

บทที่ 3

วิธีการดำเนินการวิจัย

3.1 บทนำ

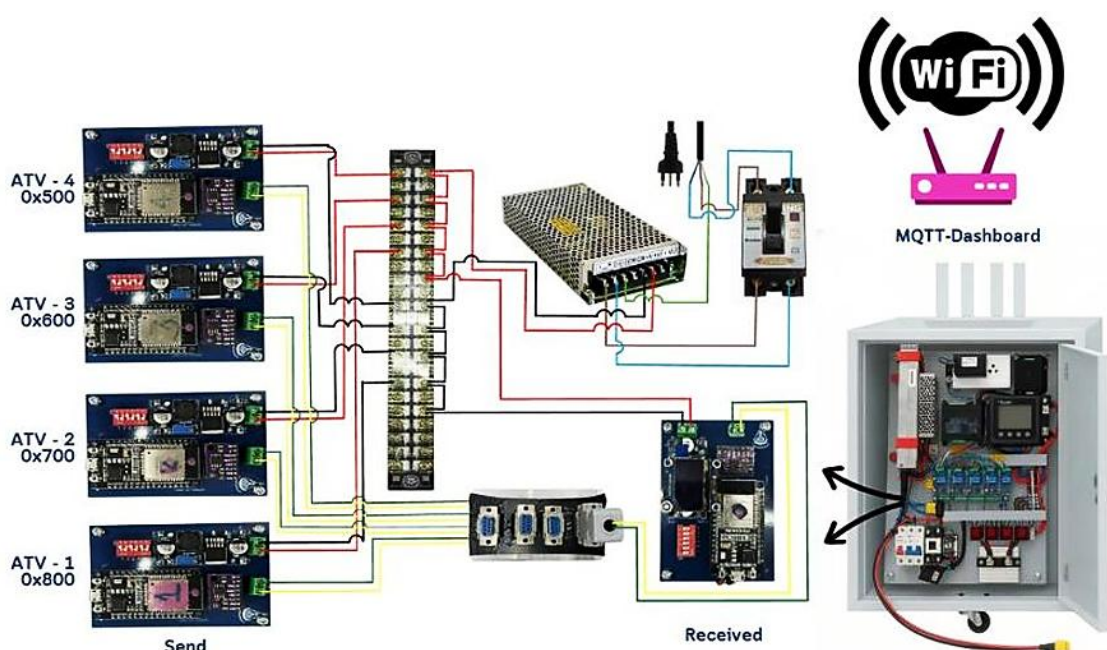
บทนี้นำเสนอวิธีดำเนินการวิจัยสำหรับการออกแบบ พัฒนา และประเมินสมรรถนะของระบบเลือกโปรไฟล์การชาร์จสำหรับแบตเตอรี่ลิเทียมไอออนชนิด NMC แรงดันต่ำของยานยนต์ไฟฟ้า ATV โดยมุ่งเน้นการควบคุมกระบวนการชาร์จแบบฟิกส์ CC-CV ให้สอดคล้องกับคุณลักษณะเฉพาะของแบตเตอรี่แต่ละระบบอย่างถูกต้อง มีเสถียรภาพ และสามารถทำงานได้อย่างต่อเนื่องภายใต้สภาวะการใช้งานจริง งานวิจัยนี้ให้ความสำคัญกับการลดความผิดพลาดจากการกำหนดค่าพารามิเตอร์ด้วยมนุษย์ ซึ่งเป็นหนึ่งในปัจจัยที่ส่งผลต่อความปลอดภัย ประสิทธิภาพ และความน่าเชื่อถือของระบบชาร์จแบตเตอรี่ โดยเฉพาะในระบบที่รองรับแบตเตอรี่หลายระดับแรงดันซึ่งต้องใช้เงื่อนไขการชาร์จแตกต่างกันในแต่ละระบบ

แนวคิดหลักของงานวิจัยคือการบูรณาการระบบสื่อสาร ระบบควบคุม และระบบติดตามข้อมูลแบบเรียลไทม์เข้าด้วยกัน ภายใต้โครงสร้างการทำงานแบบหลายระดับ (Multi-layer architecture) เพื่อให้สามารถระบุข้อมูลประจำแบตเตอรี่ เลือกโปรไฟล์การชาร์จ และกำหนดค่าพารามิเตอร์ของเครื่องชาร์จได้อย่างเป็นเชิงกำหนด (Deterministic) โดยใช้ CANopen เป็นกลไกหลักในการระบุพารามิเตอร์ของแบตเตอรี่ผ่าน CAN Identifier ซึ่งข้อมูลที่ได้รับจะถูกส่งเข้าสู่ Supervisory layer ผ่าน MQTT เพื่อให้แพลตฟอร์ม Node-RED ทำหน้าที่ประมวลผล วิเคราะห์ และเลือกโปรไฟล์การชาร์จที่เหมาะสม ก่อนถ่ายทอดคำสั่งไปยัง Execution layer ผ่าน Modbus RTU เพื่อกำหนดค่าการทำงานของเครื่องชาร์จแบบโปรแกรมได้

ในงานวิจัยนี้ การกำหนดค่าพารามิเตอร์ของเครื่องชาร์จไม่ได้อาศัยการปรับค่าผ่านซอฟต์แวร์เพียงอย่างเดียว แต่ใช้แนวทาง hardware-enforced parameter configuration ซึ่งกำหนดค่าความต้านทานและพารามิเตอร์ของระบบชาร์จไว้ล่วงหน้าในระดับฮาร์ดแวร์ เพื่อให้ค่าการชาร์จที่ถูกเลือกไม่สามารถ

เปลี่ยนแปลงได้ระหว่างกระบวนการทำงาน แนวทางดังกล่าวช่วยลดความเสี่ยงจากความผิดพลาดของระบบควบคุม เพิ่มความต่อเนื่องของกระบวนการชาร์จ และช่วยให้ระบบสามารถทำงานซ้ำได้อย่างมีประสิทธิภาพในทุกเงื่อนไขการทดลอง

นอกจากนี้ ระบบยังได้รับการออกแบบให้รองรับการติดตามข้อมูลแบบเรียลไทม์ทั้งด้านอินพุตและเอาต์พุตของระบบชาร์จ โดยข้อมูลด้านอินพุตกระแสสลับ เช่น แรงดัน กระแส กำลังไฟฟ้า และพลังงานสะสม จะถูกจัดเก็บผ่าน Google Sheet เพื่อใช้ในการวิเคราะห์พลังงานและคำนวณต้นทุนการชาร์จภายใต้อัตราค่าไฟฟ้าแบบ Time of Use (TOU) ขณะที่ข้อมูลด้านเอาต์พุตกระแสตรง เช่น ค่าแรงดัน กระแส และพฤติกรรมของแบตเตอรี่ระหว่างกระบวนการชาร์จแบบฟูลล์ CC-CV จะถูกจัดเก็บในฐานข้อมูล InfluxDB และแสดงผลแบบเรียลไทม์ผ่าน Grafana Dashboard เพื่อใช้ในการติดตามเสถียรภาพและสมรรถนะของระบบชาร์จ



รูปที่ 3.1 สถาปัตยกรรมระบบเลือกโปรไฟล์การชาร์จแบตเตอรี่ NMC ผ่าน CANopen - Modbus RTU

จากรูปที่ 3.1 สถาปัตยกรรมของระบบประกอบด้วยหน่วยสื่อสาร CANopen ที่ติดตั้งร่วมกับแบตเตอรี่แต่ละระบบ ทำหน้าที่ส่งข้อมูลพารามิเตอร์ของแบตเตอรี่ เช่น ระดับแรงดัน ความจุ และเงื่อนไขการชาร์จ ผ่านเครือข่าย CAN ไปยังโมดูลรับข้อมูลภายในตู้ควบคุม จากนั้นข้อมูลดังกล่าวจะถูกส่งผ่าน MQTT ไปยังระบบ Supervisory control เพื่อให้ Node-RED ประมวลผลและเลือกโปรไฟล์การชาร์จที่สอดคล้องกับแบตเตอรี่แต่ละระบบ ก่อนส่งคำสั่งผ่าน Modbus RTU ไปยัง

กระบวนการควบคุมในระดับ Execution layer ซึ่งทำหน้าที่กำหนดค่าพารามิเตอร์ของเครื่องชาร์จโดยตรง ขณะเดียวกัน ระบบยังเชื่อมต่อกับเครือข่าย Wi-Fi และ MQTT Dashboard เพื่อใช้ในการติดตามสถานะของระบบแบบเรียลไทม์ รวมถึงรองรับการจัดเก็บข้อมูลสำหรับการวิเคราะห์สมรรถนะพลังงาน และต้นทุนการชาร์จในภายหลัง

การดำเนินการวิจัยในบทนี้ครอบคลุมการออกแบบสถาปัตยกรรมระบบ การกำหนดโครงสร้างการสื่อสาร การพัฒนากลไกเลือกโปรไฟล์การชาร์จ การควบคุมพารามิเตอร์ในระดับปฏิบัติการ การจัดเก็บและติดตามข้อมูลแบบเรียลไทม์ ตลอดจนการออกแบบขั้นตอนการทดลองและเกณฑ์การประเมินผล เพื่อให้สามารถประเมินความถูกต้อง ความเสถียร ความต่อเนื่อง และประสิทธิภาพของระบบเลือกโปรไฟล์การชาร์จสำหรับแบตเตอรี่ NMC แรงดันต่ำได้อย่างเป็นระบบภายใต้สภาวะการใช้งานจริง

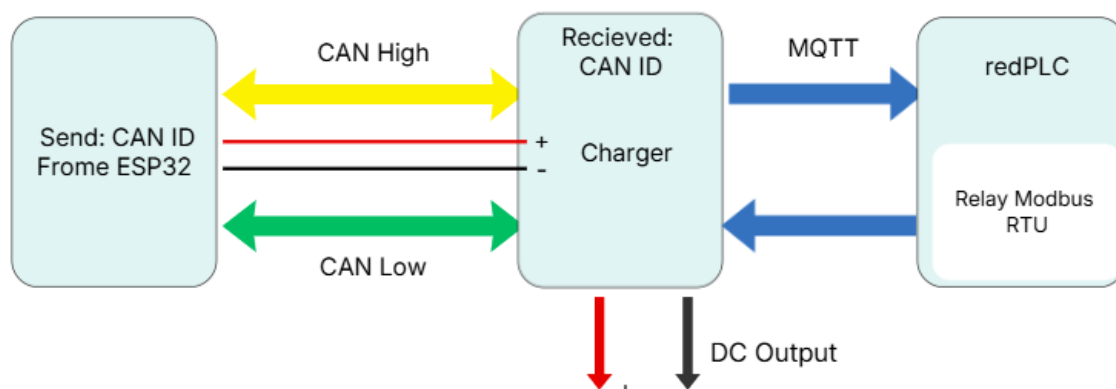
3.2 สถาปัตยกรรมระบบเลือกโปรไฟล์การชาร์จแบตเตอรี่ NMC แรงดันต่ำ

ระบบเลือกโปรไฟล์การชาร์จที่พัฒนาขึ้นในงานวิจัยนี้ถูกออกแบบภายใต้แนวคิดของการบูรณาการระบบกำลังไฟฟ้า ระบบควบคุม ระบบสื่อสาร และระบบติดตามข้อมูลแบบเรียลไทม์เข้าด้วยกัน เพื่อให้สามารถเลือกโปรไฟล์การชาร์จสำหรับแบตเตอรี่ NMC แรงดันต่ำได้อย่างถูกต้อง มีเสถียรภาพ และสามารถทำงานได้อย่างต่อเนื่องภายใต้สภาวะการใช้งานจริง โดยสถาปัตยกรรมของระบบถูกแบ่งออกเป็น Communication layer สำหรับรับส่งข้อมูลระหว่างอุปกรณ์ Supervisory layer สำหรับประมวลผลและเลือกโปรไฟล์การชาร์จ และ Execution layer สำหรับควบคุมค่าพารามิเตอร์ของเครื่องชาร์จแบบฟิกส์ CC-CV ให้สอดคล้องกับคุณลักษณะของแบตเตอรี่แต่ละระบบ นอกจากนี้ ระบบยังรองรับการติดตามข้อมูลด้านอินพุตและเอาต์พุตแบบเรียลไทม์ เพื่อใช้ในการวิเคราะห์พลังงาน ประเมินต้นทุนการชาร์จ และตรวจสอบเสถียรภาพของกระบวนการชาร์จในระหว่างการทดลอง

3.2.1 โครงสร้างระบบโดยรวม

โครงสร้างโดยรวมของระบบเลือกโปรไฟล์การชาร์จถูกออกแบบให้รองรับการทำงาน of แบตเตอรี่ NMC หลายระดับแรงดันภายในระบบเดียวกัน โดยอาศัยการบูรณาการระบบสื่อสาร ระบบควบคุม และระบบกำลังไฟฟ้าเข้าด้วยกัน เพื่อให้สามารถเลือกโปรไฟล์การชาร์จและกำหนดค่าพารามิเตอร์ของเครื่องชาร์จได้อย่างถูกต้องและเป็นเชิงกำหนด (Deterministic) ระบบเริ่มต้นจากหน่วยส่งข้อมูลที่ติดตั้งร่วมกับแบตเตอรี่ ทำหน้าที่ส่ง CAN Identifier ผ่านเครือข่าย CAN Bus ไปยังหน่วยรับข้อมูลภายในระบบควบคุม จากนั้นข้อมูลจะถูกส่งผ่าน MQTT ไปยังระบบ Supervisory control เพื่อประมวลผลและเลือกโปรไฟล์การชาร์จที่เหมาะสม ก่อนถ่ายทอดคำสั่งผ่าน Modbus RTU ไปยัง Execution layer สำหรับกำหนดค่ากระแสและแรงดันอ้างอิงของเครื่องชาร์จแบบฟิกส์

CC-CV โดยแนวทางดังกล่าวช่วยลดความผิดพลาดจากการกำหนดค่าด้วยมนุษย์ เพิ่มความต่อเนื่องในการทำงานของระบบ และช่วยให้การชาร์จแบตเตอรี่สามารถทำงานได้อย่างมีประสิทธิภาพภายใต้เงื่อนไขที่กำหนดไว้ล่วงหน้า



รูปที่ 3.2 โครงสร้างระบบเลือกโปรไฟล์การชาร์จผ่าน CANopen และ Modbus RTU

จากรูปที่ 3.2 จะเห็นได้ว่าระบบเริ่มต้นจากการส่งข้อมูล CAN ID ผ่านสายสัญญาณ CAN High และ CAN Low ไปยังหน่วยรับข้อมูลของเครื่องชาร์จ จากนั้นข้อมูลจะถูกส่งต่อผ่าน MQTT ไปยังระบบควบคุมส่วนกลางเพื่อประมวลผลและเลือกโปรไฟล์การชาร์จ ก่อนถ่ายทอดคำสั่งผ่าน Modbus RTU ไปยัง execution layer สำหรับกำหนดค่าการทำงานของเครื่องชาร์จ ขณะที่เอาต์พุตกระแสตรงของเครื่องชาร์จจะถูกส่งไปยังแบตเตอรี่เพื่อดำเนินกระบวนการชาร์จแบบฟิกส์ CC-CV ตามพารามิเตอร์ที่กำหนดไว้ในระบบควบคุม โดยโครงสร้างดังกล่าวช่วยให้ระบบสามารถเชื่อมโยงข้อมูลระหว่างระบบสื่อสารและระบบกำลังไฟฟ้าได้อย่างต่อเนื่อง และรองรับการทำงานของแบตเตอรี่หลายระดับแรงดันภายในสถาปัตยกรรมเดียวกันได้อย่างมีประสิทธิภาพ

