

THIẾT KẾ THIẾT BỊ ĐỌC DỮ LIỆU ĐỘNG PHỤC VỤ CHẨN ĐOÁN KỸ THUẬT QUA CỔNG OBD-II

Vũ Ngọc Tuấn¹, Nguyễn Đình Dũng², Trịnh Ngọc Hùng^{1,*}

¹Viện Cơ khí động lực, Trường Đại học Kỹ thuật Lê Quý Đôn

²Khoa Hàng không - Vũ trụ, Trường Đại học Kỹ thuật Lê Quý Đôn

DOI: 10.56651/lqdtu.jst.v18.n02.687

Tóm tắt

Bài báo trình bày phương pháp thiết kế mạch đọc dữ liệu động của xe thông qua cổng giao tiếp OBD-II trên cơ sở mạch Can Bus Shield và Arduino Uno. Mạch giao tiếp này cho phép giám sát mọi thông số của xe qua các cảm biến theo thời gian thực. Đồng thời, phương pháp chẩn đoán theo ngưỡng tín hiệu của các thông số nêu trên cũng được áp dụng nhằm giám sát và chẩn đoán tình trạng kỹ thuật của ô tô. Thông qua kết quả thử nghiệm thực tế, thiết bị giám sát dữ liệu động theo thời gian thực và phương pháp chẩn đoán có thể được ứng dụng cho các loại xe có cổng giao tiếp OBD-II khác nhau. Đồng thời, giám sát số lượng lớn xe đang hoạt động giúp cho nhà quản lý có kế hoạch khai thác, bảo dưỡng và chọn chế độ vận hành hợp lý cho phương tiện cũng là hướng phát triển chính tiếp theo của nghiên cứu này. Kết quả thời gian khảo sát của thiết bị đọc dữ liệu động là 0,02 s, nhanh hơn (50 lần) so với máy chẩn đoán là 0,1 s. Giá thành sản phẩm rẻ, nhỏ gọn hơn 60% so với máy chẩn đoán của hãng theo xe.

Từ khóa: Can Bus Shield; OBD-II; công cụ chẩn đoán; Arduino Uno.

1. Đặt vấn đề

Việc lấy dữ liệu động từ các cảm biến trên xe ô tô theo thời gian thực giúp cho người sử dụng hoặc kỹ sư có thể phân tích, theo dõi và kiểm soát hiệu suất của các bộ phận trên xe ô tô như động cơ, hệ thống phanh, hệ thống treo, điều hòa không khí,... Điều này giúp nghiên cứu và phân tích tình trạng hệ thống và cải thiện hiệu suất, giảm lượng khí thải ô nhiễm ra môi trường, định vị vị trí xe gặp sự cố, quản lý thống kê hoạt động nhiều xe nhằm tối ưu việc bảo trì và sửa chữa xe ô tô đúng thời điểm giúp giảm thiểu tối đa chi phí vận hành,...

Trên thế giới, một số công trình nghiên cứu liên quan đến việc quan sát, phân tích các tham số của ô tô theo thời gian thực đã được công bố. Gilman đã chỉ ra phương pháp để tăng hiệu quả tiêu hao nhiên liệu bằng cách quan sát và phân tích các tham số trên xe theo thời gian thực [1]. Mặt khác, Szalay và cộng sự đã sử dụng giao thức CAN

* Email: trinhhung987@gmail.com

(Controller Area Network) và FMS CAN Bus để đọc các thông số theo thời gian thực và kết luận rằng hai phương pháp này cho kết quả tương tự nhau [2]. Thông qua cổng giao tiếp OBD, Kushiro và cộng sự đã phân tích các mã lỗi và mối tương quan giữa chúng để xây dựng mô hình chẩn đoán các hỏng hóc tiềm ẩn và có thể ngăn chặn được các sự cố có thể xảy ra [3]. Sik đã thiết kế một mô hình dựa trên mạch CAN, OBD kết hợp với GPS để giúp người lái xe lựa chọn tuyến đường tối ưu hoặc tìm chỗ đậu xe [4]. Đối với loại xe hybrid, vừa sử dụng động cơ điện vừa sử dụng động cơ đốt trong, tác giả D'Agostino đã phân tích dữ liệu trên xe thông qua cổng OBD để xác định chế độ hoạt động cho từng loại động cơ. Kết quả là sự tiêu hao nhiên liệu và khí thải đã giảm đi một cách đáng kể [5, 6]. Ngày nay, cùng với sự phát triển của khoa học công nghệ, các nghiên cứu về giám sát và điều khiển từ xa các thông số của xe theo thời gian thực đã và đang được thực hiện và có chiều hướng tăng lên [7-9]. Tại Việt Nam, việc dựa vào các thông số dữ liệu động phục vụ chẩn đoán trên xe qua cổng OBD-II hầu như chưa có nghiên cứu. Tuy nhiên, có một số nghiên cứu liên quan đến vấn đề này đã được công bố. Nguyễn Kim và cộng sự đã thực hiện thành công đề tài “Thiết kế và thi công mô hình triển khai hệ thống điều khiển động cơ diesel điện tử common rail”. Trong đó, tác giả Nguyễn Kim và cộng sự đã sử dụng mạch Arduino Uno để thu thập dữ liệu và phần mềm LabVIEW để hiển thị giá trị các tham số đo được [10, 11].

Việc lấy dữ liệu động từ các cảm biến trên xe ô tô gặp một số khó khăn như: (i) Cần kinh nghiệm về các cảm biến trên xe ô tô và nguyên lý làm việc; (ii) Dữ liệu hiển thị từ các cảm biến trên xe ô tô thường khá phức tạp và khó hiển thị phải dùng thuật toán lập trình giúp hiển thị thông tin cần biết; (iii) Thiết bị thu nhận dữ liệu và phần mềm xử lý dữ liệu có giá thành khá cao, chỉ áp dụng cho một dòng xe, một hãng xe nhất định. Đồng thời mã code lập trình là bí mật công nghệ của từng hãng; (iv) Thường xuyên phải cập nhật phần mềm chẩn đoán cho từng xe, giá thành cao, khi hư hỏng khó sửa chữa; (v) Khi chẩn đoán kỹ thuật, máy chẩn đoán không cho ra các đề thị tổng quát chuyên sâu để phân tích đánh giá kỹ hơn tình trạng kỹ thuật của ô tô.

