

Integrated Cluster Development with Human Machine Interface

Seema Borawake¹, Rakhi Khedkar², Sankalp Hivare³, Monika Jagtap⁴, Aadarsh Harne⁵ Vaibhav Patharikar⁶

¹E&Tc, SAOE, India

²DoEEE, MIT-WPU, India

³E&Tc, SAOE, India

⁴E&Tc, SAOE, India

⁵E&Tc, SAOE, India

⁶Smart Qik Solutions LLP, India

Abstract- *In this paper, the project aims to transform car dashboards with an integrated, user-friendly interface that improves driving and vehicle management. Unlike traditional dashboards, which rely on separate indicators and analog displays, this advanced system combines multiple features into a single, comprehensive interface. Key features include real-time speed display, distance tracking, engine RPM, gear status, and online music streaming for enhanced user entertainment. The speedometer and odometer allow drivers to monitor their speed and distance travelled, while the engine RPM and gear indicator ensure efficient driving. In contrast to traditional dashboards, this system also incorporates vehicle diagnostics, enabling early detection of issues like engine problems or system malfunctions, which promotes timely maintenance and reduces repair costs. The system's wiring harness is designed to efficiently connect multiple components, reducing clutter and improving reliability compared to traditional dashboards with their complex and bulky wiring, results in decreasing weight of car and increase car mileage. It also offers a visually enhanced interface, implemented using the DGUS-Based UI Deployment Algorithm for the DWIN display, ensuring smooth rendering of critical information like speed, gear status, and temperature. The centralized display is strategically located to minimize driver distraction, ensuring safer driving by requiring only slight glances to the left instead of looking up or down.*

Additionally, the integration of a temperature gauge monitors atmospheric conditions, while online music streaming provides access to playlists for entertainment. The CAN (Controller Area Network) protocol enables seamless communication within the system, utilizing the CAN-to-UART Data Transmission Algorithm to efficiently relay data between components, enhancing performance

and reliability. Overall, this modern dashboard redefines vehicle interaction by providing a safer, more enjoyable, and efficient driving experience, surpassing the limitations of traditional dashboards.

Key-words: *Integrated interfaces, Driving experience, Real – time driving information, CAN protocol.*

1. INTRODUCTION

The proposed idea focuses on transforming the traditional car dashboard by integrating essential and modern functionalities into a single touchscreen display. By combining multiple controls and displays into one centralized system, the project aims to enhance vehicle efficiency, reduce manufacturing costs, and elevate the overall driving experience. One of the primary benefits of this system is the significant reduction in the amount of wiring typically required in traditional dashboards. In a traditional setup, each feature, such as the speedometer, temperature gauge, and entertainment system, requires separate wiring and components, which increases maintenance challenges and complicates the manufacturing process. By integrating these features into a single interface, the project minimizes the need for extensive wiring, resulting in a lighter vehicle. A lighter car not only improves fuel efficiency but also reduces wear and tear on the engine, leading to enhanced performance and a longer lifespan. The project also contributes to cost efficiency by reducing the number of physical components in the dashboard. Traditional dashboards require multiple buttons, switches, and dials, each

adding to production costs. By replacing these separate elements with a unified touchscreen interface, automobile industries can reduce material and labour costs. These savings are significant, as fewer components simplify the manufacturing process, leading to shorter production times and lower overall costs. In turn, this cost reduction can make advanced features more affordable for consumers, enhancing the marketability of vehicles equipped with this system.

The touchscreen interface is designed to be accessible to drivers of all heights and body types. Unlike traditional dashboards, where controls may be difficult to reach or view depending on the driver's seating position, this system positions the touchscreen in the middle of the dashboard. This placement ensures it is within easy reach of the driver, regardless of height or seat adjustment, enhancing both accessibility and safety. Drivers can access important information and controls, such as speed monitoring, odometer readings, entertainment options, and diagnostics, without taking their eyes off the road or making unnecessary movements. The interface is also user-friendly, making it easy to navigate through various functions efficiently.

Unlike traditional dashboards that would require physical modifications to add new features or improve functionality, this system can be updated through software. Automobile industries can introduce new features, fix bugs, or optimize system performance through simple software updates, ensuring that the vehicle remains current with the latest technological advancements. This flexibility makes the system adaptable and ready for future innovations without requiring costly hardware changes. The project also improves vehicle safety and connectivity. The touchscreen is positioned in a central location, providing quick access to real-time data and alerts. This includes warnings about vehicle maintenance, safety notifications, and access to entertainment without the need for external devices, reducing driver distraction. By consolidating these features into one interface, the project offers a safer, more convenient, and more connected driving experience. Overall, the project represents a significant leap forward in vehicle design, offering a cost-effective, efficient, and adaptable solution that enhances both vehicle performance and driver experience.

