

การพัฒนาระบบควบคุมและสื่อสารยานยนต์ไฟฟ้าด้วยระบบ CANBUS ใน
ยานยนต์ไฟฟ้าดัดแปลง



นายวัฒนา นามบุญลือ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมเมคคาทรอนิกส์
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2567

DEVELOPMENT OF A CONTROL AND COMMUNICATION SYSTEM
FOR ELECTRIC VEHICLE CONVERSION USING CAN BUS



A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Master of Engineering in Mechatronics Engineering
Suranaree University of Technology
Academic Year 2024

การพัฒนาระบบควบคุมและสื่อสารยานยนต์ไฟฟ้าด้วยระบบ CANBUS ในยานยนต์ ไฟฟ้าดัดแปลง

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้บัณฑิตวิทยาลัยฉบับนี้เป็นส่วนหนึ่งของการศึกษา
ตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์



(ผศ. ดร.อุเทน ลีตัน)

ประธานกรรมการ



(รศ. ดร.จิระพล ศรีเสรีสุผล)

กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)



(รศ. ดร.กานต์ เกิดชื่น)

กรรมการ



(รศ. ดร.ยุพาพร รักษกุลวัฒน์)

รองอธิการบดีฝ่ายวิชาการและประกันคุณภาพ



(รศ. ดร.พรศิริ จงกล)

คณบดีสำนักวิชาวิศวกรรมศาสตร์

พัฒนา นามบุญลือ : การพัฒนาระบบควบคุมและสื่อสารยานยนต์ไฟฟ้าด้วยระบบ CANBUS
ในยานยนต์ไฟฟ้าดัดแปลง (DEVELOPMENT OF A CONTROL AND COMMUNICATION
SYSTEM FOR ELECTRIC VEHICLE CONVERSION USING CAN BUS)

อาจารย์ที่ปรึกษา: รองศาสตราจารย์ ดร.จิระพล ศรีเสริมผล, 93 หน้า.

คำสำคัญ : ระบบสื่อสารในยานยนต์ (CAN BUS)/ยานยนต์ไฟฟ้าดัดแปลง (EV Conversion)/ระบบ
ควบคุมมอเตอร์ไฟฟ้า

งานวิจัยนี้มุ่งเน้นการพัฒนาระบบควบคุมและสื่อสารยานยนต์ไฟฟ้าด้วยระบบ CAN BUS ในยานยนต์ไฟฟ้าดัดแปลง (EV Conversion) เนื่องจากกล่องควบคุมมอเตอร์ไฟฟ้าที่มีขายตามท้องตลาดนั้นไม่สามารถควบคุมโหมดการทำงาน ความเร็วรอบของมอเตอร์ไฟฟ้า ความปลอดภัยในการใช้งาน รวมถึงการจัดการพลังงานภายในตัวรถได้ ดังนั้น งานวิจัยนี้จึงใช้ระบบ CAN BUS ร่วมกับกล่องควบคุม CAN I/O PLC และซอฟต์แวร์ MRS Developers Studio เพื่อควบคุมและสั่งงานระบบต่าง ๆ ของยานยนต์ผ่านการสื่อสารแบบ Controller Area Network (CAN) ตามมาตรฐาน SAE J1939 ซึ่งช่วยให้สามารถสั่งการอุปกรณ์ต่าง ๆ ภายในยานยนต์ เช่น มอเตอร์ไฟฟ้า สัญญาณไฟระบบเบรก การทำงานของคันเร่ง และโหมดการทำงานอย่างมีประสิทธิภาพ โดยการทดสอบและทดลองได้ทำการจำลองการทำงานบนรถ ATV เพื่อวิเคราะห์ความเสถียรและความแม่นยำของการสื่อสารระหว่างโมดูลควบคุมต่าง ๆ ผ่านระบบ CAN BUS ผลการทดสอบแสดงให้เห็นว่าระบบ CAN BUS สามารถทำงานได้อย่างเสถียร และสามารถส่งข้อมูลระหว่างโมดูลควบคุมได้อย่างถูกต้องและรวดเร็ว การควบคุมมอเตอร์ไฟฟ้าและระบบต่าง ๆ ของรถ ATV ทำได้ตามที่กำหนด หลังจากปรับสัญญาณ Pulse Width Modulation (PWM) ด้วยวงจร Low Pass Filter ให้เป็นสัญญาณอนาล็อกและมีการวัดกำลังไฟฟ้าด้วย DC Power meter ร่วมกับ V Box เพื่อเก็บข้อมูลการใช้พลังงานแบบเรียลไทม์ นอกจากนี้ ระบบยังสามารถแสดงผลสถานะการทำงานแบบเรียลไทม์บนจอ ATD3.5-S3 ซึ่งช่วยในการตรวจสอบและวิเคราะห์สถานะต่าง ๆ ของรถได้อย่างแม่นยำ งานวิจัยนี้แสดงให้เห็นถึงศักยภาพของระบบ CAN BUS ในการควบคุมยานยนต์ไฟฟ้าดัดแปลง และสามารถนำไปประยุกต์ใช้กับยานยนต์ประเภทอื่น ๆ ได้ในอนาคต

สาขาวิชา วิศวกรรมเมคคาทรอนิกส์
ปีการศึกษา 2567

ลายมือชื่อนักศึกษา
ลายมือชื่ออาจารย์ที่ปรึกษา.....

WATTANA NAMBUNLUE : DEVELOPMENT OF A CONTROL AND COMMUNICATION
SYSTEM FOR ELECTRIC VEHICLE CONVERSION USING CAN BUS.

THESIS ADVISOR : ASSOC. PROF. JIRAPHON SRISERTPOL, Ph.D., 93 PP.

Keywords: CAN BUS Communication System/Electric Vehicle Conversion/Electric Motor
Control System

This research focuses on the development of a control and communication system for electric vehicles using the CAN BUS system in modified electric vehicles (EV Conversion). The commercial motor control units available in the market are unable to control the operating mode, motor speed, safety features, and power management within the vehicle. Therefore, this research utilizes the CAN BUS system in conjunction with the CAN I/O PLC control unit and MRS Developers Studio software to control and command various systems of the vehicle through Controller Area Network (CAN) communication according to the SAE J1939 standard. This enables efficient control of devices within the vehicle, such as the electric motor, signal lights, braking system, throttle operation, and operating modes. The testing and experiments were conducted by simulating the operation on an ATV to analyze the stability and accuracy of communication between control modules via the CAN BUS system. The test results demonstrated that the CAN BUS system operated stably and was able to transmit data between control modules accurately and rapidly. The control of the electric motor and other systems in the ATV worked as specified after adjusting the Pulse Width Modulation (PWM) signal with a Low Pass Filter circuit to convert it to an analog signal. Power consumption was measured using a power meter along with a V Box for real-time energy usage data collection. Additionally, the system was able to display real-time status on an ATD3.5-S3 screen, allowing accurate monitoring and analysis of the vehicle's status. This research showcases the potential of the CAN BUS system in controlling modified electric vehicles and can be applied to other types of vehicles in the future.

School of Mechatronics Engineering
Academic Year 2024

Student's Signature.....Wattana
Advisor's Signature.....Srisertpol

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงด้วยดี เนื่องจากได้รับความช่วยเหลืออย่างยิ่ง ทั้งด้านวิชาการและด้านการดำเนินงานวิจัย จากบุคคลและกลุ่มบุคคลต่าง ๆ ได้แก่

ขอขอบคุณ รองศาสตราจารย์ ดร.จิระพล ศรีเสรีฐผล ซึ่งเป็นที่ปรึกษาหลักของวิทยานิพนธ์นี้ รวมถึงผู้ช่วยศาสตราจารย์ ดร.อุเทน ลีตัน ซึ่งให้คำปรึกษาและแนะนำการวิจัยเกี่ยวกับ CANBUS ตลอดจนสละเวลามอบคำแนะนำและความคิดเห็นที่เป็นประโยชน์อย่างยิ่งในด้านการวางแผนวิจัย การปรับปรุงผลงาน และการนำเสนองาน ทำให้ผู้จัดทำสามารถวิเคราะห์และวางแผนวิจัยได้อย่างครบถ้วนสมบูรณ์

ผู้จัดทำหวังเป็นอย่างยิ่งว่างานวิจัยนี้จะเป็นประโยชน์ไม่เพียงต่อคณาจารย์ที่เกี่ยวข้อง แต่ยังต่อทุกท่านที่ได้เข้ามาศึกษา หากมีข้อบกพร่องใด ๆ ที่เกิดขึ้น ผู้จัดทำขอน้อมรับไว้เพื่อปรับปรุงในโอกาสต่อไป และยินดีรับฟังคำแนะนำจากทุกท่าน เพื่อเป็นแนวทางในการพัฒนางานวิจัยต่อไปในอนาคต

วัฒนา นามบุญลือ

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญ

หน้า

บทคัดย่อ (ภาษาไทย).....	ก
บทคัดย่อ (ภาษาอังกฤษ).....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	
สารบัญตาราง.....	
สารบัญรูป.....	
คำอธิบายสัญลักษณ์และคำย่อ	
บทที่	
1 บทนำ	1
1.1 ความสำคัญและที่มาของปัญหา	1
1.2 วัตถุประสงค์การวิจัย.....	2
1.3 ขอบเขตของงานวิจัย	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
2 ปรัชศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง	4
2.1 บทนำ.....	4
2.2 ระบบสื่อสารในยานยนต์ CAN BUS (CONTROLLER AREA NETWORK)	4
2.3 ระดับของ CAN BUS.....	5
2.4 ประเภทของ CAN BUS.....	6
2.5 การใช้งานมาตรฐาน SAE J1939 ร่วมกับ CAN BUS	8
2.6 การประยุกต์ CAN I/O PLC ร่วมกับ MRS DEVELOPERS STUDIO	8
2.7 หลักการและการประยุกต์ใช้ตัวกรองความถี่ต่ำ (LOW-PASS FILTER).....	8
2.8 จอแสดงผลสถานะการทำงานของ ATV.....	8
2.9 ซอฟต์แวร์ SQUARELINE STUDIO	8
2.10 โปรแกรม VISUAL STUDIO CODE พร้อม PLATFORM IO EXTENSION.....	8
2.11 ปรัชศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง	8

สารบัญ (ต่อ)

หน้า

3	วิธีการดำเนินการวิจัย	21
3.1	บทนำ.....	21
3.2	การออกแบบระบบสื่อสารในยานยนต์ CAN BUS (CONTROLLER AREA NETWORK)..	21
3.3	การออกแบบการทำงานของยานยนต์	23
3.4	การกรองสัญญาณ PWM เป็นสัญญาณอนาล็อก.....	29
3.5	จอแสดงผลของระบบ CAN BUS.....	29
3.6	การวัดกำลังไฟ โดยใช้ V-BOX ร่วมกับ POWER METER	30
3.7	ขั้นตอนการทดลอง.....	31
4	ผลการวิจัยและอภิปรายผล	32
4.1	บทนำ.....	48
4.2	การดำเนินการเขียนโปรแกรมรถ ATV	33
4.3	การเชื่อมต่อระบบ CAN BUS ระหว่างโมดูลต่างๆสำหรับการแสดงผล	36
4.4	ประสิทธิภาพการควบคุมมอเตอร์ไฟฟ้า	44
4.5	การวิเคราะห์และอภิปรายผลการดำเนินงานวิจัย	46
5	สรุปและข้อเสนอแนะ	48
5.1	บทนำ.....	48
5.2	สรุปผลการศึกษา	48
5.3	ข้อเสนอแนะ	49
	รายการอ้างอิง	50
	ภาคผนวก	
	ภาคผนวก ก. ขั้นตอนการดำเนินงาน	51
	ภาคผนวก ข. บทความที่ได้รับการตีพิมพ์เผยแพร่	60
	ประวัติผู้เขียน	93

คำอธิบายสัญลักษณ์และคำย่อ

EV Conversion	=	การแปลงรถยนต์ที่ใช้น้ำมันให้เป็นรถยนต์ไฟฟ้า (Electric Vehicle Conversion)
CAN	=	เป็นโปรโตคอลการสื่อสารที่ใช้ในรถยนต์ (Controller Area Network)
PWM	=	การปรับความกว้างพัลส์ (Pulse Width Modulation)
ATV	=	ยานพาหนะที่วิ่งได้ในทุกสภาพภูมิประเทศ (All-Terrain Vehicle)
LPF	=	วงจรกรองความถี่ต่ำ (Low Pass Filter)
PLC	=	ตัวควบคุมลอจิกแบบโปรแกรมได้ (Programmable Logic Controller)
V	=	หน่วยของแรงดันไฟฟ้า
Hz	=	หน่วยของความถี่
W	=	หน่วยของกำลังไฟฟ้า(วัตต์)
s	=	หน่วยของเวลา(วินาที)

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของปัญหา

ปัจจุบัน รถยนต์ไฟฟ้าได้รับความนิยมอย่างแพร่หลายด้วย เทคโนโลยีที่พัฒนาอย่างต่อเนื่อง ซึ่งช่วยเพิ่มประสิทธิภาพและสมรรถนะ ของตัวรถ อีกทั้งยังช่วย ลดการปล่อยก๊าซเรือนกระจกและมลพิษ ทำให้ผู้บริโภคหันมาเลือกใช้มากขึ้น ข้อดีสำคัญของรถยนต์ไฟฟ้าคือ ความประหยัดพลังงาน เนื่องจาก ต้นทุนพลังงานไฟฟ้าต่ำกว่าน้ำมันอย่างมีนัยสำคัญ ช่วยลดค่าใช้จ่ายในการเดินทาง นอกจากนี้ ยังมีผู้บริโภคที่ต้องการดัดแปลงรถยนต์สันดาปให้เป็นรถยนต์ไฟฟ้า เพื่อรักษารถรุ่นเก่าที่ตนชื่นชอบไว้ พร้อมทั้ง ส่งเสริมการใช้พลังงาน ทำให้ความดัดแปลงรถยนต์สันดาปให้เป็นยานยนต์ไฟฟ้าดัดแปลง (EV Conversion) เพิ่มขึ้นอย่างต่อเนื่อง

ในการดัดแปลงรถยนต์สันดาปให้เป็นยานยนต์ไฟฟ้าดัดแปลง (EV Conversion) พบว่ากล่องควบคุมมอเตอร์ที่มีจำหน่ายตามท้องตลาดยังไม่สามารถตอบสนองต่อความต้องการของผู้ใช้งานได้อย่างมีประสิทธิภาพ เนื่องจากไม่สามารถปรับโหมดการทำงานให้เหมาะสมกับลักษณะการขับขี่ที่แตกต่างกัน ส่งผลให้เกิดการสูญเสียพลังงานแบตเตอรี่มากกว่าที่ควร เช่น ในกรณีที่ต้องการใช้พลังงานอย่างประหยัด กล่องควบคุมมอเตอร์ควรสามารถปรับกำลังการขับเคลื่อนให้อยู่ในระดับที่เหมาะสมกับโหมดการใช้งาน นอกจากนี้ระบบความปลอดภัยของกล่องควบคุมมอเตอร์ทั่วไปยังไม่ครอบคลุม เช่น ไม่มีระบบตัดการทำงานของคันเร่งเมื่อเหยียบเบรก ซึ่งอาจทำให้รถยังคงเคลื่อนที่แม้ว่าผู้ขับขี่ต้องการหยุดรถ ทำให้เกิดความเสี่ยงต่ออุบัติเหตุ อีกทั้งยังไม่มีโหมดเกียร์ว่าง ซึ่งเป็นฟังก์ชันที่ช่วยป้องกันไม่ให้มอเตอร์ทำงานโดยไม่ตั้งใจ หากไม่มีระบบนี้รถอาจเกิดการเคลื่อนที่ทันทีเมื่อเปิดระบบไฟฟ้าโดยไม่ได้ตั้งใจ ส่งผลให้เกิดอุบัติเหตุที่ไม่คาดคิด ดังนั้นการพัฒนากล่องควบคุมมอเตอร์ที่สามารถปรับโหมดการทำงานได้ และมีระบบความปลอดภัยที่เหมาะสมกับการใช้งานจริงจึงเป็นสิ่งสำคัญในการทำ EV Conversion เพื่อให้มั่นใจว่า ยานยนต์ไฟฟ้าดัดแปลงมีประสิทธิภาพสูงสุด ใช้พลังงานอย่างคุ้มค่า และสามารถขับขี่ได้อย่างปลอดภัย

ด้วยเหตุนี้ งานวิจัยนี้จึงมีวัตถุประสงค์เพื่อพัฒนาการใช้กล่อง CAN I/O PLC ร่วมกับโปรแกรม MRS Developers Studio ในการเขียนโปรแกรมและสร้างระบบการสื่อสารแบบ Controller Area Network (CAN) ซึ่งเป็นมาตรฐานการสื่อสารเครือข่ายในอุตสาหกรรมยานยนต์ โดยผสานมาตรฐาน SAE J1939 สำหรับการควบคุมและสั่งการระบบต่าง ๆ ของยานยนต์ไฟฟ้าดัดแปลง (EV Conversion) ตัวอย่างการใช้งานประกอบด้วยการส่งสัญญาณในรูปแบบอนาล็อกและ

ดิจิทัลเพื่อควบคุมการทำงานและปรับโหมดการขับเคลื่อน การควบคุมความปลอดภัยของผู้ขับขี่ ตลอดจนการสั่งการเปิด-ปิดอุปกรณ์ภายในยานยนต์ เช่น ระบบไฟสัญญาณ ระบบการขับเคลื่อนหน้า-ถอยหลัง และการโหมดการทำงาน เป็นต้น การทดลองในงานวิจัยนี้มุ่งเน้นการเขียนและทดสอบการสื่อสารของระบบ CAN BUS โดยการจำลองการทำงานบนรถ ATV ซึ่งเป็นการทดสอบและวิเคราะห์การทำงานของระบบควบคุมแบบ CAN BUS ในสภาพแวดล้อมที่จำลองขึ้นเพื่อประเมินประสิทธิภาพของระบบทั้งในด้านการควบคุมอุปกรณ์และการปรับปรุงสมรรถนะการทำงานของยานยนต์ในสถานการณ์ที่หลากหลาย การทดสอบนี้ช่วยยืนยันถึงเสถียรภาพของระบบสื่อสาร รวมถึงความแม่นยำในการตอบสนองของอุปกรณ์ควบคุมภายในรถ ATV นอกจากนี้ ยังแสดงสถานะการทำงานผ่านจอ ATD3.5-S3-CAN BUS Shield เพื่อเสริมความชัดเจนและความน่าเชื่อถือของข้อมูล

1.2 วัตถุประสงค์การวิจัย

1.2.1 เพื่อออกแบบและพัฒนาระบบควบคุมมอเตอร์ไฟฟ้าสำหรับรถไฟฟ้าที่แปลงจากรถยนต์สันดาปภายใน โดยคำนึงถึงความเหมาะสมในการใช้งานและประสิทธิภาพของระบบขับเคลื่อน

1.2.2 เพื่อออกแบบโปรแกรมการทำงานต่าง ๆ ที่ใช้ควบคุมระบบภายในรถยนต์ไฟฟ้า รวมถึงการควบคุมการทำงานของอุปกรณ์ต่าง ๆ ผ่านการสื่อสารแบบ Controller Area Network (CAN)

1.2.3 เพื่อวิเคราะห์และปรับปรุงการจัดการพลังงานของรถยนต์ไฟฟ้าดัดแปลงให้มีประสิทธิภาพสูงสุดโดยรูปแบบการสื่อสารแสดงสถานะต่าง ๆ ผ่านจอแสดงผลได้

1.2.4 เพื่อประเมินการทำงานของระบบควบคุมที่ได้รับการพัฒนาในสภาพแวดล้อมการใช้งานจริง โดยทดสอบในสถานการณ์ต่าง ๆ เพื่อให้แน่ใจว่าระบบสามารถตอบสนองต่อการขับขี่ได้อย่างมีประสิทธิภาพ

1.3 ขอบเขตของงานวิจัย

1.3.1 ออกแบบวงจรไฟฟ้าและระบบควบคุมภายในรถยนต์ไฟฟ้าดัดแปลง โดยให้สอดคล้องกับข้อกำหนดและมาตรฐานด้านการใช้งาน

1.3.2 พัฒนาการสื่อสารในระบบควบคุมรถยนต์โดยใช้เทคโนโลยี Control Area Network (CAN) ซึ่งทำงานร่วมกับกล่อง CAN I/O ของ MRS เพื่อให้เกิดการสื่อสารที่มีประสิทธิภาพ

1.3.3 ใช้มาตรฐาน SAE J1939 เป็นรูปแบบการสื่อสารกับอุปกรณ์ภายนอก เพื่อให้การควบคุมและการตอบสนองต่อระบบต่าง ๆ ในยานยนต์เป็นไปอย่างแม่นยำและรวดเร็ว

1.3.4 ทดสอบและปรับปรุงระบบควบคุมในสภาพแวดล้อมการใช้งานที่แตกต่างกัน เช่น การขับขึ้นทางลาดชัน ทางราบ และสภาพถนนที่มีความท้าทาย เพื่อให้แน่ใจว่าระบบสามารถทำงานได้ดีในทุกสถานการณ์

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 ได้รับระบบควบคุมสำหรับยานยนต์ไฟฟ้าที่มีประสิทธิภาพสูง สามารถตอบสนองต่อการใช้งานได้อย่างเหมาะสม และช่วยเพิ่มความมั่นใจในด้านความปลอดภัยสำหรับผู้ใช้งาน

1.4.2 ได้วิธีการเดินระบบไฟฟ้าภายในยานยนต์ไฟฟ้าที่มีความซับซ้อนน้อยลง ซึ่งจะทำให้การติดตั้งและการบำรุงรักษาทำได้ง่ายขึ้น ช่วยลดเวลาและค่าใช้จ่ายในการดำเนินการ

1.4.3 พัฒนารูปแบบการส่งสัญญาณอนาล็อกและดิจิทัลที่ตอบสนองได้รวดเร็วและแม่นยำมากขึ้น ช่วยให้การควบคุมอุปกรณ์ต่าง ๆ ภายในรถยนต์ไฟฟ้าเป็นไปอย่างมีประสิทธิภาพ



บทที่ 2

ปรัทัศนัวรรณกรรมและงานวิจัยที่เกี่ยวข้อง

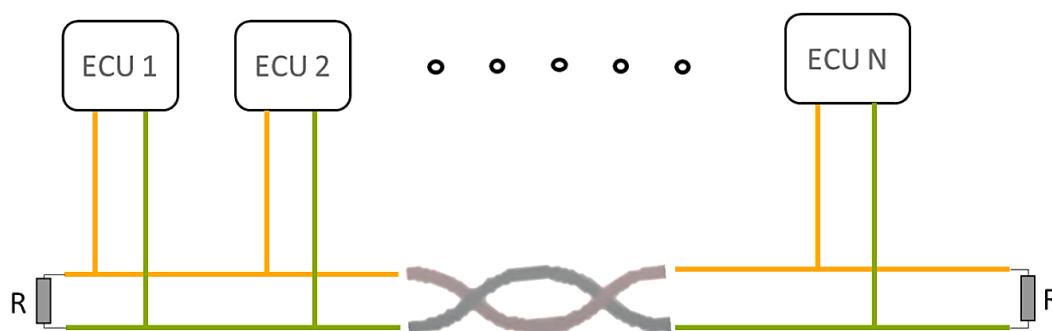
2.1 บทนำ

(Donjaroennon, 2024) การพัฒนาระบบควบคุมการสื่อสารภายในยานพาหนะไฟฟ้า ATV โดยใช้เทคโนโลยี CAN BUS และซอฟต์แวร์ MRS Developers Studio เพื่อสร้างการควบคุมที่มีประสิทธิภาพและเชื่อถือได้สำหรับการทำงานของฟังก์ชันต่าง ๆ ในยานพาหนะ เช่น เครื่องยนต์ ไฟหน้า และเกียร์ ผ่านการกำหนดค่าและจำลองระบบที่สอดคล้องกับมาตรฐาน J1939 ซึ่งจะช่วยเพิ่มประสิทธิภาพ ความปลอดภัย และลดการบำรุงรักษา รองรับการพัฒนานวัตกรรมในยานพาหนะไฟฟ้าในอนาคต

การพัฒนาประสิทธิภาพการสื่อสารระหว่างระบบภายในของยานยนต์ถือเป็นปัจจัยสำคัญที่ส่งผลกระทบต่อเพิ่มประสิทธิภาพและความปลอดภัยของยานยนต์ เทคโนโลยีหลักที่มีบทบาทสำคัญในการสนับสนุนการสื่อสารระหว่างระบบย่อยในยานยนต์คือระบบ CAN BUS ซึ่งได้รับการออกแบบมาเพื่อให้สามารถแลกเปลี่ยนข้อมูลระหว่างอุปกรณ์อิเล็กทรอนิกส์ภายในยานยนต์ได้อย่างรวดเร็วและมีประสิทธิภาพ ระบบ CAN BUS ได้รับการยอมรับอย่างแพร่หลายในอุตสาหกรรมยานยนต์เนื่องจากความน่าเชื่อถือและความทนทานในสภาวะแวดล้อมที่มีความรุนแรง เทคโนโลยีนี้ช่วยรับประกันว่าการส่งข้อมูลระหว่างหน่วยควบคุมอิเล็กทรอนิกส์ (ECU) จะเป็นไปอย่างต่อเนื่องและปลอดภัย ดังนั้นการพัฒนาซอฟต์แวร์ที่สามารถรองรับการทำงานของระบบ CAN BUS จึงมีความสำคัญในการยกระดับประสิทธิภาพการทำงานของยานยนต์ไฟฟ้า

2.2 ระบบสื่อสารในยานยนต์ CAN BUS (Controller Area Network)

ระบบสื่อสาร CAN BUS (Controller Area Network) เป็นระบบการสื่อสารแบบอนุกรมที่ได้รับการออกแบบเฉพาะสำหรับการส่งข้อมูลแบบสองทางภายในระบบควบคุมของยานยนต์ โดยระบบ CAN BUS ทำหน้าที่เป็นสื่อกลางในการแลกเปลี่ยนข้อมูลระหว่างหน่วยควบคุมอิเล็กทรอนิกส์ (ECU) ซึ่งเชื่อมต่อกันผ่านสายคู่บิดเกลียวแสดงดังรูปที่ 2.1 ระบบนี้สามารถรองรับอัตราการส่งข้อมูลได้สูงสุดถึง 1 เมกะบิตต่อวินาที ขึ้นอยู่กับความยาวของบัส สอดคล้องกับความต้องการด้านการสื่อสารที่รวดเร็ว โดยมีการรับประกันความล่าช้าในการส่งข้อมูลต่ำ อีกทั้งยังมีระบบตรวจจับข้อผิดพลาดที่มีประสิทธิภาพสูงเพื่อเสริมความน่าเชื่อถือของการสื่อสาร

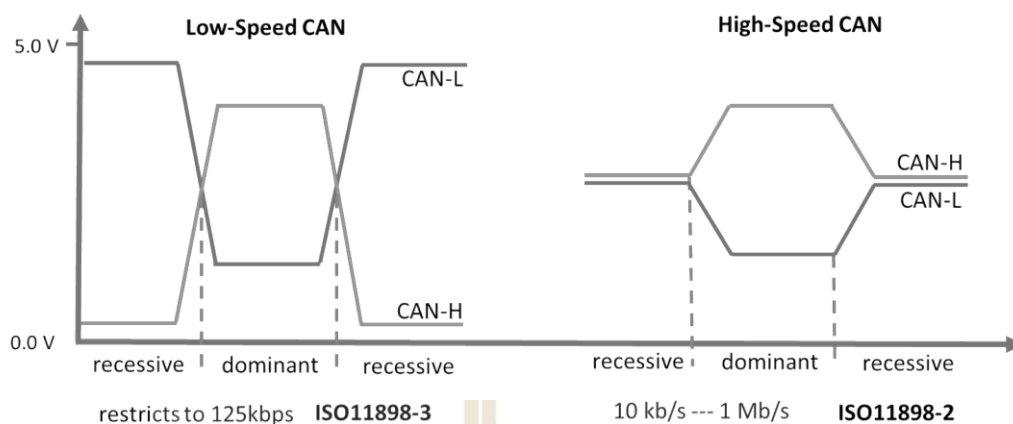


รูปที่ 2.1 ระบบสื่อสาร CAN BUS (Controller Area Network)

ระบบ CAN BUS เป็นบัสแบบหลายผู้ควบคุม (Multi-Master) ซึ่งหมายความว่าแต่ละโหนดในเครือข่ายสามารถทำหน้าที่เป็นทั้งผู้ส่งและผู้รับข้อมูลได้ โดยไม่จำเป็นต้องพึ่งพาคอมพิวเตอร์แม่ข่ายกลาง โหนดแต่ละโหนดสามารถประกอบด้วยไมโครคอนโทรลเลอร์ คอมพิวเตอร์ฝังตัว อุปกรณ์ I/O หรืออุปกรณ์เชื่อมต่อเครือข่ายอื่น ๆ คุณสมบัตินี้ทำให้การสื่อสารระหว่างอุปกรณ์ต่าง ๆ ภายในยานยนต์เป็นไปอย่างอิสระและมีประสิทธิภาพ เพิ่มความยืดหยุ่นในการเชื่อมต่อและส่งข้อมูลระหว่างอุปกรณ์

2.3 ระดับของ CAN BUS

ระบบ CAN BUS แบ่งออกเป็นสองประเภทหลัก ได้แก่ High-Speed CAN และ Low-Speed CAN โดย High-Speed CAN ซึ่งเป็นไปตามมาตรฐาน ISO 11898-2 รองรับอัตราการส่งข้อมูลตั้งแต่ 10 kbit/s ถึง 1 Mbit/s ขณะที่ Low-Speed CAN ตามมาตรฐาน ISO 11898-3 มีความเร็วจำกัดอยู่ที่ 125 kbps แสดงดังรูปที่ 2.2 ความแตกต่างสำคัญระหว่างสองระบบนี้ คือ การใช้ตัวต้านทาน (Termination Resistance) โดย High-Speed CAN ต้องการตัวต้านทานขนาด 120 โอห์มที่ปลายบัส เพื่อช่วยลดคลื่นสะท้อนและเพิ่มความเร็วในการส่งข้อมูล ในขณะที่ Low-Speed CAN มีการปิดตัวต้านทานที่ทุกอุปกรณ์ เพื่อป้องกันการเกิดคลื่นสะท้อนและเพิ่มความสามารถในการทนทานต่อความผิดพลาด (Fault Tolerant) ทำให้สามารถทำงานได้อย่างต่อเนื่องแม้ในกรณีที่เกิดปัญหาในระบบสายสัญญาณ



รูปที่ 2.2 สัญญาณ Low-Speed CAN และสัญญาณ High-Speed CAN

2.4 ประเภทของ CAN BUS

2.4.1 CAN 2.0A

มาตรฐาน CAN 2.0A เป็นหนึ่งในประเภทของระบบ CAN BUS ที่ได้รับการใช้งานอย่างแพร่หลาย โดยมาตรฐานนี้ใช้ตัวระบุข้อมูลขนาด 11 บิต (11-bit Identifier) ซึ่งมักเรียกว่ารูปแบบ Standard Frame ที่แสดงดังรูปที่ 2.3 การทำงานของ CAN 2.0A มุ่งเน้นที่ความสามารถในการส่งข้อมูลอย่างรวดเร็วและมีประสิทธิภาพสูง ทั้งยังมีค่าใช้จ่ายในการพัฒนาที่ต่ำกว่าเมื่อเปรียบเทียบกับมาตรฐาน CAN 2.0B ซึ่งใช้ตัวระบุข้อมูลขนาด 29 บิต (29-bit Identifier) โดยการใช้ CAN 2.0A จึงเหมาะสมสำหรับระบบที่ต้องการความเร็วในการส่งข้อมูลและความคุ้มค่าในด้านการพัฒนา



รูปที่ 2.3 โครงสร้างของ CAN 2.0A (Standard Frame)

องค์ประกอบของ Standard Data Frame ใน CAN 2.0A

- 1) SOF (Start of Frame) – เป็นบิตเริ่มต้นของเฟรม มีขนาด 1 บิต
- 2) ID (Identifier) – ใช้ตัวระบุขนาด 11 บิต เพื่อบอกความสำคัญของข้อความที่ส่งในเครือข่าย
- 3) RTR (Remote Transmission Request) – บิตขนาด 1 บิต ใช้สำหรับการขอข้อมูลจากโหนดอื่นในเครือข่าย
- 4) Control Field – มีความยาว 6 บิต ประกอบด้วย 1 บิตที่ใช้สำหรับ IDE (บอกว่าข้อความเป็นแบบ Standard หรือ Extended) และอีก 1 บิตสำรอง รวมถึง 4 บิตสำหรับบอกขนาดของข้อมูล (Data Length Code หรือ DLC)
- 5) Data Field – ข้อมูลที่ส่งผ่าน CAN BUS มีความยาวตั้งแต่ 0 ถึง 64 บิต
- 6) CRC (Cyclic Redundancy Check) – ใช้สำหรับการตรวจสอบความถูกต้องของข้อมูล มีความยาว 16 บิต
- 7) ACK (Acknowledgement) - ใช้ในการยืนยันการรับข้อมูล โดยมีขนาด 2 บิต
- 8) End of Frame – เป็นสัญญาณบอกจบเฟรม ขนาด 7 บิต

2.4.2 CAN 2.0B

มาตรฐาน CAN 2.0B หรือที่รู้จักในชื่อ Extended Frame เป็นมาตรฐานการสื่อสารในระบบ CAN BUS ที่ใช้ตัวระบุข้อมูลขนาด 29 บิตดังรูปที่ 2.4 ทำให้สามารถระบุตัวตนของข้อมูลได้มากกว่า CAN 2.0A ที่ใช้ตัวระบุเพียง 11 บิต การเพิ่มขนาดของตัวระบุนี้ช่วยให้สามารถส่งข้อมูลที่มีความซับซ้อนและมีรายละเอียดมากขึ้นได้ อย่างไรก็ตาม การใช้ตัวระบุขนาดใหญ่ส่งผลให้เกิดการใช้แบนด์วิธที่สูงขึ้น ซึ่งอาจกระทบต่อความเร็วในการส่งข้อมูลในกรณีที่เครือข่ายมีปริมาณข้อมูลหนาแน่น