Cụ thể, trong nghiên cứu này, nhóm tác giả trình bày phương pháp thiết kế thiết bị để thu thập các thông số trên xe theo thời gian thực dựa trên mạch Can Bus Shield và Arduino Uno. Việc phân tích các mã PID để ghi nhận và xử lý số liệu được xây dựng và lập trình trên phần mềm Arduino IDE. Dựa trên các tham số đo được theo thời gian thực, nhóm tác giả đã áp dụng phương pháp chẩn đoán theo ngưỡng để đưa ra các kết luận thông qua phần mềm LabVIEW giúp hiển thị các lỗi cảnh báo của hệ thống. Thiết bị này cho phép thu thập, quan sát, ghi nhận và phân tích giá trị các tham số trên xe ô tô

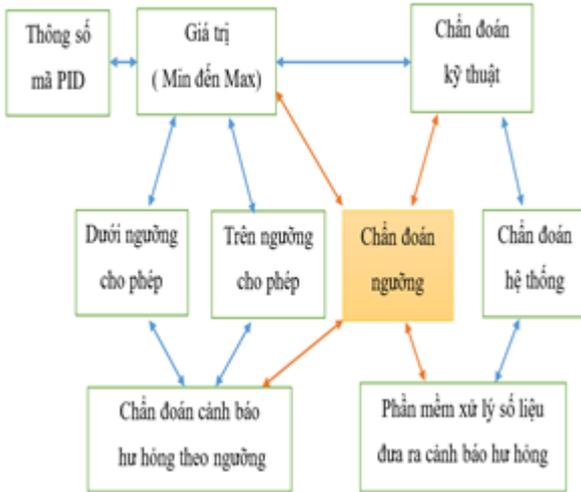
theo thời gian thực, phục vụ công tác chẩn đoán kỹ thuật, phát hiện các lỗi kỹ thuật và ngăn ngừa các sự cố có thể xảy ra. Việc thiết kế thành công thiết bị đọc dữ liệu động và thuật toán chẩn đoán có thể sử dụng được cho nhiều dòng xe, với giá thành rẻ giúp làm chủ được công nghệ giám sát, phân tích hư hỏng nhằm hạn chế rủi ro về mặt kỹ thuật và cải thiện hiệu suất của xe một cách độc lập không cần dùng máy chẩn đoán chuyên dụng.

2. Nguyên tắc chẩn đoán thông qua ngưỡng dữ liệu

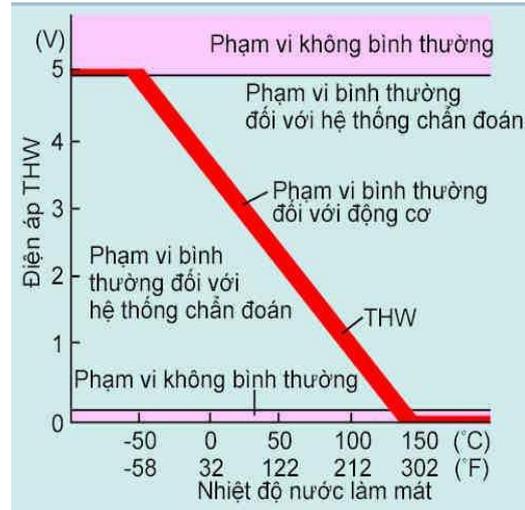
Có rất nhiều phương pháp khác nhau để chẩn đoán tình trạng kỹ thuật của các cụm và hệ thống trên xe. Dựa trên bộ tham số kết cấu, tham số chẩn đoán, các hệ thống đo lường và các lý thuyết nhằm xác định bộ các thông số chẩn đoán để xây dựng các công cụ chẩn đoán khác nhau. Đối với các máy chẩn đoán hiện tại, thông thường việc chẩn đoán và hiển thị các mã lỗi được thực hiện qua việc so sánh ngưỡng giá trị của các tín hiệu đo được thông qua các cảm biến. Tuy nhiên, phần mã nguồn của các thuật toán kiểm tra nêu trên thường được bảo mật bởi các hãng thiết kế chế tạo thiết bị chẩn đoán. Chính vì lý do đó, trong nghiên cứu này nhóm tác giả sử dụng nguyên tắc chẩn đoán kỹ thuật theo giá trị ngưỡng của các cảm biến trên xe làm nguyên tắc chính để hình thành thuật toán chẩn đoán. Tuy nhiên, các giá trị này được ghi nhận và so sánh theo thời gian thực thông qua bộ thiết bị tự thiết kế chế tạo. Các giá trị đo của các cảm biến trên xe bằng các mã PID và công thức tính toán của từng cảm biến. Thông qua phần mềm thu thập, xử lý số liệu, hiển thị dữ liệu động so sánh với một giá trị ngưỡng đã được xác định trước đó trong khoảng min tới max. Từ đó đưa ra kết luận chẩn đoán và cảnh báo về tình trạng hoạt động của các cụm và hệ thống trên xe. Hiệu quả của phương pháp này nhanh chóng, chính xác cao và dễ sử dụng không cần máy chẩn đoán công nghiệp. Đồng thời có thể làm chủ để phát triển hệ thống giám sát và chẩn đoán dữ liệu theo thời gian thực cho nhiều xe cùng một lúc, qua đó có những chính sách khai thác, bảo dưỡng, sử dụng và quản lý hợp lý phương tiện.

Khi giá trị đo của một cảm biến vượt quá giá trị ngưỡng hoặc không đạt được giá trị ngưỡng, hệ thống sẽ hiển thị một cảnh báo để cho người lái biết rằng hệ thống đang gặp vấn đề và cần được kiểm tra hoặc sửa chữa (Hình 1). Việc chẩn đoán kỹ thuật theo giá trị ngưỡng của các cảm biến trên xe là rất quan trọng để đảm bảo an toàn và nâng cao hiệu suất cho xe trong quá trình vận hành.

Dựa vào đồ thị ngưỡng chẩn đoán nếu cảm biến bị ngắn mạch (điện áp đưa vào nhỏ hơn 0,1 V) hoặc đứt dây (điện áp đầu vào lớn hơn 4,8 V). Nhiệt độ nước làm mát cao vượt ngưỡng cho phép trên 150°C như hình 2.



Hình 1. Sơ đồ nguyên tắc chẩn đoán theo ngưỡng giá trị.



Hình 2. Đồ thị chẩn đoán theo ngưỡng của nhiệt độ nước làm mát.

