High level Design Architecture for EV Vehicle Remote Monitoring System

Introduction:

The purpose of this document is to present a comprehensive solution for the implementation of the **Chipmonk Electric Vehicle Remote Monitoring System**, specifically tailored for Heavy Transport Vehicles (HTV) such as passenger buses and cars. In the rapidly evolving automotive industry, the integration of electronic control systems has become increasingly prevalent. This proposal outlines how recent advancements in battery technologies, power electronics, communications, and intelligent control systems can be harnessed to enhance the monitoring and performance of Electric Vehicles (EVs).

Background:

Modern vehicles, including Electric Vehicles, feature intricate electronic control systems. These systems play a crucial role in monitoring and managing various vehicle subsystems and Electronic Control Units (ECUs). As part of the standard equipment, all vehicles are equipped with an On-Board Diagnosis (OBD) system. The OBD system is essential for the continuous monitoring and diagnosis of different vehicle components.

In the event of a malfunction or anomaly detected by the OBD system, a Malfunction Indicator Light (MIL) on the instrument cluster is illuminated. This serves as an indicator to the driver, notifying them of an existing problem within the vehicle. However, in the context of Electric Vehicles, additional monitoring and remote diagnostic capabilities can significantly enhance overall vehicle management and efficiency.

Objectives:

The primary objectives of implementing the Chipmonk Electric Vehicle Remote Monitoring System for HTV passenger buses and cars are as follows:

1. **Enhanced Remote Monitoring:** Enable real-time monitoring of crucial parameters, including battery status, motor performance, and overall vehicle health.
2. **Proactive Maintenance:** Implement a predictive maintenance system that utilizes data analytics to anticipate potential issues, reducing downtime and maintenance costs.
3. **Optimized Performance:** Utilize intelligent control systems to optimize the performance of the electric vehicle, considering factors such as energy efficiency and environmental impact.
4. **Communication and Connectivity:** Establish robust communication channels to facilitate seamless data transfer between the vehicle's electronic systems and a central monitoring platform.
5. **User-Friendly Interface:** Develop a user-friendly interface for both drivers and fleet managers to access relevant vehicle information, diagnostics, and performance data.
6. **Security:** Implement robust security measures to safeguard the integrity and confidentiality of the data transmitted between the vehicle and the central monitoring system.

Shruti Manisha and Team Below Components are on assumption that On-Board Monitoring Unit (OMU) Hardware will be available and should be Equipped with Control Area Network(CAN-BUS) instrument.

Components of the Chipmonk Electric Vehicle Remote Monitoring System:

The proposed system will consist of the following key components:

1. **On-Board Monitoring Unit (OMU):** A dedicated hardware unit installed in the electric vehicle to collect and process real-time data from various sensors and ECUs.
2. **Communication Module:** Enables secure and reliable communication between the vehicle and the central monitoring platform. Utilizes both cellular and Wi-Fi connectivity.
3. **Central Monitoring Platform:** A cloud-based platform that aggregates and analyzes data from multiple vehicles, providing insights to fleet managers and service technicians.
4. **User Interface:** Intuitive dashboards and mobile applications for both drivers and fleet managers to access real-time information, diagnostics, and historical performance data.
5. **Data Analytics Engine:** Utilizes machine learning algorithms to analyze data patterns and predict potential issues, enabling proactive maintenance.

