Autonomous moving Bot using Arduino Duemilanove and various sensors

Abstract

This project focuses on the design and implementation of an autonomous driver using the Arduino Duemilanove microcontroller and a variety of sensors including infrared (IR), gyroscope, and sonar sensors. The main goal is to create an intelligent robot that can control itself and is environmentally conscious. Arduino Duemilanove plays an important role in the control room by managing the interaction between sensors and actuators. The addition of infrared sensors allows the robot to detect objects in its environment, while gyroscopes provide information about its orientation and speed angle. Sonar sensors increase spatial awareness by measuring distance, supporting anti-jamming and navigation.

This project involves integrating sensor data into complex control systems. The algorithm allows the robot to react to the environment by controlling its movement and decision-making. Thanks to the combination of sensor fusion and advanced control, self-driving cars are designed to provide efficient and seamless navigation. Additionally, the project is exploring the possibility of integrating additional functions such as wireless communication modules and remote monitoring management. The versatility of the system makes it suitable for many applications, from indoor navigation to tracking tasks, demonstrating the adaptability and scalability of the proposed robotic platform.

Acknowledgment

I, K. Srujan, wish to express my heartfelt gratitude to several individuals who have played a pivotal role in the successful completion of my individual project, the Autonomous Moving Bot, undertaken during my participation in a summer robotics camp in my 4th standard. Completing this project has been a personal milestone, and I am grateful for the support and encouragement from my Parents. Their contributions have made this journey both fulfilling and educational.

List of Figures

List of Abbreviations

IR	Infrared radiation
PID	Proportional – Integral – Derivative
I2C	Inter-Integrated Circuit
SPI	Serial peripheral interface
LED	Light Emitting Diode
IDE	Integrated Drive Electronics
ADC	Analog-to-Digital Converter

Contents

Chapter-1

Introduction

This project focuses on the design and implementation of an autonomous driver using the Arduino Duemilanove microcontroller and a variety of sensors including infrared (IR), gyroscope, and sonar sensors. The main goal is to create an intelligent robot that can control itself and is environmentally conscious. Arduino Duemilanove plays an important role in the control room by managing the interaction between sensors and actuators. The addition of infrared sensors allows the robot to detect objects in its environment, while gyroscopes provide information about its orientation and speed angle. Sonar sensors increase spatial awareness by measuring distance, supporting anti-jamming and navigation.

This project involves integrating sensor data into complex control systems. The algorithm allows the robot to react to the environment by controlling its movement and decision-making. Thanks to the combination of sensor fusion and advanced control, self-driving cars are designed to provide efficient and seamless navigation.

Additionally, the project is exploring the possibility of integrating additional functions such as wireless communication modules and remote monitoring management. The versatility of the system makes it suitable for many applications, from indoor navigation to tracking tasks, demonstrating the adaptability and scalability of the proposed robotic platform.

Overall, this project is not only to demonstrate the use of robots, but also to understand the integration of sensor, control machine and microcontroller programming. Autonomous mobile robots represent a major opportunity for robotics and demonstrate advances in self-control and intelligent decision-making.

Project Overview and Motivation

The Autonomous Mobile Robot project revolves around the development of a robot system that can move and navigate independently using the Arduino Duemilanove microcontroller and various sensors such as infrared (IR), gyroscope and sonar sensors. The project aims to create a versatile, adaptable robotic platform that can sense the environment and make smart decisions autonomously. The primary motivation behind this project is to advance the autonomy of robotic systems. By integrating various sensors and a robust control algorithm, the robot gains the ability to perceive its environment and respond intelligently, reducing the need for constant human intervention. Autonomous robots find applications in diverse fields, from warehouse logistics to environmental monitoring. This project seeks to explore the practical implementation of autonomous navigation, paving the way for potential applications in industries where precise and independent movement is crucial. The project serves as an educational tool for robotics enthusiasts, students, and researchers. It provides hands-on experience in designing and implementing control algorithms for autonomous systems, fostering a deeper understanding of robotics principles. The integration of sensors and microcontrollers represents the exploration of cutting-edge technologies. The project encourages the use of open-source platforms like Arduino for fostering innovation in robotics. The challenges involved in creating an autonomous moving bot, such as obstacle detection, navigation, and decision-making, contribute to problem-solving skills. Overcoming these challenges requires a multidisciplinary approach, integrating hardware and software components seamlessly.

Chapter-2

Literature Survey

2.1 Literature survey Table

Sr.No	Author Name	Paper Title
1	Mahmud Kazi ,Abdullah Nahis, Abdullah Al Mamun	Implementation Of Autonomous Line Follower Robot
2	Kumar Rishabh	Design of autonomous line follower robot with obstacle avoidance
3	Haruka Kido, James Vrtis, Luke Anderson, Tarek Elderini	Autonomous Gyroscopic 2-Wheel Differential Robot
4	Sibel Tariyan Özyer	Remote and Autonomous Controlled Robotic Car based on Arduino with Real Time Obstacle Detection and Avoidance

2.2 Advantages and Disadvantages of various Micro Controllers which can be used

Advantages of using Arduino Duemilanove:

- Ease of Use: Arduino Duemilanove is known for its simplicity and user-friendly interface. It is an excellent choice for beginners and those new to microcontroller programming.
- Large Community Support: Arduino has a vast and active community, making it easy to find resources, tutorials, and troubleshooting assistance online.
- Wide Range of Libraries: Arduino offers a wide array of libraries that simplify the programming process. These libraries cover various sensors and devices, making it easier to interface with different components.
- Versatility: Arduino Duemilanove can be used for a variety of projects, including robotics, home automation, and sensor-based applications.
- Open-Source Platform: Arduino is an open-source platform, allowing users to modify and share their code and designs freely.