รูปที่ 2.4 โครงสร้างของ CAN 2.0B (Extended Frame)

องค์ประกอบของ Extended Data Frame (CAN 2.0B)

- 1) SOF (Start of Frame) – ขนาด 1 บิต ใช้สำหรับบอกจุดเริ่มต้นของเฟรม
- 2) ID (Identifier) – ใช้ตัวระบุขนาด 11 บิตเพื่อระบุตัวตนของข้อความ

3) SRR (Substitute Remote Request) – ขนาด 1 บิต ใช้แทนบิต RTR ในเฟรมแบบ Extended Frame

4) IDE (Identifier Extension) – ขนาด 1 บิต ใช้บอกว่าเฟรมนี้เป็นแบบ Extended Frame หากเป็น Extended จะมีการเพิ่มตัวระบุ 18 บิตต่อไป

5) Extended ID (18-bit Identifier) – ข้อมูลเพิ่มเติมของตัวระบุขนาด 18 บิต รวมกับตัวระบุ 11 บิตแรกเป็นตัวระบุขนาด 29 บิตทั้งหมด

6) RTR (Remote Transmission Request) – ขนาด 1 บิต ใช้สำหรับการขอข้อมูลจากโหนดอื่นในเครือข่าย

7) Control Field – ขนาด 6 บิต โดยประกอบไปด้วย 4 บิตสำหรับ Data Length Code (DLC) และ 2 บิตสำรอง

8) Data Field – ข้อมูลที่ส่งผ่าน CAN BUS มีความยาวตั้งแต่ 0 ถึง 64 บิต

9) CRC (Cyclic Redundancy Check) – ใช้สำหรับตรวจสอบความถูกต้องของข้อมูล มีความยาว 16 บิต

10) ACK (Acknowledgement) – ขนาด 2 บิต ใช้ในการยืนยันการรับข้อมูล

11) End of Frame – ขนาด 7 บิต ใช้สำหรับบอกจุดสิ้นสุดของเฟรม

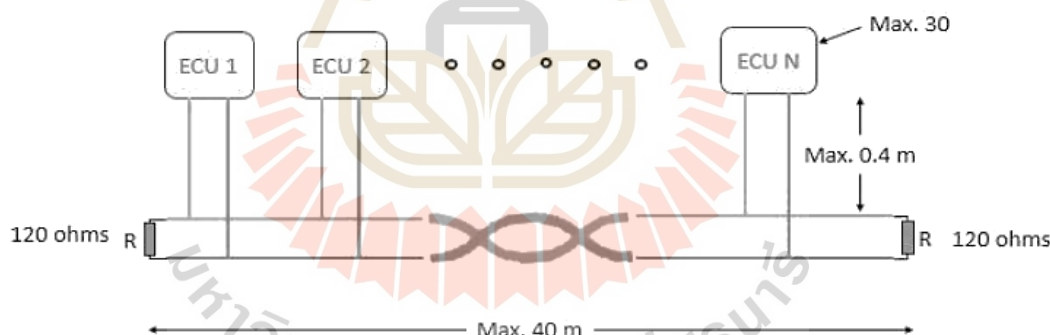
2.4.3 CAN FD (Flexible Data-Rate)

CAN FD เป็นเวอร์ชันที่ได้รับการพัฒนาขึ้นจากระบบ CAN แบบดั้งเดิม เพื่อแก้ไขข้อจำกัดด้านความเร็วและขนาดข้อมูลในการส่งผ่าน CAN BUS ให้มีประสิทธิภาพมากยิ่งขึ้น จุดเด่นสำคัญของ CAN FD คือสามารถเพิ่มอัตราการส่งข้อมูล (Data Rate) ได้สูงกว่ามาตรฐาน CAN 2.0 ทั้งแบบ 2.0A และ 2.0B โดยมีการปรับอัตราการส่งข้อมูลให้รวดเร็วขึ้นในช่วงที่ส่งข้อมูล ทำให้ลดความล่าช้าในการรับส่งได้อย่างมีนัยสำคัญ CAN FD ยังสามารถรองรับการส่งข้อมูลได้สูงสุดถึง 64 ไบต์ต่อเฟรม ซึ่งมากกว่ามาตรฐาน CAN ดั้งเดิมที่รองรับเพียง 8 ไบต์ต่อเฟรม ช่วยให้การรับส่งข้อมูลมีปริมาณมากขึ้นภายในระยะเวลาเดียวกัน อีกทั้ง CAN FD ยังสามารถทำงานร่วมกับระบบ CAN เดิมได้อย่างราบรื่นโดยไม่เกิดปัญหาใด ๆ พร้อมทั้งเพิ่มประสิทธิภาพในการตอบสนองที่รวดเร็ว จึงเหมาะสมอย่างยิ่งสำหรับการใช้งานในระบบที่ต้องการความแม่นยำและประสิทธิภาพสูง เช่น ระบบควบคุมในยานยนต์

2.5 การใช้งานมาตรฐาน SAE J1939 ร่วมกับ CAN BUS

มาตรฐาน SAE J1939 ซึ่งได้รับการพัฒนาโดย Society of Automotive Engineers (SAE) เป็นโพรโตคอลการสื่อสารที่ออกแบบมาเฉพาะสำหรับยานยนต์ขนาดใหญ่ เช่น รถบรรทุกและรถ

โดยสาร โดยใช้พื้นฐานของระบบ CAN BUS ในการอำนวยความสะดวกด้านการสื่อสารระหว่างหน่วยควบคุมอิเล็กทรอนิกส์ (ECU) ผ่าน Parameter Group Numbers (PGNs) ซึ่งเป็นตัวระบุประเภทของข้อมูล เช่น การควบคุมเครื่องยนต์ ระบบเบรก หรือการตรวจวินิจฉัยระบบต่าง ๆ โดยข้อมูลที่มีความสำคัญ เช่น ข้อมูลด้านความปลอดภัย จะได้รับการจัดลำดับความสำคัญและถูกส่งก่อนเพื่อให้การสื่อสารเกิดขึ้นอย่างทันท่วงทีและแม่นยำ มาตรฐานนี้ใช้โครงสร้างเฟรมแบบ 29-bit Extended Frame ทำให้สามารถรองรับข้อมูลที่ซับซ้อนและมีรายละเอียดสูงสุด 8 ไบต์ต่อข้อความ พร้อมทั้งมีระบบตรวจสอบข้อผิดพลาดแบบครบวงจรเพื่อเสริมความน่าเชื่อถือและเสถียรภาพของการสื่อสาร นอกจากนี้ มาตรฐาน SAE J1939 ยังขยายความสามารถของ CAN BUS ให้รองรับการเชื่อมต่อ ECU ได้สูงสุดถึง 30 โหนดต่อเครือข่ายดังรูปที่ 2.5 โดยมีการจัดการที่อยู่ต้นทางและปลายทางของข้อมูลผ่าน Protocol Data Unit (PDU) ซึ่งช่วยในการจัดลำดับความสำคัญของข้อมูล ลดปัญหาการส่งข้อความซ้ำซ้อน และเพิ่มประสิทธิภาพในการสื่อสาร SAE J1939 ได้ถูกนำมาใช้ในยานยนต์ไฟฟ้าดัดแปลง (EV Conversion) เพื่อใช้ในการควบคุมโหมดการทำงานของมอเตอร์ไฟฟ้า ระบบเบรก และการตรวจสอบสถานะพลังงาน นอกจากนี้ยังช่วยเพิ่มประสิทธิภาพในการจัดการข้อมูลด้านความปลอดภัย ซึ่งเหมาะสมกับสภาพแวดล้อมการใช้งานที่มีความท้าทายสูง

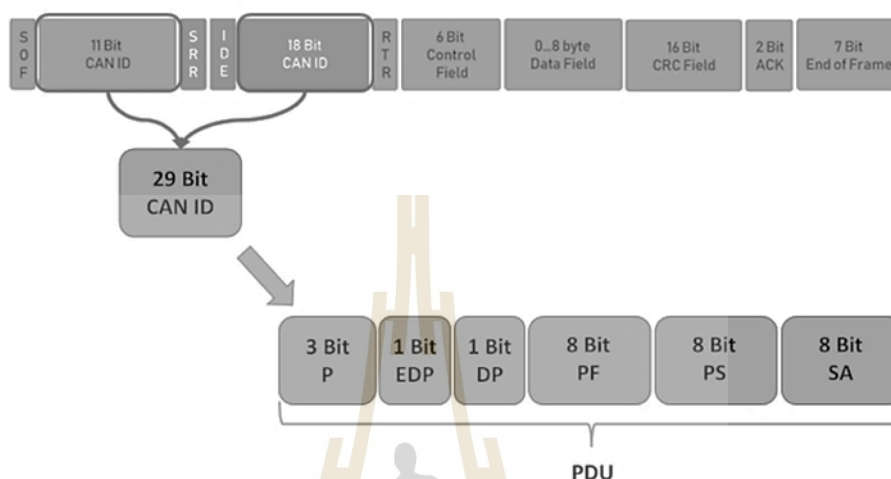


รูปที่ 2.5 การเชื่อมต่อ ECU ของมาตรฐาน SAE J1939

Protocol Data Unit (PDU)

พิจารณารูปที่ 2.4 CAN 2.0B พบว่า Identifier มีขนาด 29 บิต ก่อให้เกิดมาตรฐานที่ถูกกำหนดโดยสมาคมวิศวกรรมยานยนต์เป็น Protocol Data Unit (PDU) ดังรูปที่ 2.6 ที่ใช้สำหรับรถบรรทุกหรือรถบรรทุกขนาดใหญ่ เป็นหน่วยข้อมูลในโปรโตคอล SAE J1939 ที่มีบทบาทสำคัญในการจัดการการสื่อสารในระบบ CAN BUS โดย PDU ประกอบด้วยข้อมูลเกี่ยวกับผู้ส่งและผู้รับของข้อความ ตลอดจนข้อมูลเกี่ยวกับลำดับความสำคัญของข้อความที่ส่ง ซึ่งช่วยให้ระบบสามารถจำแนก

การสื่อสารระหว่างการส่งข้อมูลไปยังอุปกรณ์เป้าหมายที่ระบุ (point-to-point) หรือการส่งข้อมูลแบบกระจายไปยังทุกอุปกรณ์ที่เชื่อมต่อในระบบ (broadcast) การใช้ PDU ส่งผลให้การจัดการข้อมูลมีประสิทธิภาพยิ่งขึ้นและช่วยลดปัญหาความซ้ำซ้อนในการส่งข้อมูล



รูปที่ 2.6 Protocol Data Unit (PDU)

ส่วนประกอบของ PDU

- 1) P (Priority) [3 Bit] ส่วนนี้ระบุระดับความสำคัญของข้อความที่ถูกส่ง ข้อมูลที่มีค่าความสำคัญสูงจะถูกส่งก่อนในกรณีที่มีการสื่อสารหลายรายการในเครือข่าย
- 2) EDP (Extended Data Page) [1 Bit] บิตนี้ใช้สำหรับระบุว่าข้อมูลที่ถูกส่งนั้นอยู่ในโครงสร้างข้อมูลแบบขยายหรือไม่ ส่วนใหญ่จะใช้ในการขยายข้อมูลการสื่อสาร
- 3) DP (Data Page) [1 Bit] บิตนี้ใช้สำหรับการแบ่งแยกข้อมูลในแถบข้อมูลหลักหรือในหน้าเพิ่มเติม การระบุแถบข้อมูลที่แตกต่างกันช่วยให้สามารถจัดการข้อมูลหลายรูปแบบได้
- 4) PF (PDU Format) [8 Bit] ส่วนนี้เป็นฟิลด์ที่ระบุรูปแบบของข้อมูลที่ถูกส่ง สามารถแยกแยะข้อมูลระหว่างการสื่อสารแบบ point-to-point หรือการสื่อสารแบบกระจายสู่ทุกอุปกรณ์ (broadcast)
- 5) PS (PDU Specific) [8 Bit] ฟิลด์นี้จะกำหนดที่อยู่ของผู้รับในกรณีของการสื่อสาร point-to-point ถ้าเป็นการส่งข้อมูลแบบ broadcast ฟิลด์นี้จะกำหนดประเภทของข้อมูลที่จะถูกส่ง
- 6) SA (Source Address) [8 Bit] ฟิลด์นี้ระบุที่อยู่ของผู้ส่งข้อมูล ทำให้สามารถระบุแหล่งที่มาของข้อมูลได้ชัดเจน

7) Data Field [0-64 Bit] ส่วนนี้ คือ ข้อมูลจริงที่ถูกส่งผ่าน PDU ซึ่งอาจมีความยาวตั้งแต่ 0 ถึง 64 บิต ขึ้นอยู่กับประเภทและลักษณะของข้อมูลที่ต้องการสื่อสาร

Parameter Group Numbers (PGN)

PGN หรือ Parameter Group Number เป็นตัวระบุในโปรโตคอล SAE J1939 ที่ทำหน้าที่กำหนดประเภทของข้อมูลหรือข้อความที่ส่งผ่านระบบ CAN BUS โดย PGN ดังรูปที่ 2.7 ทำหน้าที่บ่งบอกถึงชนิดและลักษณะของข้อมูล เช่น ข้อมูลเกี่ยวกับการควบคุมเครื่องยนต์ ระบบเบรก หรือการวินิจฉัยปัญหาในยานยนต์ การใช้ PGN ในการจัดการข้อมูลทำให้สามารถกำหนดลำดับความสำคัญของข้อมูลที่ถูกส่งได้อย่างชัดเจน โดยข้อความที่มีความสำคัญสูงสุด เช่น ข้อมูลด้านความปลอดภัย จะถูกส่งก่อนข้อมูลที่มีความสำคัญน้อยกว่า ซึ่งช่วยเพิ่มประสิทธิภาพและเสถียรภาพของระบบการสื่อสารในยานยนต์ พร้อมทั้งเสริมความปลอดภัยในการใช้งาน



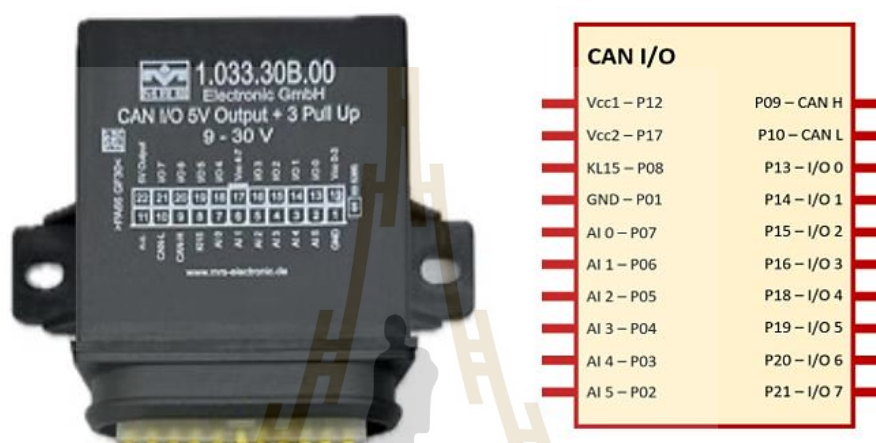
รูปที่ 2.7 Parameter Group Numbers (PGN)

2.6 การประยุกต์ CAN I/O PLC ร่วมกับ MRS Developers Studio

การใช้งาน CAN I/O PLC ของ MRS ร่วมกับ MRS Developers Studio เป็นการผสมผสานเทคโนโลยีฮาร์ดแวร์และซอฟต์แวร์ที่ช่วยในการควบคุมและสื่อสารข้อมูลระหว่างอุปกรณ์ในระบบยานยนต์หรืออุตสาหกรรมที่ใช้ CAN BUS โดยเฉพาะในยานยนต์ไฟฟ้า (EV) CAN I/O PLC มีความสามารถในการจัดการสัญญาณอินพุตและเอาต์พุตที่หลากหลาย เมื่อใช้ร่วมกับ MRS Developers Studio ซึ่งเป็นเครื่องมือสำหรับการเขียนโปรแกรมและพัฒนาระบบสื่อสารแบบ CAN จะช่วยให้การควบคุมและสื่อสารกับอุปกรณ์ต่าง ๆ ผ่านระบบ CAN BUS เป็นไปอย่างมีประสิทธิภาพและแม่นยำสูงสุด

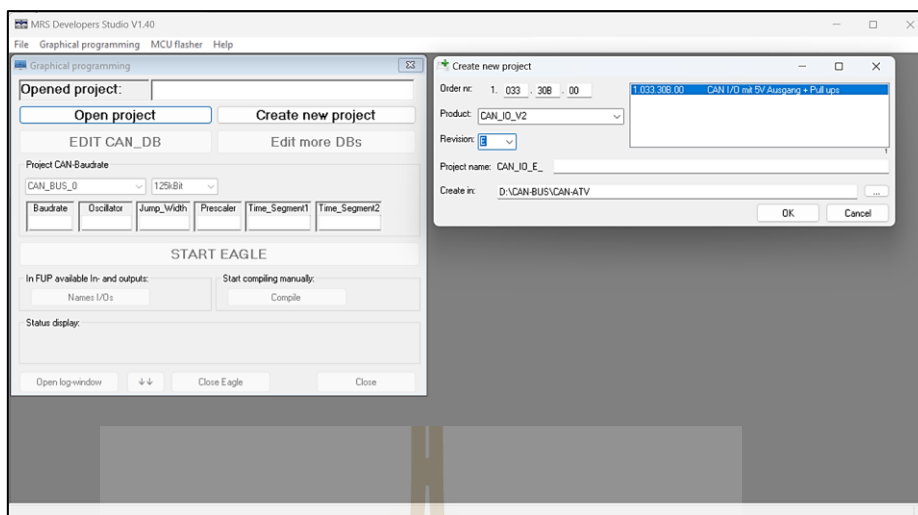
CAN I/O PLC ของ MRS ที่แสดงดังรูปที่ 2.8 เป็นอุปกรณ์ที่ออกแบบมาเพื่อทำงานร่วมกับระบบ CAN BUS โดยทำหน้าที่ควบคุมและประมวลผลการทำงานของอุปกรณ์ต่าง ๆ ในระบบอัตโนมัติแบบเรียลไทม์ เช่น เซ็นเซอร์ มอเตอร์ และอุปกรณ์ I/O ด้วยการออกแบบที่ยืดหยุ่น ทำให้ CAN I/O PLC สามารถตั้งโปรแกรมเพื่อควบคุมกระบวนการที่ซับซ้อนได้อย่างแม่นยำ อุปกรณ์นี้ยัง

รองรับการเชื่อมต่อกับอุปกรณ์ภายนอกผ่านโปรโตคอล CAN BUS ซึ่งช่วยให้สามารถสั่งการและตรวจสอบการทำงานของอุปกรณ์ต่าง ๆ ได้อย่างรวดเร็ว เช่น การควบคุมมอเตอร์ การรับข้อมูลจากเซ็นเซอร์ และการควบคุมระบบอื่น ๆ ที่ต้องการความแม่นยำสูง นอกจากนี้ การตั้งโปรแกรมการทำงานยังสามารถปรับเปลี่ยนได้ตามความต้องการเฉพาะของระบบ โดยการใช้งานร่วมกับ MRS Developers Studio ช่วยให้การควบคุมและการจัดการกระบวนการเป็นไปอย่างมีประสิทธิภาพ



รูปที่ 2.8 CAN I/O PLC รุ่น MRS Developers Studio

MRS Developers Studio เป็นซอฟต์แวร์ที่ออกแบบมาเพื่อสนับสนุนการพัฒนาระบบสื่อสารใน CAN BUS โดยเฉพาะการทำงานร่วมกับ CAN I/O PLC จากรูปที่ 2.9 ซอฟต์แวร์นี้ช่วยให้ผู้ใช้สามารถตั้งโปรแกรมการทำงานของ CAN I/O PLC ได้อย่างละเอียดตามความต้องการของระบบ พร้อมทั้งมีฟังก์ชันการจำลองการทำงานในสภาพแวดล้อมเสมือนจริง ซึ่งช่วยให้สามารถตรวจสอบและทดสอบการทำงานของระบบก่อนนำไปใช้งานจริง ลดความเสี่ยงจากปัญหาที่อาจเกิดขึ้นในระหว่างการใช้งานจริง นอกจากนี้ MRS Developers Studio ยังมีเครื่องมือที่ช่วยในการตรวจสอบและแก้ไขปัญหาที่เกิดขึ้นในระบบ CAN BUS ได้อย่างรวดเร็ว ทำให้สามารถเพิ่มเสถียรภาพของระบบได้อย่างมีประสิทธิภาพ

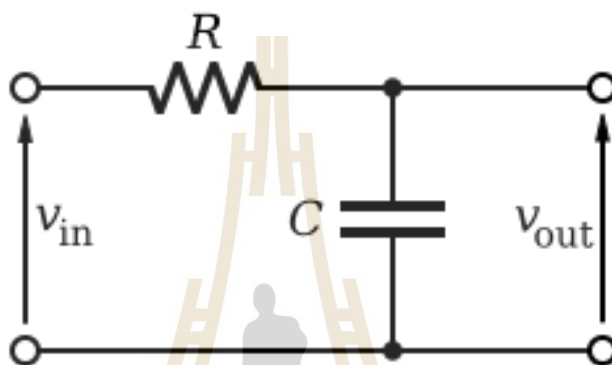


รูปที่ 2.9 ซอฟต์แวร์ MRS Developers Studio

การใช้งาน CAN I/O PLC ร่วมกับ MRS Developers Studio ช่วยให้การออกแบบและการควบคุมระบบ CAN BUS ในยานยนต์หรือระบบอุตสาหกรรมเป็นไปอย่างราบรื่นและมีประสิทธิภาพ ซอฟต์แวร์นี้ช่วยให้วิศวกรและผู้พัฒนาสามารถปรับแต่งโปรแกรมเพื่อควบคุมอุปกรณ์ต่าง ๆ ได้อย่างยืดหยุ่นและสอดคล้องกับความต้องการเฉพาะของระบบ เช่น การควบคุมมอเตอร์ไฟฟ้า การปรับโหมดการทำงานของอุปกรณ์ หรือการตรวจสอบสถานการณ์ทำงานของระบบความปลอดภัยในยานยนต์

ตัวอย่างการใช้งานหนึ่งคือการควบคุมสัญญาณนาฬิกา ดังรูปที่ 2.10 ซึ่ง CAN I/O PLC สามารถส่งสัญญาณนาฬิกา ผ่านการสื่อสารกับ CAN BUS โดยการตั้งโปรแกรมเพื่อควบคุมความเร็วเปลี่ยนโหมดการทำงาน หรือหยุดของสัญญาณ ทุกขั้นตอนการตั้งค่าและควบคุมนี้สามารถดำเนินการได้ผ่านซอฟต์แวร์ MRS Developers Studio ทำให้ระบบทำงานอย่างมีประสิทธิภาพ พร้อมทั้งเสริมความปลอดภัยและความแม่นยำในการควบคุม

นอกจากนี้ ในระบบควบคุมยานยนต์ไฟฟ้า ตัวกรองความถี่ต่ำยังมีบทบาทสำคัญในการกรองสัญญาณเซ็นเซอร์ ลดสัญญาณรบกวนในระบบ CAN Bus และช่วยให้การควบคุมมอเตอร์มีประสิทธิภาพมากขึ้น โดยสรุป ตัวกรองความถี่ต่ำเป็นองค์ประกอบสำคัญที่ช่วยให้ระบบอิเล็กทรอนิกส์และการสื่อสารทำงานได้อย่างมีประสิทธิภาพ การเลือกใช้ตัวกรองที่เหมาะสมขึ้นอยู่กับความต้องการของแต่ละระบบ ซึ่งสามารถช่วยปรับปรุงคุณภาพของสัญญาณและลดการรบกวนที่ไม่พึงประสงค์ได้อย่างมีประสิทธิภาพ

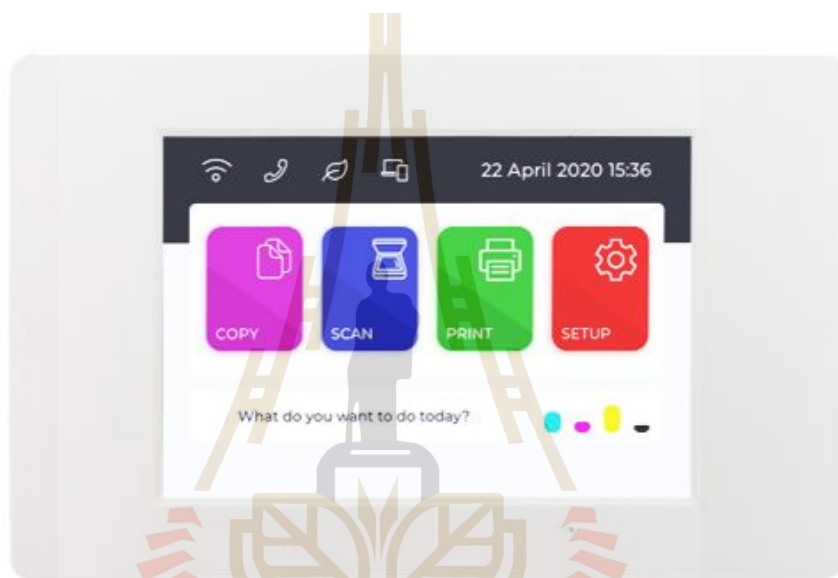


รูปที่ 2.11 ตัวอย่างการต่อวงจรกรองความถี่ต่ำ (Low-pass filter)

2.8 จอแสดงผลสถานการณ์การทำงานของระบบ CANBUS

จอแสดงผล ATD3.5-S3 เป็นบอร์ดพัฒนาที่ต่อยอดจากไมโครคอนโทรลเลอร์ ESP32-S3 ซึ่งเป็นชิปแบบ Dual-core ที่มีประสิทธิภาพในการประมวลผลสูง พร้อมหน่วยความจำ Flash ขนาด 8MB และ PSRAM ขนาด 8MB รองรับการทำงานของระบบที่ต้องการการประมวลผลกราฟิกและเสียงในระดับที่ซับซ้อน โดยบอร์ดรุ่นนี้มาพร้อมกับหน้าจอสัมผัสแบบ Capacitive ขนาด 3.5 นิ้ว ความละเอียด 480x320 พิกเซล ชนิด IPS (In-Plane Switching) ซึ่งให้ภาพที่คมชัด สีสันสมจริง และมุมมองที่กว้าง รองรับการสัมผัสหลายจุดผ่านตัวควบคุมระบบสัมผัส FT6336U ที่เชื่อมต่อผ่าน I2C บอร์ด ATD3.5-S3 มีคุณสมบัติการเชื่อมต่อแบบไร้สาย ได้แก่ Wi-Fi ความถี่ 2.4GHz และ Bluetooth 5 ช่วยให้สามารถสื่อสารกับอุปกรณ์ภายนอกหรือระบบควบคุมได้อย่างยืดหยุ่นและมีประสิทธิภาพ นอกจากนี้ยังมีองค์ประกอบฮาร์ดแวร์เพิ่มเติมที่อำนวยความสะดวกในการใช้งาน เช่น ลำโพงภายในที่สามารถขับเสียงผ่าน Digital-to-Analog Converter (DAC), ช่องเสียบการ์ด MicroSD สำหรับจัดเก็บข้อมูลเพิ่มเติม รวมถึงพอร์ตเชื่อมต่อมาตรฐาน เช่น I2C, SPI และ GPIO ซึ่งเหมาะสำหรับต่อร่วมกับเซ็นเซอร์หรืออุปกรณ์ภายนอกอื่น ๆ ในด้านการพัฒนาโปรแกรม บอร์ดนี้รองรับเครื่องมือและไลบรารีที่หลากหลาย เช่น Arduino IDE ซึ่งเหมาะสำหรับผู้เริ่มต้นและ Maker,

ไลบรารี LVGL (Light and Versatile Graphics Library) สำหรับการสร้างส่วนติดต่อผู้ใช้งาน (UI) แบบกราฟิกที่ลื่นไหล และ SquareLine Studio ซึ่งเป็นเครื่องมือออกแบบ UI แบบลากวางที่สามารถเชื่อมต่อกับ LVGL ได้โดยตรง ด้วยความสามารถข้างต้น บอร์ด ATD3.5-S3 จึงเหมาะสมอย่างยิ่งสำหรับการนำไปใช้งานในระบบที่ต้องการแสดงผลข้อมูลและตอบสนองต่อการสัมผัส เช่น ในวิจัยนี้ใช้สำหรับ แสดงผลสถานะของระบบในยานยนต์ไฟฟ้าประเภท ATV ไม่ว่าจะเป็นข้อมูลเกี่ยวกับแบตเตอรี่ ความเร็ว มอเตอร์ หรือระบบไฟฟ้าภายในตัวรถ โดยบอร์ดสามารถตอบสนองได้อย่างแม่นยำและมีประสิทธิภาพในลักษณะเรียลไทม์ ตัวอย่างจอ ATD3.5-S3 ดังรูปที่ 2.12

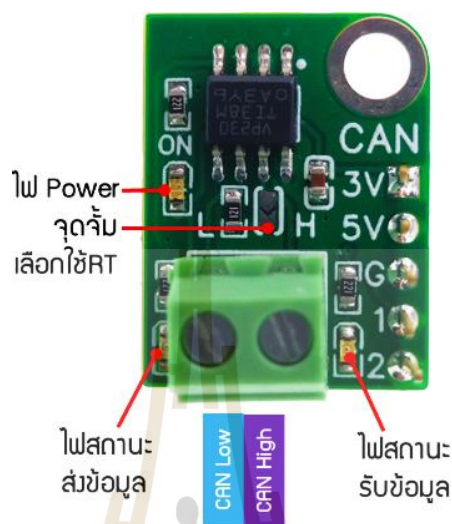


รูปที่ 2.12 จอATD3.5-S3 แสดงสถานะการทำงานของ ATV

CAN BUS Shield

โมดูล ATD3.5-S3 CAN BUS Shield เป็นอุปกรณ์เสริมที่ออกแบบมาสำหรับใช้งานร่วมกับบอร์ด ATD3.5-S3 เพื่อเพิ่มความสามารถในการสื่อสารผ่านเครือข่าย CAN BUS (Controller Area Network) ซึ่งเป็นระบบสื่อสารที่นิยมใช้อย่างแพร่หลายในงานอุตสาหกรรมและยานยนต์ โดยเฉพาะในระบบควบคุมอิเล็กทรอนิกส์ของยานพาหนะ โมดูลนี้รองรับการส่งข้อมูลด้วยความเร็วสูงสุด 1 Mbps และใช้ชิปควบคุมจากบริษัท Texas Instruments (TI) ซึ่งเป็นไปตามมาตรฐาน ISO 11898-2 ช่วยให้การส่งผ่านข้อมูลระหว่างอุปกรณ์ต่าง ๆ ในระบบ CAN BUS เป็นไปอย่างแม่นยำ เสถียร และเชื่อถือได้ แม้ในสภาพแวดล้อมที่มีสัญญาณรบกวนสูง ด้วยคุณสมบัติดังกล่าว โมดูล ATD3.5-S3 CAN BUS Shield จึงเหมาะสำหรับนำมาใช้ในงานวิจัยที่เกี่ยวข้องกับ การสื่อสารระหว่างระบบอิเล็กทรอนิกส์ในยานยนต์ เช่น การอ่านข้อมูลจากกล่องควบคุม (ECU), การแสดงผลสถานะจาก

เซ็นเซอร์ต่าง ๆ ภายในระบบรถยนต์ไฟฟ้า และการเชื่อมต่อกับระบบควบคุมส่วนอื่น ๆ ภายในยานพาหนะ รูปที่ 2.13 แสดงโมดูล ATD3.5-S3 CAN BUS Shield



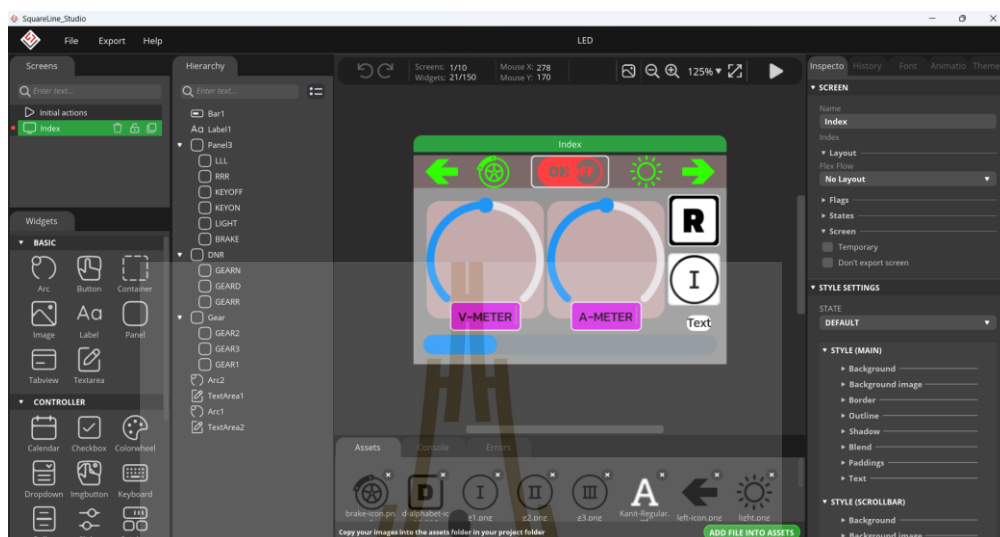
รูปที่ 2.13 CAN BUS Shield

2.9 ซอฟต์แวร์ SquareLine Studio

SquareLine Studio เป็นเครื่องมือสำหรับการออกแบบและพัฒนาอินเทอร์เฟซผู้ใช้ (User Interface: UI) ที่เน้นการใช้งานกับหน้าจอสัมผัสในระบบฝังตัว (Embedded Systems) โดยเฉพาะเมื่อใช้งานร่วมกับไลบรารี LVGL (Light and Versatile Graphics Library) ซึ่งเป็นไลบรารีกราฟิกแบบโอเพนซอร์สที่ได้รับความนิยมสูงในกลุ่มผู้พัฒนาระบบฝังตัว การทำงานร่วมกันระหว่าง SquareLine Studio และ LVGL ช่วยให้กระบวนการสร้าง UI มีความสะดวกและรวดเร็วมากยิ่งขึ้น โดยผู้พัฒนาสามารถออกแบบอินเทอร์เฟซด้วยวิธีการลากและวาง (drag-and-drop) ผ่านเครื่องมือของ SquareLine Studio และสามารถดูผลลัพธ์ในลักษณะใกล้เคียงกับการใช้งานจริงได้ทันที จากนั้นจึงสามารถส่งออกโค้ดเพื่อใช้งานร่วมกับ LVGL ได้อย่างราบรื่น

