CHAPTER 1

INTRODUCTION

The Otto robot is a cutting-edge, autonomous machine designed to revolutionize various industries and aspects of life. Named after Nikolaus August Otto, the inventor of the internal combustion engine, this robot embodies innovation and efficiency. The design of the Otto robot focuses on creating a versatile, user-friendly, and highly adaptable machine that can be applied in diverse settings, from manufacturing and logistics to healthcare and service industries. The primary objective of the project is to introduce learners to the fundamentals of robotics through hands-on experience. The process involves mechanical assembly, electronic wiring, and programming using the Arduino IDE. Once assembled, Otto can walk, dance, detect and avoid obstacles, and perform basic interactive behaviours of the key strengths of the Otto robot is its modular and customizable design. As an open-source project, it allows users to freely modify the hardware and software to create their own versions, enabling creativity and innovation. The project not only improves technical skills but also encourages logical thinking, problem-solving, and teamwork.This project is ideal for use in schools, workshops, and hobbyist communities, providing a fun and engaging way to explore STEM concepts. Overall, the Otto Robot Project serves as a practical introduction to the world of robotics and automation, laying the foundation for more advanced learning and development.

1.1 Key Design Objectives

i. Autonomy and Navigation: The Otto robot is designed to operate autonomously, navigating through complex environments with ease and precision.

ii. Modularity and Adaptability: The robot's modular design allows for easy customization and adaptation to various tasks and applications.

iii. User-Friendly Interface: The Otto robot features an intuitive interface, enabling users to easily interact with and program the robot.

iv. Safety and Reliability: The design prioritizes safety and reliability, ensuring the robot can operate efficiently and securely in various environments.

CHAPTER 2

METHODOLOGY AND SOFTWARE TOOLS

2.1 Methodology

The implementation includes several steps including design, compilation and uploading. The following is a description of the methodology in a step wise used in this project:

2.1.1 Project Setup

The Otto robot project begins with gathering all necessary components and preparing the workspace. The 3D parts are printed and assembled as per design specifications. The Arduino Nano is mounted, and all electronic components are connected according to the circuit diagram.

2.1.2 Implementation

After the hardware is assembled, the next step is writing the code to control Otto's movements and behavior. The code is developed using the Arduino IDE. Otto's basic functions include walking, turning, dancing, and obstacle avoidance. Libraries like Otto calibration are added to simplify programming and sensor integration.

2.1.3 Compilation

Once the code is written, it is compiled using the Arduino IDE. This process checks for syntax errors, verifies library dependencies, and prepares the code in machine language for the microcontroller. If there are any errors, they are debugged and corrected before moving to the next stage.

2.1.4 Uploading

After successful compilation, the code is uploaded to the Arduino Nano using a USB cable. The board and port are selected in the Arduino IDE, and the upload process is initiated. Once uploaded, the Otto robot begins executing the programmed instructions, enabling movement and interaction.

2.2 Software Tools

The primary software tool used in the design and programming of the Otto robot is the Arduino Integrated Development Environment (IDE). It plays a crucial role in writing, compiling, and uploading the code that controls the robot's actions and behaviours.

2.2.1 Arduino IDE:

The Arduino IDE is an open-source programming environment that supports Arduino-compatible microcontrollers like the Arduino Nano, which is used in the Otto robot. It allows users to write code in a simplified version of C/C++, verify it through compilation, and upload it directly to the microcontroller via USB.

Key features of Arduino IDE include:

- Easy-to-use interface for writing and editing code
- Library manager for installing required libraries such as OttoDIY,OttoCalibration and Otto avoid.
- Serial Monitor for debugging and monitoring sensor data
- Compatibility with various Arduino boards

2.2.2 Otto Avoid

Otto Avoid is an upgraded version of the original Otto DIY robot, designed with obstacle detection and avoidance capabilities. It is a small, 3D-printed bipedal robot powered by an Arduino Nano microcontroller and controlled using four servo motors for movement. The ultrasonic sensor constantly measures the distance to nearby objects. When an obstacle is detected within a certain range, the robot stops, analyzes the environment, and changes direction to avoid a collision. This makes Otto Avoid capable of basic autonomous navigation, enhancing its interactivity and practical application in learning environments.

CHAPTER 3

CIRCUIT DIAGRAM

3.1 Circuit Diagram

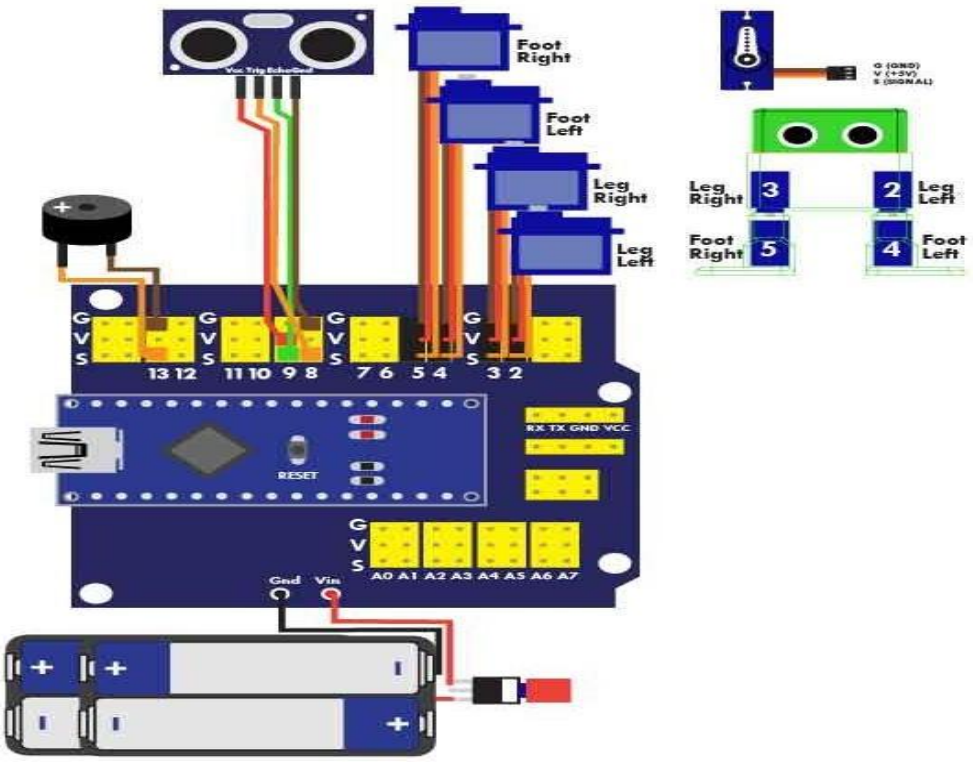


Fig.3.1 Block Diagram of Otto Robot Design

The circuit diagram shows the wiring layout for the Otto Avoid Robot, which uses an Arduino Nano as the central microcontroller. The components are connected as follows:

3.1.1 Servo Motors

4 Micro Servo Motors are used to control the legs and feet:

- Leg Left → Pin D2
- Leg Right → Pin D3
- Foot Left → Pin D4
- Foot Right → Pin D5

These servos enable Otto to walk and turn by controlling the motion of the legs and feet.

3.1.2.Ultrasonic Sensor (HC-SR04)

There were Two pins have been used. There were:

- Trigger Pin D9
- Echo Pin D8

This sensor detects obstacles in front of Otto and helps it avoid them by changing direction.

3.1.3 Buzzer

Connected to Pin D13, used for sound output (e.g., beeps or melodies).

3.1.4 Power Supply

Powered by a battery pack (typically 4x AA batteries), connected to Vin and GND to provide power to the Arduino board and the connected components.

3.1.5 Ground and Power

All components share common GND and 5V lines, ensuring proper operation and voltage levels.

3.2 Description of Components

Here's a detailed explanation of all the components used in the Otto Avoid robot design:

3.2.1 Arduino Nano (Microcontroller)

- Function: Acts as the central control unit (brain) of the Otto robot.
- Role: Executes the programmed code to control movements, read sensor data, and manage power to the servos and buzzer.
- Interfaces: Multiple digital I/O pins for connecting servos, sensor, and buzzer.

3.2.2 **Servo Motors:**

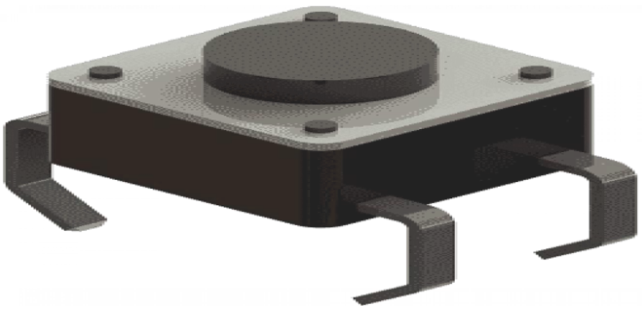
- Function: Control movement of the robot’s legs and feet.

Servo motors are used to provide the robot with movement. In the Otto robot, they control the legs, allowing the robot to walk, dance, and perform other motions. By adjusting the angle of the servo motors, the robot can change its posture and execute complex movements.

3.2.3 Ultrasonic Sensor (HC-SR04)

- Function: Detects obstacles using sound waves.
- How it works: Sends an ultrasonic pulse from the Trigger pin, and receives the reflected wave through the Echo pin. The time it takes is used to calculate distance.

3.2.4 Push Button



The push button allows the user to interact with the robot, initiating various behaviors like starting movement, triggering sounds, or activating different functions.

3.2.5 Piezo Buzzer

- Function: Provides sound output (alerts, effects, or feedback).
- Connection: Connected to Pin D13 of Arduino.
- Usage: Plays simple tones or beeps to signal activity or warnings (like when an obstacle is detected).

3.2.6 Jumper Wires

- Function: Make electrical connections between components.
- Types: Male-to-male and female-to-female wires are used depending on the board and connectors.
- Jumper wires are essential for connecting all the electronic components in the Otto robot, including the Arduino Nano, sensors, motors, and other devices. They ensure that the circuit is complete and that signals can be transferred between components for proper functioning.

3.2.7 Battery Pack

- Function: Powers the entire robot (typically 4x AA batteries).
- Voltage: Provides around 6V (regulated by Arduino to 5V).
- Connection: Connected to Vin (power input) and GND on the Arduino board.

3.2.8 3D Printed Body

- Function: Encloses and supports all components.
- Parts Include:

  Head (holds ultrasonic sensor)

  Body (houses Arduino and battery)

  Legs and feet (hold servo motors

CHAPTER 4

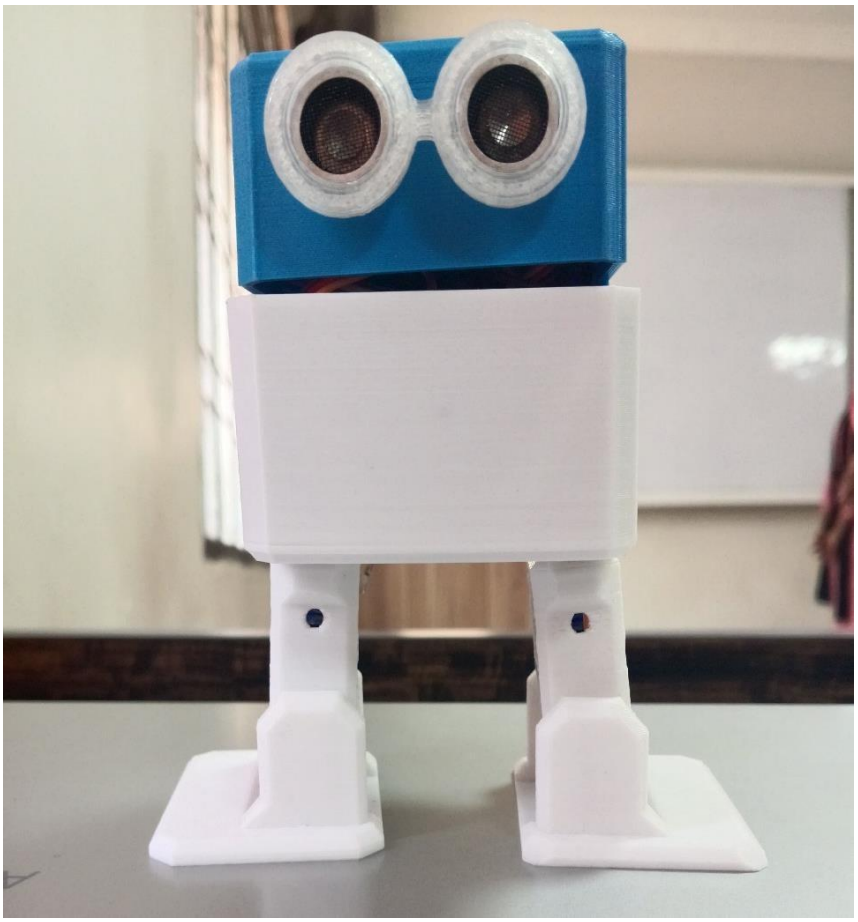
SOFTWARE IMPLEMENTATION

```
#include <Otto.h>
#define LeftLeg 2
#define RightLeg 3
#define LeftFoot 4
#define RightFoot 5
#define Buzzer  13
#define Trigger 8 // ultrasonic sensor trigger pin
#define Echo 9 // ultrasonic sensor echo pin

long ultrasound() {
  long duration, distance;
  digitalWrite(Trigger,LOW);
  delayMicroseconds(2);
  digitalWrite(Trigger, HIGH);
  delayMicroseconds(10);
  digitalWrite(Trigger, LOW);
  duration = pulseIn(Echo, HIGH);
  distance = duration/58;
  return distance;
}

void setup() {
  Otto.init(LeftLeg, RightLeg, LeftFoot, RightFoot, true, Buzzer); //Set the servo pins and Buzzer pin
```

```
  pinMode(Trigger, OUTPUT);
  pinMode(Echo, INPUT);

}

void loop() {
    if (ultrasound() <= 15) {
      Otto.sing(S_surprise);
      Otto.playGesture(OttoConfused);
      Otto.walk(2,1000,-1); // BACKWARD x2
      Otto.turn(3,1000,1); // LEFT x3
    }
    Otto.walk(1,1000,1); // FORWARD x1

}
```

CHAPTER 5

RESULTS AND DISCUSSION



The Otto robot was successfully designed, assembled, and programmed using an Arduino Nano microcontroller, ultrasonic sensor, and servo motors. The robot was able to perform various movements such as walking, turning, and dancing. With the integration of the ultrasonic sensor, Otto effectively detected obstacles in its path and responded by changing direction, demonstrating basic autonomous behavior. The 3D-printed body provided a lightweight and modular structure, making it easy to assemble and customize. The code was written and uploaded using the Arduino IDE, and the robot operated as intended with all components functioning correctly.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

The Otto robot project demonstrated a successful implementation of a simple, interactive, and educational bipedal robot. Through the use of open-source hardware and software, the project provided hands-on experience in electronics, programming, and mechanical design. The integration of the ultrasonic sensor allowed Otto to avoid obstacles, adding an element of autonomy and intelligence to its behavior. The project not only fulfilled its goal of building a functioning robot but also served as a practical learning tool for understanding basic robotics concepts. With its modular and customizable nature, Otto can be further expanded and enhanced for more advanced applications in the future. The Otto robot offers a solid foundation for exploring more advanced features in robotics and automation. Its modular design and open-source nature make it ideal for upgrades and experimentation. Future enhancements can significantly expand its capabilities and educational value:

REFERENCES

[1] "Exploring Arduino: Tools and Techniques for Engineering Wizardry" *Author:* Jeremy Blum *Publisher:* Wiley

[2] "Arduino Workshop: A Hands-On Introduction with 65 Projects" *Author:Boxall* John *Publisher No Starch Press*

[3]"Beginning Robotics with Raspberry Pi and Arduino" *Author:Jeff* cicolani *Publisher:* Apress

ASSESSMENT

SL NO	RUBRICS	FULL MARK	MARKS OBTAINED	REMARKS
1	Understanding the relevance, scope and dimension of the project	10		
2	Methodology	10		
3	Quality of Analysis and Results	10		
4	Interpretations and Conclusions	10		
5	Report	10		
	Total	**50**		

Date: **Signature of the Faculty**