

TÓM TẮT

TỔNG QUAN

Việc ứng dụng một hệ thống có thể kết nối và điều khiển các thiết bị khác thông qua một chuẩn giao thức không dây hay qua Internet thì có thể hiểu nôm na là một hệ thống IoT, việc ứng dụng như vậy sẽ giúp người dùng tiết kiệm thời gian, nhân lực nhưng vẫn có thể điều khiển, theo dõi, giám sát các thiết bị từ xa.

Đối với ngành công nghiệp ô tô (Automotive) việc sử dụng những công nghệ đó cũng không là ngoại lệ. Hiện nay trên những dòng xe ta thấy hàng ngày của các hãng như Hyundai, KIA đã cho ra mắt các ứng dụng di động có thể theo dõi giám sát chu kì hoạt động của xe cũng như là coi các thông số cần thiết của xe từ xa thông qua Internet. Những dòng xe hàng ngày ta có thể bắt gặp trên đường thường được biết đến là passenger car vậy sẽ như thế nào nếu chúng ta ứng dụng công nghệ đó vào những dòng xe nông nghiệp như máy cày, máy xúc hay là các dòng xe tải lớn, container lớn. Để từ đó có thể phát triển thêm cho ngành nông nghiệp cũng như chất lượng các nông sản thông qua việc giám sát theo dõi, ứng biến kịp thời với các vấn đề phát sinh nhờ vào ứng dụng có thể xem từ xa.

Đó cũng chính là lý do nhóm đã lựa chọn cũng như phát triển đề tài cũng như là do hiện nay nhiều tập đoàn trong ngành công nghiệp ô tô đều đã tiến đến cải tiến và phát triển dựa trên xu hướng này nên nhóm chúng tôi quyết định nghiên cứu lý thuyết liên quan đến đề tài từ đó thiết kế và thi công hệ thống giám sát, chẩn đoán dành cho xe nông nghiệp dựa trên chuẩn giao thức J1939. Để có thể nói rõ hơn về đề tài một cách ngắn gọn thì việc sử dụng điều khiển và quản lý một hệ thống trên xe cơ giới nói chung và xe nông nghiệp nói riêng đều thông qua một bộ điều khiển điện tử ECU (Electronic control unit), nó có thể giao tiếp với các thiết bị khác thông qua một giao thức mạng nối tiếp là CAN (Controller area network). Khác với các dòng xe ô tô khách, đối với các loại xe cơ giới, hiệp hội các kỹ sư trong ngành ô tô – SAE (Society of Automotive Engineers) đã nghiên cứu ra chuẩn giao thức J1939 chuyên dùng cho các dòng xe này. Chuẩn giao thức J1939 được biết đến là một giao thức tiêu chuẩn trên các xe nông nghiệp vì có thể dùng cho các hệ thống chẩn đoán cũng như liên lạc giữa các thành phần xe. Đây là giao thức cấp cao được phát triển dựa trên giao thức CAN, chịu trách nhiệm cho việc liên lạc, giao tiếp thông tin của các ECU trên tất cả các loại xe cơ giới.

MỤC TIÊU ĐỀ TÀI

Các mục tiêu nhóm đề ra bao gồm:

Thứ nhất: thiết kế hệ thống có thể xử lý, chuyển đổi các dữ liệu CAN được gửi từ ECU theo chuẩn J1939 và có thể xác định vị trí của phương tiện thông qua GPS.

Thứ hai: thiết lập và cấu hình Server có thể truyền các dữ liệu lên Database và giao tiếp với giao diện người dùng

Thứ ba: thiết kế giao diện hiển thị các thông số xe như nhiệt độ động cơ, tốc độ xe, quãng đường đã đi được, mức nhiên liệu lên trên Web và ứng dụng điện thoại, đồng thời sử dụng một màn hình GUI (graphical user interface) hiển thị để người dùng có thể trực tiếp xem các thông số trên xe.

Thứ tư: Tiến hành thiết lập để kiểm thử hệ thống và đảm bảo độ trễ truyền dữ liệu của hệ thống không quá 5s.

PHƯƠNG PHÁP NGHIÊN CỨU

Thực hiện nghiên cứu, tìm hiểu phân tích lý thuyết cách vận hành và hoạt động của một ECU. Nghiên cứu tìm hiểu các yêu cầu người dùng, từ đó phát triển hệ thống đáp ứng được các yêu cầu trên.

Thiết lập các khối theo từng thành phần và phân chia công việc dựa vào các khối trên để có thể tiến đến thiết kế chi tiết một hệ thống.