2. RELATED WORK

In [1], the development of an advanced vehicle instrument cluster using STM32H735IGDK

microcontroller focused on integrating features like odometer, speedometer, and Trip A/B readings. Hall sensors ensured precise measurements, while Liquid Crystal Display-TFT Display Controller interface enhanced data presentation on TFT-LCD screens. Tools like STM32CubeMX and Touch GFX (Touch Graphics Framework) simplified peripheral configuration and interface design. With inputs like ignition, mode, and reset switches, the cluster aimed to improve driver experience by offering real-time, reliable data, enhancing safety, and situational awareness. In [2] Modern vehicles integrate systems like navigation, monitoring, and driver assistance, necessitating reconfigurable clusters and central displays for information bundling. Enhanced interfaces ensure ergonomic usability, balancing technical and user needs. The 2006 ESOP and proposed ADAS guidelines emphasized systematic evaluation of in-vehicle systems. Night vision systems introduced further demands on HMI, making intuitive, menu-driven designs essential for driver focus and safety, reflecting the growing complexity and importance of advanced automotive HMIs.

In [3] the study explored HMI challenges in modern vehicles due to added features like navigation and media. Effective HMI design reduced driver distraction by simplifying function access. Traditional surveys proved inadequate in evaluating user behaviour, while vehicle data analysis offered precise insights. By combining quantitative data logging with driver interviews, researchers enhanced HMI usability and competitiveness. The approach focused on understanding real-world usage, improving design to minimize complexity and optimize driver interaction. In [4] author discusses the development of HMIs for two-wheeler digital instrument clusters, focusing on improving the rider's experience through clear displays, real-time data, and IoT integration. It addresses design challenges like user-friendliness and safety, ensuring connectivity with mobile devices. The paper highlights the importance of scalable software architecture for different vehicle models, details the technologies used, testing methods, and emphasizes customer feedback in refining human machine interface systems.

In [5] paper outlines the collaborative process between original equipment manufacturers and suppliers in developing automotive instrument clusters. It stresses the need for effective communication to align technical and aesthetic standards. The paper provides a

step-by-step overview of the development process, from concept to integration, focusing on technical specification alignment, market-specific customization, and quality assurance. It also discusses common challenges like time management and supplier coordination to ensure cohesive product quality. In [6] The author provides insights into HMI development within instrument clusters for modern vehicles. It explores advancements in HMI technologies and the impact of digitalization on traditional automotive instrument clusters. The paper reviews different design methodologies used in HMI systems, including UI frameworks, graphical processing, and software reliability. Key elements discussed include user-centric designs that ensure intuitive and safe interactions, even at high speeds, and the integration of complex information such as navigation, driver assistance, and infotainment. The authors highlight the significance of standardized development practices for scalability across vehicle models and the incorporation of real-time data analytics for enhanced driver awareness. The paper ultimately underscores the value of HMI systems in improving overall vehicle ergonomics and safety.

3. PROPOSED FRAMEWORK

3.1 Problem Statement – The proposed project aims to create a unified, user-friendly, and cost-effective display in cars that combines essential information and entertainment for the user, enhancing convenience, safety, and the overall driving experience.

3.2 Objectives –

1. To interface speedometer with screen and display the real time speed on screen.
2. To track the total distance travelled by the vehicle, this is crucial for maintenance and service schedules with the help of odometer.
3. To integrate the online music streaming services, which allowing drivers and passengers to enjoy a wide range of music and audio content without the need for external devices.
4. To implement a system for displaying important messages, alerts and notifications such as low fuel warnings and maintenance reminders.
5. To integrate temperature sensors for measuring both inside and outside temperatures and to display real-time temperature readings on the screen.

6. To develop advanced diagnostic tools to monitor vehicle systems to provide real-time data on vehicle health and performance for proactive maintenance.
7. To integrate a gear indicator that displays the current gear in which the car is being driven, the system will utilize real-time data from the vehicle's transmission system, ensuring accurate and updates on the dashboard display for the driver's convenience and safety.

