แสดงการโต้ตอบของชุดข้อมูล oadrPoll

5.6.4 ทดสอบการทำงานตอบสนองด้านโหลดด้วยซอฟต์แวร์ OpenADR ที่พัฒนาขึ้น

ผู้วิจัยได้ทำการพัฒนาดัดแปลงซอฟต์แวร์การตอบสนองด้านโหลดของโหนดปลายเสมือน เพื่อให้มีความสามารถในการตอบรับการเข้าร่วมมาตรการการตอบสนองด้านโหลดได้ โดยเพื่อความสะดวก ผู้ใช้งานโปรแกรมสามารถเริ่มโปรแกรมการทำงานด้วยการใส่อาร์กิวเมนต์เพิ่มอีกหนึ่งตัวคือ อาร์กิวเมนต์ค่าเริ่มต้นเข้าร่วมมาตรการ (Default opt) ตามหลังคำสั่งเรียกใช้งานโปรแกรมโหนดปลายเสมือน โดยตัวอย่างคำสั่งเรียกใช้โปรแกรมคือ

```
~$ ./venmgr http://161.200.85.66:8080/ ESID-VEN in
```

หรือ

```
~$ ./venmgr http://161.200.85.66:8080/ ESID-VEN out
```

โดยที่ `venmgr` คือ โปรแกรมสำหรับการโต้ตอบกับโหนดต้นเสมือนที่ผู้วิจัยได้พัฒนาขึ้น
`http://161.200.85.66:8080/` คือ ที่อยู่ของโหนดต้นเสมือนที่ผู้วิจัยสร้างขึ้น
`ESID-VEN` คือ ชื่อของโหนดปลายเสมือนที่ได้ทำการลงทะเบียนไว้แล้ว
`In หรือ out` คือ ค่าเริ่มต้นเข้าร่วมมาตรการของโหนดปลายเสมือน

ค่าเริ่มต้นเข้าร่วมมาตรการนั้น หากใส่ค่าเป็น “in” โหนดปลายเสมือนจะทำการตั้งค่าให้กับเหตุการณ์ต่าง ๆ เป็นเข้าร่วมมาตรการ (optIn) แต่หากใส่ค่าเริ่มต้นเป็น “out” โหนดปลายเสมือนจะตั้งค่าให้เป็นไม่เข้าร่วมมาตรการ (optOut)

ในกรณีเข้าร่วมมาตรการ (optIn) หลังจากทีโหนดปลายเสมือนส่งชุดข้อมูล oadrPoll เพื่อตรวจสอบสถานะและได้รับเหตุการณ์ที่สร้างใหม่จากโหนดต้นเสมือนด้วยชุดข้อมูล oadrDistributeEvent ซึ่งจะมีรายละเอียดของเหตุการณ์ที่สำคัญเช่น อดีเหตุการณ์ เวลาเริ่มการทำงาน และระยะเวลาการทำงาน โหนดปลายเสมือนจะทำการสร้างชุดข้อมูลเข้าร่วมมาตรการของแต่ละเหตุการณ์ (ซึ่งอาจมีหนึ่งเหตุการณ์หรือมากกว่านั้น) ด้วยชุดข้อมูล oadrCreatedEvent ตอบกลับไปยังโหนดต้นเสมือน ดังแสดงให้เห็นในรูปที่ 86 และ 87 ตามลำดับ

```

2021-08-12 10:43:34,539 INFO [default]
VEN <--- VTN
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:oadrPayload xmlns="http://www.w3.org/2005/Atom" xmlns:ns2="http://docs.oasis-open.org/ns/energyinterop/201110" xmlns:ns3="http://openadr.org/oadr-2.0b/2012/07" xmlns:ns4="http://docs.oasis-open.org/ns/energyinterop/201110/payloads" xmlns:ns5="urn:ietf:params:xml:ns:icalendar-2.0" xmlns:ns6="urn:ietf:params:xml:ns:icalendar-2.0:stream" xmlns:ns7="http://docs.oasis-open.org/ns/emix/2011/06/power" xmlns:ns8="http://www.opengis.net/gml/3.2" xmlns:ns9="http://docs.oasis-open.org/ns/emix/2011/06" xmlns:ns10="http://docs.oasis-open.org/ns/emix/2011/06/siscale" xmlns:ns11="http://www.w3.org/2000/09/xmldsig#" xmlns:ns12="http://www.w3.org/2009/xmldsig1#" xmlns:ns13="http://openadr.org/oadr-2.0b/2012/07/xmldsig-properties" xmlns:ns14="urn:un:uncece:uncefact:codeList:standard:5:ISO42173A:2010-04-07">
  <ns3:oadrSignedObject>
    <ns3:oadrDistributeEvent ns2:schemaVersion="2.0b">
      <ns2:eiResponse>
        <ns2:responseCode>200</ns2:responseCode>
        <ns2:responseDescription>OK</ns2:responseDescription>
        <ns4:requestID>5387ed0365b1b56d1e6d</ns4:requestID>
      </ns2:eiResponse>
      <ns4:requestID>e5e4717d4ca0e390a49f</ns4:requestID>
      <ns2:vtnID>EPRI_VTN</ns2:vtnID>
      <ns3:oadrEvent>
        <ns2:eiEvent>
          <ns2:eventDescriptor>
            <ns2:eventID>08648bb799a79fc84933</ns2:eventID>
            <ns2:modificationNumber>0</ns2:modificationNumber>
            <ns2:modificationReason></ns2:modificationReason>
            <ns2:priority>0</ns2:priority>
            <ns2:eiMarketContext>
              <ns9:marketContext>http://MarketContext1</ns9:marketContext>
            </ns2:eiMarketContext>
            <ns2:createdDateTime>2021-08-12T03:43:30.000Z</ns2:createdDateTime>
          </ns2:eventDescriptor>
        </ns2:eiEvent>
      </ns3:oadrEvent>
    </ns3:oadrDistributeEvent>
  </ns3:oadrSignedObject>
</ns3:oadrPayload>