3.2.2 อุปกรณ์ที่ใช้ในการพัฒนาระบบชาร์จ

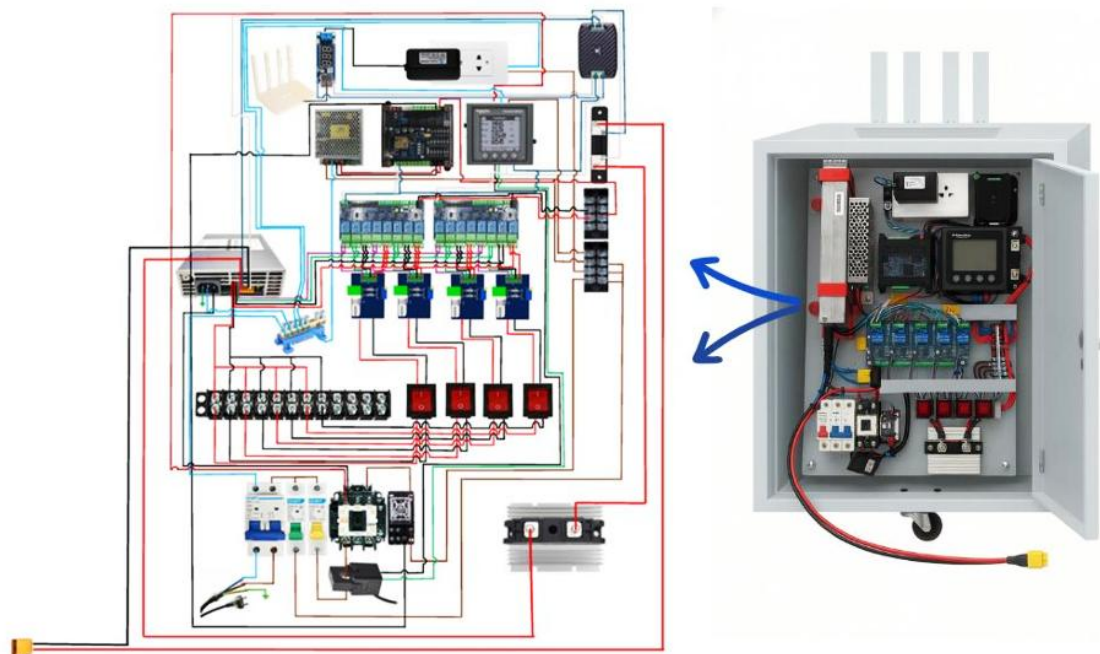
ระบบชาร์จที่พัฒนาขึ้นในงานวิจัยนี้ประกอบด้วยอุปกรณ์ด้านกำลังไฟฟ้า ระบบควบคุม ระบบสื่อสาร และระบบตรวจวัดพลังงาน ซึ่งอุปกรณ์ทั้งหมดถูกติดตั้งรวมภายในตู้ควบคุม เพื่อให้สามารถทำงานร่วมกันได้อย่างเป็นระบบ โดยเครื่องชาร์จแบบโปรแกรมได้ทำหน้าที่เป็นแหล่งจ่ายกำลังไฟฟ้าหลักสำหรับกระบวนการชาร์จแบบฟิกส์ CC-CV ขณะที่ MiniPLC-H3 ทำหน้าที่ควบคุมลำดับการทำงานของระบบ และ RS485 Relay Module ถูกใช้สำหรับเลือกชุดพารามิเตอร์การชาร์จที่กำหนดไว้ล่วงหน้า นอกจากนี้ ระบบยังประกอบด้วยอุปกรณ์ตรวจวัดพลังงานด้าน AC และ DC รวมถึงอุปกรณ์สื่อสารสำหรับเชื่อมโยงข้อมูลระหว่างระบบควบคุมและระบบติดตามข้อมูลแบบเรียลไทม์ โดยรายละเอียดของอุปกรณ์ที่ใช้ในการพัฒนาระบบชาร์จแสดงดังตารางที่ 3.1

ตารางที่ 3.1 อุปกรณ์ที่ใช้ในการพัฒนาระบบชาร์จ

รูปภาพอุปกรณ์	ชื่ออุปกรณ์-พิกัด	รูปภาพอุปกรณ์	ชื่ออุปกรณ์-พิกัด
	เครื่องชาร์จเดสก์ท็อป แบบ 0-120V/0-30A		Wi-Fi สำหรับการสื่อสาร MQTT
	รีเลย์ RS-485 จำนวน 8 Channel		เครื่องมือวัด AC Power Meter PM2230
	Mini PLC-H3 หรือ (redPLC/Node-RED)		เครื่องมือวัด DC Power Meter
	แบตเตอรี่แพ็ค NMC พิกัด 3.7V25Ah/Cell		อุปกรณ์ควบคุม Duty Cycle สำหรับ Fixed CC-CV

3.2.3 แผนภาพการเชื่อมต่อระบบไฟฟ้าและระบบควบคุม

การเชื่อมต่ออุปกรณ์ภายในตู้ชาร์จถูกออกแบบให้แยกส่วนระหว่างวงจรกำลังไฟฟ้า วงจรควบคุม และระบบสื่อสารอย่างชัดเจน เพื่อลดสัญญาณรบกวนและเพิ่มเสถียรภาพของระบบควบคุม โดยวงจรกำลังไฟฟ้าทำหน้าที่จ่ายพลังงานให้กับเครื่องชาร์จและแบตเตอรี่ ขณะที่วงจรควบคุมทำหน้าที่จัดลำดับการทำงานของระบบและเลือกพารามิเตอร์การชาร์จผ่าน Relay control และ Modbus RTU ส่วนระบบสื่อสารทำหน้าที่เชื่อมโยงข้อมูลระหว่างแบตเตอรี่ ระบบ Supervisory control และระบบติดตามข้อมูลแบบเรียลไทม์ นอกจากนี้ การจัดวางอุปกรณ์และการเดินสายภายในตู้ยังถูกออกแบบให้สามารถแยกเส้นทางของสายกำลังไฟฟ้าและสายสื่อสารออกจากกัน เพื่อลดผลกระทบจากสัญญาณรบกวนและเพิ่มความต่อเนื่องในการสื่อสารของระบบควบคุม ดังแสดงในรูปที่ 3.3



รูปที่ 3.3 แผนภาพการเชื่อมต่อระบบไฟฟ้าและระบบควบคุมภายในตู้ชาร์จ

3.3 การพัฒนาเครื่องชาร์จแบบฟิกส์ CC-CV สำหรับระบบเลือกโปรไฟล์การชาร์จ

เพื่อให้ระบบสามารถกำหนดพารามิเตอร์การชาร์จได้อย่างเป็นเชิงกำหนด (Deterministic) งานวิจัยนี้ได้พัฒนาเครื่องชาร์จแบบฟิกส์ CC-CV โดยประยุกต์ใช้เครื่องชาร์จแบบโปรแกรมได้ Delta 0–120 V, 0–30 A ร่วมกับวงจรควบคุมภายนอกที่ออกแบบขึ้นเอง แนวทางดังกล่าวแตกต่างจากการควบคุมผ่านซอฟต์แวร์ทั่วไป เนื่องจากงานวิจัยนี้ใช้วิธีการกำหนดค่าความต้านทานของวงจรแรงดัน (CV) และวงจรกระแส (CC) ผ่านฮาร์ดแวร์โดยตรง เพื่อบังคับให้เครื่องชาร์จทำงานตามค่าพารามิเตอร์ที่กำหนดไว้ล่วงหน้า โดยได้ทำการถอดวงจรปรับค่าแรงดันและกระแสจากหน้าเครื่องชาร์จออกมาพัฒนาเป็นบอร์ดควบคุมภายนอกสำหรับใช้งานร่วมกับระบบเลือกโปรไฟล์การชาร์จของงานวิจัยนี้

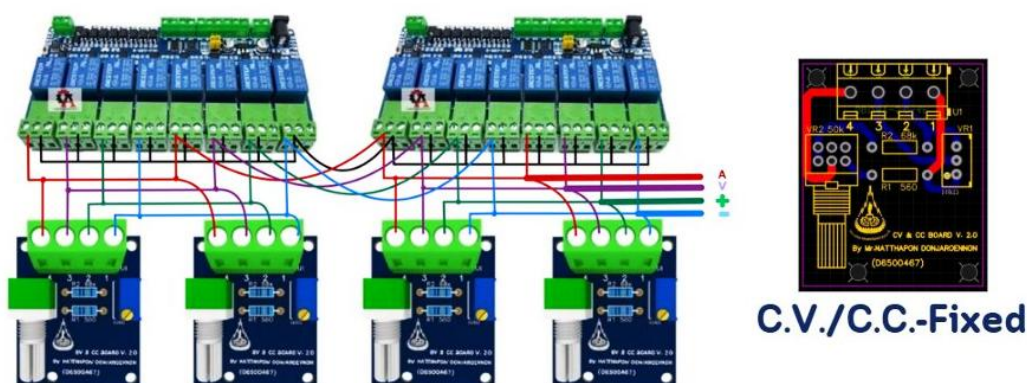
วงจรที่พัฒนาขึ้นถูกออกแบบให้รองรับการชาร์จแบตเตอรี่ NMC จำนวน 4 โปรไฟล์ ได้แก่ 24V 5A, 48V 5A, 60V 5A และ 72V 15A โดยแต่ละโปรไฟล์จะถูกกำหนดค่าความต้านทานอ้างอิงของแรงดันและกระแสไว้ล่วงหน้าบนแผงวงจรควบคุม เพื่อให้สามารถเลือกค่าพารามิเตอร์ของเครื่องชาร์จได้โดยไม่ต้องปรับตั้งใหม่ทุกครั้งที่เปลี่ยนแบตเตอรี่ หลักการดังกล่าวช่วยลดความผิดพลาดจากผู้ปฏิบัติงาน เพิ่มความต่อเนื่องของระบบ และทำให้พารามิเตอร์การชาร์จสามารถถูกบังคับใช้ได้จริงในระดับฮาร์ดแวร์ วงจรควบคุมแรงดันและกระแสเป็นดังรูปที่ 3.4



รูปที่ 3.4 วงจรควบคุมแรงดันและกระแสของเครื่องชาร์จแบบฟิกส์ CC-CV

จากรูปที่ 3.4 แสดงการพัฒนาวงจรควบคุมภายนอกสำหรับกำหนดค่าแรงดันและกระแสอ้างอิงของเครื่องชาร์จ Delta โดยใช้วงจรปรับค่า Duty Cycle ร่วมกับวงจรแบ่งแรงดันและวงจรจำกัดกระแส เพื่อสร้างพารามิเตอร์การชาร์จที่สอดคล้องกับโปรไฟล์ที่กำหนดไว้ล่วงหน้า นอกจากนี้ยังมีการออกแบบแผ่น PCB เพื่อให้สามารถติดตั้งวงจรควบคุมได้อย่างเป็นระบบและลดผลกระทบจากสัญญาณรบกวนทางไฟฟ้าในระหว่างการทำงานจริง

เพื่อให้ระบบสามารถเลือกโปรไฟล์การชาร์จได้โดยอัตโนมัติ งานวิจัยนี้ได้ประยุกต์ใช้ Relay Module ที่สื่อสารผ่าน Modbus RTU บนเครือข่าย RS485 สำหรับทำหน้าที่เลือกชุดค่าความต้านทานที่ถูกกำหนดไว้ล่วงหน้าบนบอร์ดควบคุมของแต่ละโปรไฟล์ โดยรีเลย์จะทำหน้าที่สลับเส้นทางของวงจรแรงดันและกระแสให้ตรงกับเงื่อนไขของแบตเตอรี่แต่ละระบบ ส่งผลให้เครื่องชาร์จสามารถเปลี่ยนค่าพารามิเตอร์การทำงานได้โดยอัตโนมัติในการเลือกโปรไฟล์การชาร์จภายใต้ Execution layer ของระบบควบคุม การเลือกโปรไฟล์การชาร์จแสดงดังรูปที่ 3.5



รูปที่ 3.5 การเลือกโปรไฟล์การชาร์จผ่าน Relay RS485 สำหรับระบบฟิกส์ CC-CV

จากรูปที่ 3.5 แสดงโครงสร้างการเชื่อมต่อระหว่าง Relay RS485 และบอร์ดควบคุมแรงดัน-กระแสของเครื่องชาร์จ โดยระบบจะเลือกชุดค่าความต้านทานที่สอดคล้องกับโปรไฟล์การชาร์จแต่ละชุดผ่านการสั่งงานรีเลย์ ทำให้สามารถกำหนดค่าการชาร์จของแบตเตอรี่แต่ละระดับแรงดันได้อย่างถูกต้องและเป็นเชิงกำหนด นอกจากนี้ แนวทางดังกล่าวยังช่วยให้ระบบสามารถใช้เครื่องชาร์จเพียงเครื่องเดียวรองรับแบตเตอรี่หลายรูปแบบได้ภายใต้โครงสร้างควบคุมเดียวกัน ลดความซับซ้อนของระบบและเพิ่มความยืดหยุ่นในการใช้งานจริง

3.3.1 การกำหนด CAN Identifier สำหรับแบตเตอรี่ NMC

ในการออกแบบระบบสื่อสารสำหรับการเลือกโปรไฟล์การชาร์จ งานวิจัยนี้กำหนดให้แบตเตอรี่แต่ละระบบมี CAN Identifier (CAN ID) เฉพาะ เพื่อใช้เป็นตัวระบุข้อมูลประจำแบตเตอรี่ในเครือข่าย CAN Bus โดยใช้ ESP32 เป็นหน่วยประมวลผลหลักร่วมกับโมดูล CAN Transceiver สำหรับส่งข้อมูลเข้าสู่ระบบควบคุม การกำหนด CAN ID ในลักษณะดังกล่าวช่วยให้ระบบสามารถระบุชนิดของแบตเตอรี่และเลือกโปรไฟล์การชาร์จที่เหมาะสมได้โดยอัตโนมัติ ลดความผิดพลาดจากการตั้งค่าด้วยมนุษย์ และเพิ่มความต่อเนื่องในการทำงานของระบบ

ในงานวิจัยนี้ได้ใช้แนวทาง Static Mapping โดยอาศัยค่า Chip ID หรือ MAC Address ของ ESP32 เป็นตัวกำหนด CAN ID ของอุปกรณ์แต่ละชุด เนื่องจาก Chip ID เป็นค่าประจำตัวระดับฮาร์ดแวร์ที่ไม่ซ้ำกัน ทำให้สามารถระบุแบตเตอรี่แต่ละระบบได้อย่างชัดเจน โดยระบบจะอ่านค่า Chip ID เมื่อเริ่มต้นการทำงาน จากนั้นจึงเปรียบเทียบกับตารางค่า CAN ID ที่ถูกสร้างไว้ล่วงหน้า ก่อนส่งข้อมูลเข้าสู่ระบบสื่อสาร CANopen เพื่อใช้ในการเลือกโปรไฟล์การชาร์จ ดังแสดงในตารางที่ 3.2

ตารางที่ 3.2 การกำหนด CAN ID สำหรับแบตเตอรี่แต่ละระบบ

CAN ID	ระบบแบตเตอรี่ NMC	โหมดโปรไฟล์การชาร์จ
0x500	24V 5A	Profile 1
0x600	48V 5A	Profile 2
0x700	60V 5A	Profile 3
0x800	72V 15A	Profile 4

ภายหลังจาก ESP32 อ่านค่า Chip ID แล้ว ระบบจะนำค่าดังกล่าวเข้าสู่กระบวนการเปรียบเทียบกับฐานข้อมูล CAN ID ที่กำหนดไว้ล่วงหน้า เพื่อเลือกหมายเลขประจำตัวของแบตเตอรี่ที่ถูกต้อง ก่อนส่งข้อมูลเข้าสู่เครือข่าย CAN Bus โดยตัวอย่างโค้ดที่ใช้สำหรับกำหนด CAN ID ของแต่ละระบบแสดงดังรูปที่ 3.6



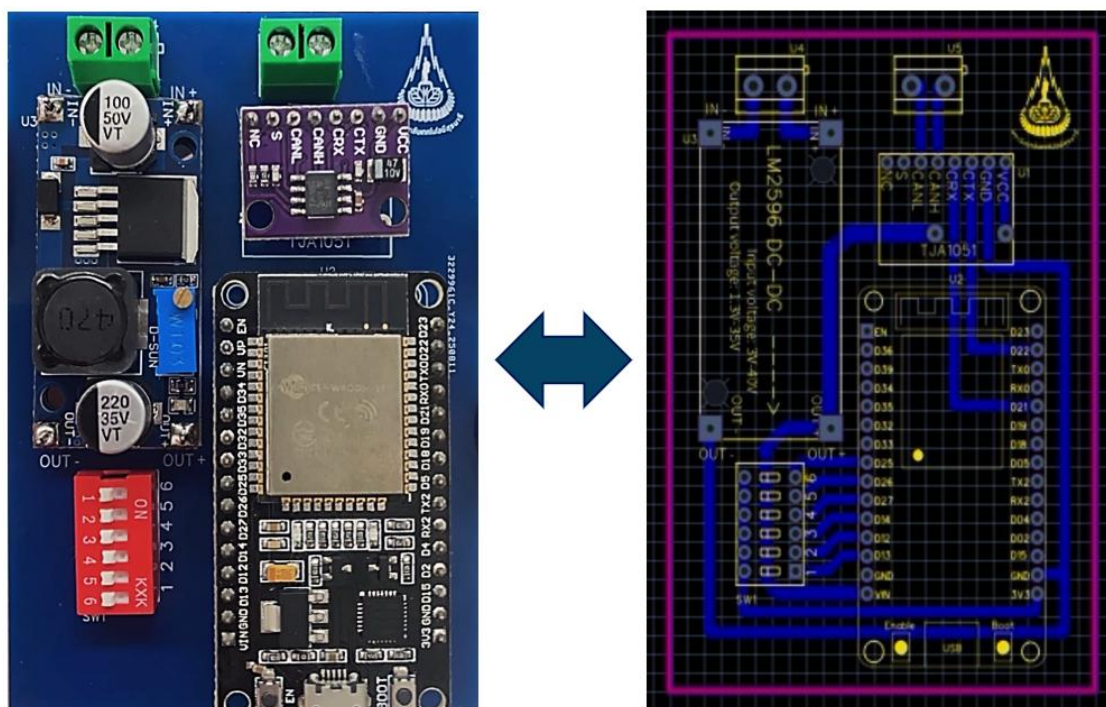
รูปที่ 3.6 การรับและส่ง CAN ID ระหว่างกล่องแบตเตอรี่ NMC และตู้ชาร์จผ่าน CANopen

