1. For OBD2 Project we can use LM35 as Temperature sensor to measure the ambient temperature inside or outside the vehicle. This information can be used for climate control systems and to provide data for the vehicle's ECU.

LM35 can also be used for Engine Temperature or Battery Temperature Monitoring

1. Potentiometer can be used as

**Throttle Position Sensor (TPS):**

A potentiometer can be used as a throttle position sensor to measure the position of the throttle valve. This information is critical for the ECU to control the air-fuel mixture and engine performance.

**Pedal Position Sensor:**

Potentiometers can be used in accelerator and brake pedals to detect the position of the pedal. This data is used by the ECU to manage acceleration and braking.

**Abstract**

The increasing complexity of modern vehicles necessitates the use of multiple Electronics Control Units (ECUs) to manage and monitor various functions. Communication between these ECUs is facilitated by the Controller Area Network (CAN) protocol, which handles the Physical and Data Link layers of the OSI model but lacks advanced diagnostic capabilities. To address this, standardized diagnostic protocols such as On-Board Diagnostics (OBD2) have been developed, providing a comprehensive framework for fault detection and in-vehicle communication. This paper presents the design and implementation of the OBD2 protocol on an embedded system using STM32F407 microcontrollers. The project aims to establish a diagnostic communication system between ECUs and Tester Tool. Primary function of this ECU is to measure ambient temperature and Throttle Position. ECU and Tester tool are connected via a CAN bus. The implementation leverages the ISO 15031 standard to ensure reliable and standardized diagnostic services.

**Keywords**: On-Board Diagnostics (OBD2), ISO 15031, Electronic Control Unit (ECU), Body Control Module (BCM), Light Control Module (LCM), Controller Area Network (CAN), Diagnostic Trouble Code (DTC), Routine Control Identifier (RID), Data Identifier (DID), ISO-TP (Transport Protocol

**Introduction**

The complexity of modern vehicles necessitates advanced diagnostic capabilities to ensure optimal performance and reliability. The On-Board Diagnostics (OBD2) protocol, standardized under ISO 15031, is a crucial tool in the automotive industry for monitoring and diagnosing vehicle systems. OBD2 provides a comprehensive framework for fault detection, emission control, and real-time data access, facilitating efficient vehicle maintenance and repair.

This paper explores the design and implementation of an OBD2-based diagnostic communication system using STM32F407 microcontrollers. The project involves two key component one is ECU and another is PC Based tester tool. ECU and Tester tool are communicate over a CAN bus and respond to OBD2 diagnostic requests. **The system supports vital OBD2 services, including retrieving Diagnostic Trouble Codes (DTCs), monitoring vehicle emissions, and accessing real-time sensor data. Mode (0x01)**

To ensure the system's robustness, fault injection techniques are employed to simulate real-world ECU failures such as over Temperature CAN bus disconnection, ECU power loss. The diagnostic responses to these failures are analyzed using a Waveshare USB-to-CAN module and Python-based OBD2 test scripts. By implementing OBD2 on embedded automotive ECUs, this project aims to demonstrate the practical application of standardized diagnostic services, enhancing the efficiency of vehicle fault detection and maintenance processes.

**System Overview**

1. **OBD2 Protocol Overview**

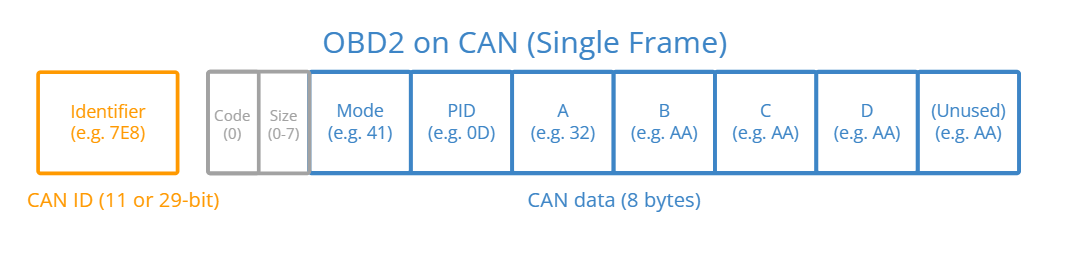
On-Board Diagnostics II (OBD2) is a standardized system implemented in modern vehicles to monitor and diagnose various aspects of vehicle performance and emissions. Defined under the ISO 15031 standard, OBD2 provides a comprehensive framework for fault detection, real-time data access, and emission control, ensuring vehicles meet regulatory requirements and maintain optimal performance. The OBD2 protocol enables communication between the vehicle's Electronic Control Units (ECUs) and diagnostic tools via the Controller Area Network (CAN) bus.

1. **Client-Server Architecture**

 The OBD2 protocol operates on a client-server architecture, where the diagnostic tool acts as the client and the vehicle's Electronic Control Units (ECUs) act as servers. Client initiates diagnostic request, and Server will respond to the request as per configuration. The diagnostic tool provides a user-friendly interface for mechanics and technicians to interact with the vehicle's diagnostic system. This interface allows users to select specific tests, view data, and perform diagnostic procedures. Client-Server architecture provide standardization, Efficiency, Scalability and Flexibility.

1. **OBD Message Structure**

OBD2 message consist of CAN ID, Mode, PID and Data as per illustrated in diagram.



CAN ID: OBD2 Protocol support both 11-Bit CAN Identifier and 29-Bit Identifier. OBD2 Protocol