Digital Control Course

Dr./ Ahmed Osman Mahmoud

Distance Control Vehicle

Members:

محمد محمود محمد غانم

حسن سامي حسن جودة

عبدالرحمن علاء احمد مصيلحي

احمد جمال عزقلاني

حسن سامح حسن محمد قاعود

محمد عادل محمد قاعود

عبدالرحمن ابراهيم محمود محمد

محمد عبدالحميد فخري

محمد احمد محمود محمد حسين

محمد مصطفي محمد احمد

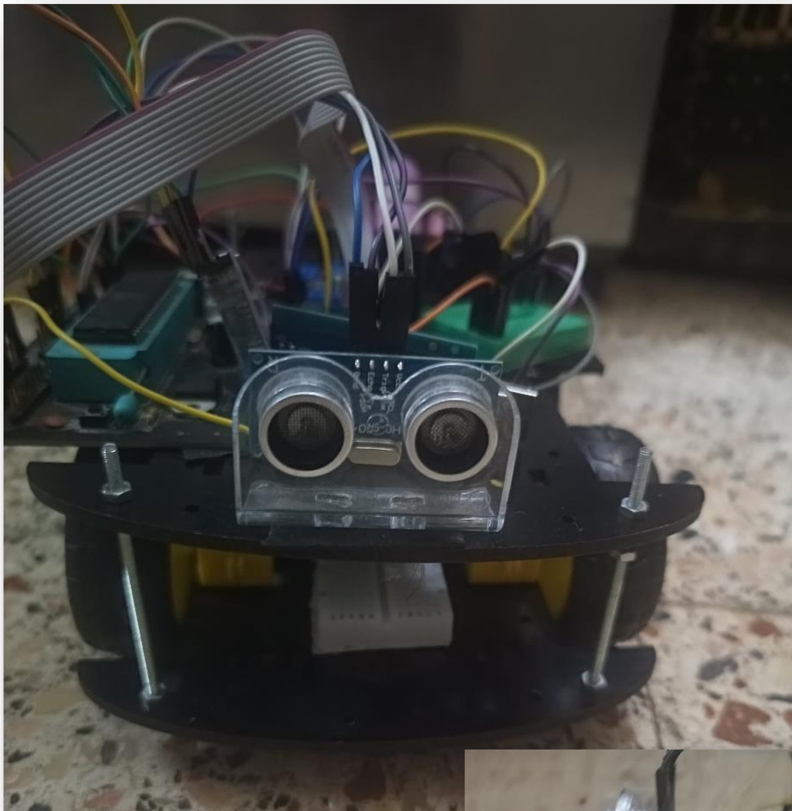
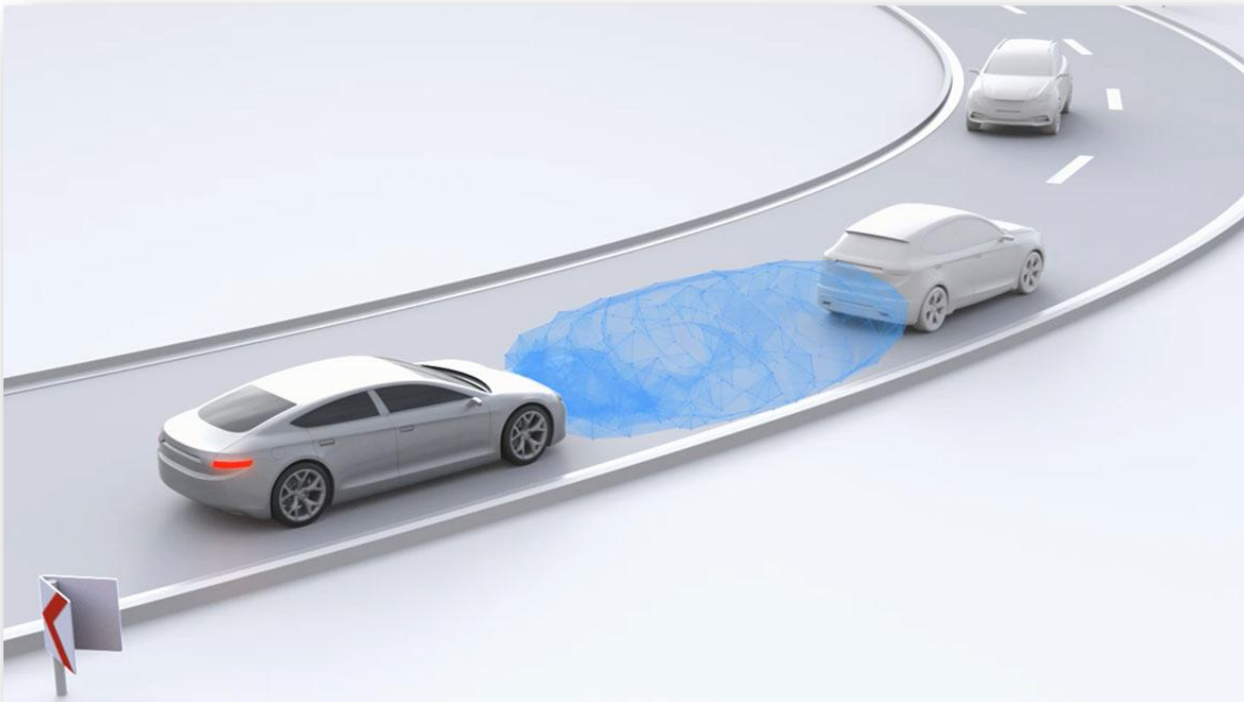