จากโค้ดข้างต้นจะเห็นได้ว่าแต่ละค่า Chip ID ถูกเชื่อมโยงกับ CAN ID เฉพาะของระบบแบตเตอรี่แต่ละชุด ทำให้ระบบสามารถกำหนดตัวตนของแบตเตอรี่ได้ในระดับฮาร์ดแวร์โดยไม่ต้องพึ่งพาการตั้งค่าจากผู้ปฏิบัติงาน ส่งผลให้การเลือกโปรไฟล์การชาร์จมีความต่อเนื่องและลดความผิดพลาดจากการกำหนดค่าที่ไม่ถูกต้อง

3.3.2 การออกแบบบอร์ดสื่อสาร CANopen สำหรับเลือกโปรไฟล์การชาร์จ

งานวิจัยนี้ได้พัฒนาบอร์ดสื่อสาร CANopen สำหรับติดตั้งร่วมกับแบตเตอรี่แต่ละระบบ เพื่อใช้ในการส่ง CAN Identifier ไปยังตู้ชาร์จสำหรับเลือกโปรไฟล์การชาร์จแบบอัตโนมัติ โดยบอร์ดดังกล่าวประกอบด้วย ESP32 สำหรับประมวลผลข้อมูล โมดูล CAN Transceiver รุ่น TJA1051 สำหรับเชื่อมต่อกับเครือข่าย CAN Bus และวงจร DC/DC Converter สำหรับจ่ายพลังงานให้กับอุปกรณ์ภายในบอร์ด

นอกจากการกำหนด CAN ID แล้ว บอร์ดสื่อสารดังรูป 3.7 ยังถูกออกแบบให้สามารถกำหนดค่าพารามิเตอร์แรงดันและกระแสของแบตเตอรี่ผ่าน Diff Switch ซึ่งใช้หลักการ Binary Encoding ในการแปลงสถานะของสวิตช์ให้เป็นค่าพารามิเตอร์ที่ต้องการใช้งานจริง โดยค่าที่ได้จะถูกนำไปประมวลผลร่วมกับ CAN ID เพื่อใช้ในการเลือกโปรไฟล์การชาร์จในระดับ Supervisory layer



รูปที่ 3.7 บอร์ดสื่อสาร CANOpen สำหรับระบบแบตเตอรี่และเลือกโปรไฟล์การชาร์จ

จากรูปที่ 3.7 จะเห็นได้ว่าบอร์ดสื่อสารประกอบด้วยวงจร DC to DC Converter สำหรับจ่ายแรงดันให้กับ ESP32 และโมดูล CAN Transceiver เพื่อเชื่อมต่อกับเครือข่าย CAN Bus ขณะที่ Diff Switch ถูกใช้เป็นอุปกรณ์กำหนดค่าพารามิเตอร์ของแบตเตอรี่ในระดับฮาร์ดแวร์ ส่งผลให้ระบบสามารถเลือกโปรไฟล์การชาร์จได้อย่างเป็นเชิงกำหนดและลดภาระของการประมวลผลในระดับซอฟต์แวร์

3.3.3 การกำหนดพารามิเตอร์ผ่าน Diff Switch

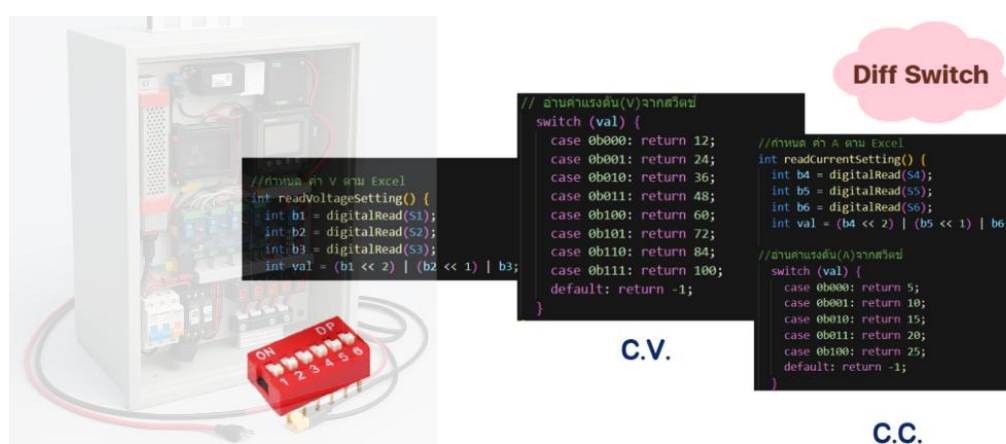
เพื่อให้ระบบสามารถเลือกโปรไฟล์การชาร์จได้ในระดับฮาร์ดแวร์ งานวิจัยนี้ได้ออกแบบระบบ Diff Switch สำหรับกำหนดค่าพารามิเตอร์ของแรงดันและกระแสผ่าน GPIO ของ ESP32 โดยใช้หลักการ Binary Encoding ในการแปลงสถานะของสวิตช์ให้เป็นค่าพารามิเตอร์ที่ต้องการใช้งานจริง ซึ่งช่วยให้ระบบสามารถกำหนดค่าการชาร์จได้อย่างรวดเร็ว มีความเสถียร และลดความผิดพลาดจากการตั้งค่าผ่านซอฟต์แวร์

ในส่วนของการกำหนดแรงดันไฟฟ้า ใช้ Diff Switch จำนวน 3 บิต เชื่อมต่อกับ GPIO25, GPIO26 และ GPIO27 ของ ESP32 ขณะที่การกำหนดค่ากระแสไฟฟ้าใช้ Diff Switch อีก 3 บิต เชื่อมต่อกับ GPIO14, GPIO12 และ GPIO13 ตามลำดับ โดยทุกขาถูกกำหนดให้ทำงานในโหมด INPUT_PULLDOWN เพื่อป้องกันสัญญาณรบกวนในระหว่างการอ่านข้อมูล ดังแสดงในตารางที่ 3.3

ตารางที่ 3.3 การเชื่อมต่อ Diff Switch กับ GPIO ของ ESP32

Diff Switch	ขา GPIO จาก ESP32
S1 - Voltage	GPIO25
S2 - Voltage	GPIO26
S3 - Voltage	GPIO27
S4 - Ampere	GPIO14
S5 - Ampere	GPIO12
S6 - Ampere	GPIO13

ภายหลังจากกำหนดตำแหน่งของ Diff Switch แล้ว ระบบจะทำการอ่านสถานะลอจิกของแต่ละ GPIO และนำค่าไบนารีที่ได้เข้าสู่กระบวนการประมวลผลเพื่อแปลงเป็นค่าระดับแรงดันและกระแสที่ต้องการใช้งานจริง โดยตัวอย่างโค้ดสำหรับอ่านค่าระดับแรงดันแสดงดังรูปที่ 3.8



รูปที่ 3.8 การกำหนดค่าแรงดันและกระแสของระบบฟิกส์ CC-CV ผ่าน Diff Switch

จากโค้ดข้างต้น ระบบจะอ่านสถานะของ Diff Switch ทั้ง 3 บิตและนำมาเรียงในรูปแบบไบนารีก่อนแปลงเป็นค่าระดับแรงดันที่ต้องการใช้งานจริง เช่น ค่า 0b000 จะถูกกำหนดให้เป็นระบบ 24V ขณะที่ค่า 0b100 จะถูกกำหนดให้เป็นระบบ 72V ส่งผลให้ระบบสามารถเลือกค่าพารามิเตอร์แรงดันได้โดยไม่ต้องแก้ไขโปรแกรมในระหว่างการใช้งานจริง ในส่วนของการกำหนดค่ากระแสไฟฟ้า ระบบใช้หลักการเดียวกันในการอ่านสถานะของ Diff Switch ผ่าน GPIO14, GPIO12 และ GPIO13 ก่อนแปลงเป็นค่ากระแสที่กำหนดไว้ล่วงหน้า โดยตัวอย่างโค้ดสำหรับอ่านค่ากระแส

จากโค้ดข้างต้น ระบบจะทำการแปลงสถานะไบนารีของ Diff Switch ให้เป็นค่ากระแสอ้างอิงของเครื่องชาร์จ ซึ่งช่วยให้สามารถเลือกค่าพารามิเตอร์ของระบบชาร์จได้ในระดับฮาร์ดแวร์และลดความซับซ้อนของการประมวลผลในระดับซอฟต์แวร์

3.3.4 กลไกการเลือกโปรไฟล์การชาร์จในระดับระบบ

ภายหลังจากระบบกำหนด CAN ID และอ่านค่าพารามิเตอร์จาก Diff Switch แล้ว ข้อมูลทั้งหมดจะถูกส่งเข้าสู่ Supervisory layer ผ่านโปรโตคอล MQTT เพื่อใช้ในการประมวลผลและเลือกโปรไฟล์การชาร์จที่เหมาะสมกับแบตเตอรี่แต่ละระบบ โดย Node-RED จะทำหน้าที่เป็นศูนย์กลางในการจัดการข้อมูลและกำหนดลำดับการทำงานของระบบ ก่อนส่งคำสั่งผ่าน Modbus RTU ไปยัง Relay RS485 สำหรับเลือกชุดค่าความต้านทานของวงจรควบคุมแรงดันและกระแสใน Execution layer

ในระดับ Execution layer รีเลย์แต่ละช่องสัญญาณจะถูกกำหนดให้เชื่อมต่อกับชุดค่าความต้านทานเฉพาะของแต่ละโปรไฟล์การชาร์จ ซึ่งค่าความต้านทานดังกล่าวถูกปรับตั้งไว้ล่วงหน้าบนบอร์ดควบคุมแรงดันและกระแสของเครื่องชาร์จแบบฟิกส์ CC-CV เมื่อระบบได้รับข้อมูล CAN ID และตรวจสอบความถูกต้องของโปรไฟล์แล้ว ระบบจะสั่งงาน Relay RS485 ให้เลือกเส้นทางของวงจรควบคุมที่ตรงกับพารามิเตอร์ของแบตเตอรี่แต่ละระบบ ส่งผลให้เครื่องชาร์จสามารถกำหนดค่าแรงดันและกระแสได้โดยอัตโนมัติ

นอกจากนี้ งานวิจัยนี้ยังออกแบบการเชื่อมโยงระหว่าง CAN ID, Modbus Slave ID และช่องสัญญาณรีเลย์ให้ทำงานสัมพันธ์กันในระดับระบบ เพื่อให้สามารถเลือกโปรไฟล์การชาร์จได้อย่างต่อเนื่องและเป็นเชิงกำหนด โดยรายละเอียดของการเชื่อมโยงระหว่าง CAN ID และระบบรีเลย์แสดงดังตารางที่ 3.4

ตารางที่ 3.4 การเชื่อมโยง CAN ID กับระบบรีเลย์ควบคุม

CAN ID	Slave ID	Address (Relay Channel)	Relay RS485
0x500	1	0-3	Relay 1
0x600	1	4-7	Relay 1
0x700	2	0-3	Relay 2
0x800	2	4-7	Relay 2

จากตารางที่ 3.4 จะเห็นได้ว่าแต่ละ CAN ID ถูกเชื่อมโยงกับ Modbus Slave ID และตำแหน่งรีเลย์ที่แตกต่างกัน ทำให้ระบบสามารถกำหนดโปรไฟล์การชาร์จได้อย่างชัดเจนและลดความซับซ้อนของการควบคุมภายใน execution layer โดยแนวทางดังกล่าวช่วยให้ระบบสามารถใช้

เครื่องชาร์จเพียงชุดเดียวรองรับแบตเตอรี่หลายระดับแรงดันได้ภายใต้สถาปัตยกรรมเดียวกันลำดับการทำงานของระบบเลือกโปรไฟล์การชาร์จสามารถสรุปได้ดังนี้

- 1) แบตเตอรี่เชื่อมต่อเข้าสู่ระบบ
- 2) ESP32 อ่านค่า Chip ID และสถานะของ Diff Switch
- 3) ระบบกำหนด CAN ID และค่าพารามิเตอร์ของแบตเตอรี่
- 4) ส่งข้อมูลผ่าน CANopen ไปยังตู้ชาร์จ
- 5) Node-RED รับข้อมูลผ่าน MQTT และประมวลผลโปรไฟล์
- 6) ส่งคำสั่งผ่าน Modbus RTU ไปยัง Relay RS485
- 7) รีเลย์เลือกชุดค่าความต้านทานของวงจร CC-CV
- 8) เครื่องชาร์จเริ่มทำงานตามโปรไฟล์ที่กำหนดไว้ล่วงหน้า

กลไกดังกล่าวช่วยให้ระบบสามารถเลือกโปรไฟล์การชาร์จได้อย่างเป็นอัตโนมัติ มีความเสถียร และลดความผิดพลาดจากการกำหนดค่าด้วยมนุษย์ เนื่องจากค่าพารามิเตอร์ทั้งหมดถูกกำหนดผ่านโครงสร้างฮาร์ดแวร์ที่ออกแบบไว้ล่วงหน้า ไม่ได้อาศัยการปรับค่าผ่านซอฟต์แวร์ระหว่างกระบวนการทำงาน ส่งผลให้ Execution layer สามารถทำงานได้อย่างต่อเนื่องและมีความน่าเชื่อถือสูงสำหรับการประยุกต์ใช้งานกับระบบชาร์จแบตเตอรี่ NMC หลายระดับแรงดันภายในระบบเดียวกัน

3.4 การสื่อสารข้อมูลระหว่างแบตเตอรี่และตู้ชาร์จผ่าน CANopen และ MQTT

ระบบสื่อสารที่พัฒนาขึ้นในงานวิจัยนี้ถูกออกแบบเพื่อรองรับการเลือกโปรไฟล์การชาร์จแบตเตอรี่ NMC แรงดันต่ำแบบอัตโนมัติ โดยอาศัยการสื่อสารระหว่างกล่องแบตเตอรี่และตู้ชาร์จผ่านโปรโตคอล CANopen ก่อนส่งข้อมูลเข้าสู่ Supervisory layer ผ่าน MQTT เพื่อใช้ในการติดตามและประมวลผลข้อมูลแบบเรียลไทม์ ภายในระบบ กล่องแบตเตอรี่จะทำหน้าที่ส่ง CAN ID พร้อมข้อมูลแรงดันและกระแสเข้าสู่ตู้ชาร์จ ขณะที่ตู้ชาร์จจะทำหน้าที่รับข้อมูล วิเคราะห์โปรไฟล์การชาร์จ และส่งข้อมูลไปยัง Node-RED ผ่านเครือข่าย Wi-Fi เพื่อใช้ในการติดตามสถานะของระบบและควบคุม Execution layer ของเครื่องชาร์จแบบฟิกส์ CC-CV โดยแนวทางดังกล่าวช่วยให้ระบบสามารถเลือกโปรไฟล์การชาร์จได้โดยอัตโนมัติ ลดความผิดพลาดจากผู้ปฏิบัติงาน และเพิ่มความน่าเชื่อถือของระบบสื่อสารในระดับปฏิบัติการ

3.4.1 การส่งข้อมูล CANopen จากกล่องแบตเตอรี่

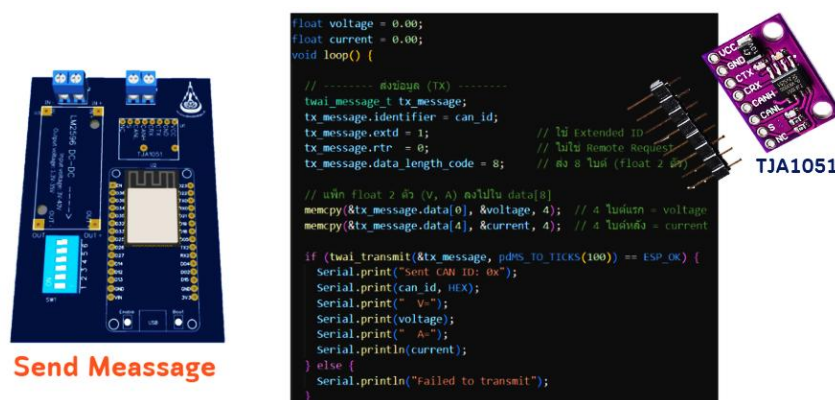
ในการเริ่มต้นกระบวนการชาร์จ กล่องแบตเตอรี่จะทำหน้าที่ส่งข้อมูลประจำตัวของแบตเตอรี่เข้าสู่ตู้ชาร์จทันทีเมื่อมีการเชื่อมต่อสายชาร์จ โดยงานวิจัยนี้ใช้ ESP32 ร่วมกับโมดูล CAN Transceiver รุ่น TJA1051 สำหรับพัฒนาระบบสื่อสารผ่านเครือข่าย CAN Bus ภายใต้มาตรฐาน CAN 2.0B เพื่อใช้เป็น Communication layer สำหรับการระบุโปรไฟล์การชาร์จของแบตเตอรี่