4. METHODOLOGY

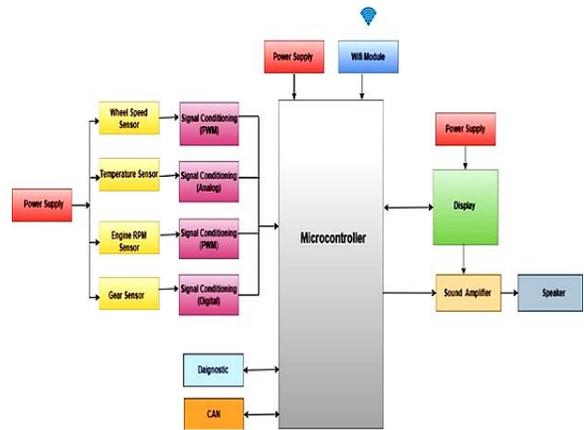


Fig.1. Block Diagram of Integrated Cluster Development with Human Machine Interface.

Fig.1 shows the block diagram of Integrated cluster development with human machine interface and its description is mention below:

Speed sensor : A speed sensor is critical in measuring vehicle speed by detecting wheel or transmission shaft rotations. It operates by capturing the rotational motion, then converting this mechanical input into an electronic pulse signal. The frequency of these pulses correlates directly to the speed higher frequencies indicate faster speeds. Speed sensors can use magnetic or optical technologies to detect rotation. Once the signal is processed, it's transmitted to the vehicle's main system, which then calculates real-time speed information, crucial for speedometer readings and safety applications such as ABS, stability control, and adaptive cruise control systems. The Fig 2 shows a wheel speed sensor also table 1 shows the different specifications of it. The equation (1) is used to calculate the actual speed by using the raw data obtained from the sensor.

$$Speed = \frac{rawSpeedData * 200}{4096} \tag{1}$$



Fig.2. Speed sensor

Table 1 Specifications of Speed Sensor

Sr. No	Parameters	Value
1.	Operating Voltage	5V to 12V DC
2.	Output Signal	PWM
3.	Operating Temperature Range	40°C to +150°C

Gear sensor : The gear sensor detects the current gear position by monitoring the gear shift lever’s movements. Positioned near the gear mechanism, it translates mechanical shifts into electrical signals, allowing the microcontroller to understand the exact gear engaged. This real-time data is critical for driver displays and automatic transmission systems, enabling smoother shifts and better fuel efficiency. By communicating the gear status to the vehicle's control system, the gear sensor supports enhanced performance by adjusting power distribution and throttle responses based on gear position, creating a more responsive and efficient driving experience. In proposed system gear indication is done over CAN protocol. The equation (2) is used for the calculation of Gear. The gear range is assumed to be 0-6 and the raw value is scaled accordingly from 0-4095.

$$Gear = \frac{(rawGear*6)}{4096} \quad (2)$$

Temperature Sensor LM32D : The temperature sensor in a vehicle measures the external ambient temperature. The outside temperature sensor, often positioned in the front grille, accurately detects ambient temperature to aid in climate control adjustments and impact engine management as needed, readings are displayed on the dashboard, helping drivers monitor external conditions for safer, more efficient driving. In

[8] the detailed specification can be observed. The Fig 3 shows the temperature sensor and table 2 shows the different specification of it. The equation (3) is used for the calculation of temperature.

$$Temperature = (rawTemperature * \frac{150}{4096}) - 45 \quad (3)$$

In the mentioned formula the rawTemperature is CAN value received from the sensor, sensor has a range of – 45° C – 105° C. The CAN message is a 12 – bit ADC value (0 - 4095). Offset of -45 ensures the correct range.

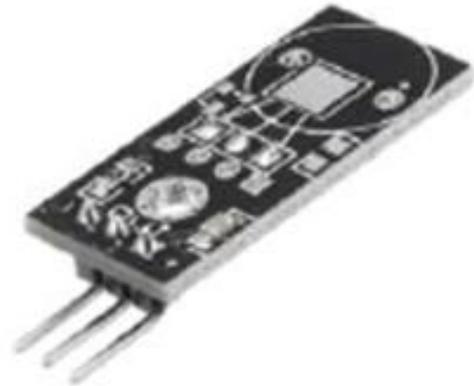


Fig.3. Temperature sensor

Table 2 Specifications of LM32D

Sr. No.	Parameter	Value
1.	Temperature Calibration	Calibrated directly in ° C
2.	Scale Factor	Linear + 10 mV/° C
3.	Accuracy	0.5° C Ensured Accuracy (at +25° C)
4.	Power Consumption	Less Than 0.3 mW
5.	Output Impedance Low Output	Low Output Impedance, 0.1 Ω (at 1mA current)