```

รูปที่ 86 แสดงการโต้ตอบเมื่อได้รับเหตุการณ์จากโหนดต้นเสมือนด้วย `oadrDistributeEvent`

```

2021-08-12 10:43:34,540 INFO [default] new event received: 08648bb799a79fc84933 | setDefaultOpt = optIn
2021-08-12 10:43:34,541 INFO [default]
VEN --> VTN
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<p1:oadrPayload xmlns:p1="http://openadr.org/oadr-2.0b/2012/07">
  <p1:oadrCreatedEvent xmlns:p3="http://docs.oasis-open.org/ns/energyinterop/201110" p3:schemaVersion="2.0b" xmlns:p2="http://docs.oasis-open.org/ns/energyinterop/201110/payloads">
    <p2:eiCreatedEvent xmlns:p3="http://docs.oasis-open.org/ns/energyinterop/201110">
      <p3:eiResponse>
        <p3:responseCode>200</p3:responseCode>
        <p3:responseDescription>OK</p3:responseDescription>
        <p2:requestID/>
      </p3:eiResponse>
      <p3:eventResponses>
        <p3:eventResponse>
          <p3:responseCode>200</p3:responseCode>
          <p3:responseDescription>OK</p3:responseDescription>
          <p2:requestID>e5e4717d4ca0e390a49f</p2:requestID>
          <p3:qualifiedEventID>
            <p3:eventID>08648bb799a79fc84933</p3:eventID>
            <p3:modificationNumber>0</p3:modificationNumber>
          </p3:qualifiedEventID>
          <p3:optType>optIn</p3:optType>
        </p3:eventResponse>
      </p3:eventResponses>
      <p2:venID>cc312a0bf5d25600b86d</p2:venID>
    </p2:eiCreatedEvent>
  </p1:oadrCreatedEvent>
</p1:oadrPayload>

```

รูปที่ 87 แสดงการโต้ตอบเมื่อตอบกลับการเข้าร่วมมาตรการด้วย `oadrCreatedEvent`

เมื่อถึงเวลาของเหตุการณ์ฯ โหนดปลายเสมือนจะทำการส่งสัญญาณการตอบสนองด้านโหลดเพื่อลดการใช้พลังงานไปยังอุปกรณ์เครื่องใช้ไฟฟ้าที่เกี่ยวข้อง ดังรูปที่ 88

```

2021-08-12 10:45:00,541 INFO [default] event start: 08648bb799a79fc84933 | Chosen optType = optIn
2021-08-12 10:45:00,541 INFO [default] event interval start: 08648bb799a79fc849335 | Remaining 900 sec

>>>> YOUR APPLIANCE IS TURNING OFF <<<<<

2021-08-12 10:45:04,896 INFO [default]
VEN ---> VTN
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<p1:oadrPayload xmlns:p1="http://openadr.org/oadr-2.0b/2012/07">
  <p1:oadrSignedObject>
    <p1:oadrPoll xmlns:p2="http://docs.oasis-open.org/ns/energyinterop/201110" p2:schemaVersion="2.0b">
      <p2:venID>cc312a0bf5d25600b86d</p2:venID>
    </p1:oadrPoll>
  </p1:oadrSignedObject>
</p1:oadrPayload>

=====