NMC แรงดันต่ำแต่ละระบบ ทั้งนี้ CAN Identifier (CAN ID) ของแบตเตอรี่แต่ละชุดถูกกำหนดไว้ล่วงหน้าเพื่อใช้เป็นตัวระบุโปรไฟล์การชาร์จ ดังที่ได้กล่าวไว้ในตารางที่ 3.2 ซึ่งช่วยให้ผู้ชาร์จสามารถตรวจสอบชนิดของแบตเตอรี่และเลือกโปรไฟล์การชาร์จที่เหมาะสมได้โดยอัตโนมัติ

ระบบส่งข้อมูล CANopen ที่พัฒนาขึ้นในงานวิจัยนี้แสดงดังรูปที่ 3.8 โดยประกอบด้วย ESP32 สำหรับประมวลผลข้อมูล โมดูล TJA1051 สำหรับเชื่อมต่อกับเครือข่าย CAN Bus และวงจร DC/DC Converter สำหรับจ่ายพลังงานให้กับระบบสื่อสาร ภายใน ESP32 จะทำหน้าที่สร้าง CAN Frame พร้อมกำหนด CAN Identifier และบันทึกข้อมูลแรงดัน-กระแสลงใน payload ของ CAN message ก่อนส่งข้อมูลไปยังตู้ชาร์จผ่านเครือข่าย CAN Bus

จากรูปที่ 3.9 จะเห็นได้ว่าบอร์ดส่งข้อมูลถูกออกแบบให้สามารถเชื่อมต่อกับระบบแบตเตอรี่และเครือข่าย CAN Bus ได้โดยตรงผ่านโมดูล TJA1051 ซึ่งทำหน้าที่เป็น Physical interface ระหว่าง ESP32 และสายสื่อสาร CAN High / CAN Low ขณะที่ ESP32 จะทำหน้าที่ประมวลผลข้อมูลและสร้าง CAN Frame สำหรับส่งข้อมูลเข้าสู่ระบบชาร์จ ทั้งนี้ การเลือกใช้ TJA1051 ช่วยเพิ่มเสถียรภาพของระบบสื่อสาร ลดผลกระทบจากสัญญาณรบกวน และรองรับการทำงานของระบบในสภาพแวดล้อมทางไฟฟ้ากำลัง

ภายในระบบ CANopen ข้อมูลแรงดันและกระแสของแบตเตอรี่จะถูกจัดเก็บอยู่ภายใน payload ของ CAN message โดยงานวิจัยนี้กำหนดให้ข้อมูลแรงดันและกระแสถูกส่งในรูปแบบ floating-point ขนาด 4 ไบต์ต่อหนึ่งพารามิเตอร์ ส่งผลให้ข้อมูลทั้งหมดภายใน CAN payload มีขนาดรวม 8 ไบต์ ซึ่งสอดคล้องกับขนาดมาตรฐานของ CAN Data Frame ภายใต้ CAN 2.0B โดยข้อมูลแรงดันจะถูกจัดเก็บในตำแหน่ง Byte 0 ถึง Byte 3 ขณะที่ข้อมูลกระแสจะถูกจัดเก็บในตำแหน่ง Byte 4 ถึง Byte 7 ของ Payload ทั้งนี้ การกำหนดรูปแบบข้อมูลในลักษณะดังกล่าวช่วยให้ตู้ชาร์จสามารถแยกข้อมูลแรงดันและกระแสออกจาก CAN message ได้อย่างถูกต้อง และลดความซับซ้อนของกระบวนการประมวลผลข้อมูลภายใน Supervisory layer



รูปที่ 3.9 การส่งข้อมูล CANopen จากกล่องแบตเตอรี่ผ่าน ESP32 ผ่านสาย CAN-high/CAN-low

ภายใน ESP32 งานวิจัยนี้ใช้ฟังก์ชัน `memcpy()` สำหรับคัดลอกข้อมูลชนิด floating-point ลงใน data field ของ CAN Frame ก่อนส่งข้อมูลผ่านเครือข่าย CAN Bus ไปยังตู้ชาร์จ โดยแนวทางดังกล่าวช่วยลดปัญหาการแปลงชนิดข้อมูลระหว่างกระบวนการรับส่งข้อมูล และช่วยให้สามารถส่งค่าพารามิเตอร์ของแบตเตอรี่ได้อย่างถูกต้องภายใต้ Communication layer ของระบบชาร์จ

จากโค้ดข้างต้น ESP32 จะทำการสร้าง CAN message ผ่านโครงสร้าง `twai_message_t` พร้อมกำหนด CAN Identifier ของแบตเตอรี่แต่ละระบบ ก่อนกำหนด payload ของข้อมูลแรงดันและกระแสผ่านฟังก์ชัน `memcpy()` และส่งข้อมูลเข้าสู่เครือข่าย CAN Bus ผ่านฟังก์ชัน `twai_transmit()` โดยภายใน CAN message จะประกอบด้วย Identifier, data length code และ Payload ของข้อมูลแรงดัน-กระแสที่ถูกส่งไปยังตู้ชาร์จแบบ Real-time ส่งผลให้ระบบสามารถระบุโปรไฟล์การชาร์จและติดตามค่าพารามิเตอร์ของแบตเตอรี่ได้อย่างต่อเนื่องระหว่างกระบวนการทำงาน นอกจากนี้ การใช้ CANopen ภายใน Communication layer ยังช่วยให้ระบบสามารถรองรับการทำงานร่วมกับ Execution layer ของเครื่องชาร์จแบบฟิกส์ CC-CV ได้อย่างมีประสิทธิภาพ เนื่องจากข้อมูลที่ถูกส่งจากกล่องแบตเตอรี่สามารถถูกนำไปใช้ในการเลือกโปรไฟล์การชาร์จและควบคุม Relay RS485 ผ่าน Modbus RTU ได้โดยอัตโนมัติ ส่งผลให้ระบบสามารถเลือกค่าพารามิเตอร์ของเครื่องชาร์จได้อย่างเป็นเชิงกำหนด ลดความผิดพลาดจากการตั้งค่าด้วยมนุษย์ และเพิ่มความน่าเชื่อถือของระบบชาร์จในระดับปฏิบัติการจริง

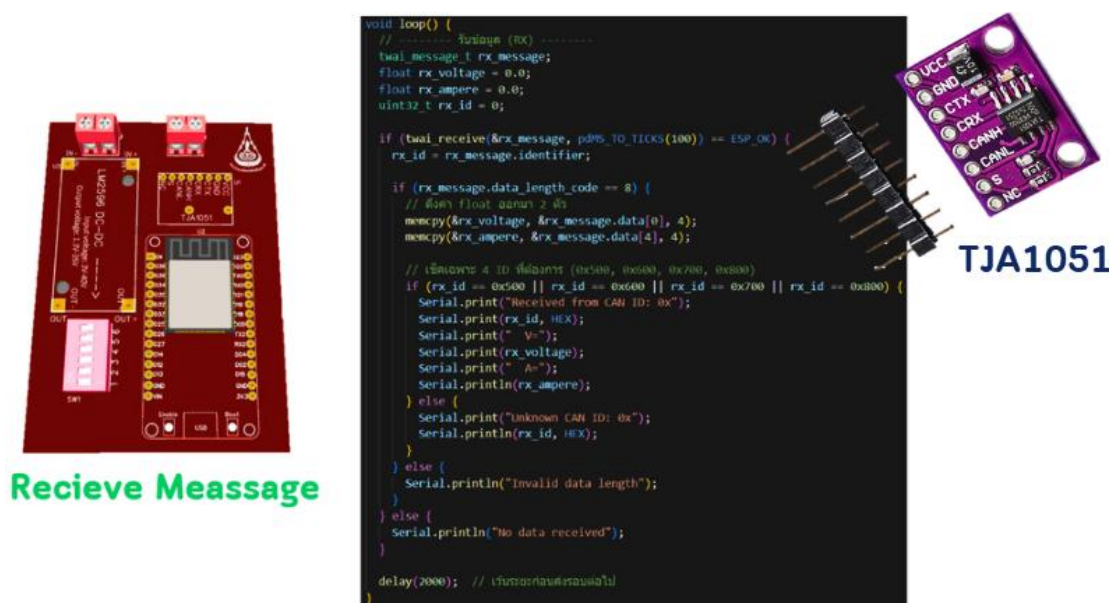
3.4.2 การรับข้อมูล CANopen ที่ตู้ชาร์จ

ภายหลังจากกล่องแบตเตอรี่ส่งข้อมูลผ่านเครือข่าย CAN Bus แล้ว ตู้ชาร์จจะทำหน้าที่รับข้อมูล CANopen ผ่าน ESP32 และโมดูล CAN Transceiver รุ่น TJA1051 เพื่อใช้ในการระบุโปรไฟล์การชาร์จและประมวลผลข้อมูลของแบตเตอรี่ก่อนส่งเข้าสู่ Supervisory layer ของระบบ โดยภายในตู้ชาร์จ ESP32 จะทำงานในโหมด CAN Receive Mode เพื่อรอรับข้อมูลจากเครือข่าย CAN Bus ตลอดเวลา เมื่อมีการเชื่อมต่อแบตเตอรี่และเริ่มส่ง CAN message ระบบจะทำการตรวจสอบ CAN Identifier และ Payload ของข้อมูลเพื่อใช้ในการเลือกโปรไฟล์การชาร์จที่เหมาะสมกับแบตเตอรี่แต่ละระบบ

โครงสร้างของระบบรับข้อมูล CANopen ภายในตู้ชาร์จแสดงดังรูปที่ 3.10 โดยประกอบด้วย ESP32 สำหรับประมวลผลข้อมูล โมดูล TJA1051 สำหรับเชื่อมต่อกับเครือข่าย CAN Bus และระบบ Supervisory layer สำหรับประมวลผลข้อมูลก่อนส่งต่อไปยัง Node-RED ผ่าน MQTT ทั้งนี้ TJA1051 ทำหน้าที่เป็น Physical interface สำหรับเชื่อมต่อสัญญาณ CAN High และ CAN Low เข้าสู่ ESP32 เพื่อให้ระบบสามารถรับข้อมูลจากเครือข่าย CAN Bus ได้อย่างมีประสิทธิภาพ และทนทานต่อสัญญาณรบกวนในระบบกำลังไฟฟ้า

จากรูปที่ 3.10 จะเห็นได้ว่าตู้ชาร์จทำหน้าที่รับข้อมูล CANopen ผ่านโมดูล TJA1051 ก่อนนำข้อมูลเข้าสู่ ESP32 เพื่อประมวลผล CAN Identifier และข้อมูลแรงดัน-กระแสของแบตเตอรี่ โดยข้อมูลดังกล่าวจะถูกใช้สำหรับระบุโปรไฟล์การชาร์จและส่งต่อไปยัง Supervisory layer ผ่าน MQTT เพื่อใช้ในการติดตามข้อมูลแบบเรียลไทม์และควบคุม Execution layer ของระบบชาร์จ

ภายในระบบรับข้อมูล CANopen งานวิจัยนี้กำหนดให้ ESP32 ตรวจสอบข้อมูล CAN Frame ที่ได้รับจากเครือข่าย CAN Bus ผ่านฟังก์ชัน twai_receive() โดยเมื่อระบบได้รับ CAN message แล้ว ESP32 จะทำการอ่านค่า Identifier และตรวจสอบขนาดของ payload ผ่าน Data length code ก่อนแยกข้อมูลแรงดันและกระแสออกจาก Payload ของ CAN message ผ่านฟังก์ชัน memcpy() ทั้งนี้ การตรวจสอบ Data length code ช่วยลดความผิดพลาดจากการรับข้อมูลที่ไม่สมบูรณ์ และเพิ่มความน่าเชื่อถือของ Communication layer ภายในระบบชาร์จ



รูปที่ 3.10 การรับข้อมูล CANopen ที่ตู้ชาร์จผ่าน ESP32 ผ่านสาย CAN-high/CAN-low

ในการประมวลผลข้อมูล CANopen ภายในตู้ชาร์จ ค่า CAN Identifier ที่ได้รับจะถูกนำไปเปรียบเทียบกับฐานข้อมูลโปรไฟล์การชาร์จของแบตเตอรี่แต่ละระบบ ดังที่ได้กล่าวไว้ในตารางที่ 3.2 เพื่อใช้ในการกำหนดหมายเลขโปรไฟล์การชาร์จของระบบ ขณะที่ข้อมูลแรงดันและกระแสจะถูกใช้สำหรับติดตามสถานะของแบตเตอรี่และส่งต่อไปยัง Supervisory layer ผ่าน MQTT สำหรับการแสดงผลและจัดเก็บข้อมูลแบบเรียลไทม์

จากโค้ดข้างต้น ESP32 ภายในตู้ชาร์จจะทำการตรวจสอบ CAN message ที่ได้รับจากเครือข่าย CAN Bus ผ่านฟังก์ชัน `twai_receive()` ก่อนอ่านค่า CAN Identifier และตรวจสอบขนาดของ Payload ผ่าน Data length code หาก Payload มีขนาด 8 ไบต์ตามที่กำหนด ระบบจะใช้ฟังก์ชัน `memcpy()` สำหรับแยกข้อมูลแรงดันและกระแสออกจาก Payload ของ CAN message ก่อนนำข้อมูลไปใช้สำหรับระบุโปรไฟล์การชาร์จและติดตามสถานะของแบตเตอรี่ นอกจากนี้ ภายใน Execution layer ของระบบ ข้อมูล CAN ID ที่ได้รับจากแบตเตอรี่ยังถูกใช้สำหรับเชื่อมโยงไปยังระบบควบคุม Relay RS485 ผ่าน Modbus RTU เพื่อเลือกชุดค่าความต้านทานของวงจรควบคุม CC-CV ที่สอดคล้องกับโปรไฟล์การชาร์จของแบตเตอรี่แต่ละระบบ ส่งผลให้ระบบสามารถเลือกค่าพารามิเตอร์ของเครื่องชาร์จได้อย่างเป็นเชิงกำหนดโดยไม่ต้องอาศัยการตั้งค่าจากผู้ปฏิบัติงาน ทั้งนี้ แนวทางดังกล่าวช่วยเพิ่มความต่อเนื่องในการทำงานของระบบ ลดความผิดพลาดจากการเลือกโปรไฟล์การชาร์จ และเพิ่มความน่าเชื่อถือของ Execution layer สำหรับการประยุกต์ใช้งานกับระบบชาร์จแบตเตอรี่ NMC หลายระดับแรงดันภายในระบบเดียวกัน

3.4.3 การส่งข้อมูล MQTT ไปยัง Node-RED

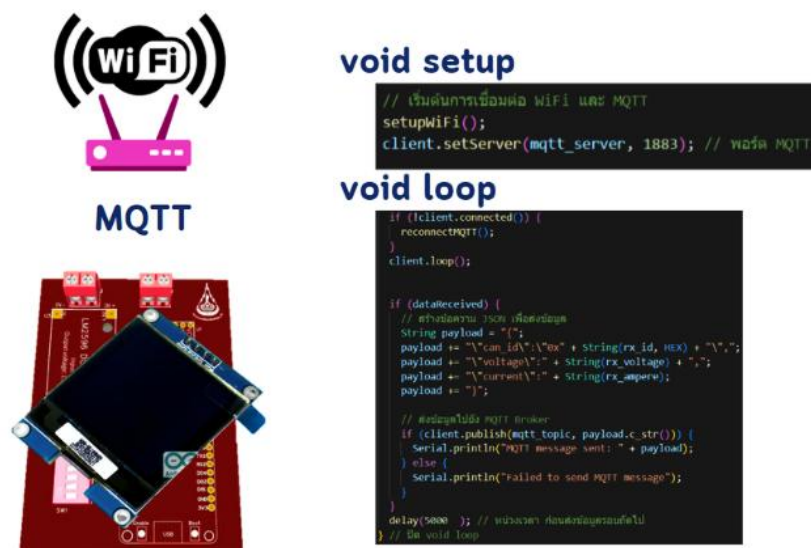
ภายหลังจากตู้ชาร์จได้รับข้อมูล CANopen และประมวลผลข้อมูลของแบตเตอรี่แล้ว ระบบจะส่งข้อมูล CAN ID, Voltage และ Current ไปยัง Node-RED ผ่านโปรโตคอล MQTT เพื่อใช้ในการติดตามข้อมูลแบบเรียลไทม์และจัดการ Supervisory control ของระบบชาร์จ โดย MQTT ถูกเลือกใช้งานเนื่องจากเป็นโปรโตคอลแบบ Publish/Subscribe ที่รองรับการสื่อสารผ่านเครือข่าย Wi-Fi มีขนาดข้อมูลต่ำ และเหมาะสมกับระบบ IoT ที่ต้องการรับส่งข้อมูลอย่างต่อเนื่องภายใต้ Distributed control architecture