Table of Contents

1. Introduction



The increasing need for safety and automation in modern vehicles has led to the development of systems capable of preventing collisions by maintaining a safe distance from obstacles. This project aims to design and implement a distance control system for a vehicle using affordable and widely available components. The system employs an ultrasonic sensor to continuously monitor the distance between the vehicle and any objects in its path.

When the distance between the vehicle and the obstacle exceeds a predefined set point, the vehicle accelerates to reduce the gap. Conversely, if the measured distance is below the set point, the vehicle reverses to restore the safe distance. This dynamic adjustment ensures that the vehicle operates safely and efficiently, avoiding potential collisions or unsafe proximity to obstacles.

The project uses an ATmega32 microcontroller as the brain of the system, which processes data from the ultrasonic sensor and controls the motors accordingly. Additional components like motor drivers, a Bluetooth module, and a crystal oscillator enhance the system's functionality and responsiveness.

This project is not only a demonstration of automation but also a step toward building foundational technologies for autonomous vehicles. It has potential applications in robotics, industrial automation, and smart delivery systems, providing practical solutions for real-world challenges.

2. Objectives

The main objectives of this project are:

- To develop an automated distance control mechanism using microcontrollers and sensors.
- To ensure smooth and responsive speed adjustments for maintaining safe distances.
- To utilize cost-effective and widely available components for implementation.
- To create a mobile application for real-time monitoring of the distance measured by the ultrasonic sensor.
- To ensure the system is reliable and operates effectively under varying environmental conditions.

3. Components Used

3.1 Chassis

- **Purpose:**
  - The chassis is the structural foundation of the vehicle, providing physical support to all the electronic components and ensuring stability during movement.

- **Features:**
  1. Material:
     - Typically made of lightweight but durable materials such as aluminum alloy, plastic, or acrylic.
     - Ensures the vehicle is robust enough to carry all components while remaining mobile.
  2. Design:
     - Designed to accommodate four DC motors for a balanced and stable four-wheel-drive system.
     - Includes pre-drilled holes or mounts for easy installation of the motor driver, ultrasonic sensor, battery, and other modules.
  3. Motor Mounts:
     - Equipped with motor brackets to securely attach the DC motors.
     - Includes slots for adjusting the position of the motors to optimize traction and balance.
  4. Space for Components:
     - Provides dedicated mounting areas for the battery, microcontroller, and motor drivers.
     - Includes a flat or layered surface to organize components neatly and reduce wiring complexity.

5. Dimensions:
   - Sized appropriately to fit all required components while maintaining a compact footprint for maneuverability.
   - For this project, a chassis of approximately 15 cm x 20 cm is recommended for compact design and stability.
6. Wheels and Tires:
   - Supports four wheels with anti-skid rubber tires for effective traction on various surfaces.
   - Ensures smooth movement and reduces slippage during operation.
7. Weight Distribution:
   - Balances the weight of the battery, motors, and electronics to prevent tipping or uneven movement.
   - Ensures uniform pressure on all wheels to improve motor efficiency.

3.2 ATmega32 Microcontroller

- **Purpose:**
  

  - Serves as the core controller for the system, handling input data, processing logic, and output signals.
  - Processes the ultrasonic sensor's distance readings and makes real-time decisions to adjust the vehicle's speed and direction.
- **Features:**
  - 32 General-Purpose Input/Output (GPIO) pins to interface with sensors, motor drivers, and other peripherals.
  - 8-bit architecture, providing sufficient processing power for embedded applications.
  - Supports communication protocols like UART, SPI, and I2C, enabling easy integration with external components such as the HC-05 Bluetooth module and other sensors.
  - Built-in timers and interrupts for real-time operations like precise distance measurements and motor control.
- **Benefits:**
  - Cost-effective and widely used in academic and industrial projects.
  - Provides flexibility and compatibility with many external components.