Tổng hợp các khối với nhau thông qua các chuẩn giao tiếp để có thể thiết kế một hệ thống và kiểm tra hoạt động của toàn bộ hệ thống.

YÊU CẦU HỆ THỐNG

Thiết kế hệ thống có thể xử lý, chuyển đổi các dữ liệu CAN được gửi từ ECU theo chuẩn J1939 và có thể xác định vị trí của phương tiện thông qua GPS

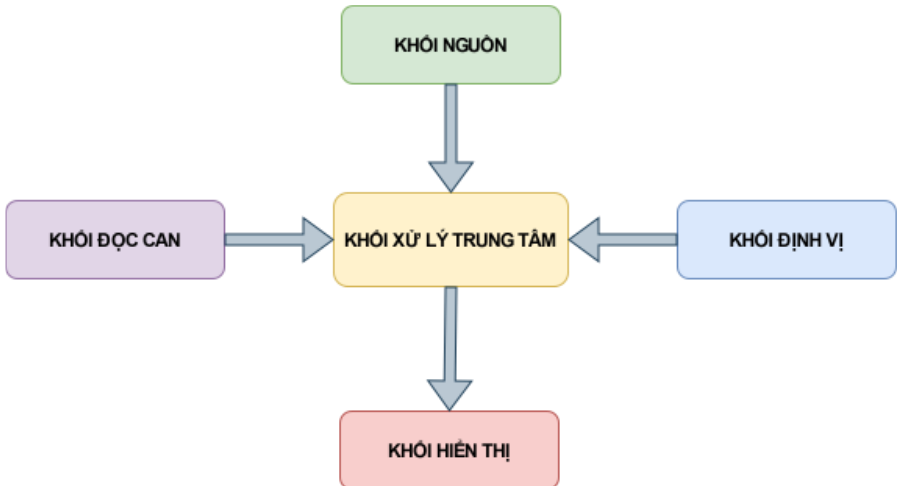
Thiết kế giao diện hiển thị các thông số Web và ứng dụng điện thoại, GUI (graphical user interface)

Thiết lập và cấu hình Server có thể truyền các dữ liệu lên Database và giao tiếp với giao diện người dùng

Tiến hành thiết lập kiểm thử hệ thống và đảm bảo độ trễ truyền dữ liệu của hệ thống không quá 5s

THIẾT KẾ PHẦN CỨNG

- Thiết kế phần cứng dựa trên sơ đồ khối chi tiết tại gateway



Hình 1: sơ đồ khối chi tiết

Khối đọc CAN: sử dụng module RS485 CAN HAT kết nối với ECU thông qua hai dây CAN HIGH và CAN LOW để đọc dữ liệu CAN và truyền các khung CAN đó qua khối xử lý trung tâm là Raspberry Pi thông qua chuẩn giao tiếp SPI.

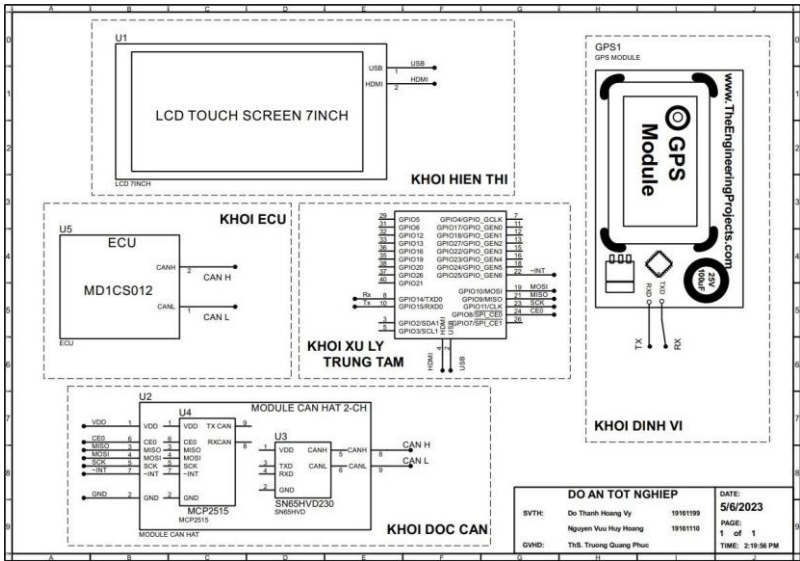
Khối xử lý trung tâm: sử dụng một máy tính thu nhỏ Raspberry Pi để đảm nhiệm cho việc quản lý và xử lý mọi hoạt động của hệ thống từ việc chuyển đổi các dữ liệu cho đến khi truyền gửi các dữ liệu đó lên Cloud.