Disadvantages of using Arduino Duemilanove:

- Limited Processing Power: Arduino Duemilanove has a relatively lower processing power compared to more advanced microcontrollers, which might limit its capabilities for complex tasks.
- Limited Memory: The onboard memory of Arduino Duemilanove is limited, restricting the size and complexity of programs that can be executed.
- Not Suitable for High-End Applications: For applications requiring advanced features, real-time processing, or complex algorithms, Arduino Duemilanove might not be the best choice.
- Older Technology: Arduino Duemilanove is based on the ATmega328 microcontroller, which is considered somewhat outdated compared to more modern microcontrollers.

Arduino Uno:

Advantages:

- Similar ease of use and community support as Duemilanove.
- Slightly updated features and capabilities.

Disadvantages:

- Still limited processing power and memory.

Raspberry Pi Pico:

Advantages:

- Lower cost compared to Arduino boards.
- Higher processing power and memory for more demanding applications.

Disadvantages:

- Different programming environment (MicroPython).

ESP8266 (NodeMCU):

Advantages:

- Integrated Wi-Fi capabilities.
- Suitable for IoT projects.

Disadvantages:

- Limited GPIO pins compared to Arduino

.

STM32 Blue Pill:

Advantages:

- More processing power and memory.
- Suitable for applications requiring higher performance.

Disadvantages:

- Steeper learning curve for beginners.
- Limited community support compared to Arduino.

Chapter-3

Project Design and Description

3.1 Description (Work Done)

Line Following Robot using Arduino Duemilanove and IR Sensors:

Working Principle: The Line Following Robot is designed to follow a predefined path marked by a contrasting line on the ground. It employs infrared (IR) sensors to detect the line and make necessary adjustments in its movement to stay on the track. The Arduino Duemilanove microcontroller processes the sensor data and controls the motors to ensure the robot's proper navigation along the line.

Components:

- Arduino Duemilanove Microcontroller: Acts as the brain of the robot, responsible for processing sensor data and controlling the motors.
- IR Sensors: These sensors are positioned on the underside of the robot, facing the ground. They consist of an IR transmitter and receiver pair. The IR transmitter emits infrared light, and the receiver detects the reflected light. When the robot is over the line, the IR light is reflected differently, allowing the sensors to detect the line.

Connections:

IR Sensor Connections:

1. Connect the VCC pin of each IR sensor to the 5V output of Arduino.
2. Connect the GND pin of each IR sensor to the GND pin of Arduino.
3. Connect the OUT pin of each IR sensor to digital input pins on Arduino (e.g., D2 and D3).

Motor Connections:

1. Connect the motor driver module to the Arduino.
2. Connect the motor terminals to the motor driver outputs.
3. The motor driver is used to control the speed and direction of the motors.

Power Supply:

Connect a power source to the Arduino and motor driver to ensure sufficient power for both the microcontroller and motors.

Line Following Algorithm:

- Read Sensor Data: The IR sensors continuously detect the color contrast between the line and the surrounding surface.
- Data Processing: Arduino processes the sensor data to determine the robot's position relative to the line.
- Motor Control: Based on the processed data, the Arduino adjusts the motor speeds to keep the robot aligned with the line.
- Continuous Loop: The robot continuously repeats the process, making real-time adjustments to maintain its course.

Obstacle Avoidance using Arduino Duemilanove and IR Sensors:

Working Principle: Obstacle avoidance is an additional feature where the robot can detect and navigate around obstacles using IR sensors.

Additional Components:

- Obstacle Detection IR Sensors: Install additional IR sensors on the sides or front of the robot to detect obstacles.

Connections:

Obstacle Detection IR Sensor Connections:

1. Connect the additional IR sensors in a similar manner to the line-following IR sensors.
2. Utilize separate digital input pins on Arduino for obstacle detection.
3. Obstacle Avoidance Algorithm:
4. Read Obstacle Sensor Data: The additional IR sensors detect obstacles.
5. Data Processing: Arduino processes the obstacle sensor data to determine the location and proximity of obstacles.
6. Motor Control: Based on the obstacle data, Arduino adjusts the motor speeds and direction to navigate around the obstacle.
7. Continue Line Following: After avoiding the obstacle, the robot resumes line following.

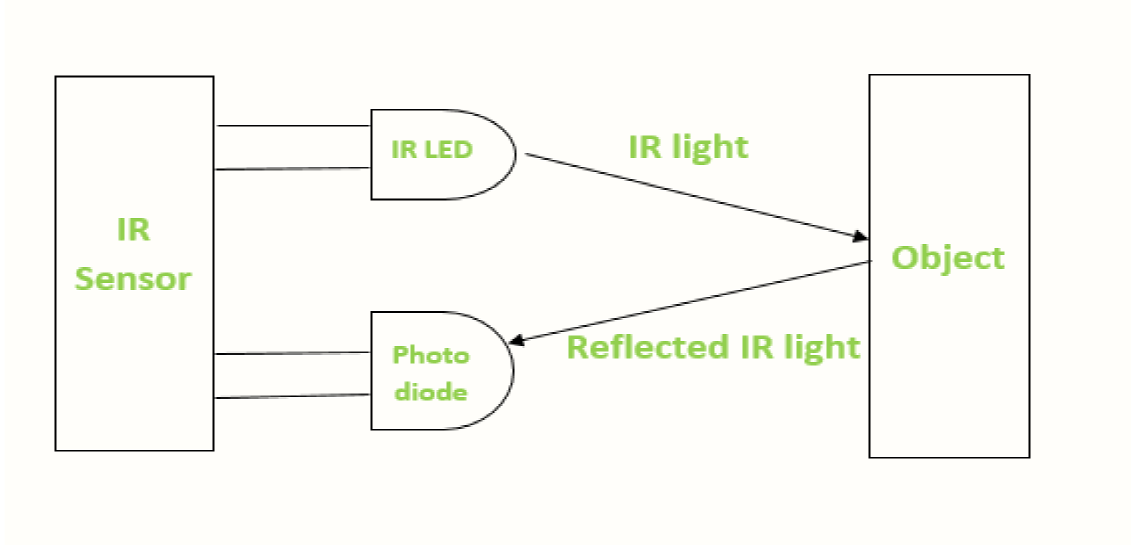


Fig1.6 :IR sensor for Obstrucle avoidance

Moving Bot with Arduino Duemilanove and Gyro Sensor: Direction Control and Connections:

Working Principle: A Gyro Sensor measures the rate of angular velocity or rotation around various axes. In this project, the Gyro Sensor provides real-time data on the bot's orientation, allowing it to make informed decisions about direction changes. This real-time feedback is crucial for achieving precise and controlled movements.