3. Thiết kế thiết bị đọc dữ liệu động

3.1. Phương thức đọc dữ liệu từ cảm biến

Dựa vào các công thức tính toán trong bảng mã PID OBD-II [12], tác giả sử dụng ngôn ngữ lập trình phần mềm Arduino để tính toán ra các thông số và thu nhận các bộ dữ liệu động từ các cảm biến trên xe ô tô như sau:

- Nhiệt độ nước làm mát động cơ (N_{dc}) mã PID 05 được tính theo công thức như sau:

$$N_{dc} = A - 40 \text{ [có giá trị ngưỡng từ } -40 \text{ đến } 215^{\circ}\text{C]} \quad (1)$$

trong đó, A - giá trị byte thứ 3.

- Tốc độ động cơ (V_{dc}) mã PID 0C được tính theo công thức như sau:

$$V_{dc} = \frac{256A + B}{4} \text{ [có giá trị ngưỡng từ } 0 \text{ đến } 16.383 \text{ vòng/phút]} \quad (2)$$

trong đó: A - giá trị byte thứ 3; B - giá trị byte thứ 4.

- Cảm biến lưu lượng không khí MAF (V_{maf}) mã PID 10 được tính theo công thức như sau:

$$V_{maf} = \frac{256A + B}{100} \text{ [có giá trị ngưỡng từ } 0 \text{ đến } 655,35 \text{ g/s]} \quad (3)$$

trong đó: A - giá trị byte thứ 3; B - giá trị byte thứ 4.

- Nhiệt độ không khí nạp (N_{kn}) mã PID OF được tính theo công thức như sau:

$$N_{kn} = A - 40 \text{ [có giá trị ngưỡng từ } -40 \text{ đến } 215^{\circ}\text{C]} \quad (4)$$

trong đó: A - giá trị byte thứ 3.

- Vị trí bướm ga (V_{bg}) mã PID 11 được tính theo công thức như sau:

$$V_{bg} = \frac{100}{255} A \text{ [có giá trị ngưỡng từ 0 đến 100%]} \quad (5)$$

trong đó: A - giá trị byte thứ 3.

- Cảm biến Oxy (C_{ox}) mã PID 14 được tính theo công thức như sau:

$$C_{ox} = \frac{A}{255} \text{ [có giá trị ngưỡng từ 0 đến 1.275 V]} \quad (6)$$

trong đó: A - giá trị byte thứ 3.

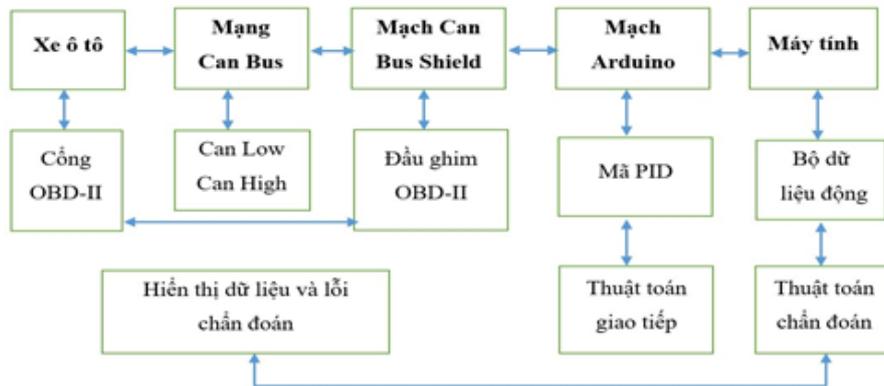
3.2. Lựa chọn tính toán thiết bị

Trên thị trường hiện tại có rất nhiều mạch, thiết bị thu thập dữ liệu động từ mạch kết nối với hệ thống OBD-II như ESP32, ESP8266, ELM327, Can Bus Shield, Arduino Uno và các máy chẩn đoán lỗi như Autel, Fcar, Otofex. Đa phần các mạch, thiết bị và máy chẩn đoán trên đều sử dụng ngôn ngữ bí mật của nhà sản xuất, được sử dụng cho một hãng xe nhất định, hạn chế quá trình khai thác và sử dụng.

Trong nghiên cứu thiết kế lấy dữ liệu động, nhận thấy rằng việc sử dụng mạch Can Bus Shield và Arduino lấy dữ liệu động từ hệ thống OBD-II (Hình 3) rất tiện lợi, linh hoạt, giá thành thấp, độ tin cậy cao... Tính linh hoạt của mạch Can Bus Shield có thể nối với các thiết bị khác nhau thông qua chân cắm đầu nối và cổng kết nối, giúp cho việc kết nối với hệ thống OBD-II dễ dàng hơn. Giá thành của mạch Arduino và Can Bus Shield là những sản phẩm có giá thành rẻ, do đó có thể tiết kiệm được chi phí khi sử dụng để kết nối với hệ thống OBD-II. Độ tin cậy chính xác cao khi kết nối với hệ thống OBD-II vì mạch Arduino và Can Bus Shield sử dụng giao thức CAN để truyền thông tin, đây được coi là giao thức tốt nhất để truyền thông giữa các điểm cuối, đảm bảo tín hiệu truyền đi được đáng tin cậy và giảm thiểu lỗi. Mạch Arduino là một nền tảng phát triển mạch rất dễ sử dụng và có nhiều tài liệu hướng dẫn cho người dùng, giúp họ dễ dàng tiến hành các thao tác cài đặt và vận hành. Ngoài ra, Can Bus Shield cũng có tính tương thích cao với các thư viện phần mềm của hệ điều hành Arduino, giúp người dùng tiết kiệm được thời gian và tăng hiệu quả công việc.