ภายในระบบตู้ชาร์จจะทำหน้าที่เป็น MQTT Client สำหรับเชื่อมต่อกับ MQTT Broker ผ่านเครือข่าย Wi-Fi ก่อน publish ข้อมูลจากระบบ CANopen ไปยัง Supervisory layer ขณะที่ Node-RED จะทำหน้าที่ Subscribe ข้อมูลจาก MQTT Broker เพื่อใช้ในการแสดงผลข้อมูลแบบเรียลไทม์ วิเคราะห์สถานะของระบบ และควบคุม Execution layer ของเครื่องชาร์จแบบฟิกส์ CC-CV ต่อไป โดยแนวทางดังกล่าวช่วยให้ Communication layer และ Supervisory layer สามารถทำงานร่วมกันได้อย่างต่อเนื่องภายใต้ระบบสื่อสารเดียวกัน

โครงสร้างของระบบส่งข้อมูล MQTT จากตู้ชาร์จไปยัง Node-RED แสดงดังรูปที่ 3.10 โดย ESP32 ภายในตู้ชาร์จจะทำหน้าที่เชื่อมต่อกับเครือข่าย Wi-Fi และสร้างข้อมูลในรูปแบบ JSON payload ก่อนส่งข้อมูลผ่าน MQTT Broker ไปยัง Node-RED เพื่อใช้ในการติดตามสถานะของแบตเตอรี่และระบบชาร์จแบบเรียลไทม์

จากรูปที่ 3.11 จะเห็นได้ว่าตู้ชาร์จทำหน้าที่ Publish ข้อมูลจากระบบ CANopen ไปยัง MQTT Broker ผ่านเครือข่าย Wi-Fi ก่อนส่งต่อไปยัง Node-RED เพื่อใช้ในการแสดงผลและ

ติดตามสถานะของระบบแบบเรียลไทม์ โดยข้อมูลที่ถูกส่งประกอบด้วย CAN ID ค่าแรงดัน และค่ากระแสของแบตเตอรี่แต่ละระบบ ซึ่งถูกจัดเก็บในรูปแบบ JSON payload เพื่อให้ง่ายต่อการประมวลผลภายใน Supervisory layer



รูปที่ 3.11 การส่งข้อมูล CAN ID, Voltage และ Current ไปยัง Node-RED ผ่าน MQTT

ในงานวิจัยนี้ ESP32 ภายในตู้ชาร์จจะทำหน้าที่เชื่อมต่อกับ MQTT Broker ผ่านไลบรารี PubSubClient โดยระบบจะทำการตรวจสอบสถานะการเชื่อมต่อกับ MQTT Broker อย่างต่อเนื่อง หากการเชื่อมต่อถูกตัดขาด ระบบจะทำการ reconnect อัตโนมัติเพื่อรักษาความต่อเนื่องของการสื่อสารข้อมูลระหว่าง communication layer และ supervisory layer ทั้งนี้ การออกแบบในลักษณะดังกล่าวช่วยเพิ่มเสถียรภาพของระบบสื่อสารและลดโอกาสการสูญหายของข้อมูลระหว่างกระบวนการชาร์จ ข้อมูลที่ถูกส่งผ่าน MQTT จะอยู่ในรูปแบบ JSON payload ซึ่งประกอบด้วยค่า CAN ID ค่าแรงดัน และค่ากระแสของแบตเตอรี่ โดย CAN ID จะถูกใช้เป็นตัวระบุโปรไฟล์การชาร์จขณะที่ข้อมูลแรงดันและกระแสจะถูกใช้สำหรับติดตามสถานะของระบบและประเมินสมรรถนะของกระบวนการชาร์จในระหว่างการทดลอง ทั้งนี้ งานวิจัยนี้กำหนดให้ Node-RED subscribe ข้อมูลผ่าน MQTT Topic ที่ถูกกำหนดไว้ล่วงหน้า เพื่อให้ supervisory layer สามารถแยกประเภทข้อมูลและประมวลผลได้อย่างถูกต้อง

จาก JSON payload ข้างต้น จะเห็นได้ว่าข้อมูล CAN ID, Voltage และ Current ถูกจัดเก็บในรูปแบบ key-value structure ซึ่งช่วยให้ Node-RED สามารถอ่านข้อมูลและแยกพารามิเตอร์ต่าง ๆ ออกจาก payload ได้อย่างง่ายดาย นอกจากนี้ การใช้ JSON ยังช่วยให้ระบบสามารถขยายโครงสร้างข้อมูลเพิ่มเติมในอนาคตได้โดยไม่กระทบต่อ communication layer เดิม

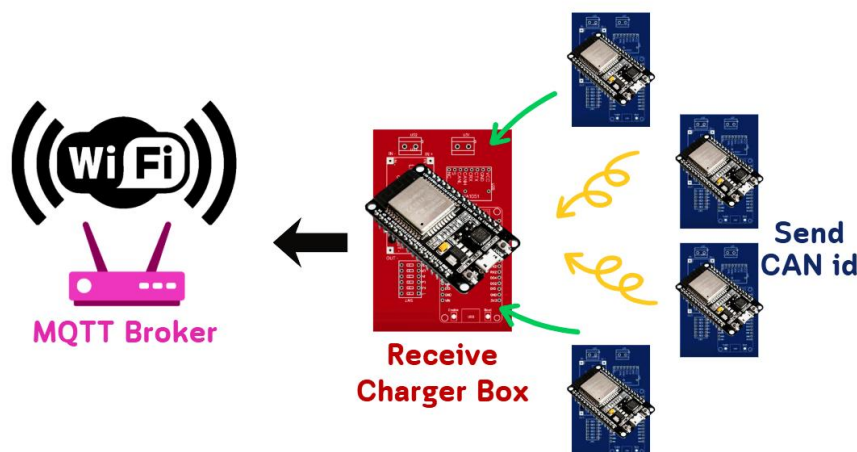
ภายใน Node-RED ระบบจะทำการ subscribe ข้อมูลจาก MQTT Broker ผ่าน topic ที่กำหนดไว้ล่วงหน้า ก่อนนำข้อมูลเข้าสู่ dashboard visualization และระบบฐานข้อมูลสำหรับจัดเก็บข้อมูลแบบเรียลไทม์ โดยข้อมูลแรงดันและกระแสที่ได้รับจาก MQTT จะถูกใช้สำหรับติดตามสถานะของแบตเตอรี่ ประเมินพฤติกรรมของกระบวนการชาร์จ และใช้ในการวิเคราะห์พลังงานของระบบในขั้นตอนถัดไปของงานวิจัย

จากโค้ดข้างต้น ESP32 ภายในตู้ชาร์จจะทำการสร้าง JSON payload จากข้อมูล CAN ID ค่าแรงดัน และค่ากระแสที่ได้รับจากระบบ CANopen ก่อน publish ข้อมูลผ่าน MQTT Broker ไปยัง Node-RED ผ่านเครือข่าย Wi-Fi โดยภายใน supervisory layer ข้อมูลดังกล่าวจะถูกนำไปใช้สำหรับติดตามสถานะของแบตเตอรี่ แสดงผลข้อมูลแบบเรียลไทม์ และเชื่อมโยงเข้าสู่ execution layer ของระบบฟิกส์ CC-CV เพื่อใช้ในการควบคุมกระบวนการชาร์จของแบตเตอรี่แต่ละระบบอย่างต่อเนื่องและเป็นเชิงกำหนด

3.5 โครงสร้างการแลกเปลี่ยนข้อมูลสำหรับเลือกโปรไฟล์การชาร์จ

งานวิจัยนี้ได้ออกแบบโครงสร้างการแลกเปลี่ยนข้อมูลระหว่างกล่องแบตเตอรี่และตู้ชาร์จเพื่อใช้สำหรับระบุโปรไฟล์การชาร์จและส่งข้อมูลเข้าสู่ระบบควบคุมแบบฟิกส์ CC-CV โดยระบบถูกออกแบบภายใต้แนวคิดของ Distributed communication architecture ซึ่งแบ่งการทำงานออกเป็น Communication layer, Supervisory layer และ Execution layer เพื่อให้การรับส่งข้อมูลและการควบคุมระบบสามารถทำงานร่วมกันได้อย่างต่อเนื่องและเป็นเชิงกำหนด ทั้งนี้ กล่องแบตเตอรี่จะทำหน้าที่ส่งข้อมูล CANopen ผ่านเครือข่าย CAN Bus ไปยังตู้ชาร์จ ขณะที่ตู้ชาร์จจะทำหน้าที่รับข้อมูล CAN ID และข้อมูลแรงดัน-กระแส ก่อนส่งข้อมูลเข้าสู่ Node-RED ผ่าน MQTT เพื่อใช้ในการติดตามข้อมูลแบบเรียลไทม์และเชื่อมโยงไปยังระบบควบคุมการชาร์จ

ภายในโครงสร้างดังกล่าว CANopen ถูกใช้สำหรับการรับส่งข้อมูลระหว่างกล่องแบตเตอรี่และตู้ชาร์จ เนื่องจากรองรับการสื่อสารแบบ Real-time communication และมีความทนทานต่อสัญญาณรบกวนในระบบกำลังไฟฟ้า ขณะที่ MQTT ถูกใช้เป็นโปรโตคอลสำหรับส่งข้อมูลเข้าสู่ Supervisory layer ผ่านเครือข่าย Wi-Fi เพื่อให้ Node-RED สามารถติดตามข้อมูลของระบบและจัดการข้อมูลภายใน Execution layer ได้อย่างต่อเนื่อง ทั้งนี้ ข้อมูล CAN ID ที่ได้รับจากแบตเตอรี่จะถูกใช้สำหรับระบุโปรไฟล์การชาร์จและเชื่อมโยงไปยังระบบควบคุมฟิกส์ CC-CV ผ่าน Relay RS485 และ Modbus RTU ดังแสดงในรูปที่ 3.12



รูปที่ 3.12 การสื่อสารข้อมูลจาก CAN Bus สู่ระบบ redPLC-NodeRED ผ่านโปรโตคอล MQTT

จากรูปที่ 3.12 จะเห็นได้ว่าระบบเริ่มต้นจากกล่องแบตเตอรี่ทำการส่ง CAN ID ผ่านเครือข่าย CAN Bus ไปยังตู้ชาร์จ เมื่อ ESP32 ภายในตู้ชาร์จได้รับข้อมูลแล้ว ระบบจะทำการประมวลผลข้อมูล CANopen ก่อนสร้างข้อมูลในรูปแบบ JSON payload และส่งข้อมูล CAN ID, Voltage และ Current ผ่าน MQTT Broker ไปยัง Node-RED ผ่านเครือข่าย Wi-Fi เพื่อใช้ในการติดตามสถานะของแบตเตอรี่และเชื่อมโยงเข้าสู่ Execution layer ของระบบฟิกส์ CC-CV สำหรับเลือกโปรไฟล์การชาร์จของแบตเตอรี่แต่ละระบบ

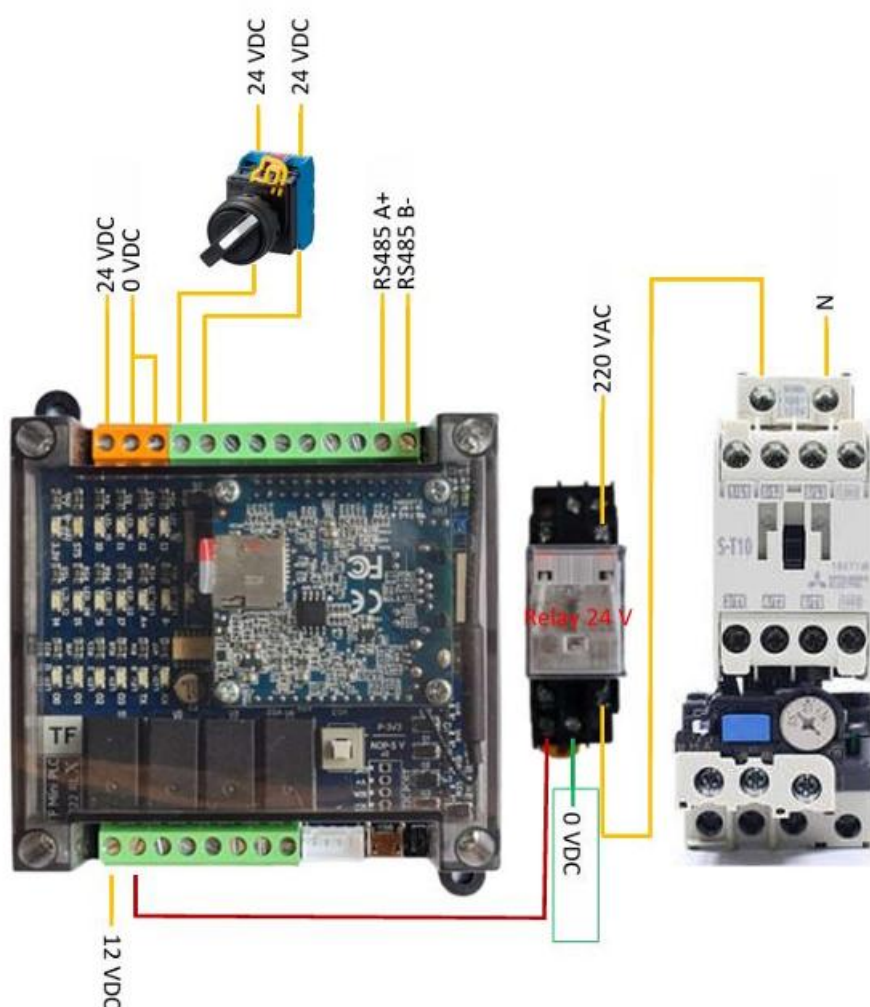
3.6 การพัฒนาระบบควบคุมด้วย Mini PLC-H3 (redPLC or Node-RED)

ภายหลังจากระบบสามารถรับ-ส่งข้อมูลระหว่างกล่องแบตเตอรี่และตู้ชาร์จผ่าน CANopen และ MQTT ได้แล้ว งานวิจัยนี้ได้พัฒนาระบบควบคุมฝั่ง Hardware โดยใช้ Mini PLC-H3 หรือ redPLC เป็น Execution controller หลักสำหรับควบคุมการทำงานของระบบฟิกส์ CC-CV ภายในตู้ชาร์จ ทั้งนี้ redPLC จะทำหน้าที่รับข้อมูลโปรไฟล์การชาร์จจาก Supervisory layer ก่อนส่งคำสั่งควบคุมไปยังอุปกรณ์ภายในระบบผ่านโปรโตคอล Modbus RTU บนเครือข่าย RS485 เพื่อใช้สำหรับเลือกโปรไฟล์การชาร์จ ควบคุม Relay Module และเชื่อมโยงข้อมูลกับอุปกรณ์วัดพลังงานไฟฟ้าภายในระบบ

ในการพัฒนาระบบดังกล่าว งานวิจัยนี้เลือกใช้ Mini PLC-H3 เนื่องจากสามารถรองรับการสื่อสารผ่าน RS485 ได้โดยตรง มี Digital Output สำหรับควบคุม Relay และสามารถทำงานร่วมกับ Node-RED ผ่าน Modbus RTU ได้อย่างมีประสิทธิภาพ นอกจากนี้ redPLC ยังถูกใช้เป็นศูนย์กลางในการเชื่อมต่อกับอุปกรณ์ Power Meter ผ่านอินเทอร์เฟซ RS485 เพื่ออ่านค่าพลังงานไฟฟ้า แรงดัน กระแส และกำลังไฟฟ้าของระบบชาร์จ ทั้งนี้การออกแบบในลักษณะดังกล่าวช่วยให้ Execution

layer และ monitoring layer สามารถทำงานร่วมกันผ่านโครงสร้างการสื่อสารเดียวกันได้อย่างต่อเนื่อง

โครงสร้างของระบบควบคุมผ่าน Mini PLC-H3 แสดงดังรูปที่ 3.12 โดยภายในระบบใช้ redPLC เชื่อมต่อกับ Relay RS485 ผ่านเครือข่าย RS485 Bus เพื่อควบคุมวงจรฟิกส์ CC-CV ของเครื่องชาร์จเตลต้าจำนวน 4 โพรไฟล์ ประกอบด้วย 24V 5A, 48V 5A, 60V 5A และ 72V 15A ทั้งนี้ Relay Module แต่ละชุดจะเชื่อมต่อกับวงจรค่าความต้านทานของระบบควบคุมแรงดันและกระแส ซึ่งถูกกำหนดค่าไว้ล่วงหน้าเพื่อใช้สำหรับเลือกค่าพารามิเตอร์ของเครื่องชาร์จให้ตรงกับโปรไฟล์ของแบตเตอรี่แต่ละระบบ



รูปที่ 3.13 การเชื่อมต่อ Mini PLC-H3 ร่วมกับอุปกรณ์ป้องกันฝั่งกำลังของเครื่องชาร์จ

