Tri-nit Hackathon

ECO-DRIVING DRIVER ASSISTANCE SYSTEM

DATE: 12/02/2023

TEAM NAME: Jobbers

TEAM MEMBERS:

1. Mohan V
2. Soorya S
3. Taushiq B

IOT01

Topic: ECO-DRIVING DRIVER ASSISTANCE SYSTEM

PROBLEM STATEMENT:

Vehicles have a major contribution to the total energy demand as well as to harmful emissions. In light of this, an important factor in creating sustainable cities is the reduction of energy required to power automobiles. A method of achieving the same is eco-driving. Eco-driving is a term used to describe the energy-efficient driving of vehicles. By this principle, a driver follows certain principles while driving that help him reduce the total energy consumed. In this problem statement, you are required to design a driver assistance system that will remind a driver to follow eco-driving principles when a certain principle is violated. The driver assistance system must look out for the following principles.

● The Cruising Speed of the vehicle must be around 90 km/h

 ● Acceleration shouldn't exceed 1.4705 m/s2

 ● If the vehicle is idling (Idling is a working condition in which the engine is idling but does not output power, resulting in fuel consumption) for more than half a minute, recommend that the engine be switched off.

● The number of braking must be kept minimal.

● Detect obstacles ahead and suggest an optimal deceleration

● Track the location of the vehicle. The driver assistance system must, at fixed intervals of time, transfer the vehicle's above-mentioned parameters to a remote server. After processing the parameters, the remote server must alert the user when a principle is being violated on an application or make the suggestions mentioned above. (The application can be a simple python client application). The more useful suggestions the driver assistance system can make, the better.

The driver assistance system must, at fixed intervals of time, transfer the vehicle's above-mentioned parameters to a remote server. After processing the parameters, the remote server must alert the user when a principle is being violated on an application or make the suggestions mentioned above. (The application can be a simple python client application). The more useful suggestions the driver assistance system can make, the better.

HARDWARE COMPONENTS:

- ESP 32 microcontroller
- Ultrasonic Sensor HC SR04
- 16x2 LCD Display
- Mpu6050 6-Axis Accel & Gyro Sensor
- LED
- 2 Push Button

Circuit Diagram:
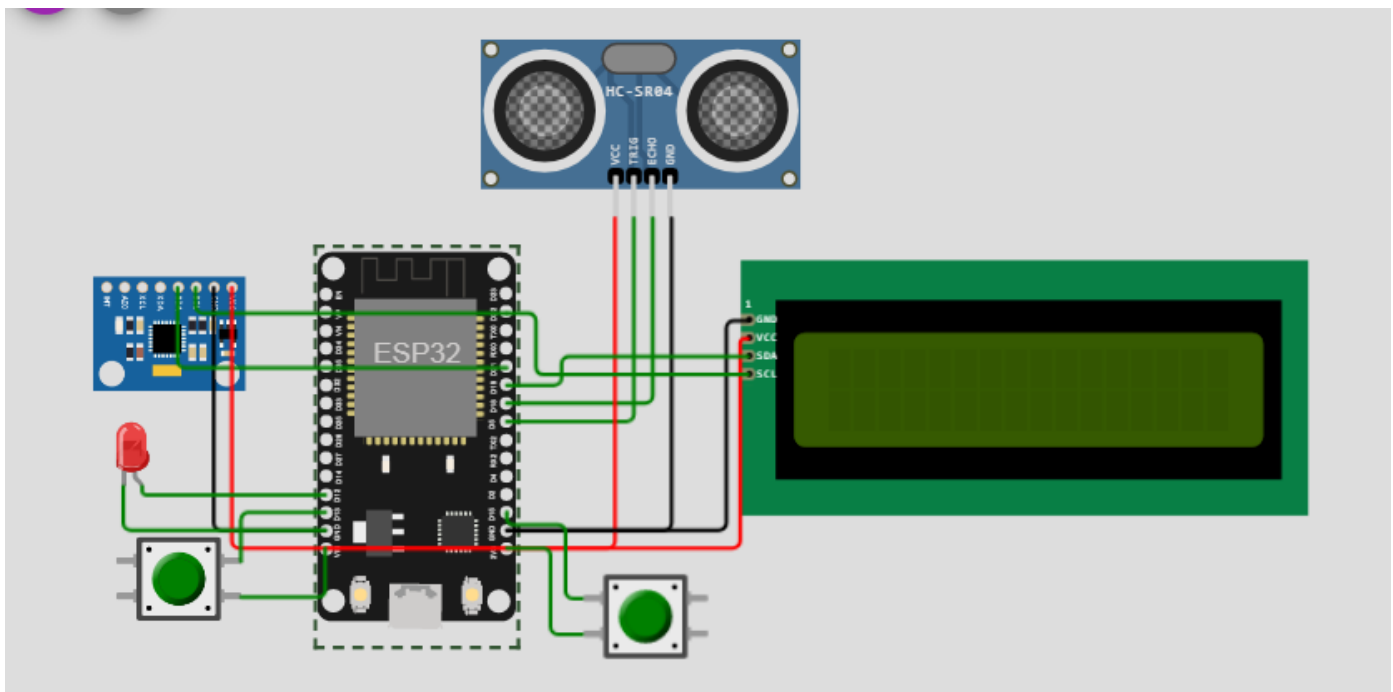
SOLUTION:

- We have employed acceleration from accelerometer to keep the cruising speed of the vehicle around 90 km/h.
- Accelerometer has been used to check that the acceleration doesn't exceed 1.4705 m/s^2.
- To indicate the idle state, a push button is placed under accelerator, alert message is sent to LCD saying "Switch off engine", if the vehicle is idle for more than half a minute.
- A push button is placed under brake, if it is being pressed for more than 5 times within 10 seconds, alert message is sent, This keeps the number of braking minimal.
- Ultrasonic Sensor HC SR04 has been used to detect obstacles ahead and suggest an optimal deceleration.

Cloud Server : Thingspeak (distance and acceleration data sent)