Engine RPM sensor: The engine RPM sensor monitors the crankshaft’s rotational speed by detecting specific teeth or markers on a rotating part, like a gear or flywheel. Each rotation generates a pulse signal proportional to the engine’s revolutions per minute (RPM). This data is sent to the vehicle’s microcontroller, which interprets RPM to aid in fuel injection, ignition timing, and transmission control. RPM sensors are essential for both manual and automatic transmission systems to ensure efficient gear shifting and optimal engine performance. The RPM

information is also displayed on the dashboard, providing the driver with real-time insights into engine activity. In proposed system engine RPM is done over CAN protocol. The equation (4) is used for the calculation of Engine RPM. The engine RPM range is assumed to be 0-8000 RPM and the raw value is scaled accordingly from 0-4095.

$$Engine\ RPM = \frac{rawRPM * 8000}{4096} \tag{4}$$

STM32F446RE (Microcontroller): The STM32F446RE is a high-performance microcontroller from STMicroelectronics, part of the STM32F4 series, tailored for digital power conversion, motor control, and advanced analog applications. It's based on a 32-bit ARM Cortex-M4 core running at 180 MHz and includes a Floating Point Unit (FPU) and Digital Signal Processing (DSP) instructions, making it ideal for precision and computationally intensive tasks. In [7] the detailed specification for this microcontroller can be observed.

DWIN Touch Screen Display : The DWIN Touch Screen Display with an 800x480 resolution is a versatile, high-quality display module commonly used in Human-Machine Interface (HMI) applications, including automotive, industrial, and home automation systems. It typically features a 7-inch TFT LCD panel with a capacitive or resistive touch screen that allows for intuitive, multi-touch operation, and supports gestures like swiping, pinching, and tapping. The Fig 4 and 5 shows the front and back side of DWIN screen respectively. In [10] the detailed information for this screen can be observed, table 3 shows the specification of the DWIN screen.



Fig.4. DWIN screen front



Fig.5. DWIN screen back

Table 3 Specification of DWIN screen

Sr No.	Parameters	Value
1.	Resolution	800×480pixels
2.	Baud Rate	3150~3225600bps
3.	Operating Voltage	4.5~5.5V, typical value of 5V
4.	Operating Current	1.8 A
5.	Operating Temperature	-10°C~60°C

CAN (Controller Area Network): The CAN protocol enables efficient, reliable communication among various sensors and modules within the vehicle. This protocol employs a dual-wire bus system, supporting multiple nodes, like sensors and the microcontroller, to communicate seamlessly. CAN is resilient to electromagnetic interference and includes built-in error detection mechanisms, ensuring data accuracy and integrity. By prioritizing messages, it maintains low latency for critical data transmission, which is crucial for real-time vehicle control systems. CAN enables centralized monitoring and diagnostics, simplifying maintenance and enhancing system integration, making it indispensable for modern automotive electronic architectures. The Fig 6 illustrates the CAN transceivers that can be used for the communication between the microcontrollers and different sensors.

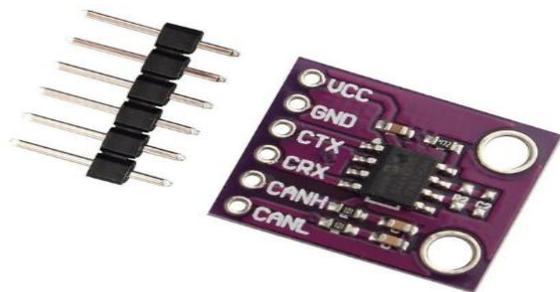


Fig.6. CAN transceiver

5. IMPLEMENTATION

a. Hardware Implementation :

The hardware used in integrated cluster development in HMI consist of the following components:

- STM32F446RE: To control and analyse the signal coming from the different type of sensors and give input to the display.
- CAN bus : It is used for the communication for the different MCUs preset in the car.
- Sensors : In proposed system different types of sensors are used to collect the real time raw data that is further evaluate, analyse and convert into the suitable form for the display input.
- DWIN screen: All the data coming from the sensors are integrated on this screen. It is responsible for the user interaction.