THE REMOTE MONITORING PLATFORM

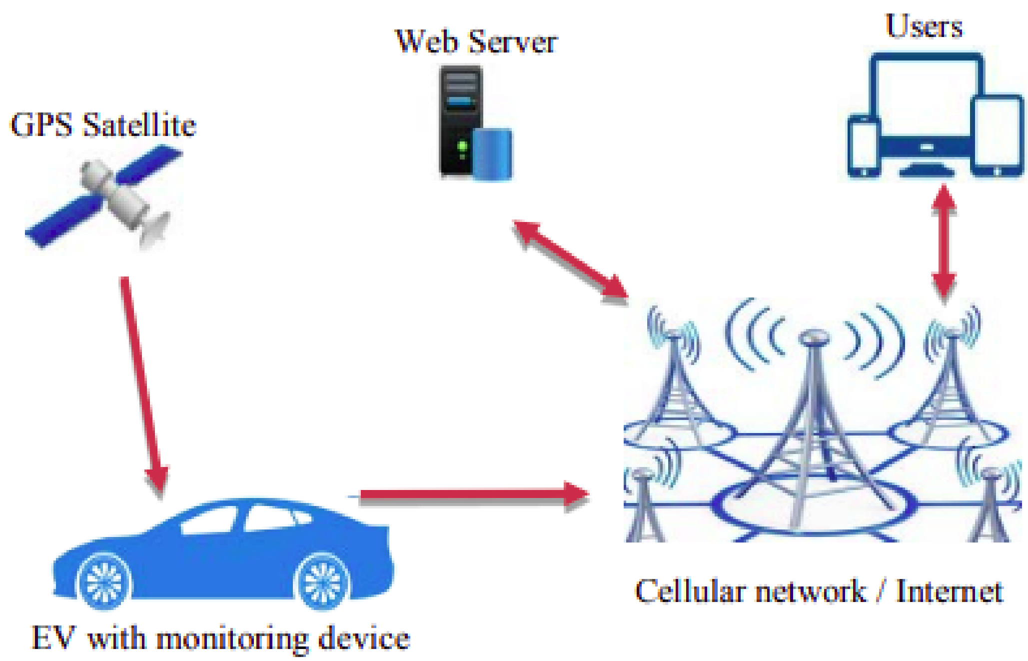


Fig : 1

The GPS satellite transmits signals that allow the monitoring device EVCANMon embedded in the EV to know its location. Also, the monitoring device is connected to the CAN network of the EV and collects data from different ECUs of the vehicle: the controller of the electric motor, the BMS and the charger (see Fig: 2). Then, all the gathered data are transmitted every minute or less via a local cellular network to a database located on a web server. Finally, the users can access the data online from the server using web browsers on their desktop computers, laptops or mobile devices and they can track the EV in real-time.

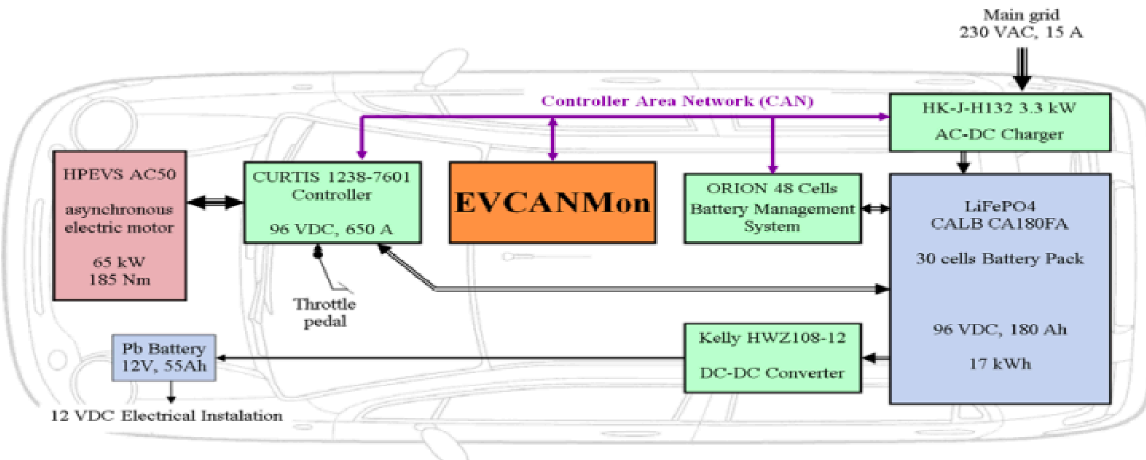


Fig: 2

Fig. 2 illustrates the architecture of the testing platform for EVs, which is installed in the laboratory and in which the monitoring device EVCANMon is integrated .

CAN Network Setup

The CAN network of the EV testing platform consists of four CAN nodes (motor controller, charger, BMS and monitoring device EVCANMon) connected using a pair of twisted wires CAN-H/CAN-L, see Fig. 3. Because the CAN bus is bidirectional, both ends of the bus must be terminated with 120 Ω resistors. The bus speed is set to 250 kbps.