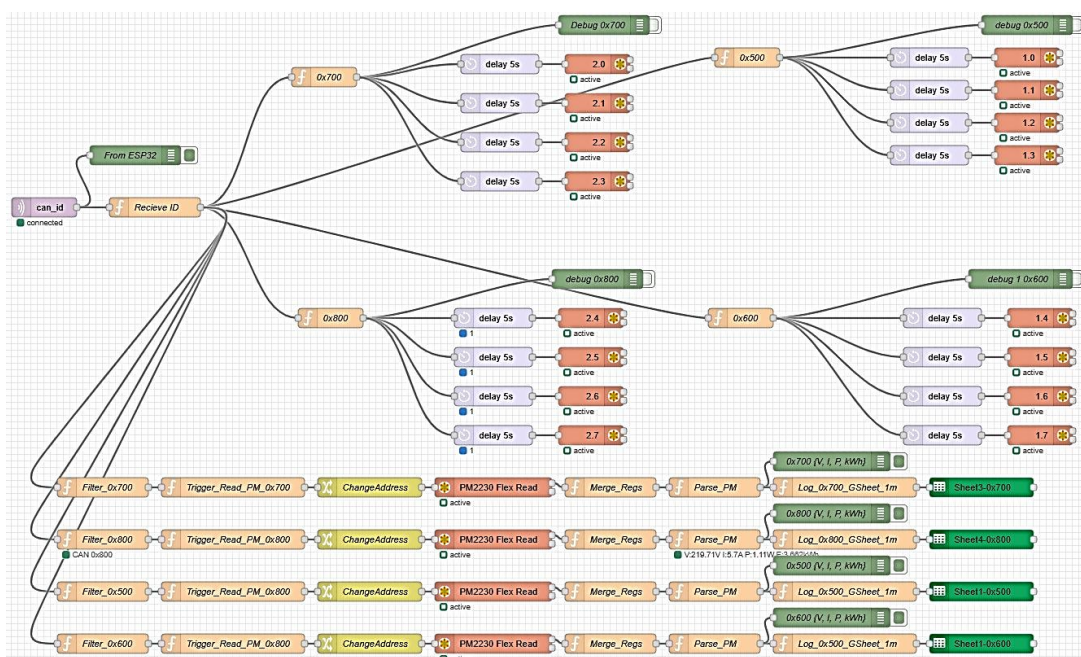
จากรูปที่ 3.13 จะเห็นได้ว่าระบบใช้ redPLC เป็นอุปกรณ์หลักในการควบคุม Relay RS485 ผ่านเครือข่าย RS485 Bus โดย Relay แต่ละชุดจะทำหน้าที่เลือกวงจรค่าความต้านทานของระบบ

ฟิกส์ CC-CV สำหรับควบคุมค่าแรงดันคงที่ (Constant Voltage: CV) และค่ากระแสคงที่ (Constant Current: CC) ของเครื่องชาร์จเซลล์ทำให้สอดคล้องกับโปรไฟล์ของแบตเตอรี่แต่ละระดับแรงดัน ทั้งนี้ redPLC ยังสามารถเชื่อมต่อกับ Power Meter ผ่านอินเทอร์เฟซ RS485 เดียวกันเพื่ออ่านค่าพลังงานไฟฟ้าและส่งข้อมูลเข้าสู่ Supervisory layer ผ่าน Node-RED ได้อีกด้วย ส่งผลให้ระบบสามารถควบคุมและติดตามสถานะของระบบชาร์จได้ภายใต้ execution layer เดียวกันอย่างมีประสิทธิภาพ

3.6.1 การพัฒนาระบบ Supervisory Control บน Node-RED

ระบบ Supervisory Control ถูกพัฒนาขึ้นบนแพลตฟอร์ม Node-RED เพื่อใช้เป็นศูนย์กลางในการประมวลผลข้อมูล การจัดการเส้นทางการทำงานของระบบ และการเชื่อมโยงข้อมูลระหว่าง Communication layer และ Execution layer ของระบบชาร์จแบตเตอรี่ NMC แรงดันต่ำ โดย Node-RED จะทำหน้าที่รับข้อมูลจาก ESP32 ภายในตู้ชาร์จผ่านโปรโตคอล MQTT ก่อนนำข้อมูล CAN ID, Voltage และ Current มาประมวลผลเพื่อใช้ในการระบุโปรไฟล์การชาร์จของแบตเตอรี่แต่ละระบบ ทั้งนี้ข้อมูล CAN ID ที่ได้รับจาก Communication layer จะถูกใช้เป็นตัวแยกเส้นทางการทำงาน (Routing Path) ภายใน Node-RED เพื่อเลือกการทำงานของ Execution layer ให้สอดคล้องกับโปรไฟล์ของแบตเตอรี่แต่ละระดับแรงดัน นอกจากนี้ Node-RED ยังทำหน้าที่เป็น Supervisory controller สำหรับควบคุมลำดับการทำงานของระบบทั้งหมด ตั้งแต่การรับข้อมูลจาก communication layer การเลือกโปรไฟล์การชาร์จ การส่งคำสั่งไปยัง redPLC ไปจนถึงการจัดเก็บข้อมูลของระบบชาร์จเข้าสู่ฐานข้อมูลและระบบวิเคราะห์ข้อมูล

ภายใน Supervisory layer ของระบบ Node-RED ถูกออกแบบให้สามารถทำงานร่วมกับอุปกรณ์หลายชนิดภายใต้โครงสร้างการสื่อสารเดียวกัน โดยระบบจะทำการ Subscribe ข้อมูลจาก MQTT Broker อย่างต่อเนื่องเพื่อติดตามสถานะของแบตเตอรี่แบบ Real-time เมื่อได้รับข้อมูล CAN ID แล้ว ระบบจะทำการตรวจสอบหมายเลขประจำตัวของแบตเตอรี่และแยกเส้นทางข้อมูลเข้าสู่โปรไฟล์การชาร์จที่กำหนดไว้ล่วงหน้า ก่อนส่งคำสั่งไปยัง Mini PLC-H3 (redPLC) ผ่าน Modbus RTU เพื่อควบคุม Relay RS485 สำหรับเลือกวงจรฟิกส์ CC-CV ของเครื่องชาร์จเซลล์ นอกจากนี้ Node-RED ยังถูกใช้สำหรับอ่านค่าพลังงานไฟฟ้าจาก Power Meter รุ่น PM2200 ผ่านเครือข่าย RS485 เพื่อเก็บข้อมูลแรงดัน กระแส กำลังไฟฟ้า และพลังงานไฟฟ้าของระบบชาร์จ ทั้งนี้ข้อมูลทั้งหมดจะถูกจัดเก็บเข้าสู่ Google Sheets และระบบฐานข้อมูลสำหรับการติดตามสถานะของระบบ วิเคราะห์พฤติกรรมกรรมการชาร์จ และประเมินสมรรถนะของระบบในขั้นตอนถัดไป ดังแสดงในรูปที่ 3.14



รูปที่ 3.14 การพัฒนาระบบ Supervisory Control บน Node-RED สำหรับเลือกโปรไฟล์การชาร์จ และติดตามข้อมูลพลังงาน

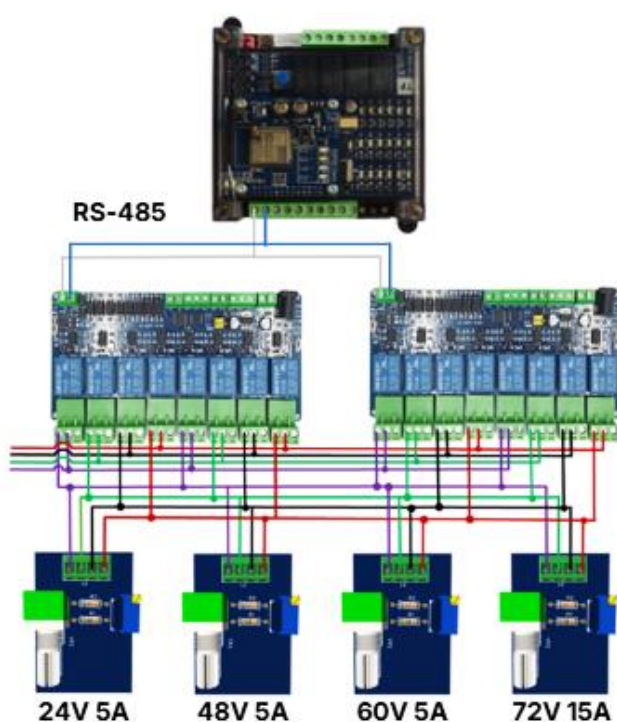
จากรูปที่ 3.14 จะเห็นได้ว่าโครงสร้างของ Node-RED ถูกแบ่งออกเป็นหลายส่วนตามหน้าที่การทำงานของระบบ โดยส่วนแรกทำหน้าที่รับข้อมูล MQTT จาก ESP32 ผ่าน MQTT Broker ก่อนนำข้อมูล CAN ID เข้าสู่กระบวนการตรวจสอบและแยกเส้นทางข้อมูลตามโปรไฟล์ของแบตเตอรี่แต่ละระบบ ขณะที่ส่วนกลางของระบบจะทำหน้าที่ควบคุม Execution layer ผ่าน redPLC และอ่านค่าพลังงานไฟฟ้าจาก PM2200 ผ่าน Modbus RTU ก่อนนำข้อมูลเข้าสู่กระบวนการแยกและจัดรูปแบบข้อมูล (Data Parsing) สำหรับสร้างข้อมูลด้านแรงดัน กระแส กำลังไฟฟ้า และพลังงานไฟฟ้าของระบบชาร์จ ทั้งนี้ส่วนสุดท้ายของระบบจะทำหน้าที่จัดเก็บข้อมูลเข้าสู่ Google Sheets และระบบฐานข้อมูลสำหรับการติดตามสถานะของระบบและวิเคราะห์พฤติกรรมของกระบวนการชาร์จแบตเตอรี่ในระดับ Supervisory layer ต่อไป

3.6.2 การควบคุม Relay RS485 สำหรับเลือกโปรไฟล์การชาร์จ

ภายใน Execution layer ของระบบ งานวิจัยนี้ได้ออกแบบการควบคุมโปรไฟล์การชาร์จผ่าน Relay RS485 โดยใช้ Mini PLC-H3 (redPLC) เป็นอุปกรณ์หลักในการส่งคำสั่ง Modbus RTU เพื่อเลือกวงจรฟลักซ์ CC-CV ของเครื่องชาร์จเซลล์ทำให้สอดคล้องกับโปรไฟล์ของแบตเตอรี่ NMC แต่ละระดับแรงดัน ทั้งนี้เมื่อ Node-RED ได้รับข้อมูล CAN ID จาก Communication layer แล้วระบบ Supervisory control จะทำการประมวลผลข้อมูลและส่งคำสั่งไปยัง redPLC ผ่านเครือข่าย

RS485 เพื่อเลือกช่องสัญญาณของ Relay Module สำหรับกำหนดค่าพารามิเตอร์แรงดันและกระแสของเครื่องชาร์จ

โครงสร้างของระบบควบคุม Relay RS485 แสดงดังรูปที่ 3.15 โดยภายในระบบใช้ redPLC เชื่อมต่อกับ Relay Module จำนวน 2 ชุดผ่านอินเทอร์เฟซ RS485 เพื่อควบคุมวงจรฟิกส์ CC-CV ของเครื่องชาร์จเตลต้าจำนวน 4 โพรไฟล์ ประกอบด้วย 24V 5A, 48V 5A, 60V 5A และ 72V 15A ทั้งนี้ Relay แต่ละช่องจะเชื่อมต่อกับชุดค่าความต้านทานของวงจรควบคุมแรงดันและกระแส ซึ่งถูกกำหนดค่าไว้ล่วงหน้าตามโปรไฟล์ของแบตเตอรี่แต่ละระบบ



รูปที่ 3.15 การควบคุม Relay RS485 สำหรับเลือกโปรไฟล์ฟิกส์ CC-CV ของเครื่องชาร์จเตลต้า

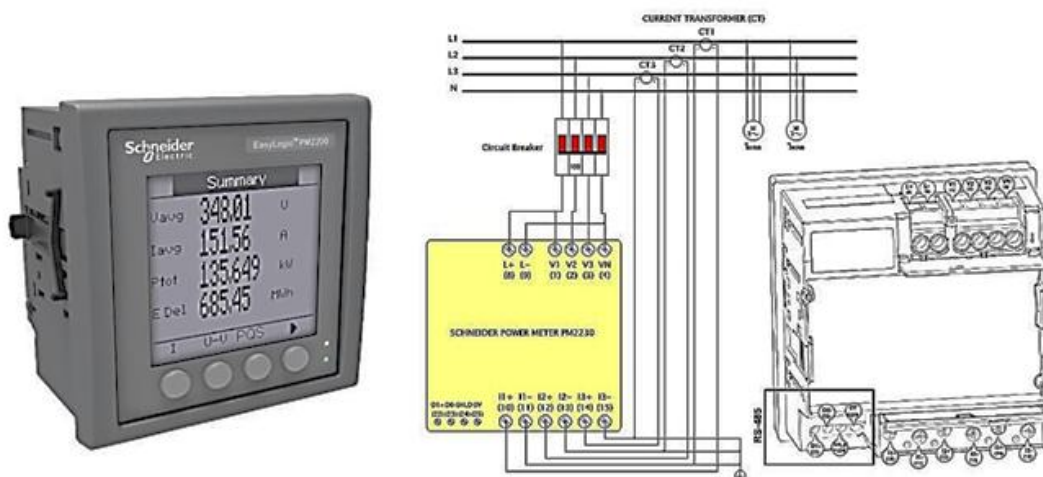
จากรูปที่ 3.15 จะเห็นได้ว่าระบบใช้เครือข่าย RS485 เป็น Communication interface สำหรับเชื่อมต่อระหว่าง redPLC และ Relay Module โดย Relay Module ชุดแรกทำหน้าที่ควบคุมโปรไฟล์ 24V 5A และ 48V 5A ขณะที่ Relay Module ชุดที่สองใช้สำหรับควบคุมโปรไฟล์ 60V 5A และ 72V 15A ทั้งนี้เมื่อ redPLC ได้รับคำสั่งจาก Node-RED ระบบจะทำการเลือกช่องสัญญาณของรีเลย์เพื่อเชื่อมต่อวงจรค่าความต้านทานของระบบฟิกส์ CC-CV ให้ตรงกับโปรไฟล์ของแบตเตอรี่ ส่งผลให้เครื่องชาร์จสามารถกำหนดค่าแรงดันคงที่ (Constant Voltage: CV) และค่ากระแสคงที่ (Constant Current: CC) ได้ตามพารามิเตอร์ที่ออกแบบไว้ล่วงหน้า ภายในงานวิจัยนี้

จากรูปที่ 3.16 จะเห็นได้ว่าภายใน Node-RED ระบบถูกออกแบบให้สามารถแยกเส้นทางข้อมูลของ CAN ID แต่ละระบบออกจากกันอย่างชัดเจน โดยหลังจากตรวจสอบหมายเลข CAN ID แล้ว ระบบจะส่งคำสั่ง Modbus RTU ผ่าน RS485 Interface ไปยัง Relay Module เพื่อเลือกช่องสัญญาณของระบบฟิกส์ CC-CV ให้ตรงกับโปรไฟล์ของแบตเตอรี่แต่ละระดับแรงดัน เช่น 24V 5A, 48V 5A, 60V 5A และ 72V 15A ส่งผลให้ Execution layer ของระบบสามารถกำหนดค่าพารามิเตอร์ของเครื่องชาร์จได้อย่างต่อเนื่องและลดความผิดพลาดจากการเลือกโปรไฟล์การชาร์จภายในระบบเดียวกัน

3.6.3 การติดตามและบันทึกข้อมูลพลังงานฝั่ง AC ของระบบชาร์จ

ภายในงานวิจัยนี้ ระบบติดตามข้อมูลพลังงานฝั่งกระแสสลับ (AC Side Measurement) ถูกพัฒนาขึ้นเพื่อใช้สำหรับตรวจสอบพฤติกรรมการใช้พลังงานของเครื่องชาร์จเตลต้า วิเคราะห์สมรรถนะของระบบชาร์จ และใช้เป็นข้อมูลอ้างอิงสำหรับตรวจสอบความถูกต้องของ Execution layer ภายในระบบชาร์จแบตเตอรี่ NMC แรงดันต่ำ ทั้งนี้การตรวจวัดพลังงานฝั่ง AC มีความสำคัญต่อการวิเคราะห์ความสัมพันธ์ระหว่างพลังงานไฟฟ้าขาเข้าของเครื่องชาร์จและพฤติกรรมของระบบฟิกส์ CC-CV โดยเฉพาะในช่วง Constant Current (CC) และ Constant Voltage (CV) ซึ่งมีการเปลี่ยนแปลงของกำลังไฟฟ้าและกระแสไฟฟ้าตลอดกระบวนการชาร์จ ดังนั้นข้อมูลที่ได้จากระบบตรวจวัดจึงถูกใช้สำหรับติดตามสถานะของระบบ วิเคราะห์พฤติกรรมการใช้พลังงาน และประเมินสมรรถนะของเครื่องชาร์จในแต่ละโปรไฟล์การชาร์จ