5.1 Microcontroller Flowchart (Hardware) Implementation:

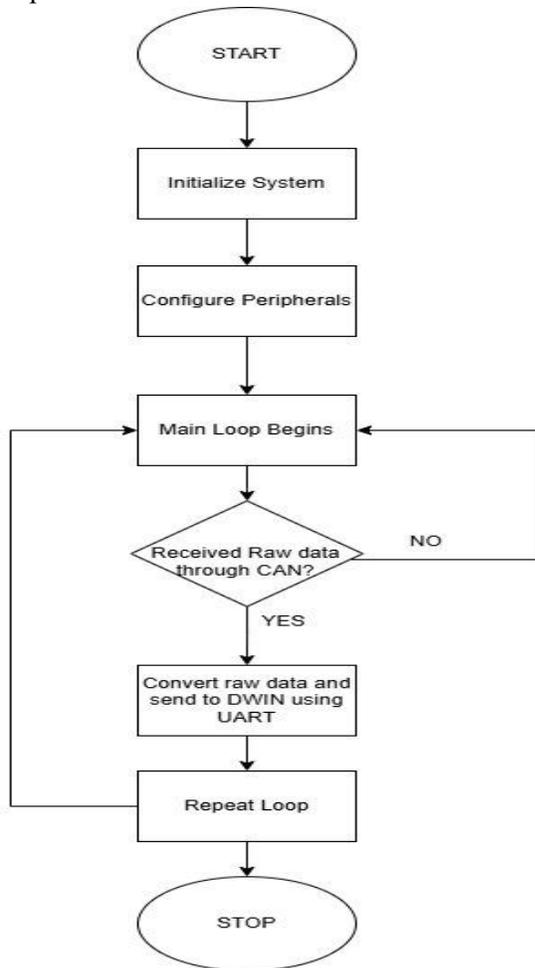


Fig.7.Flowchart of STM32F446RE execution

The Fig 7 visualise flow of STM32F446RE microcontroller execution.

CAN-to-UART Data Transmission Algorithm :

- 1 : Begin the system operation for execution of embedded system.
- 2 : Initialize the microcontroller and essential hardware components.
- 3 : The Controller Area Network module is configured to receive and transmit data by setting the baud rate, message filters and error handling mechanism.
- 4 : Configure and initialize the UART module for communication with DWIN display.
- 5 : Enter the continuous loop for real-time data processing.
- 6 : Check if new row data is received via the CAN protocol. If data is available, proceed to step 7, otherwise continue monitoring the CAN bus.
- 7 : Send the processed data to the DWIN display using UART.
- 8 : Return to step 6 and continue processing new incoming data.

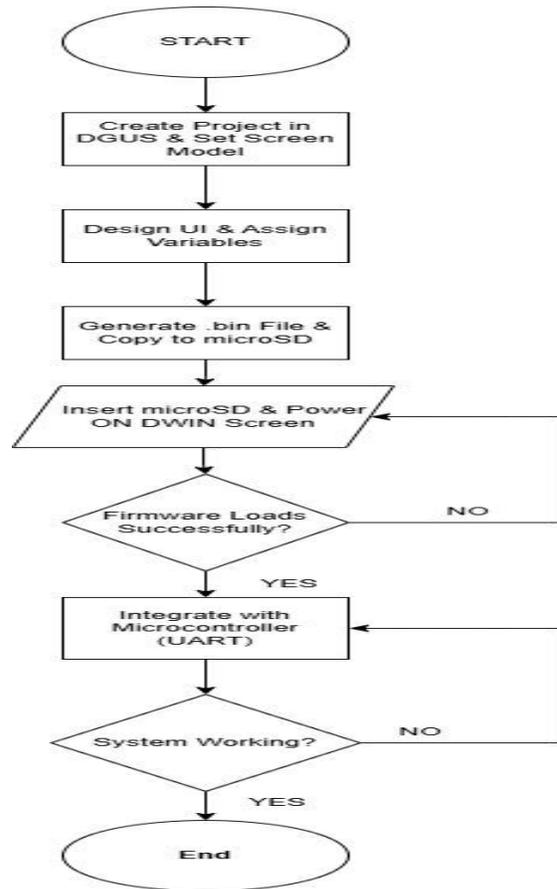


Fig.8. Flowchart for DWIN Screen

DGUS-Based UI Deployment Algorithm for DWIN Display :

- 1 : Start
- 2 : Create a Project in DGUS
- 3 : Design the User Interface, Arrange UI components such as buttons, images, and text fields.
- 4 : Compile the project to generate the .bin file. Copy the .bin file along with the DWIN_SET folder to a formatted microSD card.
- 5 : Insert the microSD card into the DWIN screen and power on the screen and wait for it to read the data.
- 6 : If the firmware loads successfully, proceed to the next step. If not, check the microSD card formatting and file structure, then retry.
- 7 : Connect the DWIN screen to the microcontroller using UART. Set up proper baud rates and communication protocols.
- 8 : If the system works as expected, proceed to the final step. If not, debug the microcontroller communication and modify configurations.
- 9 : Stop.