LVGL เป็นไลบรารีที่ถูกออกแบบมาให้มีความยืดหยุ่นและสามารถประมวลผลกราฟิกที่ซับซ้อนได้อย่างมีประสิทธิภาพ แม้จะทำงานบนฮาร์ดแวร์ที่มีทรัพยากรจำกัด เช่น ไมโครคอนโทรลเลอร์ทั่วไป โดยสามารถรองรับฟังก์ชันต่าง ๆ ได้ครบถ้วน ไม่ว่าจะเป็นปุ่มกด แถบเลื่อน เมนู ป๊อปอัพ หรือการจัดการเลย์เอาต์แบบหลายหน้า ด้วยคุณสมบัติเหล่านี้ SquareLine Studio จึงกลายเป็นเครื่องมือสำคัญในการพัฒนาอินเทอร์เฟซผู้ใช้ที่มีความซับซ้อน สวยงาม และตอบสนองต่อผู้ใช้ได้อย่างมีประสิทธิภาพ โดยเฉพาะเมื่อนำมาใช้ในงานที่เกี่ยวข้องกับระบบแสดงผลผ่านหน้าจอ

สัมผัส เช่น ระบบแสดงผลสถานะของยานยนต์ไฟฟ้า ATV ที่ต้องการ UI ที่ใช้งานง่ายและตอบสนองแบบเรียลไทม์ หน้าต่างการทำงานของ SquareLine Studio แสดงดังภาพที่ 2.14



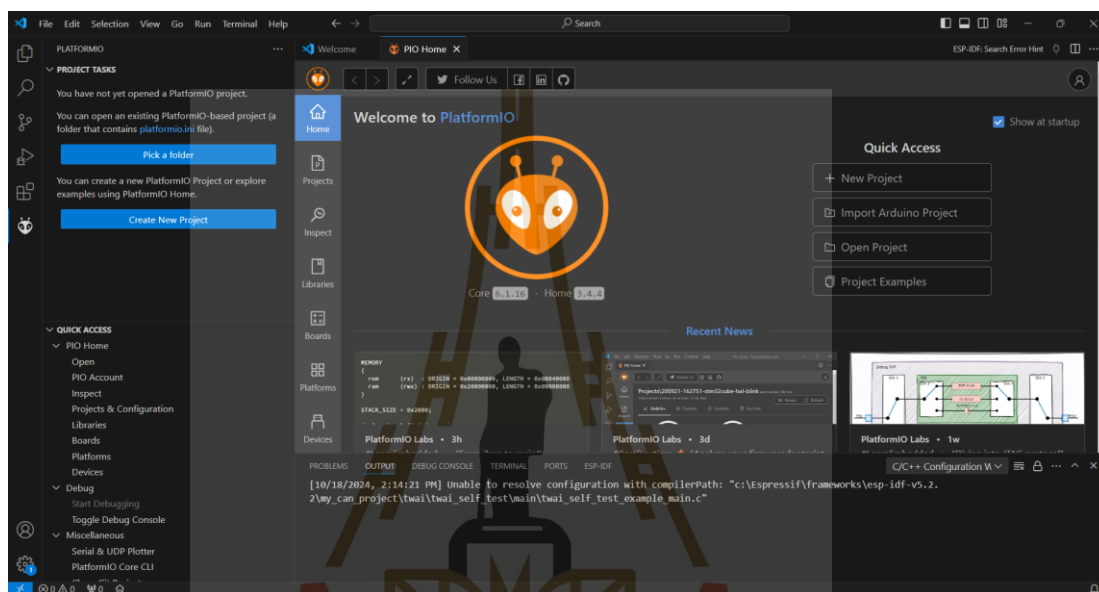
รูปที่ 2.14 ซอฟต์แวร์ SquareLine Studio

2.10 โปรแกรม Visual Studio Code และ Platform IO Extension

Visual Studio Code (VS Code) เป็นโปรแกรมแก้ไขโค้ด (Code Editor) ที่ได้รับความนิยมสูงในหมู่นักพัฒนาซอฟต์แวร์ เนื่องจากมีความยืดหยุ่นในการใช้งาน และสามารถรองรับภาษาการเขียนโปรแกรมได้หลากหลาย ไม่ว่าจะเป็น C/C++, Python, JavaScript, HTML, หรือภาษาอื่น ๆ อีกมากมาย จุดเด่นของ VS Code คือ ความสามารถในการปรับแต่งและขยายฟังก์ชันการทำงานผ่านระบบ ส่วนเสริม (Extensions) ที่ช่วยให้ผู้ใช้งานสามารถปรับแต่งสภาพแวดล้อมให้เหมาะสมกับลักษณะงานเฉพาะทางได้ หนึ่งในส่วนเสริมที่สำคัญและได้รับความนิยมอย่างมากในกลุ่มผู้พัฒนาโปรแกรมสำหรับไมโครคอนโทรลเลอร์และระบบ Internet of Things (IoT) คือ PlatformIO Extension ซึ่งเป็นเครื่องมือที่ช่วยในการพัฒนาโปรแกรมบนไมโครคอนโทรลเลอร์หลายแพลตฟอร์ม เช่น ESP32, Arduino, STM32 และอีกหลากหลายรุ่น

PlatformIO Extension ทำให้ผู้พัฒนาสามารถเขียนโค้ด คอมไพล์ อัปโหลดโปรแกรม และดีบั๊ก (Debug) อุปกรณ์ได้ภายใน VS Code โดยไม่จำเป็นต้องสลับไปใช้โปรแกรมอื่น เพิ่มความสะดวกและลดเวลาในการพัฒนาอย่างมีประสิทธิภาพ นอกจากนี้ PlatformIO ยังมีระบบจัดการไลบรารี (Library Management) ที่ช่วยให้สามารถติดตั้งและอัปเดตไลบรารีที่จำเป็นต่องานที่ทำได้อย่างง่ายดาย รวมถึงมีระบบการจัดการบอร์ดที่ยืดหยุ่น สามารถเลือกบอร์ดและกำหนดค่าเบื้องต้น

ของโปรเจกต์ได้อย่างสะดวก ด้วยคุณสมบัติข้างต้น VS Code ร่วมกับ PlatformIO Extension จึงเป็นเครื่องมือที่เหมาะสมอย่างยิ่งในการพัฒนาโครงงานด้านระบบฝังตัว (Embedded Systems) และระบบ IoT โดยเฉพาะในโครงงานที่เกี่ยวข้องกับไมโครคอนโทรลเลอร์และการสื่อสารกับอุปกรณ์ภายนอก เช่น การแสดงผลผ่านจอสัมผัส หรือการควบคุมอุปกรณ์ในระบบยานยนต์ไฟฟ้า โดยภาพรวมของโปรแกรม แสดงดังรูปที่ 2.15



รูปที่ 2.15 โปรแกรม Visual Studio Code พร้อม Platform IO

2.11 ปรัชญาวรรณกรรมและงานวิจัยที่เกี่ยวข้อง

(Nambunlue, 2024) งานวิจัยนี้นำเสนอการประยุกต์ใช้เทคโนโลยี Internet of Things (IoT) ร่วมกับ Power Meter และ V-BOX เพื่อตรวจสอบและจัดการการใช้พลังงานไฟฟ้าแบบเรียลไทม์และย้อนหลัง โดยมีวัตถุประสงค์เพื่อเพิ่มประสิทธิภาพการจัดการพลังงานในภาคอุตสาหกรรม ระบบ IoT ที่พัฒนาขึ้นนี้ประกอบด้วย Power Meter สำหรับวัดค่าพารามิเตอร์ทางไฟฟ้าต่าง ๆ เช่น แรงดัน กระแส กำลังไฟฟ้า และความถี่ V-BOX ซึ่งเป็นอุปกรณ์ IoT Gateway ที่ทำหน้าที่รับส่งข้อมูลจาก Power Meter ไปยังระบบคลาวด์ และระบบคลาวด์ที่ใช้ซอฟต์แวร์ V-NET สำหรับแสดงผลข้อมูล วิเคราะห์ และแจ้งเตือน ข้อมูลจาก Power Meter จะถูกส่งไปยัง V-BOX ผ่านโปรโตคอล MODBUS และ V-BOX จะส่งข้อมูลต่อไปยังระบบคลาวด์ผ่านเครือข่ายอินเทอร์เน็ต ผู้ใช้สามารถเข้าถึงข้อมูลและตรวจสอบสถานะการทำงานของระบบได้ผ่านเว็บเบราว์เซอร์หรือแอปพลิเคชันบนมือถือ ระบบนี้มีความสามารถในการแสดงผลข้อมูลแบบเรียลไทม์

แสดงข้อมูลย้อนหลังในรูปแบบกราฟ แจ้งเตือนเมื่อค่าพารามิเตอร์ทางไฟฟ้าผิดปกติ และส่งข้อความแจ้งเตือนผ่านแอปพลิเคชัน LINE นอกจากนี้ ผลการทดลองแสดงให้เห็นว่าระบบ IoT ที่พัฒนาขึ้นสามารถทำงานได้อย่างมีประสิทธิภาพในการตรวจสอบและจัดการการใช้พลังงานไฟฟ้าแบบเรียลไทม์ และย้อนหลัง ระบบสามารถแสดงผลข้อมูลได้อย่างถูกต้องแม่นยำ แจ้งเตือนเมื่อเกิดความผิดปกติ และควบคุมอุปกรณ์ไฟฟ้าจากระยะไกลได้ตามต้องการ

(Rattanachan, 2021) งานวิจัยนี้มีวัตถุประสงค์ในการพัฒนาระบบควบคุมประตูไฟฟ้าสำหรับรถจักรยานยนต์โดยเปลี่ยนจากระบบนิวมेटริกส์ที่ใช้อยู่เดิมมาเป็นระบบไฟฟ้า ซึ่งควบคุมผ่าน MRS Developers Studio และใช้การสื่อสารแบบ CAN-Bus ตามมาตรฐาน J1939 ที่เป็นมาตรฐานสากลในอุตสาหกรรมยานยนต์ทั่วโลก การพัฒนานี้เกิดขึ้นเพื่อรองรับข้อกำหนดด้านความปลอดภัยของกรมการขนส่งทางบกเกี่ยวกับประตูและทางออกฉุกเฉินสำหรับยานพาหนะขนส่งผู้โดยสารที่ประกาศในปี 2014 ระบบที่พัฒนานี้ประกอบด้วยหน่วยควบคุมอิเล็กทรอนิกส์ (ECU) หลัก 3 หน่วย ได้แก่ ECU1 สำหรับควบคุมไฟและระบบแจ้งเตือนของประตู ECU2 สำหรับควบคุมมอเตอร์เพื่อการเปิด-ปิดประตู และ ECU3 เป็นคอมพิวเตอร์ที่ใช้ส่งและรับคำสั่งควบคุมระบบ โดยในกระบวนการเปิดและปิดประตู ระบบจะทำการตรวจสอบสถานะของประตูว่ามีการเปิดหรือปิดสมบูรณ์หรือไม่ และจะเปิดเสียงแจ้งเตือนและไฟกระพริบหากพบสิ่งกีดขวาง ทั้งนี้ หากพบว่าไม่มีสิ่งกีดขวาง ระบบจะสั่งให้มอเตอร์หมุนตามเข็มนาฬิกาเพื่อปิดประตู การออกแบบวงจรไฟฟ้าและการเขียนโปรแกรมควบคุมมีการแบ่งงานออกเป็นสองส่วน ได้แก่ การออกแบบวงจรที่ใช้โมดูล CAN I/O สำหรับควบคุมการทำงานของไฟและระบบแจ้งเตือน และการเขียนโปรแกรมใน MRS Developers Studio เพื่อควบคุมการสั่งการมอเตอร์ให้หมุนตามทิศทางที่ต้องการ การควบคุมดังกล่าวสามารถตรวจสอบสถานะการเปิดหรือปิดของประตูได้อย่างแม่นยำ ทำให้สามารถควบคุมการเคลื่อนที่ของรถจักรยานยนต์ไม่ให้เริ่มเดินหน้าหรือถอยหลังได้หากประตูยังเปิดอยู่ ผลการทดลองพบว่า การเปิดและปิดประตูสามารถดำเนินการได้ภายใน 3-6 วินาที โดยในขั้นตอนการเปิดประตู กระแสไฟฟ้าจะเพิ่มขึ้นสูงสุดที่ 3.3 แอมป์ในช่วงแรกและค่อยๆ ลดลงตามลำดับ ในขณะที่การปิดประตูเมื่อมีวัตถุขวาง กระแสไฟจะเพิ่มขึ้นก่อนที่จะลดลงเมื่อประตูเริ่มเคลื่อนที่ย้อนกลับเพื่อเปิดประตูอีกครั้ง งานวิจัยนี้จึงได้แสดงให้เห็นถึงประสิทธิภาพและความปลอดภัยของระบบควบคุมประตูไฟฟ้าที่พัฒนาขึ้น สามารถนำไปใช้เป็นต้นแบบในการพัฒนาเพิ่มเติมในอนาคตได้อย่างมีประสิทธิภาพ

(Adly, 2023) งานวิจัยนี้นำเสนอแนวทางป้องกันการโจมตีในระบบเครือข่ายควบคุม (CAN) ของยานพาหนะไฟฟ้าอัตโนมัติ โดยมีเป้าหมายเพื่อลดความเสี่ยงจากโค้ดที่เป็นอันตรายซึ่งถูกฉีดเข้าสู่หน่วยควบคุมอิเล็กทรอนิกส์ (ECU) และอาจส่งผลกระทบต่อความปลอดภัยในการขับขี่ ระบบ secure boot ถูกนำมาใช้เพื่อตรวจสอบความถูกต้องของซอฟต์แวร์ที่ทำงานบน ECU โดยอิงตามอัลกอริทึมการเข้ารหัสข้อมูล เพื่อให้แน่ใจว่าเฉพาะซอฟต์แวร์ที่ผ่านการตรวจสอบและไม่มีการ

เปลี่ยนแปลงใด ๆ เท่านั้นที่สามารถทำงานได้ งานวิจัยยังได้ทดสอบอัลกอริทึมการรักษาความปลอดภัย 5 ชนิดที่แตกต่างกัน ได้แก่ CMAC และ ECDSA ซึ่งผลการทดสอบพบว่า CMAC ให้ประสิทธิภาพสูงที่สุดในการทำงาน ส่วน ECDSA ให้ระดับความปลอดภัยที่สูงกว่า และในขณะเดียวกันยังมีการพัฒนาอัลกอริทึม ECDSA ใหม่ที่ใช้ CMAC ซึ่งให้ประสิทธิภาพเพิ่มขึ้นถึง 19% เมื่อเทียบกับ ECDSA แบบดั้งเดิม

(Pham, 2022) งานวิจัยนี้นำเสนอการออกแบบระบบจัดการแบตเตอรี่ (Battery Management System - BMS) สำหรับยานพาหนะไฟฟ้าไร้คนขับ (Unmanned Electric Vehicles - UVs) โดยใช้เทคโนโลยี CAN BUS และ Internet of Things (IoT) ระบบ BMS นี้มีการออกแบบทั้งในส่วนฮาร์ดแวร์และซอฟต์แวร์เพื่อตรวจสอบสถานะการชาร์จ (State of Charge - SoC) และอุณหภูมิการทำงาน รวมถึงป้องกันการเสียดสีของเซลล์แบตเตอรี่ โครงสร้างของระบบนี้ประกอบด้วยโมดูลหลัก (Master module) และโมดูลย่อย (Slave modules) ที่ทำหน้าที่ตรวจสอบและป้องกันชุดแบตเตอรี่หลายเซลล์จากสถานะอันตราย โดยข้อมูลจากโมดูลย่อยจะถูกส่งไปยังโมดูลหลักผ่าน CAN BUS ซึ่งสามารถส่งต่อข้อมูลนี้ไปยังสถานีควบคุมเพื่อการวิเคราะห์แบบเรียลไทม์ อีกทั้งยังใช้ IoT เพื่อให้สามารถเฝ้าระวังและควบคุมสถานะแบตเตอรี่จากระยะไกลได้อย่างมีประสิทธิภาพ

(Alzahrani, 2023) งานวิจัยนี้มุ่งพัฒนาและออกแบบโปรโตคอลการสื่อสาร SAE J1939 และ Modbus สำหรับยานพาหนะไฟฟ้า (EV) โดยเน้นการใช้โปรโตคอลการสื่อสารแบบ CAN Bus และการควบคุมการชาร์จผ่านระบบ Modbus เพื่อเพิ่มประสิทธิภาพการถ่ายโอนพลังงานระหว่าง EV กับสถานีชาร์จและปรับปรุงความปลอดภัยในการทำงาน ระบบที่พัฒนานี้ใช้ไมโครคอนโทรลเลอร์ Arduino ในการควบคุมการสื่อสาร โดย CAN Bus ใช้การส่งข้อมูลระหว่างระบบจัดการแบตเตอรี่ (BMS) กับคอนโทรลเลอร์ของสถานีชาร์จผ่านอินเทอร์เฟซ PCAN View ทำให้สามารถตรวจสอบและบันทึกการสื่อสารได้อย่างมีประสิทธิภาพ ขณะที่โปรโตคอล Modbus ถูกนำมาใช้ในการจัดการกับโมดูลพลังงานของสถานีชาร์จเพื่อควบคุมการแปลงไฟฟ้า AC/DC และ DC/DC ตามความต้องการพลังงานของ EV ทั้งนี้ การใช้โปรโตคอลทั้งสองแบบช่วยให้สามารถเพิ่มระยะทางระหว่างยานพาหนะไฟฟ้ากับสถานีชาร์จ และการทดสอบยังได้แสดงให้เห็นถึงความสามารถของระบบในการส่งและรับข้อมูลที่สอดคล้องกับมาตรฐาน SAE J1939 อีกด้วย

(Osman, 2021) งานวิจัยนี้มุ่งเน้นการพัฒนาความน่าเชื่อถือในการสื่อสารระหว่างยานพาหนะกับทุกสิ่ง (V2X) โดยใช้เทคนิค Deep Learning เพื่อให้ได้ข้อมูลจราจรที่มีประสิทธิภาพและเพิ่มความปลอดภัยทางถนน โมเดลที่พัฒนาขึ้นใช้ระบบการควบคุมพลังงานแทรกสอดที่เหมาะสมซึ่งช่วยปรับปรุงคุณภาพบริการ (QoS) ของการสื่อสาร V2X ผ่านการประเมินอัตราข้อมูลที่ทำได้ (Rb), อัตราการส่งข้อมูลที่ประสบความสำเร็จ (PDR), อัตราการสูญเสียข้อมูล (PLR), และความล่าช้าเฉลี่ย

แบบ end-to-end (E2E) โดยใช้แบบจำลองการเรียนรู้เชิงลึก 1D-CNN เพื่อตรวจจับและควบคุมพลังงานการรบกวนให้เหมาะสมกับสถานะต่าง ๆ ของเครือข่าย



บทที่ 3

วิธีดำเนินการวิจัย

3.1 บทนำ

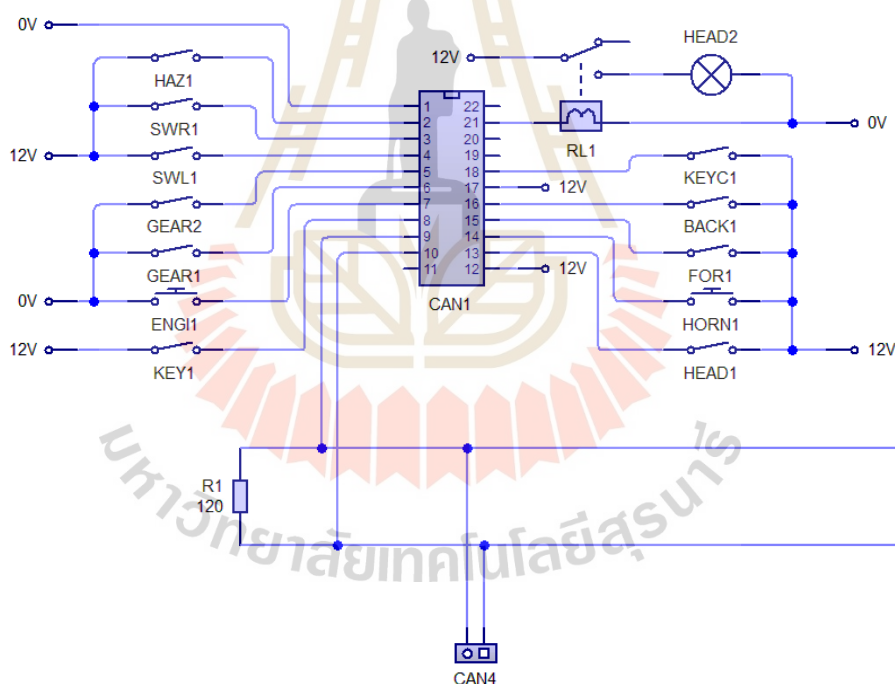
ในการศึกษานี้ กระบวนการและขั้นตอนการดำเนินการวิจัยได้รับการวางแผนและดำเนินการอย่างเป็นระบบ โดยครอบคลุมถึงการออกแบบการทดลอง การเก็บรวบรวมข้อมูล การวิเคราะห์ข้อมูล ตลอดจนการประยุกต์ใช้เครื่องมือและอุปกรณ์ต่าง ๆ เพื่อให้ผลลัพธ์ที่ได้มีความถูกต้องและน่าเชื่อถือ วิธีการดำเนินการวิจัยถูกกำหนดขึ้นอย่างมีแบบแผน เพื่อให้สามารถนำผลลัพธ์ไปใช้ในการพัฒนาและแก้ไขปัญหาตามวัตถุประสงค์ของการศึกษา งานวิจัยนี้ใช้เทคโนโลยี CAN BUS ร่วมกับอุปกรณ์ CAN I/O PLC และซอฟต์แวร์ MRS Developers Studio ในการสื่อสารข้อมูลภายในระบบยานยนต์ โดยอ้างอิงมาตรฐาน SAE J1939 ซึ่งใช้ควบคุมระบบสื่อสารและกำหนดที่อยู่ของข้อมูล นอกจากนี้ยังมีการตรวจสอบและทดสอบการทำงานของระบบเพื่อให้มั่นใจว่าข้อมูลที่ส่งผ่านระบบมีความถูกต้อง และสามารถนำไปใช้งานได้จริง เนื่องจาก กล้อง CAN I/O PLC ไม่สามารถส่งสัญญาณอนาล็อก ได้โดยตรง จึงใช้หลักการ Low-pass filter เพื่อแปลงสัญญาณ PWM ให้เป็นสัญญาณอนาล็อก เพื่อให้สามารถสื่อสารกับกล่องควบคุมมอเตอร์ได้ นอกจากนี้ยังมีการวัดกำลังไฟฟ้าผ่าน Power meter ร่วมกับ V-Box และส่งข้อมูลไปยัง V-Net เพื่อบันทึกข้อมูลการใช้พลังงานแบบเรียลไทม์ เพื่อนำไปวิเคราะห์ผลการทดลอง สำหรับการแสดงผลข้อมูล จอสัมผัส ATD3.5-S3 ขนาด 3.5 นิ้ว ซึ่งเชื่อมต่อกับ บอร์ด ESP32-S3 ถูกนำมาใช้เพื่อแสดงผลข้อมูลแบบเรียลไทม์จากระบบ CAN BUS ข้อมูลที่แสดงรวมถึงสถานะการทำงานของระบบ เช่น สัญญาณจากปั๊มสตาร์ท สัญญาณไฟเลี้ยว (ซ้าย-ขวา-ฉุกเฉิน) สัญญาณคันเร่ง และระบบเบรกของยานพาหนะ นอกจากนี้ จอ ATD3.5-S3 ยังสามารถแสดงการสื่อสารระหว่างโมดูล CAN ทั้งสามโมดูล ที่ติดตั้งในตัวรถ ซึ่งช่วยให้การตรวจสอบและทดสอบการทำงานของระบบ CAN BUS มีความแม่นยำและละเอียดรอบคอบมากขึ้น เพื่อให้มั่นใจว่าข้อมูลที่ได้รับและผลลัพธ์ของการวิจัยมีความถูกต้องและสามารถนำไปใช้ได้อย่างมีประสิทธิภาพ

3.2 การออกแบบระบบสื่อสารในยานยนต์ CAN BUS (Controller Area Network)

งานวิจัยนี้ได้ใช้มาตรฐาน SAE J1939 ในการออกแบบระบบสื่อสารภายในยานยนต์ โดยเลือกใช้ High-Speed CAN เป็นรูปแบบในการเชื่อมต่อระหว่างโมดูลต่าง ๆ ซึ่งระบบนี้สามารถรองรับ

การเชื่อมต่อโมดูล CAN ได้สูงสุดถึง 30 โมดูล และจำกัดความยาวของวงจรสื่อสารไม่เกิน 40 เมตร เพื่อให้เหมาะสมกับการใช้งาน งานวิจัยนี้ได้นำโมดูล CAN I/O PLC จำนวน 3 โมดูลมาใช้ในการจำลองระบบสื่อสาร โดยแต่ละโมดูลมีหน้าที่แตกต่างกันไปในการจัดการข้อมูลและสัญญาณต่าง ๆ ภายในระบบ โดยมีการกำหนดหน้าที่ของแต่ละโมดูลแตกต่างกันออกไป ดังนี้

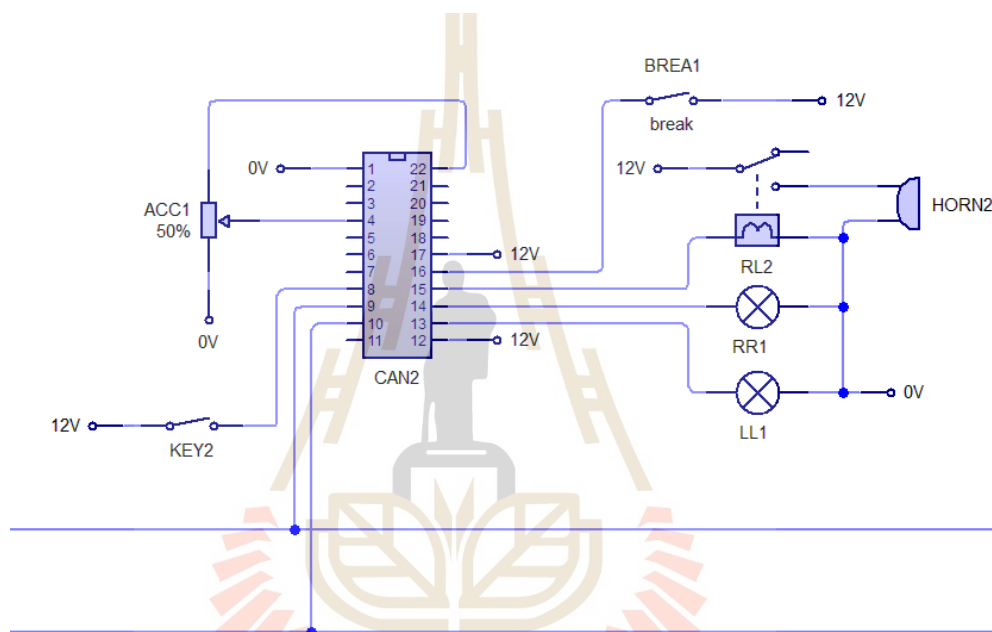
โมดูลที่ 1 จะถูกติดตั้งภายในตัวรถและมีหน้าที่สำคัญในการรับสัญญาณจากสวิตช์ควบคุมต่าง ๆ ที่ติดตั้งในระบบของรถ ATV โมดูลนี้จะทำการรับข้อมูลจากสวิตช์ควบคุมที่ใช้ในการปรับการทำงาน of ระบบต่าง ๆ ภายในรถ เช่น การเปิด-ปิดไฟหน้า สัญญาณไฟเลี้ยว สัญญาณเดินหน้าถอยหลัง โหมมการทำงาน และการสั่งงานฟังก์ชันอื่น ๆ ที่เกี่ยวข้อง จากนั้นโมดูลจะส่งข้อมูลเหล่านี้ไปยังโมดูลอื่น ๆ ภายในระบบผ่านช่องทางสื่อสาร CAN โดยรูปที่ 3.1 แสดงวงจรการเชื่อมต่อของโมดูลนี้ ซึ่งจะสามารถแสดงให้เห็นถึงการเชื่อมต่อระหว่างสวิตช์ควบคุมต่าง ๆ และโมดูล CAN I/O PLC เพื่อให้ระบบทำงานได้อย่างถูกต้อง



รูปที่ 3.1 แผนผังการต่อสายไฟ ของโมดูล CAN ที่ 1

โมดูลที่ 2 จะถูกติดตั้งที่ด้านหน้าของรถ ATV และมีบทบาทสำคัญในการรับสัญญาณจากเซนเซอร์คันเร่งและเซนเซอร์เบรก โดยจะทำการตรวจจับสถานะต่าง ๆ ของระบบ เช่น การกดคันเร่งหรือการเบรก เพื่อส่งข้อมูลไปยังโมดูลอื่น ๆ ในระบบเพื่อให้สามารถประมวลผลและดำเนินการต่อไป

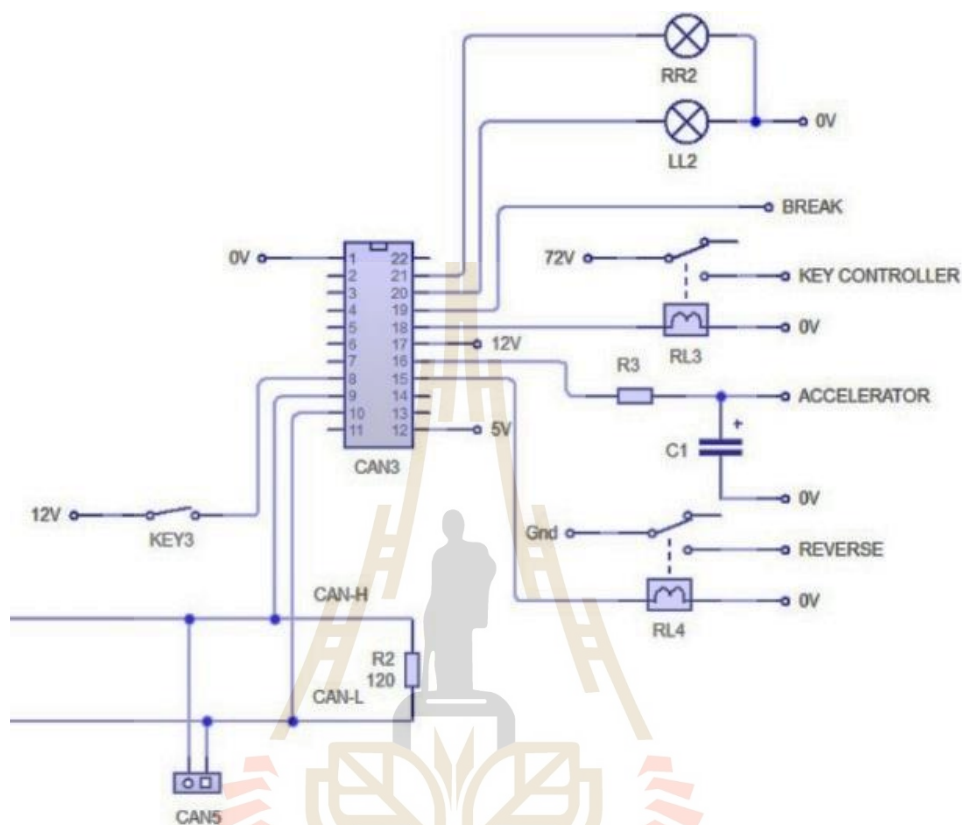
ได้อย่างถูกต้อง นอกจากนี้ โมดูลที่ 2 ยังมีหน้าที่ในการส่งออกสัญญาณไฟเลี้ยวและแตรที่ได้รับค่าจากโมดูลที่ 1 เพื่อให้การแสดงผลของไฟเลี้ยวและการทำงานของแตรตรงตามการควบคุมของผู้ขับขี่ การเชื่อมต่อวงจรของโมดูลนี้จะแสดงในรูปที่ 3.2 ซึ่งจะช่วยให้เห็นการเชื่อมต่อระหว่างเซนเซอร์ต่าง ๆ ที่รับสัญญาณจากคันเร่งและเบรก รวมถึงการรับส่งสัญญาณไฟเลี้ยวและแตรจากโมดูลที่ 1 ผ่านระบบ CAN อย่างไรก็ตาม ในกรณีของแตรนั้นมีการใช้กระแสไฟฟ้าสูง ซึ่งทำให้โมดูล CAN ไม่สามารถจ่ายไฟฟ้าให้กับแตรได้โดยตรง ดังนั้นจึงจำเป็นต้องใช้เลยในการช่วยส่งงานแตร เพื่อป้องกันไม่ให้โมดูล CAN ได้รับความเสียหายจากการจ่ายไฟที่มากเกินไป



