



การออกแบบบอร์ดแปลงสัญญาณระหว่าง CAN bus และ Modbus สำหรับรถขนส่งผู้โดยสารอัตโนมัติ

ก่อเกียรติ อ้อดทรัพย์¹ และ สุรัชชัย วงษ์ฟูเกียรติ^{2*}

¹ สาขาวิศวกรรมไฟฟ้า, คณะวิศวกรรมศาสตร์, มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี

² หลักสูตรวิศวกรรมยานยนต์ไฟฟ้า, คณะเทคโนโลยีอุตสาหกรรม, มหาวิทยาลัยราชภัฏนครราชสีมา

* ผู้ประสานงานเผยแพร่ (Corresponding Author), surachai.w@nrru.ac.th

วันที่รับบทความ: 19 มิถุนายน 2567; วันที่ทบทวนบทความ: 22 กันยายน 2567; วันที่ตอบรับบทความ: 24 ตุลาคม 2567

วันที่เผยแพร่ออนไลน์: 19 ธันวาคม 2567

บทคัดย่อ: รถขนส่งผู้โดยสารอัตโนมัติ (Automated People Mover หรือ APM) เป็นระบบขนส่งมวลชนอัตโนมัติเต็มรูปแบบที่นิยมในการขนส่งผู้โดยสารระหว่างอาคารผู้โดยสารและอาคารเทียบเครื่องบินระบบนี้สามารถควบคุมได้ทั้งจากภายในตัวรถและจากสถานีควบคุม โดยสถานีจะต้องสามารถตรวจสอบสถานะการทำงานของรถได้อย่างต่อเนื่อง การวิจัยด้านการออกแบบและพัฒนาบอร์ดแปลงสัญญาณอัตโนมัติในประเทศไทย รวมถึงการพัฒนาชุดฝึกเพื่อจำลองการทำงานของรถขนส่งผู้โดยสารอัตโนมัติ นิยมใช้โครงสร้างพื้นฐานจากระบบไฟฟ้าซึ่งใช้โปรโตคอลการสื่อสารแบบแคนบัส (CAN bus) ขณะที่การสื่อสารภายในสถานีใช้โปรโตคอลมอดบัส (Modbus) ซึ่งเป็นมาตรฐานในอุตสาหกรรมอัตโนมัติเพื่อเชื่อมโยงการทำงานระหว่างโปรโตคอลทั้งสองจึงจำเป็นต้องมีตัวกลางเชื่อมต่องานวิจัยนี้มุ่งเน้นการพัฒนาบอร์ดสำหรับแปลงสัญญาณการสื่อสารระหว่างมอดบัสอาร์ทียู (Modbus RTU) และแคนบัส (CAN bus) โดยใช้โปรโตคอล SAE J1939 ซึ่งบอร์ดนี้มีชื่อว่า CAN2RS485 ผู้ใช้งานสามารถกำหนดแอดเดส (CAN ID) ได้ตามความต้องการ และสามารถกำหนดให้ข้อมูลจากแอดเดสส่งไปยังรีจิสเตอร์ของมอดบัสได้ตามความต้องการเพื่อแสดงผลการทำงานของรถขนส่งผู้โดยสารอัตโนมัติที่สถานี นอกจากนี้ยังสามารถรับข้อมูลจากรีจิสเตอร์ของมอดบัสและส่งไปยังแคนบัสเพื่อควบคุมการทำงานของรถขนส่งผู้โดยสารอัตโนมัติได้ ในส่วนของมอดบัส บอร์ดยังสามารถกำหนดให้ทำงานในโหมดมาสเตอร์หรือสลาฟได้ตามความต้องการ

คำสำคัญ: แคนบัส; มอดบัส; รถขนส่งผู้โดยสารอัตโนมัติ; แปลงสัญญาณ

Designing a Signal Conversion Circuit Board between CAN bus and Modbus for Automated People Mover

Kokiat Aodsup¹ and Surachai Wongfookeat^{2*}

¹ Department of Electrical Engineering, Faculty of Engineering, Rajamangala University of Technology Lanna Tak

² Electric Vehicle Engineering Program, Faculty of Industrial Technology, Nakhon Ratchasima Rajabhat University

* Corresponding author, E-mail: surachai.w@nrru.ac.th

Received: 19 June 2024; Revised: 22 September 2024; Accepted: 24 October 2024

Online Published: 19 December 2024

Abstract: The Automated People Mover (APM) is a fully automated mass transit system commonly used for transporting passengers between terminal buildings and boarding gates. This system can be controlled both from within the vehicle and from a control station, with the station being required to continuously monitor the vehicle's operational status. Research on the design and development of automated people movers in Thailand, including the development of training kits to simulate the operation of APMs, often utilizes infrastructure based on electric buses, which use the CAN bus communication protocol. Meanwhile, station communications typically use the Modbus protocol, a standard in industrial automation. To link the operations between these two protocols, an intermediary connector is necessary. This research focuses on the development of a board for converting communication signals between Modbus RTU and CAN bus using the SAE J1939 protocol. This board, called CAN2RS485, allows users to set the CAN ID as desired, and to configure data from a selected CAN ID to be sent to the Modbus registers for displaying the APM's operational status at the station. Additionally, it can receive data from Modbus registers and send it to CAN bus to control the operation of the APM. In the Modbus section, the board can also be configured to operate in either master or slave mode as needed.

Keywords: CAN bus; Modbus; APM; Converter



1. บทนำ

รถขนส่งผู้โดยสารอัตโนมัติ หรือ เอพีเอ็ม (Automated People Mover, APM) เป็นระบบขนส่งมวลชนที่ควบคุมด้วยระบบอัตโนมัติเต็มรูปแบบ นิยมใช้ในพื้นที่ขนาดเล็ก เช่น สนามบิน เพื่อขนส่งผู้โดยสารระหว่างอาคารผู้โดยสาร (Terminal) และอาคารเทียบเครื่องบิน ในประเทศไทย เอพีเอ็มถูกใช้งานในสนามบินสุวรรณภูมิและโครงการรถไฟฟ้าสายสีทอง ปัจจุบันมีงานวิจัยหลายชิ้นที่มุ่งเน้นการออกแบบและพัฒนาระบบควบคุมการทำงานของเอพีเอ็มเพื่อเพิ่มประสิทธิภาพ รวมถึงงานวิจัยเกี่ยวกับความปลอดภัยและการสร้างมาตรฐานสำหรับระบบเอพีเอ็ม นอกจากนี้ ยังมีงานวิจัยที่นำมาตราฐาน IEC 61850 ซึ่งเป็นโปรโตคอลการสื่อสารสำหรับสถานีไฟฟ้าย่อย มาประยุกต์ใช้กับเอพีเอ็มอีกด้วย [1-4]

ในประเทศไทยมีงานวิจัยหลายชิ้นที่มุ่งเน้นการพัฒนาเอพีเอ็ม โดยเฉพาะด้านการออกแบบและสร้างต้นแบบระบบควบคุมการเดินรถเอพีเอ็ม [5] รวมถึงการออกแบบและสร้างต้นแบบ เอพีเอ็มของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ และการพัฒนาต้นแบบชุดฝึกเอพีเอ็ม ของมหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี โดยใช้โครงสร้างพื้นฐานจากรถบัสไฟฟ้า สำหรับการสื่อสารภายในตัวรถเอพีเอ็ม จะใช้โปรโตคอลแคนบัส (Controller Area Network, CAN bus) ในการสื่อสาร ส่วนการสื่อสารระหว่างตัวรถกับสถานีควบคุมการเดินรถจะใช้โปรโตคอลมอดบัส (Modbus) ซึ่งทำให้เกิดปัญหาในการเชื่อมต่อระหว่างสองโปรโตคอลนี้ งานวิจัยหนึ่ง [6] ได้นำเสนอการออกแบบวงจรแปลงสัญญาณจากแคนบัส เป็น RS-485 แต่ยังไม่มีการนำไปใช้งานจริง ขณะที่ในท้องตลาดมีโมดูลสำหรับ