5.2 Software Implementation:

The software used in Integrated cluster development with HMI consisted of the following components:
 Cube IDE : Used to program STM32F446RE and implement the CAN bus for the hardware system.
 DGUS Run time environment : This run time environment is provided by DWIN for developing the graphical user interface (GUI) for the DWIN screen. It is user-friendly that helps user to make and load the graphics into the screen. It has virtual screen which helps user to see the results even if the screen is not available.

5.2.1- Screen Transition Diagram :

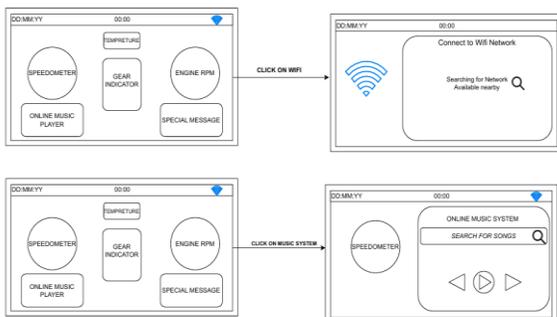


Fig.9. Screen transition diagram

Fig 9 illustrates the screen transition process. Since the screen is touch-enabled, each option displayed is interactive and can be clicked or tapped. The functionalities include a speedometer, odometer, gear indicator, online music streaming, and more. When a specific option is selected, the screen updates to display relevant functions. For instance, tapping the Wi-Fi symbol brings up a screen with detailed Wi-Fi information, while selecting the music player reveals additional controls such as play, pause, and song selection.

5.5 Circuit Diagram :

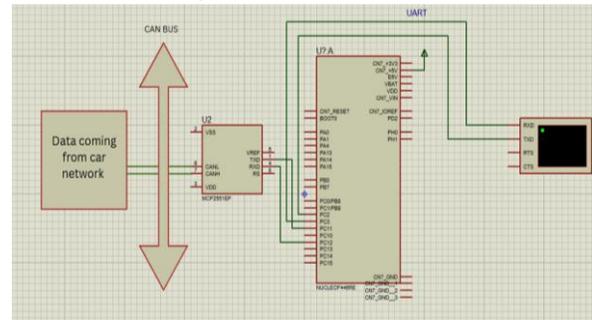


Fig.10. Circuit diagram for Integrated cluster development with HMI

This Fig 10 is circuit diagram for Integrated cluster development with HMI which utilizes actual sensors, MCP2551 CAN transceiver, STM32 microcontroller, and an HMI display. The block data coming from car network is responsible for acquiring raw sensor data from various vehicle sensors, including speed sensors, engine RPM sensors, gear position sensors, and temperature sensors. Since the sensors does not have built-in CAN support, an MCP2515 CAN controller is used along with the MCP2551 CAN transceiver to facilitate communication over the CAN bus as shown in Fig.10. The MCP2551 converts the digital data from the sensors into CAN signals and transmits them over the CAN_H and CAN_L lines to the STM32 microcontroller, which is equipped with a native CAN interface. The STM32 receives the raw CAN data, processes it by applying necessary calculations (speed in km/h, RPM conversion, gear detection, and temperature interpretation), and formats it for display. After processing, the STM32 forwards the final data to an HMI display via UART. The display presents real-time vehicle parameters such as speedometer, gear indicator, engine RPM, and temperature, enhancing the driver's dashboard experience. This system is designed

to ensure efficient communication between components, leveraging the CAN protocol for reliable and robust data exchange, making it suitable for automotive applications.