ในการตรวจวัดพลังงานไฟฟ้าฝั่ง AC งานวิจัยนี้เลือกใช้ Power Meter รุ่น PM2200 สำหรับติดตั้งบริเวณด้าน Input ของเครื่องชาร์จเตลต้า โดยอุปกรณ์ดังกล่าวสามารถตรวจวัดค่าพารามิเตอร์ทางไฟฟ้าที่สำคัญ ได้แก่ ค่าแรงดันไฟฟ้า (Voltage) กระแสไฟฟ้า (Current) กำลังไฟฟ้า (Active Power) และพลังงานไฟฟ้าสะสม (Energy) ผ่าน Current Transformer (CT) ที่ติดตั้งบริเวณสายป้อนของระบบ ทั้งนี้ PM2200 สามารถสื่อสารข้อมูลผ่านโปรโตคอล Modbus RTU บนเครือข่าย RS485 ทำให้สามารถเชื่อมโยงข้อมูลเข้าสู่ supervisory layer ของระบบได้โดยตรง นอกจากนี้ข้อมูลที่ได้จาก PM2200 ยังถูกใช้สำหรับประเมินความต่อเนื่องของระบบชาร์จ ตรวจสอบพฤติกรรมการใช้พลังงานของเครื่องชาร์จ และใช้สำหรับเปรียบเทียบพลังงานไฟฟ้าระหว่างแต่ละโปรไฟล์การชาร์จภายในงานวิจัย ดังแสดงในรูปที่ 3.17



รูปที่ 3.17 การติดตามพลังงานไฟฟ้าฝั่ง AC ด้วย PM2200 ผ่าน Modbus RTU

จากรูปที่ 3.17 จะเห็นได้ว่ามิเตอร์ PM2200 ถูกติดตั้งอยู่ด้าน Input ของเครื่องชาร์จเตลต้าเพื่อใช้สำหรับตรวจวัดพลังงานไฟฟ้าฝั่ง AC โดยระบบจะใช้ Current Transformer (CT) สำหรับวัดค่ากระแสไฟฟ้าของเครื่องชาร์จ ขณะที่ค่าแรงดันไฟฟ้า กำลังไฟฟ้า และพลังงานไฟฟ้าสะสมจะถูกประมวลผลภายในมิเตอร์ก่อนส่งข้อมูลผ่านเครือข่าย RS485 ไปยัง Node-RED ผ่านโปรโตคอล Modbus RTU ทั้งนี้ข้อมูลที่ถูกตรวจวัดจะถูกใช้สำหรับติดตามพฤติกรรมการใช้พลังงานของเครื่องชาร์จในแต่ละช่วงของกระบวนการชาร์จ รวมถึงใช้เป็นข้อมูลอ้างอิงสำหรับประเมินสมรรถนะของ Execution layer ภายในระบบชาร์จ

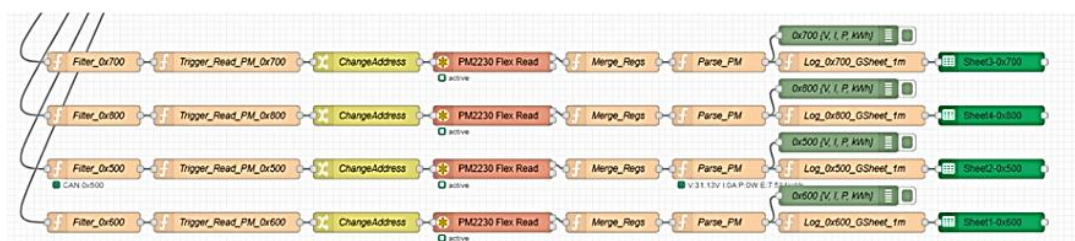
ในการอ่านค่าพลังงานจาก PM2200 งานวิจัยนี้กำหนดให้ Node-RED อ่านข้อมูลผ่านรีจิสเตอร์ของ Modbus RTU ซึ่งค่าพารามิเตอร์ทางไฟฟ้าแต่ละชนิดจะถูกจัดเก็บอยู่ภายใน Address ของรีจิสเตอร์ที่แตกต่างกัน ทั้งนี้ข้อมูลที่ใช้ภายในงานวิจัยประกอบด้วยค่าแรงดันไฟฟ้า กระแสไฟฟ้า กำลังไฟฟ้า และพลังงานไฟฟ้าสะสม โดยข้อมูลทั้งหมดถูกจัดเก็บในรูปแบบ 32-bit Register เพื่อเพิ่มความละเอียดของข้อมูลสำหรับการวิเคราะห์พฤติกรรมของระบบชาร์จ ดังแสดงในตารางที่ 3.5

ตารางที่ 3.5 การแมปรีจิสเตอร์ Modbus RTU สำหรับตรวจวัดพลังงานไฟฟ้าฝั่ง AC ของ PM2200

พารามิเตอร์	Address (Register)	จำนวน Register	หน่วย	หมายเหตุ
แรงดันไฟฟ้า Voltage: V	3027-3028	2 (32 bits)	V	ค่าเฉลี่ยแรงดันระบบ (Line to Line)
กระแสไฟฟ้า Current: I	3001-3002	2 (32 bits)	A	ค่าเฉลี่ยกระแสไฟฟ้า
กำลังไฟฟ้า Active Power: P	3053-3054	2 (32 bits)	kW	กำลังไฟฟ้าจริงของระบบ
พลังงานไฟฟ้า Energy: kWh	3203-3204	2 (32 bits)	kWh	พลังงานสะสม

จากตารางที่ 3.5 จะเห็นได้ว่าค่าแรงดันไฟฟ้าของระบบถูกจัดเก็บอยู่ที่รีจิสเตอร์ Address 3027–3028 ขณะที่ค่ากระแสไฟฟ้าจะอยู่ที่ Address 3001–3002 ส่วนค่ากำลังไฟฟ้าจริงของระบบจะถูกอ่านจาก Address 3053–3054 และค่าพลังงานไฟฟ้าสะสมจะถูกจัดเก็บอยู่ที่ Address 3203–3204 ทั้งนี้ Node-RED จะทำหน้าที่อ่านค่าข้อมูลจากรีจิสเตอร์ดังกล่าวผ่าน Modbus RTU ก่อนนำข้อมูลเข้าสู่กระบวนการรวมข้อมูล (Merge Register) และแปลงข้อมูลให้อยู่ในรูปแบบที่สามารถใช้งานได้ภายใน supervisory layer ของระบบชาร์จต่อไป

ภายใน Supervisory layer งานวิจัยนี้ได้พัฒนาระบบบน Node-RED สำหรับอ่านค่าพลังงานจาก PM2200 แบบ real-time โดย Node-RED จะทำหน้าที่ Trigger การอ่านค่าจากมิเตอร์ผ่าน Modbus RTU ตาม CAN ID ของแบตเตอรี่แต่ละระบบ ก่อนนำข้อมูลรีจิสเตอร์เข้าสู่กระบวนการรวมข้อมูล (Merge Register) และแปลงข้อมูลให้อยู่ในรูปแบบที่สามารถใช้งานได้ภายในระบบ ทั้งนี้ข้อมูลที่ถูกประมวลผลจะประกอบด้วยค่าแรงดันไฟฟ้า กระแสไฟฟ้า กำลังไฟฟ้า และพลังงานไฟฟ้าสะสม ก่อนถูกส่งเข้าสู่ระบบฐานข้อมูลและ dashboard สำหรับใช้ในการติดตามสถานะของระบบแบบ Real-time ดังแสดงในรูปที่ 3.18



รูปที่ 3.18 การอ่านและบันทึกข้อมูลพลังงานจาก PM2200 บน Node-RED

จากรูปที่ 3.18 จะเห็นได้ว่าภายใน Node-RED ระบบถูกออกแบบให้สามารถแยกเส้นทางการอ่านค่าพลังงานตาม CAN ID ของแบตเตอรี่แต่ละระบบ โดยเมื่อระบบได้รับข้อมูล CAN ID จาก Communication layer แล้ว Node-RED จะทำการ Trigger การอ่านข้อมูลจาก PM2200 ผ่าน Modbus RTU ก่อนเข้าสู่กระบวนการ Parse ข้อมูลเพื่อแยกค่าแรงดันไฟฟ้า กระแสไฟฟ้า กำลังไฟฟ้า และพลังงานไฟฟ้าออกจากรีจิสเตอร์ของมิเตอร์ จากนั้นระบบจะทำการจัดเก็บข้อมูลเข้าสู่ Google Sheets แยกตามโปรไฟล์การชาร์จของแบตเตอรี่แต่ละระบบ เช่น 0x500, 0x600, 0x700 และ 0x800 ทั้งนี้งานวิจัยนี้กำหนดให้ระบบทำการบันทึกข้อมูลทุก ๆ 1 นาที เพื่อใช้สำหรับติดตามพฤติกรรมการใช้พลังงานของเครื่องชาร์จในระหว่างกระบวนการชาร์จแบบ Real-time และใช้เป็นฐานข้อมูลสำหรับวิเคราะห์สถานะการชาร์จและการคำนวณค่าไฟฟ้าต่อการชาร์จในขั้นตอนถัดไป

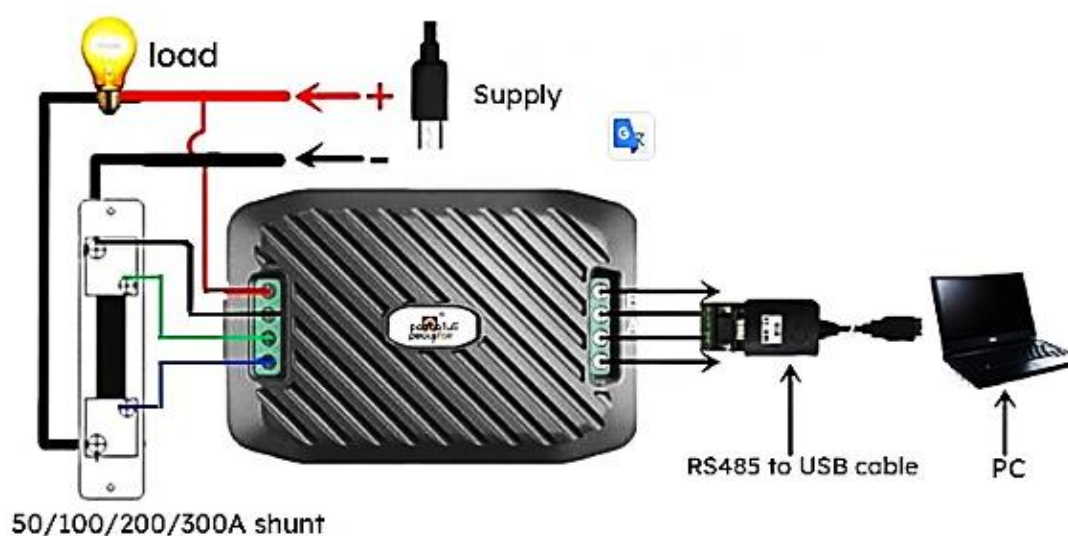
นอกจากนี้ การใช้ Google Sheets สำหรับจัดเก็บข้อมูลยังช่วยให้ระบบสามารถตรวจสอบข้อมูลย้อนหลัง วิเคราะห์พฤติกรรมการใช้พลังงาน และติดตามค่าพารามิเตอร์ของเครื่องชาร์จผ่านระบบออนไลน์ได้อย่างต่อเนื่อง โดยข้อมูลที่ถูกบันทึกจะถูกจัดเก็บในรูปแบบ Time-series data ซึ่งสามารถนำไปใช้สำหรับวิเคราะห์พฤติกรรมของเครื่องชาร์จในแต่ละโปรไฟล์ เปรียบเทียบพลังงานไฟฟ้าระหว่างช่วง CC และ CV รวมถึงใช้สำหรับประเมินแนวโน้มของระบบชาร์จในระยะยาวได้อย่างมีประสิทธิภาพ ทั้งนี้แนวทางดังกล่าวช่วยให้ Supervisory layer ของระบบสามารถทำงานร่วมกับ Execution layer และ Monitoring layer ได้อย่างสมบูรณ์ภายใต้โครงสร้างการสื่อสารเดียวกันของงานวิจัย

3.6.4 การติดตามและบันทึกข้อมูลพลังงานฝั่ง DC ของระบบชาร์จ

ระบบติดตามข้อมูลพลังงานฝั่งกระแสตรง (DC Side Measurement) ถูกพัฒนาขึ้นเพื่อใช้สำหรับตรวจสอบพฤติกรรมของกระบวนการชาร์จแบตเตอรี่ NMC แรงดันต่ำในระดับ Output ของเครื่องชาร์จเตลต้า โดยการตรวจวัดพลังงานฝั่ง DC มีความสำคัญต่อการวิเคราะห์พฤติกรรมของระบบฟิกส์ CC-CV เนื่องจากค่ากระแสไฟฟ้าและแรงดันไฟฟ้าจะเปลี่ยนแปลงตามลักษณะการชาร์จในแต่ละช่วงของกระบวนการชาร์จ ทั้งนี้ข้อมูลที่ตรวจวัดได้จะถูกใช้สำหรับตรวจสอบช่วงกระแสคงที่ (Constant Current: CC) และช่วงแรงดันคงที่ (Constant Voltage: CV) รวมถึงใช้สำหรับประเมิน

สมรรถนะของเครื่องชาร์จและความสอดคล้องของ Execution layer กับโปรไฟล์การชาร์จที่กำหนดไว้ภายในระบบ

ในการตรวจวัดพลังงานไฟฟ้าฝั่ง DC งานวิจัยนี้เลือกใช้ Power Meter รุ่น PZEM-017 ร่วมกับ Shunt resistor ขนาด 50A สำหรับติดตั้งบริเวณเอาต์พุตของเครื่องชาร์จเตลต้าระหว่างเครื่องชาร์จและแบตเตอรี่ โดย PZEM-017 สามารถตรวจวัดค่าแรงดันไฟฟ้า กระแสไฟฟ้า กำลังไฟฟ้า และพลังงานไฟฟ้าสะสมของระบบได้โดยตรง ทั้งนี้หลักการทำงานของระบบตรวจวัดจะอาศัยการวัดแรงดันตกคร่อมบน Shunt resistor เพื่อนำไปคำนวณค่ากระแสไฟฟ้าที่ไหลผ่านระบบ ขณะที่ค่าแรงดันไฟฟ้าของแบตเตอรี่จะถูกตรวจวัดจากเอาต์พุตของเครื่องชาร์จโดยตรง ส่งผลให้ระบบสามารถติดตามพฤติกรรมของกระบวนการชาร์จในระดับ DC ได้อย่างต่อเนื่อง ดังแสดงในรูปที่ 3.19



รูปที่ 3.19 การตรวจวัดพลังงานไฟฟ้าฝั่ง DC ด้วย PZEM-017 ร่วมกับ Shunt Resistor 50A

จากรูปที่ 3.19 จะเห็นได้ว่า PZEM-017 ถูกติดตั้งอยู่บริเวณเอาต์พุตของเครื่องชาร์จเตลต้าเพื่อใช้สำหรับตรวจวัดค่าพารามิเตอร์ทางไฟฟ้าของระบบชาร์จฝั่ง DC โดยอุปกรณ์ดังกล่าวสามารถสื่อสารข้อมูลผ่านโปรโตคอล Modbus RTU บนเครือข่าย RS485 ก่อนส่งข้อมูลเข้าสู่ Supervisory layer ของระบบ ทั้งนี้ข้อมูลที่ได้จาก PZEM-017 จะถูกใช้สำหรับติดตามพฤติกรรมของแรงดันและกระแสไฟฟ้าในช่วง CC และ CV รวมถึงใช้สำหรับวิเคราะห์สมรรถนะของเครื่องชาร์จในแต่ละโปรไฟล์การชาร์จของแบตเตอรี่

การอ่านค่าพลังงานจาก PZEM-017 งานวิจัยนี้กำหนดให้ Node-RED อ่านข้อมูลผ่านรีจิสเตอร์ของ Modbus RTU โดยข้อมูลที่ใช้ภายในระบบประกอบด้วยค่าแรงดันไฟฟ้า กระแสไฟฟ้า กำลังไฟฟ้า พลังงานไฟฟ้าสะสม และสถานะแรงดันเกิน (High Voltage Alarm) ซึ่ง

ค่าพารามิเตอร์แต่ละชนิดจะถูกจัดเก็บอยู่ในรีจิสเตอร์ Address ที่แตกต่างกัน ทั้งนี้ข้อมูลแรงดันและกระแสจะถูกจัดเก็บในรูปแบบ 16-bit Register ขณะที่ค่ากำลังไฟฟ้าและพลังงานไฟฟ้าจะถูกจัดเก็บในรูปแบบ 32-bit Register เพื่อเพิ่มความละเอียดของข้อมูลสำหรับการวิเคราะห์พฤติกรรมของระบบชาร์จ ดังแสดงในตารางที่ 3.6