รูปที่ 3.2 แผนผังการต่อสายไฟ ของโมดูล CAN ที่ 2

โมดูลที่ 3 จะถูกติดตั้งที่ด้านท้ายของรถ ATV และมีหน้าที่สำคัญในการส่งสัญญาณที่ได้รับจากทั้งโมดูลที่ 1 และโมดูลที่ 2 เพื่อนำไปควบคุมกล่องควบคุมมอเตอร์และอุปกรณ์ต่าง ๆ ภายในระบบของรถ เช่น การควบคุมคันเร่ง ระบบเบรก สวิตช์กุญแจ เกียร์เดินหน้า-ถอยหลัง โหมดการทำงาน และไฟเลี้ยวด้านหลัง โมดูลนี้จะทำหน้าที่ประมวลผลข้อมูลที่ได้รับจากทั้งสองโมดูล ก่อนที่จะส่งสัญญาณไปยังอุปกรณ์ที่เกี่ยวข้อง เพื่อให้สามารถทำงานได้ตามที่ผู้ขับขี่ต้องการ ในส่วนของสัญญาณคันเร่งนั้น เนื่องจากโมดูล CAN ไม่สามารถส่งคำสั่งเป็นสัญญาณอนาล็อกได้ แต่สามารถส่งเป็นสัญญาณ PWM (Pulse Width Modulation) ได้ จึงต้องมีการแปลงรูปสัญญาณจาก PWM เป็นสัญญาณอนาล็อก โดยการใช้วงจร Low Pass Filter เพื่อทำการแปลงสัญญาณดังกล่าวให้เหมาะสมกับการควบคุมอุปกรณ์ที่ต้องการสัญญาณอนาล็อก การเชื่อมต่อวงจรของโมดูลนี้จะแสดงในรูปที่ 3.3

ซึ่งจะช่วยให้เห็นการเชื่อมต่อระหว่างโมดูลที่ 3 กับอุปกรณ์ต่าง ๆ ที่อยู่ในระบบของรถ เพื่อให้ระบบสามารถทำงานได้สอดคล้องกับการควบคุมของผู้ขับขี่

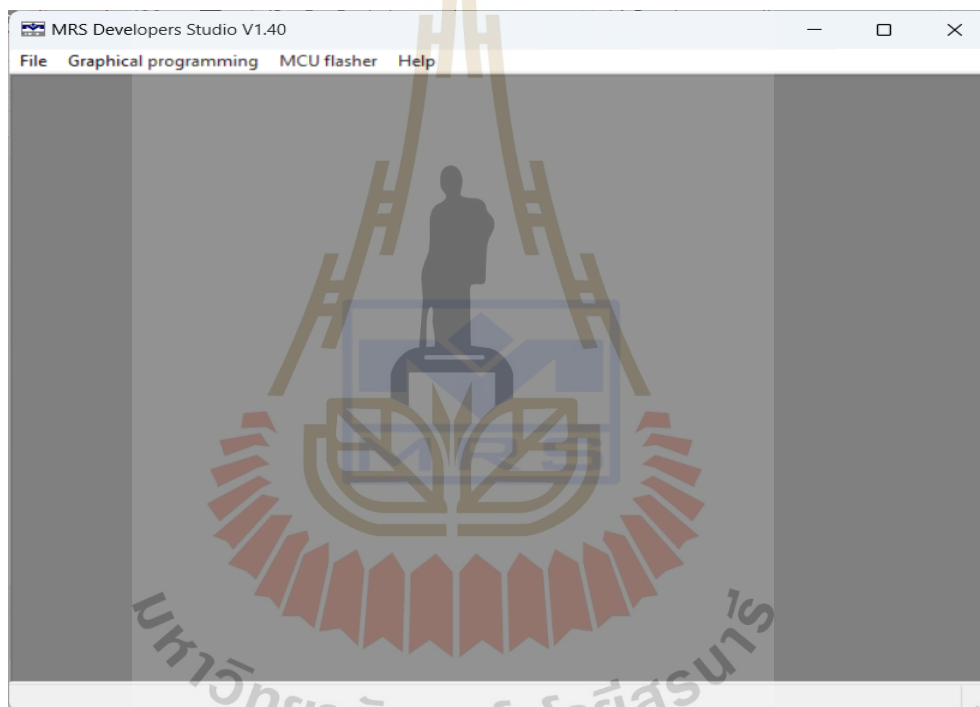


รูปที่ 3.3 แผนผังการต่อสายไฟ ของโมดูล CAN ที่ 3

การออกแบบระบบสื่อสารนี้ช่วยให้โมดูลทั้งสามสามารถทำงานร่วมกันได้อย่างมีประสิทธิภาพภายใต้มาตรฐาน SAE J1939 ซึ่งเป็นมาตรฐานที่รองรับการสื่อสารในระบบยานยนต์ โดยการใช้เทคโนโลยี High-Speed CAN ช่วยให้ระบบสามารถส่งข้อมูลระหว่างโมดูลต่าง ๆ ได้อย่างรวดเร็วและมีเสถียรภาพสูง การกำหนดข้อจำกัดทางด้านความยาววงจรและจำนวนโมดูลที่เชื่อมต่อได้ ทำให้ระบบสามารถทำงานร่วมกันได้อย่างเหมาะสมและไม่เกิดปัญหาความล่าช้าหรือการสูญเสียข้อมูล ระบบนี้ช่วยให้การทำงานของอุปกรณ์ต่าง ๆ ในรถ ATV เป็นไปอย่างราบรื่นและมีความแม่นยำ เนื่องจากโมดูลทั้งสามมีการประมวลผลข้อมูลและสั่งงานได้อย่างมีประสิทธิภาพ รวมถึงการเชื่อมต่อที่เสถียรและรองรับการทำงานที่หลากหลาย เช่น การควบคุมคันเร่ง เบรก สวิตช์กุญแจ เกียร์ ไฟเลี้ยว และอื่น ๆ ได้อย่างถูกต้องและทันเวลา ทำให้การทำงานของระบบสื่อสารภายในรถ ATV มีความเสถียรและเหมาะสมต่อการใช้งานในสภาพแวดล้อมต่าง ๆ

3.3 การออกแบบการทำงานของยานยนต์

การออกแบบวงจรการทำงานของแต่ละโมดูล CAN I/O PLC ได้รับการกำหนดให้มีการทำงานที่แตกต่างกันไปตามหน้าที่เฉพาะของแต่ละโมดูล ซึ่งจำเป็นต้องมีการเขียนโปรแกรมควบคุมเพื่อทำให้การทำงานของโมดูลเหล่านี้สอดคล้องกันและทำงานได้อย่างมีประสิทธิภาพ ผ่านซอฟต์แวร์ MRS Developers Studio เครื่องมือที่ใช้ในการตั้งค่าและเขียนโปรแกรมสำหรับโมดูล CAN I/O PLC ยี่ห้อ MRS โดยซอฟต์แวร์ดังกล่าวช่วยให้สามารถตั้งค่าการทำงานของแต่ละโมดูลได้อย่างละเอียดและตรงตามความต้องการของการใช้งานในแต่ละกรณี โครงสร้างของซอฟต์แวร์ที่ใช้งานได้แสดงอยู่ในรูปที่ 3.4



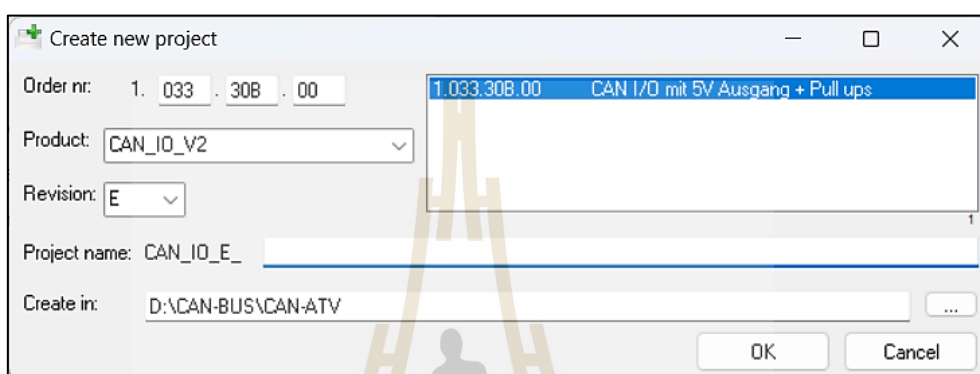
รูปที่ 3.4 หน้าต่างโปรแกรม MRS Developers Studio

3.3.1 ขั้นตอนการสร้างโปรเจกต์ใหม่ในการออกแบบการควบคุมระบบการทำงาน ATV

การเริ่มต้นสร้างโปรเจกต์ใหม่สำหรับ CAN I/O PLC ยี่ห้อ MRS จำเป็นต้องเริ่มจากการกำหนดรุ่นของโมดูลที่ใช้ให้ถูกต้อง โดยในกรณีนี้จะต้องระบุรุ่นเป็น Order nr: 1.033.30B.00 ซึ่งเป็นรุ่นที่กำหนดให้ใช้ Product: CAN_IO_V2 และ Revision: E ตามที่แสดงในรูปที่ 3.5 การกำหนดค่ารุ่นเหล่านี้มีความสำคัญอย่างยิ่ง เนื่องจากแต่ละรุ่นอาจมีคุณสมบัติทางเทคนิคหรือข้อกำหนดที่แตกต่างกัน การตั้งค่าเริ่มต้นให้ตรงกับฮาร์ดแวร์ที่ใช้งานจริงจะช่วยลดความผิดพลาดที่

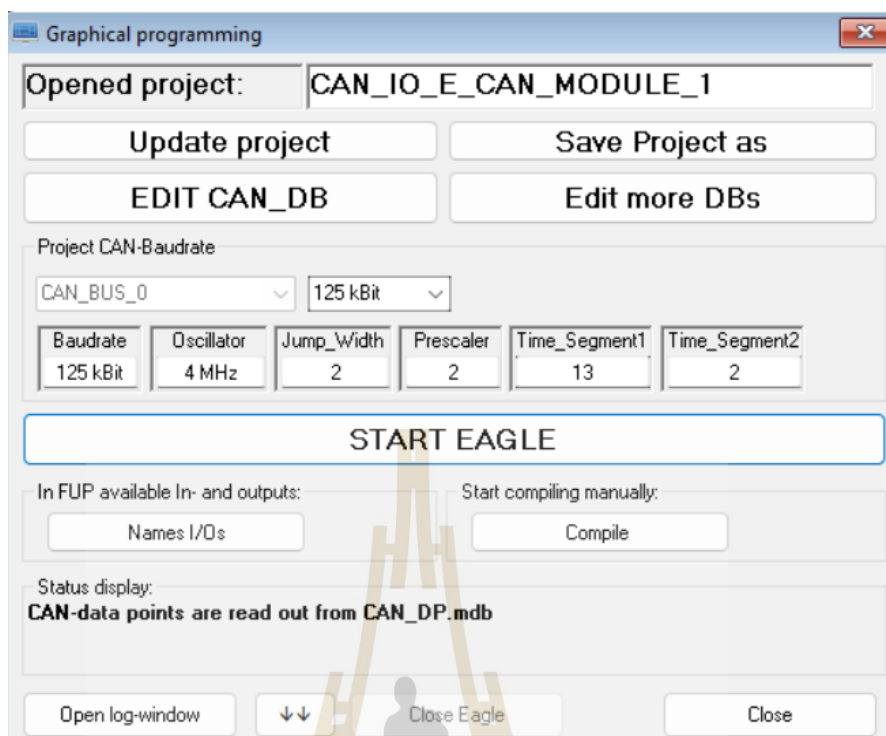
อาจเกิดขึ้นในการเขียนโปรแกรม และช่วยให้การเชื่อมต่อกับระบบ CAN BUS เป็นไปอย่างถูกต้อง และมีเสถียรภาพ

นอกจากนี้ การกำหนดค่าที่ตรงตามฮาร์ดแวร์ยังช่วยให้ซอฟต์แวร์ MRS Developers Studio สามารถประมวลผลคำสั่งและควบคุมฟังก์ชันต่าง ๆ ของโมดูลได้อย่างแม่นยำ ซึ่งส่งผลให้การทำงานของระบบเป็นไปตามแผนที่วางไว้



รูปที่ 3.5 หน้าต่างการตั้งค่าโมดูล PLC CAN I/O

เมื่อสร้างโปรเจกต์ใหม่ในซอฟต์แวร์ MRS Developers Studio หน้าต่าง Graphical Programming จะปรากฏขึ้น แสดงส่วนต่าง ๆ ของโปรเจกต์ที่กำลังเปิดใช้งาน ดังรูป 3.6 ผู้ใช้งานสามารถออกแบบและกำหนดโปรแกรมผ่านบล็อกคำสั่งต่าง ๆ ได้อย่างสะดวก



รูปที่ 3.6 หน้าต่าง Graphical Programming

การทำงานสามารถตั้งค่าโปรแกรมได้ดังนี้

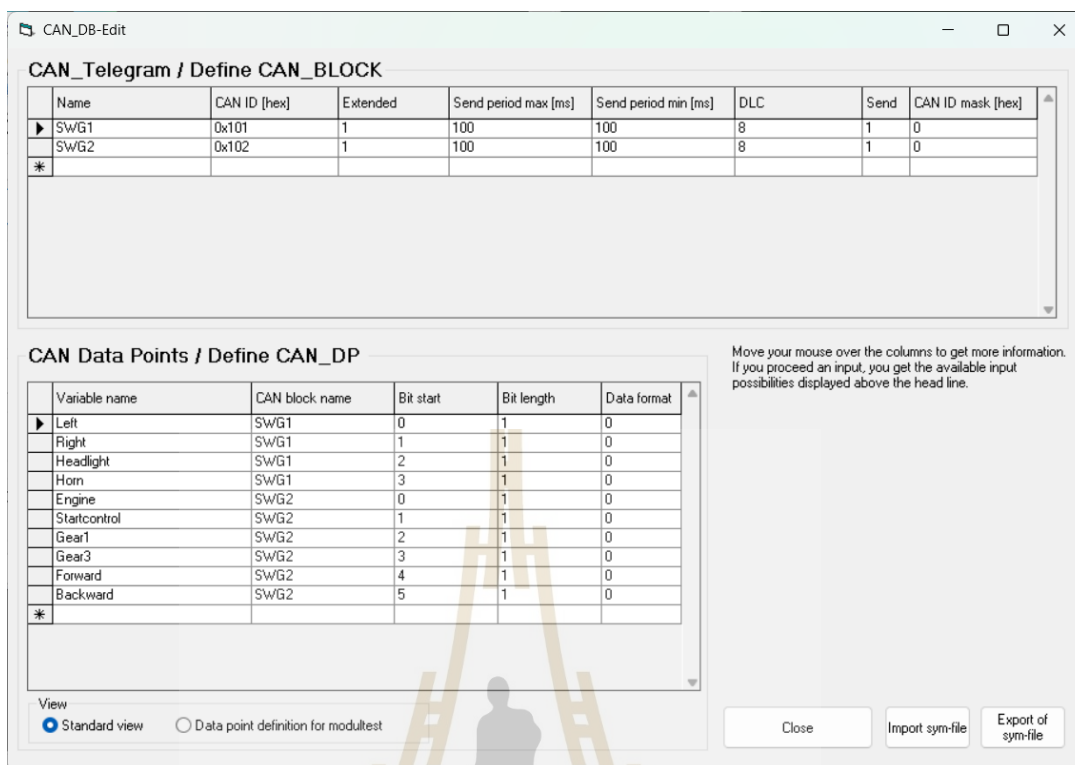
- 1) Opened Project: แสดงชื่อของโปรเจกต์ที่กำลังเปิดใช้งานอยู่ เช่น "CAN_IO_E_CAN_MODULE_1"
- 2) Update Project / Save Project as: ใช้เพื่ออัปเดตหรือบันทึกโปรเจกต์ปัจจุบันในชื่อใหม่
- 3) Edit CAN_DB / Edit more DBs: ใช้สำหรับแก้ไขข้อมูลฐานข้อมูล CAN หรือฐานข้อมูลอื่นที่เกี่ยวข้อง
- 4) Project CAN-Baud rate: กำหนดอัตราความเร็วในการสื่อสารของ CAN BUS (Baud rate) ซึ่งในตัวอย่างคือ 125 Kbit
- 5) Baud rate / Oscillator / Jump Width / Pre Scaler / Time Segment1 / Time Segment2: เป็นพารามิเตอร์ที่ใช้ในการตั้งค่าความเร็วและการสื่อสารของ CAN BUS
- 6) Start Eagle: ใช้เพื่อเริ่มการทำงานของโปรเจกต์ที่สร้างขึ้น
- 7) Names I/O: ชื่อของอินพุต/เอาต์พุตที่ใช้งานในโปรแกรม
- 8) Compile: ใช้เพื่อคอมไพล์โปรแกรมก่อนที่จะทำการอัปโหลด
- 9) Status Display: แสดงสถานะปัจจุบันของข้อมูลที่ถูกอ่านจากฐานข้อมูล

3.3.2 การตั้งค่า CAN-ID และตัวแปรสำหรับการรับส่งข้อมูล

การพัฒนาโปรแกรมควบคุมโมดูล CAN I/O PLC จำเป็นต้องดำเนินการตั้งค่าการรับส่งข้อมูลในหน้า Edit CAN_DB ซึ่งเป็นหน้าต่างสำหรับการกำหนดข้อมูลที่ต้องการรับหรือส่งผ่านระบบ CAN BUS โดยผู้ใช้งานต้องสร้างตัวแปรที่ใช้ในการจัดเก็บข้อมูล เพื่อให้ระบบสามารถประมวลผลและจัดเก็บข้อมูลการสื่อสารได้อย่างถูกต้องและแม่นยำ หลังจากการตั้งค่าข้อมูลและตัวแปรเรียบร้อยแล้ว จึงสามารถดำเนินการเขียนโปรแกรมและตั้งค่าฟังก์ชันที่จำเป็นสำหรับการควบคุมระบบของรถ ATV

ในส่วนของการตั้งค่า CAN-ID และตัวแปรสำหรับการรับส่งข้อมูลในโมดูล CAN ที่ 1 ซึ่งรับค่าจากสวิตช์เพียงอย่างเดียว จะถูกแบ่งออกเป็นหมวดหมู่ของสวิตช์ทั้งหมด 2 กลุ่ม โดยกลุ่มแรกมีชื่อว่า SWG1 ซึ่งถูกกำหนดชื่อตัวแปรแทน CAN-ID: 0X101 โดยมีการกำหนดความเร็วในการส่งข้อมูลที่ 100 ms ในตัวแปร SWG1 หรือ CAN-ID: 0X101 จะเก็บข้อมูลที่เป็นส่วนของสวิตช์ควบคุมสัญญาณไฟภายในตัวรถ เช่น สวิตช์ไฟเลี้ยว ไฟหน้า แตร เป็นต้น ข้อมูลเหล่านี้จะถูกรวมอยู่ใน CAN-ID นี้ เพื่อแยกเป็นกลุ่มของข้อมูลที่เกี่ยวข้องกับการควบคุมไฟต่าง ๆ ภายในรถ และในกลุ่มที่ 2 มีชื่อว่า SWG2 ซึ่งถูกกำหนดชื่อตัวแปรแทน CAN-ID: 0X102 จะเก็บข้อมูลที่เป็นส่วนของสวิตช์ควบคุมการทำงานของมอเตอร์ขับเคลื่อน เช่น สวิตช์สตาร์ท สวิตช์เลือกโหมด และสวิตช์เดินหน้า-ถอยหลัง เป็นต้น ซึ่งสวิตช์แต่ละตัวก็จะถูกตัวตัวแปรย่อยเพื่อไว้เก็บข้อมูลแต่ละตัวต่างกันออกไป การตั้งค่า CAN-ID และรูปแบบการรับส่งข้อมูลการสื่อสารแสดงในโมดูล CAN ที่ 1 แสดงดังรูปที่ 3.7

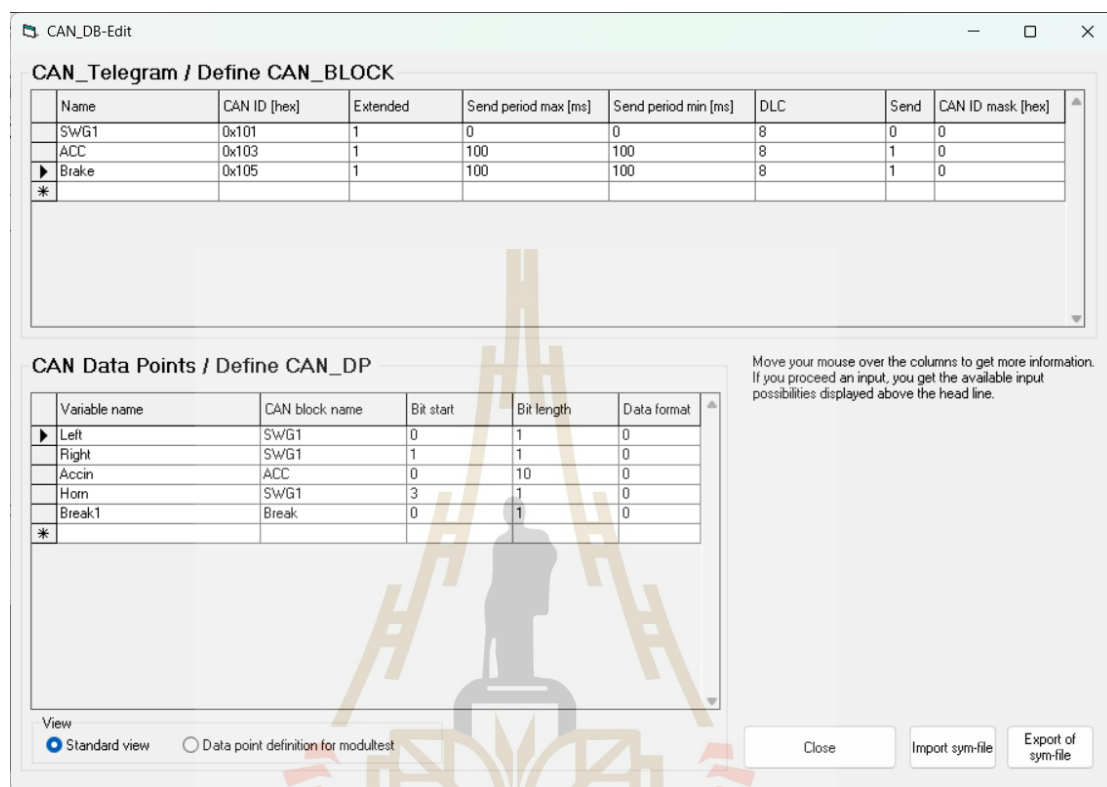




รูปที่ 3.7 หน้าต่าง Edit CAN_DB สำหรับการตั้งค่าการสื่อสารในโมดูล CAN ที่ 1

ในส่วนของการตั้งค่า CAN-ID และตัวแปรสำหรับการรับส่งข้อมูลใน โมดูล CAN ที่ 2 โมดูลนี้จะมีการกำหนดให้รับค่าของสวิตช์ไฟเลี้ยงที่ได้รับจาก โมดูล CAN ที่ 1 โดยโมดูล CAN ที่ 2 จะรับค่าจาก CAN-ID: 0X101 ซึ่งเป็นที่อยู่ของข้อมูลสวิตช์ไฟเลี้ยงที่ส่งมาจากโมดูล CAN ที่ 1 นั้น หมายความว่าโมดูล CAN ที่ 2 จะต้องสามารถรับข้อมูลจากโมดูล CAN ที่ 1 ซึ่งมีการกำหนดค่า CAN-ID เป็น 0X101 เพื่อให้สามารถรับข้อมูลที่ถูกส่งจากโมดูลนั้นได้ นอกจากนี้ โมดูล CAN ที่ 2 ยังมีการสร้าง CAN-ID ใหม่ ขึ้นมาสำหรับการสื่อสารกับโมดูล CAN อื่น ๆ ในระบบ โดยมี CAN-ID: 0X103 ซึ่งใช้ชื่อว่า ACC สำหรับส่งข้อมูลเกี่ยวกับการควบคุมคันเร่ง และ CAN-ID: 0X105 ในชื่อ Brake เพื่อใช้ในการส่งข้อมูลที่เกี่ยวข้องกับสัญญาณเบรก โมดูล CAN ที่ 2 มีหน้าที่รับข้อมูลจากโมดูล CAN ที่ 1 และส่งข้อมูลไปยังโมดูล CAN อื่น ๆ โดยผ่าน CAN-ID ที่ตั้งไว้ในกรณีของ ตัวแปรย่อย Accin ซึ่งเป็นตัวแปรที่ใช้เก็บข้อมูลเกี่ยวกับคันเร่ง ค่าของคันเร่งนั้นเป็น สัญญาณอนาล็อก ซึ่งมีลักษณะเป็นค่าต่อเนื่อง ไม่ใช่แค่เปิดหรือปิดเหมือนสวิตช์ ทำให้ต้องใช้พื้นที่ในการเก็บข้อมูลมากขึ้น ตัวแปร Accin จึงถูกกำหนดให้สามารถเก็บข้อมูลได้มากถึง 10 ตำแหน่ง เพื่อรองรับการส่งข้อมูลที่มีหลายค่าในตัวแปรเดียว เมื่อเทียบกับข้อมูลจากสวิตช์ที่มีแค่สถานะเปิดหรือปิด ซึ่งใช้พื้นที่ในการเก็บข้อมูลเพียงแค่ว่า

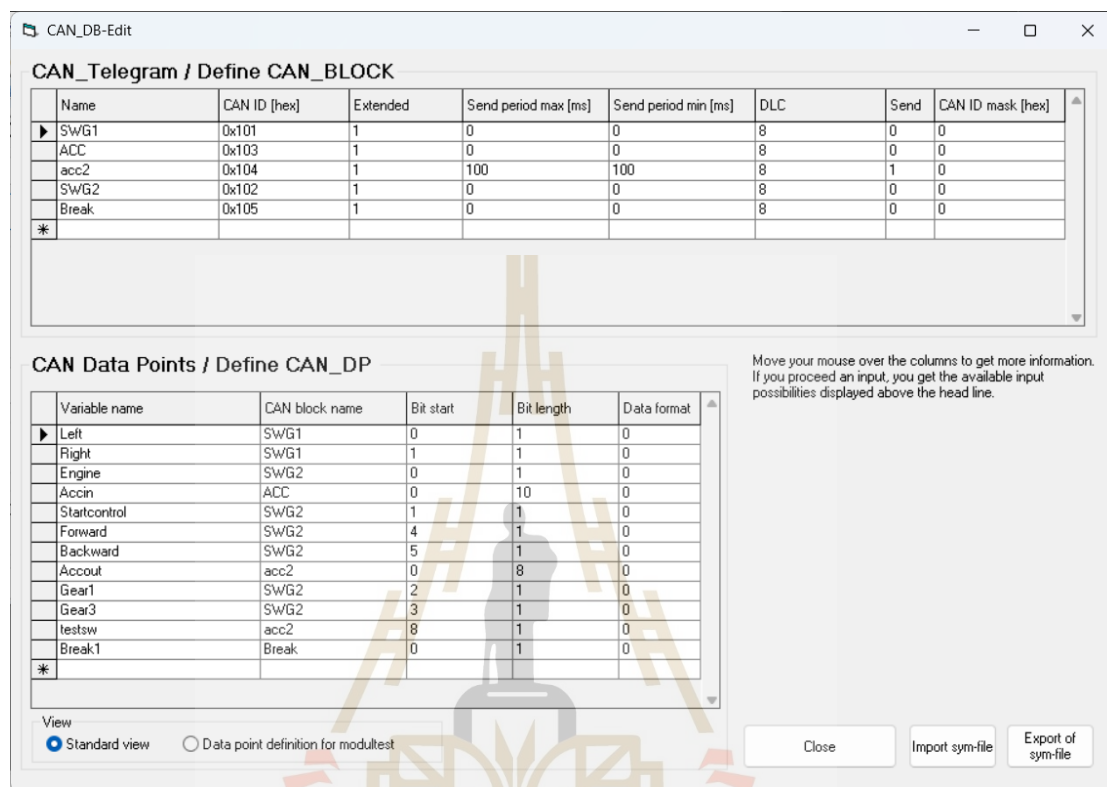
1 ตำแหน่ง การตั้งค่า CAN-ID และรูปแบบการรับส่งข้อมูลต่าง ๆ ในการสื่อสารของโมดูล CAN ที่ 2 จะถูกแสดงในรูปที่ 3.8



รูปที่ 3.8 หน้าต่าง Edit CAN_DB สำหรับการตั้งค่าการสื่อสารในโมดูล CAN ที่ 2

ในส่วนของการตั้งค่า CAN-ID และตัวแปรสำหรับการรับส่งข้อมูลใน โมดูล CAN ที่ 3 โมดูลนี้ จะไม่มีการรับค่าจากสวิตช์หรือเซ็นเซอร์ แต่จะทำหน้าที่ในการส่งข้อมูลออกจากตัวโมดูลเพื่อไป ควบคุมการทำงานของกล่องควบคุมมอเตอร์ โมดูล CAN ที่ 3 จะทำการรับคำสั่งจาก โมดูล CAN ที่ 1 และ โมดูล CAN ที่ 2 มาเพื่อทำการประมวลผลก่อนที่จะส่งข้อมูลไปยังกล่องควบคุมมอเตอร์ โดยการ ทำงานในลักษณะนี้ทำให้โมดูล CAN ที่ 3 จะเป็นตัวกลางในการประสานข้อมูลจากโมดูลที่ 1 และ 2 เพื่อควบคุมการทำงานของมอเตอร์ การตั้งค่า CAN-ID และตัวแปรสำหรับการรับส่งข้อมูลใน โมดูล CAN ที่ 3 จะมีการรับข้อมูลจากโมดูลอื่น ๆ เท่านั้น โดยจะรับค่า CAN-ID: 0X101 และ CAN-ID: 0X102 จากโมดูล CAN ที่ 1 ซึ่งเป็นค่าของสวิตช์ควบคุมภายในตัวรถ เช่น สวิตช์ไฟเลี้ยว ไฟหน้า หรือ แตร และจะเก็บข้อมูลเหล่านี้ไว้ในตัวแปรต่าง ๆ จากนั้นโมดูล CAN ที่ 3 ยังจะรับค่า CAN-ID: 0X103 และ CAN-ID: 0X105 จากโมดูล CAN ที่ 2 ซึ่งเป็นค่าคันเร่งและสัญญาณเบรก และเก็บข้อมูลเหล่านี้ ไว้ในตัวแปรเช่นเดียวกัน การตั้งค่า CAN-ID และรูปแบบการรับส่งข้อมูลในการสื่อสารของโมดูล CAN

ที่ 3 จะแสดงในรูปที่ 3.9 โดยหลังจากการตั้งค่าต่าง ๆ เสร็จสิ้นแล้ว ข้อมูลที่ถูกเก็บไว้ในตัวแปรเหล่านี้ จะถูกนำไปใช้ในการเขียนโปรแกรมควบคุมการทำงานของมอเตอร์ในขั้นตอนถัดไป



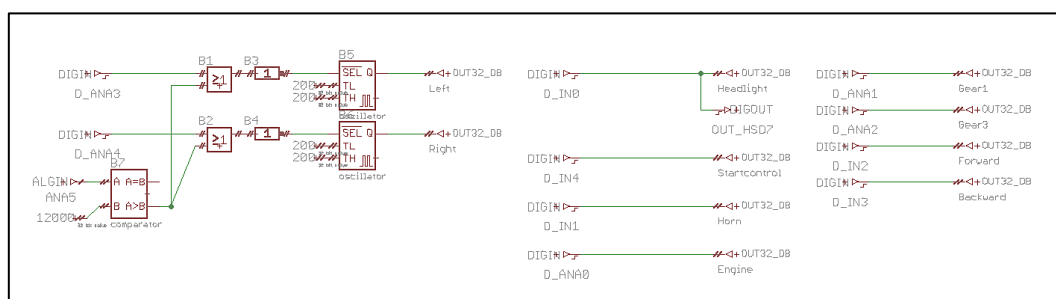
รูปที่ 3.9 หน้าต่าง Edit CAN_DB สำหรับการตั้งค่าการสื่อสารในโมดูล CAN ที่ 3

3.3.3 การเขียนโปรแกรมควบคุม CAN I/O PLC

จากตัวอย่างในรูปที่ 3.10 การเขียนโปรแกรมควบคุมโมดูล CAN I/O PLC ใช้การประมวลผลผ่านโปรแกรม EAGLE 7.30 Light ซึ่งเป็นเครื่องมือที่ช่วยให้สามารถเชื่อมต่อและควบคุมการทำงานของอุปกรณ์ต่าง ๆ ในระบบ CAN BUS ได้อย่างมีประสิทธิภาพ โปรแกรมนี้ช่วยให้ผู้ใช้งานสามารถกำหนดและออกแบบกระบวนการทำงานของระบบได้ง่ายขึ้น โดยใช้วิธีการลากและจัดเรียงเครื่องมือหรือฟังก์ชันตามลำดับการทำงานที่ต้องการ การเขียนโปรแกรมจะเริ่มต้นจากการเลือกฟังก์ชันต่าง ๆ ที่จำเป็นสำหรับการประมวลผลคำสั่ง เช่น การรับส่งข้อมูลจากเซ็นเซอร์ การสั่งการอุปกรณ์ หรือการประมวลผลสัญญาณจากโมดูลต่าง ๆ ในระบบ CAN โดยผู้ใช้งานสามารถวางเครื่องมือในลำดับที่เหมาะสมเพื่อให้การทำงานของระบบเป็นไปตามที่ต้องการ ฟังก์ชันเหล่านี้จะช่วยให้การควบคุมอุปกรณ์ต่าง ๆ เป็นไปได้อย่างถูกต้อง