```

รูปที่ 88 แสดงการทำงานเมื่อถึงเวลาของเหตุการณ์การตอบสนองด้านโหลดกรณีเข้าร่วมมาตรการ

หลังจากที่เหตุการณ์เสร็จสิ้น ในรูปที่ 89 โหนดปลายเสมือนจะทำการส่งสัญญาณไปยังอุปกรณ์ที่เกี่ยวข้องเพื่อเปิดใช้งานอุปกรณ์ตามปกติ เป็นอันเสร็จสิ้นกระบวนการการตอบสนองด้านโหลดแบบอัตโนมัติด้วยโพรโทคอล OpenADR

```

2021-08-12 11:00:00,541 INFO [default] event complete: 08648bb799a79fc84933 | Chosen optType = optIn

>>>> YOUR APPLIANCE IS TURNING ON <<<<<

2021-08-12 11:00:08,545 INFO [default]
VEN ---> VTN
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<p1:oadrPayload xmlns:p1="http://openadr.org/oadr-2.0b/2012/07">
  <p1:oadrSignedObject>
    <p1:oadrPoll xmlns:p2="http://docs.oasis-open.org/ns/energyinterop/201110" p2:schemaVersion="2.0b">
      <p2:venID>cc312a0bf5d25600b86d</p2:venID>
    </p1:oadrPoll>
  </p1:oadrSignedObject>
</p1:oadrPayload>

=====

```

รูปที่ 89 แสดงการทำงานเมื่อสิ้นสุดเหตุการณ์การตอบสนองด้านโหลด

ในทางกลับกัน หากตั้งค่าเริ่มต้นเข้าร่วมมาตรการเป็นไม่เข้าร่วม (optOut) โหนดปลายเสมือนจะไม่มี การตอบสนองด้านโหลดใด ๆ ถึงแม้ว่าโหนดปลายเสมือนจะได้รับเหตุการณ์การตอบสนองด้านโหลดก็ตาม ดังแสดงให้เห็นในรูปที่ 90

```

2021-08-12 13:15:00,139 INFO [default] event start: 08648bb799a79fc84933 | Chosen optType = optOut
2021-08-12 13:15:00,140 INFO [default] event interval start: 08648bb799a79fc849335 | Remaining 900 sec
2021-08-12 13:15:04,737 INFO [default]
VEN ---> VTN
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<p1:oadrPayload xmlns:p1="http://openadr.org/oadr-2.0b/2012/07">
  <p1:oadrSignedObject>
    <p1:oadrPoll xmlns:p2="http://docs.oasis-open.org/ns/energyinterop/201110" p2:schemaVersion="2.0b">
      <p2:venID>cc312a0bf5d25600b86d</p2:venID>
    </p1:oadrPoll>
  </p1:oadrSignedObject>
</p1:oadrPayload>

=====

```

รูปที่ 90 แสดงการทำงานเมื่อถึงเวลาของเหตุการณ์การตอบสนองด้านโหลดกรณีไม่เข้าร่วมมาตรการ

บทที่ 6

สรุปผลงานวิจัยและข้อเสนอแนะ

6.1 สรุปผลงานวิจัย

จากผลการทดลองในบทที่ 5 จะเห็นได้ว่ามิเตอร์อัจฉริยะต้นแบบที่ผู้วิจัยได้พัฒนาขึ้นสามารถวัดค่าพลังงานทางไฟฟ้าตามข้อกำหนดตามมาตรฐาน IEC 62053-21 Class 1 ซึ่งเป็นมาตรฐานสำหรับทดสอบมิเตอร์ที่ใช้กับระบบจำหน่ายไฟฟ้าแรงดันต่ำ (ไม่เกิน 1,000 โวลต์) โดยมีมิเตอร์อัจฉริยะนี้สามารถอ่านค่าพลังงานทางไฟฟ้า ได้แก่

- ค่าแรงดันไฟฟ้า (Voltage)
- ค่ากระแสไฟฟ้า (Current)
- ค่ากำลังไฟฟ้าจริง (Active Power)
- ค่ากำลังไฟฟ้ารีแอกทีฟ (Reactive Power)
- ค่ากำลังไฟฟ้าปรากฏ (Apparent Power)
- ค่าตัวประกอบกำลังไฟฟ้า (Power Factor) และ
- ค่าความถี่ทางไฟฟ้า (Frequency)

ESP32 สามารถเชื่อมต่อสัญญาณอินเทอร์เน็ตไร้สาย เพื่อดึงค่าเวลาจากเซิร์ฟเวอร์ฐานเวลาอ้างอิงในการนำมาใช้เป็นตราเวลาในการเก็บลงในฐานข้อมูล อีกทั้งยังสามารถส่งข้อมูลไปยังคลาวด์แพลตฟอร์มเพื่อแสดงผลค่าพลังงานไฟฟ้าได้ตามเวลาจริง (Real-time) ทำให้สามารถเฝ้าสังเกตการณ์ผ่านเว็บไซต์จากทุกที่ทุกเวลา

ในงานวิจัยฉบับนี้ มิเตอร์อัจฉริยะจะมีความสามารถสอบเทียบตัวเองได้ โดยผลการทดสอบในบทที่ 5 ทำให้เห็นว่ามีเตอร์อัจฉริยะต้นแบบนี้สามารถทำการสอบเทียบตัวเองเพื่อรักษาเสถียรภาพความแม่นยำ ซึ่งจะส่งผลให้การวัดค่าพลังงานทางไฟฟ้าเป็นไปได้อย่างมีประสิทธิภาพอยู่เสมอ

มิเตอร์อัจฉริยะต้นแบบนี้ยังสามารถเก็บข้อมูลลงในหน่วยความจำเมื่อมีการวัดค่าพลังงานทางไฟฟ้าด้วยซอฟต์แวร์ฐานข้อมูลขนาดเล็กอย่าง SQLite ด้วยภาษาโปรแกรม SQL ทำให้ข้อมูลที่เก็บลงในฐานข้อมูลนั้นมีความเป็นระเบียบ เมื่อมีการนำข้อมูลที่เก็บลงในฐานข้อมูลดังกล่าวมาใช้จะสามารถใช้คำสั่งเรียกใช้ได้อย่างง่ายด้วยคำสั่งเพียงไม่กี่บรรทัด

นอกจากนี้ มิเตอร์อัจฉริยะที่ผู้วิจัยได้พัฒนาขึ้นมีความสามารถในการสื่อสารโต้ตอบผ่านโพรโทคอล OpenADR 2.0 ซึ่งเป็นมาตรฐานที่สำคัญสำหรับการตอบสนองด้านโหลดอัตโนมัติ ทำให้

สามารถทำการตอบสนองด้านโหลดเพื่อจัดการปรับเปลี่ยนการใช้งานโหลดได้อย่างรวดเร็วโดยไม่ต้องมีเจ้าหน้าที่ทำการควบคุมเครื่องใช้ไฟฟ้า แน่ใจว่าจะทำให้สถานะในโครงข่ายไฟฟ้ามีเสถียรภาพที่ดีขึ้น มีความยืดหยุ่นมากขึ้น อันเนื่องมาจากการปรับเปลี่ยนพฤติกรรมการใช้พลังงานในช่วงเวลาที่มีความต้องการไฟฟ้าสูง

6.2 ข้อเสนอแนะและแนวทางในการพัฒนาต่อ

จากการทดลองที่ผ่านมา การตอบสนองด้านโหลดอัตโนมัติด้วยโพรโทคอล OpenADR ที่ผู้วิจัยได้พัฒนาขึ้นมานั้นยังมีความสามารถที่จำกัด โดยที่ผู้วิจัยได้ทำการควบคุมเครื่องใช้ไฟฟ้าโดยตรงไปที่ตัวเครื่องใช้ไฟฟ้าเมื่อถึงเวลาของเหตุการณ์การตอบสนองด้านโหลด ซึ่งผู้วิจัยเล็งเห็นแนวทางในการพัฒนาต่อโดยสามารถเพิ่มโพรโทคอลสื่อสารระหว่างโหนดปลายเสมือนกับอุปกรณ์เครื่องใช้ไฟฟ้าในการควบคุมอุปกรณ์อย่างโพรโทคอล ECHONET Lite ซึ่งจะทำให้การควบคุมอุปกรณ์ปลายทางนั้นสามารถทำได้อย่างเป็นระบบระเบียบ และมีประสิทธิภาพที่มากกว่า