แปลงสัญญาณระหว่างแคนบัส และ RS-485 จำหน่าย แต่ส่วนใหญ่จะเป็นแบบมอดบัสอาร์ทียูสลาฟ (Modbus RTU slave) และใช้กับแคนโอเพ่น (CANopen) ซึ่งออกแบบมาใช้ในระบบอัตโนมัติในภาคอุตสาหกรรม สำหรับการใช้งานแคนบัสในอุตสาหกรรมยานยนต์หรือยานพาหนะ มักนิยมใช้โปรโตคอล SAE J1939

งานวิจัยนี้มุ่งเน้นการพัฒนาบอร์ดสำหรับแปลงสัญญาณการสื่อสารระหว่างมอดบัสอาร์ทียู ซึ่งสามารถตั้งค่าให้ทำงานในโหมดมาสเตอร์หรือสลาฟได้ กับแคนบัส โดยใช้โปรโตคอล SAE J1939 เพื่อเชื่อมต่อการสื่อสารระหว่างระบบตัวรถเอพีเอ็ม กับสถานีควบคุม

2. โปรโตคอลการสื่อสารของเอพีเอ็ม

2.1 โปรโตคอลแคนบัส

แคนบัส เป็นมาตรฐานการสื่อสารผ่านสายที่ออกแบบมาให้อุปกรณ์ต่างๆ ใช้สื่อสารกัน โดยเฉพาะในรถยนต์ ซึ่งนิยมใช้งานอย่างแพร่หลายเนื่องจากเป็นการสื่อสารที่มีความน่าเชื่อถือสูงและสามารถทำงานแบบเรียลไทม์ ระบบการสื่อสารนี้จะสื่อสารด้วยสาย 2 เส้น โดยที่ปลายทั้งสองด้านจะต้องต่อตัวต้านทาน 120Ω (Terminating Resistor) และมีอัตราการส่งข้อมูล (Bitrate) มีค่าอยู่ระหว่าง 12.5 กิโลบิต/วินาที (kbit/s) ถึง 1 เมกะบิต/วินาที (Mbit/s) ในการส่งข้อมูลสามารถทำงานพร้อมกันได้ โดยที่ทุกอุปกรณ์จะได้รับข้อมูลเหมือนกัน แต่สามารถเลือกรับเฉพาะข้อมูลที่ต้องการได้ และหากมีอุปกรณ์ใดๆ เกิดการชำรุดเสียหาย อุปกรณ์อื่นๆ ระบบยังคงสามารถสื่อสารกันได้ กรณีที่มีการส่งข้อมูลพร้อมกัน ข้อมูลที่มีความสำคัญมากกว่าจะได้รับสิทธิในการส่งก่อน ส่วนข้อมูลที่มีความสำคัญน้อยกว่าจะได้โอกาสส่งใหม่ภายหลัง โดยจะมีตัวนับเรียกว่า ทีอีซี (Transmit Error Counter, TEC) และอาร์อีซี



(Receive Error Counter, REC) ทำหน้าที่นับจำนวนความผิดพลาดที่เกิดจากการรับ-ส่งข้อมูล โดยสถานะความผิดพลาด (Error States) แบ่งเป็น

- ความผิดพลาดแบบแอคทีฟ (Error Active) หมายถึง ทีอีซี และ อาร์อีซี นับได้น้อยกว่า 128 การรับ-ส่งข้อมูลยังปกติ แต่จะส่งเฟรมความผิดพลาด (Error frame) ทุกครั้งที่ตรวจพบความผิดพลาด
- ความผิดพลาดแบบพาสซีฟ (Error Passive) หมายถึง ทีอีซี และ อาร์อีซี นับได้มากกว่าหรือเท่ากับ 128 การรับ-ส่งข้อมูลยังปกติ แต่จะส่งเฟรมความผิดพลาด (Error frame) ทุกครั้งที่ตรวจพบความผิดพลาด
- ตัดการเชื่อมต่อ (Bus-Off) หมายถึง ทีอีซี นับได้มากกว่าหรือเท่ากับ 256 ในสถานะนี้ แคนคอนโทรลเลอร์ (CAN controller) จะไม่สามารถรับ-ส่งข้อมูลบนบัสอีกต่อไปจนกว่าจะได้รับการแก้ไข

การรับ-ส่งข้อมูลผ่านระบบแคน จะใช้แคนเฟรม (CAN frame) ซึ่งแบ่งเป็น 4 ชนิดดังนี้

1. เฟรมข้อมูล (Data Frame) จะทำหน้าที่ส่งข้อมูลไปยังอุปกรณ์อื่นๆ
2. เฟรมรีโมท (Remote Frame) จะทำหน้าที่ร้องขอข้อมูลจากอุปกรณ์อื่นๆ
3. เฟรมความผิดพลาด (Error Frame) จะทำหน้าที่แจ้งทุกอุปกรณ์ในระบบว่ามีความผิดพลาดเกิดขึ้น
4. เฟรมโอเวอร์โหลด (Overload Frame) จะทำหน้าที่เมื่ออุปกรณ์ใช้เวลาเพิ่มขึ้นในการประมวลผลข้อมูล หรือเมื่อเกิดข้อผิดพลาด

โครงสร้างของเฟรมข้อมูลจะแบ่งเป็น

1. เฟรมเริ่มต้น (Start of Frame, SOF) จำนวน 1 บิต จะเป็นตัวบอกให้อุปกรณ์รู้ว่ากำลังจะมีการส่งข้อมูล
2. ไอดี (Identifier, ID) จำนวน 11 บิต เป็นหมายเลขเฉพาะของข้อมูลและเป็นตัวกำหนดความสำคัญของข้อมูล
3. อาร์ทีอาร์ (Remote Transmission Request, RTR) จำนวน 1 บิต จะเป็นตัวกำหนดว่าเป็นข้อมูล หรือเฟรมรีโมท
4. ตัวควบคุม (Control) จำนวน 6 บิต ประกอบด้วย ส่วนขยายไอดี (Identifier Extension Bit, IDE), สำรอง (Reserved) และ ความยาวรหัสข้อมูล (Data Length Code, DLC)
5. ข้อมูลที่ต้องการส่ง (Data) จำนวนตั้งแต่ 0 ถึง 64 บิต
6. ซีอาร์ซี (Cyclic Redundancy Check, CRC) จำนวน 16 บิต สำหรับตรวจสอบความถูกต้องของข้อมูล
7. ยืนยันการรับข้อมูล (Acknowledge, ACK) จำนวน 2 บิต
8. สิ้นสุดแคนเฟรม (End of Frame, EOF) จำนวน 7 บิต

2.2 โพรโตคอล SAE J1939 [8]

แคนบัสมีประสิทธิภาพสูงในรถยนต์ แต่ไม่เหมาะสมสำหรับงานที่ต้องการข้อมูลมากกว่า 8 ไบต์ (byte) SAE J1939 เป็นโพรโตคอลเลเยอร์สูง (High Layer) ที่ใช้แคนบัสเป็นชั้นกายภาพสำหรับการสื่อสาร โดยใช้สายคู่บิดเกลียวหุ้มฉนวน ความยาวสูงสุด 40 เมตร และมีอัตราการส่งข้อมูลมาตรฐาน 250 กิโลบิต/วินาที