3.2 Ultrasonic Sensor

- **Model:** HC-SR04

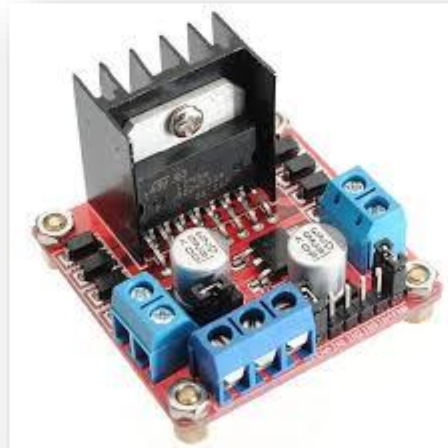
- **Purpose:**
  - Measures the distance between the vehicle and obstacles using ultrasonic waves.
  - Provides high accuracy for detecting objects within its range.
- **Working Principle:**
  - The sensor sends ultrasonic waves and calculates the time taken for the reflected wave (echo) to return.
  - The distance is determined using the formula:

$$\text{Distance} = \frac{\text{Time (microseconds)} \times \text{Speed of Sound (340 m/s)}}{2}$$

- **Specifications:**
  - **Range:** 2 cm to 400 cm.
  - **Accuracy:** ±3 mm.
  - **Operating Voltage:** 5V DC.
  - **Pins:** Trigger (input), Echo (output), VCC, and GND.
- **Benefits:**
  - Low cost, high reliability, and easy to interface with microcontrollers.

3.3 L298 Motor Driver

- **Purpose:**



  - Alternative motor driver with higher current handling capability compared to L293.
  - Suitable for driving heavier motors with increased torque requirements.

- **Features:**
  - Dual H-bridge configuration to drive two DC motors.
  - Supports motor voltages up to 46V and currents up to 2A per channel.
  - Built-in diodes for protection against back-EMF.
- **Benefits:**
  - Offers higher current output, enabling the use of larger, more powerful motors.
  - Ensures system reliability under high-load conditions.

3.4 HC-05 Bluetooth Module

- **Purpose:**

  - Facilitates wireless communication for debugging and remote control of the vehicle.
  - Enables real-time monitoring and command sending through Bluetooth-enabled devices.

- **Features:**
  - Supports serial communication with a baud rate range of 9600 to 115200.
  - Operates on 3.3V to 5V DC.
  - Standard range: Up to 10 meters in open space.
- **Applications in Project:**
  - Allows developers to monitor sensor readings and motor behavior.
  - Offers flexibility to adjust the set point dynamically via a mobile app or PC.

3.5 DC Motors (4 Units)

- **Purpose:**

  - Propels the vehicle forward and backward.
  - Adjusts the speed and direction of movement based on microcontroller signals.
- **Features:**
  - High torque for efficient propulsion, even with additional load.
  - Typically rated for 6V to 12V operation.
  - Compact size, suitable for embedded robotic applications.
- **Motor Placement:**
  - Two motors for forward/reverse drive.
  - Two motors for additional stability and traction.

3.6 8 MHz Crystal Oscillator

- **Purpose:**

  - Provides stable clock signals for the ATmega32 microcontroller, ensuring precise timing and reliable operation.

- **Features:**
  - Frequency: 8 MHz, suitable for most microcontroller applications.
  - Enables accurate timing for tasks like ultrasonic signal processing and motor control.

- **Benefits:**
  - Ensures consistent performance by reducing timing errors.