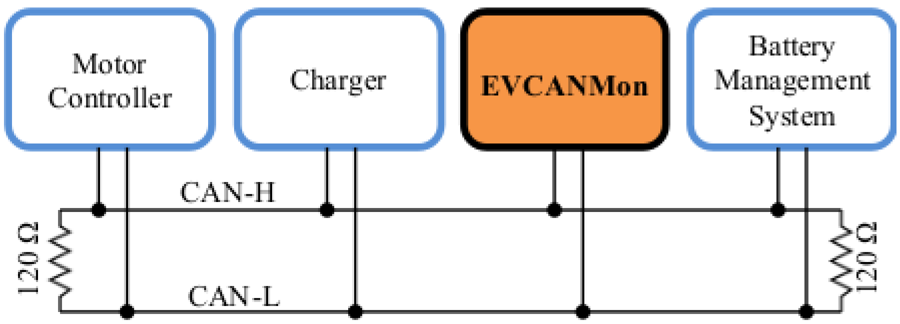


Fig:3

A Hardware Implementation

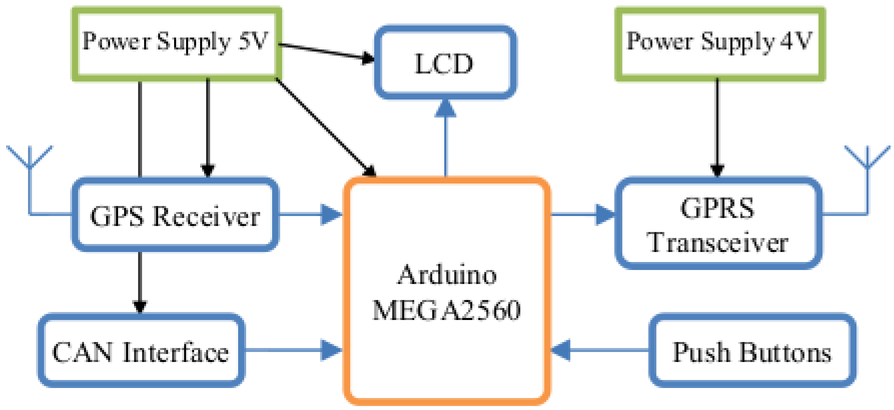


Fig:4

A Software Implementation

The software for the microcontroller allows the user to control the device and to select different options based on the menus displayed on LCD. Also, it is responsible for the communication with the GPS receiver and CAN interface to get the data, printing it on LCD and sending it to the web server. Secondly, the application displays to the user in his web browser different pages with data from the database using tables and graphs. Once the monitoring device is powered on, the main menu is displayed on LCD with the following options: Local Monitoring, Sending data to server, and Sending SMS to user.

Web Base Monitoring System

The developed web application is based on more PHP scripts, combined with HTML and Java scripts(Our choice of comfortable software). Preferred (LAMP) Linux , Apache,Mysql,PHP

Step 1:
Firstly, the application assures the reception and storage of data transmitted by the embedded monitoring device. The application connects to the MySQL database using the server name and user name and password, inserts a new row in the database with the received data, and closes the connection.

Step 2:
Secondly, the application displays to the user in his web browser different pages with data from the database using tables and graphs.

The main web page has three buttons: **Google Maps, Parameters,** and **Graphs**. If the user presses one button, then the corresponding page is displayed.

Google Maps:

For the Google Maps page(Fig:5) the application interrogates the MySQL database, extracts latitude and longitude, and places the markers accordingly on the map using Google Maps API and Java scripts.

This will display in LCD Monitoring Screen and webpages

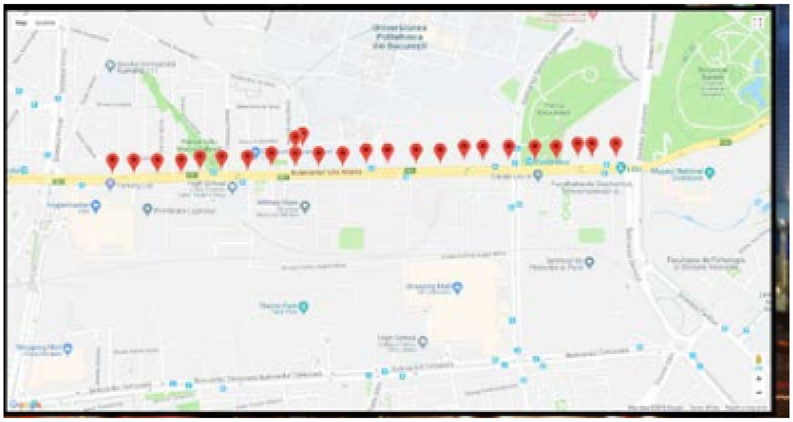


Fig:5

Parameters:

For the Parameters page the application requires the user to introduce a time interval and then displays a table with the acquired data from that interval. HTML script is used to draw the table.