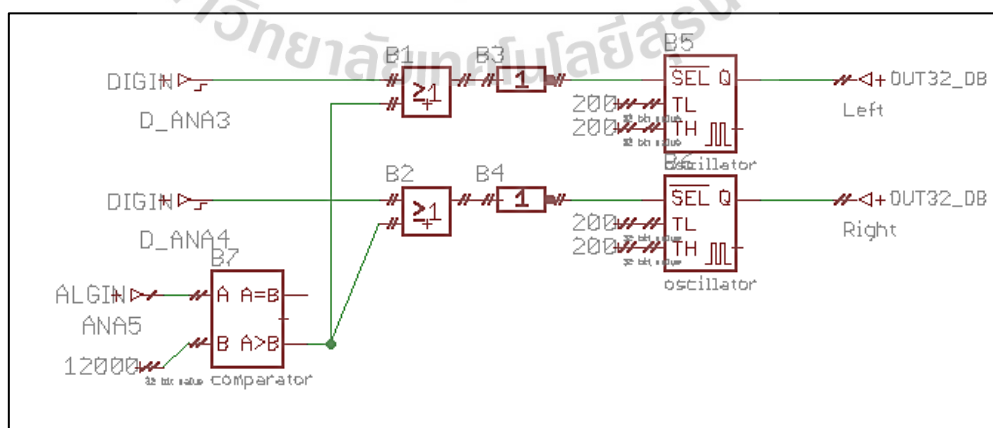
3.3.4 โปรแกรมควบคุมการทำงานของรถATV

โปรแกรมควบคุมการทำงานของรถ ATV โมดูลที่ 1 ได้ถูกตั้งโปรแกรมให้รับสัญญาณจากสวิตช์ควบคุมต่าง ๆ ภายในตัวรถ เช่น สวิตช์กุญแจ สวิตช์สตาร์ท สวิตช์เบรค สวิตช์เดินหน้า-ถอยหลัง และสวิตช์โหมดการทำงาน จากนั้นจะมีการประมวลผลในส่วนของสัญญาณไฟเลี้ยวก่อนที่จะส่งข้อมูลไปยังโมดูลที่ 2 และโมดูลที่ 3 ผ่านระบบ CAN BUS โดยที่โมดูลที่ 2 และโมดูลที่ 3 จะรอรับข้อมูลดังกล่าวเพื่อนำไปสั่งการให้ไฟและอุปกรณ์ต่าง ๆ ทำงานสอดคล้องตามข้อมูลที่ได้รับจากโมดูลที่ 1 เพื่อให้ระบบทั้งหมดทำงานประสานกัน รูปแบบการเขียนโปรแกรมแสดง ดังรูปที่ 3.11



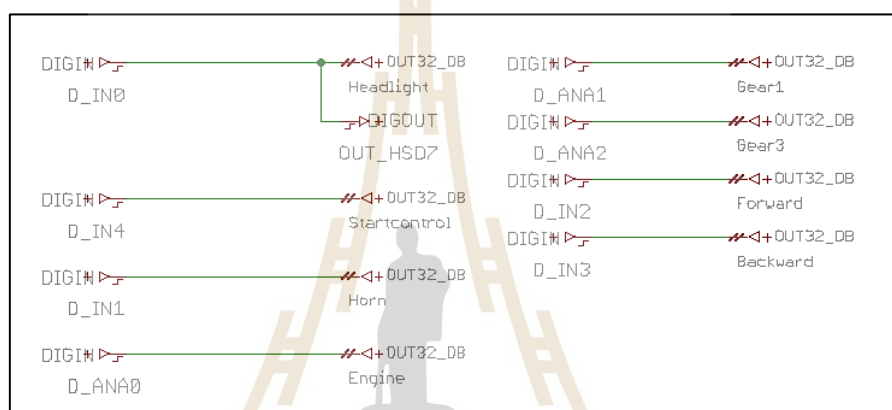
รูปที่ 3.11 โปรแกรมควบคุมการทำงานของรถ ATV โมดูลที่ 1

ในการประมวลผลของสัญญาณไฟเลี้ยว เมื่อมีการรับค่าสวิตช์ไฟเลี้ยวแล้ว โปรแกรม จะทำการประมวลผลให้เกิด สัญญาณกระพริบ และส่งคำสั่งไปยัง โมดูล CAN ที่ 2 และ โมดูล CAN ที่ 3 เพื่อหลีกเลี่ยงการเขียนโปรแกรมซ้ำซ้อนในส่วนของโปรแกรมควบคุมอื่น ๆ การส่งสัญญาณไฟ กระพริบนี้ค่อนข้างซับซ้อน เนื่องจากความสามารถของขารับสัญญาณใน โมดูล CAN นั้นแตกต่างกัน ไป ตัวอย่างเช่น ขารับสัญญาณไฟเลี้ยวซ้าย D_ANA3 และ ขารับสัญญาณไฟเลี้ยวขวา D_ANA4 จะ รับสัญญาณแบบ PULL_DOWN ซึ่งทำให้ค่าเริ่มต้นที่ออกมาคือ 0 ตลอดเวลาจนกว่าจะมีการเปิด สัญญาณไฟ การทำงานนี้ทำให้เครื่องมือ Oscillator เมื่อรับค่าเป็น 0 จะสร้างสัญญาณไฟกระพริบ ขึ้นมา ซึ่งไม่ตรงกับความต้องการ เพื่อแก้ไขปัญหานี้ จำเป็นต้องเพิ่มเครื่องมือ NOT (B3 และ B4) เพื่อทำการกลับค่าสัญญาณจากขา D_ANA3 และ D_ANA4 ก่อนที่จะส่งสัญญาณออกไปยังโมดูล CAN อื่น ๆ ซึ่งจะใช้ชื่อของตัวแปรที่ได้ตั้งไว้ก่อนหน้านี้ การใช้เครื่องมือ NOT นี้จะช่วยให้สามารถ ควบคุมและส่งสัญญาณไฟกระพริบไปยังโมดูล CAN ได้ตามต้องการ และในการสร้างสัญญาณไฟผ่า หมาก โมดูลจะรับค่าจากขารับสัญญาณ ANA5 ซึ่งขานี้จะรับสัญญาณแบบอนาล็อก โดยการเทียบค่า แรงดันไฟฟ้า 1 mV เท่ากับ 1 digit ดังนั้น เมื่อกดสัญญาณไฟผ่าหมากจะมีแรงดันที่ 12.3V หรือ 12300 digit การทำงานนี้จะใช้เครื่องมือ Comparator ช่วยในการส่งค่าเปิดไฟผ่าหมาก โดย เครื่องมือ Comparator จะเทียบค่าสัญญาณเมื่อเปิดแล้วว่า สัญญาณไฟมีค่ามากกว่าค่าที่กำหนดไว้ 12000 digit เมื่อถึงค่าดังกล่าวแล้ว ระบบจะใช้เครื่องมือ AND (B1 และ B2) เพื่อให้สัญญาณส่งค่า ออกไปโดยในวงจรการสื่อสาร CAN BUS นี้จะใช้คำสั่ง Out32_DB เพื่อส่งค่าที่ได้ไปเก็บไว้ในตัวแปร ย่อยที่ได้กำหนดไว้ก่อนหน้านี้ ซึ่งเป็นตัวแปรที่ใช้ในการควบคุมการทำงานของอุปกรณ์ต่าง ๆ หลังจากนั้น เมื่อโมดูลอื่น ๆ รับค่าแล้ว ก็จะสามารถทำงานพร้อมกันตามคำสั่งที่ส่งไปจากโมดูลที่มีการ ประมวลผล รูปแบบการเขียนโปรแกรมควบคุมไฟเลี้ยว แสดงดังรูปที่ 3.12



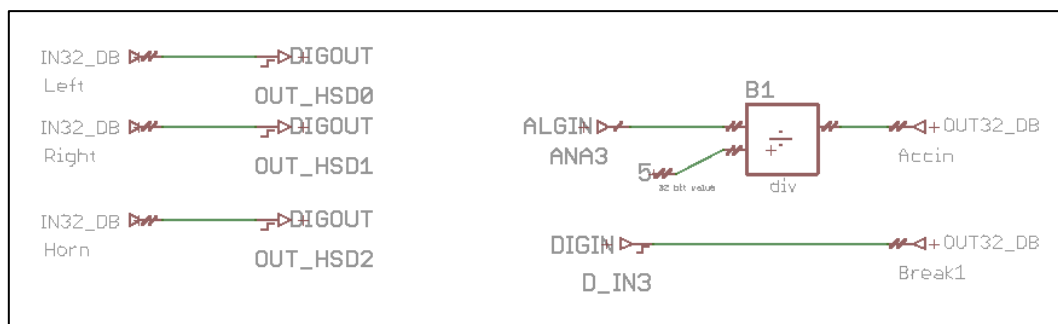
รูปที่ 3.12 โปรแกรมควบคุมการทำงานของไฟเลี้ยวในรถATV

ในอีกส่วนหนึ่งของการประมวลผลที่รับสัญญาณมาจากสวิทช์ ยังไม่ได้มีการประมวลผลทันที เนื่องจากจำเป็นต้องทำการประมวลผลร่วมกับสัญญาณที่ได้รับจากโมดูล CAN ที่ 2 เพื่อให้สามารถทำการควบคุมและประมวลผลข้อมูลได้อย่างถูกต้อง ดังนั้นในขั้นตอนนี้สัญญาณจากสวิทช์จะถูกส่งเข้าไปเก็บไว้ในตัวแปรย่อย ที่ได้มีการกำหนดไว้ก่อนหน้านี้เพื่อรอการประมวลผลในขั้นตอนถัดไป โดยการเก็บข้อมูลในตัวแปรนี้จะช่วยให้สามารถนำข้อมูลไปใช้งานร่วมกับสัญญาณจากโมดูล CAN ที่ 2 ในภายหลัง รูปแบบการเขียนโปรแกรมเพื่อรับค่าจากสวิทช์และเก็บข้อมูลไว้ในตัวแปรย่อยจะถูกแสดงใน รูปที่ 3.13



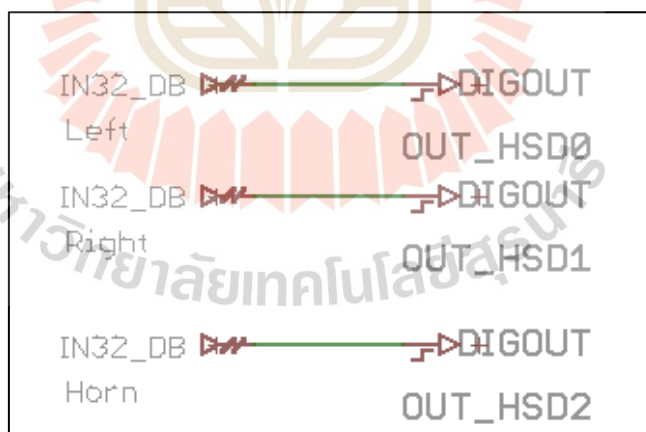
รูปที่ 3.13 โปรแกรมรับค่าสวิทช์และเก็บข้อมูลไว้ในตัวแปรย่อยในโมดูล CAN ที่ 1

โปรแกรมควบคุมการทำงานของรถ ATV สำหรับโมดูลที่ 2 ได้ถูกตั้งโปรแกรมให้รับสัญญาณ CAN BUS จากโมดูลที่ 1 เช่น สัญญาณไฟเลี้ยว และแตร โดยโมดูลที่ 2 จะทำการสั่งการให้หลอดไฟเลี้ยวซ้าย-ขวาและแตร ทำงานตามคำสั่งที่ได้รับจากโมดูลที่ 1 ทำให้การควบคุมอุปกรณ์ต่าง ๆ ในระบบไฟฟ้าภายในรถ ATV สามารถทำได้ตามต้องการ นอกจากนี้ โมดูลที่ 2 ยังมีหน้าที่ในการรับ สัญญาณจากเซ็นเซอร์เบรก และคันเร่ง เพื่อนำข้อมูลเหล่านี้ส่งต่อไปยัง โมดูลที่ 3 ซึ่งจะทำให้การควบคุมการทำงานของกล่องคอนโทรลเลอร์หลักในระบบรถ ATV โดยข้อมูลจากเซ็นเซอร์เบรกและคันเร่งจะถูกใช้ในการประมวลผลเพื่อควบคุมการทำงานของมอเตอร์และระบบต่าง ๆ ที่เกี่ยวข้อง ในภายหลังรูปแบบการเขียนโปรแกรมในการรับสัญญาณและการส่งต่อข้อมูลจากโมดูลที่ 2 แสดงในรูปที่ 3.14



รูปที่ 3.14 โปรแกรมควบคุมการทำงานของรถ ATV โมดูลที่ 2

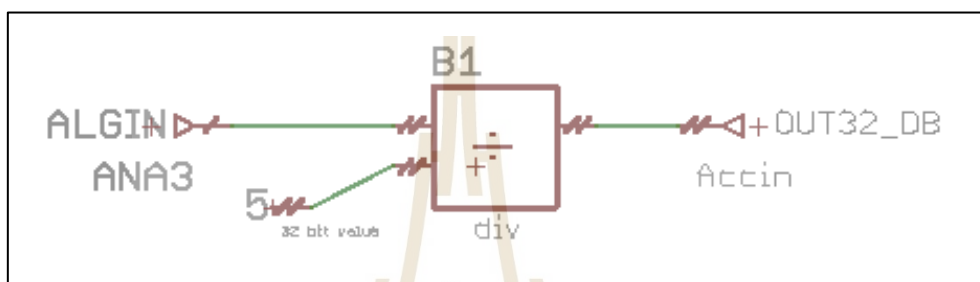
ในส่วนของการรับค่าสัญญาณนั้น โปรแกรมจะทำการเรียกตัวแปรย่อยที่ได้กำหนดไว้ก่อนหน้านี้ โดยใช้คำสั่ง IN32_DB เพื่อดึงข้อมูลที่เก็บไว้ในตัวแปรย่อยที่เกี่ยวข้องออกมา เมื่อข้อมูลถูกดึงมาแล้ว โปรแกรมจะทำการส่งค่าออกไปยังขาสัญญาณที่เชื่อมต่อกับอุปกรณ์ต่าง ๆ อยู่ โดยใช้คำสั่ง DIGOUT เพื่อส่งให้อุปกรณ์ที่เชื่อมต่อกับขาสัญญาณ เช่น หลอดไฟ หรือ รีเลย์ นั้นทำงานตามคำสั่งที่ได้รับ การส่งออกสัญญาณนี้จะขึ้นอยู่กับการควบคุมจากโมดูล CAN ที่ 1 ซึ่งรับคำสั่งจากผู้ใช้งาน เพื่อทำให้หลอดไฟหรือรีเลย์ทำงานตามที่ต้องการในระบบการควบคุม รูปแบบการเขียนโปรแกรมดึงค่าจากตัวแปรมาใช้งาน แสดงในรูปที่ 3.15



รูปที่ 3.15 โปรแกรมดึงค่าจากตัวแปรมาใช้งานของรถ ATV โมดูลที่ 2

ในอีกส่วนหนึ่งของโปรแกรมการทำงานของ โมดูล CAN ที่ 2 คือการรับค่าสัญญาณจาก คันเร่ง ซึ่งคันเร่งจะส่งออกสัญญาณอนาล็อกที่มีค่าตั้งแต่ 0 ถึง 5V โมดูล CAN จะรับค่าได้ในรูปแบบอนาล็อก ซึ่งจะมีค่าตั้งแต่ 0 ถึง 5000 digit โดยที่ 1 mV เท่ากับ 1 digit นั้นหมายความว่า

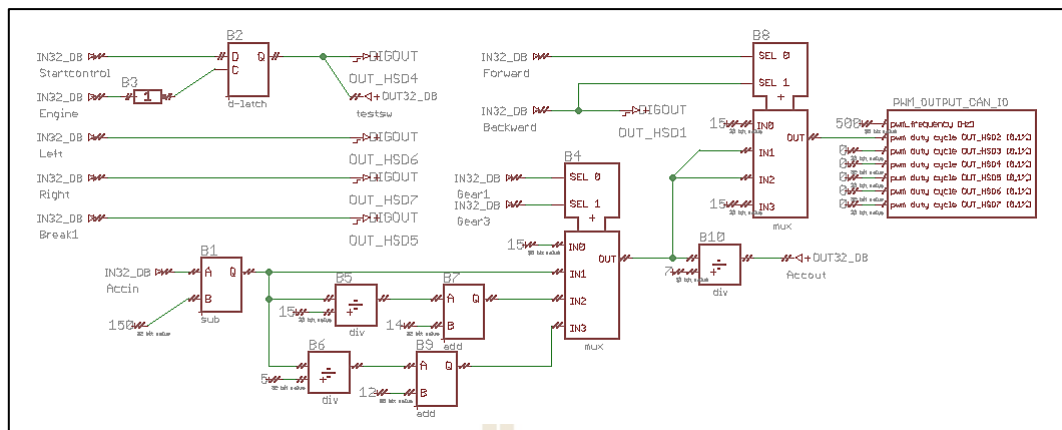
เมื่อคันเร่งส่งค่า 5V ก็จะถูกแปลงเป็น 5000 digit ซึ่งเท่ากับ 5V หรือค่าที่สูงสุดที่โมดูล CAN สามารถรับได้อย่างไรก็ตาม การส่งค่า 5000 digit นั้นจะต้องใช้พื้นที่ในการจัดเก็บข้อมูลมากถึง 13 bit ซึ่งมีข้อมูลที่เยอะเกินไป ทำให้การส่งข้อมูลในระบบ CAN BUS ช้าลง เนื่องจากต้องใช้เวลาในการส่งข้อมูลมากขึ้นเพื่อแก้ไขปัญหานี้ จึงทำการหารค่าที่อ่านได้จากคันเร่ง เพื่อลดขนาดข้อมูลที่ส่งไปยังโมดูลอื่น ๆ วิธีการนี้ช่วยลดความยาวของข้อมูล และทำให้การส่งข้อมูลเป็นไปอย่างรวดเร็วขึ้น ตัวอย่างการเขียนโปรแกรมเพื่อรับค่าคันเร่ง และลดขนาดข้อมูลจะแสดงใน รูปที่ 3.15



รูปที่ 3.16 โปรแกรมรับค่าสัญญาณคันเร่งของโมดูล CAN ที่ 2

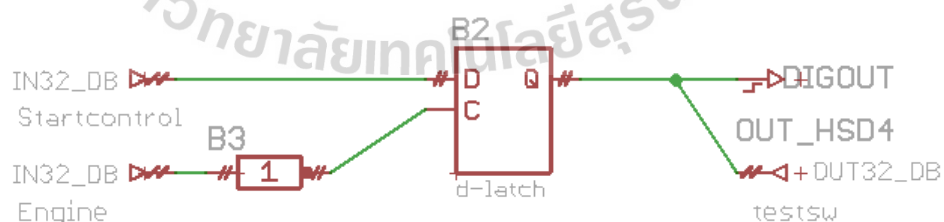
โปรแกรมควบคุมการทำงานของรถ ATV ถูกออกแบบให้โมดูลที่ 3 รับสัญญาณ CAN BUS จากโมดูลที่ 1 และโมดูลที่ 2 เพื่อนำมาประมวลผลและควบคุมระบบต่าง ๆ เช่น สวิตช์สตาร์ท และ สวิตช์เดินหน้า-ถอยหลัง โหมดการทำงาน โดยกำหนดเงื่อนไขเพื่อเพิ่มความปลอดภัยต่อการใช้งาน โดยผู้ใช้ต้องกดสวิตช์สตาร์ทเพื่อเปิดระบบมอเตอร์ก่อน จากนั้นต้องเลือกโหมดเดินหน้าหรือถอยหลังก่อนที่คันเร่งจะสามารถส่งงานให้มอเตอร์หมุนได้ นอกจากนี้ มีการเขียนโปรแกรมให้มี 3 โหมดการขับขี่ เพื่อปรับการทำงานของคันเร่งให้เหมาะสมกับการใช้งาน ได้แก่ โหมด Eco, โหมด Normal และโหมด Sport โดยโหมด Eco จำกัดกำลังคันเร่งไว้ที่ 10% ของความสามารถสูงสุดของกล่องควบคุมมอเตอร์ แม้ว่าผู้จะใช้จะเหยียบคันเร่งเต็มที่ ส่วนโหมด Normal จำกัดกำลังคันเร่งไว้ที่ 20% และโหมด Sport ให้คันเร่งทำงานที่ 100%

นอกจากนี้ โมดูลที่ 3 ยังควบคุมคันเร่งให้ทำงานตามโหมดที่ได้รับคำสั่ง โดยโมดูล CAN I/O PLC ส่งสัญญาณ PWM ไปยังวงจร Low Pass Filter เพื่อแปลงสัญญาณเป็นอนาล็อกไปควบคุมกล่องควบคุมมอเตอร์ รวมถึงควบคุมระบบเบรกและไฟเลี้ยวซ้าย-ขวา เพื่อให้การขับขี่ของรถ ATV เป็นไปตามการควบคุมที่ต้องการ รูปแบบการเขียนโปรแกรมในการรับสัญญาณจากโมดูลที่ 2 แสดงในรูปที่ 3.17



รูปที่ 3.17 โปรแกรมควบคุมการทำงานของรถ ATV โมดูลที่ 3

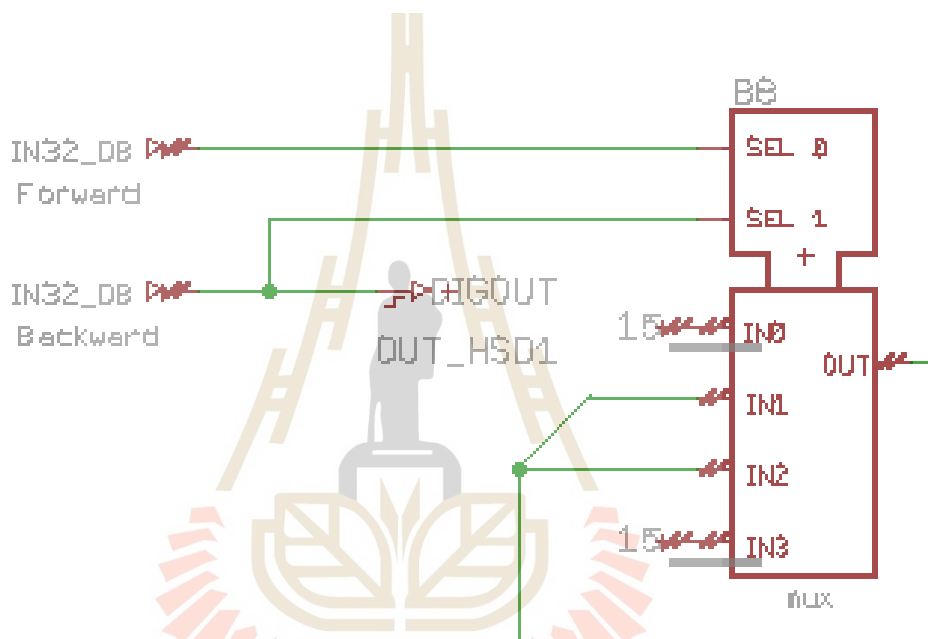
ในการเขียนโปรแกรมเงื่อนไขด้านความปลอดภัยของการใช้งานนั้น ได้ถูกกำหนดเงื่อนไขการใช้งานไว้ว่า เมื่อเปิดสวิตช์กุญแจแล้วรถยังไม่สามารถขับเคลื่อนได้จนกว่าจะกด ปุ่ม Engine Start เพื่อป้องกันการหลงลืมของผู้ใช้งาน และเพื่อความปลอดภัยในการใช้งานรถ ATV โดยใช้เครื่องมือ d-latch เพื่อคงค่าสถานะของ ปุ่ม Engine Start ซึ่งจะช่วยให้สถานะการกดปุ่ม Engine Start คงอยู่แม้ผู้ใช้งานจะปล่อยมือจากปุ่ม เมื่อกดปุ่ม Engine Start แล้ว รถจะสามารถขับเคลื่อนได้ตามปกติ แต่หากผู้ใช้งาน ปิดกุญแจ แล้วเปิดใหม่ รถจะไม่สามารถขับเคลื่อนได้จนกว่าจะกด ปุ่ม Engine Start อีกครั้งเพื่อเริ่มการใช้งานใหม่ ซึ่งเป็นการเพิ่มความปลอดภัยในการใช้งานรถอย่างมีประสิทธิภาพรูปแบบการเขียนโปรแกรมด้านความปลอดภัยของสวิตช์ Engine Start แสดงในรูปที่ 3.18



รูปที่ 3.18 โปรแกรมเงื่อนไขด้านความปลอดภัยของการใช้งาน Engine Start

อีกหนึ่งเงื่อนไขด้านความปลอดภัยของการใช้งานคือ สวิตช์สั่งงานเดินหน้า, ถอยหลัง และเกียร์ว่าง ซึ่งมีการเขียนเงื่อนไขในส่วนนี้เพิ่มเข้ามาเพื่อเพิ่มความปลอดภัยในการใช้งาน เนื่องจากเดิมกล่องควบคุมมอเตอร์นั้นไม่มีการกำหนดเกียร์ว่าง เมื่อเหยียบคันเร่งรถก็จะเคลื่อนที่ทันที

ซึ่งอาจเกิดอันตรายได้ในบางกรณี เพื่อป้องกันปัญหานี้ จึงมีการเขียนโปรแกรมให้ตัดการทำงานของคันเร่ง ในกรณีที่ไม่ได้อยู่ในเกียร์ที่เลือกไว้ เช่น เกียร์เดินหน้าหรือถอยหลัง โดยเมื่อเหยียบคันเร่งแล้วรถจะไม่สามารถเคลื่อนที่ได้ ถ้าไม่ได้อยู่ในตำแหน่งเกียร์ที่ถูกต้อง โปรแกรมนี้ใช้เครื่องมือ MUX ซึ่งจะช่วยเลือกค่าของคันเร่งตามคำสั่งที่ได้รับจากการสั่งงานเดินหน้าหรือถอยหลัง ก่อนที่จะส่งค่าดังกล่าวไปควบคุมการทำงานของกล่องควบคุมมอเตอร์ การใช้เครื่องมือ MUX จะช่วยให้โปรแกรมสามารถเลือกค่าได้อย่างถูกต้องตามคำสั่งและทำให้การทำงานของมอเตอร์เป็นไปตามที่ต้องการ รูปแบบการเขียนโปรแกรมด้านความปลอดภัยของ สวิตช์เดินหน้า, ถอยหลัง และเกียร์ว่าง จะแสดงในรูปที่ 3.19



รูปที่ 3.19 โปรแกรมด้านความปลอดภัยของ สวิตช์เดินหน้า, ถอยหลัง และเกียร์ว่าง

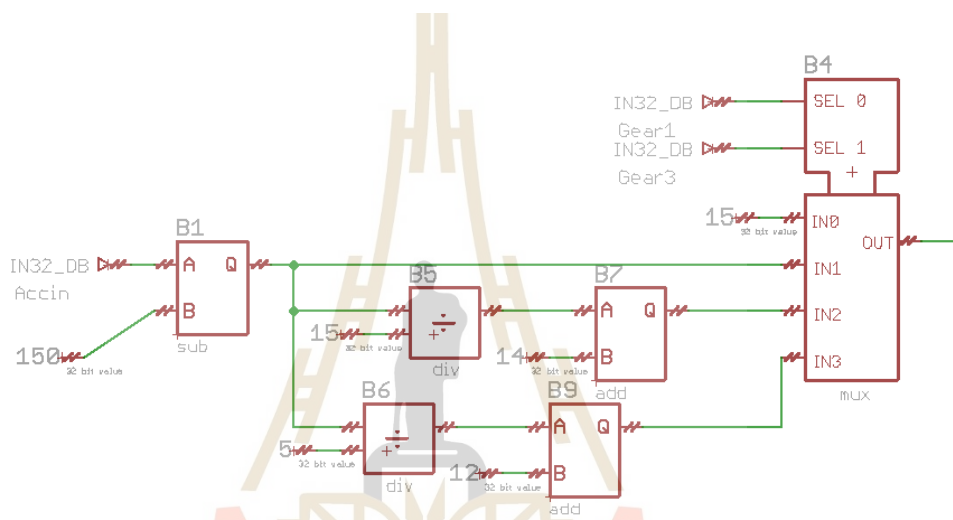
ในส่วนของโหมดการทำงาน โปรแกรมได้รับการออกแบบให้สามารถจัดการการทำงานของ คันเร่งให้ทำงานแตกต่างกันในแต่ละโหมด โดยมีการกำหนดโหมดหลัก ๆ คือ Eco, Normal, และ Sport โปรแกรมจะเริ่มต้นจากการรับค่าคันเร่ง เข้ามามาก่อนที่จะทำการประมวลผลการประมวลผลค่าคันเรานั้นจะใช้เครื่องมือ MUX เพื่อเลือกและคำนวณค่าของคันเร่งตามโหมดที่เลือกไว้

โหมด ECO: ค่าคันเร่งที่รับเข้ามาจะถูกหารด้วย 15 ผลลัพธ์ที่ได้จะเป็นเพียง 10% ของค่าคันเร่งที่ได้รับ ซึ่งจะช่วยลดการตอบสนองของคันเร่งเพื่อประหยัดพลังงานและเพิ่มความประหยัดในโหมดนี้

โหมด Normal: ค่าคันเร่งจะถูกหารด้วย 5 ผลลัพธ์ที่ได้จะออกมาเพียง 20% ของค่าคันเร่งที่ได้รับ ซึ่งจะทำให้การตอบสนองของคันเร่งเป็นปกติ สามารถขับขี่และควบคุมรถได้ง่าย

โหมด Sport: ในโหมดนี้ ค่าคันเร่งจะไม่ถูกหารแต่จะส่งออกไปโดยตรง ซึ่งทำให้การตอบสนองของคันเร่งเต็มที่และรวดเร็วที่สุด

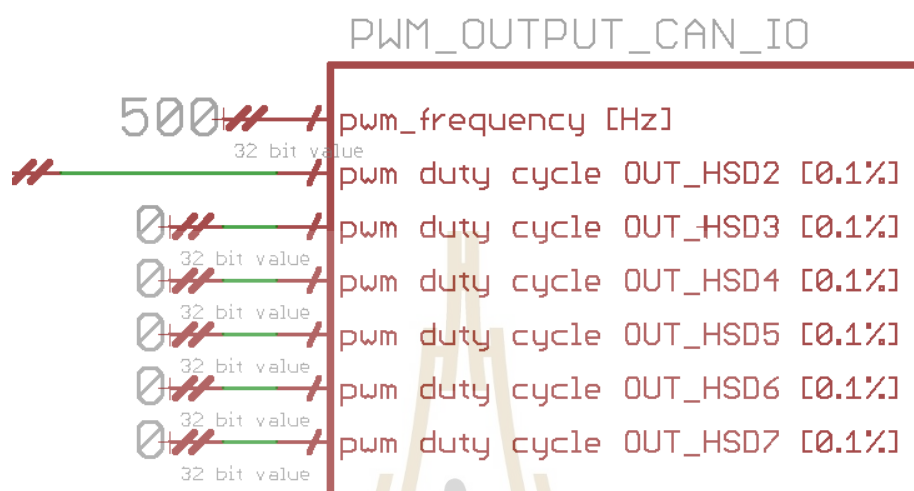
การประมวลผลในแต่ละโหมดจะช่วยให้การขับขี่มีความหลากหลายและเหมาะสมกับสภาพการขับขี่หรือความต้องการของผู้ขับขี่ รูปแบบการเขียนโปรแกรมเลือกโหมดการใช้งานแสดงดังในที่ 3.20



รูปที่ 3.20 โปรแกรมเลือกโหมดการใช้งาน

การส่งค่าคันเร่งเพื่อควบคุมการทำงานของกล่องควบคุมมอเตอร์นั้น โปรแกรมจะทำการนำค่าคันเร่งที่ได้รับการประมวลผลจากการเลือกโหมด (Eco, Normal, Sport) แล้ว เพื่อสร้างเป็น สัญญาณ PWM โดยใช้เครื่องมือ PWM_OUTPUT_CAN_IO ซึ่งเครื่องมือนี้จะช่วยในการสร้างสัญญาณ PWM ที่ต้องการ การสร้างสัญญาณ PWM นี้จะต้องมีการกำหนดให้สัญญาณออกที่ขาส่งสัญญาณที่กำหนดไว้ในระบบ และต้องตั้งค่าความถี่ของสัญญาณ PWM โดยความถี่ที่เหมาะสมสำหรับการใช้งานนี้คือ 500 Hz ซึ่งเหมาะสมสำหรับการควบคุมความเร็วของมอเตอร์ในระบบ อย่างไรก็ตามเนื่องจากกล่องควบคุมมอเตอร์ไม่สามารถใช้สัญญาณ PWM ในการควบคุมการทำงานได้โดยตรง จึงจำเป็นต้องมีการใช้วงจร Low Pass Filter เพื่อแปลงสัญญาณ PWM ที่ได้รับมาให้กลายเป็นสัญญาณอนาล็อกที่สามารถใช้งานร่วมกับกล่องควบคุมมอเตอร์ การทำงานของ Low Pass Filter จะช่วยกรองสัญญาณ PWM และให้ผลลัพธ์เป็นสัญญาณอนาล็อกที่คงที่ ซึ่งสามารถส่งไปควบคุมการทำงานของ

มอเตอร์ได้อย่างแม่นยำและเหมาะสม รูปแบบการเขียนโปรแกรมสร้างสัญญาณPWM แสดงดังในรูปที่ 3.21



รูปที่ 3.21 โปรแกรมสร้างสัญญาณPWM

3.4 การกรองสัญญาณ PWM เป็นสัญญาณอนาล็อก

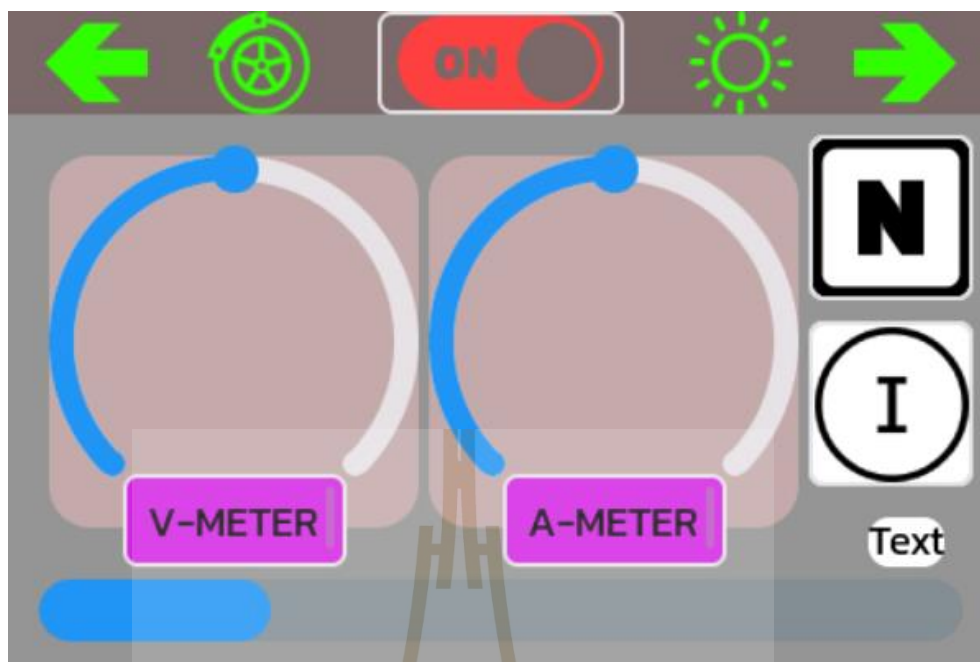
เนื่องจากโมดูล CAN I/O PLC ไม่สามารถส่งสัญญาณอนาล็อก ออกไปควบคุมกล่องควบคุมมอเตอร์ได้ แต่สามารถส่งสัญญาณ Pulse Width Modulation (PWM) ได้ ดังนั้น จึงจำเป็นต้องแปลงสัญญาณ PWM ให้เป็นสัญญาณอนาล็อก เพื่อให้กล่องควบคุมมอเตอร์สามารถรับและประมวลผลได้อย่างถูกต้อง การแปลงสัญญาณนี้ใช้ หลักการของ Low Pass Filter (LPF) ซึ่งทำหน้าที่กรองความถี่สูงของสัญญาณ PWM ออกไป ทำให้สัญญาณที่ได้เป็นแรงดันเฉลี่ยที่สามารถใช้แทนสัญญาณอนาล็อก ได้ ในงานวิจัยนี้ได้คำนวณจากตัวแปลงที่มีอยู่ คือ สัญญาณ PWM แรงดัน 5V ความถี่ที่ 500 Hz จะใช้ความต้านที่ 4,700 โอห์ม และคาปาซิเตอร์ 10 ไมโครฟารัด การต่อวงจรกรองความถี่ Low Pass Filter แสดงในรูปที่ 3.22 วิธีนี้ช่วยให้กล่องควบคุมมอเตอร์สามารถรับค่าและตอบสนองต่อการควบคุมจากโมดูล CAN I/O PLC ได้อย่างถูกต้องและราบรื่น



รูปที่ 3.22 การต่อวงจรกรองความถี่ Low Pass Filter

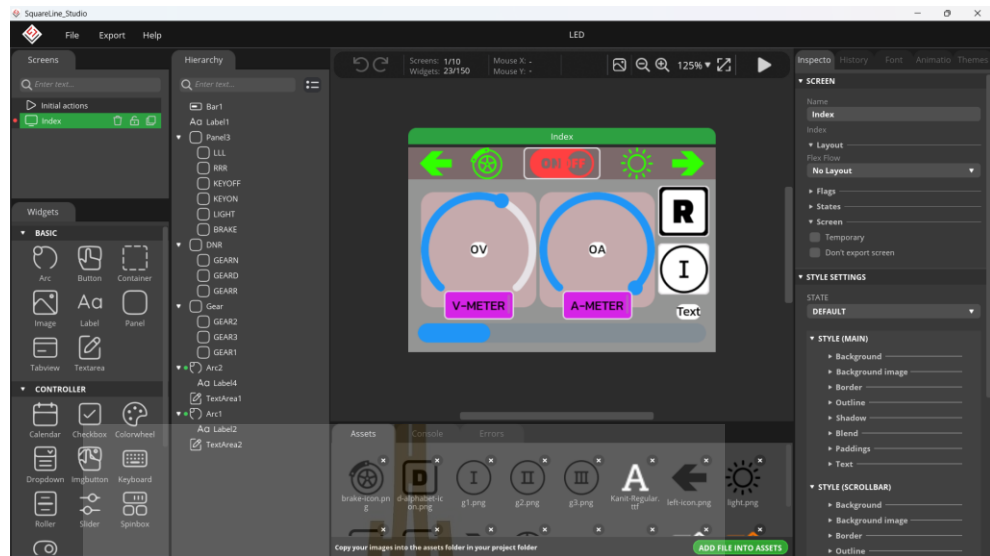
3.5 จอแสดงผลของระบบ CAN BUS

จอแสดงผลของระบบ CAN BUS ถูกเลือกใช้จอ ATD3.5-S3 ร่วมกับ CAN BUS Shield โดยเริ่มจากการออกแบบอินเทอร์เฟซผ่านโปรแกรม Square Line Studio ซึ่งเป็นเครื่องมือที่ช่วยในการสร้างอินเทอร์เฟซและสามารถออกแบบได้ตามต้องการ ในหน้าจอนี้ถูกออกแบบให้แสดงสถานะการทำงานต่าง ๆ ภายในตัวรถ ข้อมูลทั้งหมดจะถูกส่งและแสดงผลแบบเรียลไทม์บนจอแสดงผล ซึ่งการออกแบบหน้าจอแสดงในรูปที่ 3.23 หลังจากออกแบบหน้าจอเสร็จแล้ว การเขียนโปรแกรมควบคุมดำเนินการผ่าน Visual Studio Code พร้อมติดตั้ง Platform IO Extension โดยใช้ไลบรารี LVGL เพื่อสร้างหน้าจอตามที่ออกแบบใน Square Line Studio และใช้ไลบรารี twai เพื่ออ่านค่าจาก CAN BUS และนำข้อมูลมาแสดงผลบนหน้าจอ

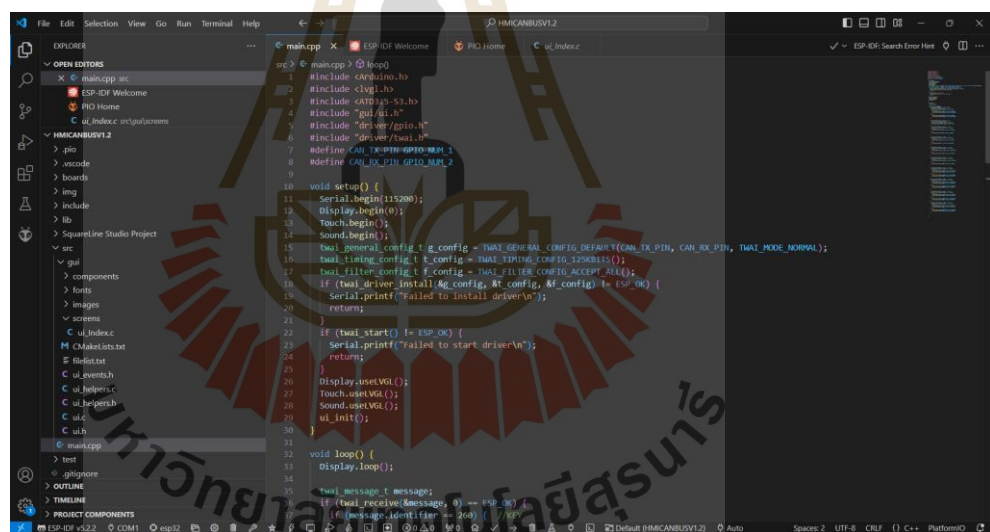


รูปที่ 3.23 ตัวอย่างจอแสดงผลของระบบ CAN BUS

ในการสร้างหน้าจอแสดงผลนั้น ขั้นตอนแรกคือการออกแบบหน้าจอแสดงผล ในขั้นตอนนี้จะทำการออกแบบหน้าจอโดยใช้โปรแกรม Square Line Studio ซึ่งเป็นเครื่องมือที่ช่วยให้ผู้ใช้งานสามารถออกแบบ อินเทอร์เฟซ ได้อย่างสะดวก โดยเริ่มจากการสร้าง ไอคอน และ องค์ประกอบต่าง ๆ ที่จะใช้แสดงผลบนหน้าจอให้ครบถ้วนตามที่ต้องการ เช่น การแสดงสถานะของ คันเร่ง เบรก ไฟเลี้ยว โหมดการทำงาน หรือสถานะของมอเตอร์ รวมไปถึงกราฟิกต่าง ๆ ที่จะช่วยให้ผู้ใช้งานเข้าใจข้อมูลได้ง่าย โปรแกรมการออกแบบหน้าจอแสดงในรูปที่ 3.24



รูปที่ 3.23 โปรแกรม Square Line Studio

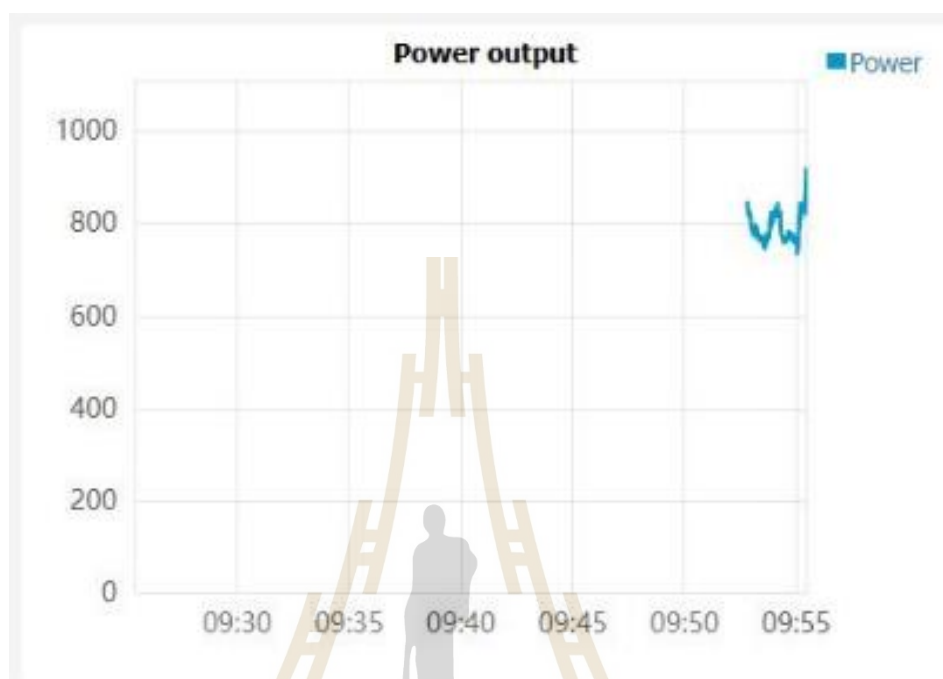


รูปที่ 3.13 โปรแกรมคำสั่งจอร์ับข้อมูลจาก CAN BUS

3.6 การวัดกำลังฟ้า โดยใช้ V-BOX ร่วมกับ Power Meter

เนื่องจาก BMS ของแบตเตอรี่ไม่สามารถสื่อสารผ่าน CAN BUS ได้โดยตรง จึงใช้ DC Power Meter ในการอ่านค่าพลังงานที่ใช้จากแบตเตอรี่ จากนั้นใช้ V-BOX ที่เชื่อมต่อกับ DC Power Meter เพื่อส่งค่าพลังงานเข้าสู่ V-Net ซึ่งทำหน้าที่เก็บข้อมูลการใช้กำลังไฟฟ้าแบบเรียลไทม์ แสดงในรูปที่ 3.14 ข้อมูลที่ได้จาก V-Net ถูกนำมาใช้ในการวิเคราะห์ผลการทดลอง เพื่อประเมินแนวโน้มการใช้

กำลังไฟฟ้าของรถ ATV ในแต่ละโหมดการทำงาน ซึ่งช่วยให้สามารถวิเคราะห์ประสิทธิภาพและปรับปรุงระบบให้เหมาะสมกับการใช้งานได้



รูปที่ 3.14 ตัวอย่างข้อมูลการวัดกำลังไฟฟ้าที่ใช้ในรถ ATV

3.3 ขั้นตอนการทดลอง

ในการทดสอบระบบ CAN BUS สำหรับควบคุมการทำงานของรถ ATV ได้ดำเนินการตามลำดับขั้นตอนดังต่อไปนี้ โดยเริ่มจากการทดสอบระบบ CAN BUS ซึ่งทำการตรวจสอบสัญญาณการสื่อสารระหว่างโมดูลทั้งสามโมดูล โดย โมดูลที่ 1 ทำหน้าที่รับข้อมูลจากสวิตช์ควบคุมต่าง ๆ เช่น สวิตช์กุญแจและไฟเลี้ยว จากนั้นส่งข้อมูลไปยังโมดูลอื่น โมดูลที่ 2 ทำหน้าที่รับสัญญาณจากคันเร่งและเบรก เพื่อนำไปแสดงผลและส่งต่อไปยัง โมดูลที่ 3 ซึ่งทำหน้าที่รับค่าการทำงานของเกียร์และโหมดการทำงาน พร้อมทั้งแสดงสถานะต่าง ๆ เช่น เกียร์เดินหน้า ถอยหลัง หรือเกียร์ว่าง นอกจากนี้ยังได้ทดสอบการแสดงผลแบบเรียลไทม์ของค่าต่าง ๆ บนหน้าจอ ATD3.5-S และทำการวิเคราะห์ค่าที่แสดงเพื่อพิจารณาความถูกต้องของข้อมูลที่รับ-ส่งผ่านระบบ CAN BUS หลังจากทดสอบระบบ CAN BUS แล้ว จึงดำเนินการทดสอบการควบคุมมอเตอร์ไฟฟ้า โดยเริ่มจากการส่งสัญญาณจากโมดูล CAN BUS ไปยังกล่องควบคุมมอเตอร์ไฟฟ้า จากนั้นทำการทดสอบการควบคุมคันเร่งในโหมดการทำงานต่าง ๆ ได้แก่ Eco, Normal และ Sport ซึ่งในขั้นตอนนี้ได้มีการใช้งานวงจร Low Pass Filter เพื่อแปลงสัญญาณ PWM ให้เป็นสัญญาณอนาล็อก เพื่อให้การควบคุมมอเตอร์ไฟฟ้ามีความแม่นยำยิ่งขึ้น

สุดท้ายได้ทำการบันทึกค่าความเร็วสูงสุดและกำลังไฟฟ้าที่ใช้ในแต่ละโหมด เพื่อเปรียบเทียบการทำงานของระบบและวิเคราะห์ผลการทดสอบ



บทที่ 4

ผลการวิจัยและอภิปรายผล

4.1 บทนำ

จากวิธีการดำเนินการวิจัยที่ได้กล่าวมา บทนี้จะนำเสนอผลลัพธ์ที่ได้จากการศึกษาระบบ CAN BUS ซึ่งทำหน้าที่เป็นระบบสื่อสารหลักระหว่างโมดูลควบคุมต่าง ๆ ภายในระบบควบคุมรถ ATV ที่ถูกออกแบบและพัฒนาเพื่อเพิ่มประสิทธิภาพในการควบคุมและสั่งการ โดยเฉพาะการควบคุมผ่าน CAN I/O PLC ที่ทำหน้าที่เป็นหน่วยประมวลผลและส่งคำสั่งระหว่างโมดูลต่าง ๆ ผลการศึกษาแสดงให้เห็นว่า CAN BUS สามารถทำงานร่วมกับอุปกรณ์ไฟฟ้าต่าง ๆ ในรถ ATV ได้อย่างเสถียร โดยระบบสามารถรับส่งข้อมูลระหว่างโมดูลได้อย่างรวดเร็วและแม่นยำ ซึ่งการสื่อสารผ่าน CAN BUS ช่วยให้การควบคุมอุปกรณ์ต่าง ๆ ภายในรถ ATV มีความสอดคล้องกัน เช่น ระบบไฟเลี้ยว คันเร่ง และเบรก เป็นต้น นอกจากนี้ การใช้จอ ATD3.5-S3 สำหรับแสดงผลข้อมูลแบบเรียลไทม์ถือเป็นอีกหนึ่งผลลัพธ์ที่สำคัญ เนื่องจากจอนี้ช่วยให้ผู้ใช้งานสามารถตรวจสอบสถานะการทำงานของระบบได้ตลอดเวลา ข้อมูลที่แสดงผลบนจอประกอบไปด้วยข้อมูลการทำงานของคันเร่ง เบรก โหมดการทำงาน และสัญญาณไฟต่าง ๆ ซึ่งช่วยให้ผู้ใช้งานสามารถตรวจสอบและประเมินความพร้อมในการทำงานของระบบได้อย่างทันทั่วทั้งนี้ การแสดงผลแบบเรียลไทม์ช่วยให้การวิเคราะห์ปัญหาและความผิดปกติของระบบเป็นไปได้อย่างแม่นยำและมีประสิทธิภาพ ซึ่งถือเป็นการเพิ่มความปลอดภัยและความน่าเชื่อถือให้กับระบบควบคุมที่พัฒนาขึ้น

การใช้โปรแกรม MRS Developers Studio ในการพัฒนาและควบคุมระบบ CAN BUS ช่วยให้ระบบตอบสนองคำสั่งได้รวดเร็วและมีประสิทธิภาพ โปรแกรมนี้ช่วยในการเขียนและปรับแต่งการทำงานของ CAN I/O PLC เพื่อควบคุมอุปกรณ์ไฟฟ้าต่าง ๆ ทั้งการส่งสัญญาณจากสวิตช์และการรับข้อมูลจากเซ็นเซอร์ ผลการทดลองแสดงให้เห็นว่า ระบบ CAN BUS ที่พัฒนาสามารถทำงานร่วมกับ CAN I/O PLC ได้ดี มีความเสถียรและความแม่นยำ การแสดงผลข้อมูลแบบเรียลไทม์ช่วยให้การควบคุมและตรวจสอบทำได้อย่างรวดเร็ว ซึ่งมีศักยภาพในการนำไปใช้กับยานพาหนะไฟฟ้าดัดแปลงในอนาคต

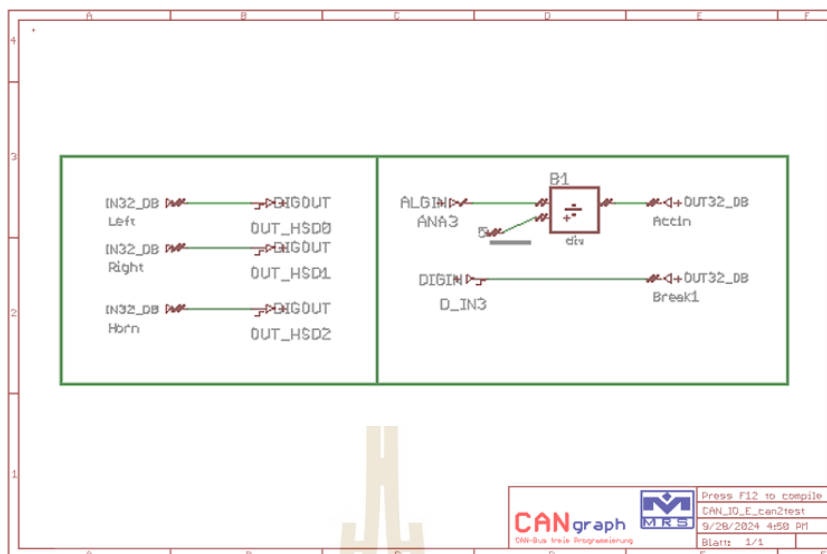
4.2 การดำเนินงานการเขียนโปรแกรมรถ ATV

ผลการดำเนินงานแสดงให้เห็นถึงความสำเร็จในการติดตั้งและการทำงานของระบบภายในรถ ATV อย่างมีประสิทธิภาพ ไม่ว่าจะเป็นระบบไฟหน้า ไฟเลี้ยว ไฟฉุกเฉิน ระบบเบรก และระบบคันเร่ง

ตารางที่ 4.1 CAN Database กลุ่มควบคุมที่ 1

Name	CAN ID [hex]	Extended	Send period max [ms]	Send period min [ms]	DLC	Send	CAN ID mask
SWG1	0x101	1	100	100	8	1	0
SWG2	0x102	1	100	100	8	1	0
CAN Data Points/Define CAN_DP							
Variable name	CAN block name	Bit start	Bit length	Data format			
Left	SWG1	0	1	0			
Right	SWG1	1	1	0			
Headlight	SWG1	2	1	0			
Horn	SWG1	3	1	0			
Engine	SWG2	0	1	0			
Start control	SWG2	1	1	0			
Gear1	SWG2	2	1	0			
Gear3	SWG2	3	1	0			
Forward	SWG2	4	1	0			
Backward	SWG2	5	1	0			

กล่อง CAN I/O ที่ 2 ทำหน้าที่รับสัญญาณไฟเลี้ยวและแสดงผลสัญญาณจากกล่อง CAN I/O ที่ 1 นอกจากนี้ กล่อง CAN I/O ที่ 2 ยังส่งสัญญาณคันเร่งไปแสดงผลที่กล่อง CAN I/O ที่ 3 และ CAN I/O กล่องที่ 2 ยังมีระบบเบรกประมวลผลภายในตัว ผลการเขียนโปรแกรมสำหรับกล่อง CAN BUS ที่ 2 แสดงในรูปที่ 4.2



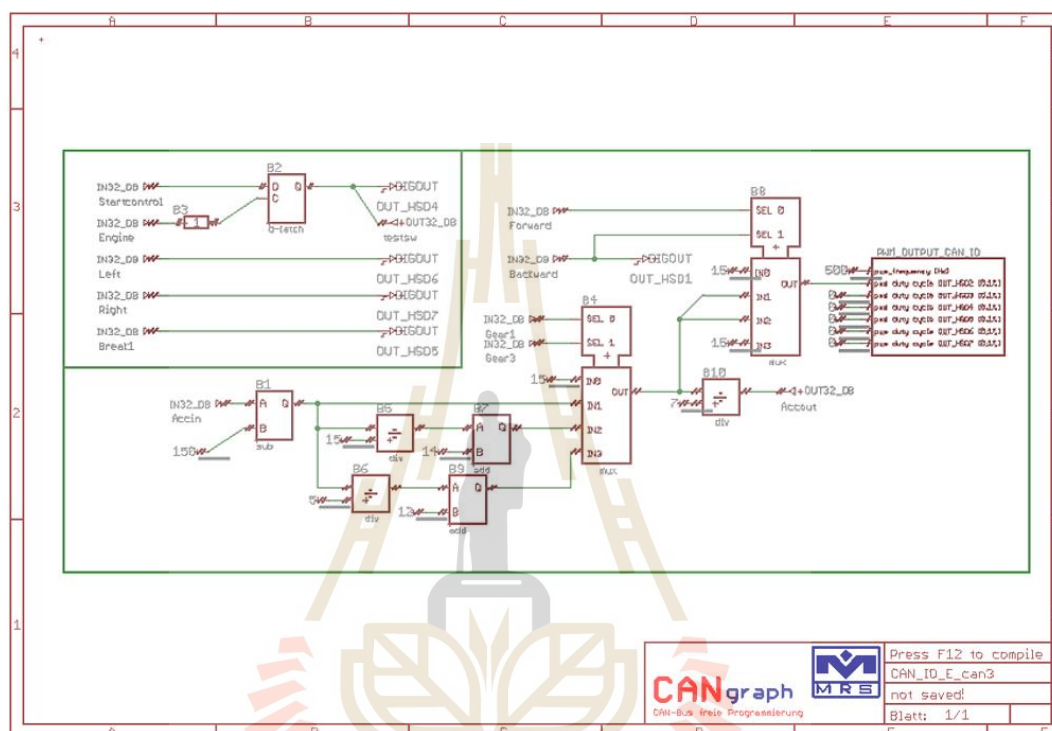
รูปที่ 4.2 CAN I/O กล่องที่ 2

สำหรับการตั้งค่าและการกำหนดพารามิเตอร์ในการเขียนโปรแกรมของตัวควบคุม CAN BUS (รูปที่ 4.2 กล่องควบคุม CAN I/O กล่องที่ 2) การตั้งค่าตัวควบคุม CAN Database สามารถกำหนดได้ตามตารางที่ 4.2 โดยพารามิเตอร์จะถูกกำหนดไว้ดังนี้

ตารางที่ 4.2 CAN Database กล่องควบคุมที่ 2

Name	CAN ID [hex]	Extended	Send period max [ms]	Send period min [ms]	DLC	Send	CAN ID mask
SWG1	0x101	1	0	0	8	1	0
ACC	0x102	1	100	100	8	1	0
Break	0x105	1	100	100	8	1	0
CAN Data Points/Define CAN_DP							
Variable name	CAN block name	Bit start	Bit length	Data format			
Left	SWG1	0	1	0			
Right	SWG1	1	1	0			
Accin	ACC	0	10	0			
Horn	SWG1	3	1	0			
Break1	Break	0	1	0			

กล่อง CAN I/O ที่ 3 มีหน้าที่รับและแสดงผลสัญญาณจากกล่อง CAN BUS ที่ 1 ที่เป็นระบบ ไฟเลี้ยงหลังรถ นอกจากนี้ กล่อง CAN I/O ที่ 3 ยังรับสัญญาณเกียร์เดินหน้าและถอยหลังจากกล่อง CAN I/O ที่ 1 และสัญญาณคันเร่งจากกล่อง CAN I/O ที่ 2 เพื่อนำมาแสดงผลการควบคุมระบบ โดยการเขียนโปรแกรมสำหรับการทำงานนี้แสดงในรูปที่ 4.3



รูปที่ 4.3 CAN I/O กล่องที่ 3

สำหรับการตั้งค่าและการกำหนดพารามิเตอร์ในการเขียนโปรแกรมของตัวควบคุม CAN BUS (รูปที่ 4.3 กล่องควบคุม CAN I/O กล่องที่ 3) การตั้งค่าตัวควบคุม CAN Database สามารถกำหนดได้ตามตารางที่ 4.3 โดยพารามิเตอร์จะถูกกำหนดไว้ดังนี้

ตารางที่ 4.3 CAN Database กลุ่มควบคุมที่ 3

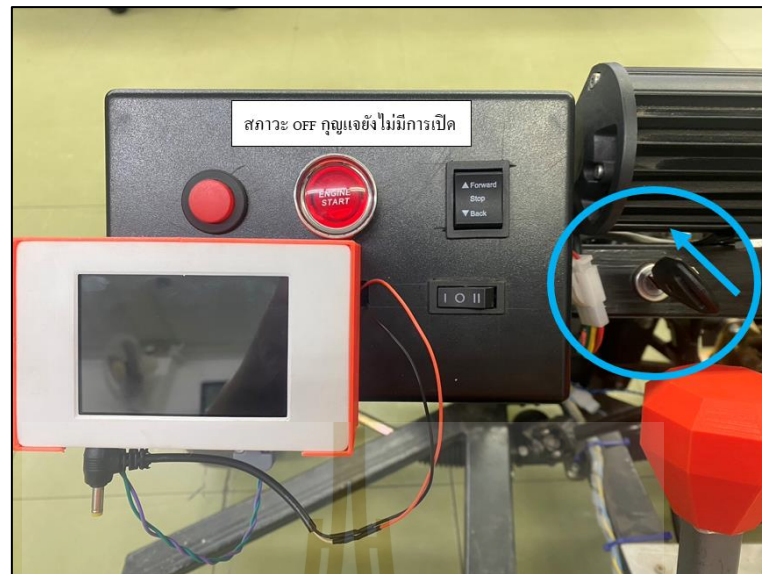
Name	CAN ID [hex]	Extended	Send period max [ms]	Send period min [ms]	DLC	Send	CAN ID mask
SWG1	0x101	1	0	0	8	1	0
ACC	0x103	1	0	0	8	1	0
acc2	0x104	1	100	100	8	1	0
SWG2	0x102	1	0	0	8	1	0
Break	0x105	1	0	0	8	1	0
CAN Data Points/Define CAN_DP							
Variable name	CAN block name	Bit start	Bit length	Data format			
Left	SWG1	0	1	0			
Right	SWG1	1	1	0			
Engine	SWG2	0	1	0			
Accin	ACC	0	10	0			
Startcontrol	SWG2	1	1	0			
Forward	SWG2	4	1	0			
Backward	SWG2	5	1	0			
Account	acc2	0	8	0			
Gear1	SWG2	2	1	0			
Gear3	SWG2	3	1	0			
testsw	acc2	8	1	0			
Break1	Break	0	1	0			

4.3 การเชื่อมต่อระบบ CAN BUS ระหว่างโมดูลต่าง ๆ สำหรับการแสดงผล

ผลลัพธ์จากการวิจัยนี้เป็นการทดสอบและพัฒนาระบบ CAN BUS ที่ใช้ควบคุมการทำงานของรถ ATV ซึ่งมีการเชื่อมต่อและสื่อสารระหว่างโมดูล CAN I/O PLC และจอแสดงผล ATD3.5-S3 โดยผลการทดสอบแบ่งออกเป็น 3 ส่วนหลัก ดังนี้

4.3.1 การแสดงผลของ CAN I/O กลุ่มที่ 1 และหน้าจอแสดงผล

- 1) สภาวะปิดกุญแจอยู่ในสถานะ OFF หน้าจอแสดงผลจะไม่มีการแสดงสถานะพิจารณาได้จากรูปที่ 4.4



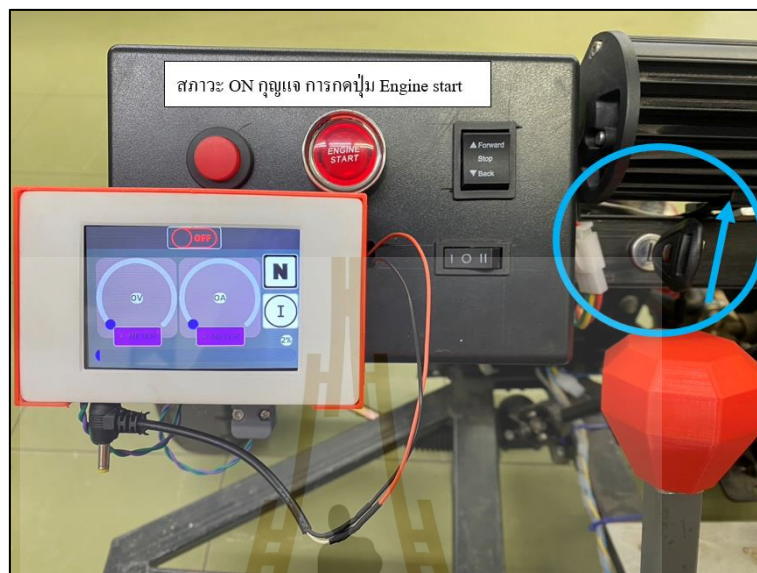
รูปที่ 4.4 สภาวะปิดทุจยงัม่ม่การเปิดในสถานะ OFF

- 2) สภาวะเปิดทุจยงัม่ม่การกดปุ่ม Engine start
จอแสดงผลจะมีการแสดงสถานะ พิจารณาได้จากรูปที่ 4.5



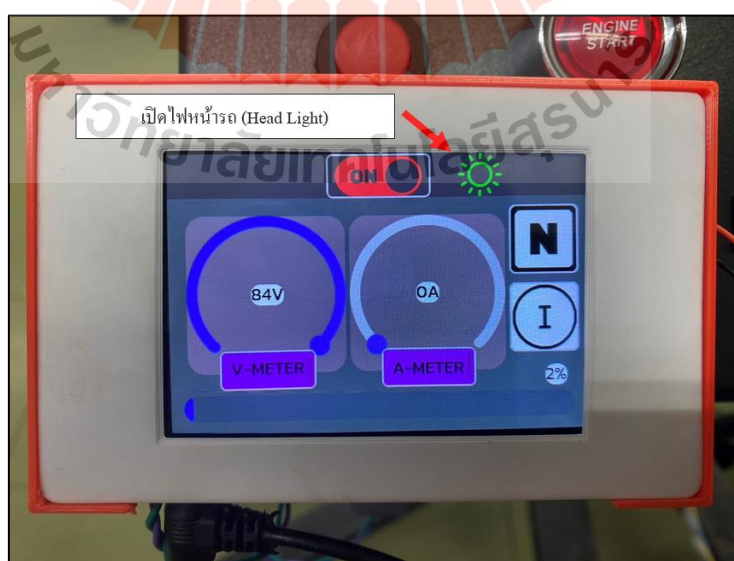
รูปที่ 4.5 สภาวะเปิดทุจยงัม่ม่การกดปุ่ม Engine start

- 3) สภาวะเปิดกุญแจอยู่ในสถานะ ON มีการกดปุ่ม Engine start จอแสดงผลจะมีการแสดงสถานะ พิจารณาได้จากรูปที่ 4.6



รูปที่ 4.6 สภาวะเปิดกุญแจอยู่ในสถานะ ON มีการกดปุ่ม Engine start

- 4) สภาวะเปิดไฟหน้ารถ และแบตเตอรี่เต็ม 84 V จอแสดงผลจะมีการแสดงสถานะ พิจารณาได้จากรูปที่ 4.7



รูปที่ 4.7 สภาวะเปิดไฟหน้ารถ

- 5) สภาวะเปิดไฟเลี้ยวซ้าย (หน้ารถและหลังรถ ทำงานพร้อมกัน) และแบตเตอรี่เต็ม 84 V จอแสดงผลจะมีการแสดงสถานะ พิจารณาได้จากรูปที่ 4.8



รูปที่ 4.8 สภาวะเปิดไฟเลี้ยวซ้าย (หน้ารถและหลังรถ ทำงานพร้อมกัน)

- 6) สภาวะเปิดไฟเลี้ยวขวา (หน้ารถและหลังรถ ทำงานพร้อมกัน) และแบตเตอรี่เต็ม 84 V จอแสดงผลจะมีการแสดงสถานะ พิจารณาได้จากรูปที่ 4.9



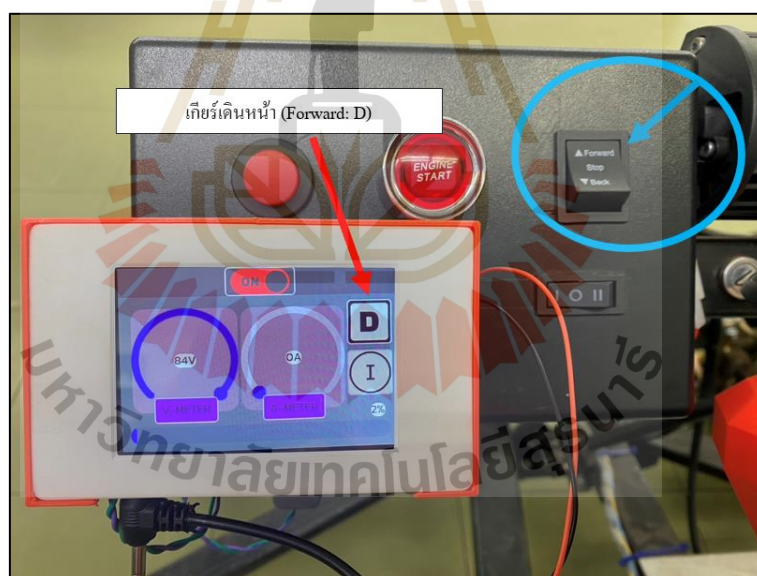
รูปที่ 4.9 สภาวะเปิดไฟเลี้ยวขวา (หน้ารถและหลังรถ ทำงานพร้อมกัน)

- 7) สภาวะเปิดไฟฉุกเฉิน (หน้ารถและหลังรถ ทำงานพร้อมกัน) และแบตเตอรี่เต็ม 84 V จอแสดงผลจะมีการแสดงสถานะ พิจารณาได้จากรูปที่ 4.10



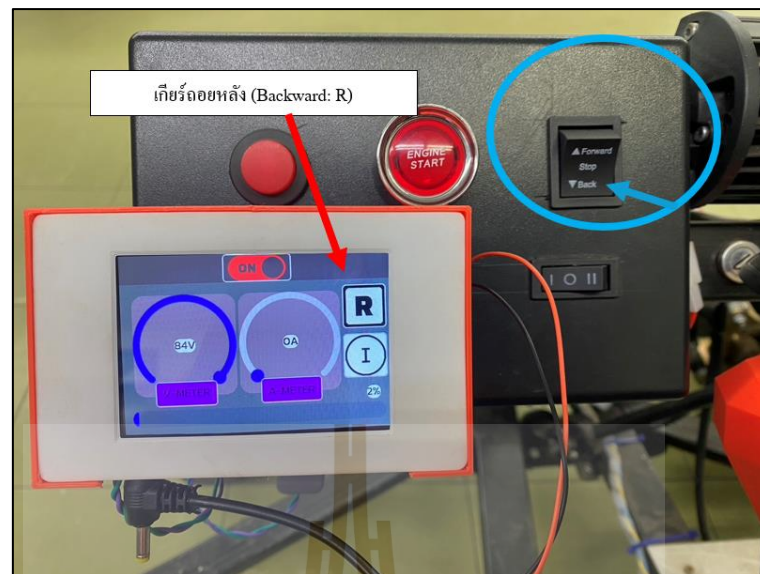
รูปที่ 4.10 สภาวะเปิดไฟฉุกเฉิน (หน้ารถและหลังรถ ทำงานพร้อมกัน)

- 8) สภาวะกดเกียร์เดินหน้า (Forward: D) และแบตเตอรี่เต็ม 84 V จอแสดงผลจะมีการแสดงสถานะ พิจารณาได้จากรูปที่ 4.11



รูปที่ 4.11 สภาวะกดเกียร์เดินหน้า (Forward: D)

- 9) สภาวะกดเกียร์ถอยหลัง (Backward: R) และแบตเตอรี่เต็ม 84 V จอแสดงผลจะมีการแสดงสถานะ พิจารณาได้จากรูปที่ 4.12



รูปที่ 4.12 สภาวะกดยกเกียร์ถอยหลัง (Backward: R)

- 10) สภาวะกดยกเกียร์ว่างในโหมด Eco โดยสภาวะคันเร่งไฟฟ้าอยู่ที่ 0-8% และ แบตเตอรี่เต็ม 84 V จอแสดงผลจะมีการแสดงสถานะ พิจารณาได้จากรูปที่ 4.13



รูปที่ 4.13 สภาวะกดยกเกียร์ว่างในโหมด Eco โดยสภาวะคันเร่งไฟฟ้าอยู่ที่ 0-8%

- 11) สภาวะกดเกียร์ว่างในโหมด Normal โดยสภาวะคันเร่งไฟฟ้าอยู่ที่ 0-20% และ แบตเตอรี่เต็ม 84 V จอแสดงผลจะมีการแสดงสถานะ พิจารณาได้จากรูปที่ 4.14



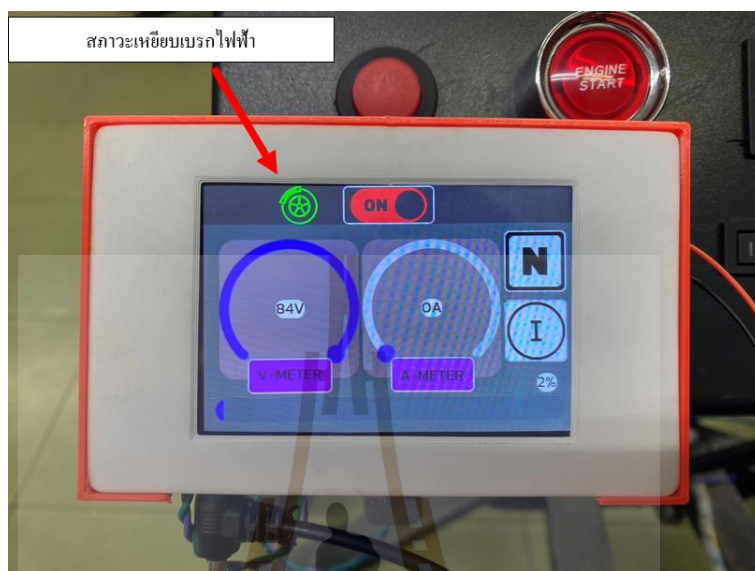
รูปที่ 4.14 สภาวะกดเกียร์ว่างในโหมด Normal โดยสภาวะคันเร่งไฟฟ้าอยู่ที่ 0-20%

- 12) สภาวะกดเกียร์ว่างในโหมด Sport โดยสภาวะคันเร่งไฟฟ้าอยู่ที่ 0-100% และ แบตเตอรี่เต็ม 84 V จอแสดงผลจะมีการแสดงสถานะ พิจารณาได้จากรูปที่ 4.15



รูปที่ 4.15 สภาวะกดเกียร์ว่างในโหมด Sport โดยสภาวะคันเร่งไฟฟ้าอยู่ที่ 0-100%

- 13) สภาวะเหยียบเบรกไฟฟ้า และแบตเตอรี่เต็ม 84 V จอแสดงผลจะมีการแสดงสถานะ พิจารณาได้จากรูปที่ 4.16



รูปที่ 4.16 สภาวะเหยียบเบรกไฟฟ้า

จากการทดสอบระบบ CAN BUS ที่เชื่อมต่อระหว่างโมดูลทั้ง 3 (โมดูลที่ 1, โมดูลที่ 2, และโมดูลที่ 3) พบว่า การส่งสัญญาณระหว่างโมดูลทำงานได้ดี โดยโมดูลที่ 1 ซึ่งทำหน้าที่รับสัญญาณจากสวิตช์ควบคุมต่าง ๆ เช่น สวิตช์กุญแจ, สวิตช์สตาร์ท, และสวิตช์ไฟเลี้ยว สามารถส่งข้อมูลผ่านระบบ CAN BUS ไปยังโมดูลที่ 2 และ 3 ได้อย่างถูกต้อง และโมดูลที่ 2 ส่งสัญญาณคันเร่งและเบรกไปหาโมดูลที่ 3 ได้อย่างถูกต้องเช่นเดียวกัน ผลการทดสอบแสดงให้เห็นว่าการสั่งงานของคันเร่ง เบรก และการควบคุมไฟเลี้ยว สามารถทำงานร่วมกันได้โดยไม่มี ความล่าช้า ซึ่งบ่งชี้ว่าระบบการสื่อสาร CAN BUS ในรถ ATV ที่พัฒนาขึ้นมีความเสถียรและสามารถใช้งานได้ดีและในการควบคุมการทำงานของรถ ระบบนี้จึงมีความเหมาะสมสำหรับการนำไปใช้งานในสภาพแวดล้อมที่ต้องการการสื่อสารระหว่างอุปกรณ์หลาย ๆ ตัวภายในยานยนต์



รูปที่ 4.17 โมดูล CAN I/O PLC ที่ 1 และหน้าจอแสดงผล



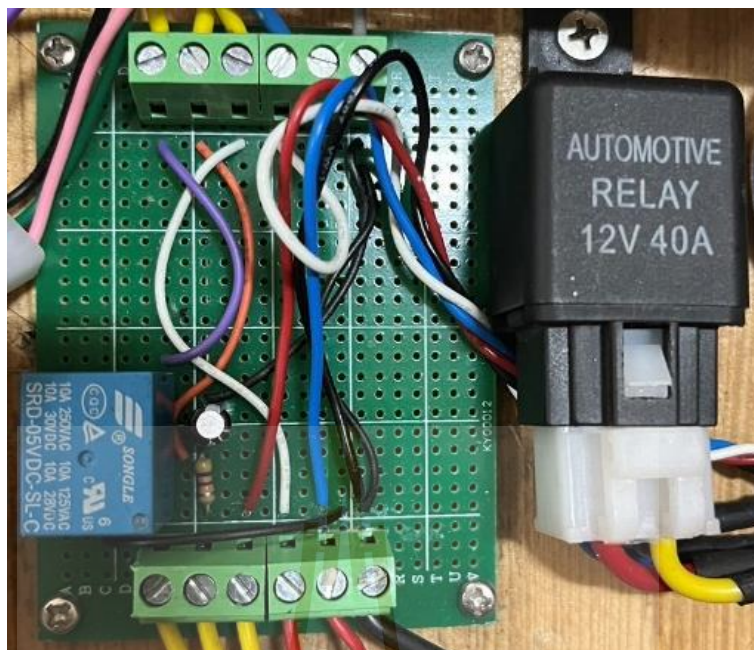
รูปที่ 4.18 โมดูล CAN I/O PLC ที่ 2 และหลอดไฟเลี้ยวด้านหน้า



รูปที่ 4.19 โมดูล CAN I/O PLC ที่ 3 และหลอดไฟเลี้ยวด้านหลัง

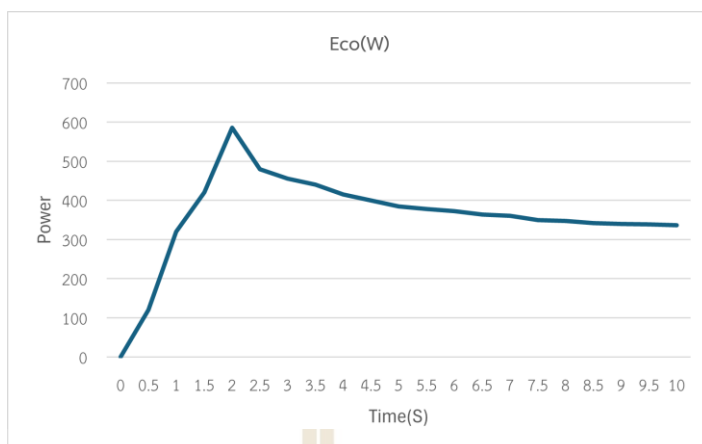
4.4 ประสิทธิภาพการควบคุมมอเตอร์ไฟฟ้า

ในการทดลองส่งสัญญาณจากโมดูลที่ 3 ไปยังกล่องควบคุมมอเตอร์ไฟฟ้า พบว่าระบบไม่สามารถปรับการทำงานของมอเตอร์ได้โดยตรง เนื่องจากสัญญาณที่ส่งออกมาจากโมดูล CAN I/O PLC เป็นสัญญาณ Pulse Width Modulation (PWM) ซึ่งไม่สามารถสื่อสารกับกล่องคอนโทรลเลอร์มอเตอร์โดยตรง ดังนั้น จึงจำเป็นต้องใช้วงจรแปลงสัญญาณ Low Pass Filter (รูปที่ 4.20) เพื่อแปลงสัญญาณ PWM ให้เป็นสัญญาณอนาล็อกที่สามารถควบคุมกล่องคอนโทรลเลอร์มอเตอร์ได้ หลังจากทำการแปลงสัญญาณแล้ว ระบบสามารถควบคุมคันเร่งและเบรกได้อย่างแม่นยำ และเมื่อทดสอบการทำงานในโหมดต่าง ๆ (Eco, Normal, Sport) พบว่าโหมดการทำงานถูกปรับตามที่กำหนดและระบบตอบสนองต่อคำสั่งได้อย่างรวดเร็ว การใช้วงจรแปลงสัญญาณจึงช่วยให้ระบบควบคุมมอเตอร์ไฟฟ้าทำงานได้ตามความต้องการและมีประสิทธิภาพ

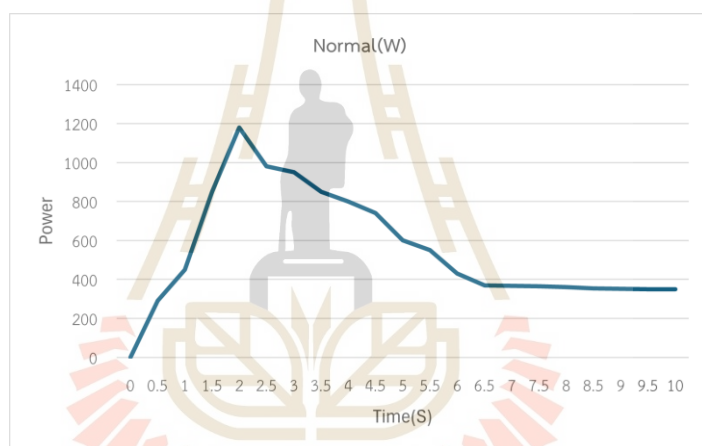


รูปที่ 4.20 วงจรแปลงสัญญาณคันเร่งและสั่งงานควบคุมส่วนต่าง ๆ ภายในรถ

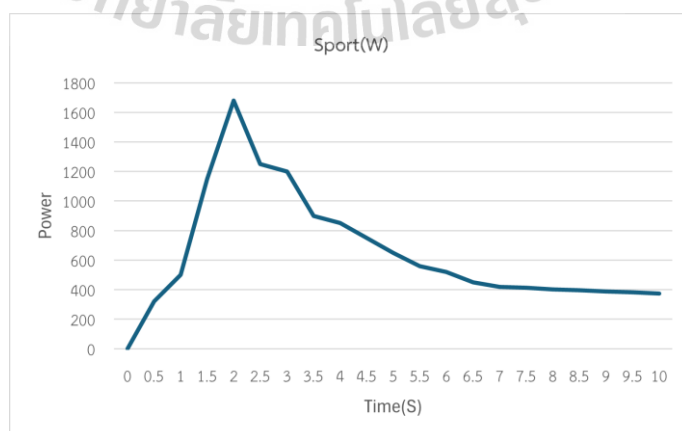
ในการทดสอบกำลังไฟฟ้าที่ใช้งานในโหมดต่าง ๆ ของรถ ATV ซึ่งประกอบด้วยโหมด Eco, Normal และ Sport พบว่าคุณสมบัติของแต่ละโหมดมีความแตกต่างกันอย่างชัดเจน โดยในโหมด Eco รถสามารถทำความเร็วสูงสุดได้ 17 กิโลเมตรต่อชั่วโมง และใช้กำลังไฟฟ้าสูงสุดที่ประมาณ 580 วัตต์ ซึ่งแสดงผลการทดสอบในรูปที่ 4.21 สำหรับโหมด Normal พบว่ารถสามารถทำความเร็วสูงสุดได้ 24 กิโลเมตรต่อชั่วโมง และใช้กำลังไฟฟ้าสูงสุดที่ประมาณ 1,100 วัตต์ ซึ่งแสดงผลการทดสอบในรูปที่ 4.22 ส่วนโหมด Sport รถยังคงมีความเร็วสูงสุดที่ 24 กิโลเมตรต่อชั่วโมง เช่นเดียวกับโหมด Normal แต่มีการใช้กำลังไฟฟ้าสูงสุดที่ประมาณ 1,600 วัตต์ ซึ่งแสดงผลการทดสอบในรูปที่ 4.23 โดยในโหมด Sport นี้ รถจะสามารถเร่งไปถึงความเร็วสูงสุดได้ในระยะเวลาที่สั้นกว่าโหมด Normal



รูปที่ 4.21 กราฟแสดงกำลังไฟฟ้าเทียบกับเวลาในโหมด Eco



รูปที่ 4.22 กราฟแสดงกำลังไฟฟ้าเทียบกับเวลาในโหมด Normal



รูปที่ 4.23 กราฟแสดงกำลังไฟฟ้าเทียบกับเวลาในโหมด Sport

4.5 การวิเคราะห์และอภิปรายผลการดำเนินงานวิจัย

จากผลการศึกษาและการทดสอบที่ได้จากการวิจัย พบว่าระบบ CAN BUS ที่ถูกพัฒนาเพื่อควบคุมการทำงานของรถ ATV มีความเสถียรและสามารถสื่อสารระหว่างโมดูลต่าง ๆ ได้อย่างมีประสิทธิภาพ การเชื่อมต่อระหว่าง CAN I/O PLC ทั้งสามโมดูลทำงานได้อย่างราบรื่น สัญญาณจากสวิตช์ควบคุมสามารถถูกส่งผ่าน CAN BUS ไปยังโมดูลต่าง ๆ ได้อย่างถูกต้องและทันท่วงที โดยเฉพาะในส่วนของการควบคุมคันเร่ง เบรก และไฟเลี้ยว ซึ่งไม่มีความล่าช้าหรือการขาดหายของสัญญาณ ส่งผลให้การทำงานของระบบมีความแม่นยำสูง

ในส่วนของการควบคุมมอเตอร์ไฟฟ้า แม้ว่าการสื่อสารระหว่าง CAN I/O PLC และกล่องควบคุมมอเตอร์จะไม่สามารถทำงานได้โดยตรง เนื่องจากสัญญาณ PWM ไม่สามารถสื่อสารกับกล่องควบคุมมอเตอร์ได้ จึงต้องใช้วงจรแปลงสัญญาณ Low Pass Filter เพื่อแปลงสัญญาณ PWM เป็นสัญญาณอนาล็อก การแปลงสัญญาณนี้ช่วยให้ระบบสามารถควบคุมคันเร่งและเบรกได้อย่างแม่นยำมากขึ้น ซึ่งการตอบสนองในโหมดการทำงานที่แตกต่างกัน (Eco, Normal, Sport) แสดงให้เห็นว่าระบบมีการตอบสนองที่รวดเร็วและแม่นยำ โดยสามารถปรับโหมดการทำงานตามที่กำหนดได้อย่างมีประสิทธิภาพ

สำหรับการแสดงผลข้อมูลแบบเรียลไทม์บนจอ ATD3.5-S3 พบว่าการแสดงสถานะการทำงานของรถ ATV มีความเสถียรและสามารถตรวจสอบได้อย่างถูกต้อง ข้อมูลที่แสดงผลแบบเรียลไทม์ช่วยให้ผู้ใช้งานสามารถติดตามการทำงานของระบบ CAN BUS ได้อย่างรวดเร็ว และสามารถวิเคราะห์หาความผิดปกติที่อาจเกิดขึ้นได้อย่างมีประสิทธิภาพ การแสดงผลนี้ยังช่วยเพิ่มความรวดเร็วและความแม่นยำในการควบคุมและปรับปรุงการทำงานของรถ ทำให้การตรวจสอบสถานะต่าง ๆ เป็นไปได้อย่างราบรื่นและสอดคล้องกับการทำงานของระบบโดยรวม

จากการวิเคราะห์ผลที่ได้ แสดงให้เห็นว่าการประยุกต์ใช้เทคโนโลยี CAN BUS, CAN I/O PLC, และการแสดงผลผ่านจอ ATD3.5-S3 นั้นสามารถสนับสนุนการควบคุมการทำงานของรถ ATV ได้อย่างมีประสิทธิภาพ โดยช่วยให้ระบบมีความเสถียรในการส่งข้อมูล ความแม่นยำในการควบคุม และความสามารถในการตรวจสอบและวิเคราะห์ข้อมูลได้แบบเรียลไทม์

สำหรับการควบคุมมอเตอร์ไฟฟ้าในโหมดการทำงานต่าง ๆ พบว่าระบบสามารถตอบสนองต่อคำสั่งได้อย่างรวดเร็ว และมีประสิทธิภาพในการควบคุม ทั้งนี้จากการทดลองพบว่าการเลือกโหมดการทำงานต่าง ๆ มีผลต่อการใช้พลังงานและความเร็วของรถ ATV ซึ่งสอดคล้องกับวัตถุประสงค์ของการพัฒนา

บทที่ 5

สรุปและข้อเสนอแนะ

5.1 บทนำ

ในการวิจัยนี้ได้ดำเนินการศึกษาการใช้ระบบ CAN BUS ร่วมกับการควบคุมโมดูลต่าง ๆ ในรถ ATV โดยใช้ CAN I/O PLC และจอ ATD3.5-S3 เพื่อตรวจสอบการทำงานของระบบ รวมถึงการประยุกต์ใช้ซอฟต์แวร์ MRS Developers Studio เพื่อพัฒนาและควบคุมการทำงานของระบบ ซึ่งผลการศึกษาและการทดลองได้รับการวิเคราะห์อย่างละเอียดในบทก่อนหน้านี้

5.2 สรุปผลการศึกษา

จากการทดสอบระบบ CAN BUS ที่เชื่อมต่อระหว่างโมดูลทั้ง 3 (โมดูลที่ 1, โมดูลที่ 2, และโมดูลที่ 3) พบว่าระบบสามารถทำงานได้อย่างเสถียรและมีประสิทธิภาพ โมดูลที่ 1 สามารถส่งสัญญาณจากสวิตช์ควบคุมไปยังโมดูลที่ 2 และ 3 ได้อย่างแม่นยำ โดยโมดูลที่ 2 ทำหน้าที่ควบคุมการทำงานของไฟเลี้ยวและเบรก และส่งสัญญาณจากเซนเซอร์คันเร่งและเบรกไปยังโมดูลที่ 3 สำหรับควบคุมมอเตอร์ไฟฟ้า ซึ่งหลังจากปรับสัญญาณ PWM ด้วยวงจร Low Pass Filter แล้ว ระบบสามารถควบคุมการทำงานของมอเตอร์ไฟฟ้าได้อย่างมีประสิทธิภาพตามโหมดต่าง ๆ ได้แก่ Eco, Normal, และ Sport นอกจากนี้ การแสดงผลบนจอ ATD3.5-S3 ที่เชื่อมต่อกับระบบ CAN BUS สามารถแสดงข้อมูลแบบเรียลไทม์ได้อย่างถูกต้องและทันท่วงที ข้อมูลที่แสดงผล ได้แก่ สถานะการทำงานของคันเร่ง, เบรก, ไฟเลี้ยว, และโหมดการทำงานของรถ ATV (Eco, Normal, Sport) การแสดงผลมีความเสถียรและตอบสนองอย่างรวดเร็ว ทำให้สามารถตรวจสอบสถานะการทำงานของระบบได้แบบเรียลไทม์ เพิ่มความสะดวกในการตรวจสอบสถานะและการทำงานของระบบ ทำให้การควบคุมรถ ATV ทำได้อย่างแม่นยำและปลอดภัย

ในส่วนของการวัดพลังงานที่ใช้ในแต่ละโหมด พบว่าระบบสามารถบันทึกค่าพลังงานที่ใช้ได้อย่างแม่นยำ จากการทดสอบดังกล่าว พบว่าโหมด Eco มีประสิทธิภาพในการประหยัดพลังงานสูงสุด แต่ความเร็วของรถต่ำที่สุด ในขณะที่โหมด Sport มีการใช้พลังงานสูงสุดแต่สามารถเร่งความเร็วได้เร็วกว่าโหมดอื่น ๆ ข้อมูลเหล่านี้สามารถนำไปใช้ในการวิเคราะห์และปรับปรุงระบบให้เหมาะสมกับการใช้งานในสภาพแวดล้อมต่าง ๆ ได้ต่อไป

5.3 ข้อเสนอแนะ

1) พัฒนาระบบควบคุมโดยการแยกโมดูลให้มากกว่านี้ เพราะจะได้แยกเป็นสัดส่วนและแยกระบบการทำงานอย่างชัดเจน ลดความซับซ้อนในการควบคุมและบำรุงรักษา ทำให้ระบบสามารถตรวจสอบและแก้ไขปัญหาได้ง่ายขึ้นในกรณีที่เกิดข้อผิดพลาด

2) พัฒนาวงจรการควบคุมให้เปลี่ยนมาใช้โมดูลที่สามารถรองรับการส่งสัญญาณอนาล็อกได้โดยตรง เพื่อลดข้อผิดพลาดที่อาจเกิดขึ้นจากการแปลงสัญญาณ เช่น การใช้วงจร Low Pass Filter เพื่อแปลงสัญญาณ PWM ให้เป็นอนาล็อก

3) เพิ่มการแสดงผลสถานะต่าง ๆ บนจอแสดงผล ATD3.5-S3 เพื่อให้สามารถแสดงข้อมูลการทำงานเพิ่มเติมของระบบ CAN BUS เช่น อุณหภูมิของมอเตอร์, สถานะการใช้พลังงาน, การตรวจจับและแจ้งเตือนข้อผิดพลาดในระบบแบบเรียลไทม์ ซึ่งจะช่วยเพิ่มความสามารถในการตรวจสอบและติดตามการทำงานของรถ ATV ได้อย่างมีประสิทธิภาพมากยิ่งขึ้น

4) เปลี่ยนมาใช้โมดูล CAN BUS ที่มีราคาถูกกว่า CAN I/O PLC เพื่อลดต้นทุนในการพัฒนาระบบ แต่ยังคงประสิทธิภาพในการสื่อสารข้อมูลระหว่างโมดูลต่าง ๆ ของระบบ



รายการอ้างอิง

- Donjaroennon, N.; Nambunlue, W.; Nuchkum, S.; Leeton, U. Development of an Improved Communication Control System for ATV Electric Vehicles Using MRS Developers Studio. *World Electr. Veh. J.* 2024, 15, 303. <https://doi.org/10.3390/wevj15070303>
- Nambunlue, W., Donjaroennon, N., Leeton, U., Nuchkum, S., Singhand, K., Promlungd, N., & Srisertpol, J. (2024). Internet of Things (IoT) With the Power Meter Using V-BOX. *ISCEAS-0055(1)*, 183-192.
- S. Rattanachan, S. Nuchkum and U. Leeton, "Development of Electric Door Control System using MRS Developers Studio for Low-floor Bus," 2021 9th International Electrical Engineering Congress (iEECON), Pattaya, Thailand, 2021, pp. 285 - 288, doi: 10.1109/IEECON51072.2021.9440298.
- Adly, S.; Moro, A.; Hammad, S.; Maged, S.A. Prevention of Controller Area Network (CAN) Attacks on Electric Autonomous Vehicles. *Appl. Sci.* 2023, 13, 9374. <https://doi.org/10.3390/app13169374>
- Pham, N.N.; Leuchter, J.; Pham, K.L.; Dong, Q.H. Battery Management System for Unmanned Electric Vehicles with CAN BUS and Internet of Things. *Vehicles* 2022, 4, 639 - 662. <https://doi.org/10.3390/vehicles4030037>
- Alzahrani, A.; Wangikar, S.M.; Indragandhi, V.; Singh, R.R.; Subramaniaswamy, V. Design and Implementation of SAE J1939 and Modbus Communication Protocols for Electric Vehicle. *Machines* 2023, 11, 201. <https://doi.org/10.3390/machines11020201>
- J1939/16_201811; SAE International Recommended Practice, Automatic Baud Rate Detection Process. SAE Standard: Warrendale, PA, USA, 2018.
- J1939DA_202406; SAE International Supplement, J1939 Digital Annex. SAE Standard: Warrendale, PA, USA, 2024.
- Osman, R.A.; Saleh, S.N.; Saleh, Y.N.M.; Elagamy, M.N. Enhancing the Reliability of Communication between Vehicle and Everything (V2X) Based on Deep Learning for Providing Efficient Road Traffic Information. *Appl. Sci.* 2021, 11, 11382. <https://doi.org/10.3390/app112311382>

Yin, L.; Xu, J.; Wang, Z.; Wang, C. A Provable Secure Session Key Distribution Protocol Based on NSSK for In-Vehicle CAN Network. *Mathematics* 2022, 10, 2903. <https://doi.org/10.3390/math10162903>

Babangida, A.; Light Odazie, C.M.; Szemes, P.T. Optimal Control Design and Online Controller-Area-Network Bus Data Analysis for a Light Commercial Hybrid Electric Vehicle. *Mathematics* 2023, 11, 3436. <https://doi.org/10.3390/math11153436>





ภาคผนวก ก

ขั้นตอนการดำเนินงาน

ชุดทดลองยานยนต์ไฟฟ้า



รูปที่ 6.1 ด้านหน้ารถ ATV



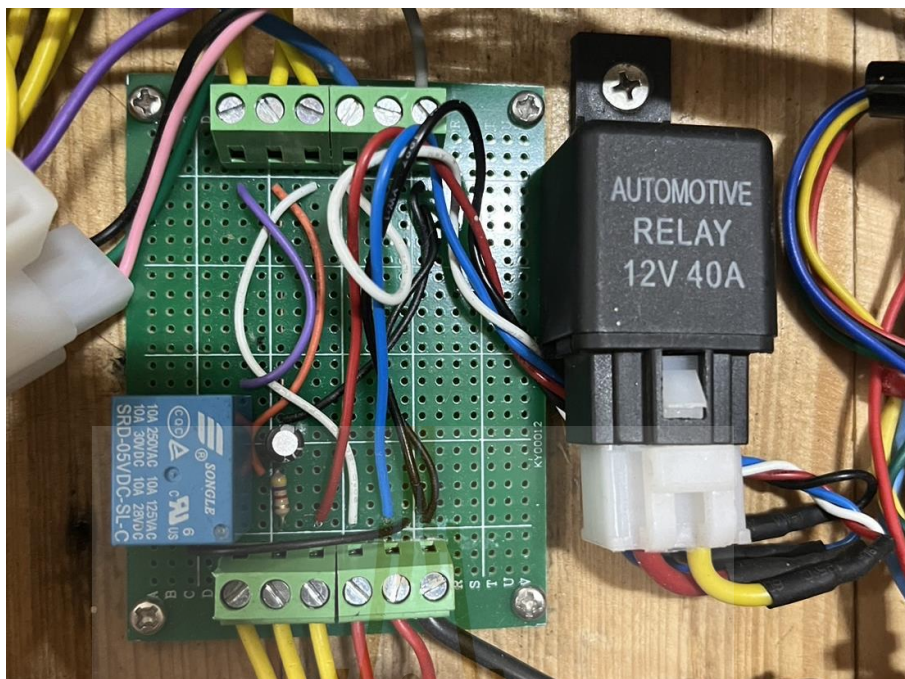
รูปที่ 6.2 ด้านท้ายรถ ATV



รูปที่ 6.3 ชุดควบคุมด้านหน้ารถ ATV



รูปที่ 6.4 หน้าจอแสดงสถานะรถ ATV



รูปที่ 6.5 ชุดแปลงสัญญาณ Low Pass Filter



รูปที่ 6.6 กล่องควบคุมมอเตอร์

CAN_DB-Edit

CAN_Telegram / Define CAN_BLOCK

Name	CAN ID [hex]	Extended	Send period max [ms]	Send period min [ms]	DLC	Send	CAN ID mask [hex]
▶ SWG1	0x101	1	100	100	8	1	0
SWG2	0x102	1	100	100	8	1	0
*							

CAN Data Points / Define CAN_DP

Variable name	CAN block name	Bit start	Bit length	Data format
▶ Left	SWG1	0	1	0
Right	SWG1	1	1	0
Headlight	SWG1	2	1	0
Horn	SWG1	3	1	0
Engine	SWG2	0	1	0
Startcontrol	SWG2	1	1	0
Gear1	SWG2	2	1	0
Gear3	SWG2	3	1	0
Forward	SWG2	4	1	0
Backward	SWG2	5	1	0
*				

Move your mouse over the columns to get more information. If you proceed an input, you get the available input possibilities displayed above the head line.