Components:

- Arduino Duemilanove: The central control unit, Arduino Duemilanove, serves as the brain of the autonomous bot. It processes data from various sensors, including the Gyro Sensor, and adjusts the bot's behavior accordingly.
- Gyro Sensor: The Gyro Sensor connects to the Arduino Duemilanove, typically using the I2C or SPI communication protocol. Ensure proper pin connections, referencing the datasheets of both the sensor and the Arduino for accurate wiring.
- Power Supply: Both the Arduino Duemilanove and the Gyro Sensor require a stable power supply. Connect the power sources according to the specifications of each component. It's crucial to provide sufficient power for seamless operation.
- Motor Control: If the autonomous bot has wheels with motors, connect the motor control pins to the Arduino. The Gyro Sensor's data will be used to adjust motor speeds and achieve precise turns.

Benefits of Gyro Sensor Integration:

- Precision: The Gyro Sensor provides precise rotational data, enhancing the bot's ability to maintain accurate headings.
- Real-time Adjustments: Continuous updates from the Gyro Sensor enable the Arduino to make instantaneous adjustments, improving responsiveness.
- Improved Maneuverability: With better direction control, the bot can navigate through tight spaces and adapt to changing environments effectively.

3.3 Architecture

Architecture of Arduino Duemilanove:

The Arduino Duemilanove relies on the ATmega328 microcontroller, employing an 8-bit AVR architecture running at 16 MHz. This microcontroller serves as the brain of the system, managing the execution of the programmed code, handling input and output operations, and interfacing with various sensors and actuators. The board features 32 KB of flash memory for storing the user's program, commonly known as the Arduino sketch. Additionally, it is equipped with 2 KB of SRAM for storing variables and data during program execution, and 1 KB of EEPROM for non-volatile data storage. The availability of digital and analog I/O pins facilitates the connection of sensors and peripherals, while the USB interface enables programming and serial communication with an external computer.

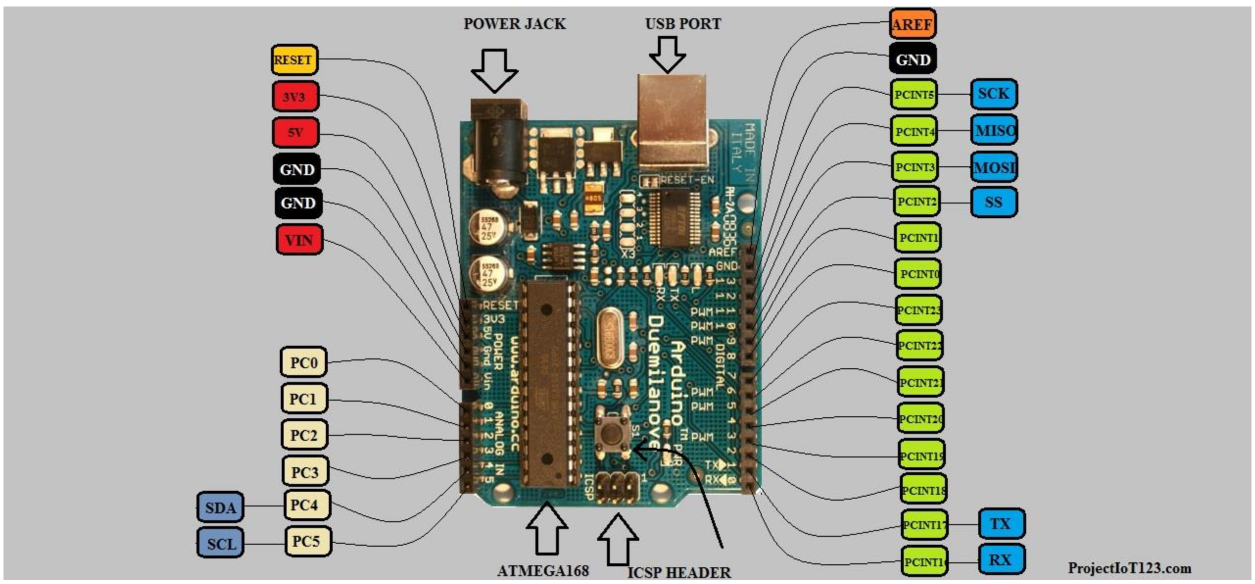


Fig 1.0:Arduino Duemilanove Architecture

IR Sensor Architecture:

In the realm of robotics, IR sensors play a crucial role in detecting proximity and reflectivity. These sensors typically consist of an IR emitter and receiver pair. The IR emitter, usually composed of IR LEDs, emits infrared light. The IR receiver detects the reflected IR light, and the intensity of the received signal varies based on the proximity and reflectivity of objects in the sensor's field. Signal processing circuitry within the IR sensor converts the analog signal from the receiver into a digital signal that can be interpreted by the Arduino microcontroller.