Khối định vị: dùng module định vị GPS NEO-6M với kích thước nhỏ gọn và độ nhạy cao thích hợp cho việc giám sát, định vị phương tiện.

Khối hiển thị: sử dụng màn hình Waveshare 7 Inch kết nối HDMI với Raspberry Pi, có tích hợp cảm ứng trên màn hình, dùng để hiển thị trực tiếp các thông số nhiệt độ động cơ, tốc độ xe, mức nhiên liệu, quãng đường đã đi trên xe.

Khối nguồn: dùng máy phát nguồn để cấp trực tiếp cho ECU hoạt động và nguồn chuyên dụng của Raspberry để hoạt động.

➤ Dựa trên sơ đồ khối chi tiết sẽ thiết kế sơ đồ nguyên lý:



Hình 2: sơ đồ nguyên lý phân cứng hệ thống

THIẾT KẾ PHẦN MỀM

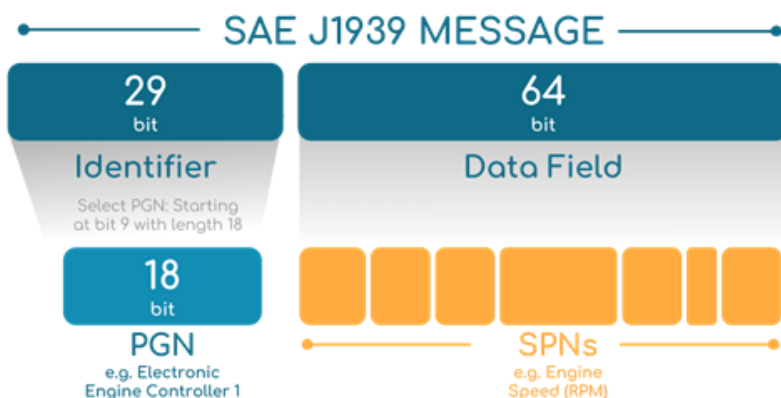
- **Lý thuyết:**

J1939 là một chuẩn giao thức truyền thông phổ biến trong ngành công nghiệp ô tô và các phương tiện di chuyển khác, được phát triển bởi Hiệp hội Kỹ sư ô tô Hoa Kỳ (SAE International). Chuẩn này giúp các nhà sản xuất xe ô tô khác nhau kết nối các bộ phận điện tử và truyền tín hiệu giữa chúng một cách tương thích, đồng thời giúp giảm chi phí phát triển sản xuất các hệ thống điện tử.

J1939 đóng vai trò quan trọng trong việc truyền tải các thông tin về động cơ, hộp số, phanh, hệ thống đánh lái, hệ thống giám sát và các thông tin khác giữa các nút trên mạng. Các thông tin này được đóng gói trong các tin nhắn có định dạng chuẩn và được truyền tải trên các kênh truyền thông riêng biệt. Mỗi tin nhắn được định danh bằng một địa chỉ ID riêng, giúp các nút khác trên mạng có thể xác định và phân tích nội dung của tin nhắn. J1939 được sử dụng rộng rãi trong ngành công nghiệp ô tô và các phương tiện di chuyển khác như xe tải, xe buýt, máy công trình, tàu thủy và máy bay. Đóng vai trò quan trọng trong việc giúp các

hệ thống điện tử trên các phương tiện này hoạt động một cách hiệu quả và đảm bảo tính an toàn cho người sử dụng.

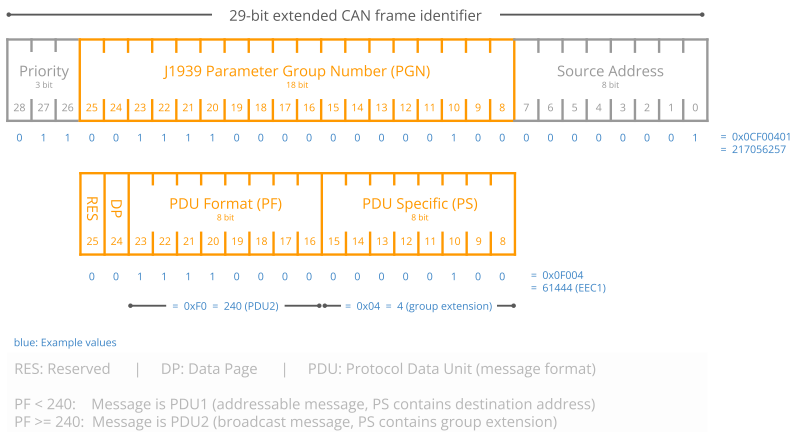
J1939 sử dụng giao thức truyền thông CAN (Controller Area Network) để truyền thông giữa các thiết bị trên mạng. Vì vậy, các khung tin truyền thông (message frames) của J1939 cũng tuân theo định dạng khung tin truyền thông CAN.



Hình 3: một khung truyền cơ bản CAN, J1939

Với khung truyền của J1939 và CAN về cơ bản sẽ bao gồm hai phần chính, một sẽ là Identifier (ID) để có thể nhận diện cũng như biết được giá trị của khung đó đang là gì, tiếp đến sẽ là Data Field đây là vùng sẽ chứa các giá trị từ đó có thể chuyển đổi qua

một số vật lý nhất định cho người dùng có thể đọc được. Hiện nay về khung truyền có thể mở rộng lên đến 29 bit đối với vùng Identifier, có thể biết đây là khung Extended ID đối với CAN và J1939.

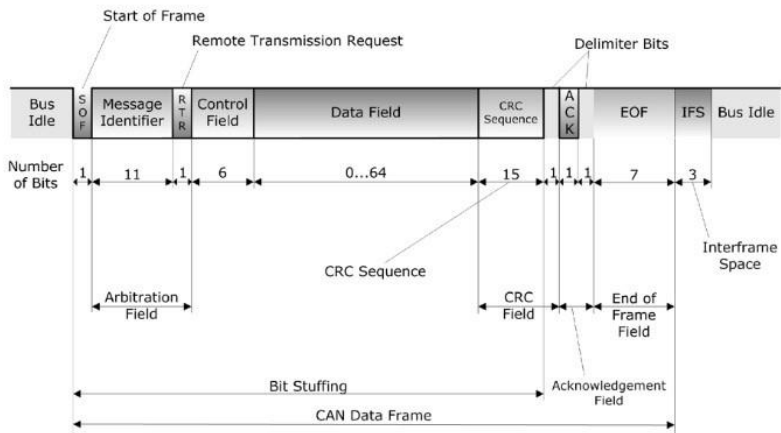


Hình 4: chi tiết khung truyền Identifier trong J1939

Một khung tin truyền thông CAN của J1939 bao gồm 4 phần chính: khung tiêu đề (header) hay là Arbitration Field, trường dữ liệu (data field), khung điều khiển (control field) và khung kết thúc (end of frame).

Khung tiêu đề (header): Đây là phần đầu tiên của khung tin truyền thông sẽ bao gồm các phần về Identifier của khung truyền, bao gồm các trường sau:

- + Priority: Xác định mức độ ưu tiên của khung tin truyền thông. Trong J1939, giá trị priority được sử dụng để xác định ưu tiên truyền thông giữa các thông điệp khác nhau. Các thông điệp cần truyền thông quan trọng hơn sẽ có giá trị priority cao hơn.
- + PDU format (Protocol Data Unit format): Xác định định dạng PDU được sử dụng trong khung tin truyền thông.
- + Source address: Xác định địa chỉ của thiết bị gửi thông điệp.



Hình 5: chi tiết khung truyền CAN, J1939

Khung điều khiển (Control field): Đây là phần đặc biệt của J1939, bao gồm các trường sau:

+ Data page (trang dữ liệu): Xác định trang dữ liệu được sử dụng trong trường dữ liệu.

+ Number of packets (số lượng gói tin): Xác định số lượng gói tin được sử dụng để chuyển đổi trường dữ liệu.

+ Packet sequence number (số thứ tự gói tin): Xác định số thứ tự của gói tin trong chuỗi gói tin.

+ End of message (kết thúc tin nhắn): Xác định kết thúc tin nhắn.

Trường dữ liệu (Data field): Đây là phần chứa thông tin cần truyền thông. Các trường dữ liệu của J1939 có thể bao gồm các thông số như tốc độ động cơ, nhiên liệu còn lại, áp suất lốp, nhiệt độ nước làm mát.

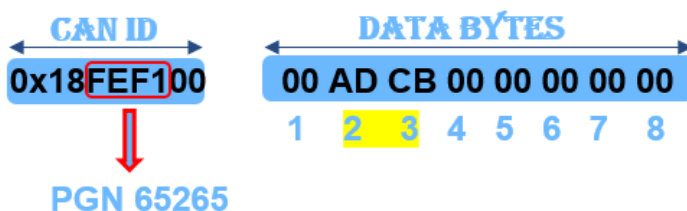
Khung CRC (Cyclic Redundancy Check): đây là khung để kiểm tra lỗi.

Khung kết thúc (End of frame): Đây là phần kết thúc của khung tin truyền thông, bao gồm một bit CRC để kiểm tra lỗi và các bit tạm dừng (interframe space) để tạo khoảng thời gian giữa các khung tin truyền thông.