6. RESULTS AND DISCUSSION



Fig.11. Lock screen of HMI

The Fig 11 illustrate the lock screen of the HMI, which will be the very first screen displayed when the HMI is powered on. The "WELCOME" text is prominently displayed to greet the user. Additionally, on the left side of the screen, there are several icons, including home, settings, audio, and brightness controls, providing quick access to essential functions. This screen is fully customizable, allowing users to change the background, modify the welcome text, or add personalized elements according to their preferences. The flexibility in customization ensures that the HMI can be tailored to match different themes, branding, or user requirements, making it a versatile and user-friendly interface.

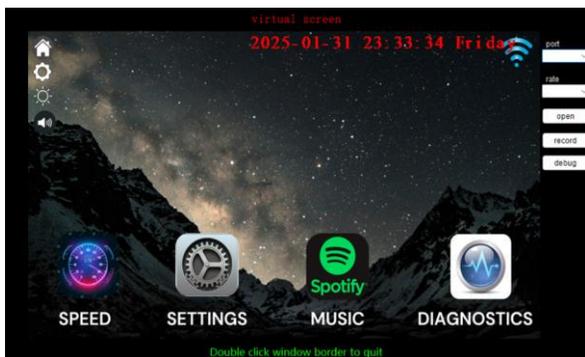


Fig.12. Home screen of HMI

The Home Screen is the primary interface from which users can navigate to various features. It displays essential dashboard functions such as speedometer, music, Wi-Fi, allowing quick access to each feature with just a tap. The time, date and temperature is varied accordingly and updated real time. The layout is designed for ease of use while driving, with large icons and a clear, organized arrangement.

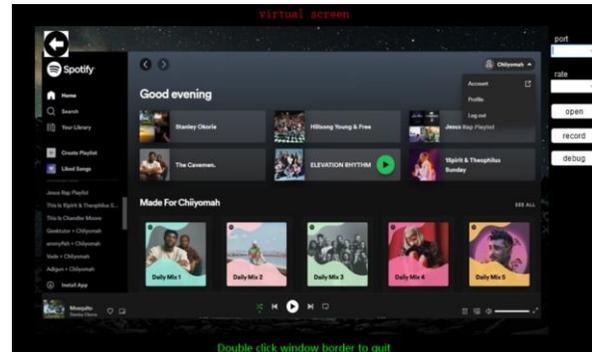


Fig.13. Window appeared after clicking on music

Fig.13 illustrates Music Window After selecting the Music option, users are taken to a dedicated screen for music controls. This screen includes options for play/pause, skip, and volume adjustment. It also connects to an internet-based music service if available, displaying album art, song title, and artist information for an enhanced entertainment experience.



Fig.14. Window appeared after clicking on Speedometer

Fig. 14 shows Speedometer Window. The Speedometer Window provides real-time speed data to the user. It is designed with a clear digital display, showing the vehicle's current speed in a prominent format. Additional elements, such as odometer, engine rpm and gear indicator also displayed to provide comprehensive driving metrics.

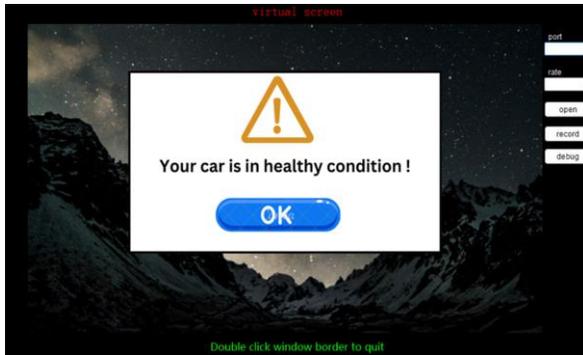


Fig. 15. Window appeared after clicking on Diagnostics

When the diagnostics option is selected, the HMI system performs a comprehensive check on the vehicle's health by analysing various parameters. If no issues are detected, a confirmation message appears, as shown in the image, stating "Your car is in healthy condition!" along with an OK button for the user to acknowledge. However, if the system detects any fault or malfunction, it will generate an alert message with details about the issue. These alerts help the user quickly identify and address potential problems, ensuring timely maintenance and enhanced vehicle safety.

7. FUTURE SCOPE

The proposed Integrated Cluster Development Using HMI in cars has immense potential for future enhancements to improve driver assistance, safety, and overall user experience. Some key advancements that can be integrated into future versions of this system include:

1. Advanced Gear Suggestions – Implementing an intelligent gear recommendation system based on vehicle speed, engine RPM, and road conditions to optimize fuel efficiency and performance.
2. Voice Control Integration – Enabling voice commands to control various vehicle functions such as navigation, music, climate control, and diagnostics, ensuring a hands-free and safer driving experience.
3. ADAS (Advanced Driver Assistance Systems) – Incorporating ADAS features like lane departure warning, collision detection, blind-spot monitoring, and adaptive cruise control, enhancing safety and driving comfort.