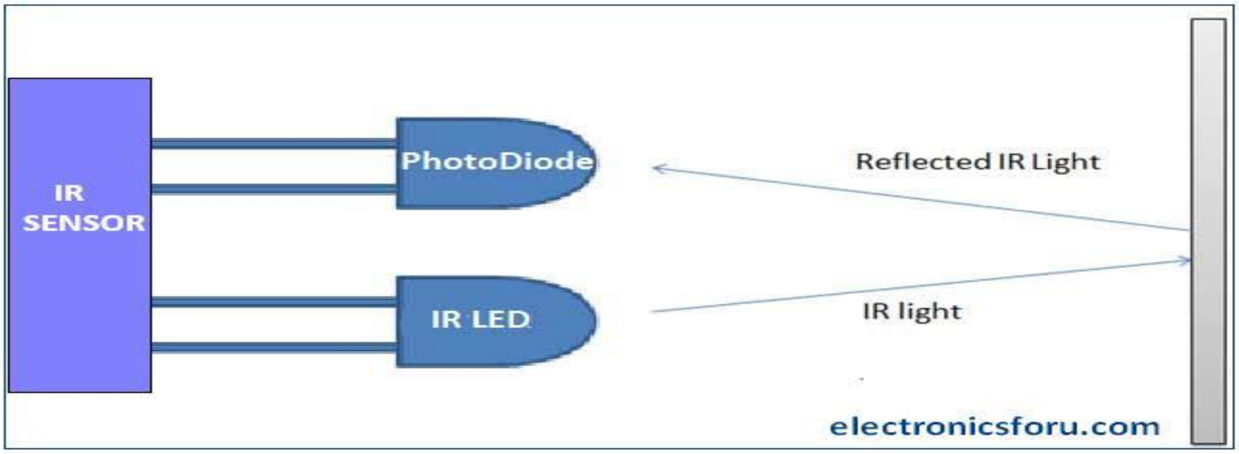


Fig1.7:IR Sensor working

Gyro Sensor Architecture:

Gyroscopes, essential for measuring orientation and rotation, are a key component in the navigation system. A typical gyroscope setup includes a gyroscopic sensor responsible for detecting rotational motion. This sensor produces an analog output, which is converted into digital signals by an Analog-to-Digital Converter (ADC). The output interface then provides processed data, such as angular velocity or orientation, to the Arduino microcontroller. The gyroscope's ability to sense changes in orientation enables the robot to adjust its movements based on changes in direction.

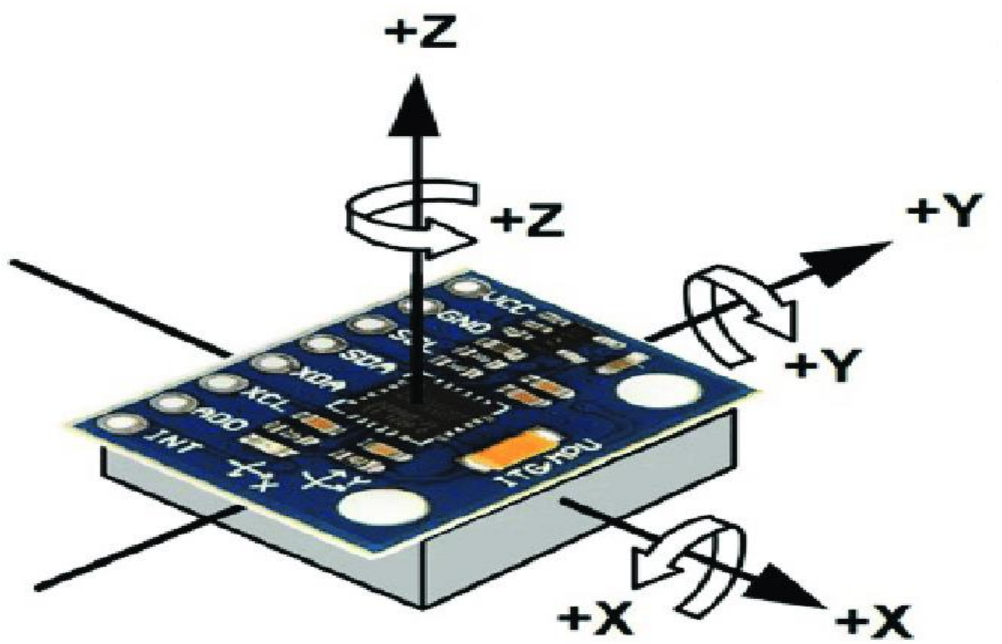
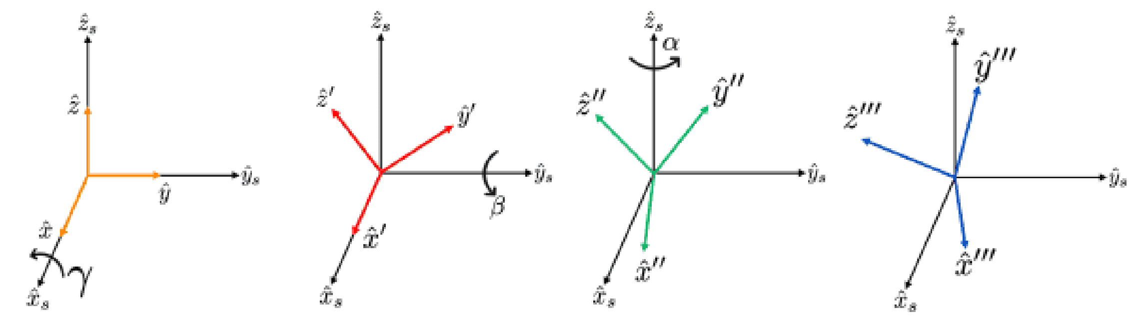


Fig 1.8:Gyroscope sensor working

Maneuvering Gyro Directional Algorithm:

The maneuvering gyro directional algorithm is a crucial aspect of autonomous navigation. The process involves continuous data acquisition, where gyro sensors measure the robot's angular velocity. The Arduino microcontroller reads these measurements at regular intervals. Using the gyro data, the microcontroller calculates the current orientation or rotational angle of the robot. The directional control is then adjusted based on this information. Optionally, a Proportional, Integral, and Derivative (PID) control system may be employed to fine-tune rotational stability. PID parameters are adjusted based on gyro feedback to minimize errors. Finally, the Arduino sends control signals to motor drivers, adjusting wheel speed and direction. Differential drive algorithms enhance precise maneuvering based on gyro feedback, allowing the robot to autonomously navigate and adjust its direction based on its surroundings. The effectiveness of this system depends on factors such as sensor accuracy, algorithm design, and PID tuning.