Ví dụ chuyển đổi:



Khi gateway nhận được một khung CAN theo chuẩn J1939. Ta cần tách CAN ID để nhận được các thông tin PNG của khung truyền



Dựa theo lý thuyết nằm từ byte thứ 3 đến byte thứ 6 của khung dữ liệu. PGN nhận được từ khung dữ liệu là 65265. Dựa theo tài liệu của chuẩn ta thu được các thông tin chưa trong trường dữ liệu của khung

(R) PGN 65265 Cruise Control/Vehicle Speed 1			
Cruise Control/Vehicle Speed 1			
Transmission Repetition Rate:	100 ms		
Data Length:	8		
Extended Data Page:	0		
Data Page:	0		
PDU Format:	254	PGN Supporting Information:	
PDU Specific:	241		
Default Priority:	6		
Parameter Group Number:	65265 (0x00FEF1)		
Start Position	Length	Parameter Name	SPN
1.1	2 bits	Two Speed Axle Switch	69
1.3	2 bits	Parking Brake Switch	70
1.5	2 bits	Cruise Control Pause Switch	1633
1.7	2 bits	Park Brake Release Inhibit Request	3507
2-3	2 bytes	Wheel-Based Vehicle Speed	84
4.1	2 bits	Cruise Control Active	555
4.3	2 bits	Cruise Control Enable Switch	596
4.5	2 bits	Brake Switch	597
4.7	2 bits	Clutch Switch	598
5.1	2 bits	Cruise Control Set Switch	599
5.3	2 bits	Cruise Control Coast (Decelerate) Switch	600
5.5	2 bits	Cruise Control Resume Switch	601
5.7	2 bits	Cruise Control Accelerate Switch	602
6	1 byte	Cruise Control Set Speed	86
7.1	5 bits	PTO Governor State	976
7.6	3 bits	Cruise Control States	527
8.1	2 bits	Engine Idle Increment Switch	968
8.3	2 bits	Engine Idle Decrement Switch	967
8.5	2 bits	Engine Test Mode Switch	966
8.7	2 bits	Engine Shutdown Override Switch	1237

Hình 6: Bảng thông tin PGN 65265

Ta có thể thấy trường dữ liệu của khung chứa rất nhiều thông tin, ở đây nhóm chọn mẫu thông tin về tốc độ của xe “vehicle speed”, có vị trí dữ liệu nằm ở byte số 2 và 3 của khung dữ liệu và thông số có chỉ số SPN tương ứng là 84, tiếp tục dựa trên tài liệu được cung cấp ta có thể lấy được giá trị “Resolution” và “offset” trong trường hợp này thì “Resolution” và “offset” tương ứng là 1/256 và 0 như hình dưới đây

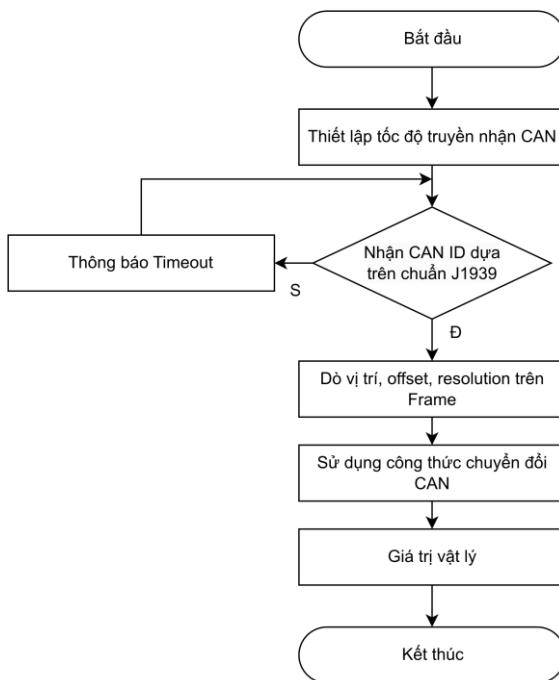
SPN 84 Wheel-Based Vehicle Speed	
Speed of the vehicle as calculated from wheel or tailshaft speed.	
Data Length:	2 bytes
Resolution:	1/256 km/h per bit, 0 offset
Data Range:	0 to 250.996 km/h
Type:	Measured
Supporting Information:	
PGN reference:	65265

Hình 7: Bảng thông tin SPN 84

Sau khi có các thông tin cần thiết ta tiến hành áp vào công thức để thu được tốc của xe hiện tại.

$$\begin{aligned}\text{Physical} &= \text{Network} * \text{Resolution} + \text{Offset} \\ &= \text{CB AD} * 1/256 + 0 \\ &= 52141 * 1/256 + 0 \\ &= 204 \text{ (Km/h)}\end{aligned}$$

Lưu đồ giải thuật trên gateway:



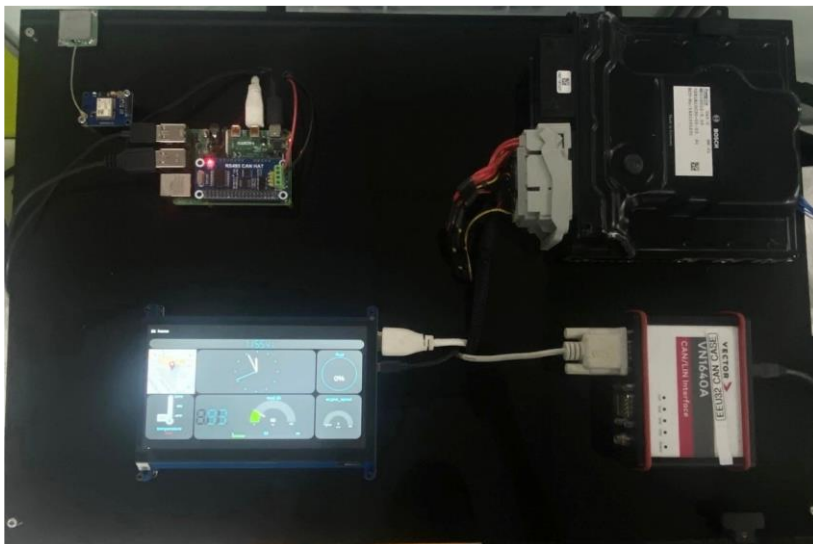
Thiết lập tốc độ truyền nhận CAN với BaudRate là 500000, sử dụng thư viện Python-Can để thiết lập chọn lọc các dữ liệu dựa trên

những CAN ID được lựa chọn dựa theo tiêu chuẩn J1939, khi không có dữ liệu đúng với CAN ID (PGN) được lựa chọn sẽ xuất hiện thông báo Timeout. Nếu hiển thị đúng các CAN ID thì dựa vào chuẩn J1939 với vị trí tương ứng và offset cũng như resolution của các giá trị hiển thị trên Frame, từ đó áp dụng công thức chuyển đổi giá trị của CAN để có thể chuyển đổi sang giá trị vật lý.

$$Physical\ value = CAN\ value * resolution + offset$$

KẾT QUẢ

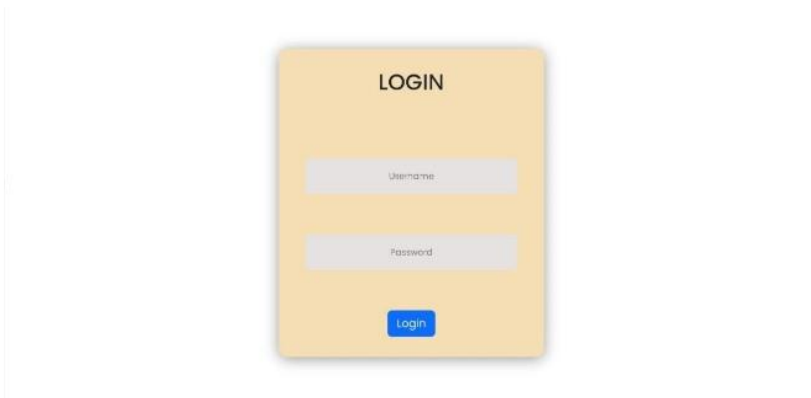
Kết quả phần cứng:



Hình 8: kết quả phần cứng

Đối với phần cứng thì nhóm đã thiết kế và đi dây dựa trên mô hình được giới thiệu cho khách hàng trong ở công ty Bosch. Với thiết kế theo một dạng sa bàn đặt các thiết bị phần cứng lên trên một bàn mà người dùng có thể quan sát được các thiết bị hoạt động cũng như có thể kiểm tra, theo dõi các linh kiện như ECU và cục Vector thuộc công ty Bosch giúp truyền gửi CAN Frame.

Kết quả giao diện web

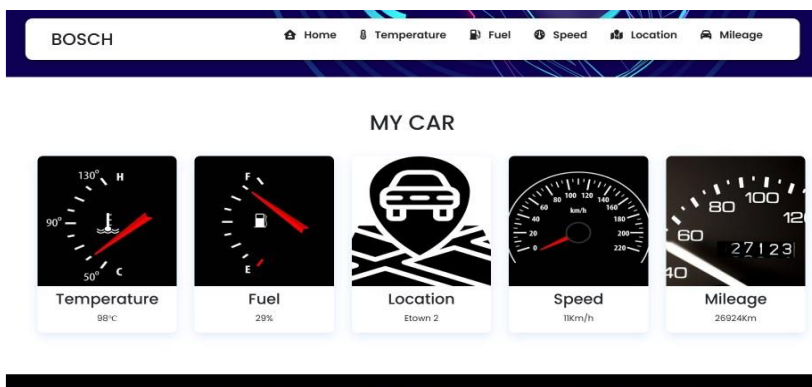


Hình 9: Giao diện đăng nhập web

Cách vận hành của giao diện này cũng sẽ giống với cách vận hành hoạt động của ứng dụng trên điện thoại, chỉ khác cách bố trí cũng như màu sắc, sắp xếp các khối. Về tài khoản người dùng thì cũng sẽ được cấp bởi nhà cung cấp, vừa có thể dùng tài khoản đó đăng nhập vào ứng dụng trên điện thoại và cả trên web này. Để có thể gửi các thông tin và kiểm tra xem có đúng với các tài khoản được định sẵn trên database hay không, sẽ sử dụng giao thức HTTP, truyền gửi một cách nhanh chóng cũng như đảm bảo về tính bảo mật, an toàn.

Cách để có thể nhận diện khi mình đăng nhập có đúng tài khoản hay mật khẩu không thì trên giao diện sẽ hiển thị. Chỉ khi đăng nhập sai thì trên giao diện sẽ hiện dòng chữ “Wrong username or

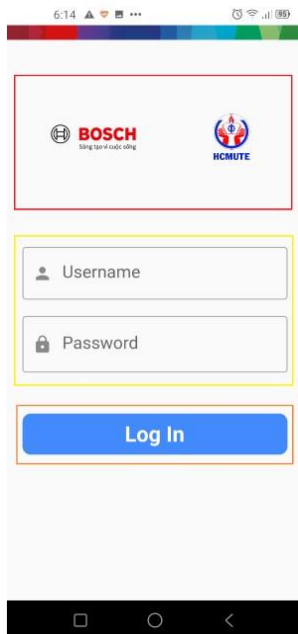
password” như Hình 4.15 thì người dùng cần phải nhập đúng với tài khoản, ngược lại khi đăng nhập thành công thì sẽ chuyển thẳng vào trang sau.



Hình 10: Giao diện chính web

Hiện thị toàn bộ các thông số của xe như tốc độ xe, nhiệt độ động cơ, mức nhiên liệu và quãng đường đi được, đây cũng là trang chính nhằm giúp cho người dùng xem tổng quát về các giá trị thông số mà nhóm lựa chọn. Ngoài ra thì có thể ẩn vào từng thông số để có thể xem chi tiết các thông số giao diện hiển thị như gauge, nhiệt kế cũng như có thể nhận được các thông báo chẩn đoán, cảnh báo đến cho người dùng

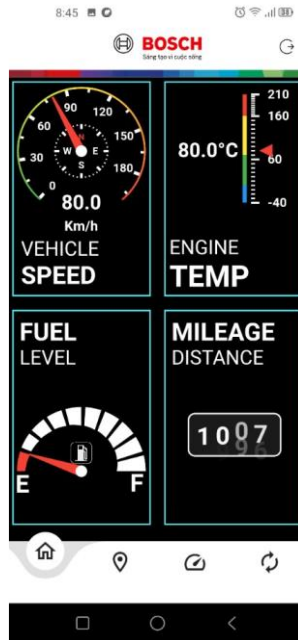
Kết quả giao diện App



Hình 11: giao diện đăng nhập App

Ở giao diện này sẽ được kết nối với Server thông qua API nhằm mục đích xác minh đúng với các tài khoản ở trên Database đang chứa hay không nếu được thì sẽ hiển thị tiếp theo đến trang chính còn ngược lại thì sẽ hiển thị sai mật khẩu hoặc tên đăng nhập. Ở phần trên thì nhóm đã sử dụng cả hai logo một là logo của trường UTE và logo của công ty Bosch đã hỗ trợ các thiết bị, công cụ để nhóm có thể làm được đề tài này. Hai khung dưới sẽ là nơi nhập Username và Password mà giả định rằng chủ xe đã

được cấp bởi nhà cung cấp xe, tiếp đến là nút Log In để có thể đăng nhập vào trang tiếp theo.



