

บทที่ 2

ปฐกษัตริ์รณกรรมและงานวิจัยที่เกี่ยวข้อง

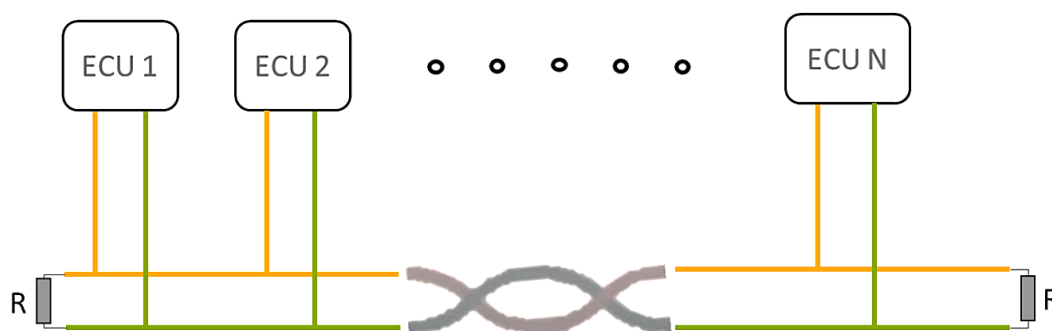
2.1 บทนำ

(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

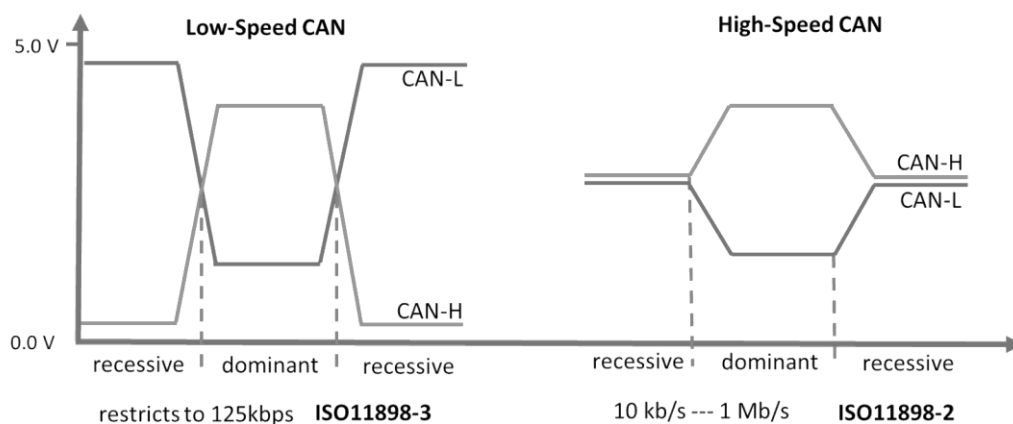


รูปที่ 2.1 ระบบสื่อสาร CAN BUS (Controller Area Network)

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 เพื่อช่วยลดคลื่นสะท้อนและเพิ่มความเสถียรในการส่งข้อมูล ในขณะที่ Low-Speed CAN มีการปิดตัวต้านทานที่ทุกอุปกรณ์ เพื่อป้องกันการเกิดคลื่นสะท้อนและเพิ่มความสามารถในการทนทานต่อความผิดพลาด (Fault Tolerant) ทำให้สามารถทำงานได้อย่างต่อเนื่องแม้ในกรณีที่เกิดปัญหาในระบบสายสัญญาณ

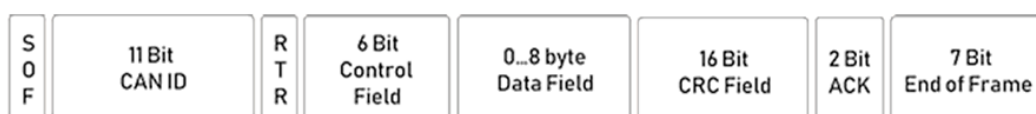


รูปที่ 2.2 สัญญาณ Low-Speed CAN และสัญญาณ High-Speed CAN

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) โดยการเลือกใช้ 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