{b} is initially coincident with {s}

frame achieved after a rotation about space frame's x-axis by γ

frame achieved after a rotation about space frame's y-axis by β

frame achieved after a rotation about space frame's z-axis by α

Fig1.9: Gyro Directional Algorithm

3.4 Tools and Technology

Software Used in Arduino Duemilanove:

The Arduino Duemilanove relies on the Arduino Integrated Development Environment (IDE), a robust software platform designed for programming and interacting with Arduino microcontrollers. Below, we delve into the software specifics and explore its diverse applications:

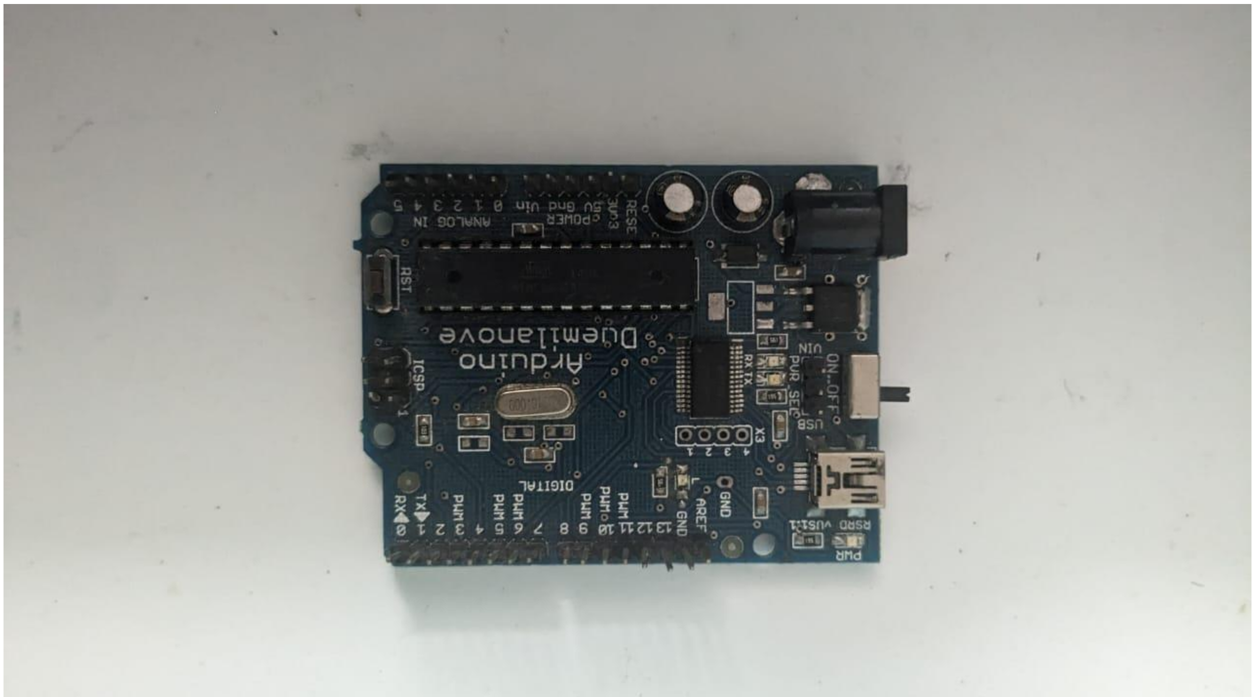


Fig1.1:Arduino Duemilanove Micro Controller

Arduino IDE:

Functionality: The Arduino IDE serves as the primary software interface for users to write, compile, and upload code (sketches) onto the Arduino Duemilanove.

Ease of Use: Known for its user-friendly nature, the IDE abstracts much of the complexity of embedded systems programming, making it accessible to beginners and experts alike.

Programming Language: Arduino IDE uses a simplified version of C/C++, facilitating a smoother learning curve.

Real world Applications :

- Education and Training: The Arduino Duemilanove, coupled with the Arduino IDE, is extensively employed in educational settings to teach programming, electronics, and robotics. Its simplicity encourages hands-on learning.

- Prototyping and Development: Engineers and hobbyists leverage the Arduino Duemilanove for prototyping projects, enabling rapid development and testing of electronic ideas before moving to production.
- Home Automation: The microcontroller is utilized in creating smart home systems, controlling devices such as lights, thermostats, and security systems.

Possible Applications:

- Autonomous Robots: Arduino Duemilanove can be integral in developing autonomous robots, finding applications in surveillance, exploration, or automated guided vehicles (AGVs).
- Environmental Monitoring: Employing sensors for temperature, humidity, or pollution, the Arduino Duemilanove can contribute to creating real-time environmental monitoring systems.
- Healthcare Devices: In healthcare, the microcontroller can be utilized to develop patient monitoring systems, wearable health devices, or assistive technologies.
- Smart Agriculture: The microcontroller is applicable in smart farming, contributing to projects involving automated irrigation systems, soil health monitoring, and crop monitoring devices.
- Smart Infrastructure: Arduino Duemilanove plays a role in smart city applications, including traffic management, waste management, and energy-efficient lighting systems.