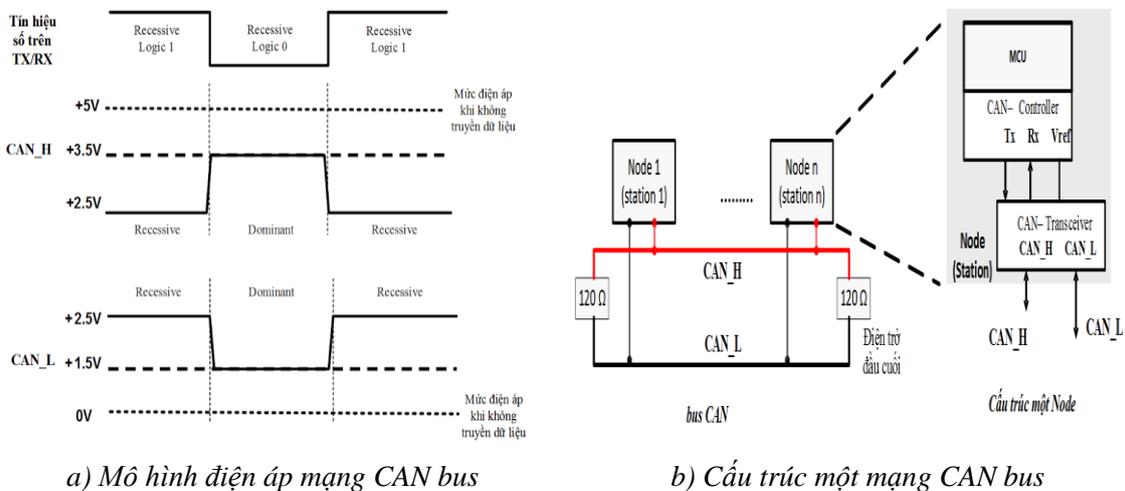
Thiết bị đọc dữ liệu động thu nhận tín hiệu từ các cảm biến trên xe ô tô thông qua cổng OBD-II. Nhờ có mạch Can Bus Shield giao tiếp với mạng Can Bus trên ô tô và xử lý tín hiệu truyền về mạch Arduino là nơi trung tâm nhận các tín hiệu gửi về máy tính để theo dõi, giám sát đánh giá bộ dữ liệu động phục vụ quá trình chẩn đoán kỹ thuật. Thiết bị đọc dữ liệu động được phát triển được chia thành hai phần: Giai đoạn thu thập dữ liệu, thiết kế giao diện đồ họa để trực quan hóa và theo dõi một số thông số dữ liệu động thông qua phần mềm hiển thị phục vụ chẩn đoán kỹ thuật.

Việc sử dụng mạch Can Bus Shield và Arduino để lấy dữ liệu động trên xe ô tô từ các cảm biến có nhiều ứng dụng hữu ích như: Theo dõi và giám sát trạng thái hoạt động của các bộ phận trên xe ô tô như động cơ, hộp số, hệ thống điện, hệ thống phanh, hệ thống lái... giúp người sử dụng dễ dàng phát hiện và phân tích được các vấn đề về an toàn, bảo trì và sửa chữa của xe. Cung cấp thông tin về hiệu suất và tiết kiệm năng lượng của xe ô tô, giúp người sử dụng có thể dễ dàng tối ưu hóa chi phí vận hành. Tạo ra các ứng dụng và sản phẩm liên quan đến xe ô tô như hệ thống định vị GPS, thiết bị đo lường tiêu thụ nhiên liệu. Mạch hoạt động ổn định và chính xác trong thời gian dài, giúp người sử dụng dễ dàng nâng cao khả năng giám sát và quản lý các thông tin liên quan đến xe.

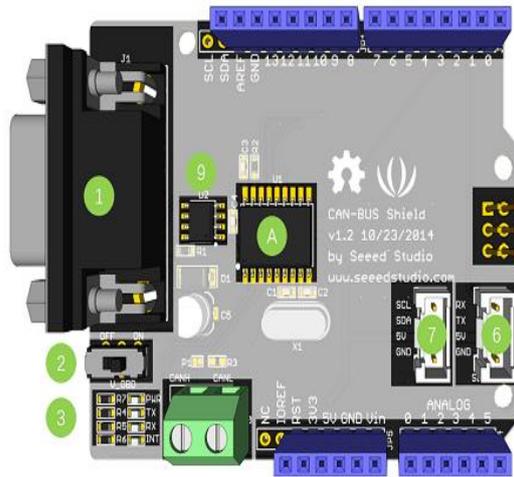


Hình 3. Sơ đồ nguyên tắc giao tiếp dữ liệu động.

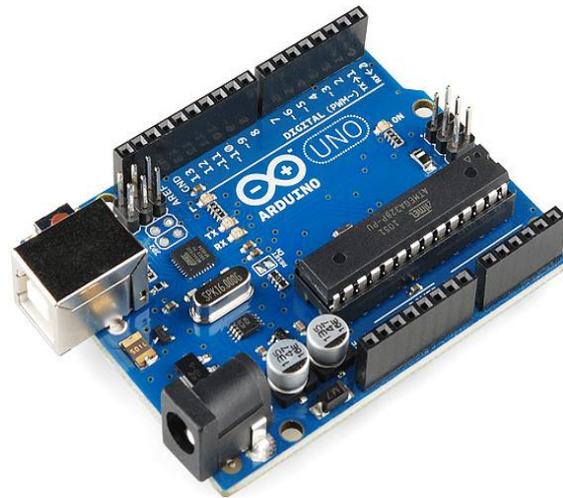
Trong xe ô tô cũng giống như một cơ thể người, mạng giao tiếp CAN là công nghệ mạng nối tiếp có tốc độ cao có hai dây. Cấu trúc mô hình điện áp của mạng CAN Bus như hình 4a và cấu trúc của một mạng CAN Bus như hình 4b.



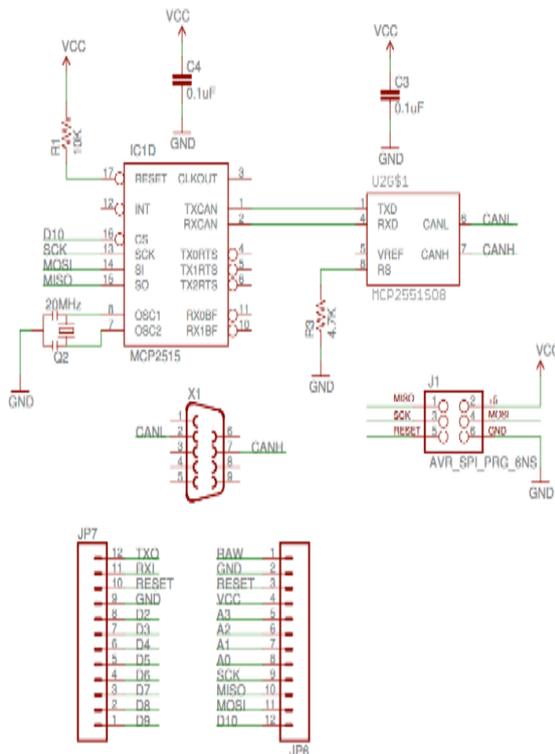
Hình 4. Cấu trúc mạng CAN.