รองรับจำนวนโหนด (Node) สูงสุดจำนวน 30 โหนด และรองรับแอปพลิเคชัน (Application) สูงสุด 253 แอปพลิเคชัน รองรับการสื่อสารแบบเพียร์ทูเพียร์ (Peer-to-peer) และบรอดแคสต์ (Broadcast) ความยาวข้อความสูงสุด 1785 ไบต์ โดยมีหมายเลขกลุ่มพารามิเตอร์ (Parameter Group Numbers, PGNs) สำหรับยานพาหนะที่กำหนดไว้ล่วงหน้า และรองรับการจัดการเครือข่าย

2.2.1 หมายเลขกลุ่มพารามิเตอร์ (PGN)

SAE J1939 เป็นโปรโตคอลที่ออกแบบอย่างชาญฉลาดโดยจะใช้ประโยชน์จากไอดีข้อความจำนวน 29 บิต ของแคนบัส จากตารางพารามิเตอร์ที่กำหนดไว้ล่วงหน้าเพื่อให้โปรโตคอลเข้าใจง่าย กลุ่มพารามิเตอร์ เช่น อุณหภูมิเครื่องยนต์ ประกอบด้วย อุณหภูมิน้ำหล่อเย็น อุณหภูมิน้ำมันเชื้อเพลิง และ อุณหภูมิน้ำมัน กลุ่มพารามิเตอร์และหมายเลขของกลุ่มพารามิเตอร์ จะถูกระบุใน SAE J1939 และ SAE J1939/71 ซึ่งประกอบด้วยคำจำกัดความของกลุ่มพารามิเตอร์และหมายเลขของพารามิเตอร์

2.2.2 เอส พี เอ็น (Suspect Parameter Numbers, SPN)

เอสพีเอ็น คือหมายเลขที่กำหนดโดย SAE ให้กับพารามิเตอร์เฉพาะ โดยจะอธิบายรายละเอียดพารามิเตอร์ เช่น ความยาวข้อมูล ชนิดข้อมูล รายละเอียด ออฟเซต (Offset) ช่วง และป้ายอ้างอิง (Label)

2.2.3 ช่วงของหมายเลขกลุ่มพารามิเตอร์

- 0x0000 – 0xEEFF: ข้อความแบบเพียร์ทูเพียร์จำนวน 239 ข้อความที่กำหนดโดย SAE
- 0xEF00 – 0xEFFF: ข้อความแบบเพียร์ทูเพียร์จำนวน 1 ข้อความสำหรับการใช้งานเฉพาะ
- 0xF000 – 0xFEFF: ข้อความบรอดแคสต์จำนวน 3840 ข้อความที่กำหนดโดย SAE
- 0xFF00 – 0xFFFF: ข้อความบรอดแคสต์จำนวน 256 ข้อความสำหรับการใช้งานเฉพาะ

2.3 โปรโตคอล RS-485

โปรโตคอล RS-485 เป็นมาตรฐานการสื่อสารแบบอนุกรมที่กำหนดลักษณะทางไฟฟ้าของตัวส่งและตัวรับเพื่อใช้งานในระบบดิจิทัลแบบสมดุลหลายจุด มีการใช้งานอย่างแพร่หลายในอุตสาหกรรมและการพาณิชย์ เนื่องจากความทนทานต่อสัญญาณรบกวนสูงและความสามารถในการสนับสนุนการสื่อสารระยะไกล ใช้การส่งสัญญาณแบบต่างระดับในการส่งข้อมูล ซึ่งช่วยลดสัญญาณรบกวนและปรับปรุงความสมบูรณ์ของสัญญาณในระยะทางไกล นอกจากนี้ยังรองรับอุปกรณ์สูงสุดถึง 32 อุปกรณ์บนบัสเดียว สามารถสื่อสารได้ในระยะสูงสุด 4,000 ฟุต ที่ความเร็วต่ำ และรองรับอัตราข้อมูลสูงถึง 10 เมกะบิต/วินาที ในระยะทางสั้น รองรับทั้งการสื่อสารทางเดียวและสองทาง โดยโหมดการสื่อสารสองทางต้องใช้สายสองคู่



2.4 โพรโตคอลมอดบัส

มอดบัสเป็นโพรโตคอลการสื่อสารที่ออกแบบมาใช้กับ พีแอลซี (Programmable Logic Controller, PLC) และได้กลายเป็นมาตรฐานในระบบอัตโนมัติ เนื่องจากมีความเรียบง่าย น่าเชื่อถือ และติดตั้งง่าย ทำงานบนสถาปัตยกรรมมาสเตอร์-สลาฟ โดยที่อุปกรณ์มาสเตอร์จะทำหน้าที่เริ่มการสื่อสารและอุปกรณ์สลาฟจะเป็นตัวทำหน้าที่ตอบสนองสามารถมีได้หลายตัว สนับสนุนการสื่อสารทั้งแบบอนุกรมและแบบเครือข่าย โดยการทำงานของมอดบัสจะมีการกำหนดรหัสฟังก์ชัน (Function Codes) เพื่อระบุการดำเนินการ เช่น การอ่านรีจิสเตอร์ (Read Registers) หรือการเขียนรีจิสเตอร์ (Write Registers) นอกจากนี้ยังมีกลไกการตรวจสอบข้อผิดพลาด (Error Checking) เช่น CRC (Cyclic Redundancy Check) เพื่อความสมบูรณ์และความถูกต้องของข้อมูลที่ส่งผ่านเครือข่าย ซึ่งช่วยให้การสื่อสารระหว่างอุปกรณ์มีความน่าเชื่อถือและปลอดภัย

2.4.1 ประเภทของมอดบัส

- มอดบัส อาร์ทียู (Modbus RTU) การทำงานจะใช้การเข้ารหัสแบบไบนารี (Binary) และเป็นการสื่อสารแบบอนุกรม (RS232, RS485)
- มอดบัส แอสกี (Modbus ASCII) จะใช้อักขระแอสกี (ASCII) ในการเข้ารหัสเป็นการสื่อสารแบบอนุกรมเช่นเดียวกับ มอดบัส อาร์ทียู แต่มีประสิทธิภาพน้อยกว่า มอดบัส อาร์ทียู
- มอดบัส ทีซีพี/ไอพี (Modbus TCP/IP) จะใช้อีเทอร์เน็ต (Ethernet) ในการสื่อสารและอนุญาตให้มีการสื่อสารผ่านเครือข่าย ทีซีพี/ไอพี (TCP/IP)

2.4.2 การจัดระเบียบข้อมูลในมอดบัส

ข้อมูลในการสื่อสารแบบมอดบัส จะประกอบด้วย รีจิสเตอร์ (Register) ทั้งหมด 4 รูปแบบ ซึ่งสามารถใช้งานโดยเฉพาะ และแต่ละรูปแบบจะมีที่อยู่ของข้อมูล (data address) อยู่ระหว่าง 0000-270E โดยรีจิสเตอร์แต่ละรูปแบบจะทำหน้าที่ต่าง ๆ ดังนี้

- เอาต์พุตคอล์ย (Coils) จะเป็นเอาต์พุตดิจิทัลที่สามารถอ่านหรือเขียนได้ ตำแหน่งที่อยู่จะเริ่มต้นที่ 0x
- อินพุตแบบดิสครีต (Discrete Inputs) จะเป็น อินพุตดิจิทัลที่สามารถอ่านได้เท่านั้น ตำแหน่งที่อยู่จะเริ่มต้นที่ 1x
- อินพุต รีจิสเตอร์ (Input Registers) จะเป็น อินพุตอนาล็อกที่สามารถอ่านได้เท่านั้น ตำแหน่งที่อยู่จะเริ่มต้นที่ 3x
- โฮลดิ้ง รีจิสเตอร์ Holding Registers จะเป็น เอาต์พุตอนาล็อกที่สามารถอ่านหรือเขียนได้ ตำแหน่งที่อยู่จะเริ่มต้นที่ 4x

3. การออกแบบบอร์ด CAN2RS485

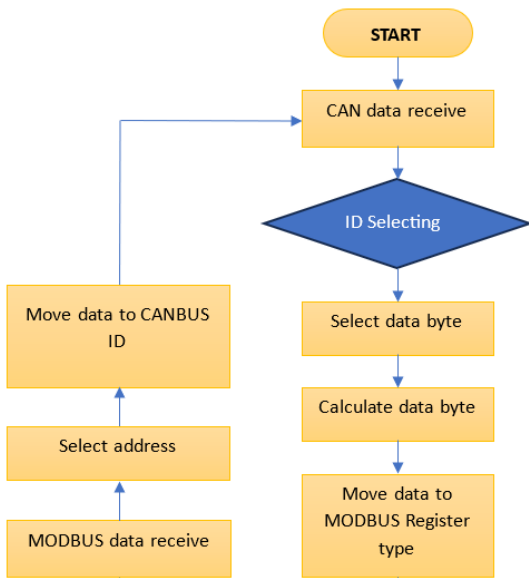
ในการออกแบบบอร์ดแปลงสัญญาณระหว่างแคนบัส กับมอดบัส โดยให้ชื่อว่า CAN2RS485 ซึ่งทำหน้าที่สื่อสารระหว่างตัวรถเอพีเอ็ม กับสถานี โดยการสื่อสารระหว่างตัวรถ กับบอร์ด จะเป็นการสื่อสารแบบแคนบัสภายใต้โพรโตคอล SAE J1939 ส่วนการสื่อสารระหว่างสถานีกับบอร์ด จะเป็นสื่อสารผ่าน RS-485 ภายใต้โพรโตคอล มอดบัส อาร์ทียู

3.1 การออกแบบการทำงาน

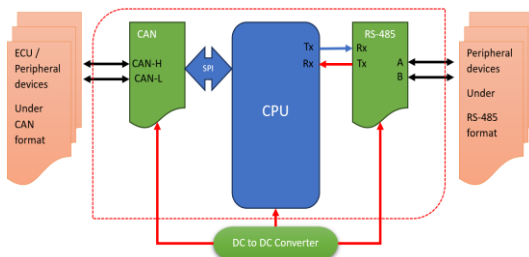
การทำงานของบอร์ด CAN2RS485 ในส่วนของการแปลงสัญญาณแคนบัสให้เป็นมอดบัส โดยจะ



แปลงค่า แคนไอดี และแคนเฟรม ให้อยู่ในรูปแบบ สลาฟ, ที่อยู่, รหัสฟังก์ชัน, ข้อมูล และ ซีอาร์ซี ของ มอดบัส ส่วนในกรณีการแปลงกลับก็จะทำในลักษณะ ตรงกันข้าม โดยแปลง สลาฟ, ที่อยู่, รหัสฟังก์ชัน, ข้อมูล และ ซีอาร์ซี ของมอดบัส ให้อยู่ในรูป แคนไอดี และแคนเฟรม การทำงาน ดังแสดงในรูปที่ 1 และ 2



รูปที่ 1 แผนผังแสดงการทำงาน



รูปที่ 2 โดอะแกรมการทำงานของวงจร

3.2 การออกแบบภาคจ่ายไฟ

การออกแบบภาคจ่ายไฟจะเลือกใช้ DC/DC คอนเวอร์เตอร์ เบอร์ LM2596S ซึ่งเป็นไอซีเรกูเลเตอร์ (Regulator) ที่สามารถรองรับอินพุตระหว่าง 12VDC ถึง 30VDC ให้เป็น 5VDC เพื่อใช้สำหรับเป็นแหล่งจ่าย ไฟฟ้าให้กับวงจรต่างๆ ภายในบอร์ด รายละเอียดวงจร ดังแสดงในรูปที่ 3

3.3 การออกแบบภาคการสื่อสารระหว่าง RS-485 กับ ซีพียู ของบอร์ด CAN2RS485

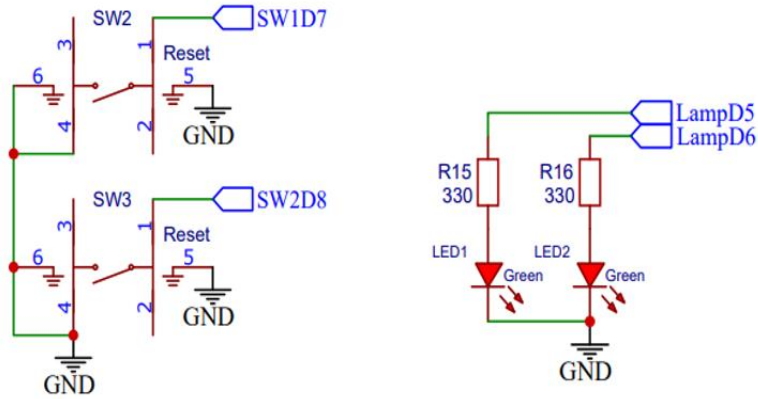
ภาคการสื่อสาร RS-485 ทำงานโดยรับสัญญาณ แบบอนุกรมยูอาร์ที(UART) จากซีพียู ผ่าน Rx และ Tx และใช้ TM74HC04 เพื่อสร้างแรงดันไฟฟ้าให้กับ LED เพื่อแสดงการรับ-ส่งข้อมูล โดยมีตัวต้านทาน R8 ทำหน้าที่ปรับอิมพีแดนซ์ของวงจร ดังแสดงในรูปที่ 4

3.4 แคนบัส กับ ซีพียู ของบอร์ด CAN2RS485

ภาคการสื่อสารแคนบัส ทำงานโดยรับสัญญาณเอส พีไอ (SPI) จากซีพียู ส่งมาที่ MCP2515 เพื่อแปลงเป็น สัญญาณแคนอนุกรม(CAN serial) จากนั้นส่งสัญญาณ ไปที่ MCP2551 เพื่อจัดการรับ/ส่ง โดยมี R11 ทำหน้าที่ ปรับอิมพีแดนซ์ของสายสื่อสาร ดังแสดงในรูปที่ 5

3.5 การออกแบบระบบสำหรับทดสอบการทำงาน

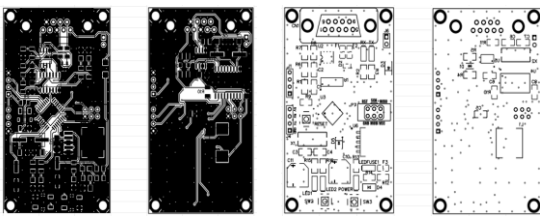
โดยออกแบบอินพุตสวิตช์ จำนวน 2 ตัว เชื่อมต่อกับขา D7 และ D8 ของ ซีพียู และเอาต์พุต แอลอีดี (LED) จำนวน 2 ตัว เชื่อมต่อกับขา D5 และ D6 ของ ซีพียู เพื่อใช้ในการทดสอบการทำงานของบอร์ด ดัง แสดงในรูปที่ 6



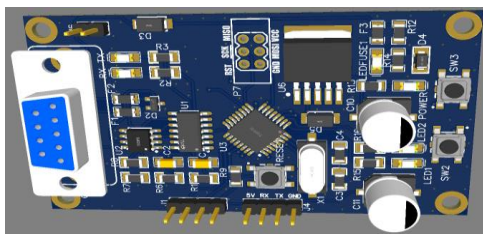
รูปที่ 6 วงจรสวิตช์อินพุต และ แอลอีดี เอาต์พุต สำหรับทดสอบการทำงาน

3.6 การออกแบบบอร์ด CAN2RS485

จากขั้นตอนการออกแบบในหัวข้อที่ 3.2 ถึง 3.5 นำมาออกแบบ พีซีบี (PCB) และตำแหน่งการลงอุปกรณ์ ดังแสดงในรูปที่ 7 และรูปภาพแสดงบอร์ด CAN2RS485 ดังแสดงรูปที่ 8



รูปที่ 7 วงจร พีซีบี และตำแหน่งอุปกรณ์

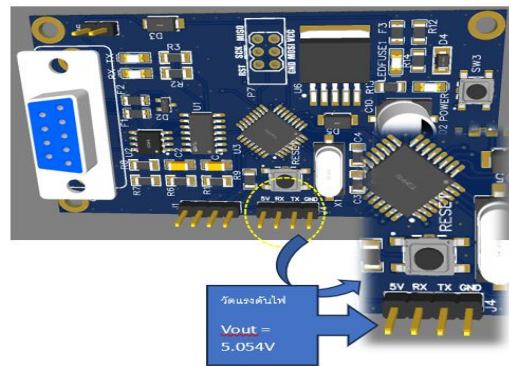


รูปที่ 8 บอร์ด CAN2RS485

4. การทดสอบการทำงาน

4.1 การทดสอบภาคจ่ายไฟ

การทดสอบนี้เป็นการทดสอบย่านการทำงานของอินพุตภาคจ่ายไฟเพื่อให้สามารถรองรับการทำงานของเอพีเอ็ม โดยจ่ายไฟให้กับวงจรระหว่าง 12 - 30 VDC จากนั้นวัดแรงดันเอาต์พุตที่จุดทดสอบได้ค่า 5.054 VDC ซึ่งใกล้เคียงกับค่าที่ออกแบบไว้ ดังแสดงในรูปที่ 9



รูปที่ 9 ผลการทดสอบภาคจ่ายไฟ

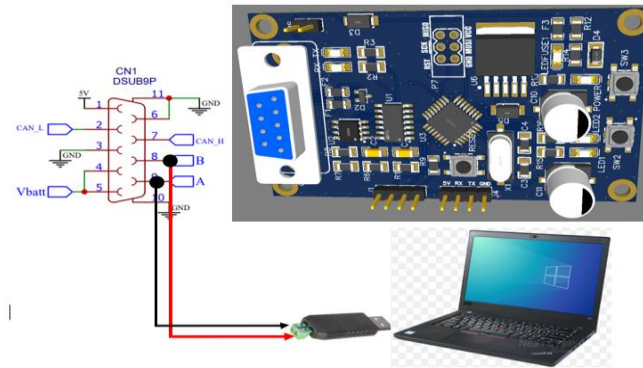


4.2 การทดสอบการสื่อสารด้วยโปรโตคอลมอดบัสอาร์ทียู

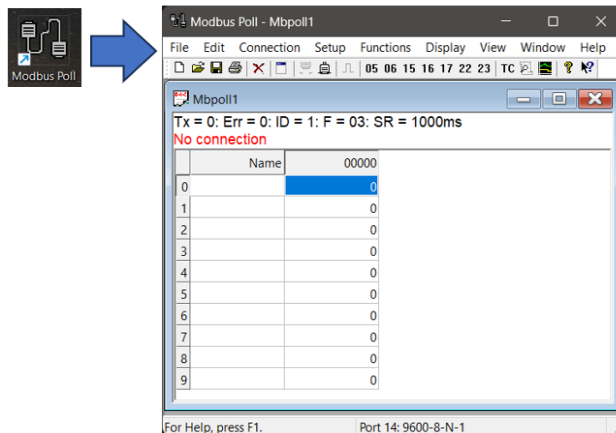
การทดสอบนี้เป็นการทดสอบภาคการสื่อสารด้วยโปรโตคอลมอดบัส โดยจะใช้คอมพิวเตอร์เพื่อทำหน้าที่ในสื่อสารกับบอร์ด CAN2RS485 ภายใต้โปรโตคอลมอดบัสอาร์ทียู โดยจะกำหนดให้คอมพิวเตอร์เป็นมาสเตอร์ และ บอร์ด CAN2RS485 เป็น สลาฟ วงจรการเชื่อมต่อเพื่อใช้ในการทดสอบการทำงาน ดังแสดงในรูปที่ 10 โปรแกรมมอดบัสโพล เพื่อใช้ในการแสดงผลการรับ/ส่งข้อมูล(Modbus Poll) ดังแสดงในรูปที่ 11

4.3 การทดสอบการสื่อสารด้วยโปรโตคอล SAE J1939

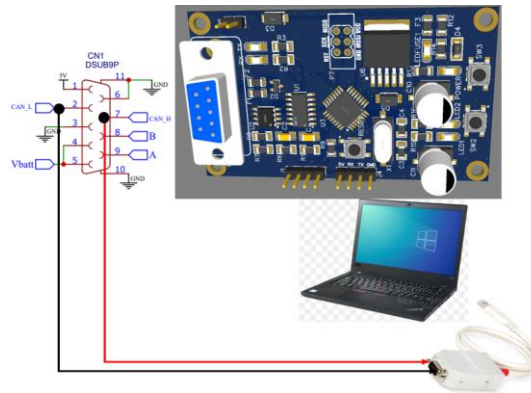
การทดสอบนี้เป็นการทดสอบภาคการสื่อสารแคนบัส ด้วยโปรโตคอล SAE J1939 ซึ่งเป็นส่วนที่จะนำไปใช้ในการสื่อสารภายในตัวรถเอพีเอ็ม โดยในการทดสอบจะใช้คอมพิวเตอร์เป็นตัวกลางทำหน้าที่ในสื่อสารกับบอร์ด CAN2RS485 วงจรการเชื่อมต่อเพื่อใช้ในการทดสอบการทำงาน ดังแสดงในรูปที่ 12 โปรแกรมพีแคนวิว(PCAN View) เพื่อใช้ในการแสดงผลการรับ/ส่งข้อมูล ดังแสดงในรูปที่ 13



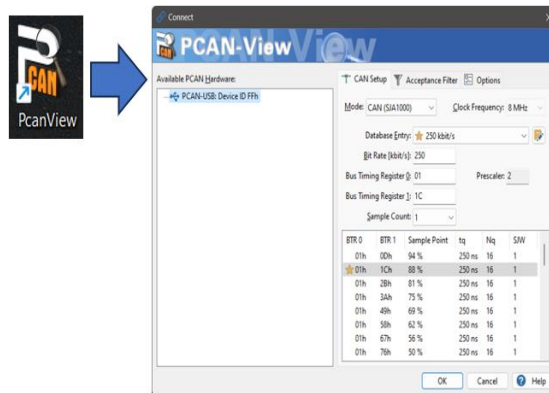
รูปที่ 10 วงจรการทดสอบ มอดบัส อาร์ทียู



รูปที่ 11 หน้าต่างโปรแกรม มอดบัสโพล



รูปที่ 12 วงจรการทดสอบ CAN BUS



รูปที่ 13 หน้าต่างโปรแกรม พีแคนวิว

5. ผลการทดสอบการสื่อสาร

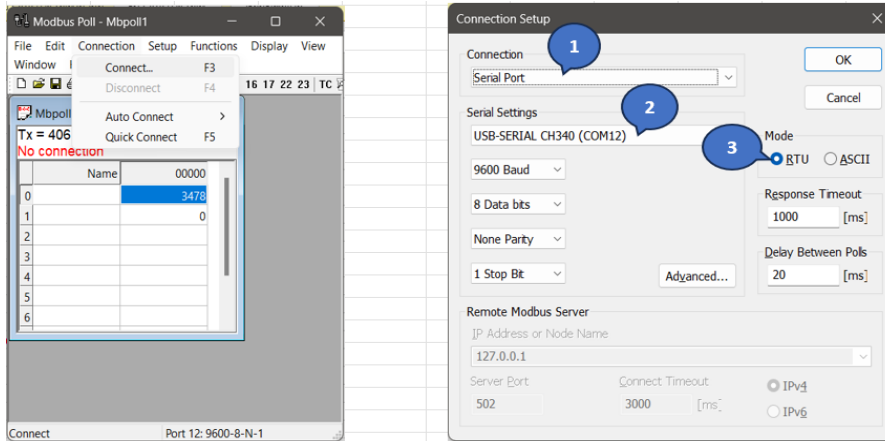
5.1 การทดสอบการสื่อสาร มอดบัส อาร์ทียู

เมื่อเปิดโปรแกรม มอดบัสโพล ตามรูปที่ 11 ซึ่งแสดงหน้าต่างโปรแกรมมอดบัสโพล จากนั้นเมื่อต้องการเชื่อมต่อการสื่อสารให้ดำเนินการตามขั้นตอนดังรูปที่ 14

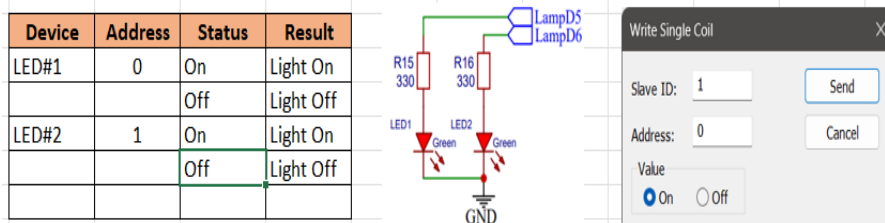
5.1.1 ผลการทดสอบการส่งข้อมูล

การทดสอบนี้จะเป็นการทดสอบการส่งข้อมูลจากโปรโตคอลมอดบัสเพื่อควบคุมการทำงานของหลอดแอลอีดี ที่อยู่ภายในบอร์ด CAN2RS485 โดยการ

ทดสอบจะเริ่มจากการต่อวงจรการเชื่อมต่อตามรูปที่ 10 และตั้งค่าการสื่อสารด้วยโปรแกรมมอดบัสโพล ตามรูปที่ 11 จากนั้นทดสอบการส่งข้อมูลผ่านรีจิสเตอร์คอยล์ (Coils : 0x) ซึ่งเป็นรีจิสเตอร์แบบบิต เพื่อควบคุมการทำงานของหลอดแอลอีดี ผลการทดสอบพบว่าเมื่อข้อมูลบิตถูกส่งจากโปรแกรมมอดบัสโพล ไปยังบอร์ด CAN2RS485 โดยกำหนดให้ไอดี หมายเลข 1 ส่งสัญญาณสั่งงานหลอดแอลอีดี ทั้ง 2 หลอด พบว่า LED1 และ LED2 สามารถทำงานได้ตามข้อมูลที่ส่ง ดังแสดงในรูปที่ 15



รูปที่ 14 แสดงขั้นตอนการเชื่อมต่อระหว่าง
มอดบัสโพล กับคอมพิวเตอร์



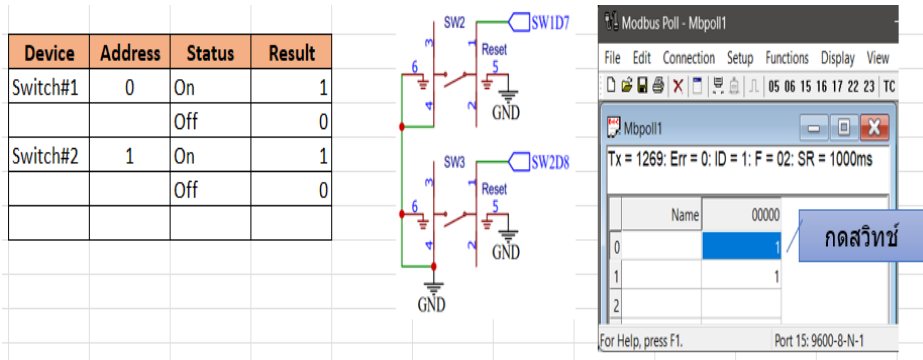
รูปที่ 15 แสดงผลการส่งข้อมูล

5.1.2 ผลการทดสอบการรับข้อมูลอินพุตแบบไม่ต่อเนื่อง

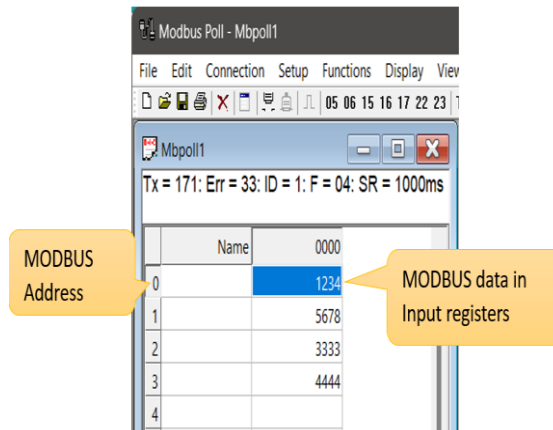
การทดสอบนี้เป็นการส่งค่าสถานะการกดคันสวิทช์ 2 ตัว ผ่านรีจิสเตอร์อ่านอินพุตแบบไม่ต่อเนื่อง (Discrete Inputs : 1x) โดยโปรแกรม มอดบัสโพล สามารถแสดงสถานะของสวิทช์ได้ตามเงื่อนไขการกด ดังแสดงในรูปที่ 16

5.1.3 ผลการทดสอบการรับข้อมูลอินพุตรีจิสเตอร์

การทดสอบการส่งข้อมูล 16 บิต เลขฐานฐานจำนวน 4 ข้อมูล ได้แก่ 04D2 162E 0D05 115C ผ่านบอร์ด CAN2RS485 โดยข้อมูลที่ส่งไปยัง มอดบัส ที่อินพุตรีจิสเตอร์ (Input Registers :3x) ที่อยู่ 0, 1, 2 และ 3 จากนั้นจะใช้โปรแกรม มอดบัสโพล อ่านข้อมูลพบว่าข้อมูลที่ส่งไปแสดงค่าเป็นเลขฐาน 10 ได้ตรงตามค่าที่ส่ง ดังแสดงในรูปที่ 17



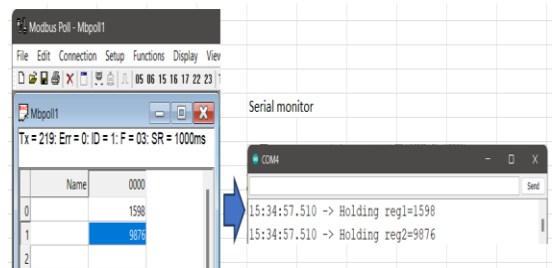
รูปที่ 16 แสดงผลการรับข้อมูลอินพุตแบบไม่ต่อเนื่อง



รูปที่ 17 แสดงผลการรับข้อมูลอินพุตรีจิสเตอร์

5.1.4 ผลการทดสอบการรับข้อมูลโฮลดิ้งรีจิสเตอร์ (Holding Register)

ในการทดสอบนี้จะเป็นการทดสอบการส่งข้อมูลโฮลดิ้งรีจิสเตอร์ (Read Holding Registers: 4x) ขนาด 16 บิต จำนวน 2 ที่อยู่ โดยใช้โปรแกรมมอดบัลโพล ทำหน้าที่ส่งข้อมูลผ่านบอร์ด CAN2RS485 เพื่อนำข้อมูลไปแสดงผลที่ส่วนแสดงผลแบบอนุกรม (Serial Monitor) จากการทดสอบพบว่าข้อมูลที่ส่งจากโฮลดิ้งรีจิสเตอร์ มีค่าตรงกับข้อมูลที่ส่งไปยังส่วนแสดงผลแบบอนุกรม ดังแสดงในรูปที่ 18



รูปที่ 18 แสดงผลการรับข้อมูลโฮลดิ้งรีจิสเตอร์



5.2 การทดสอบการสื่อสารแคนบัส

การทดสอบโดยต่อวงจรตามรูปที่ 12 และเลือกอัตราการส่งข้อมูล (bit rate) ที่ 250 Kbit/s

5.2.1 ผลการทดสอบการส่งข้อมูลแคนเฟรม (Data CAN Frame) ด้วยโปรแกรมพีแคนวี

การทดสอบกำหนดให้ แคนไอดี = 0x18A57F27 และ ข้อมูลแคนเฟรม = 11 22 33 44 55 66 77 88 ตามลำดับ ผลการทดสอบพบว่า แคนไอดี และ ข้อมูลแคนเฟรม สามารถส่งผ่านไปแสดงที่โปรแกรม พีแคนวี ได้ ตรงกับค่าของข้อมูลที่ส่ง ดังแสดงในรูปที่ 19

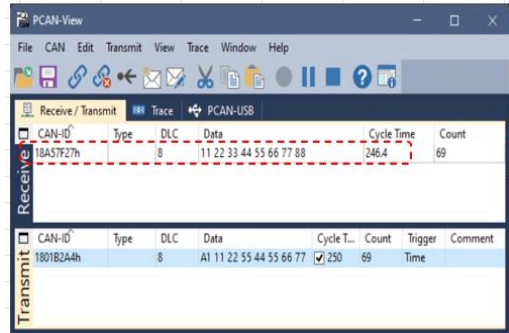
5.2.2 ผลการทดสอบการรับข้อมูลแคนเฟรม ด้วยโปรแกรม พีแคนวี

การทดสอบนี้เป็นการทดสอบการสื่อสารของระบบแคนบัส ของบอร์ด CAN2RS485 โดยกำหนดให้ แคนไอดีเป็น 0x08A57F28 และ ข้อมูลแคนเฟรม = 1A 2B 3C 4D 5E 6F A7 B8 ตามลำดับ เพื่อใช้ในการทดสอบการสื่อสาร โดยผลการทดสอบพบว่า แคนไอดี และข้อมูลแคนเฟรม สามารถส่งผ่านบอร์ด CAN2RS485 ไปแสดงที่ส่วนแสดงผลแบบอนุกรมได้ถูกต้องตรงตามค่าที่ส่ง ดังแสดงในรูปที่ 20

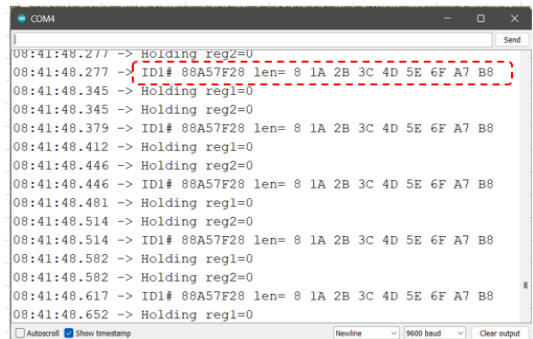
5.3 การทดสอบการสื่อสารระหว่าง แคนบัส และ มอดบัส

5.3.1 ผลการทดสอบการส่งข้อมูลแคนเฟรม ไปแสดงผลที่ โปรแกรม มอดบัสโพล

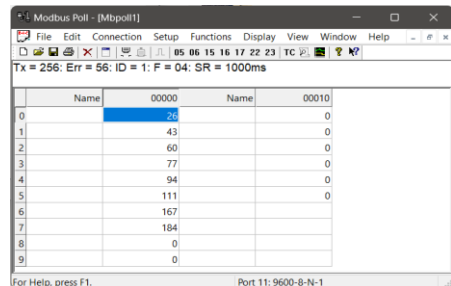
การทดสอบนี้จะทดสอบการนำข้อมูลจากแคนบัสไปแสดงผลที่มอดบัส โดยกำหนดให้ แคนไอดีเป็น 0x08A57F28 และข้อมูลแคนเฟรม = 1A 2B 3C 4D 5E 6F A7 B8 ตามลำดับ และส่งข้อมูลจำนวน 8 ไบต์ไปเก็บที่ อินพุตรีจิสเตอร์ ของมอดบัส โดยจะเริ่มต้นที่



รูปที่ 19 แสดงผลการส่งข้อมูลแคนเฟรม



รูปที่ 20 ผลทดสอบการรับข้อมูลแคนเฟรม



รูปที่ 21 ผลทดสอบการส่งข้อมูลแคนเฟรม ไปแสดงผลที่โปรแกรม มอดบัสโพล

300001 ผลการทดสอบพบว่า แคนไอดี และข้อมูลแคนเฟรม สามารถส่งผ่านไปแสดงที่อินพุตรีจิสเตอร์ ของมอดบัส ได้ตรงตามค่าที่ส่ง ดังแสดงในรูปที่ 21



5.3.2 ทดสอบการส่งค่าโฮลดิ้งรีจิสเตอร์ไปแสดงผลที่โปรแกรม พีแคนวิว

การทดสอบกำหนดให้ โฮลดิ้งรีจิสเตอร์ = $0x1234_{10}$ อินพุตรีจิสเตอร์ของ มอดบัส จะเริ่มต้นที่ 400001 ผลการทดสอบพบว่าข้อมูลโฮลดิ้งรีจิสเตอร์ สามารถส่งผ่านบอร์ด CAN2RS485 ไปแสดงที่โปรแกรม พีแคนวิว ในรูปแบบเลขฐานสิบหก ซึ่งตรงกับข้อมูลที่ส่ง ดังแสดงในรูปที่ 22

5.4 การทดสอบไปใช้งานกับ เอพีเอ็ม

5.4.1 ผลการทดสอบการควบคุมเอพีเอ็มด้วยตัวรถ

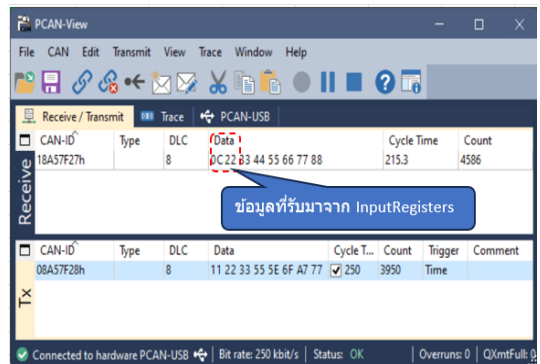
ในการทดสอบนี้จะทดสอบควบคุมการทำงานของเอพีเอ็มผ่านแคนบัส โดยกำหนดให้ แคนไอดีของตัวรถเป็น $0x18A72721$ โดยที่ 18 หมายถึงลำดับความสำคัญ (Priority : P) 3 บิต สำรอง (Reserved : RES) 1 บิต และหน้าของข้อมูล (Data Page : DP) 1 บิต A727 คือส่วนของหมายเลขกลุ่มพารามิเตอร์ โดยแบ่งเป็น A7 เป็นส่วนรูปแบบของหน่วยโปรโตคอลข้อมูล (Protocol Data Unit Format : PF) และ 27 เป็นส่วนของหน่วยโปรโตคอลข้อมูลเฉพาะ (Protocol Data Unit Specific : PS) 21 คือ ที่อยู่ต้นทาง (Source Address : SA) [11]

ในการทดสอบจะใช้สวิทช์สำหรับติดตั้งในตัวรถจำนวน 8 สวิทช์ ในการส่งข้อมูลแคนเฟรมจำนวน 2 ไบต์ โดยกำหนดให้ ไบต์ที่ 1 จะติดต่อกับสวิทช์ 1-4 และไบต์ที่ 2 จะติดต่อกับสวิทช์ 5-8 ซึ่งจะทำหน้าที่ในการควบคุมการทำงานของตัวรถ หลังจากนั้นจะส่งสัญญาณมาที่ บอร์ด CAN2RS485 เพื่อแปลงสัญญาณเป็นมอดบัส ส่งไปที่อินพุตแบบดิสครีต (1X) เพื่อนำข้อมูลสถานะการทำงานไปแสดงผลที่สถานี ผลการ

ทดสอบพบว่า สามารถควบคุม เอพีเอ็มได้ และสามารถส่งสถานะไปแสดงผลที่สถานีได้ ดังแสดงในรูปที่ 23

5.4.2 ผลการทดสอบการควบคุมเอพีเอ็มด้วยสถานี

ในการทดสอบโดยควบคุมผ่านจอทัชสกรีนที่ติดตั้งในสถานี เพื่อควบคุมการทำงานของตัวรถ เอพีเอ็ม ผ่านบอร์ด CAN2RS485 ซึ่งจะแปลงสัญญาณมอดบัส รับมาจากเอาต์พุตคอล์ย (0X) ให้เป็นสัญญาณเป็น แคนบัส โดยจะกำหนด แคนไอดีของบอร์ด CAN2RS485 = $0x18A72717$ เพื่อส่งข้อมูลไปควบคุมการทำงานของตัวรถ ผลการทดสอบพบว่า สามารถควบคุมเอพีเอ็มจากสถานีได้ ดังแสดงในรูปที่ 24



รูปที่ 22 ผลทดสอบการอ่านค่าโฮลดิ้งรีจิสเตอร์แสดงผลที่โปรแกรม พีแคนวิว



รูปที่ 23 ผลทดสอบการควบคุม เอพีเอ็ม ด้วยตัวรถ



รูปที่ 24 ผลทดสอบการควบคุมเอพีเอ็มด้วยสถานี

6. สรุปผลการดำเนินงานและข้อเสนอแนะ

6.1 สรุปผลการดำเนินงาน

งานวิจัยนี้มุ่งเน้นการพัฒนาบอร์ดแปลงสัญญาณ CAN2RS485 สำหรับใช้ในระบบรถขนส่งผู้โดยสารอัตโนมัติ หรือ เอพีเอ็ม ซึ่งเป็นระบบขนส่งอัตโนมัติที่นิยมใช้ในพื้นที่ขนาดเล็ก การสื่อสารภายในตัวรถเอพีเอ็มใช้โปรโตคอลแคนบัส ขณะที่สถานีควบคุมใช้โปรโตคอลมอดบัส ซึ่งจำเป็นต้องมีบอร์ดเชื่อมต่อเพื่อแปลงสัญญาณระหว่างสองโปรโตคอลนี้ บอร์ดที่พัฒนาขึ้นนี้รองรับการสื่อสารตามมาตรฐาน SAE J1939 และมอดบัสอาร์ทียู โดยสามารถกำหนดแคนไอดี และการตั้งค่ารีจิสเตอร์ของมอดบัสได้ตามต้องการ นอกจากนี้ บอร์ด CAN2RS485 ยังสามารถทำงานในโหมดมาสเตอร์หรือสลาฟได้ ผลการทดสอบพบว่า บอร์ดสามารถแปลงสัญญาณจากแคนบัสเป็นมอดบัส โดยสามารถแสดงผลการทำงานของเอพีเอ็มที่สถานีควบคุมได้ และสามารถรับข้อมูลจากมอดบัสแปลงกลับเป็นแคนบัส เพื่อควบคุมการทำงานของเอพีเอ็มได้ โดยการสื่อสารระหว่างตัวรถเอพีเอ็มและสถานีควบคุมสามารถทำงานได้อย่างถูกต้อง

6.2 ข้อเสนอแนะ

การพัฒนาบอร์ดนี้มีศักยภาพในการนำไปประยุกต์ใช้กับระบบขนส่งอื่น ๆ เช่น ระบบรถไฟฟ้าหรือระบบขนส่งมวลชนที่ต้องการการเชื่อมต่อระหว่างโปรโตคอลที่แตกต่างกัน โดยสามารถปรับแต่งเพื่อเพิ่มความยืดหยุ่นในการทำงานได้ แม้ว่าบอร์ดนี้จะทำงานได้ดีในการทดสอบ แต่หากนำไปใช้ในระบบเอพีเอ็มขนาดใหญ่ หรือระบบที่มีปริมาณข้อมูลการสื่อสารมาก จำเป็นต้องพิจารณาเรื่องความเร็วในการรับส่งข้อมูล รวมถึงการจัดลำดับความสำคัญของแคนไอดี เพื่อความปลอดภัยในการทำงานด้วย นอกจากนี้ ควรมีการตรวจสอบเพิ่มเติมเกี่ยวกับความเข้ากันได้กับอุปกรณ์อื่น ๆ ที่เชื่อมต่อกับระบบ เพื่อให้สามารถรองรับการทำงานในสภาพแวดล้อมจริงได้อย่างราบรื่น

7. เอกสารอ้างอิง

- [1] S.D. Srivastava and R. Agrawal, Automated people movers: A futuristic approach to modern transportation planning, *IOSR Journal of Mechanical and Civil Engineering*, 2014, 11(3), 01-11.
- [2] M. van Doorne, G. Lodewijks, and W. Beelaerts van Blokland, Adapting automated people mover capacity on airports to real time demand, in M. Dressner, Ed., *Proc. 23rd World Conf. Air Transport Research Society (ATRS 2019)*, Proceeding, 2019, 45-52.
- [3] American Society of Civil Engineers, *Automated people mover standards*, ASCE Standard, Virginia, USA, 2024.



- [4] E.A. Perondi, J.R.F. Bertoldo, and V.H.C. de Albuquerque, Modeling and simulating the controller behavior of an automated people mover using IEC 61850 communication requirements, *IEEE Transactions on Industrial Electronics*, 2024, 70(1), 123-130.
- [5] B. Khemkladmuk, T. Kulworawanichpong, and S. Wongfookeat, Design and prototype development of traffic control systems for automated passenger transport, *The Journal of King Mongkut's University of Technology North Bangkok*, 2022, 32(3), 1-17. (in Thai)
- [6] Q. Yuan, B. Zhang, and Q. Yuan, Design of a CAN-485 conversion circuit, 10th International Conference on Applied Science, Engineering and Technology (ICASET 2021), *Proceeding*, 2021, 120-125.
- [7] SAE J1939-21: Data link layer, SAE International, Warrendale, PA, USA, 2013.
- [8] <https://www.csselectronics.com/pages/j1939-explained-simple-intro-tutorial> (Accessed on 1 September 2024).