Infrared (IR) Sensor in Autonomous Moving Bot Project:

The implementation of an Infrared (IR) sensor is a key component in the Autonomous Moving Bot project using Arduino Duemilanove. The IR sensor serves multiple purposes, contributing to both line following and obstacle avoidance functionalities. Below is an in-depth exploration of the IR sensor's role, its working principle, and its connections within the project:

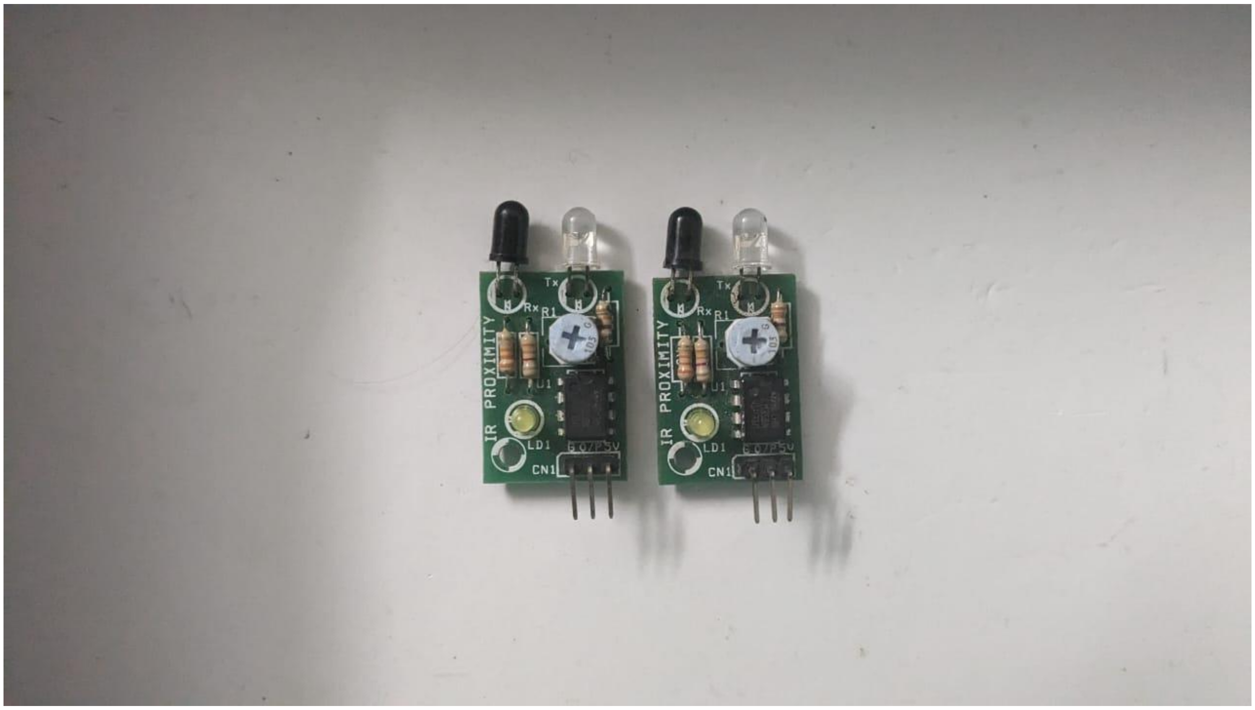


Fig1.2: IR Sensors

Working Principle:

- Detection of Infrared Radiation: IR sensors operate by detecting infrared radiation emitted or reflected by objects in their proximity. In the context of the project, these sensors help the bot perceive its environment.
- Emitter and Receiver Pair: Typically, an IR sensor comprises an emitter and a receiver. The emitter emits infrared light, and the receiver detects the reflected light. An object's presence alters the reflection, allowing the sensor to identify obstacles or follow a line.

Line Following Mechanism:

- Array of Sensors: In line following, an array of IR sensors is strategically positioned under the bot. These sensors continuously monitor the surface below, distinguishing between contrasting colors or reflective materials.

- Algorithmic Response: Based on the input from the sensors, the Arduino Duemilanove executes an algorithm to ensure the bot stays on the desired path. For example, if the left sensor detects the line, the bot adjusts its direction accordingly.

Obstacle Avoidance:

- Front-Facing Sensors: Additional IR sensors, positioned at the front of the bot, act as proximity detectors. They identify obstacles in the path of the bot.
- Decision-Making: When an obstacle is detected, the Arduino Duemilanove triggers a decision-making process. It may involve stopping the bot, adjusting its direction, or initiating a maneuver to avoid the obstacle.

Connections:

- Wiring Setup: The IR sensors are connected to specific pins on the Arduino Duemilanove. The emitter and receiver pairs require careful wiring to ensure proper functionality.
- Analog/Digital Signals: Depending on the type of IR sensor, analog or digital signals are sent to the Arduino, conveying information about the detected infrared radiation.

Advantages:

- Cost-Effective: IR sensors are relatively inexpensive, making them a cost-effective choice for projects with budget constraints.
- Robust Performance: They operate well in various lighting conditions, making them versatile for indoor and outdoor applications.
- Simplicity: IR sensors are easy to integrate, providing a straightforward solution for obstacle detection and line following.

Challenges:

- Limited Range: IR sensors may have a limited detection range, requiring careful placement to ensure comprehensive coverage.
- Interference: External sources of infrared radiation, such as sunlight, may introduce interference, impacting sensor accuracy.

<u>Gyroscope Sensor in Autonomous Moving Bot Project:</u>

The Gyroscope sensor plays a pivotal role in the Autonomous Moving Bot project, interfacing with the Arduino Duemilanove to provide precise directional control. Below is an elaborate explanation of the Gyroscope sensor's functionality, its working principle, and how it contributes to the autonomous movement of the bot:

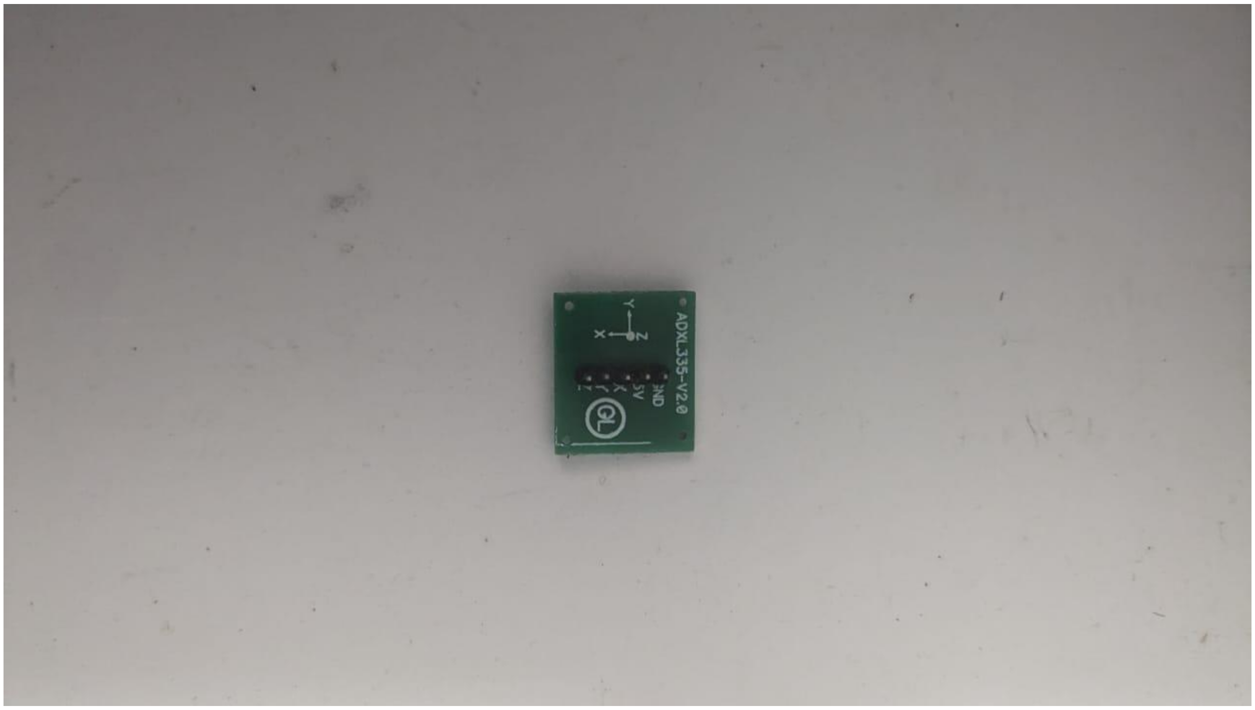


Fig1.3:Gyroscope Sensor

Working Principle:

- Measurement of Angular Velocity: A Gyroscope sensor measures the rate of rotation or angular velocity around a specific axis. In this project, the gyroscope detects changes in the bot's orientation, helping to maintain a straight path or adjust direction when needed.
- Three-Axis Sensing: Gyroscopes often feature three axes of sensing: X, Y, and Z. This triaxial capability allows the sensor to detect rotations in three-dimensional space.

Directional Control:

- Integration with Arduino Duemilanove: The Gyroscope sensor is connected to the Arduino Duemilanove microcontroller, providing real-time data on the bot's orientation.
- Algorithmic Response: Through programmed algorithms, the Arduino interprets the gyroscope data to determine if the bot is deviating from its intended direction. Subsequently, it triggers adjustments to maintain a consistent and accurate path.

Maneuvering Algorithm:

- Angular Rate Calculation: The gyroscope's data is processed to calculate the bot's angular rate, indicating how fast it is rotating.
- Directional Adjustment: When a deviation is detected, the Arduino executes a maneuvering algorithm. For example, if the bot drifts to the right, the algorithm may instruct the motors to correct the orientation and resume the desired course.

Connections:

- Wiring Setup: The Gyroscope sensor is connected to specific pins on the Arduino Duemilanove, ensuring a reliable flow of data.
- Analog/Digital Signals: Depending on the gyroscope model, it may transmit analog or digital signals to the Arduino, conveying information about angular velocity.

Advantages:

- High Precision: Gyroscopes provide accurate measurements of angular velocity, enabling precise control over the bot's direction.
- Real-Time Feedback: The real-time data from the gyroscope allows the Arduino to make instant adjustments, enhancing the bot's responsiveness.
- Versatility: Gyroscopes are versatile sensors suitable for various robotic applications, offering stability and control.

Challenges:

Integration Complexity: Proper integration with the Arduino Duemilanove may require calibration and careful consideration of the gyroscope's orientation on the bot.

Cost: Higher-precision gyroscope sensors may come at a higher cost, impacting the overall project budget.

<u>Power Supply in Autonomous Moving Bot Project:</u>

In the Autonomous Moving Bot project, a 9V battery is commonly used as the power supply. This section provides a detailed explanation of the power supply choice, its implications, and its role in ensuring the seamless operation of the project:

Fig1.4: 9v Battery

Power Requirement:

- Voltage: The choice of a 9V battery aligns with the voltage requirements of the components in the project, including the Arduino Duemilanove and other associated electronics. It provides a stable and standardized power source.

Components Powered:

- Arduino Duemilanove: The primary microcontroller, Arduino Duemilanove, typically operates within the voltage range provided by a 9V battery. It powers the logic circuits and control functions of the bot.
- Motors and Motor Driver: The regular 5V motors and associated motor driver circuits that propel the bot's movement are also powered by the 9V supply.

Connection to Arduino:

- Voltage Regulation: While the Arduino Duemilanove can handle 9V, it usually incorporates onboard voltage regulators to provide stable lower voltages for its internal components. This ensures that the microcontroller receives a consistent and regulated power supply.

Advantages of 9V Battery:

- Portable: 9V batteries are compact and portable, making them suitable for robotic projects where mobility is a crucial factor.
- Widespread Availability: 9V batteries are commonly available, making them convenient for hobbyist and educational purposes.

Challenges:

- Limited Capacity: While convenient, 9V batteries may have limited capacity compared to larger battery options. This could impact the runtime of the bot, especially if the project demands prolonged operation.

Power Management:

- Energy-Efficient Design: To maximize the battery life, the project may incorporate energy-efficient coding practices, and the bot's components might be designed to operate optimally within the constraints of a 9V supply.