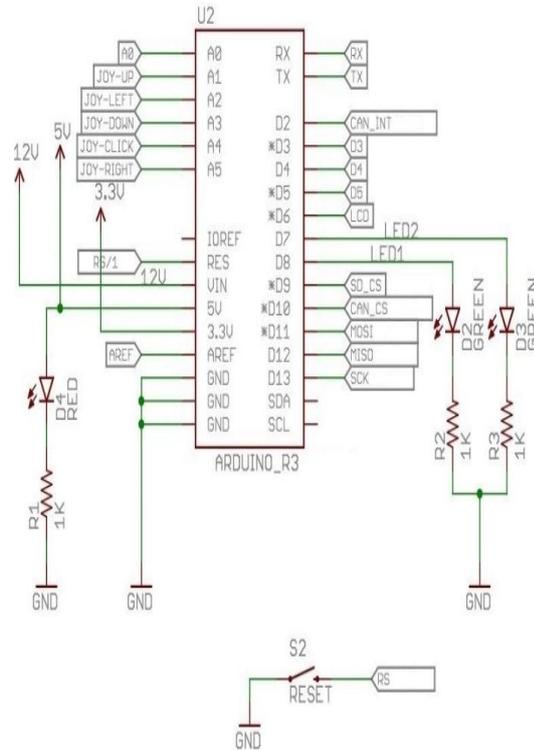
a) Mạch Can Bus Shield



b) Mạch Arduino Uno



c) Sơ đồ mạch Can Bus Shield



d) Sơ đồ mạch Arduino Uno

Hình 5. Sơ đồ cấu tạo, sơ đồ mạch Can Bus Shield và mạch Arduino Uno.

Kết nối mạch Can Bus Shield với mạch Arduino Uno để lấy dữ liệu động thông qua các chân như sau:

- Chân GND của mạch Can Bus Shield với chân GND của mạch Arduino Uno.

- Chân VCC của mạch Can Bus Shield với chân 5V của mạch Arduino Uno.
- Chân CS (Chip Select) của mạch Can Bus Shield với chân 10 của mạch Arduino Uno.
- Chân MOSI (Master Out Slave In) của mạch Can Bus Shield với chân 11 của mạch Arduino Uno.
- Chân MISO (Master In Slave Out) của mạch Can Bus Shield với chân 12 của mạch Arduino Uno.
- Chân SCK (Serial Clock) của mạch Can Bus Shield với chân 13 của mạch Arduino Uno.

Trong mạch Can Bus Shield chứa hai bộ điều khiển MCP2551 và MCP2515 là hai thành phần chính của mạch điều khiển CAN Bus hoạt động như sau:

Mạch Can Bus Shield có bộ điều khiển MCP2551 xử lý tín hiệu CAN Bus (Hình 5c) trên xe ô tô thông qua đầu ghim OBD-II giao tiếp với hai dây mạng CAN Low và CAN High.

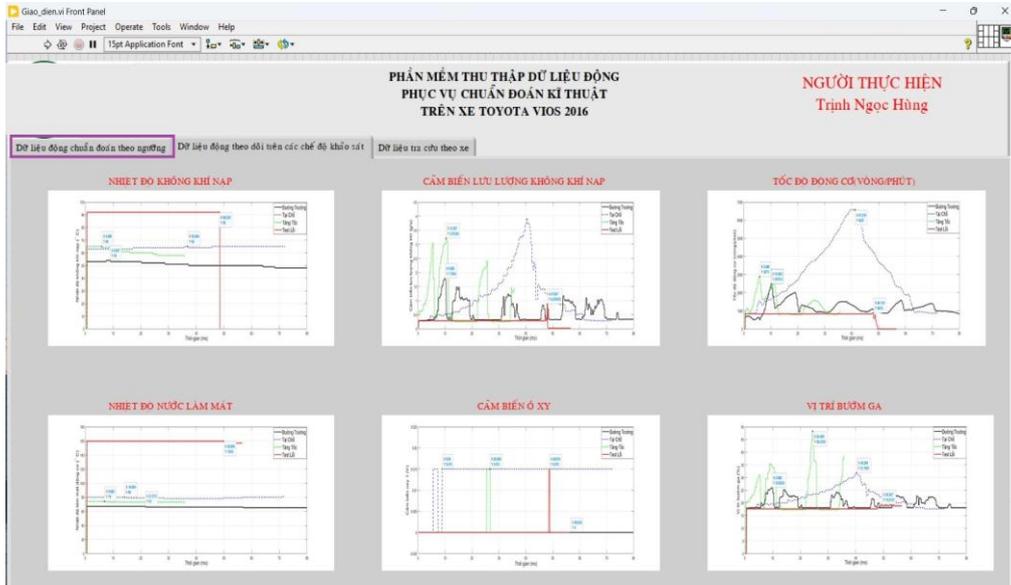
- Bộ điều khiển MCP2551: Là bộ khuếch đại tín hiệu cho mạng CAN Bus. Nhiệm vụ của bộ điều khiển MCP2551 là tạo ra mức tín hiệu đủ lớn cho phép dữ liệu trên mạng CAN Bus có thể được truyền đi xa hơn, với độ chính xác cao. Bộ điều khiển MCP2551 sẽ nhận tín hiệu dữ liệu từ mạng CAN Bus thông qua chân TX hoặc RX. Sau đó, khuếch đại tín hiệu mạng CAN Bus ở mức độ tương thích để có thể truyền đi. Ngoài ra, còn có chức năng giảm nhiễu tín hiệu bằng cách loại bỏ các nhiễu tạp âm có thể xuất hiện trên tín hiệu mạng CAN.

Tín hiệu CAN Bus thông qua hai chân TXD và RXD nhận về mạch Arduino theo giao thức SPI như hình 5d là mạch trung tâm chịu trách nhiệm xử lý các tín hiệu kết nối với PC thông qua phần mềm hiển thị. Bộ điều khiển MCP2515 có nhiệm vụ và nhận tín hiệu trên mạng CAN Bus và truyền tín hiệu được sử dụng để xử lý các tín hiệu CAN nhận và điều khiển truyền tín hiệu CAN Bus đến mạch Arduino Uno xử lý trung tâm. Đầu tiên các tín hiệu dữ liệu được đưa vào MCP2515 thông qua các chân nối đến mạng CAN Bus. Thiết bị đọc dữ liệu động được khảo sát thí nghiệm trên xe Toyota Vios 2016 (Hình 6).