Hình 12: giao diện chính của App

Dựa vào hình trên, giao diện sẽ hiển thị các thông số như tốc độ xe, nhiệt độ động cơ, mức nhiên liệu và quãng đường đi được tương ứng với 4 ô khác nhau đi theo thứ tự trên trên xuống và từ trái qua phải, ngoài ra còn có cách hiển thị các thông số dựa trên các hoạt ảnh động giúp tạo ấn tượng đối với người dùng.

Kết quả giao diện GUI



Hình 13: trang chính trên GUI

Trong giao diện trang “Home”, các thông số vật lý đặc trưng của động cơ được thu thập thông qua chuẩn J1939 sẽ được thể hiện. Các thông số của động cơ sau khi được chuyển đổi từ CAN sẽ được lưu vào các tệp lưu trữ dữ liệu, GUI sẽ đọc dữ liệu trực tiếp từ các tệp lưu trữ ấy và hiển thị lên trang “home”, thứ tự hiển thị lần lượt là:

1. Đồng hồ thể hiện thời gian thực.
2. Bản đồ thu nhỏ, thể hiện vị trí hiện tại của phương tiện.
3. Đồ thị hiển thị lượng nhiên liệu hiện tại của phương tiện, đơn vị (%).

4. Đồ thị hiển thị nhiệt độ hiện tại của động cơ, đơn vị ($^{\circ}\text{C}$).
5. Đồ thị thể hiện quãng đường mà phương tiện đã đi được, đơn vị (Km).
6. Đồ thị hiển thị vận tốc của phương tiện ở thời điểm hiện tại, đơn vị (Km/h).

Vì GUI đọc dữ liệu trực tiếp từ các tệp lưu trữ mà không thông qua bất cứ giao thức truyền thông nào nhờ vậy hạn chế được tối đa thời gian trễ và không cần phụ thuộc vào đường truyền mạng, đồng thời các tệp lưu trữ sẽ liên tục được ghi đè dữ liệu, tránh gây hiện tượng tràn bộ nhớ.

Controller

ENGINE_SPEED

TEMPERATURE

FUEL

MILEAGE

STORE

store_table_speed

ID	Engine_Speed	Date	Time
120	73	4/5/2023	5:53:40
121	78	4/5/2023	5:53:45
122	33	4/5/2023	5:53:50
123	91	4/5/2023	5:53:55
124	61	4/5/2023	5:54:0
125	25	4/5/2023	5:54:5
126	65	4/5/2023	5:54:10
127	30	4/5/2023	5:54:15
128	30	4/5/2023	5:54:20
129	5	4/5/2023	5:54:25
130	40	4/5/2023	5:54:30
131	49	4/5/2023	5:54:35
132	45	4/5/2023	5:54:40
133	42	4/5/2023	5:54:45
134	20	4/5/2023	5:54:50

store_scheul

V-KeyBoard On

4/5/2023

MAX
99

AVG
53

MIN
2

store

ENGINE_SPEED **TEMPERATURE** **MILEAGE**

Hình 14: bảng trích xuất dữ liệu trên GUI

Trên hình là bảng trích xuất dữ liệu tốc độ của xe ngày 4/5/2023, ta có thể thấy bảng hiển thị thông số ở phía tay trái (Hình 4.34), đồng thời ta có thể biết tốc độ của xe cao nhất trong ngày 4/5/2023 là 99 Km/h, tốc độ của xe thấp nhất trong ngày 4/5/2023 là 2 Km/h, tốc độ của xe trung bình trong ngày 4/5/2023 là 53 Km/h.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Ưu điểm:

Hệ thống thực hiện được các chức năng theo yêu cầu người dùng. Các tính năng hoạt động ổn định, đáp ứng được các tiêu chí yêu cầu về thời gian dưới 5s đối với một mô hình truyền gửi thực tế trên xe. Tính ứng dụng cao, hiện nay về ý tưởng mô hình đã được tích hợp vào một số hệ thống của các hãng xe hiện nay.

Nhược điểm:

Đọc CAN từ Raspberry Pi vẫn còn trễ với một thời gian ngắn khi gửi liên tục từ ECU. Cơ sở dữ liệu còn bị hạn chế dung lượng lưu trữ. Chưa tối ưu hóa được các hiển thị giao diện bản đồ.

Hướng phát triển:

Phát triển thêm phân quyền từ đó có thể xem xét mở rộng thêm với việc quản lý mỗi tài khoản đối với nhiều xe khác nhau.

Kết hợp dữ liệu thu thập với trí tuệ nhân tạo để hệ thống có thể tự tính toán và đoán trước được .

Đưa tín hiệu điều khiển ngược về ECU, từ đó có thể điều khiển được động cơ phương tiện.