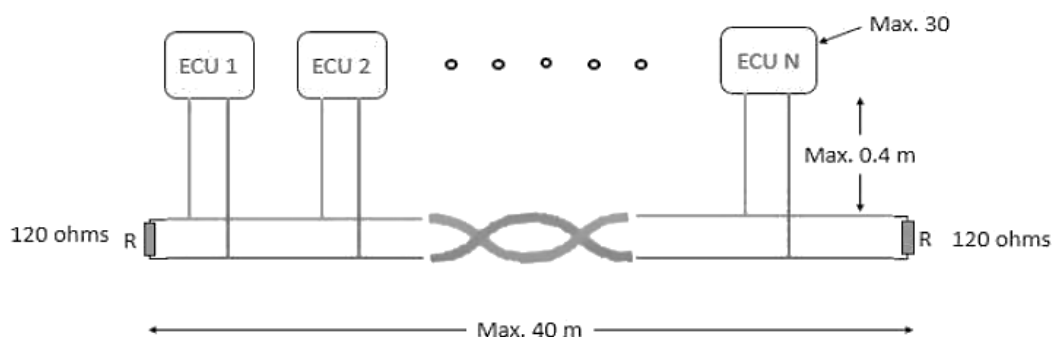
2.4.3 CAN FD (Flexible Data-Rate)

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

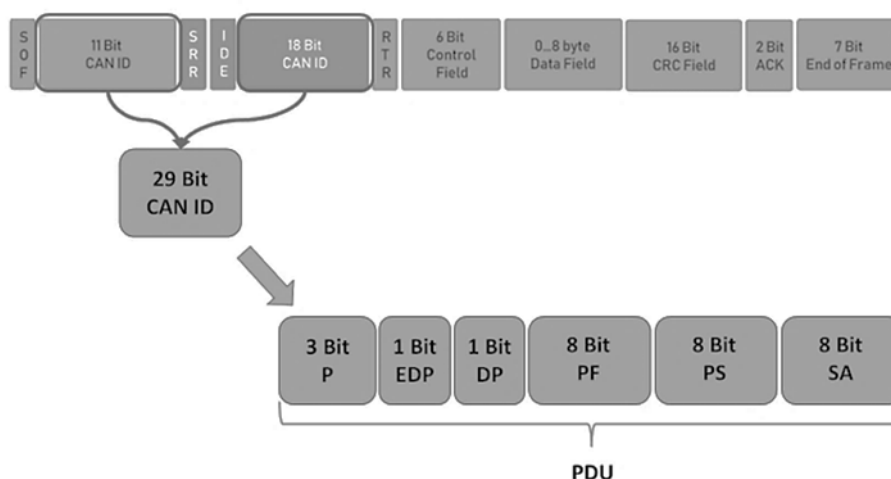


รูปที่ 2.5 การเชื่อมต่อ ECU ของมาตรฐาน SAE J1939

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)