4. SOS Emergency System – Adding an automatic SOS alert system that detects accidents using impact sensors and notifies emergency contacts or rescue services with the vehicle's real-time location.
5. Artificial Intelligence (AI) for Predictive Maintenance – Utilizing AI and machine learning algorithms to analyze vehicle data trends and predict potential failures before they occur, reducing downtime and maintenance costs.

8. ACKNOWLEDGEMENT

We would like to extend our heartfelt gratitude to everyone who contributed to the successful completion of this project. First and foremost, we are grateful to the IEEE Smart City Lab at Sinhgad Academy of Engineering for providing us with access to advanced resources and facilities, which were instrumental in the development of our HMI system. A special thank you to Mr. Vaibhav Patharikar for his belief in our vision and for the financial support that made this project possible.

Finally, we would like to acknowledge the invaluable guidance and encouragement of our project guide, Dr. Rakhi Khedkar and Prof. Seema Borawake for their expertise, advice, and unwavering support were essential to navigating the challenges of this project. This accomplishment is a testament to the collective efforts and dedication of all those involved.

9. CONCLUSION

This integrated HMI system enhances automotive dashboards by consolidating essential functionalities such as speed, RPM, temperature, gear position, and odometer readings into a single, user-friendly interface. Built using an STM32 microcontroller, various sensors, and a DWIN touchscreen, it ensures real-time, reliable data communication via the CAN protocol. The CAN-to-UART Data Transmission Algorithm enables smooth data acquisition and display, while the DWIN Screen Initialization Algorithm ensures proper UI loading and seamless touch-based interactions. Additionally, Wi-Fi-enabled music streaming enhances entertainment by providing connectivity for online music services. The system also contributes to reducing vehicle weight by replacing traditional analog clusters

with a lightweight digital interface, which in turn improves fuel efficiency and increases mileage. With DGUS runtime simplifying UI design and implementation, this project highlights the potential for smart, connected vehicle dashboards that prioritize driver convenience, safety, and an enriched user experience.

[10] <https://www.dwin-global.com/t5l-uart-lcd-7-0inch-module-product/>

REFERENCE

- [1] M. Wang, J. Parker, N. Wong, S. Mehrotra, S. C. Roberts, W. Kim, A. Romo, and W. J. Horry, "Human-Machine Interfaces and Vehicle Automation: The Effect of HMI Design on Driver Performance and Behaviour," *AAA Foundation for Traffic Safety*, vol. 67, no. 3, pp. 450-461, July 2023.
- [2] M. Monika, "Advanced Instrument Cluster Development with Odometer and Speedometer Using STM32H735IGDK Controller," *R.V.R & J.C College of Engineering, Guntur, India*, vol. 4, no. 1, pp. 152-159, June 2022.
- [3] P. M. Knoll, "An Integrated HMI Concept for Driver Information and Driver Assistance Systems," *University of Karlsruhe, Faculty of Electrical Engineering*, vol. 19, no. 2, pp. 88-102, 2021.
- [4] EMBITEL, "HMI Development for Two-Wheeler Digital Instrument Cluster," *Embitter IoT Case Studies*, vol. 1, no. 1, pp. 12-19, [Online]. Available: <https://www.embitel.com/iot-casestudies/hmi-development-for-two-wheeler-digital-instrument-cluster>.
- [5] ResearchGate, "Example of an Instrument Cluster Development Process Between OEM and Suppliers," *ResearchGate*, vol. 8, no. 4, pp. 214-220, 2021. [Online]. Available: https://www.researchgate.net/figure/Example-of-an-instrument-cluster-development-process-between-OEM-and-suppliers_fig1_259688216.
- [6] IEEE, "HMI Development in Instrument Cluster Systems," *IEEE Xplore*, vol. 51, no. 5, pp. 972-979, Sept. 2020. [Online]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9179697>.
- [7] <https://www.st.com/resource/en/datasheet/stm32f446re.pdf>
- [8] <https://www.alldatasheet.com/datasheetpdf/pdf/840761/ELM-TECH/ELM32D548A-S.html>
- [9] <https://ww1.microchip.com/downloads/en/devicedoc/20001667g.pdf>