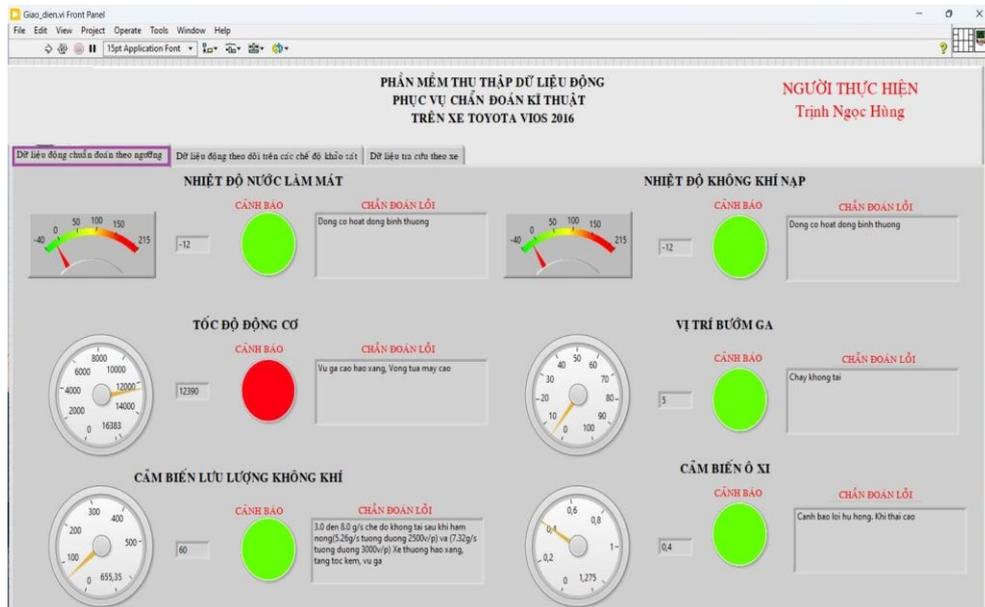


Hình 6. Thiết bị đọc dữ liệu động thực tế khi khảo sát trên xe Toyota Vios 2016.

3.3. Thiết kế phần mềm ghi nhận, xử lý số liệu, chẩn đoán theo ngưỡng dữ liệu động của các cảm biến được giám sát phân tích



Hình 7. Phần mềm thu thập dữ liệu động đưa ra đồ thị tổng quát.



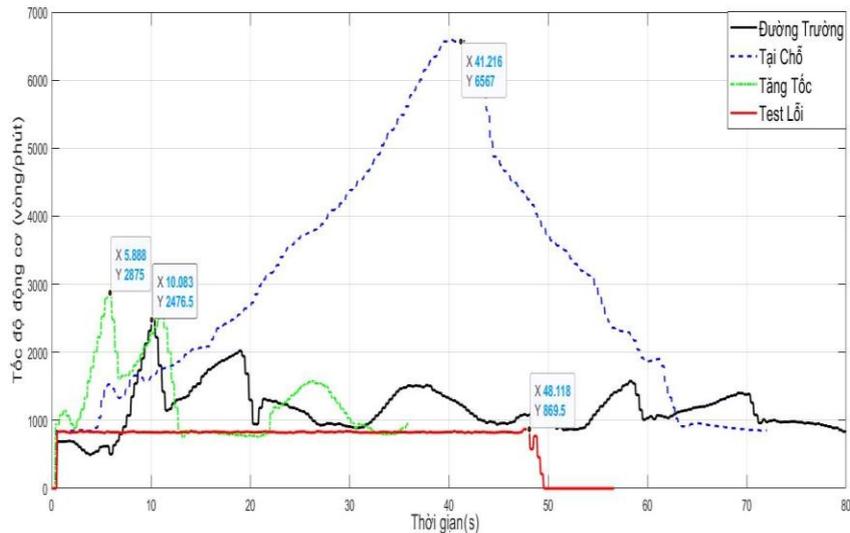
Hình 8. Phần mềm thu thập dữ liệu động đưa ra các cảnh báo.

4. Kết quả thử nghiệm và bàn luận

Nghiên cứu thông qua xe Toyota Vios 2016 là đối tượng để truy xuất dữ liệu từ ECU qua cổng truyền OBD-II trên phương tiện. Các giá trị đã được truy xuất khi lái xe

trên một tuyến đường dài 2 km đường dân sinh và 2 km đường cao tốc với các chế độ khảo sát khác nhau như hình 10 đồ thị tổng quát của cảm biến tốc độ động cơ. Nhiệt độ môi trường là 28°C lúc 9 giờ sáng và toàn bộ dự án kéo dài 10 phút hoặc 600 s.

Khi test lỗi xe bị hỏng cảm biến lưu lượng không khí và cảm biến nhiệt độ không khí nạp. Khi các cảm biến bị lỗi thì động cơ vẫn hoạt động nhưng hệ thống ECU sẽ nhận tín hiệu dữ liệu bị sai lệch, từ đó đưa ra cảnh báo trên bảng điều khiển. Các cảm biến nhiệt độ làm mát động cơ tăng cao 159°C tương ứng thời gian 49 s, cảm biến lưu lượng không khí bị hỏng tại thời điểm 47 s tương ứng 0,8 g/s, cảm biến vị trí bướm ga bị hỏng khi mở 19% tương ứng 49 s, cảm biến oxy bị lỗi khi có giá trị 0,015 tương ứng 48 s, nhiệt độ không khí nạp bị hỏng khi có giá trị 92°C tương ứng 48 s. Các kết quả thử nghiệm được thể hiện trên hình 9-11 như sau:



Hình 9. Đồ thị tốc độ động cơ ở các chế độ khảo sát.

Tại thời điểm cảm biến tốc độ động cơ gặp lỗi hư hỏng khi hoạt động ở 869 vòng/phút tương ứng 48 s bắt đầu đi xuống gần như bằng 0 như hình 9. Khi các cảm biến hư hỏng sẽ có phần mềm cảnh báo hư hỏng như hình 10 và 11.



Hình 10. Phần mềm hiển thị tốc độ động cơ hoạt động bình thường.



Hình 11. Phần mềm hiển thị tốc độ động cơ cảnh báo vòng tua máy cao gây hao xăng.

Trong quá trình thử nghiệm, các thông số được giám sát như tốc độ phương tiện và nhiệt độ nước làm mát và so sánh các tham số này ở trên máy tính và trên bảng điều khiển trung tâm của xe. Kết quả cho thấy các thông số này ở trên bảng đồng hồ điều khiển và trên máy tính đều giống nhau. Điều này cho thấy rằng mạch đọc dữ liệu động đã hoạt động chính xác và hiệu quả.

Tốc độ trung bình của ô tô là 28 km/h và cảm biến lưu lượng không khí nạp đo được dao động từ 2 g/s đến 3 g/s ở thời điểm vị trí bướm ga mở ở chế độ không tải trung bình từ 28% đến 30%. Nhiệt độ nước làm mát trung bình 85°C đến 90°C với cảm biến oxy dao động từ 0,15 V đến 0,18 V và nhiệt độ không khí nạp dao động 80°C đến 86°C.

Trong quá trình thử nghiệm, xe ô tô được chạy cả trên đường giao thông và đường cao tốc. Các kết quả thử nghiệm cho thấy rằng giá trị của các thông số hiển thị trên máy tính (thông qua mạch đọc dữ liệu động) và trên bảng đồng hồ là giống nhau. Cụ thể, đó là các thông số tốc độ động cơ, tốc độ xe, nhiệt độ nước làm mát.

Thời gian lấy mẫu được khảo sát ở nhiều chế độ khác nhau (thời gian lấy mẫu thay đổi từ 0,01 s đến 0,1 s) như sau:

Ưu điểm:

- Tần suất thu thập dữ liệu động cao hơn giúp đo lường được chính xác và độ chính xác cao hơn, đặc biệt là trong việc lấy dữ liệu động theo thời gian thực.
- Với thiết bị thu thập dữ liệu động có thể được sử dụng để tối ưu hóa hiệu suất xe và cải thiện hiệu quả nhiên liệu, khí thải...

Nhược điểm:

- Thu thập dữ liệu với tần suất cao như thế này có thể tạo ra một lượng lớn dữ liệu, khó khăn để lưu trữ và xử lý.

Sau khi thực hiện thử nghiệm nhiều lần, thời gian lấy mẫu được chọn là 0,02 s.

Khi lập trình lấy dữ liệu với số lượng lớn cảm biến thì việc sử dụng mạch Can Bus Shield và mạch Arduino để lấy mẫu với tần suất 0,02 s có thể có những ưu việt so với lấy mẫu với tần suất 0,1 s của máy chẩn đoán như sau:

Độ chính xác phù hợp với tất cả các cảm biến: Thời gian lấy mẫu 0,02 s tương ứng delay 20 trong ngôn ngữ lập trình Arduino được đánh giá đo lường chính xác tất cả các thông số liên quan đến ô tô như tốc độ phun nhiên liệu, cảm biến lưu lượng không khí, tỉ lệ nhiên liệu động cơ, cũng như các thông tin về hệ thống trên xe. Khi thời gian lấy mẫu quá dài sẽ làm giảm độ chính xác của dữ liệu.

Tốc độ truyền thông: Thời gian lấy mẫu càng ngắn, thời gian đáp ứng càng nhanh, giúp hệ thống các cảm biến và bộ xử lý giảm thiểu thời gian trễ trong việc thu thập và xử lý dữ liệu, do đó hệ thống ô tô sẽ có phản hồi nhanh hơn. Mạch Can Bus Shield cho phép truyền thông với tốc độ tối đa 1 Mbps, trong khi máy chẩn đoán thông thường chỉ hỗ trợ tốc độ truyền thông thấp hơn khoảng từ 5 Kbps đến 500 Kbps. Can Bus Shield với tốc độ truyền thông cao hơn này có thể cải thiện đáng kể khả năng thu thập dữ liệu chính xác và thời gian đáp ứng khi thu thập dữ liệu từ các cảm biến trên ô tô. Vì vậy, Arduino và Can Bus Shield có thể thu thập dữ liệu với tốc độ cao hơn và chính xác hơn. Độ linh hoạt: Arduino là một nền tảng phổ biến trong việc phát triển các ứng dụng có liên quan đến thu thập dữ liệu và điều khiển thiết bị. Bên cạnh đó, Can Bus Shield hỗ trợ nhiều loại sensor và giao thức truyền thông, do đó, Arduino và Can Bus Shield có thể linh hoạt hơn trong việc kết nối với các thiết bị cảm biến khác và thu thập dữ liệu khác nhau.

Chi phí: Thời gian lấy mẫu 0,02 s đủ để đáp ứng yêu cầu của nhiều ứng dụng thu thập dữ liệu trên ô tô mà không cần sử dụng các linh kiện cảm biến đắt tiền hay phần mềm xử lý phức tạp hơn, do đó giảm thiểu chi phí cho các ứng dụng kiểm tra độ chính xác và hiệu quả trong thực tế.

Tóm lại, thời gian lấy mẫu 0,02 s là sự lựa chọn phù hợp để đáp ứng được yêu cầu về độ chính xác và thời gian đáp ứng của nhiều ứng dụng thu thập dữ liệu trên ô tô, đồng thời giảm thiểu chi phí cho phần cứng và phần mềm.

Với các kết quả thử nghiệm được, nhóm tác giả tiếp tục tiến hành các nghiên cứu tiếp theo dựa trên nghiên cứu này. Đó là, dựa trên bộ dữ liệu động nhóm tác giả đã giám sát, phân tích theo dõi chất lượng khí thải, giám sát hành trình, giám sát mức tiêu hao nhiên liệu... nhằm thu nhận một bộ dữ liệu động phục vụ mục đích chẩn đoán kỹ thuật các cụm và hệ thống khi phát sinh các hư hỏng. Điều này sẽ giúp cho kỹ thuật viên và lái xe khai thác, sửa chữa, bảo dưỡng và sử dụng xe được hiệu quả hơn rất nhiều.

5. Kết luận và hướng phát triển

Kết quả thử nghiệm trên xe Toyota Vios 2016 cho thấy thiết bị đọc dữ liệu động trên cơ sở mạch Can Bus Shield và Arduino thông qua cổng OBD-II hoạt động tốt và chính xác cao, thời gian nhanh hơn (50 lần) so với máy chẩn đoán. Giá thành sản phẩm được thiết kế rẻ, nhỏ gọn hơn (60%) so với máy chẩn đoán của hãng theo xe. Đồng thời, đã thiết kế được công cụ chẩn đoán theo thời gian thực bằng phương pháp so sánh

ngưỡng giá trị giúp chẩn đoán tình trạng kỹ thuật hoạt động tốt và hiệu quả. Thông tin về dữ liệu động, lỗi hư hỏng của hệ thống được giám sát và hiển thị liên tục trong suốt quá trình vận hành xe thông qua giao diện phần mềm được phát triển. Việc thực hiện kết nối thiết bị với ô tô cũng như quá trình vận hành đơn giản, dễ dàng, thuận tiện cho người sử dụng. Dựa trên các kết quả thử nghiệm, có thể kết luận rằng việc kết hợp giữa giám sát dữ liệu động theo thời gian thực và phương pháp chẩn đoán nêu trên có thể được ứng dụng cho các loại xe có cổng giao tiếp OBD-II khác nhau.