View
☒ Standard view ☐ Data point definition for moduletest

Close Import sym-file Export of sym-file

รูปที่ 6.7 CAN_DB กล้อง CANBUS ที่ 1

CAN_DB-Edit

CAN_Telegram / Define CAN_BLOCK

Name	CAN ID [hex]	Extended	Send period max [ms]	Send period min [ms]	DLC	Send	CAN ID mask [hex]
▶ SWG1	0x101	1	0	0	8	0	0
ACC	0x103	1	100	100	8	1	0
Break	0x105	1	100	100	8	1	0
*							

CAN Data Points / Define CAN_DP

Variable name	CAN block name	Bit start	Bit length	Data format
▶ Left	SWG1	0	1	0
Right	SWG1	1	1	0
Accin	ACC	0	10	0
Horn	SWG1	3	1	0
Break1	Break	0	1	0
*				

Move your mouse over the columns to get more information. If you proceed an input, you get the available input possibilities displayed above the head line.

View
☒ Standard view ☐ Data point definition for moduletest

Close Import sym-file Export of sym-file

รูปที่ 6.8 CAN_DB กล้อง CANBUS ที่ 2

CAN_DB-Edit

CAN_Telegram / Define CAN_BLOCK

Name	CAN ID [hex]	Extended	Send period max [ms]	Send period min [ms]	DLC	Send	CAN ID mask [hex]
SWG1	0x101	1	0	0	8	0	0
ACC	0x103	1	0	0	8	0	0
acc2	0x104	1	100	100	8	1	0
SWG2	0x102	1	0	0	8	0	0
Break	0x105	1	0	0	8	0	0
*							

CAN Data Points / Define CAN_DP

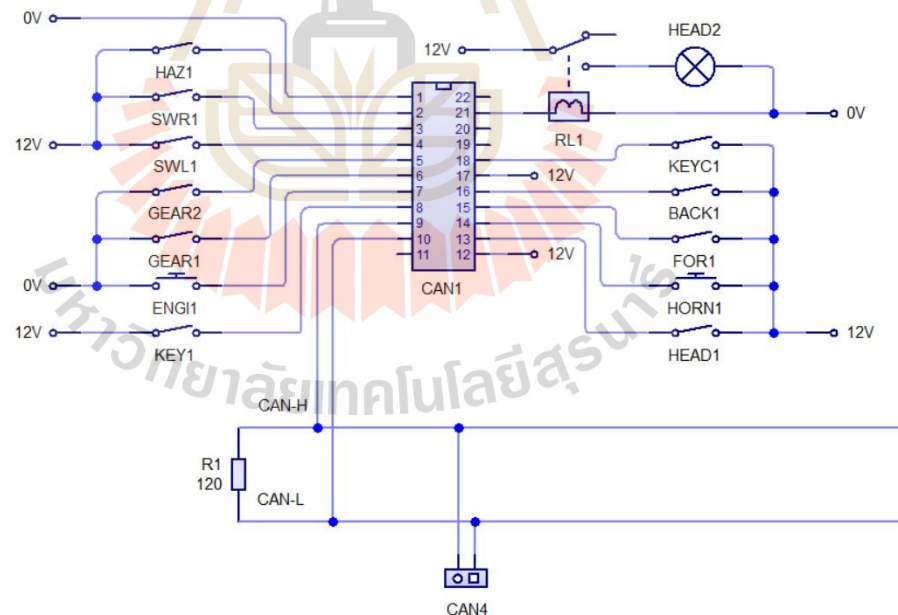
Variable name	CAN block name	Bit start	Bit length	Data format
Left	SWG1	0	1	0
Right	SWG1	1	1	0
Engine	SWG2	0	1	0
Accin	ACC	0	10	0
Startcontrol	SWG2	1	1	0
Forward	SWG2	4	1	0
Backward	SWG2	5	1	0
Accout	acc2	0	8	0
Gear1	SWG2	2	1	0
Gear3	SWG2	3	1	0
testsw	acc2	8	1	0
Break1	Break	0	1	0
*				

View
☒ Standard view ☐ Data point definition for moduletest

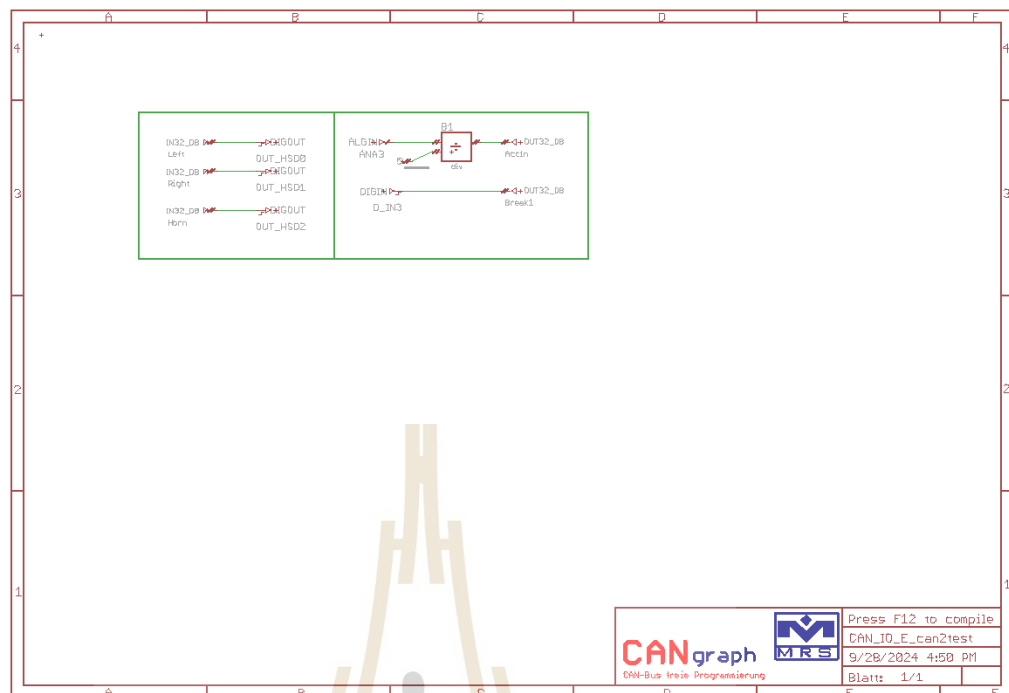
Close Import sym-file Export of sym-file

Move your mouse over the columns to get more information.
 If you proceed an input, you get the available input possibilities displayed above the head line.

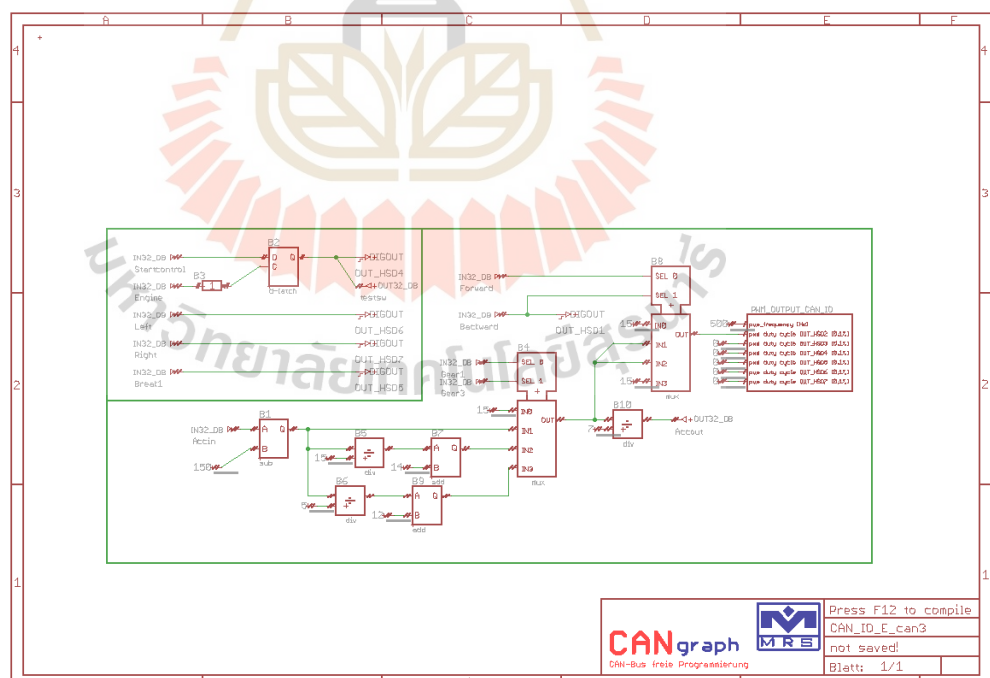
รูปที่ 6.9 CAN_DB กล้อง CANBUS ที่ 3



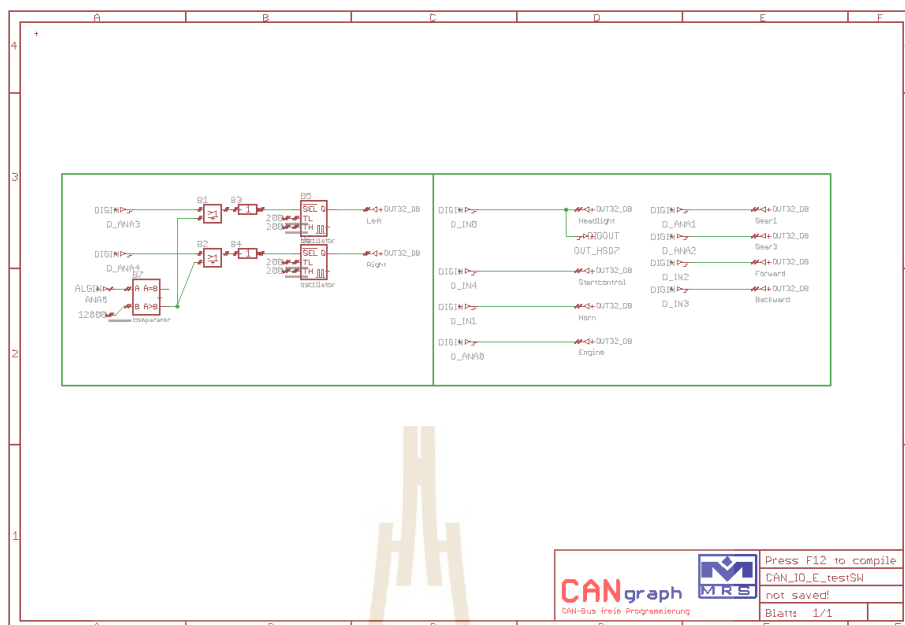
รูปที่ 6.10 วงจรไฟฟ้า กล้อง CANBUS ที่ 1



รูปที่ 6.13 โปรแกรมกล่อง CANBUS ที่ 1



รูปที่ 6.14 โปรแกรมกล่อง CANBUS ที่ 2



รูปที่ 6.15 โปรแกรมกล่อง CANBUS ที่ 3

The PCAN-View software interface displays the following data:

Receive / Transmit	CAN-ID	Type	Length	Data	Cycle Time	Count
Receive	00000103h		8	A4 00 00 00 00 00 00 00	99.8	577
	00000102h		8	15 00 00 00 00 00 00 00	103.0	557
	00000104h		8	02 00 00 00 00 00 00 00	100.6	554
	00000101h		8	03 00 00 00 00 00 00 00	104.2	558
	00000105h		8	00 00 00 00 00 00 00 00	99.9	577
Transmit	<Empty>					

PCAN-View status bar: Connected to hardware PCAN-USB, Device ID FFh, Bit rate: 125 kbit/s, Status: OK, Overruns: 0, QXmtFull: 0.

รูปที่ 6.16 ข้อมูลการสื่อสารของ CANBUS



ภาคผนวก ข

บทความที่ได้รับการตีพิมพ์เผยแพร่

รายชื่อบทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่

Nambunlue, W., Donjaroennon, N., Leeton, U., Nuchkum, S., Singhand, K., Promlungd, N., & Srisertpol, J. (2024). Internet of Things (IoT) With the Power Meter Using V-BOX. ISCEAS-0055(1), 183-192.



ISCEAS-0055

Internet of Things (IoT) With the Power Meter Using V-BOX

Wattana Nambunlue^a, Natthapon Donjaroennon^b, Uthen Leeton^b, Suphatchakan Nuchkum^c, Kajornsak Singhan^d, Noppalith Promlung^d, Jiraphon Srisertpol^{*}

^a School of Mechatronic Engineering, Institute of Engineering, Suranaree University of Technology Nakhon Ratchasima, Thailand.

E-mail address: teicon11w@gmail.com

^b School of Electrical Engineering, Institute of Engineering, Suranaree University of Technology Nakhon Ratchasima, Thailand.

E-mail address: PreNatthapon@gmail.com, uthen.leeton@gmail.com

^c School of Mechatronic Engineering, Institute of Engineering, Suranaree University of Technology Nakhon Ratchasima, Thailand.

E-mail address: suphatchakan.nuchkum@gmail.com

^d Faculty of Industrial Technology, Nakhon Phanom University, Thailand.

E-mail address: kjssh05082514@hotmail.com, noppalith5333@gmail.com

^{*} School of Mechanical Engineering, Institute of Engineering, Suranaree University of Technology Nakhon Ratchasima, Thailand.

E-mail address: jiraphon@sut.ac.th

Abstract

This paper is a study of working and applying the Internet of thing (IoT) together with Power Meter, bringing data from Power Meter into the Cloud, such as History Data, Alarm, graphing or monitoring. Display required data in the form of Web Cloud SCADA. And data can be viewed through Computer/Laptop or Smartphone that measures and collects energy consumption data both in real-time and in retrospect. In addition, it can notify system errors and send notification messages to the application line to help users obtain and analyze data immediately and maximize efficiency and this paper will use V-BOX to validate power meter data.

Keywords: Internet of Thing (IoT), Power Meter, V-Box Software.

1. INTRODUCTION

Nowadays, in the industrial sector, it is an era where manufacturers and entrepreneurs are accelerating to transform themselves into a full industry 4.0. Everyone is aware of the benefits of incorporating technology into their operations to increase work efficiency even further. Digital technology and the Internet are important factors in the development of industries in the age of Industry 4.0, especially the Internet of Things (IoT) [1], which plays an important role in

helping collect data to analyze, improve and increase the efficiency of industrial processes [2]. And can facilitate the worker.

The Internet of Things is an important part that will enable the industry to move towards Industry 4.0, which facilitates operators. What makes Industry 4.0 better than any other era [3]. That is, there is a development in data recording, data allocation, and various resources that connect the industry. Industry 4.0 makes every job easier, more detailed and can make changes to plans efficiently [4].

Thailand's industry today has developed rapidly resulting in the use of energy in various forms in addition, every year, therefore, it is necessary to manage energy sufficiently and appropriately [5]. There are many forms of energy management, such as improving the power factor and managing electricity usage (Demand Controller) and choosing high-efficiency appliances. (Electrical appliances or devices that work more but consume less electricity), etc. Peak Demand Configurator which is an important component of the electricity bill in some factories, especially factories that operate machinery only during the daytime, may find that 30 - 35 % of the total electricity bill is a part of the calculation of the peak demand, which has a high electricity price. Therefore, controlling the use of electric power appropriately will help the factory to reduce the electricity bill to be lower [6]. Or pay the same electricity cost but can use more electricity.

2. SMART IOT OF THE POWER METER

2.1 T330 Power Meter

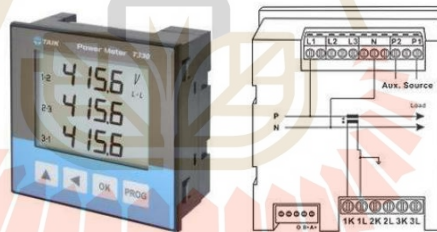


Fig. 1: T330-Power Meter

T330 Power Meter sends and receives data in the form of MODBUS Frame, which is communication and framing based on hexadecimal code, valid MODBUS address ranges from 1-255 address, Function code: 03H for reading data, 06H for Writing data, Data field: Starting address of register N WORDS reading and N writing, Error check: 16bit CRC, Bit Per Byte: Access to sub-menus 485>>FrAE to set-up. That is a communication using hexadecimal numbers set in Table 1. bit and byte transmission from the power meter and Table 2. addresses

used to send data.

Table 1: BIT AND BYTE TRANSMISSION FROM THE POWER METER

Start Bit	Data Bit	Parity	Stop	Frame
1	octal number	NONE	2	N-8-2
1	octal number	ODD	1	O-8-1
1	octal number	EVEN	1	E-8-1
1	octal number	NONE	1	N-8-1

Table 2: ADDRESSES USED TO SEND DATA

Adress	Hex-number	Parameter	Type	Word	Access	Unit
4096	1000H	I_R L1	F	2	R	Amp
4098	1002H	V_RN	F	2	R	Vol
4100	1004H	V_RS	F	2	R	Vol
4102	1006H	VA_R	F	2	R	VA
4104	1008H	W_R	F	2	R	Watt
4106	100AH	Var_R	F	2	R	Var
4108	100CH	PF_R	F	2	R	PF
4138	102AH	ΣA	F	2	R	Amp
4140	102CH	ΣVO	F	2	R	Vol
4142	102EH	ΣVL	F	2	R	Vol
4144	1030H	ΣVA	F	2	R	VA
4146	1032H	ΣW	F	2	R	Watt
4148	1034H	ΣVar	F	2	R	Var
4150	1036H	ΣPF	F	2	R	PF
4152	1038H	ΣHZ	F	2	R	Hz

* F is Float

2.2 V-BOX Wecon IoT Gateway

V-NET is a data management program for V-BOX. That acts like a SCADA program. V-BOX can create display graphics, create ALARM tables, generate power buttons to control connected devices. And most importantly, can control and manage many V-BOX simultaneously and located on CLOUD SERVER which does not need to install any program. All done through a web browser, Following in Fig. 2 Private Cloud Solution.

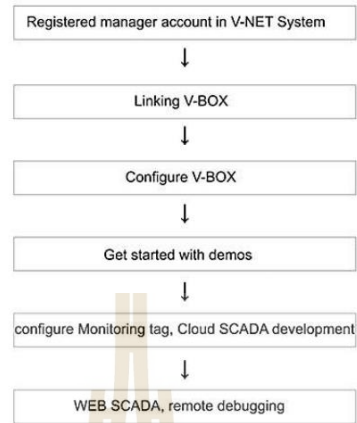


Fig. 2: Private Cloud Solution.

3. METHODOLOGY

3.1 Design and Control system

The design of the data transmission control system is shown in Figure 3. The equipment consists of the T330 Power Meter, Current Transformer, Switching, Sensor and V-BOX Control.

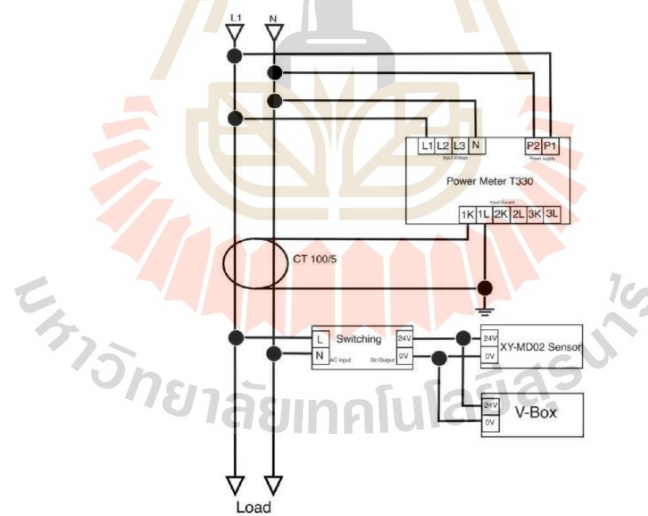


Fig. 3: Power Line Diagram

V-BOX has S Series, E Series and H Series, and each Series V-BOX supports three usage modes. V-BOX is the basic hardware of Industrial Internet of Things, it is a data exchange and protocol conversion device for IoT cloud platform to communicate with the system. In addition, the WECON LCM Series remote IO modules can communicate directly with the V-BOX without using a PLC and implement simple logic control and analog acquisition using edge computing [7]. The V-BOX communication is shown in Figure 4.

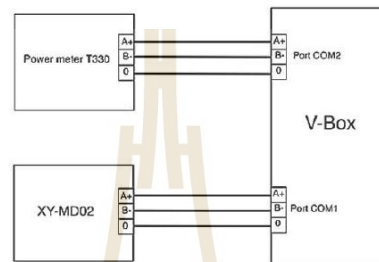


Fig. 4: V-BOX Communication

3.2 Load Lamps

Figure 5. shows the lamp working load of the circuit. The analysis will collect electrical parameters, humidity and temperature. To be adapted as an IoT control system, including notifying the working status sent back to the application line



Fig. 5: Communicate Diagram.

3.3 V-Box Use and Connection Procedure

V-BOX is a device that receives and sends data from control devices such as PLC, Remote-I/O, Power Meter, etc. by connecting with standard protocols (Modbus Protocol) such as RS485, RS422 or Ethernet and sends data to V- NET on the Cloud [8] or through the Internet, or even send data to Brokers with the MQTT Protocol by using LUA Script, which can be written into V-BOX immediately, starting from the V-BOX login. As in Figure 6, when

logging in, the V-NET window will appear for designing the IoT system to be used for control as in Figure 7.



Fig. 6: V-BOX Software

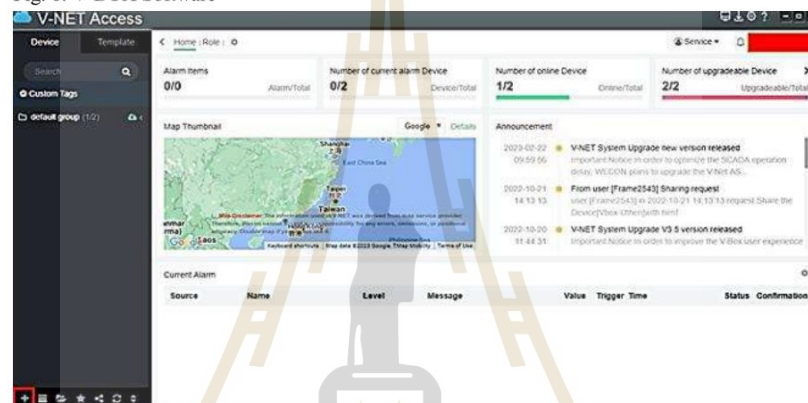


Fig. 7: V-NET Access

3.4 Code to notify information via the Application line

This paper shows the code used to write part of the paper. That will show the lamp status check and Alarm notification.

- Check the lamp status and notify Alarm

```
local LineToken = "IXJc3fwGjcFuZaILRdPP2nccP0I5zcRpLNSAZDIM1Er"
local https = require("https")
local ltn12 = require("ltn12")
function getHttpsUrl(url,header,reqbody)
    local headers,status = https.request{
        method = "POST",
        url = url,
        source = ltn12.source.string(reqbody),
        headers = header }
end
```

```

end
function getMessageUrl(lineMessage)
    local url = "https://notify-api.line.me/api/notify"
    local reqMess = "message="..lineMessage
    local headers = {["Authorization"] = "Bearer "..LineToken,
["Content-Type"] = "application/x-www-form-urlencoded",
["Content-Length"] = #reqMess }
    getHttpsUrl(url, headers, reqMess)
end

function Line_Notify.main()
    local bitValue = addr_getfloat("@Current")
    local message = ''
    local C=bitValue*1000
    C=math.ceil(C)
    C=C/1000
    if bitValue > 0.3 and bitValue<0.6 then
        message = ' V-Box Current, Load on 1 Lamp Value='..C
        getMessageUrl(message)
    elseif bitValue > 0.7 and bitValue< 1 then
        message = ' V-Box Current, Load on 2 Lamp Value='..C
        getMessageUrl(message)
    elseif bitValue > 1.1 and bitValue<1.5 then
        message = ' V-Box Current, Load on 3 Lamp Value='..C
        getMessageUrl(message)
    elseif bitValue > 1.5 and bitValue<1.9 then
        message = ' V-Box Current, Load on 4 Lamp Value='..C
        getMessageUrl(message)
    elseif bitValue > 1.9 and bitValue<2.3 then
        message = ' V-Box Current, Load on 5 Lamp Value='..C
        getMessageUrl(message)
    elseif bitValue > 2.3 then
        message = ' V-Box Current, Load on 6 Lamp Value='..C..'n!!! V-Box Alarm, Over Current
Value !!!'
        getMessageUrl(message)
    end
end
end

```

- Parameter notifications sent to line applications

```

local LineToken = "IXJc3fwGjcFuZaILRdPP2nccP0I5zcRpLNSAZDIM1Er"
local https = require("https")
local ltn12 = require("ltn12")
function getHttpsUrl(url,header,reqbody)
    local headers, status = https.request{
        method = "POST",
        url = url,
        source = ltn12.source.string(reqbody),
        headers = header }
end
function getMessageUrl(lineMessage)
    local url = "https://notify-api.line.me/api/notify"
    local reqMess = "message="..lineMessage
    local headers = {["Authorization"] = "Bearer " ..LineToken,
["Content-Type"] = "application/x-www-form-urlencoded",
["Content-Length"] = #reqMess }
    getHttpsUrl(url, headers, reqMess)
end
function Value_Voltage.main()
    local Current = addr_getfloat("@Current")
    local Voltage = addr_getfloat("@Voltage")
    local ActiveP = addr_getfloat("@Active Power")
    local AppearanceP = addr_getfloat("@Appearance Power")
    local PF = addr_getfloat("@PowerFactor")
    local Freq = addr_getfloat("@Frequency")
    local Watthour = addr_getfloat("@WH")
    local Temp = addr_getfloat("@Temperature 1")
    local Humu = addr_getfloat("@Humidity 1")
    local T=Temp*1000
    T=math.ceil(T)/1000
    local H=Humu*1000
    H=math.ceil(H)/1000
    local C=Current*1000
    C=math.ceil(C)/1000
    local V=Voltage*1000
    V=math.ceil(V)/1000
    local pf=PF*1000

```

```

pf=math.ceil(pf)/1000
pf=math.abs(pf)
local f=Freq*1000
f=math.ceil(f)/1000
local WH=Watthour/1000
local P=math.ceil(ActiveP)
P=P/1000
S=math.ceil(AppearanceP)
S=S/1000
local message = "
message = "V-Box ,Load Current Value = '..C..' A\nVoltage Value = '..V..' V\nActive Power Value = '..P..'
kW\nAppearance Power Value = '..S..' kVA \nPower Factor Value = '..pf..'\nFrequency Value = '..f..'
Hz\nWatt-hour Value = '..WH..' kWh\nTemperature Value = '..T..' C\nHumidity Value (Percent RH) = '..H
getMessageUrl(message)
end

```

4. RESULT

Figure 8. shows real-time data display in the Cloud SCADA system, users can view through the V-NET program or on the browser and mobile application. That is a remote data display online.

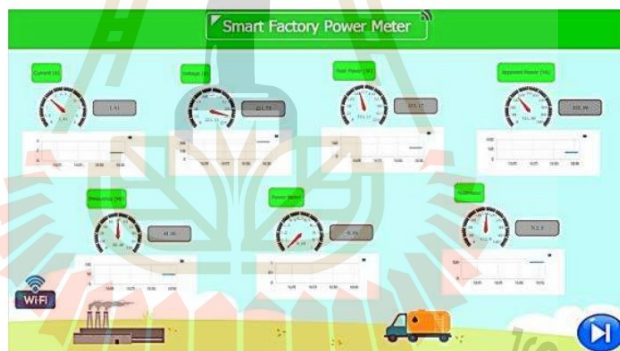


Fig. 8. Real-time Power Meter Data Visualization on Cloud SCADA.

Alarm display on the Cloud SCADA web page, there will be an alarm notification sound asset following in Figure 9. Users can view the alarm through the browser website or V-NET program and mobile application. It is a remote display on the system. Online, To set, Alarm, 5 alarm parameters are set: Voltage Alarm $\pm 10\%$, Current Alarm > 2.3 A (6 Lamps), Temperature Alarm > 70 C, Humidity Alarm $> 45\%RH$

Name	Alarm Content	Value	Alarm time	Status	Confirm user name	Marked As
Alarm Over Current	Over Current	2.13	2023-03-16 15:31:27	Unconfirmed	USDA	Confirm

Fig. 9. Real-time data alarm results in Cloud SCADA.

Real-time data display on V-NET program that users can view real-time data through V-NET program and edit data. It is a way to display remote information and modify the system remotely online. Following in Figure 10-11.

Select All	Status	Name	Value	Port	Read Address	Edit
☐	☒	Appearance Power	85.86	VA	COM2 1:4 4102	☒ Edit ☒ Copy ☒ Move ☒ Delete
☐	☒	Wht	518.0	Wht	COM2 1:4 4100	☒ Edit ☒ Copy ☒ Move ☒ Delete
☐	☒	Frequency	49.99	Hz	COM2 1:4 4152	☒ Edit ☒ Copy ☒ Move ☒ Delete
☐	☒	PowerFactor	0.91	COM2	1:4 4150	☒ Edit ☒ Copy ☒ Move ☒ Delete
☐	☒	Active Power	78.58	W	COM2 1:4 4104	☒ Edit ☒ Copy ☒ Move ☒ Delete
☐	☒	Voltage	221.70	V	COM2 1:4 4098	☒ Edit ☒ Copy ☒ Move ☒ Delete
☐	☒	Current	0.39	A	COM2 1:4 4096	☒ Edit ☒ Copy ☒ Move ☒ Delete

Fig. 10. Real-time data display of Power Meter

Select All	Status	Name	Value	Port	Read Address	Edit
☐	☒	Humidity	37.2	%RH	COM1 2:3 2	☒ Edit ☒ Copy ☒ Move ☒ Delete
☐	☒	Temperature	22.5	C	COM1 2:3 1	☒ Edit ☒ Copy ☒ Move ☒ Delete

Fig. 11. Real-time data display of Sensor XY-MD02 (Temp&Hum).

Historical Data collection results show the measurement results of the Power Meter and Sensor [9] XY-MD02 of all 6 lamp loads. It can display remote data and modify the system remotely online as Figure 12, can be used to analyze the operation and various problems in the system to fix and improve in the future.



Fig. 12. Notify daily parameters via the application line.

5. CONCLUSION

Internet of Things (IoT) technology is an intermediary measuring device. In this paper are Power meters and temperature and humidity sensors. Send data to the Internet on the cloud (Cloud), which connects via Port RS485 and Internet signal (Wifi), can view values online via Web Cloud SCADA, which records data backward and real-time, can alert the error of system according to predetermined conditions. In addition, notifications can be sent to the application line by taking data from those saved values for analysis. Management of controlling or improving the use of electricity with maximum efficiency.

ACKNOWLEDGMENT

This work was supported by the Suranaree University of Technology.

REFERENCES

- [1] A. R. Biswas and R. Giaffreda, "IoT and cloud convergence: Opportunities and challenges," 2014 IEEE World Forum on Internet of Things (WF-IoT), Seoul, Korea (South), 2014, pp. 375-376.
- [2] L. Salman et al., "Energy efficient IoT-based smart home," 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), Reston, VA, USA, 2016, pp. 526-529.
- [3] C. Sharma and N. K. Gondhi, "Communication Protocol Stack for Constrained IoT Systems," 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU), Bhimtal, India, 2018, pp. 1-6.
- [4] J. Kumar and P. R. Ramesh, "Low-Cost Energy Efficient Smart Security System with Information Stamping for IoT Networks," 2018 3rd International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), Bhimtal, India, 2018, pp. 1-5.
- [5] P. Yadav and S. Vishwakarma, "Application of Internet of Things and Big Data towards a Smart City," 2018 3rd International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), Bhimtal, India, 2018, pp. 1-5.
- [6] A. K. Gupta and R. Johari, "IOT based Electrical Device Surveillance and Control System," 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), Ghaziabad, India, 2019, pp. 1-5.

- [7] J. Fox, A. Donnellan and L. Doumen, "The deployment of an IoT network infrastructure, as a localised regional service," 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland, 2019, pp. 319-324.
- [8] Y. Shoji, K. Nakauchi, W. Liu, Y. Watanabe, K. Maruyama and K. Okamoto, "A Community-Based IoT Service Platform to Locally Disseminate Socially-Valuable Data: Best effort local data sharing network with no conscious effort," 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland, 2019, pp. 724-728.
- [9] L. R. Prando, E. R. de Lima, L. S. de Moraes, M. Biehl Hamerschmidt and G. Fraindenraich, "Experimental Performance Comparison of Emerging Low Power Wide Area Networking (LPWAN) Technologies for IoT," 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland, 2019, pp. 905-908.



ประวัติผู้เขียน

นายวัฒนา นามบุญลือ เกิดเมื่อวันที่ 13 พฤษภาคม พ.ศ. 2543 จังหวัดนครราชสีมา สำเร็จการศึกษาระดับประกาศนียบัตรวิชาชีพ โครงการวิทยาลัยเทคโนโลยีฐานวิทยาศาสตร์ วิทยาลัยเทคนิคสุรนารี อำเภอโชคชัย จังหวัดนครราชสีมา ในปีการศึกษา 2561 และสำเร็จการศึกษาระดับปริญญาตรีจากคณะวิศวกรรมศาสตร์ สาขาวิศวกรรมเมคคาทรอนิกส์ มหาวิทยาลัยเทคโนโลยีสุรนารี ในปีการศึกษา 2564 หลังจากสำเร็จการศึกษาได้เข้าการศึกษาต่อในระดับปริญญาโท สาขาวิชาวิศวกรรมเมคคาทรอนิกส์ มหาวิทยาลัยเทคโนโลยีสุรนารี ในปี พ.ศ. 2565 จนถึงปัจจุบัน