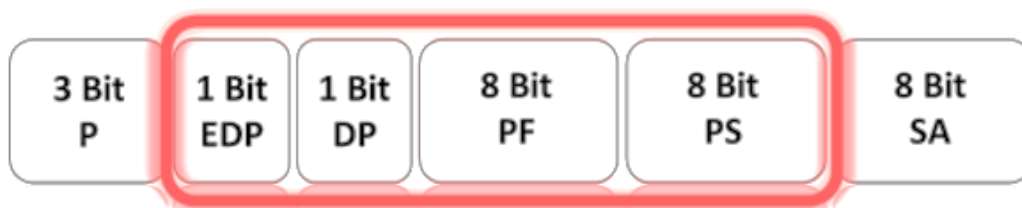
ส่วนประกอบของ PDU

- 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

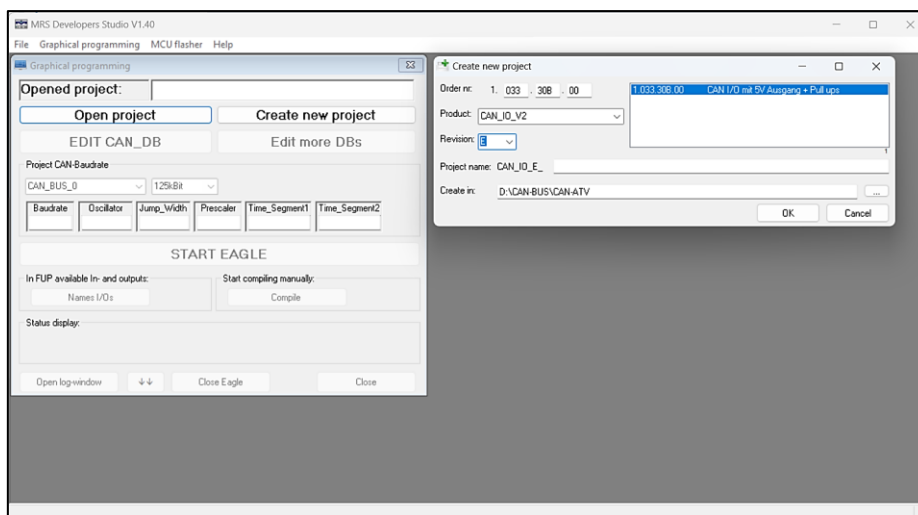
CAN I/O PLC ของ MRS ที่แสดงดังรูปที่ 2.8 เป็นอุปกรณ์ที่ออกแบบมาเพื่อทำงานร่วมกับระบบ CAN BUS โดยทำหน้าที่ควบคุมและประมวลผลการทำงานของอุปกรณ์ต่าง ๆ ในระบบอัตโนมัติแบบเรียลไทม์ เช่น เซ็นเซอร์ มอเตอร์ และอุปกรณ์ I/O ด้วยการออกแบบที่ยืดหยุ่น ทำให้ CAN I/O PLC สามารถตั้งโปรแกรมเพื่อควบคุมกระบวนการที่ซับซ้อนได้อย่างแม่นยำ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

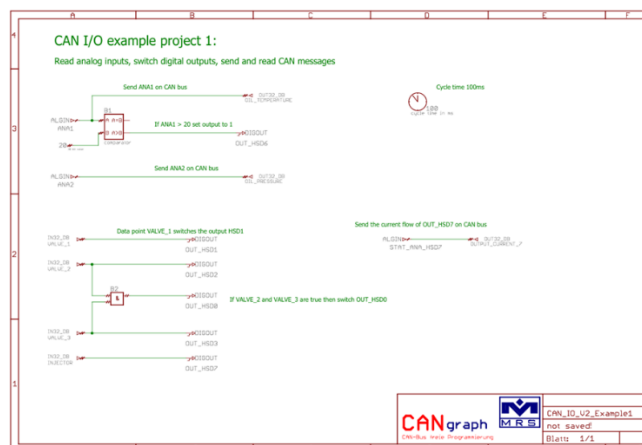
MRS Developers Studio เป็นซอฟต์แวร์ที่ออกแบบมาเพื่อสนับสนุนการพัฒนาระบบสื่อสารใน CAN BUS โดยเฉพาะการทำงานร่วมกับ CAN I/O PLC จากรูปที่ 2.9 ซอฟต์แวร์นี้ช่วยให้ผู้ใช้สามารถตั้งโปรแกรมการทำงานของ CAN I/O PLC ได้อย่างละเอียดตามความต้องการของระบบ พร้อมทั้งมีฟังก์ชันการจำลองการทำงานในสภาพแวดล้อมเสมือนจริง 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



รูปที่ 2.10 ตัวอย่างการใช้งาน CAN I/O PLC ร่วมกับ MRS Developers Studio

2.7 หลักการและการประยุกต์ใช้ตัวกรองความถี่ต่ำ (Low-pass Filter)