<u>Motor Driver in the Autonomous Moving Bot Project:</u>

In the Autonomous Moving Bot project using Arduino Duemilanove, a motor driver plays a crucial role in controlling and powering the motors that drive the robot's movement. Let's delve into the details of the motor driver, its functions, and its connection to Arduino:

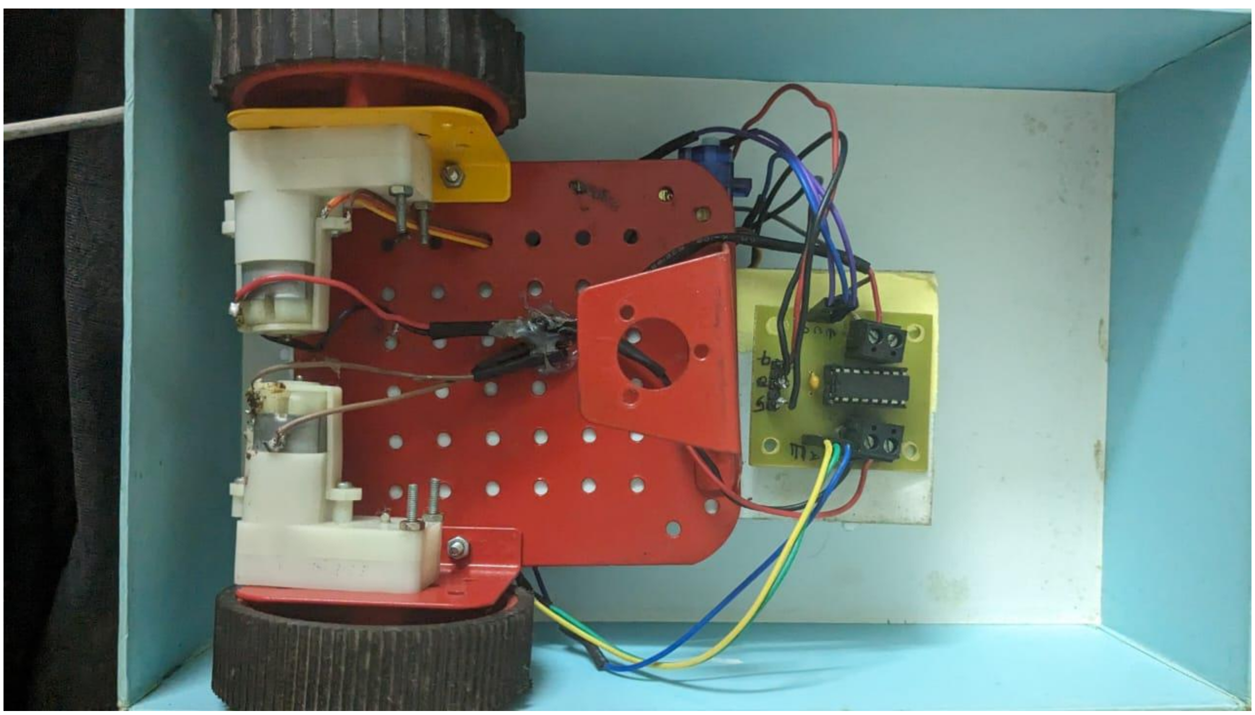


Fig1.5:Motors and Motor Driver

Motor Driver Overview:

- Purpose: The motor driver is an electronic component designed to regulate and control the speed and direction of motors. In this project, it ensures that the movements of the robot, powered by regular 5V motors, are precisely managed.

Motor Driver Functions:

- Direction Control: The motor driver facilitates the control of motor rotation direction. It allows the robot to move forward, backward, turn left, and turn right by managing the motor polarity.
- Speed Regulation: Some motor drivers also offer speed control features, allowing the user to adjust the speed of the motors for varied applications.

Motor Driver Connection to Arduino Duemilanove:

- Pins and Wiring: The motor driver is connected to specific pins on the Arduino Duemilanove to enable communication and control. It typically requires connections for motor power, ground, and control pins.
- Input Signals: Arduino outputs control signals to the motor driver, specifying the desired direction and speed of the motors. The motor driver interprets these signals and adjusts the motor behavior accordingly.

Usage of an H-Bridge:

- H-Bridge Configuration: Many motor drivers utilize an H-Bridge configuration. An H-Bridge is an arrangement of transistors that enables bidirectional control of the current flowing through the motors. It allows the motors to rotate forward or backward based on the input signals received from Arduino.

Advantages of Motor Drivers:

- Precise Control: Motor drivers provide precise control over motor movements, crucial for applications like robotics where accurate navigation is essential.
- Current Regulation: They often include features to regulate the current supplied to the motors, preventing damage due to excessive current flow.

Challenges:

- Complexity: Depending on the complexity of the motor driver used, understanding and configuring the driver settings may require careful attention.

Chapter-4

4.1 References Links

https://www.researchgate.net/publication/260249901_Implementation_Of_Autonomous_Line_Follower_Robot

https://www.researchgate.net/publication/351942942_Design_of_autonomous_line_follower_robot_with_obstacle_avoidance

https://commons.und.edu/cgi/viewcontent.cgi?article=1009&context=ee-stu chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://arxiv.org/pdf/1807.05511.pdf

https://www.researchgate.net/publication/330702209_Remote_and_Autonomous_Controlled_Robotic_Car_based_on_Arduino_with_Real_Time_Obstacle_Detection_and_Avoidance

https://www.researchgate.net/publication/354630395_Autonomous_Obstacle_Avoidance_Robot_using_IR_Sensors_Programmed_in_Arduino_UNO

https://www.academia.edu/21247219/Design_And_Analysis_Of_Carrying_Robot_Using_Arduino_Board_Control_System

https://www.diva-portal.org/smash/get/diva2:1462134/FULLTEXT01.pdf

https://portal.bazeuniversity.edu.ng/student/assets/thesis/2020100117124936373621.pdf

https://www.jetir.org/view?paper=JETIR2106835