Autonomous embedded Evaluation Interfacing Stack Task

Amr Ismail

Objective:

This task is designed to evaluate your fundamental skills in embedded systems within the s. You are expected to design and implement a small, functional subsystem that simulates a a situation a vehicle can face.

Background:

In autonomous vehicles, subsystems are responsible for sensing the environment, processing information, and communicating with other vehicle modules. This task represents a simplified version of an obstacle detection and response module like those used in active safety systems.

Task Description:

Develop a system that:
Detects obstacles using a proximity sensor (e.g., ultrasonic sensor).
Triggers a response when an object is detected within a defined range.
Sends data (e.g., distance) through a communication protocol to the actuating system.

Requirements:

The system is composed of modules:

- One for sensing the outer environment which will be connected to:
  - Any proximity sensor such as Ultrasonic sensor to feedback the distance to an obstacle.
  - Potentiometer as that simply describes the measured motor speed.
- The other for the actuating system which will be connected to a motor simulating the vehicle's movement ensuring the speed don't exceed 80% of the maximum motor speed so that the vehicle must stop the movement as the obstacle is in a range of 20cm or less and an indicator to illustrate obstacle detected such as turning on an LED.
- Communication Protocol:
  - Use SPI communication protocol to transmit sensor data to the actuating system.
  - Use I2C communication protocol to feedback motor speed to the actuating system.

Bonus (Optional):

- Use a display as LCD to show live distance and system status.
- Add servo control to simulate automatic steering adjustment.

Allowed Platforms:
You may use any embedded platform of your choice:

- Arduino (UNO, Nano, Mega, etc.)
- ESP32 / ESP8266
- STM32 (Blue Pill, Nucleo, etc.)
- Simulators (Proteus, Tinkercad, etc.)

Task Handing:

Submit a ZIP file containing with your name for example **"Mohamed_Ahmed_embTask.zip"** containing:

- Short demo video or screen recording showing the working setup.
- The simulation file and code.

Deadline: 3 days from the day of submission.

Notes:

- Focus on clarity, functionality, and system thinking.
- This task is not about complexity as you can see, but about how well you can structure and implement a clean embedded solution that simulates real automotive behaviour.