IOT BASED PREDICTIVE MAINTENANCE FOR EVs

PARVESH KUMAR. P

Problem Statement

The rapid global transition toward electric vehicles (EVs) is hindered by concerns regarding their safety and reliability—particularly battery overheating issues that have led to incidents of fire and explosion. These safety concerns, coupled with a lack of awareness among users on how to maintain EVs based on environmental conditions, usage patterns, and early damage indicators, have slowed the adoption of electric vehicles. As a result, the EV market is not growing in sync with the increasing energy demand and environmental expectations.

Proposed Solution

This project proposes an IoT-based predictive maintenance system for EVs by integrating multiple sensors to monitor critical parameters like:

- Battery voltage and efficiency
- Heat emission from key components
- Physical wear and tear (especially tyre health and mechanical vibration)

Sensors such as a DHT22 (temperature and humidity), potentiometer (voltage simulation), and vibration sensors are interfaced with an ESP32 microcontroller. These sensor readings are transmitted over WiFi to the ThingSpeak IoT platform for live monitoring and cloud analysis. Additionally, a digital twin of the EV is developed using Blender and Unity to simulate the real-time behaviour of the vehicle and help visualize anomalies or faults.

Project Description

The project implements a complete predictive maintenance architecture for EVs, starting from real-time data acquisition using embedded systems to visualization using digital twin technology. The system monitors environmental and mechanical parameters, triggers alerts when unsafe thresholds are breached, and uploads sensor readings to a cloud dashboard.

Methodology

1. **Sensor Deployment**

    - **DHT22**: Measures temperature and humidity levels (detects battery overheating).
    - **Potentiometer**: Simulates EV battery voltage.
    - **Vibration Sensor**: Detects abnormal mechanical vibrations (indicative of engine or wheel faults).
    - **Tyre Sensor (Analog Input)**: Simulates tyre wear and tear condition.
    - **Pushbutton**: Manually simulates fault triggering for emergency alert conditions.
2. **Microcontroller and Networking**
    - **ESP32**: Central control unit for acquiring sensor data and transmitting it via WiFi.
    - **WiFi Communication**: Uses ESP32's built-in WiFi module to connect to the internet.
3. **Data Handling and Cloud Integration**
    - **ThingSpeak**: Used for data storage, live monitoring, and visualization of sensor readings through dedicated fields.
    - **Field Mapping**:
        - Field 1 – Temperature (°C)
        - Field 2 – Battery Voltage (V)
        - Field 3 – Vibration level
        - Field 4 – Tyre health
        - Field 5 – Simulated mileage (seconds since start)
        - Field 6 – Fault flag
4. **Digital Twin**
    - **Blender**: Designed a 3D model of the EV vehicle and its environment.
    - **Unity**: Used to create an interactive simulation, linking real-time sensor data with 3D behaviour.

Tools and Portals Used

Component	Purpose
Wokwi	Embedded system simulation (ESP32, sensors)
ThingSpeak	Cloud-based IoT data monitoring and visualization
Blender	3D modelling for Digital Twin representation
Unity	Real-time simulation with data-driven interaction
Arduino IDE	Programming and code deployment for ESP32
Libraries Used	`DHTesp`, `WiFi.h`, `ThingSpeak.h`

Hardware and Software Components

- ESP32 Development Board
- DHT22 Temperature & Humidity Sensor
- Analog Vibration Sensor
- Potentiometer (voltage simulator)
- Pushbutton (manual fault trigger)

- OLED Display (optional for real-time alerts)
- Wokwi Platform for simulation
- ThingSpeak for real-time IoT cloud dashboard
- Blender + Unity for digital twin development

Expected Outcomes

- A live monitoring system for core EV parameters
- Real-time alert system for overheating or other fault conditions
- Predictive insights based on historical sensor data
- Digital twin model reflecting physical status of the EV

Future Scope

- Integration with Machine Learning to predict component failure
- Expansion to include GPS and driving pattern analytics
- Deployment in real-time physical EV prototypes
- Mobile app-based alert and dashboard system for users