SmartTupBot: A Tupper Mobile Robot with Manual Control, Obstacle Avoidance, and Wall Following Capabilitie

Marianthi Thodi

November 2023

Contents

1 Abstract

This paper examines the control of a robotic car using an Arduino, Bluetooth, and two sensors. The user can interact with the robot through an application, activating, deactivating, and selecting functions by pressing buttons. Available functions include moving forward, backward, left, and right. Additionally, an "Avoid Obstacles, Wall Follow" function is provided, allowing the robot to autonomously avoid obstacles during its movement and also follow walls while simultaneously avoiding obstacles. The

work combines ease of use with the robot's autonomy.

2 Introduction

Robotics is experiencing impressive growth, becoming a significant branch of modern technology. A particularly interesting area of development is in the field of robotic cars, where the convergence of various technologies creates a flexible system that offers utility. The functionality of the robotic car allows the user to interact either through manual control or by activating the obstacle avoidance or wall-following function via Bluetooth. Built-in distance sensors enable the robot to dynamically adjust its route. This function allows the robot to maintain a safe distance from obstacles and walls while navigating its environment. Many such robotic cars are applied in environments where human access is impossible or dangerous, thus demonstrating their contribution to various fields of application. For example, in smart cities, these functions help robots to navigate efficiently along buildings, avoid obstacles in crowded areas, ensure pedestrian safety, and assist in traffic management. They also have applications in exploration, such as in difficult terrain, where obstacle avoidance allows for autonomous maneuvering, minimizing the risk of damage. Wall-following capabilities aid in adapting to the topography, allowing robots to navigate along structures and optimize paths in unknown terrains. Together, these capabilities contribute to the flexibility and adaptability of mobile robots in dynamic urban environments and uncharted areas.

3 Chapter 1 : Technological Implementation

Below are the components I used as well as a few details about each.

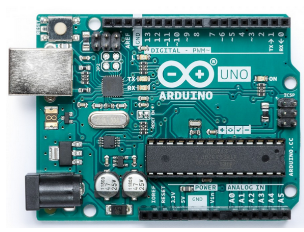
3.1 Introduction to Arduino

Arduino is an open-source platform for designing and controlling electronic projects. Two of its main advantages are its low cost, as a development board can cost as little as 10 euros, and its development environment. The development environment is written in Java, offering portability to more operating systems. Additionally, it provides a compiler for C and C++ programming languages, as well as a terminal for serial communication between the computer and the board.

3.1.1 Arduino Uno Rev3

It is a programmable board with the ATmega328P microcontroller as its central component, which acts as the "brain" of the system, controlling 14 digital pins, of which 6 support PWM output, as well as 6 analog pins. It also includes a reset button to reset the program, a connector for external power supply, and a USB port for code transfer and power supply. The 32KB flash memory stores the code, the 2KB RAM is used during execution, while the 1KB EEPROM memory retains data even after power off. Additionally, it operates with an external power supply in the voltage range of 6 to 20 Volts. Finally, Arduino boards function as microcontrollers, executing C/C++ programs stored in their hardware, without having their own operating system like the Raspberry Pi.

Figure 1: Arduino Chip



3.1.2 Arduino IDE

Figure 2: Development Environment

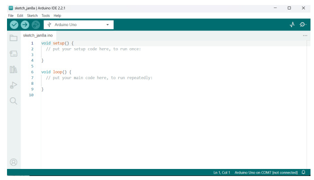


Figure 3: Checks the code for errors

Figure 4: Uploads the code to the arduino board

3.2 Ultrasonic Sensor HC-SR04

3.2.1 General

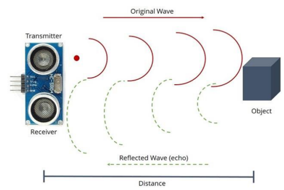
The ultrasonic sensor SR04 is an affordable and user-friendly sensor that uses ultrasound to measure distances. It consists of two ultrasonic transducers, a transmitter and a receiver. The transmitter generates ultrasonic pulses, while the receiver detects the pulses that return from objects. By measuring the time it takes for the pulses to return, the sensor calculates the distance between itself and the object. It has a detection range from 2 cm to 4 m and an accuracy of 3 mm.

3.2.2 Mode Operation

The operation of the sensor is based on measuring the time required for the return of the ultrasonic pulses. The equation in Figure 5 is used to calculate the distance, where the time is multiplied by the speed and divided by 2, since the time measured is for the sound wave to reach the obstacle and return. The operation of the sensor is illustrated in Figure 6.

Figure 5: Distance = Velocity * (Time/2)

Figure 6: Mode Operation Sensor



3.3 Motor Driver L293D

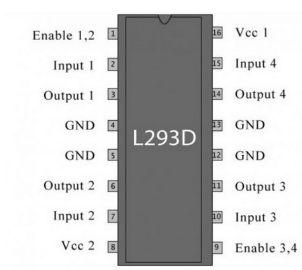
3.3.1 General for IC

A motor driver, such as the L293D, is like an amplifier that takes a low control signal from a microcontroller and generates a high current signal that can drive a DC motor. In general, the L293D is a motor driver that uses two half H-bridges, one for each motor. Each full H-bridge allows control of a motor's direction and speed. Therefore, with two full H-bridges, I can control two motors independently, each in its own direction and speed.

- Pin 1 (enable1,2): When this pin is HIGH or 1, the left side of the IC is enabled. When it is LOW or 0, it is disabled.
- Pin 2 (input 1): When it is HIGH or 1, output 1 becomes HIGH, allowing current to flow to motor 1.
- Pin 3 (output 1): Connects to one of the terminals of motor 1.
- Pins 4, 5 (GND): Connects to ground (GND).
- Pin 6 (output 2): Connects to one of the terminals of motor 1.
- Pin 7 (input 2): When it is HIGH or 1, output 2 becomes HIGH, allowing current to flow to motor 1.
- Pin 8 (vcc2): Provides the supply voltage for motor 2. It is recommended to be equal to or greater than the IC's supply voltage.

- Pin 9 (enable3,4): When it is HIGH or 1, the right side of the IC is enabled. When it is LOW or 0, it is disabled.
- Pin 10 (input 3): When it is HIGH or 1, output 3 becomes HIGH, allowing current to flow to motor 2.
- Pin 11 (output 3): Connects to one of the terminals of motor 2.
- Pins 12, 13 (GND): Connects to ground (GND).
- Pin 14 (output 4): Connects to one of the terminals of motor 2.
- Pin 15 (input 4): When it is HIGH or 1, output 4 becomes HIGH, allowing current to flow to motor 2.
- Pin 16 (vcc1): Provides power to the IC. It should be powered with 5V.

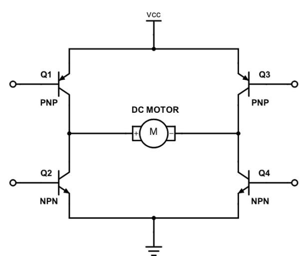
Figure 7: Diagram L239D



3.3.2 Operating principle of the IC

The H-bridge is an electronic circuit that allows voltage to be applied to a load in either direction and consists of four switches (or transistors) that create a "bridge" over the motor. During operation, when the switches are closed in a specific sequence, current can flow from one side of the bridge to the other, determining the direction of the motor's rotation. The operation is as follows (Figure 8): Q1 and Q4 closed switches: forward motion, Q2 and Q3 closed switches: reverse direction. Therefore, having low power requirements, I found the L293D to be the ideal and economical choice.

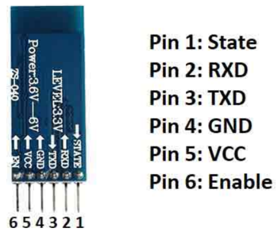
Figure 8: Circuit H-Bridge



3.4 HC-05 Bluetooth Module

The HC-05 Bluetooth module is affordable and compatible with many devices, supporting both master and slave functions. It uses the SPP (Serial Port Profile) protocol to communicate with the Arduino via UART, adjusting the baud rate (data transmission rate = 9600 bps) for serial communication. The ease of connection and reliable data transfer make the HC-05 a good choice for wireless communication.

Figure 9: HC-05



4 Chapter 2 : Mechanical Construction

The construction of the car is shown in Figure 10. Generally, the dimensions of the wheels and the distance between them significantly affect the robot's control capability. Two motors control the two wheels, allowing forward, backward, left, and right movement, thus providing two degrees of freedom. A "caster wheel" is used to maintain the stability of the vehicle. In the case of a two-wheeled mobile robot (differential drive), the position is determined by three coordinates on a plane (x, y, θ), while the robot's velocities can be described as shown in Figure 12.

Figure 10: Construction of a wheelchair base

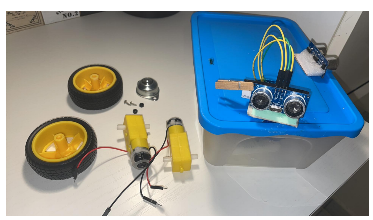


Figure 11: Mechanical servo interface with sensor

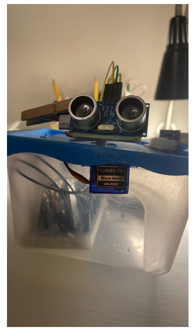
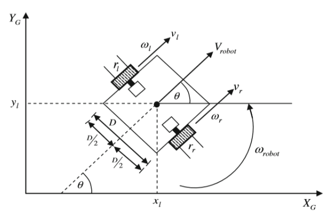


Figure 12:
1. $V_{robot} = v_r + v_l/2$ $W_{robot} = v_r - v_l/D$
2. $v_i = r * w_i \rightarrow$ linear velocity
3. $w_i \rightarrow$ angular velocity
4. $D \rightarrow$ distance between wheels
5. $r \rightarrow$ wheel radius



4.1 Μηχαναλογικά Στοιχεία

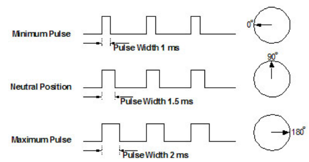
- Tupper: 14 cm ×10 cm
- Wheels: Diameter: 65mm Width : 28mm
- Castel Wheel 15mm
- Servo Motor

4.1.1 Servo Motor

The servo motor is a type of motor that includes an integrated system allowing it to move precisely to a specific angle or position. It consists of a small motor, a set of gears, and a control circuit. The control circuit receives signals in the form of electrical pulses and adjusts the motor's position accordingly. It is connected to a PWM (Pulse Width Modulation) signal, which is a series of pulses with a constant frequency. The duration of each pulse, known as the pulse width, determines the position of the servo motor.In the robot, the function of the servo motor is to support the ultrasonic sensor for creating a radar system. It controls the angle of rotation of the sensor, allowing it to scan the environment. I chose the SG90 micro servo because it was cost-effective, lightweight, and compact in size.

Figure 13: Servo Motor

Figure 14: Working Principle



4.1.2 DC Geared Motors

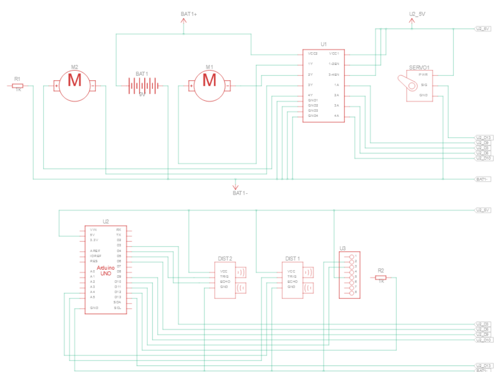
DC motors, such as the DC Geared TT motor, operate using the interaction of magnetic fields. When an electric current flows through the coil in the motor's armature, it creates a magnetic field that interacts with the stationary magnetic field provided by the stator. This interaction generates a force, causing the motor to rotate. The TT motor includes a gearbox, which contains gears that control the speed and torque. The choice of these motors was made because they have low power requirements and are cost-effective.

Figure 15: DC Motors and Wheels

5 Chapter 3 : Hardware Architecture

The hardware architecture includes the ATmega328 microcontroller as the "brain" of the Arduino Uno and the overall system. An L293D motor driver is added for motor control. The input unit includes sensors and a Bluetooth module, while the output unit is responsible for motor movement. The distance sensors are connected to digital pins, using data to control the motors. Adding a second distance sensor to the robot helps maintain a specific distance from the wall and assists in obstacle detection. The regular rotation of the front sensor is replaced by the second sensor on the left, providing faster response times. The servo motor is connected to a PWM pin to determine its position. The Bluetooth module is connected for serial communication, with the TX of the Bluetooth connected to the RX of the Arduino and the RX of the Bluetooth connected to the TX of the Arduino. Additionally, it is worth noting a design detail where a voltage divider (R1=1K, R2=2K) is used on the RX pin of the Bluetooth module to adjust logic levels due to the TTL standard operating at 3.3V.

Figure 16: Schematic illustration of hardware architecture using Tinkercad



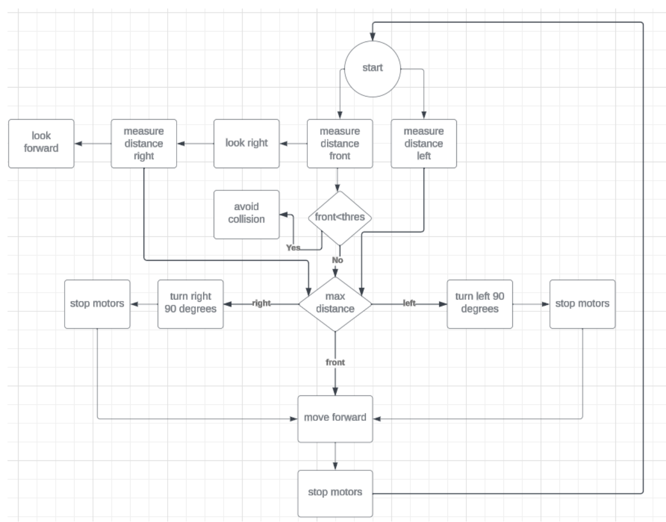
6 Chapter 4 : Software Architecture

The software implementation is responsible for translating the commands given by the user through the application into robot movement according to the selected mode. It is assumed that the user can change the robot's mode at any time. The software consists of three components, which are:

6.1 Avoid Obstacles

The process starts with measuring distances in three directions using an ultrasonic sensor. The findMaxDistance function determines the direction with the maximum distance, allowing the robotic system to choose the path with the most available space. The continuous computational process ensures obstacle avoidance and collision prevention while the robot maintains continuous movement in the direction that offers the greatest safety.

Figure 17: Flowchart via Lucidchart

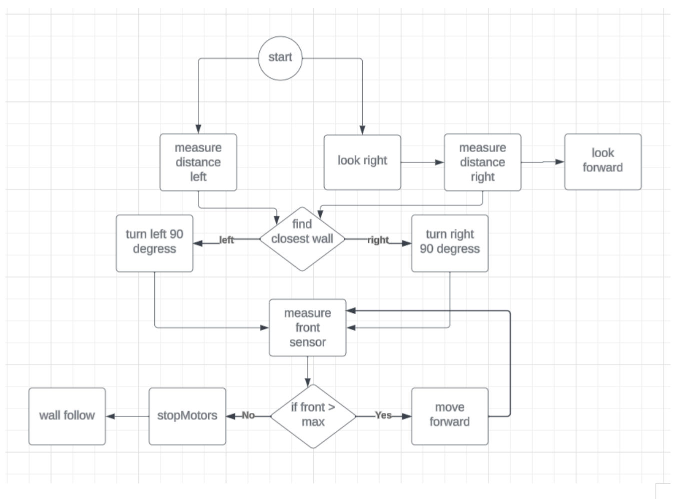


6.2 Wall Following

6.2.1 Nearest wall detection

Initially, the robot performs a recognition of the closest wall, either on the left or right, and then turns towards the direction where the distance to the wall is smaller. After that, the robot moves forward until it reaches the desired distance from the wall. Subsequently, depending on whether it is following a right or left wall, the robot adjusts its orientation to begin the wall-following process.

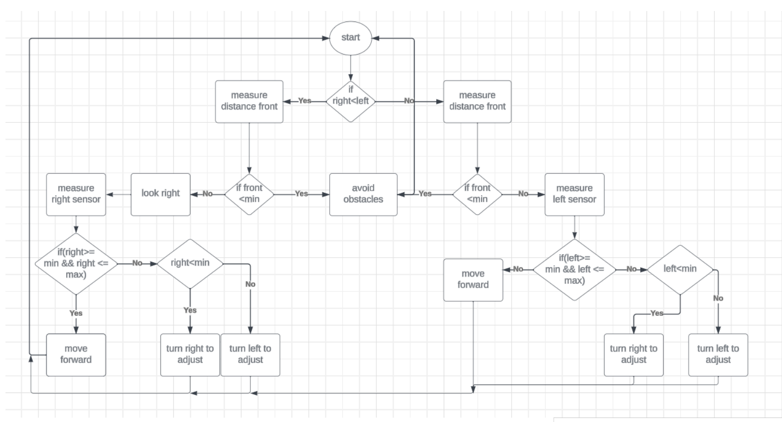
Figure 18: Nearest wall recognition, diagram via Lucidchart



6.2.2 Wall tracking process

During the execution of the wall-following process, the robot responds to two possible situations: either it detects an obstacle in front of it or it observes a decrease or increase in the distance from the wall. If an obstacle is detected or a change in angle is noted at a specific distance, the robot reacts by turning right or left (depending on whether it is following the right or left wall) to avoid the obstacle (in the case of an obstacle) or to continue with the wall-following process (in the case of a change in angle).
In the absence of an obstacle (when following the left wall), the robot continues moving forward. Additionally, if the distance from the left wall decreases below a certain threshold, the robot responds by turning right and increasing the speed of the left motor until the distance returns to an acceptable maximum value, and vice versa. This logic allows the robot to dynamically adjust its movement to avoid obstacles and maintain a consistent distance from the wall.

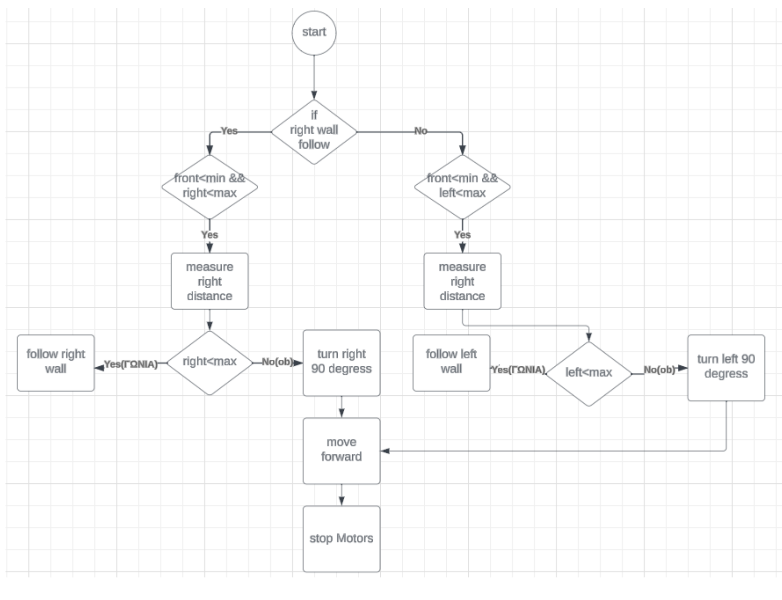
Figure 19: Wall Follow right/left , diagram via Lucidchart



6.2.3 Avoid Obstacles

First, it checks for obstacles or angles in front of it, considering the wall it is following. If the robot is following the right wall and detects an obstacle or angle, it proceeds to avoid it by initially turning left, taking the appropriate actions to either avoid the obstacle or continue following the wall (in the case of an angle). If the robot is following the left wall, it applies similar logic for avoidance. This logic allows the robot to dynamically respond to obstacles and angles, ensuring it continues to move safely along the wall.

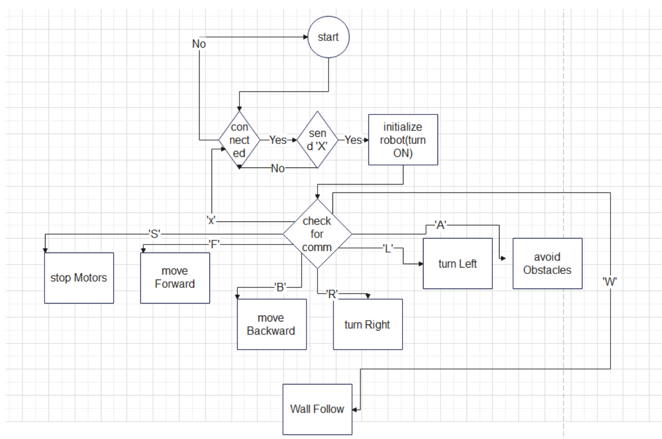
Figure 20: Obstacle avoidance logic, diagram via Lucidchart



6.3 Communication with User Interface

The robot initializes Bluetooth communication, ensuring that the HC-05 module is connected to a device before proceeding. Once connected, the robot 'listens' for commands from a paired device, and only upon receiving the 'X' command does the robot activate and enter the ON state. Additionally, the 'x' command is used to deactivate the robot (OFF state), initiating a shutdown process that stops the motors and prompts the user to restart the robot by sending the 'X' command when ready. This approach allows the user to remotely control the robot via Bluetooth commands.

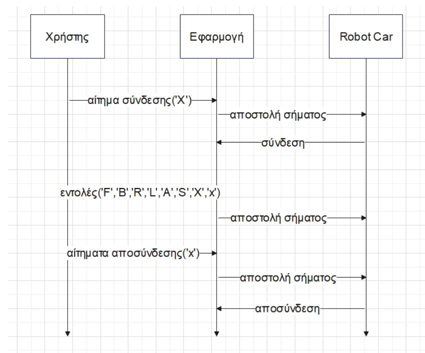
Figure 21: Interface and System Mode of Operation, diagram via Lucidchart



7 Chapter 5 : System Design

The system is divided into subsystems that work together to ensure the smooth operation of the overall system, providing the following functionalities with a simple and user-friendly interface. Below is a brief analysis of what each subsystem does and how it contributes to the overall system.

Figure 22: Sequence Diagram

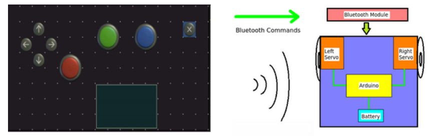


7.1 User Interface and Bluetooth Communication

It consists of a Bluetooth module, a user interface, and serves as the core for controlling the robot. Users can perform control actions for the robot, such as moving forward, moving backward, turning left, turning right, stopping, wall following/avoiding obstacles mode, and activation/deactivation, through the buttons provided in the application.

- Red : Stop Engines
- Green : Mode Avoid Obstacles
- Blue : Mode Wall Following
- X : ON,OFF
- Arrors : move forward/backward, turn right/left

Figure 23: Mode of Operation interface and HC-05



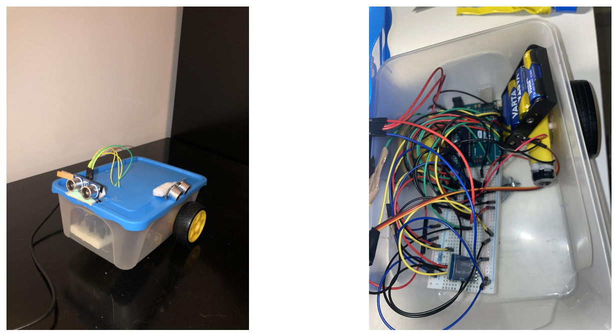


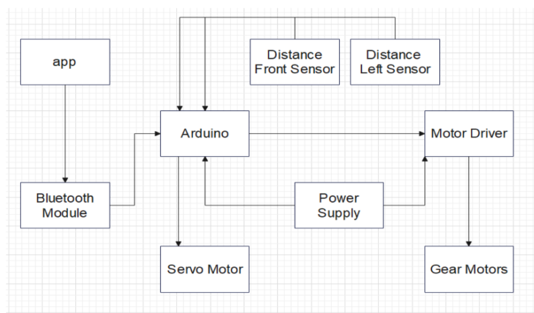
Figure 24: System circuit and wheelchair

7.2 Robot Car

7.2.1 Central Control and Data Processing Subsystem

The central control and data processing subsystem serves as the "brain" of the robot. Here, commands are translated into movements via motor control, while sensors collect environmental data, enabling obstacle avoidance and wall-following. Bluetooth communication ensures wireless interaction with the user (via the app) through the Bluetooth module. Together, these subsystems are coordinated by the microprocessor, providing a comprehensive control system.

Figure 25: Component diagram



7.2.2 Motor Power and Control

For power, a power supply is used, which provides sufficient power for continuous operation. The motor driver, powered by 4xAA batteries, achieves greater overall capacity compared to a 9V battery. The Arduino's USB port provides an additional 5V, powering all subsystems. The motor control subsystem consists of two gear motors and one servo motor, offering precise control of position and the necessary movement for the robot's operation. The servo motor provides position control through a closed-loop system with positive feedback, while the gear motors include an electric motor with a series of gears.

Figure 26: Subsystem Diagram using Lucidchart

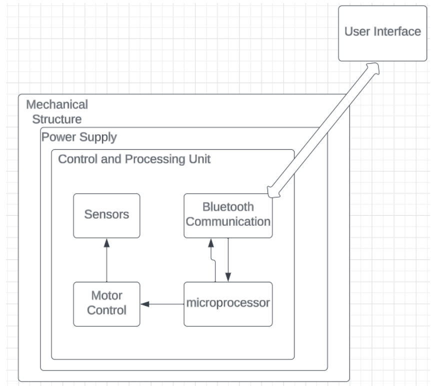


Figure 27: Gear Motor Mode of Operation

ΔΡΑΣΗ	ΔΕΞΙΟΣ ΚΙΝΗΤΗΡΑΣ	ΑΡΙΣΤΕΡΟΣ ΚΙΝΗΤΗΡΑΣ
ΕΜΠΡΟΣ	CLOCKWISE	ANTICLOCKWISE
ΠΙΣΩ	ANTICLOCKWISE	CLOCKWISE
ΔΕΞΙΑ ΣΤΡΟΦΗ	STOP	ANTICLOCKWISE
ΑΡΙΣΤΕΡΗ ΣΤΡΟΦΗ	CLOCKWISE	STOP

Figure 28: Mode Operation Servo Motor

ΜΠΡΟΣΤΑ ΑΙΣΘΗΤΗΡΑΣ	SERVO MOTOR
ΑΡΙΣΤΕΡΑ	90 μοίρες anticlockwise και μετά επιστρέψτε στην αρχική θέση
ΔΕΞΙΑ	90 μοίρες clockwise και μετά επιστρέψτε στην αρχική θέση

Experimental Measurements and Procedures for Checking Correct Operation Marianthi Thodi November 2023

8 Experimental Measurements and Procedures for Checking Correct Operation

8.1 HC-SR04 Calibration

Calibrating a sensor, such as the ultrasonic distance sensor HC-SR04, involves adjusting its measurements to align with the actual distances. The Linear Regression model is chosen for calibration because it is a simple method to minimize the sum of the squared differences between predicted and actual values. By training the model on a dataset that contains real and recorded values, it learns the scaling factor and offset required to correct inaccuracies in the sensor's measurements. This process is crucial because sensors can introduce systematic errors, affecting the reliability of distance measurements. Calibration ensures that the sensor's measurements are more accurate and consistent, enhancing the overall precision and reliability of the sensor output.

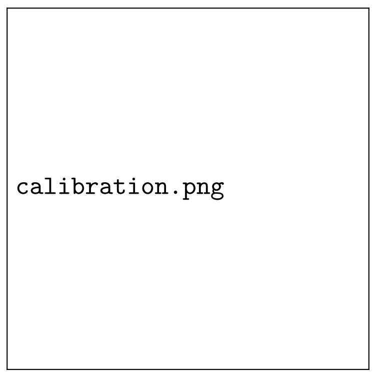


Figure 29: Comparison of actual and optimized measurements

8.2 Robot Adjusts Turning Based on Max Distance

The experiment aims to demonstrate a basic obstacle avoidance behavior in a robot. Based on the distance sensor data, the robot dynamically selects the direction with the most available space, showcasing the principles of autonomous navigation.

Experiments	Left Distance (cm)	Right Distance (cm)	Forward Distance (cm)	Status
1	81	57.75	156.82	move forward
2	56.73	77.97	0.0 (obstacle)	turn left
3	83.02	55.72	163.90	move forward
4	55.72	91.11	141.66	move forward
5	54.71	99.20	54.71	move forward
6	45.61	169.97	0.0 (obstacle)	turn right
7	209.40	103.24	55.72	move forward
8	102.23	156.83	0.0 (obstacle)	turn right
9	291.29	151.17	96.16	turn left
10	105.26	98.19	0.0 (obstacle)	turn left

Figure 30: Experiment data and motion state for robot vehicle

8.3 Sensor Testing for Avoid Obstacles (Stop Robot)

In my experiment, I have determined to take a measurement using the front sensor of the robotic car. If the robot detects an obstacle in front of it, at a distance less than 20 centimeters, it immediately stops moving. Subsequently, it begins to move backward while simultaneously rotating based on the maximum distance detected by the side sensors—either left or right. Through this specific experiment, I aim to demonstrate that the robot responds effectively by stopping immediately and taking appropriate actions when it detects an obstacle in front of it.

Experiments	Forward Distance (cm)	Status
1	12.25	stop motors
2	18.89	stop motors
3	20.0	stop motors
4	6.23	stop motors
5	8.28	stop motors
6	15.77	stop motors

Figure 31: Experiment data and motion state for robot vehicle

8.4 Sensor Testing for Wall Following

The robot determines which side (left or right) has the closest wall. Based on this, it turns towards the direction with the shorter distance (making the robot perpendicular to the wall). When it reaches a minimum distance from the wall, a command is given for the robot to turn in the opposite direction, aiming to make the robot parallel to the chosen wall—either right or left.

Experiments	Left Distance (cm)	Right Distance (cm)	Status
1	66.84	60.78	right closest wall
2	55.72	62.80	left closest wall
3	27.92	39.55	left closest wall
4	7.20	61.70	left closest wall
5	48.65	44.60	right closest wall
6	182.10	45.61	right closest wall
7	52.69	55.72	left closest wall

Figure 32: Experiment data and motion state for robot vehicle

9 Optimization

Transitioning from 2 wheels to 4 wheels on this robot will enhance its maneuverability and accuracy of movement. The four wheels provide a broader base, making the robot more stable and less prone to tipping over, especially during turns or on uneven terrain. They allow for a variety of movements, including lateral movement and rotation, making the robot highly versatile in different environments and expanding its range of motions. Additionally, incorporating a PID controller increases the control of position and direction, providing more stable and accurate movements, thus representing a significant step towards improving the control system. The PID controller ensures automatic adjustment of the motors, allowing for quick adaptation to the desired position with minimal error.

10 Extensions

The further development of the robotic system for exploration missions and smart cities involves the integration and advanced development of technologies such as SLAM (Simultaneous Localization and Mapping), deep learning, and AI. By leveraging SLAM, these vehicles will enhance mapping capabilities, facilitating precise navigation in unexplored environments and complex urban spaces. Deep learning algorithms, such as YOLO, will be used for obstacle recognition, ensuring effective route planning. As artificial intelligence continues to play a central role, the adaptability of robotic vehicles will improve through continuous learning, allowing for enhanced decision-making. The integration of these technologies is poised to create the next generation of intelligent, efficient, and adaptive mobile robotic vehicles, shaping the future of exploration missions and navigation in smart cities.

11 Bibliography

[1] Gandha, Gutama Nurcipto, Dedi. (2020). The Performance Improvement of the Low-Cost Ultrasonic Range Finder (HC-SR04) Using Newton's Polynomial Interpolation Algorithm. JURNAL INFOTEL. 11. 10.20895/infotel.v11i4.456.

[2]Dika, Mahardika Suciyati, Sriwahyu Supriyanto, Amir Pauzi, Ahmad. (2023). Smart Green Box Trash Design Based on HC-SR04 Sensor Arduino Uno Integrated. Journal of Energy, Material, and Instrumentation Technology. 4. 89-98. 10.23960/jemit.v4i1.115.

[3] Hasan, Md Masum, Md Hafizur Rahman Chowdhury, Kantish. (2022). Bluetooth Controlled Arduino Based Autonomous Car. 10.13140/RG.2.2.28191.23208.

[4] Goswami, Shankha Sahoo, Sushil Kumar. (2023). Design of a Robotic Vehicle to Avoid Obstacle Using Arduino Microcontroller and Ultrasonic Sensor. 10.3233/ATDE231041.

[5] Chhotray, Animesh Pradhan, Madhuri Pandey, Krishna Parhi, Dayal. (2016). Kinematic Analy-

sis of a Two-Wheeled Self-Balancing Mobile Robot. 10.1007/978-81-322-3589-7-9.

[6] Ullah, Dr. Saleem Mumtaz, Zain Liu, Shuo Abubaqr, Mohammad A., Mahboob, Madni, Hamza. (2019). Single-Equipment with Multiple-Application for an Automated Robot-Car Control System. Sensors. 19. 662. 10.3390/s19030662.

[7] Jayasinghe, Himash Godigamuwa, Llakshi Kudagoda, Bihesha Dilshan, Manuka. (2022). Development of a Fuzzy Logic Wall Following Mobile Robot.