3.7 Battery

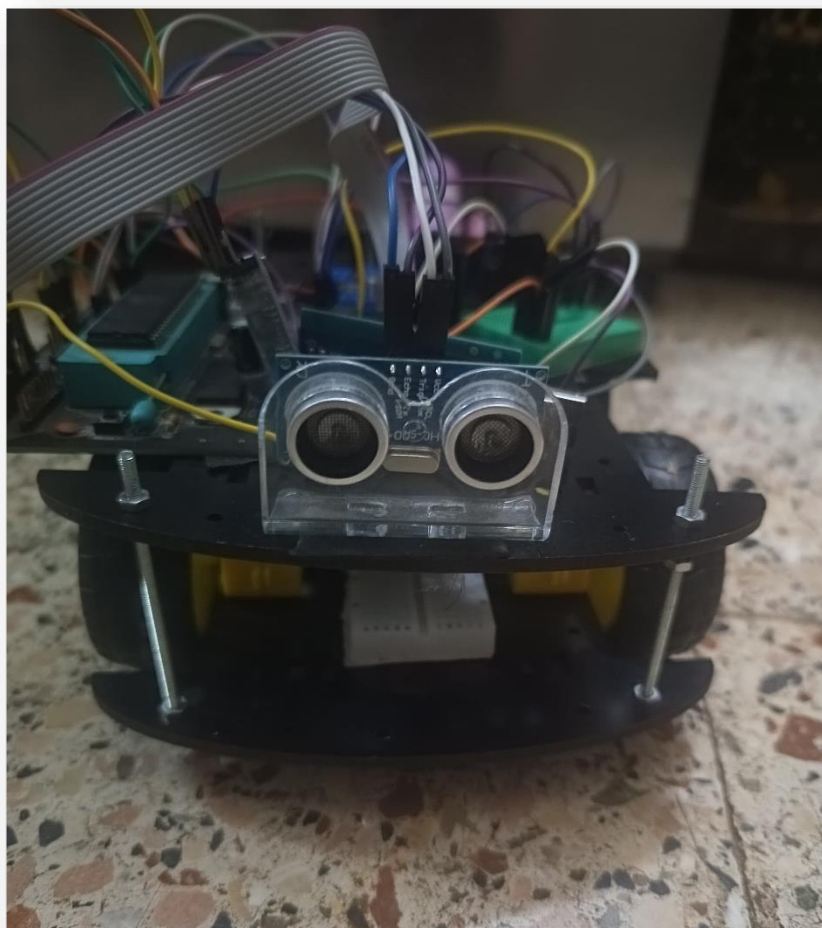
- **Purpose:**

  - Supplies power to all electronic components, including motors, sensors, and the microcontroller.

- **Specifications:**
  - Voltage: 12V, rechargeable.
  - Capacity: Typically 1500mAh to 3000mAh for moderate runtime.

- **Power Distribution:**
  - Motors are powered directly from the battery via the motor driver.
  - The microcontroller and sensors are powered through a voltage regulator (e.g., 7805) to step down to 5V.
- **Benefits:**
  - Portable power source for uninterrupted operation.

4. System Overview

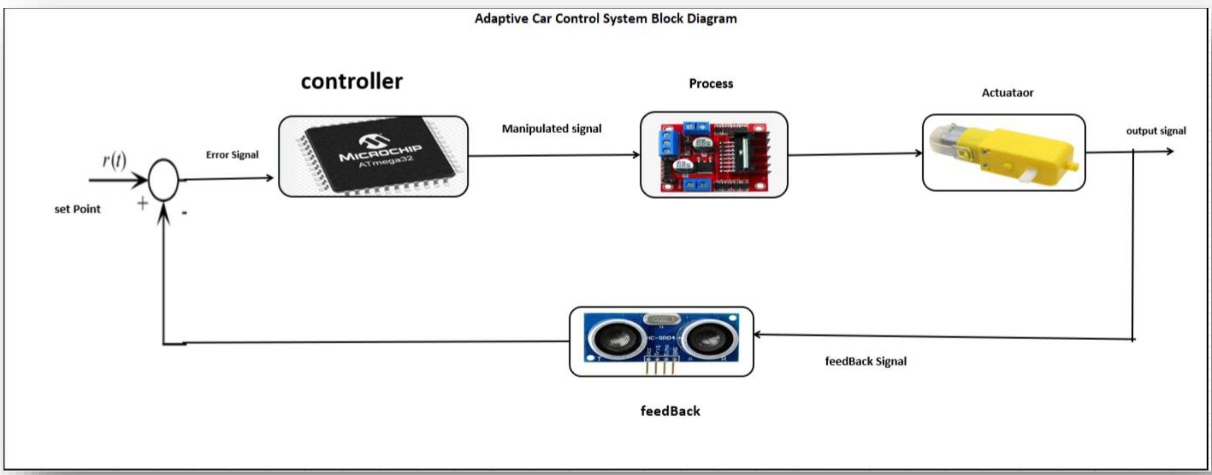


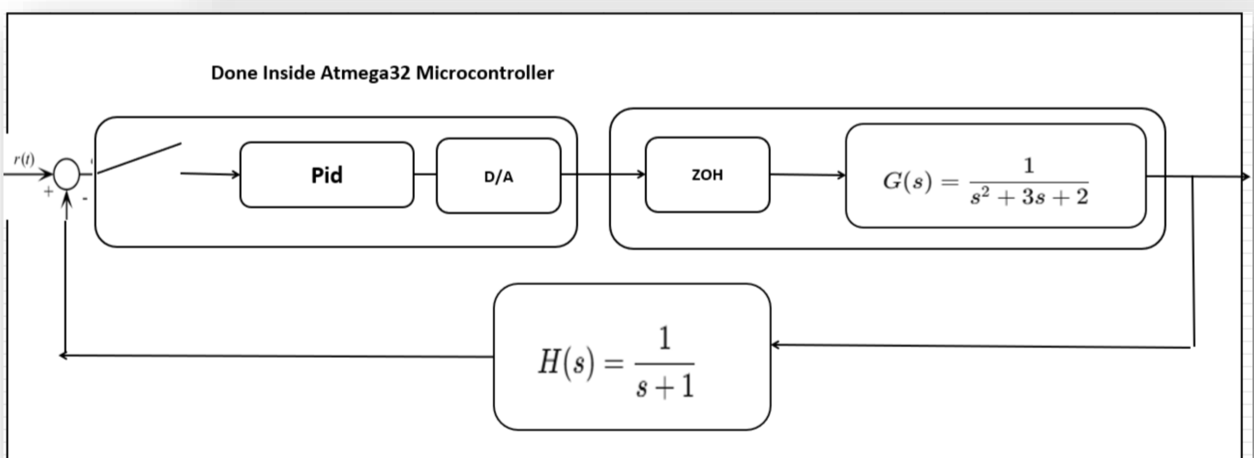
The vehicle uses an ultrasonic sensor to measure the distance to objects in its path. Based on the measured distance:

- If the distance is greater than the set point, the vehicle accelerates forward until the safe distance is maintained.
- If the distance is less than the set point, the vehicle reverses to restore the safe distance.

All components work together to create an efficient and responsive system.

5.Block Diagram :





Z-Transformation for PID controller :

C(z)=Kp+Ki (z−1)/ z +Kd z/ (z−1)

$$\frac{U(z)}{E(z)} = \frac{K_0 + K_1 z^{-1} + K_2 z^{-2}}{1 - z^{-1}}$$

Difference Equation for PID controller:

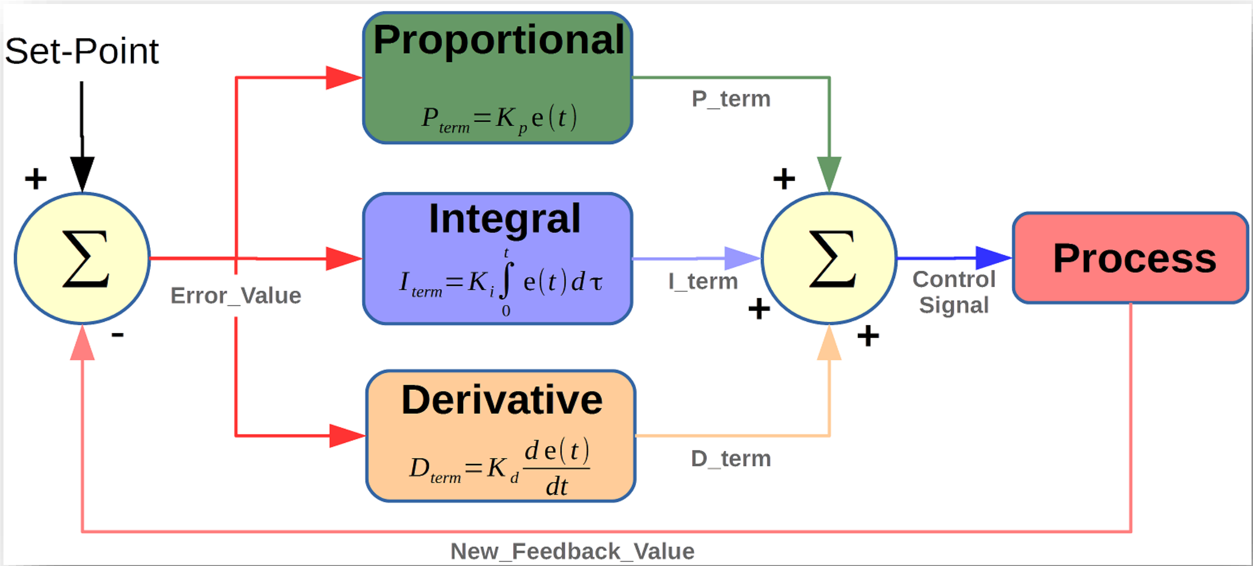
Un = Un-1 + K0 * en + K1 * en-1 + K2 * en-2

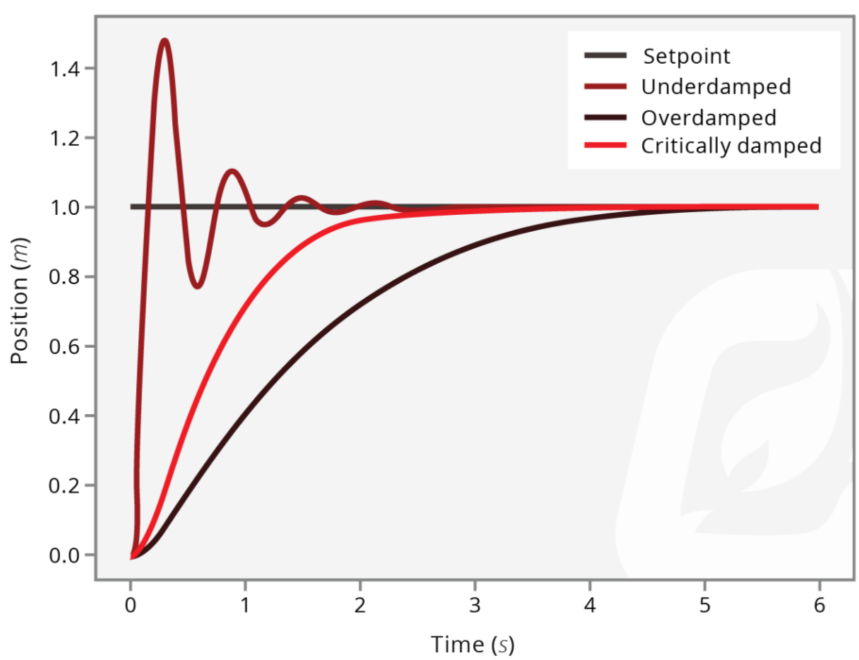
$$u(k) = K_p e(k) + K_i \sum e(k)\Delta t + K_d \frac{e(k)-e(k-1)}{\Delta t}$$

Where:

- u(k) : Control output at time k.
- e(k): Error at time k
  (e(k)=desired distance−measured distance(k)
- Kp,Ki,Kd: PID gains for proportional, integral, and derivative terms.
- Δt: sampling time.

PID Controller :

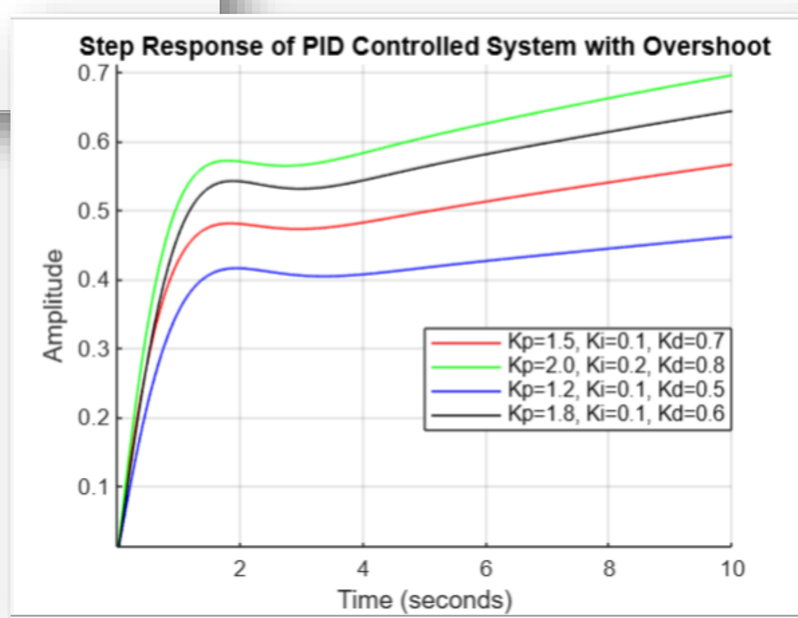




PID Controller Response Types

Command Window

```
>> % Define the Laplace variable
s = tf('s');

% Define sets of PID parameters
params = [1.5, 0.1, 0.7;   % First set of Kp, Ki, Kd
          2.0, 0.2, 0.8;   % Second set of Kp, Ki, Kd
          1.2, 0.05, 0.5;  % Third set of Kp, Ki, Kd
          1.8, 0.15, 0.6]; % Fourth set of Kp, Ki, Kd

% Define a plant with higher-order dynamics (second-order underdamped system)
Plant = 1 / (s^2 + 2*s + 2); % Example plant with overshoot

% Initialize the figure
figure;
hold on;

% Colors for each set of PID parameters
colors = ['r', 'g', 'b', 'k'];

% Simulation time vector
t_sim = 0:0.01:10; % From 0 to 10 seconds

% Loop through each set of parameters
for i = 1:size(params, 1)
    Kp = params(i, 1);
    Ki = params(i, 2);
    Kd = params(i, 3);

    % Define the PID controller
    PID = Kp + Ki/s + Kd*s;

    % Closed-loop system
    ClosedLoop = feedback(PID * Plant, 1);

    % Compute the step response
    [y, t] = step(ClosedLoop, t_sim); % Simulate over the defined time

    % Plot the step response
    plot(t, y, colors(i), 'DisplayName', ...
        sprintf('Kp=%.1f, Ki=%.1f, Kd=%.1f', Kp, Ki, Kd));
end

% Customize the plot
title('Step Response of PID Controlled System with Overshoot');
xlabel('Time (seconds)');
ylabel('Amplitude');
legend('show'); % Show the legend
grid on;
hold off;
```



```
>> % Define the coefficients
Kp = 1.5; % Proportional gain (example value, replace as needed)
Ki = 0.1; % Integral gain (example value, replace as needed)
Kd = 0.7; % Derivative gain (example value, replace as needed)

% Sampling time (specify Ts)
Ts = 0.05; % Example sampling time (adjust according to your system)

% Define the z-transform terms
z = tf('z', Ts);

% Controller C(z)
Cz = Kp + Ki * (z - 1)/z + Kd * z/(z - 1);

% Display the transfer function
disp('The transfer function C(z) is:')
Cz

% Convert C(z) to C(s) using Tustin method
Cs = d2c(Cz, 'tustin');

% Display the continuous-time transfer function
disp('The continuous-time transfer function C(s) is:');
Cs
The transfer function C(z) is:
```

```
The transfer function C(z) is:

Cz =

  2.3 z^2 - 1.7 z + 0.1
  ---------------------
         z^2 - z

Sample time: 0.05 seconds
Discrete-time transfer function.

The continuous-time transfer function C(s) is:

Cs =

  2.05 s^2 + 88 s + 560
  ---------------------
       s^2 + 40 s

Continuous-time transfer function.
```

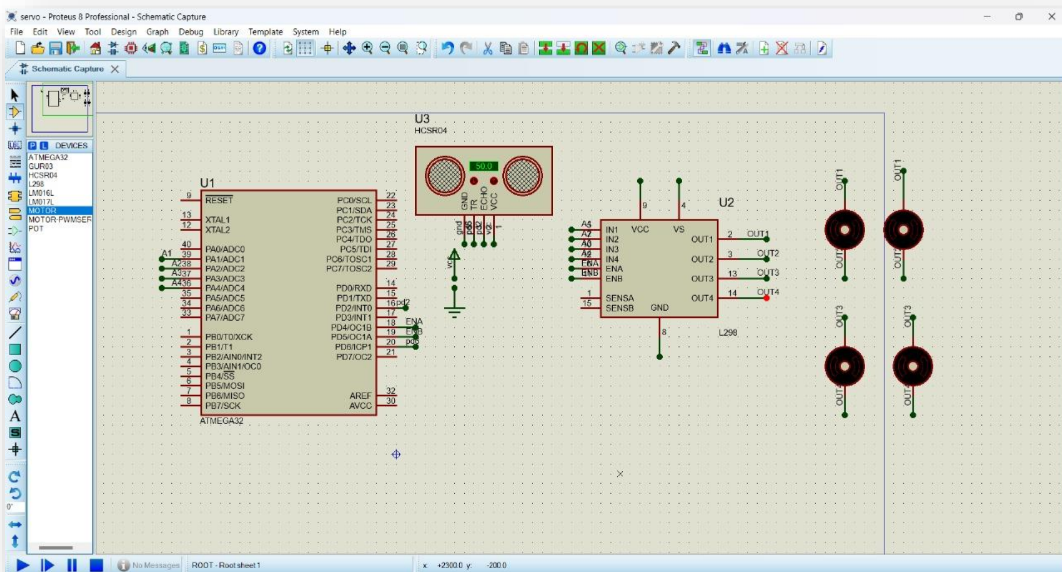
6. Working Principle

1. **Distance Measurement:** The ultrasonic sensor emits sound waves and measures the time taken for the echo to return. This time is converted into distance.
2. **Decision Making:** The ATmega32 processes the distance data and compares it with the set point.

3. **Motor Control:**

    - If the distance > set point: Increase motor speed.
    - If the distance < set point: Reverse motor direction.

7. Circuit Design

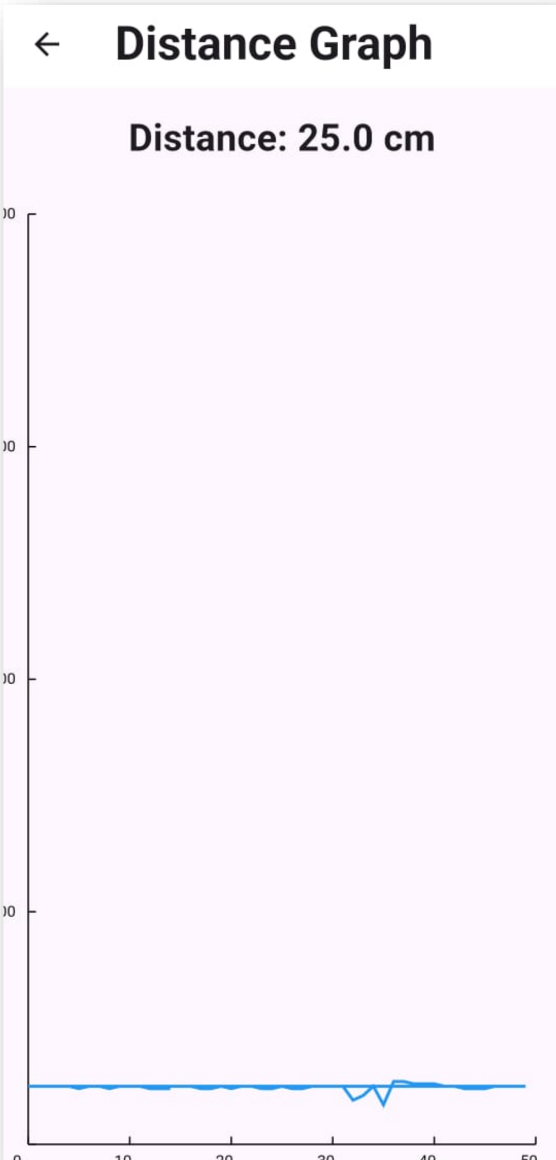


- **Connections:**
    - The ultrasonic sensor is connected to the microcontroller's I/O pins for data input.
    - The L298 motor driver is connected to the microcontroller for controlling motor operations.
    - The HC-05 Bluetooth module is connected to the UART pins of the ATmega32.
    - The 8 MHz crystal oscillator is connected to the microcontroller for clock stability.
- **Power Supply:**

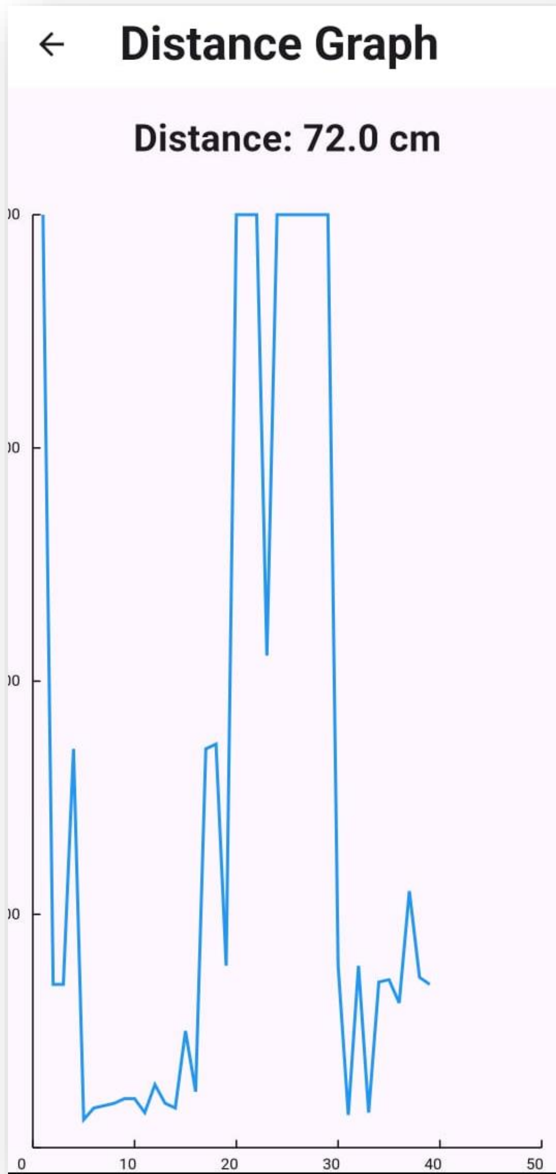
- The battery supplies 12V to the motor driver and 5V to the microcontroller via a voltage regulator.

8. Mobile Application

at set point 25 cm



at distance 72 cm



A mobile application is developed to enhance the project's usability. Key

features include:

- **Real-Time Distance Display**: The app shows the distance measured by the ultrasonic sensor, updated continuously via Bluetooth.
- **User Interface**: A simple and intuitive interface with numerical and graphical representations of the distance.
- **Debugging Tool**: The app helps in identifying any discrepancies in sensor readings or system behavior.
- **Technology Used**: The app is built using Android Studio with flutter and communicates via the HC-05 Bluetooth module.

9. Software Implementation

1. **Microcontroller Programming:**
    - The ATmega32 is programmed in C using AVR Studio.
    - Key tasks include:
        - Reading distance data from the ultrasonic sensor.
        - Comparing measured distance with the set point.
        - Sending motor control signals based on the comparison.

2. **Algorithm:**
    - Initialize all components.
    - Continuously read distance from the ultrasonic sensor.
    - Compare the distance to the set point.
    - Adjust motor speed and direction accordingly.

10. Applications

- Collision avoidance in autonomous vehicles.
- Industrial robots requiring safe operation near humans.
- Automated delivery systems in warehouses.

11. Links of Video and Code

Link Videos :

https://drive.google.com/drive/folders/1yKzVJq0gCqcXVeQLlLvyWVCB7KxVGWhu?usp=sharing

Link Code on GitHub :

https://github.com/Ghanem-MO/Distance-Control-vehicle.git

12. Conclusion

This project successfully demonstrates a distance control mechanism using ATmega32 and an ultrasonic sensor. It highlights the potential of simple, cost-effective components in building efficient autonomous systems.

13. References

- **Datasheets**
  - **ATMEGA32 Microcontroller Datasheet**:
    Source: https://www.microchip.com
  - **HC-05 Bluetooth Module Datasheet**:
    Source: https://components101.com
  - **Ultrasonic Sensor (HC-SR04) Datasheet**:
    Source: https://components101.com
  - **L298 Motor Driver Datasheet**:
    Source: https://www.st.com