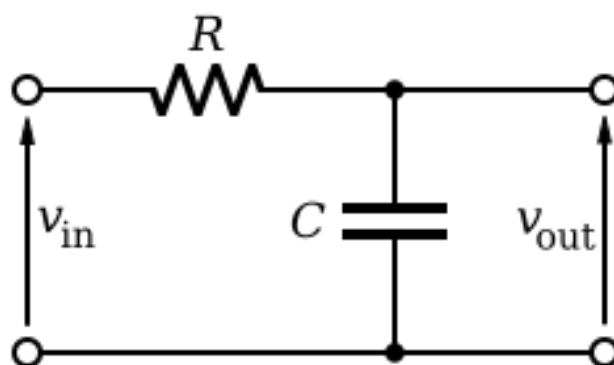
ตัวกรองความถี่ต่ำ (Low-pass filter) เป็นวงจรที่ช่วยให้สัญญาณความถี่ต่ำผ่านไปได้ ขณะที่สัญญาณความถี่สูงจะถูกลดทอนหรือตัดออก ซึ่งมีบทบาทสำคัญในระบบอิเล็กทรอนิกส์ ระบบสื่อสาร และการประมวลผลสัญญาณ โดยตัวกรองนี้สามารถสร้างขึ้นได้จากองค์ประกอบพื้นฐาน เช่น ตัวต้านทาน (R) และตัวเก็บประจุ (C) ซึ่งสามารถกำหนดค่าความถี่ตัด (cutoff frequency) ได้จากสมการ

$$f_c = \frac{1}{2\pi RC}$$

เมื่อ f_c คือ ความถี่ตัด, R คือ ค่าความต้านทาน, และ C คือ ค่าความจุของตัวเก็บประจุ

ตัวอย่างการต่อวงจรกรองความถี่ต่ำ ดังรูปที่ 2.11 (Low-pass filter) ประเภทของตัวกรองความถี่ต่ำแบ่งออกเป็นสองประเภทหลัก ได้แก่ ตัวกรองแบบพาสซีฟ (Passive Low-pass Filter) ซึ่งใช้เฉพาะตัวต้านทานและตัวเก็บประจุ ไม่ต้องการแหล่งจ่ายไฟเพิ่มเติม และตัวกรองแบบแอคทีฟ (Active Low-pass Filter) ซึ่งมีการใช้แอมป์โอป (Op-Amp) เพื่อเพิ่มประสิทธิภาพและสามารถขยายสัญญาณได้ การประยุกต์ใช้งานของตัวกรองความถี่ต่ำมีหลากหลาย เช่น ในการประมวลผลสัญญาณเสียงเพื่อกำจัดสัญญาณรบกวนความถี่สูง ทำให้เสียงมีความชัดเจนขึ้น ใช้ในระบบประมวลผลภาพเพื่อลดสัญญาณรบกวน และใช้ในระบบสื่อสารเพื่อกรองสัญญาณที่ไม่ต้องการ

นอกจากนี้ ในระบบควบคุมยานยนต์ไฟฟ้า ตัวกรองความถี่ต่ำยังมีบทบาทสำคัญในการกรองสัญญาณเซ็นเซอร์ ลดสัญญาณรบกวนในระบบ CAN Bus และช่วยให้การควบคุมมอเตอร์มีประสิทธิภาพมากขึ้น โดยสรุป ตัวกรองความถี่ต่ำเป็นองค์ประกอบสำคัญที่ช่วยให้ระบบอิเล็กทรอนิกส์และการสื่อสารทำงานได้อย่างมีประสิทธิภาพ การเลือกใช้ตัวกรองที่เหมาะสมขึ้นอยู่กับความต้องการของแต่ละระบบ ซึ่งสามารถช่วยปรับปรุงคุณภาพของสัญญาณและลดการรบกวนที่ไม่พึงประสงค์ได้อย่างมีประสิทธิภาพ