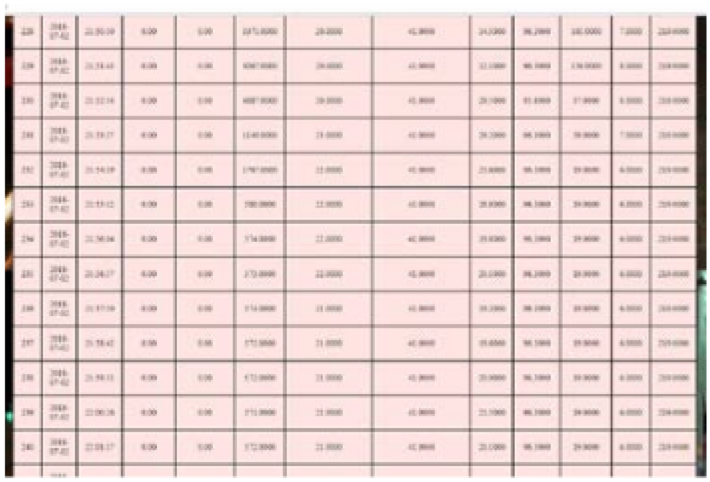


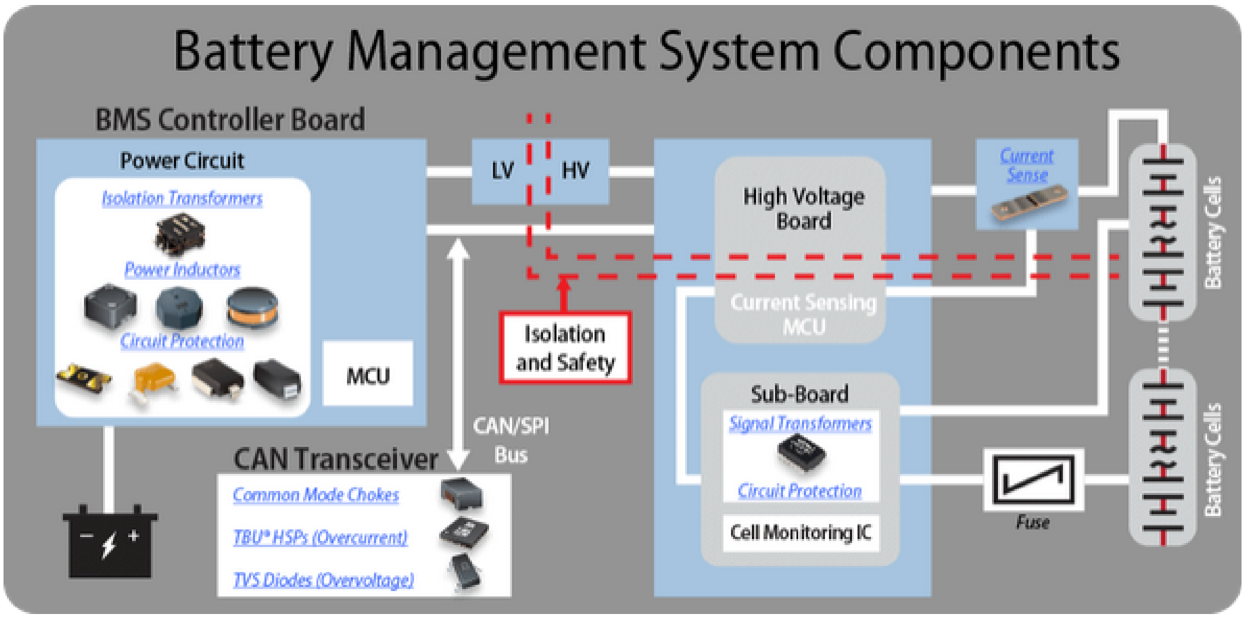
Fig:6

Graphs:

For the Graphs page (Fig:7) the application requires the user to choose from the top buttons which graphics must be displayed: motor speed, controller temperature, motor temperature, and battery voltage. Java script is used to visualize the data graphs

Fig:7

1. **Battery Management System**



BMS is used to monitor the state of the battery and is responsible to take the necessary measurements i.e. SOC, SOH. BMS performs cell balancing to deliver the best efficiency output from a battery pack, and a small ECU is used in this for communication with other components.
Apart from these important components, there are multiple hardware and software used in EV powertrain architecture. There are small monitoring ECUs placed for the specific function and communication is done by CAN protocol.

Block Diagram: