ISSN 2277-2685
IJESR/Jan-Mar. 2026/ Vol-16/Issue-1/264-270
Sampeta Ankitha *et. al.*, /International Journal of Engineering & Science Research

Obstacle Avoiding Robot

Dr.K.Ashok Kumar[1], Sampeta Ankitha[2], Vattikoti Gnaneshwari[3], Mallela Neha[4]
[1]Associate Professor; Bhoj Reddy Engineering College For Women Department Of Electronics And Communication Engineering
[2,3,4]B.Tech Students; Bhoj Reddy Engineering College For Women Department Of Electronics And Communication Engineering
sampetaankitha@gmail.com[2],Vgnaneshwari12@gmail.com[3],mallelaneha29@gmail.com[4]

Abstract

In Today's World Robotics is a fast-growing and very interesting field. The concept of Robotics is now used in every sector whether it is in manufacturing industry, medical, transport, etc.

The project is designed to build an obstacle avoidance robotic vehicle using ultrasonic sensors for its movement. Arduino is used to achieve the desired operation. A Robot is a machine that can perform task automatically or with Guidance. The Project proposes Robotic vehicle that has an intelligence built in it such that it directs itself whenever an obstacle comes in its path. This Robotic vehicle is built, using Arduino. An Ultrasonic sensor is used to detect any obstacle ahead of it and sense a command to the Arduino. Depending on the input signal received, the Arduino redirects the robot to move in an alternate direction by activating the motors which are interfaced to it through a motor driver. At the same time it can control steering gear to realize the obstacle avoidance function. The Robot car uses front axle steering, rear wheel drive arrangement. Two drive tyres are driven by two DC Motors with gear reduction mechanisms.

Keywords: *Robotics, Obstacle Avoidance, Arduino, Ultrasonic Sensor, Autonomous Robot, Embedded Systems, Motor Driver, DC Motors, Sensor-Based Navigation, Mobile Robot, Collision Detection, Control Systems*

INTRODUCTION

In Robotics, Obstacle Avoidance is the task of satisfying some control objective subject to non-intersection or non-collision positions constraints. In unmanned air vehicles, it is a hot topic. What is critical about obstacle avoidance concept in this area is the growing need of usage of unmanned aerial vehicles in urban areas for especially military applications where it can be very useful in city wars.

Normally Obstacle Avoidance is considered to be distinct from path planning in that one is usually implemented as a reactive control law while the other involves the pre- computation of an obstacle free path which a controller will then guide a robot along. With recent advanced in the autonomous vehicles sector, a good and dependable obstacle avoidance feature of a driverless platform is also required to have a robust obstacle detection module.

Reactive obstacle avoidance is a behaviour-based control strategy in a robot. It is a task similar to the navigation problem and produces a collision free motion.

This capability allows the visual warning lights to remain passive until an aircraft is detected and known to be tracking on an unsafe heading. This leaves the night time sky free of unnecessary light pollution thus decreasing public annoyance issues while improving the environmental habitat.

An autonomous robot is a robot that is capable of moving on its own in an unknown and unstructured environment. An autonomous robot is equipped with software intelligence to sense its environment, detect obstacles in its path and move around an unknown environment overcoming the obstacles. There are many robotic designs that are employed in designing of autonomous robots. These designs are usually developed considering the physical environment in which the robot has to be deployed. There are autonomous robots like snake robots, walking robots, autonomous drones and autonomous robotic cars or rovers.

This ROBOT has sufficient intelligence to cover the maximum area of provided space. It has an ultrasonic sensor which are used to sense the obstacles coming in between the path of ROBOT.

It will move in a particular direction and avoid the obstacle which is coming in its path. The main motto of designing such type of Robot or the technology is that this technology can be used in today's very fast transportation to avoid the accident generally happen in congested by applying emergency break. If we use this technology in the car or any vehicle, it will automatically sense the obstacles then it will take a side to the available free space. An Obstacle may be a living thing or any object.

Autonomous Intelligent Robots are robots that can perform desired tasks in unstructured environments without continuous human guidance. Thus, by using this technology in vehicles we make the drive safe.

AIM OF THE PROJECT

The Aim of the Project "Obstacle Avoiding Robot

using Arduino Uno" is to design and build a Robotic system that can autonomously navigate an environment by detecting and avoiding obstacles. This involves integrating sensors (such as ultrasonic or infrared sensors), actuators (such as motors and wheels), and control logic programmed into the Arduino Uno. The key objectives of the project include:

Autonomous Navigation: Enabling the robot to move through its environment without human intervention.
Obstacle Detection: Using sensors to detect obstacles in the robot's path.
Path Planning and Avoidance: Implementing algorithms that allow the robot to change its path to avoid obstacles.
Real-Time Decision Making: Ensuring the robot can process sensor data in real-time and make decisions quickly.
Robustness and Reliability: Creating a system that can handle various types of obstacles and navigate different environments effectively.

MOTIVATION

The motivation behind building an obstacle-avoiding robot using Arduino is to design a smart and autonomous system that can navigate through a dynamic environment without human intervention. By integrating sensors, motors, and Arduino's microcontroller, the robot can detect and respond to obstacles, enhancing its efficiency and safety. This project aims to develop a cost-effective and innovative solution for various applications, such as surveillance, exploration, and assistance, while also providing a platform for learning and experimentation in robotics, programming, and sensor technologies. By overcoming obstacles, the robot can achieve its goals and demonstrate its potential in real-world scenarios.

OBJECTIVES

Navigation: To move from a starting point to a destination without colliding with obstacles.
Obstacle Detection: To accurately detect and identify obstacles in the robot's path.
Path Planning: To calculate an optimal path that avoids obstacles and reaches the destination.
Collision Avoidance: To prevent collisions with obstacles and ensure safe navigation.
Adaptability: To adapt to changing environments and unexpected obstacles.
Efficiency: To navigate through environments with minimum time and energy consumption.
Accuracy: To accurately detect and respond to obstacles with high precision.
Reliability: To consistently perform obstacle avoidance tasks without failure.
Autonomy: To operate independently without human intervention.
Flexibility: To navigate through various environments and scenarios.

Software And Hardware Requirements

2.1

Introduction To Arduino Ide

IDE stands for Integrated Development Environment. Pretty fancy sounding, and should make you feel smart any time you use it. The IDE is a text editor-like program that allows you to write Arduino code. When you open the Arduino program, you are opening the IDE. It is intentionally streamlined to keep things as simple and straightforward as possible. When you save a file in Arduino, the file is called a sketch – a sketch is where you save the computer code you have written. The coding language that Arduino uses is very much like C++ ("see plus plus"), which is a common language in the world of computing. The code you learn to write for Arduino will be very similar to the code you write in any other computer language – all the basic concepts remain the same – it is just a matter of learning a new dialect should you pursue other programming languages. The code you write is "human readable", that is, it will make sense to you (sometimes), and will be organized for a human to follow. Part of the job of the IDE is to take the human readable code and translate it into machine-readable code to be executed by the Arduino.

Figure 2.1 Arduino

This process is called compiling. The process of compiling is seamless to the user. All you have to do is press a button. If you have errors in your computer code, the compiler will display an error message at the bottom of the IDE and highlight the line of code that seems to be the issue.

Figure 2.2: Arduino Desktop(Default Layout)

The error message is meant to help you identify what you might have done wrong – sometimes the message is very explicit, like saying, "Hey – you forget a semicolon", sometimes the error message is vague. Why be concerned with a semicolon you ask? A semicolon is part of the Arduino language syntax, the rules that govern how the code is written. It is like grammar in writing. Say for example we didn't use periods when we wrote – everyone would have a heck of a time trying to figure out when sentences started andended. Or if we didn't employ the comma, how would we convey a dramatic pause to the reader?

Introduction To Arduino Uno

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language(based on Wiring), and the Arduino Software (IDE), based on Processing.

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

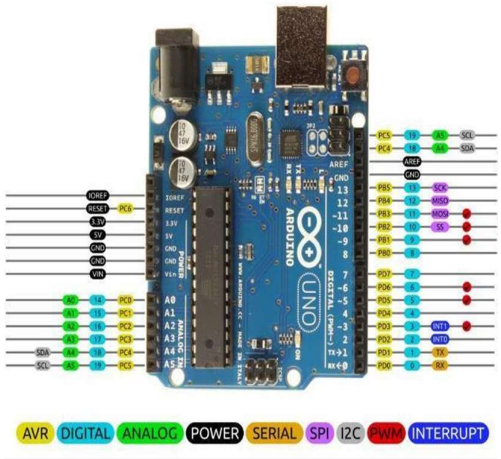

Figure 2.3: Arduino UNO R3 Pinout

WHY ARDUINO

Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low-cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes,

musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker

Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step-by-step instructions of a kit, or sharing ideas online with other members of the Arduino community.

ADVANTAGES OF ARDUINO

Inexpensive- Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and

even the pre-assembled Arduino modules cost less than $50

Cross-platform- The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

- Simple, clear programming environment- The Arduino Software (IDE) is easy-

to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.

Open source and extensible software- The Arduino software is published as opensource tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

Open source and extensible hardware- The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

Features Of Arduino Uno

The Arduino Uno is a microcontroller board based on the ATmega328. Arduino is an open source, prototyping platform and its simplicity makes it ideal for hobbyists to use as well as professionals. The Arduino Uno has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable.

Features of the Arduino UNO:

- Microcontroller: ATmega328
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Input Voltage (limits): 6-20V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 6
- DC Current per I/O Pin: 40 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB of which 0.5 KB used by bootloader
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Clock Speed: 16 MHz

2.1.2 L293D MOTOR DRIVER IC

Figure 2.4 L293D Pinout Diagram

L293d IC is known as a motor driver. It is a low voltage operating device like other ICs. The other ICs could have the same functions like L293d but they cannot provide the high voltage to the motor. L293d provides the continuous bidirectional Direct Current to the Motor.

The Pin Configuration Details are:

- **Pin1 (Enable):** Pin 1 is known as the enable pin. It has a major effect on Input and output. If there is High logical signal on enable pin (EN) then there will be input and output between pin 2,3,6 & 7 (Input 1, Output 1, Input 2 & Output 2).
- **Pin2 (Input 1):** Mostly input means where we provide the input to give the output. But here Input 1 means which polarity we want to give at Output 1.
- **Pin3 (Output 1):** Output 1 is the input of the first motor/Motor 1. It attaches to its one end.
- **Pin4 (Ground):** The ground pin will attach to the ground of the circuit.
- **Pin5 (Ground):** The ground pin will be attached to the ground, and it will remain common with the previous ground.
- **Pin6 (Output 2):** Output 2 will attach to the input of the first motor/Motor 1. It will attach to its second end.
- **Pin7 (Input 2):** Input 2 will attach to the control button or device to control the Output 2 just like Input 1.

- **Pin8 (Vcc):** Pin8 is the voltage pin for Motor. It will device how much power we are going to attach the Motor. This Power should not be more than 36 volts and should not be less than 4.5 volts.
- **Pin9 (Enable):** Pin 9 is also the same as Pin 1. It controls the input and output signals. Pin 9 Controls the connection between Input 3, Input 4, Output 3 and Output 4. It also enables the connection when the logic signal will be High (1).
- **Pin10 (Input 3):** Input 3 will control the output polarity of the Pin 11 (Output 3) by logic signals.
- **Pin11 (Output 3):** Output 3 will be connected to the one end of the second motor.
- **Pin 12 (Ground):** Pin 12 will attach to the common ground with all other grounds.
- **Pin13 (Ground):** Pin 13 will also be attached to the common ground with all other grounds.
- **Pin14 (Output 4):** Pin 14 will attach to the second end of the second motor.
- **Pin15 (Input 4):** Pin 15 will control the output polarity of the Pin 14 (Output 4) by logic signals.
- **Pin16 (Vcc):** Pin 16 will the Power we will provide to the L293D to activate it or to turn it on. The power level of Pin 16 should be 4.5 – 7Volts.

BLOCK DIAGRAM EXPLANATION

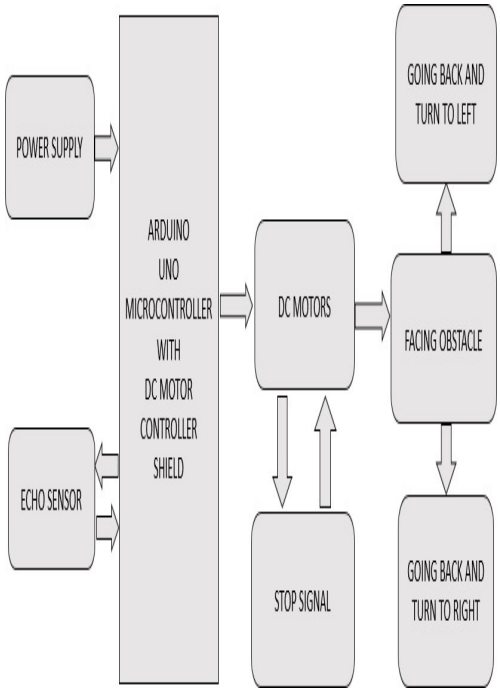

Figure 3.1: Block Diagram

The basic block diagram of the obstacle avoiding robot is shown in above figure. Mainly this block diagram consists of the following essential blocks:

ARDUINO UNO:

The brain of the robot responsible for processing sensor data and controlling the motors. Its runs the obstacle avoiding algorithm,interpreting sensor inputs and generating motor outputs.The Arduino UNO is a microcontroller board based on the ATmega328P chip.It is used for controlling all the operation and assign task to each device.

3.1.1. **ULTRASONIC SENSOR:**

These sensors use high frequency sound waves to detect obstacles .They transmit and receive sound waves ,calculating distance based on time delay .The sensors provide digital output indicating the presence or absence of an obstacle.

3.1.3 **MOTOR DRIVER(L293D):**

The L293D is a dual H-bridge driver ,allowing for forward and backward motion .This component enables the Arduino to control the motors .It takes the digital output from the Arduino generates the necessary voltage and current to drive the motors.

3.1.4 DC MOTORS:

These motors provide the locomotion for the robot. They are controlled by the motor driver, reviving the necessary voltage and current to move the robot. The Motors are designed for efficient movement and precise control.

3.1.5 POWER SUPPLY:

This component provides the necessary power to the robot. It supplies voltage and current to the Arduino, motors, and sensors. A Suitable power supply ensures stable operation and prevents damage to the components.

Circuit Diagram And Working Principle

This circuit diagram shows the connections for building an obstacle avoiding robot using Arduino, featuring an Arduino board, ultrasonic sensor (HC-SR04) for detecting obstacles, motor driver (L293D) for controlling movement, DC motors for propulsion, power supply, and breadboard connections. The Arduino reads ultrasonic sensor data, processes it, and controls the motor driver to move the robot accordingly, enabling it to navigate and avoid obstacles in its environment.

CIRCUIT DIAGRAM EXPLANATION

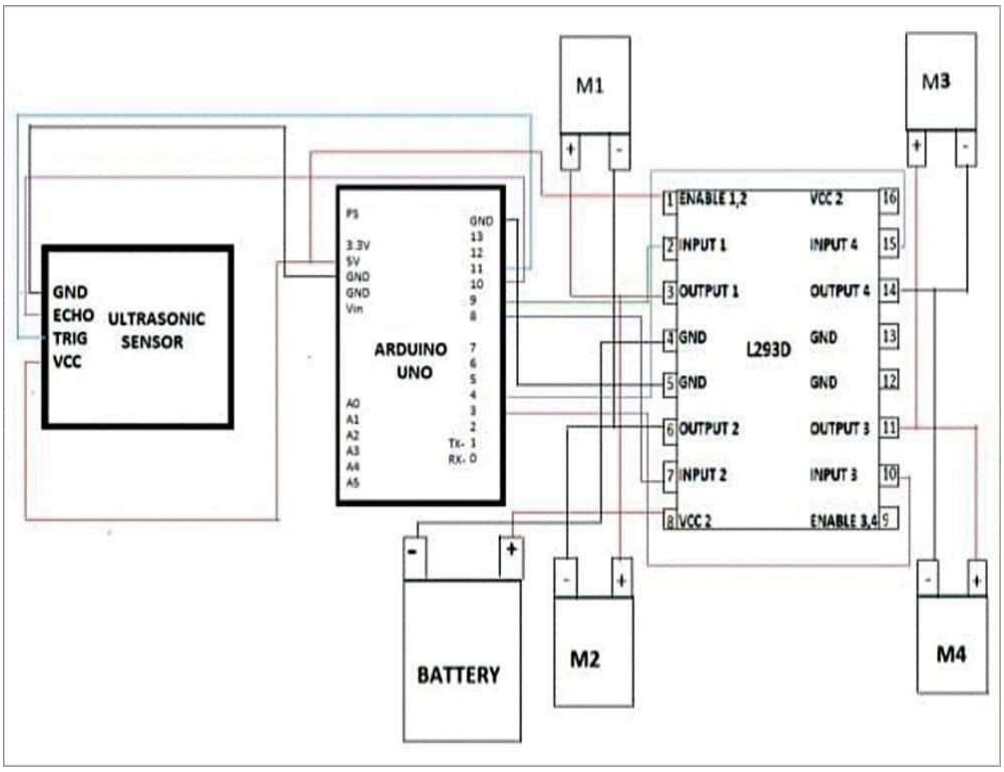

Figure 4.1: Circuit Diagram

RESULT

The Obstacle Avoiding Robot using Arduino Uno is a remarkable project that showcases the potential of robotics and automation in navigating unknown environments. Through the successful integration of ultrasonic sensors, Arduino Uno, and L293D motor driver, the robot demonstrates its ability to detect and avoid obstacles with precision and efficiency.

The robot's navigation system relies on the accurate detection of obstacles by the ultrasonic sensors, which transmit and receive sound waves to calculate distance. The sensors provide a digital output indicating the presence or absence of an obstacle, which is then processed by the Arduino Uno board. The board interprets the sensor data and generates motor outputs, controlling the robot's movement through the L293D motor driver.

The L293D motor driver plays a crucial role in managing the motor speed and direction, ensuring smooth and precise movement. The driver's enable pins allow for efficient control of the motor's direction, while the input pins receive logic signals from the Arduino Uno to control the motor's speed. The driver's output pins connect to the motors, providing the necessary voltage and current to drive the robot.

The robot's ability to avoid obstacles is a testament to the effectiveness of the Arduino Uno in processing sensor data and controlling the motors. The board's microcontroller, the ATmega328P, processes the sensor data at a speed of 16 MHz, enabling real-time processing and decision-making. The board's digital I/O pins connect to the sensors and motor driver, providing a seamless interface for data transmission and control.

The project demonstrates the versatility and ease of use of Arduino Uno in robotics and automation. The board's simplicity and flexibility make it an ideal platform for prototyping and development, allowing for quick and easy modifications to the robot's design and functionality. The project's success highlights the potential of Arduino Uno in enabling innovative solutions for real-world applications, such as surveillance, exploration, and assistive technology.

In conclusion, the Obstacle Avoiding Robot using Arduino Uno is a remarkable project that showcases the potential of robotics and automation in navigating unknown environments. Through the successful integration of sensors, Arduino Uno, and motor driver, the robot demonstrates its ability to detect and avoid obstacles with precision and efficiency. The project highlights the versatility and ease of use of Arduino Uno, paving the way for further innovations and developments in the field.

CONCLUSION

Obstacle avoidance capability needs to be considered when designing mobile robots for different applications. The low-cost ultrasonic sensor for mobile robot is aim to design and implement a helpful tool that improves the ability of mobile robot to avoid obstacle successfully. A series of test were done to check the reliability of the system.

In our experiment the ultrasonic distance sensing element was accustomed to offer a large field of detection. Which can be implemented on mobile robots both remotely controlled and also on autonomous mode, once in the autonomous mode, the initial loading of the code needs no user intervention throughout its operation.

The project is “obstacle detection and the avoidance robot” is practically proved by using the Ultrasonic sensor for sensing the robot, Motor Shield Driver for the driving the dc motors, dc motor is used for the movement of the robot with the help of the Arduino Microcontroller.

A lot of factors determined the accuracy of the robot we designed. These factors were the environmental phenomenon in which the robot was tested, the number of obstacles present making the test space crowded or relatively less crowded the type and shape of the obstacle (the robot is designed for a uniform shaped obstacle).

These factors majorly affected the sensors. The accuracy of the robot is dependent on the sensors used.

Thus, the nature of the sensor and its accuracy defined the accuracy of the robot.

When it is placed in an unknown setting with obstacles, it runs while avoiding all obstacles with significant accuracy. RESULT demonstrated high accuracy of the ultrasonic sensor to avoid obstacle.

REFERENCES

[1]. M. A. Ahasan, S. A. Hossain, A. U. Siddique & Rahman 2012 Obstacles Invariant Navigation of An Autonomous Robot based on GPS. Khulna University.

[2]. Kam, M., ZHU, X. & Kalata, P 1997 Sensor Fusion for Mobile Robot Navigation. Proceedings of the IEE 85(1) 108 – 119.

[3]. Azeta Joseph, Bolu Christian, Abioye A. Abiodun and Festus Oyawale 2018 A review on humanoid robotics in healthcare. MATEC Web Conf.., 153 02004.

[4]. C. A Ryther, & O. B. Madsen 2009 Obstacle Detection and Avoidance for Mobile Robots.

[5]. The Technical University of Denmark.

[6]. Ramaswamy, S., Sabatini, R., Gardi, A. & Liu, J 2016 LIDAR obstacle warning and avoidance system for unmanned aerial vehicle sense-and-avoid. Aerospace Science and Technology, 55, 344-358.