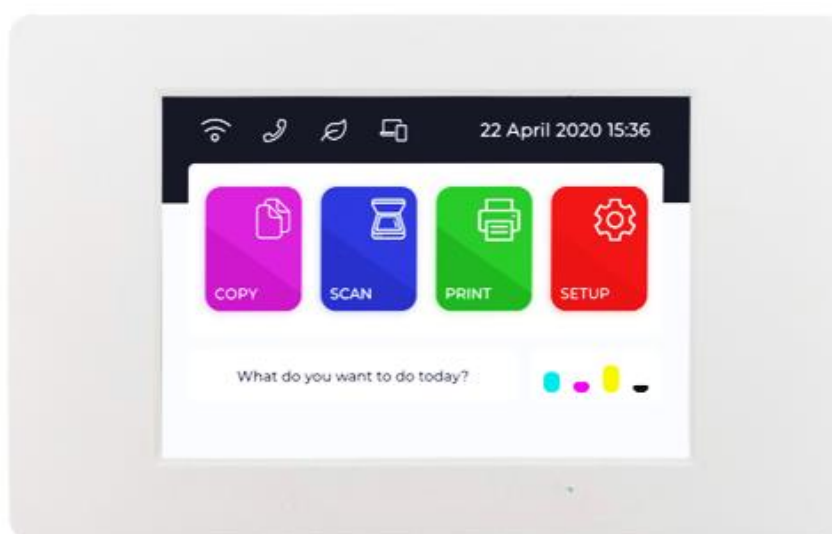


รูปที่ 2.11 ตัวอย่างการต่อวงจรกรองความถี่ต่ำ (Low-pass filter)

2.8 จอแสดงผลสถานการณ์ทำงานของระบบ CANBUS

จอแสดงผล ATD3.5-S3 เป็นบอร์ดพัฒนาที่ต่อยอดจากไมโครคอนโทรลเลอร์ ESP32-S3 ซึ่งเป็นชิปแบบ Dual-core ที่มีประสิทธิภาพในการประมวลผลสูง พร้อมหน่วยความจำ Flash ขนาด 8MB และ PSRAM ขนาด 8MB รองรับการทำงานจากระบบที่ต้องการการประมวลผลกราฟิกและเสียงในระดับที่ซับซ้อน โดยบอร์ดรุ่นนี้มาพร้อมกับหน้าจอสัมผัสแบบ Capacitive ขนาด 3.5 นิ้ว ความละเอียด 480x320 พิกเซล ชนิด IPS (In-Plane Switching) ซึ่งให้ภาพที่คมชัด สีสมจริง 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