Hướng nghiên cứu tiếp theo tập trung vào việc hoàn thiện thiết kế chế tạo thiết bị trên trở thành sản phẩm IoT công nghiệp nhằm: (i) Thực hiện giám sát và theo dõi cùng một lúc nhiều thông số kỹ thuật cho số lượng lớn xe đang hoạt động nhằm giúp cho nhà quản lý có kế hoạch khai thác, bảo dưỡng và chọn chế độ vận hành hợp lý cho phương tiện, giảm lượng khí thải, tăng hiệu suất sử dụng nhiên liệu; (ii) Cải thiện tốc độ lấy dữ liệu của thiết bị từ 0,02 s xuống 0,002 s; (iii) Mở rộng giao thức lấy dữ liệu như sử dụng mạch điều khiển bằng Bluetooth, Wifi, ZigBee; (iv) Thiết kế các ứng dụng hiển thị và quản lý dữ liệu trên các nền tảng máy tính và thiết bị di động để theo dõi, giám sát dữ liệu động cho nhiều thiết bị lắp trên nhiều xe cùng một lúc.

Tài liệu tham khảo

- [1] E. Gilman, A. Keskinarkaus, S. Tamminen, S. Pirttikangas, J. Rönning, and J. Rieki, "Personalised assistance for fuel-efficient driving," *Transp. Res. Part C Emerg. Technol.*, Vol. 58, pp. 681-705, 2015. DOI: 10.1016/j.trc.2015.02.007
- [2] Z. Szalay *et al.*, "ICT in road vehicles - Reliable vehicle sensor information from OBD versus CAN," in *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, 2015, pp. 469-476. DOI: 10.1109/MTITS.2015.7223296
- [3] N. Kushiro, Y. Oniduka, and Y. Sakurai, "Initial Practice of Telematics-Based Prognostics for Commercial Vehicles: Analysis Tool for Building Faults Progress Model for Trucks on Telematics Data," *Procedia Comput. Sci.*, Vol. 112, pp. 2155-2164, 2017. DOI: 10.1016/j.procs.2017.08.244
- [4] D. Sik, T. Balogh, P. Ekler, and L. Lengyel, "Comparing OBD and CAN Sampling on the go with the SensorHUB Framework," *Procedia Eng.*, Vol. 168, pp. 39-42, 2016. DOI: 10.1016/j.proeng.2016.11.133
- [5] M. D'Agostino, M. Naddeo, and G. Rizzo, "Development and validation of a model to detect active gear via OBD data for a Through-The-Road Hybrid Electric Vehicle," *IFAC Proc. Volumes*, Vol. 47 No. 3, pp. 6618-6623, 2014. DOI: 10.3182/20140824-6-ZA-1003.01166
- [6] G. Charalampidis, A. Papadakis, and M. Samarakou, "Power estimation of RF energy harvesters," *Energy Procedia*, Vol. 157, pp. 892-900, 2019. DOI: 10.1016/j.egypro.2018.11.255

- [7] M. A. C. Din, M. T. A. Rahman, H. A. Munir, A. Rahman, and A. F. A. Hamid, "Development of CAN Bus Converter for on Board Diagnostic (OBD-II) System," *IOP Conf. Ser. Mater. Sci. Eng.*, Vol. 705 No. 1, p. 12011, 2019. DOI: 10.1088/1757-899X/705/1/012011
- [8] M. Malik and R. Nandal, "A framework on driving behavior and pattern using On-Board diagnostics (OBD-II) tool," *Mater. Today Proc.*, 2021. DOI: 10.1016/j.matpr.2021.07.376
- [9] Medashe Michael Oluwaseyi and Abolarin Matthew Sunday, "Specifications and Analysis of Digitized Diagnostics of Automobiles: A Case Study of on Board Diagnostic (OBD II)," *Int. J. Eng. Res.*, Vol. V9 No. 01, pp. 91-105, 2020. DOI: 10.17577/ijertv9is010045
- [10] K. Nguyen and K. T. Le and T. T. Vu, "Design and manufacturing deployment of diesel engine common rail electronic control system," *J. Tech. Educ. Sci.*, No. 45 SE-Research Articles, pp. 86-2, Jan. 2018, [Online]. Available: <https://jte.hcmute.edu.vn/index.php/jte/article/view/380>
- [11] K. Nguyen and K. T. Le, "Design and implement the engine control system of Toyota Camry 2AR-FE and fault creation system using IoT technology," *J. Tech. Educ. Sci.*, No. 53 SE-Research Articles, pp. 66-71, Jul. 2019, [Online]. Available: <https://jte.hcmute.edu.vn/index.php/jte/article/view/225>
- [12] Bảng PID OBD-II và công thức của các cảm biến (https://en.wikipedia.org/wiki/OBD-II_PIDs).

DESIGNING THE BOARD TO READ REAL-TIME VEHICLE DATA FOR TECHNICAL DIAGNOSING THROUGH THE OBD-II INTERFACE

Abstract: *This article presents a method of designing a circuit that reads dynamic data of the vehicle through the OBD-II interface on the basis of the Can Bus Shield and Arduino Uno circuit. This communication circuit allows monitoring all vehicle parameters through sensors in real time. At the same time, the diagnostic method according to the signal threshold of the above parameters is also applied to monitor and diagnose the technical condition of cars. Through the actual test results, the real-time dynamic data monitoring device and diagnostic method can be applied to different types of vehicles with OBD-II interface. At the same time, monitoring a large number of vehicles in operation to help managers plan for operation, maintenance and choose a reasonable operating mode for vehicles is also the next main development direction of this research. The survey time of the dynamic data reading device is 0.02 s, faster (50 times) than the diagnostic machine is 0.1 s. The product price is cheap, 60% more compact than the car's diagnostic machine.*

Keywords: Can Bus Shield; OBD-II; diagnosing vehicle data; Arduino Uno.

Nhận bài: 25/04/2023; Hoàn thiện sau phản biện: 13/06/2023; Chấp nhận đăng: 31/07/2023