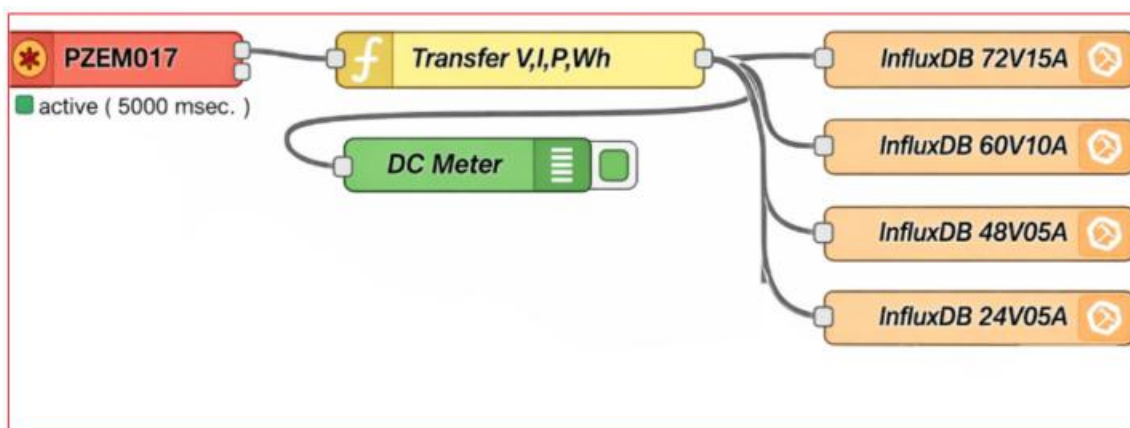
ตารางที่ 3.6 การแมปรีจิสเตอร์ Modbus RTU สำหรับตรวจวัดพลังงานไฟฟ้าฝั่ง DC ของ PZEM-017

พารามิเตอร์	Address (Register)	จำนวนรีจิสเตอร์	หน่วย	Resolution
แรงดันไฟฟ้า (Voltage, V)	0x0000	1 (16-bit)	V	1 LSB = 0.01 V
กระแสไฟฟ้า (Current, I)	0x0001	1 (16-bit)	A	1 LSB = 0.01 A
กำลังไฟฟ้า (Power, P)	0x0002 – 0x0003	2 (32-bit)	W	1 LSB = 0.1 W
พลังงานไฟฟ้า (Energy, kWh)	0x0004 – 0x0005	2 (32-bit)	kWh	1 LSB = 1 Wh
สถานะแรงดันเกิน (High Voltage Alarm)	0x0006	1	-	0xFFFF = Alarm

จากตารางที่ 3.6 จะเห็นได้ว่าค่าแรงดันไฟฟ้าของระบบถูกจัดเก็บอยู่ที่รีจิสเตอร์ Address 0x0000 ขณะที่ค่ากระแสไฟฟ้าจะอยู่ที่ Address 0x0001 ส่วนค่ากำลังไฟฟ้าจะถูกอ่านจาก Address 0x0002–0x0003 และค่าพลังงานไฟฟ้าสะสมจะถูกจัดเก็บอยู่ที่ Address 0x0004–0x0005 ทั้งนี้ Node-RED จะทำหน้าที่อ่านค่าข้อมูลจากรีจิสเตอร์ดังกล่าวผ่าน Modbus RTU ก่อนนำข้อมูลเข้าสู่กระบวนการประมวลผลภายใน Supervisory layer เพื่อใช้สำหรับติดตามพฤติกรรมของระบบชาร์จและจัดเก็บข้อมูลเข้าสู่ฐานข้อมูลของระบบ

ภายใน Supervisory layer งานวิจัยนี้ได้พัฒนาระบบบน Node-RED สำหรับอ่านและจัดเก็บข้อมูลพลังงานจาก PZEM-017 แบบ Real-time โดยระบบจะทำหน้าที่รับข้อมูลแรงดันไฟฟ้า กระแสไฟฟ้า กำลังไฟฟ้า และพลังงานไฟฟ้าสะสมผ่าน Modbus RTU ก่อนนำข้อมูลเข้าสู่กระบวนการจัดรูปแบบข้อมูลให้อยู่ในลักษณะ Time-series Data สำหรับใช้ในการจัดเก็บข้อมูลเข้าสู่ฐานข้อมูล InfluxDB ทั้งนี้ข้อมูลจะถูกแยกจัดเก็บตามโปรไฟล์การชาร์จของแบตเตอรี่แต่ละ

ระบบ เช่น 24V5A, 48V5A, 60V5A และ 72V15A เพื่อให้สามารถเปรียบเทียบพฤติกรรมของ กระบวนการชาร์จในแต่ละโปรไฟล์ได้อย่างชัดเจน ดังแสดงในรูปที่ 3.20



รูปที่ 3.20 การจัดเก็บข้อมูลพลังงานฝั่ง DC จาก PZEM-017 ลงใน InfluxDB ผ่าน Node-RED

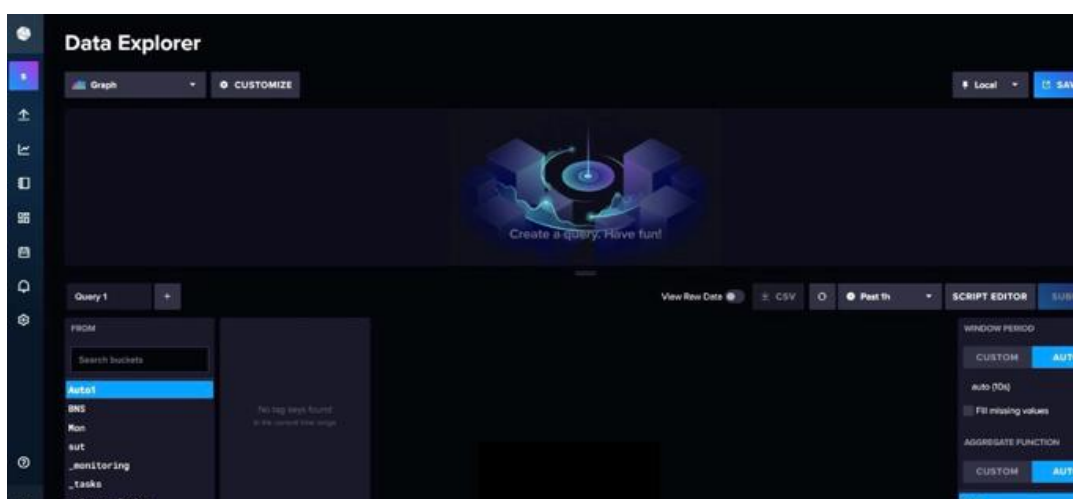
จากรูปที่ 3.20 จะเห็นได้ว่า Node-RED ถูกออกแบบให้สามารถรับข้อมูลจาก PZEM-017 ผ่าน Modbus RTU ก่อนนำข้อมูลเข้าสู่กระบวนการ Transfer ข้อมูลให้อยู่ในรูปแบบที่เหมาะสมสำหรับการจัดเก็บในฐานข้อมูล InfluxDB โดยข้อมูลที่ถูกจัดเก็บจะประกอบด้วยค่า แรงดันไฟฟ้า กระแสไฟฟ้า กำลังไฟฟ้า และพลังงานไฟฟ้าสะสมของระบบชาร์จ ทั้งนี้การจัดเก็บข้อมูล ในลักษณะ Time-series Data ช่วยให้สามารถติดตามพฤติกรรมของระบบฟิกส์ CC-CV วิเคราะห์การ เปลี่ยนแปลงของแรงดันและกระแสระหว่างกระบวนการชาร์จ รวมถึงใช้สำหรับประเมินสมรรถนะ ของเครื่องชาร์จในแต่ละโปรไฟล์ได้อย่างมีประสิทธิภาพ นอกจากนี้การใช้ InfluxDB ยังช่วยให้ระบบ สามารถเชื่อมโยงข้อมูลเข้าสู่ Grafana สำหรับแสดงผลข้อมูลแบบ Real-time และใช้ในการวิเคราะห์ ข้อมูลย้อนหลังของระบบชาร์จในขั้นตอนถัดไปของงานวิจัยได้อย่างต่อเนื่อง

3.6.5 การแสดงผลและวิเคราะห์ข้อมูลผ่าน Grafana Dashboard

ระบบแสดงผลและวิเคราะห์ข้อมูลถูกพัฒนาขึ้นบนแพลตฟอร์ม Grafana เพื่อใช้เป็น Visualization layer สำหรับติดตามสถานะของระบบชาร์จแบตเตอรี่ NMC แรงดันต่ำแบบ Real-time โดย Grafana ถูกใช้ร่วมกับฐานข้อมูล InfluxDB ซึ่งทำหน้าที่จัดเก็บข้อมูลในรูปแบบ Time-series data จากระบบตรวจวัดพลังงานทั้งฝั่ง AC และ DC ภายในระบบชาร์จ ทั้งนี้ข้อมูลที่ถูกจัดเก็บ จะประกอบด้วยค่าแรงดันไฟฟ้า กระแสไฟฟ้า กำลังไฟฟ้า พลังงานไฟฟ้า และข้อมูล CAN ID ของ แบตเตอรี่แต่ละระบบ ซึ่งช่วยให้สามารถติดตามพฤติกรรมของระบบฟิกส์ CC-CV และวิเคราะห์ สมรรถนะของเครื่องชาร์จได้อย่างต่อเนื่อง

ในการพัฒนาระบบแสดงผล งานวิจัยนี้กำหนดให้ Grafana เชื่อมต่อกับฐานข้อมูล InfluxDB ผ่าน Data Source เพื่อใช้สำหรับดึงข้อมูลจาก Supervisory layer มาแสดงผลในรูปแบบ Dashboard โดยระบบสามารถแสดงผลข้อมูลได้ทั้งในรูปแบบ Time-series Graph, Gauge และ Table ส่งผลให้สามารถติดตามการเปลี่ยนแปลงของแรงดันไฟฟ้า กระแสไฟฟ้า กำลังไฟฟ้า และพลังงานไฟฟ้าระหว่างกระบวนการชาร์จได้อย่างต่อเนื่อง ทั้งนี้ข้อมูลทั้งหมดถูกแยกตามโปรไฟล์การชาร์จของแบตเตอรี่แต่ละระบบ เช่น 24V5A, 48V5A, 60V5A และ 72V15A เพื่อให้สามารถเปรียบเทียบพฤติกรรมของระบบชาร์จในแต่ละเงื่อนไขได้อย่างชัดเจน

โครงสร้างของระบบฐานข้อมูลสำหรับใช้ร่วมกับ Grafana แสดงดังรูปที่ 3.21 โดยภายในงานวิจัยนี้เลือกใช้ InfluxDB เป็นฐานข้อมูลหลักสำหรับจัดเก็บข้อมูล Time-series จากระบบชาร์จ ทั้งนี้ข้อมูลจาก Node-RED จะถูกส่งเข้าสู่ InfluxDB ผ่าน MQTT และ Modbus RTU ก่อนถูกจัดเก็บในรูปแบบ Measurement เพื่อใช้สำหรับวิเคราะห์ข้อมูลของระบบชาร์จในระยะยาว



รูปที่ 3.21 ระบบฐานข้อมูล InfluxDB สำหรับจัดเก็บข้อมูลแบบ Time-series ของระบบชาร์จ

จากรูปที่ 3.21 จะเห็นได้ว่า InfluxDB ถูกใช้เป็นศูนย์กลางสำหรับจัดเก็บข้อมูลจากระบบชาร์จแบตเตอรี่ โดยข้อมูลที่ถูกรวบรวมจะประกอบด้วยค่าแรงดันไฟฟ้า กระแสไฟฟ้า กำลังไฟฟ้า พลังงานไฟฟ้า และข้อมูล CAN ID ของแบตเตอรี่แต่ละระบบ ทั้งนี้การจัดเก็บข้อมูลในลักษณะ time-series data ช่วยให้สามารถวิเคราะห์พฤติกรรมของระบบชาร์จย้อนหลัง ติดตามแนวโน้มของการใช้พลังงาน และตรวจสอบความต่อเนื่องของ execution layer ภายในระบบได้อย่างมีประสิทธิภาพ

ภายใน Grafana งานวิจัยนี้ได้กำหนดให้ InfluxDB ทำหน้าที่เป็น Data Source สำหรับเชื่อมโยงข้อมูลจากฐานข้อมูลเข้าสู่ Dashboard โดยระบบจะทำการดึงข้อมูลจาก Measurement ของแต่ละโปรไฟล์การชาร์จมาใช้สำหรับสร้างกราฟและเครื่องมือแสดงผลแบบ

Real-time ทั้งนี้ การกำหนด Data Source ของ Grafana ช่วยให้ระบบสามารถเชื่อมโยงข้อมูลจาก Supervisory layer เข้ากับ Visualization layer ได้อย่างสมบูรณ์ ดังแสดงในรูปที่ 3.22

รูปที่ 3.22 การกำหนด Data Source ของ Grafana สำหรับเชื่อมต่อฐานข้อมูล InfluxDB

Grafana ถูกพัฒนาขึ้นเพื่อใช้เป็น Visualization layer สำหรับแสดงผลข้อมูลของระบบชาร์จแบตเตอรี่ NMC แรงดันต่ำแบบ Real-time โดยระบบจะเชื่อมต่อกับฐานข้อมูล InfluxDB ผ่าน Flux Query Language เพื่อใช้สำหรับดึงข้อมูล Time-series จาก Supervisory layer มาแสดงผลบน Dashboard ทั้งนี้ข้อมูลจากแต่ละโปรไฟล์การชาร์จจะถูกจัดเก็บแยกตาม Bucket ของ InfluxDB เช่น 24V5A, 48V5A, 60V5A และ 72V15A ส่งผลให้ระบบสามารถติดตามพฤติกรรมของการใช้พลังงานในแต่ละเงื่อนไขได้อย่างชัดเจน นอกจากนี้ Grafana ยังสามารถดึงข้อมูลด้านแรงดันไฟฟ้า กระแสไฟฟ้า กำลังไฟฟ้า และพลังงานไฟฟ้าสะสมมาแสดงผลในรูปแบบ Time-series Data ได้อย่างต่อเนื่อง ทำให้สามารถติดตามสถานะของระบบชาร์จได้แบบ Real-time ภายใต้ Supervisory system

นอกจากนี้ Grafana Dashboard ยังถูกใช้สำหรับวิเคราะห์พฤติกรรมของระบบฟิกส์ CC-CV ภายในกระบวนการชาร์จแบตเตอรี่ โดยในช่วง Constant Current (CC) ระบบจะสามารถสังเกตเห็นค่ากระแสไฟฟ้าคงที่ ขณะที่แรงดันไฟฟ้าจะเพิ่มขึ้นตามช่วงเวลาของการชาร์จ ส่วนในช่วง Constant Voltage (CV) ค่าแรงดันไฟฟ้าจะคงที่และกระแสไฟฟ้าจะค่อย ๆ ลดลงตามลักษณะของการชาร์จแบตเตอรี่ลิเทียมไอออน ทั้งนี้ข้อมูลที่ถูกจัดเก็บภายใน InfluxDB และแสดงผล

ผ่าน Grafana ยังช่วยให้สามารถตรวจสอบความต่อเนื่องของ Execution layer วิเคราะห์แนวโน้มของการใช้พลังงาน และใช้สำหรับประเมินสมรรถนะของระบบชาร์จในแต่ละโปรไฟล์ได้อย่างมีประสิทธิภาพ โดยผลการแสดงกราฟและการวิเคราะห์พฤติกรรมของระบบชาร์จจะถูกนำเสนอเพิ่มเติมในบทถัดไป

3.7 สรุป

บทนี้นำเสนอขั้นตอนการออกแบบและพัฒนาระบบชาร์จแบตเตอรี่ NMC แรงดันต่ำที่ใช้สถาปัตยกรรมการสื่อสารและควบคุมแบบบูรณาการ ตั้งแต่การระบุโปรไฟล์การชาร์จผ่าน CANopen การส่งข้อมูลด้วย MQTT การประมวลผลและควบคุมผ่าน Node-RED และ Mini PLC-H3 (redPLC) ตลอดจนการเลือกวงจรพิกส์ CC-CV ผ่าน Relay RS485 ด้วยโปรโตคอล Modbus RTU นอกจากนี้ยังครอบคลุมการตรวจวัดและจัดเก็บข้อมูลพลังงานทั้งฝั่ง AC และ DC ผ่าน PM2200 และ PZEM-017 รวมถึงการจัดเก็บข้อมูลในฐานข้อมูล InfluxDB และการแสดงผลผ่าน Grafana Dashboard เพื่อใช้ในการติดตาม วิเคราะห์ และประเมินพฤติกรรมของระบบชาร์จแบบ Real-time โครงสร้างที่พัฒนาขึ้นช่วยให้ระบบสามารถรองรับแบตเตอรี่หลายโปรไฟล์ภายในตู้ชาร์จเดียวกันได้อย่างเป็นลำดับ มีความต่อเนื่องในการทำงาน และสามารถเชื่อมโยง Communication layer, Execution layer และ Visualization layer เข้าด้วยกันได้อย่างสมบูรณ์ ซึ่งเป็นพื้นฐานสำคัญสำหรับการวิเคราะห์ผลการทดลองและประเมินสมรรถนะของระบบในบทถัดไป ทั้งนี้ตัวอย่างโปรแกรมและซอร์สโค้ดที่ใช้สำหรับการสื่อสาร CANopen การประมวลผลบน Node-RED และการควบคุมอุปกรณ์ภายในระบบ จะถูกนำเสนอเพิ่มเติมในภาคผนวก ข ของวิทยานิพนธ์ฉบับนี้