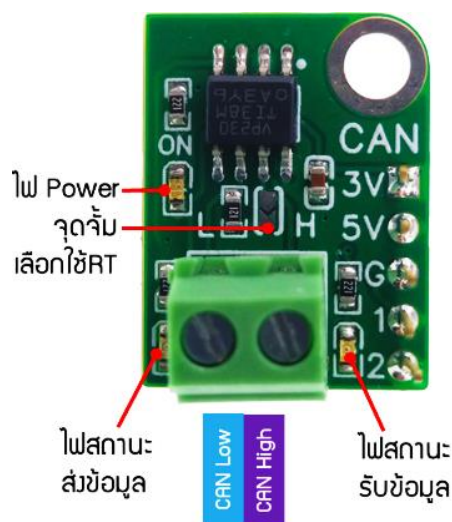


รูปที่ 2.12 จอATD3.5-S3 แสดงสถานะการทำงานของ ATV

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



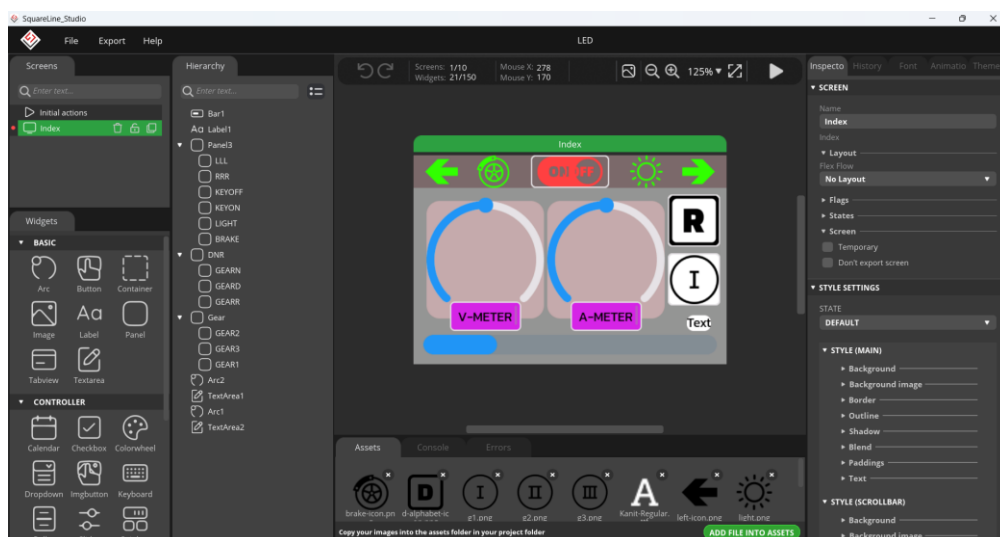
รูปที่ 2.13 CAN BUS Shield

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



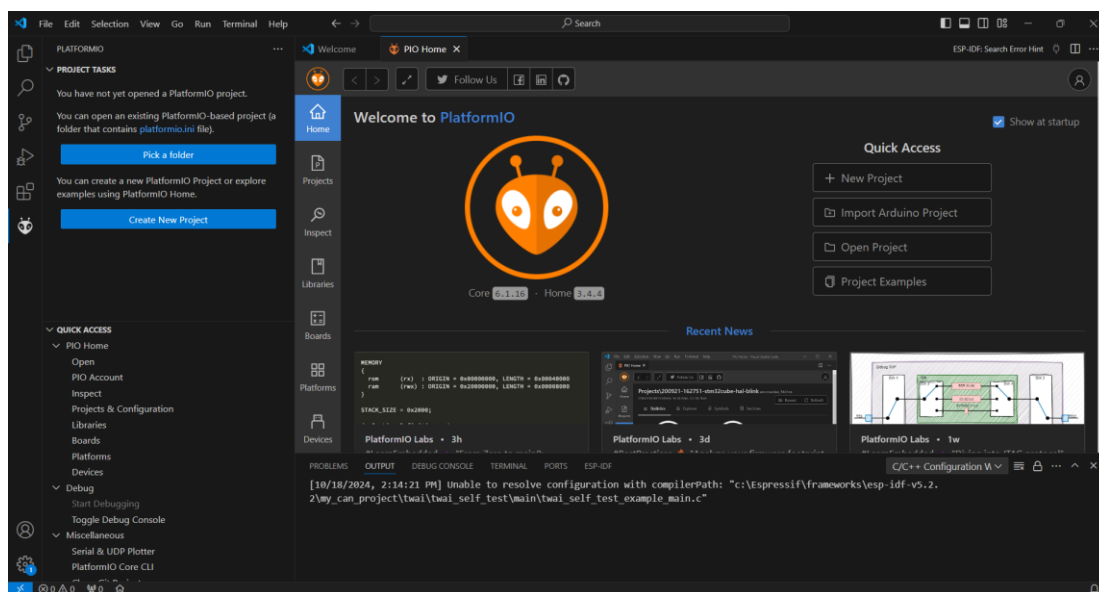
รูปที่ 2.14 ซอฟต์แวร์ SquareLine Studio

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 ปรัชศัณวรรณกรรมและงานวิจัยที่เกี่ยวข้อง

(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 การออกแบบวงจรที่ใช้ไมโครคอนโทรลเลอร์ 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 รวมถึงป้องกันการเสียมูลของเซลล์แบตเตอรี่ 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 เพื่อตรวจจับและควบคุม
พลังงานการรบกวนให้เหมาะสมกับสภาวะต่าง ๆ ของเครือข่าย