นอกจากนี้ยังสามารถประยุกต์ใช้โพรโทคอล OpenADR ให้มีความสะดวกในการใช้งานมากขึ้นโดยการนำ OpenADR ไปประยุกต์ใช้งานเข้ากับ NETPIE เพื่อใช้ในการแสดงผลสถานะต่าง ๆ ผ่านเว็บไซต์ ทำให้ผู้ใช้ไฟฟ้าสามารถควบคุมการเข้าร่วมมาตรการ (optIn/optOut) ได้สะดวกมากยิ่งขึ้น หรือแสดงผลสถานะทางไฟฟ้าของโหนดต่าง ๆ ที่เชื่อมต่ออยู่กับโหนดปลายเสมือน เป็นต้น หรืออีกกรณีหนึ่งคือนำไปสร้างแอปพลิเคชันมือถือทำให้มีความสามารถในการแจ้งเตือนผู้ใช้งานไฟฟ้าเมื่อมีเหตุการณ์การตอบสนองด้านโหลดใหม่ หรือแสดงผลค่าพลังงานที่สามารถช่วยลดระหว่างดำเนินการตอบสนองด้านโหลด เป็นต้น

บรรณานุกรม

1. Mohassel, R.R., et al., *A survey on advanced metering infrastructure*. International Journal of Electrical Power & Energy Systems, 2014. 63: p. 473-484.
2. Strategy, N.M.G., *Advanced metering infrastructure*. US Department of Energy Office of Electricity and Energy Reliability, 2008.
3. สถาบันวิจัยพลังงาน จุฬาลงกรณ์มหาวิทยาลัย โดย ผศ. ดร. วันเฉลิม โปรา, ความรู้พื้นฐานเกี่ยวกับมาตรอัจฉริยะ (*Smart Meter*) และแนวทางการจัดทำข้อกำหนดเฉพาะมาตรอัจฉริยะ. 2014: p. 12.
4. ผศ. ดร. ประยูร เขียววัฒนา, *สมาร์ททกริด นวัตกรรมสำหรับระบบไฟฟ้าในอนาคต*. Technology Promotion, 2012. 39.
5. สำนักงานนโยบายและแผนพลังงาน กระทรวงพลังงาน, *คู่มือโครงข่ายไฟฟ้าอัจฉริยะ (Smart Grid)*. 2019: p. 40.
6. กระทรวงพลังงาน, *แผนแม่บทการพัฒนา ระบบโครงข่ายสมาร์ททกริดของประเทศไทย พ.ศ. 2558-2579*. 2015.
7. Wang, S., X. Xue, and C. Yan, *Building power demand response methods toward smart grid*. Hvac&r Research, 2014. 20(6): p. 665-687.
8. Dodrill, K., *Demand Dispatch—Intelligent Demand for a More Efficient Grid*. 2011.
9. สำนักงานนโยบายและแผนพลังงาน กระทรวงพลังงาน, *การพัฒนารูปแบบธุรกิจการตอบสนองด้านโหลด (Demand Response) สำหรับประเทศไทย*. 2019.
10. OpenADR Alliance, *OpenADR 2.0 Demand Response Program, Implementation Guide*. 2016. 1.1.
11. National Electronics and Computer Technology Center (NECTEC), *คู่มือการใช้งาน NETPIE-An Official Guide to NETPIE*. 2017.
12. Analog Devices, Inc. *Energy metering IC with Autocalibration*. 2018.
13. OpenADR Alliance, *OpenADR 2.0 Profile Specification B Profile*. 2013. 1.0.
14. Electric Power Research Institute, Inc. *OpenADR 2.0 Open Source Virtual Top Node - VTN User's Manual, Software Version 0.9.7*. 2016.
15. Electric Power Research Institute, Inc. *OpenADR 2.0b Open-Source Virtual End Node C++ Library User's Manual, Software Version 0.7.0*. 2016.
16. SATEC Ltd., *Accuracy class: A small "s" that makes a big difference*. 2018 (July

2018): p. 6.

17. Electricity metering equipment (a.c.) - Particular requirements - Part 21: Static meters for active energy (classes 1 and 2), *IEC 62053-21:2003*.





จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ประวัติผู้เขียน

ชื่อ-สกุล	นพเมธ ปวีรพาณิชย์
วัน เดือน ปี เกิด	4 กุมภาพันธ์ 2538
สถานที่เกิด	นนทบุรี
วุฒิการศึกษา	จบการศึกษาหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมอิเล็กทรอนิกส์ การบิน (ต่อเนื่อง) สถาบันการบินพลเรือน ในปี 2561 และเข้าศึกษาต่อ หลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ภาควิชาวิศวกรรมไฟฟ้า คณะ วิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2562
ที่อยู่ปัจจุบัน	กรุงเทพฯ
ผลงานตีพิมพ์	A Single-phase Smart Meter With Autocalibration and Load Control Conformed to OpenADR Protocol

