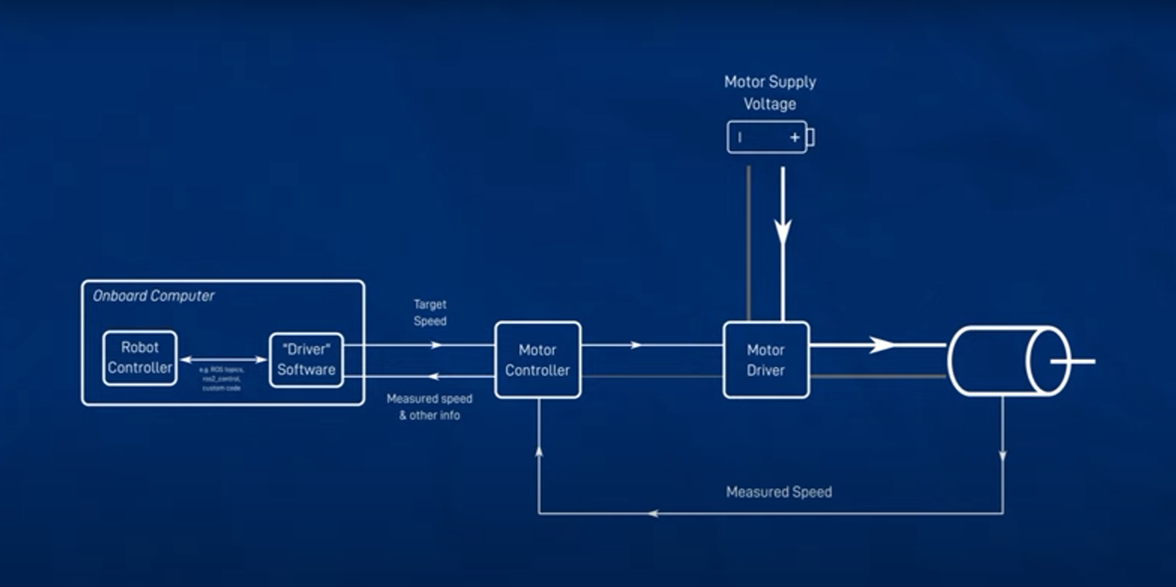
Functional Block diagram:

A diagram of a system

Description automatically generated

Block diagram for the motors:



Suggested solution:

* Raspberry Pi:

Utilize a Raspberry Pi board (Raspberry Pi 4) for its versatility and GPIO (General Purpose Input/Output) pins, enabling easy interfacing with sensors and motor controllers. Raspbian OS or a specialized robotics operating system like ROS (Robot Operating System) can be used.

* Ultrasonic Sensor:

Implement an ultrasonic distance sensor (HC-SR04) for obstacle detection. Use the sensor's echo and trigger pins to measure distances accurately. Implement a simple distance calculation algorithm to detect obstacles within a certain range.

* Infrared Sensor:

Use infrared (IR) sensors (TCRT5000) for line following. The IR sensors can detect light reflection differences between the surface and the line. Implement a PID (Proportional-Integral-Derivative) control algorithm to adjust the robot's direction based on the readings from these sensors.

* Camera:

Utilize a Raspberry Pi Camera Module or a USB camera for capturing images or video frames. Implement computer vision algorithms, such as OpenCV, for object detection. You can use pre-trained machine learning models like YOLO (You Only Look Once) or Mobile Net for efficient object detection.

* DC Motor and Servo Motor:

Use a motor controller board (L298N) to control the DC motor for propulsion and the servo motor for steering. Implement PWM (Pulse Width Modulation) control for precise motor movements. PID control can also be applied to the DC motor to maintain a steady speed and direction.

* Power Supply:

Select a battery with the suitable voltage and capacity to meet the power requirements of the Raspberry Pi, motors, and other components, ensuring it can deliver enough current for all components to operate effectively.

First Optimal Solution:

* Line Following:

Sensor: Utilize two infrared sensors mounted on the front of the car to detect the line.

Control Algorithm: Implement a PID control algorithm in the Raspberry Pi to adjust the speed and direction of the DC motor's wheels based on the data from the infrared sensors. This algorithm will keep the car centered on the line.

* Obstacle Detection and Avoidance:

Sensors: Incorporate an ultrasonic sensor mounted at the front of the car, facing forward, and use the Raspberry Pi to process its data.

Control Logic: When the ultrasonic sensor detects an obstacle within a specified range (e.g., 20 cm), trigger an obstacle avoidance routine in the Raspberry Pi:

Stop the DC motor.

Activate the servo motor to steer the car away from the obstacle.

After obstacle avoidance is complete, resume line following.

* Object Detection:

Camera: Use a Raspberry Pi Camera module to capture images of the surroundings.

Software: Employ a pre-trained YOLOv3 (You Only Look Once) neural network model for object detection and

process the data on the Raspberry Pi in real-time.

Action: If an object of interest (e.g., a person or another vehicle) is detected within the camera's field of view, take appropriate actions, such as slowing down or stopping the car, and notify the user through the remote interface.

* Integration and Control:

Raspberry Pi: Develop a Python-based software application running on the Raspberry Pi that integrates sensor data, camera input, and control algorithms.

Feedback: Implement feedback loops for each component (line following, obstacle avoidance, object detection) to ensure responsive and adaptable behavior based on real-time data.

* User Interface and Communication:

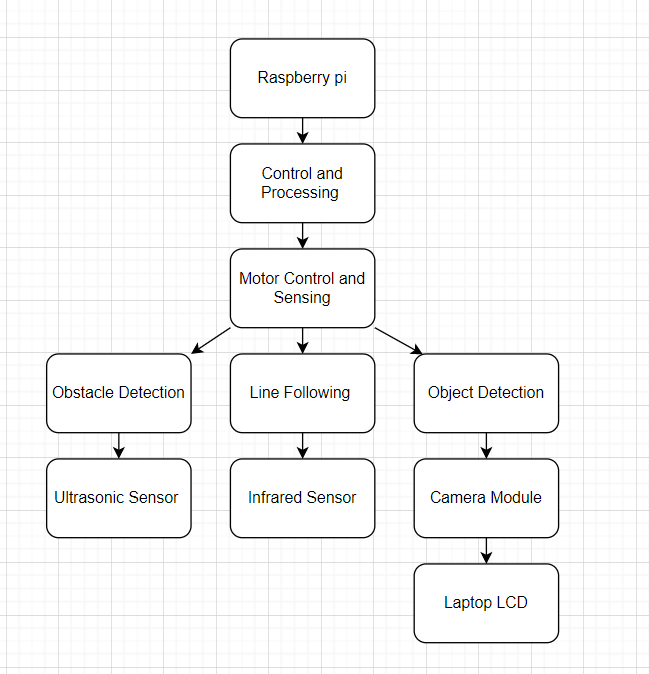
Web Interface: Create a web-based user interface hosted on the Raspberry Pi, accessible via Wi-Fi. This interface will display the camera feed, sensor data, and control options.

Communication: Use WebSocket communication for real-time interaction with the car. Allow users to start and stop the car, change its behavior, and receive updates on its status via the web interface.

* Power Supply:

Battery: Utilize a high-capacity lithium-ion rechargeable battery pack to power the Raspberry Pi, motors, and other components.

General block diagram representation:



Sensors and actuators to choose from:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Function | Solution 1 | Solution 2 | Solution 3 | Solution 4 |
| Switching system on, off | Classic on off switch | Voice activation | Touch activation |  |
| User Interface | Laptop Touch screen | GUI | Keypad | Switches |
| Power | DC (12v -Battery) | AC |  |  |
| Control System | manual | Analog | Microcontroller (STM32 inside 4WD expansion board) |  |
| Actuator | Electric DC motor |  |  |  |
| Drive | Analog H-Bridge | IC -H-Bridge | 4WD expansion board |  |
| Path detection | LDR-LED | Color sensor | Programmed to move in a specific path |  |
| Object recognition | Camera |  |  |  |
| Obstacle detection | Ultrasonic sensor | Infrared sensor | Proximity sensors |  |
| Speed sensor | Tachometer |  |  |  |
| Manual control | Switches | Joystick | Wireless from laptop |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |