A
Project Report
on

“Automated Wheelchair for Disabled people”

Submitted By

Sr No	PRN	Name
1	21410038	Mr. Shardul Shastri
2	21410039	Mr. Atharva Tambade
3	21410050	Mr. Vikas Gouda

Under the guidance of

Dr. B.G. Patil

Department of Electronics Engineering

WALCHAND COLLEGE OF ENGINEERING, SANGLI

(Government-Aided Autonomous Institute)

2023-2024

(This page is intentionally kept blank)

Contents

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The innovative "Automated Wheelchair for Individuals with Disabilities" initiative is dedicated to crafting a robotic solution controllable through hand gestures. Users will navigate the robot by employing specific hand motions, detected via mounted sensors. These gestures will prompt the robot to execute diverse movements and tasks. This pioneering project spans robotics, automation, and assistive tech realms, promising enhanced ease and intuitive interaction with robots. With its potential to streamline tasks, it stands poised to revolutionize our engagement with modern technology, opening new horizons in accessibility and automation for all.

ACKNOWLEDGEMENT

We would like to extend our heartfelt gratitude to Dr. B.G. Patil Sir, our esteemed Project Guide from the Department of Electronics Engineering, for generously dedicating his valuable time and unwavering assistance throughout our project journey. Our sincere appreciation also goes to the authors of various references and articles that served as pivotal resources for this endeavor.

CHAPTER 1

1. Introduction

The realm of robotics has seen significant progress recently, integrating robots into various applications. However, controlling these machines traditionally, like using a remote control or keyboard, can be complex and demands specialized training. Moreover, situations arise where these methods aren't safe or feasible. Using hand gestures for control offers distinct advantages over these norms. It permits users to control wheelchairs remotely without physical contact or direct visibility, proving valuable in scenarios where traditional control methods pose risks, such as hazardous environments or distant locations.

In essence, the Automated Wheelchair for Individuals with Disabilities project is an innovative and thrilling venture that provides a more intuitive and user-friendly way to manage robots. Its potential applications across different domains mark it as a significant contribution to the field of robotics and automation.

1.1. Background

In our rapidly evolving world, the significance of technology cannot be overstated. It's not just about utilizing technology but comprehending how to craft it. For an engineer, a robust understanding of other disciplines is imperative. Many projects often confine themselves within narrow disciplinary boundaries, constraining innovation and creativity. This wheelchair project, however, aims to transcend these limits, encouraging interdisciplinary connections. Rather than isolating topics, it seeks to intertwine mechanical, electronic, electrical, and programming skills, fostering a holistic approach to creation and design.

1.2. Motivation

Following are reasons due to which we find this project important:

- **Innovation:** Developing a hand gesture control robot is a cutting-edge project that involves the use of modern technologies and provides an opportunity to innovate in the field of robotics and automation.
- **Accessibility:** The project aims to create a more intuitive and user-friendly method of controlling robots that is accessible to a wider range of users without specialized training.
- **Skill Development:** Working on this project involves learning new skills and gaining valuable experience in coding, electronics, and sensor technology.
- **Personal Interest:** Developing a hand gesture control robot is a fulfilling and enjoyable project for Students like us who are interested in exploring new technologies.

1.3. Problem Description

Traditional methods of controlling robots can be complex and require specialized training, limiting the accessibility of robots to a wider range of users. Additionally, controlling a robot can be challenging in scenarios where traditional control methods are not feasible, such as hazardous environments or remote locations. Direct contact and line of sight are often required, limiting the accessibility and functionality of the robot. Furthermore, traditional control methods often require specialized training or knowledge, making it difficult for inexperienced users to operate the robot effectively. The hand gesture control robot project aims to address these challenges by developing a more intuitive and user-friendly method of controlling robots that is accessible to a wider range of users without the need for direct contact or line of sight, and without the need for specialized training.

1.4. Objectives

The objectives of the project are:

1. Develop a wheelchair controlled through hand gestures by integrating sensors, motors, and communication modules.
2. Create a wireless communication system for hand gesture recognition.
3. Execute a cost-effective project utilizing Arduino and Bluetooth technology with versatile applications.
4. Enhance accessibility for individuals with physical disabilities reliant on conventional input devices.

CHAPTER 2

2.1 Technology and Literature Survey

1. **"Hand Gesture Controlled Robot Using Arduino" by Suryarajsinh T. Vala**

Now-a-days, as a result of the advancements in technology, human-machine interaction is widely increasing, which reduces the gap between machines and humans for an easy standard of living. This paper describes regarding how the conventional hand gestures can control a robot and perform our desired tasks

2. **"Wireless Gesture Controlled Robot using Arduino and Bluetooth Module" by A. B.M. H. Rashid (2017)**

This paper describes a wireless gesture-controlled robot using Arduino and Bluetooth.The authors use an accelerometer and a gyroscope to detect hand movements, and transmit the data via Bluetooth to the robot for control.

3. **Gesture Control Robot by Namira Khan and Sharmin**

The foremost goal of the project work is to govern robotics with gestures and the use of hands.

The accelerometer relies upon the gestures of the hand. Through accelerometer, a passage of statistics sign is acquired and it's far processed with the help of Arduino microcontroller.

2.2 Block Diagram

Hand Gesture Control Wheelchair project is divided in to following blocks:

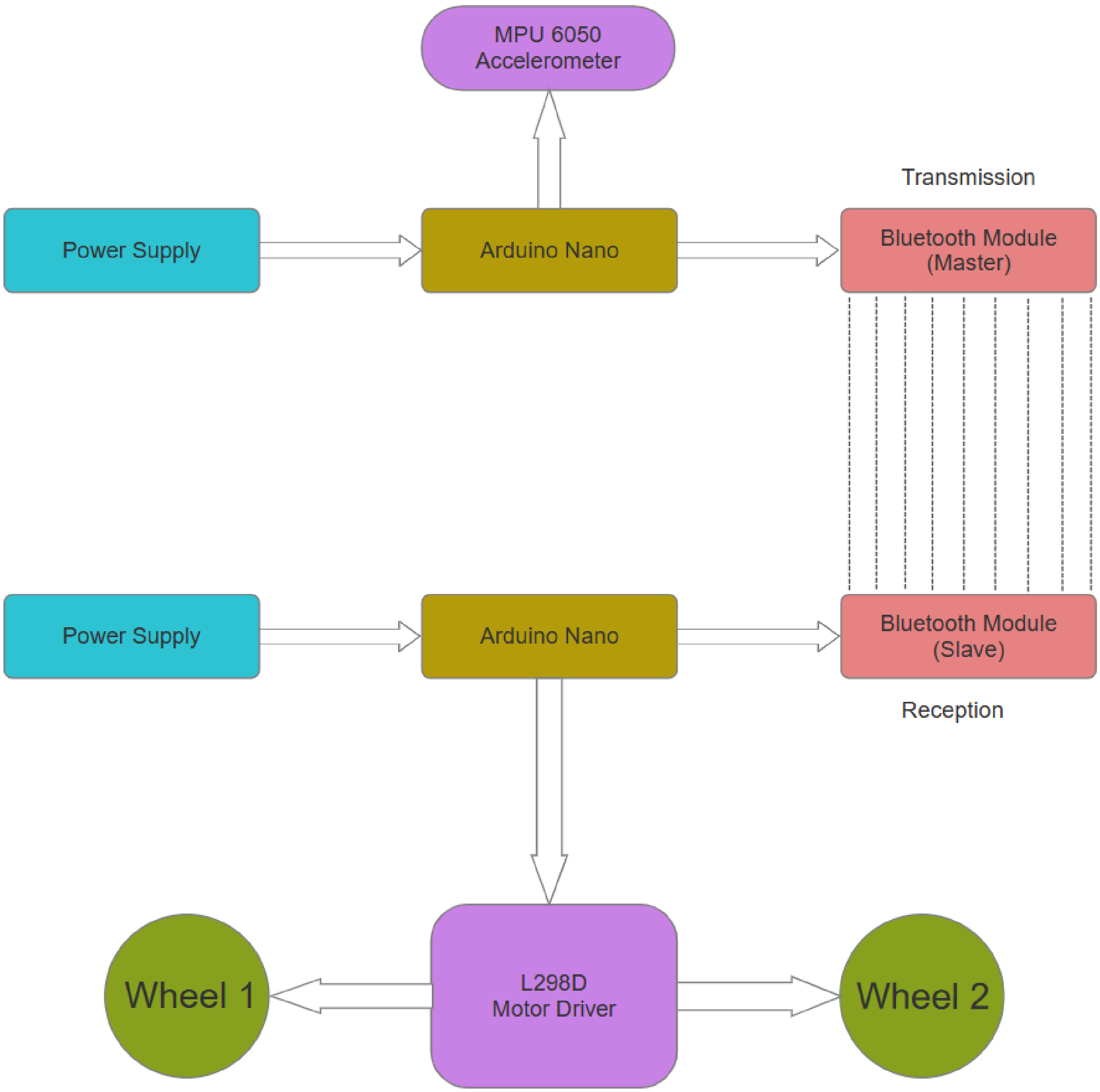

Figure 2.1: Block Diagram of Hand Gesture Control Wheelchair

2.3 Hardware Required

2.3.1. Arduino Nano

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board and IDE that runs on your computer, used to write and upload computer code to the physical board. The Arduino IDE uses a simplified version of C++, making it easier to learn to program.

- Processor: ATmega328 running at 16 MHz
- Flash Memory: 32 KB
- SRAM: 2 KB
- Operating Voltage: 5V
- Input Voltage: 7-12V
- Analog Inputs: 6
- Digital I/O Pins: 14 (including 6 PWM outputs)

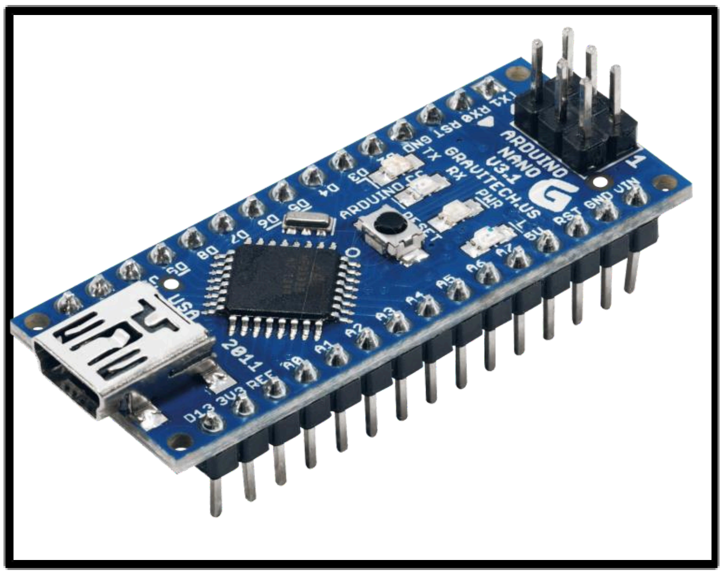

Figure 2.2: Arduino Nano

2.3.2 Bluetooth -Module

A Bluetooth module is a small electronic device that enables wireless communication between devices over short distances. Bluetooth technology is widely used in consumer electronics, such as smartphones, wireless headphones, and smart home devices. Bluetooth module is one of trending and fast technology used today. It acts as Wireless and low energy connection between two devices. The main principle of this is When Bluetooth- enabled devices are close to each other, they automatically detect each other. In this project we are using two Bluetooth modules in which one is master and other is slave. Master is used as transmitter and Slave as receiver. We can configure Bluetooth modules using AT commands.

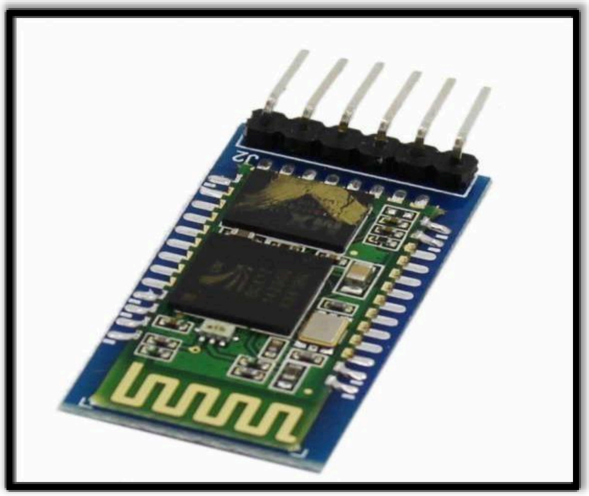

Figure 2.4 Bluetooth *Module*

2.3.3 MPU6050 Accelerometer

The MPU6050 is a commonly used accelerometer and gyroscope sensor module that can measure acceleration, tilt, and rotation in three dimensions. It is often used in robotics projects to enable motion sensing and control. To use the MPU6050 module with the Arduino, you can use a library like the "MPU6050" library, which provides functions for initializing the sensor, reading sensor data, and performing calibration. In our project MPU6050 accelerometer helps to measure the movements of a hand by detecting changes in acceleration in three axes (X, Y, and Z). When you move your hand, the accelerometer detects these changes in acceleration Produces an output signal that reflects the direction and magnitude of the movement.

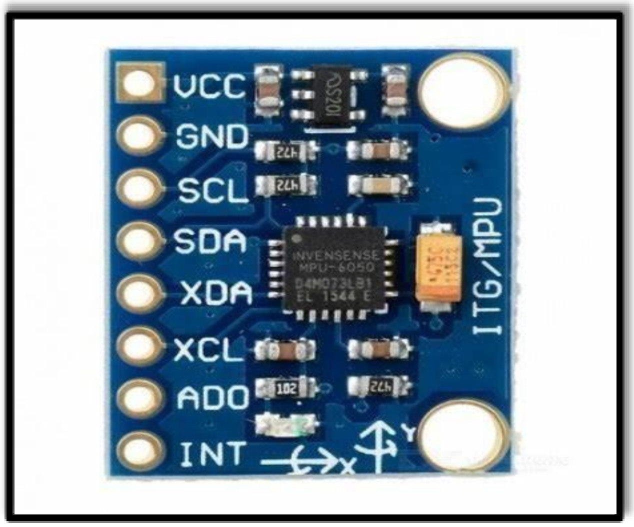

Figure 2.5 MPU6050 accelerometer

2.3.4 Motor Driver

A motor driver is an electronic circuit or device that controls the speed, direction, and torque of an electric motor. The use of a motor driver is essential in many applications where precise motor control is required. It is essential in many applications, including robotics, automation, and industrial control systems. In our project we are using a driver motor to accelerate the wheels of the robot.

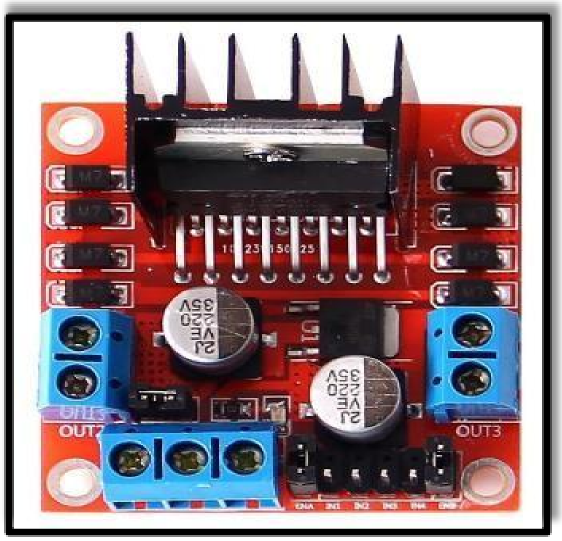

Figure 2.6 Motor Driver

2.3.5 Wheels and Chassis

Wheels and chassis are two important components of a vehicle that work together to provide stability, maneuverability, and support. Wheels are round structures that are typically made of metal or rubber and are attached to the vehicle's axles. The chassis is the framework of a vehicle that supports the body and engine. In our project for making the main body of the robot we are requiring 4 wheels and chassis.

Figure 2.6: Wheels and Chassis

2.3.6 DC Motor

A DC (direct current) motor is a type of electric motor that converts electrical energy into mechanical energy through the use of a magnetic field. It operates by applying a voltage to the motor's terminals, which creates a magnetic field that interacts with the motor's armature, causing it to rotate. DC motors are commonly used in robotics and automation applications because they can be easily controlled and provide high torque at low speeds. They are also relatively simple and inexpensive compared to other types of motors. To use a DC motor in a hand gesture control robot, we need a motor driver circuit that can provide the appropriate voltage and current to the motor.

Figure 2.7 DC Motor

2.3 Software Required

For coding and uploading the sketch, the Arduino IDE is used.

2.4 Flow Chart

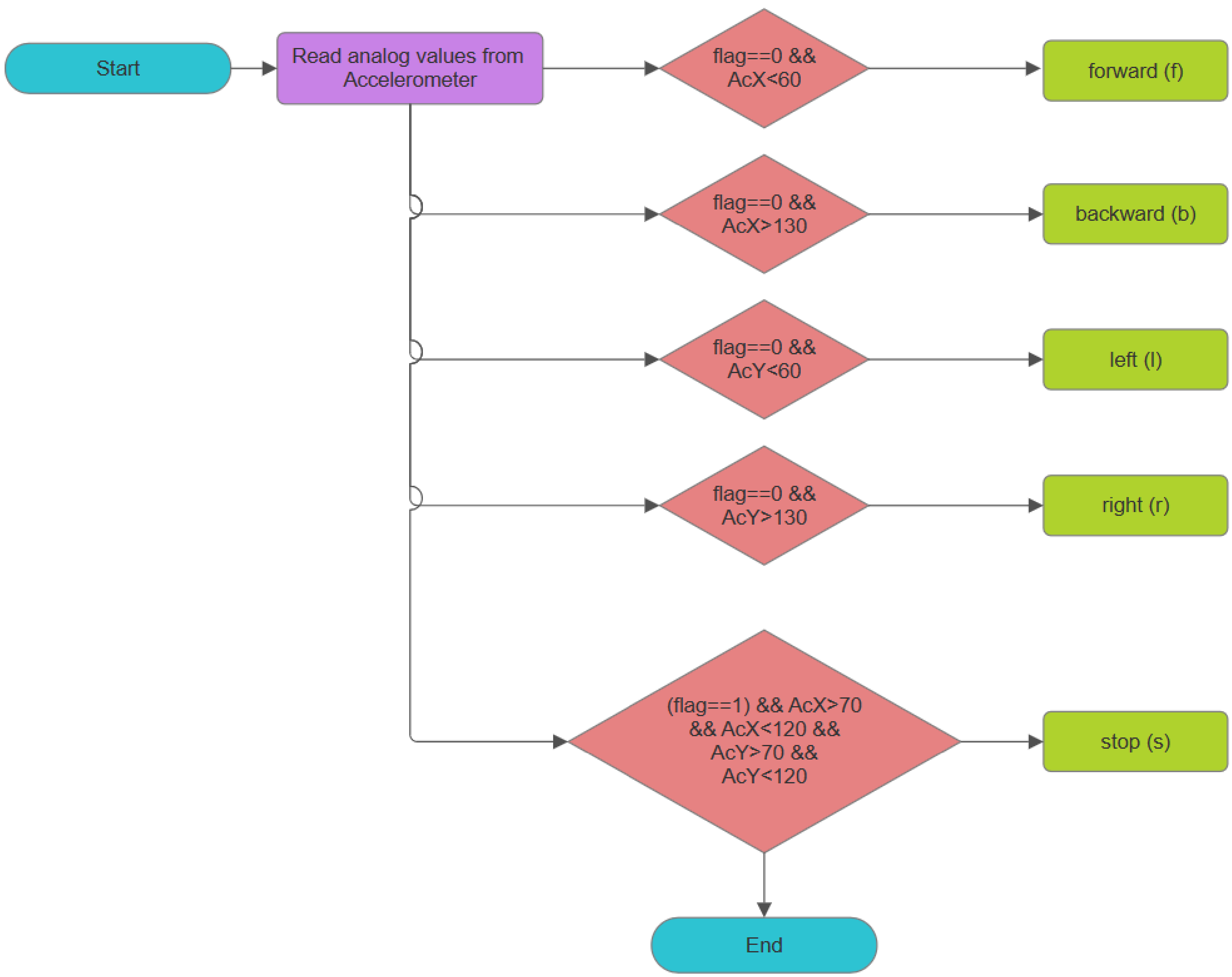

Fig. 3.1 Flow Chart

CHAPTER 3

3 Design and Implementation

3.1 Schematic

Required Components for schematic are:

- Arduino Uno.
- GSM Module
- LCD 16X2 Display
- DC Motor
- Relay
- Connectors to join the different boards to form one functional device. Each of the hardware is dissected and was designed/implemented separately for their functionality and later incorporated as one whole application. This helped in the debugging processes. We can prepare by using this.

3.2 Hardware

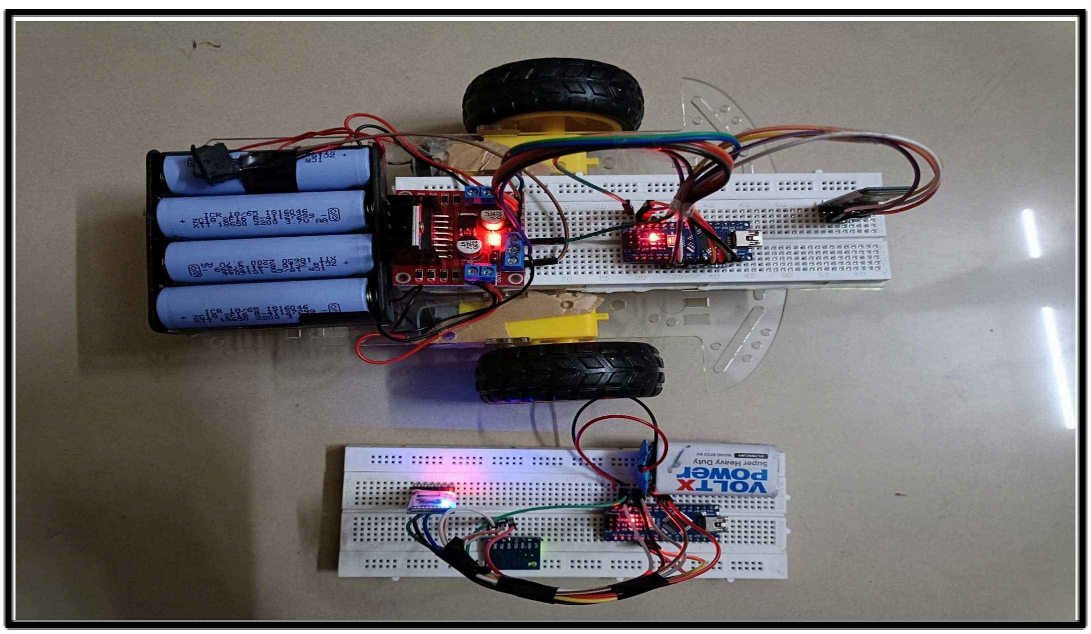

Fig. 3.2 Hardware Implementation

3.3 Working of Hand Gesture Control Wheelchair

1. **Transmitter Section:** The transmitter section includes a glove with an accelerometer sensor. The accelerometer sensor detects the hand gestures/motions made by the user wearing the glove.
2. **Receiver Section:** The receiver section includes an Arduino Uno and a Bluetooth module, which receives the signals transmitted from the glove. The Arduino Uno acts as the brain of the robot, receiving the signals from the accelerometer sensor and sending commands to the driver motor.
3. **Driver Motor:** The driver motor is connected to the wheels of the robot and is controlled by the Arduino Uno. It is responsible for accelerating the wheels and changing directions according to the gestures detected by the accelerometer sensor.
4. **Arduino Code:** Two Arduino boards will be used in the project, one for detecting the gestures and driving the robot and the other for giving signals of gestures. The code on the Arduino boards will be programmed to receive signals from the accelerometer sensor and send commands to the driver motor to control the movement of the robot.
5. **Operation:** The user wearing the glove will make hand gestures, which will be detected by the accelerometer sensor. The sensor will then send signals to the Arduino Uno, which will process the signals and send commands to the driver motor. The driver motor will then control the movement of the wheels of the robot according to the gestures made by the user.

In short, the hand gesture control wheelchair operates by detecting hand gestures/motions using an accelerometer sensor, transmitting the signals to an Arduino Uno via Bluetooth, and then using the Arduino Uno to control a driver motor that accelerates the wheels and changes directions accordingly.

CHAPTER 4

4.1. Applications

1. **In Surgery**: Hand gesture control in surgery can allow surgeons to manipulate medical tools and equipment without physical contact, this will lead to reduce the risk of contamination and increasing precision
2. **For handicapper people**: Hand gesture control technology can also help people with disabilities to operate machines or devices using simple hand movements. By using hand gesture robot, they can able to perform tasks easily which are difficult and which are not possible for them.
3. **Military**: In military applications, hand gesture control can be used to control robots and drones remotely without depending on traditional methods of input such as a joystick or keyboard.
4. **Gaming:** Hand gesture control in gaming allows Gamers to use natural hand movements to control characters or interact with the game environment.

4.2. Advantages

1. **Intuitive and natural:** Controlling a robot using hand gestures is an intuitive and natural way of interacting with technology, which can make it easier for users to learn and use the robot.
2. **Hands-free operation:** The hands-free operation of a hand gesture control robot can be useful in situations where the user's hands are occupied or where manual control is not feasible.
3. **No additional equipment required:** A hand gesture control robot does not require any additional equipment such as remote controls or joysticks, which can make it a more convenient and cost- effective option for users.
4. **Increased safety:** In some situations, such as working in hazardous environments, a hand gesture control robot can increase safety by allowing users to control the robot from a distance.

4.3. Disadvantages

1. **Limited range and accuracy:** The range and accuracy of hand gestures can be limited, which may make it difficult to control the robot precisely or from a distance.
2. **Limited range of motion:** Hand gestures can only control the robot within a limited range of motion. This can make it difficult to control the robot in certain situations or to perform complex tasks

4.4. Conclusion

Hand gesture control robots offer a promising and natural way of controlling robots without additional equipment. They can be useful in various applications, including healthcare, rehabilitation, manufacturing, and industrial automation. However, hand gesture control technology has limitations in terms of range, accuracy, complexity, functionality, fatigue, and susceptibility to interference. The success of a hand gesture control robot depends on many factors, including the sensors' accuracy and range, the control system's complexity, and the user's training and experience. Despite the challenges, hand gesture control technology has the potential to revolutionize the way we interact with robots and other devices, making them more efficient, productive, and safer.

COST ESTIMATION

Sr. No.	Name of Component	Quantity	Price Rs.
1	Arduino Nano	2	**289**
2	Driver motor	1	**115**
3	3.7V Li battery	2	**120**
4	Wheels	2	**90**
5	DC motors	2	**100**
6	MPU 6050	1	**250**
7	HC05 Bluetooth Module	2	**400**
Total Rs.			**1773/-**

Table 2: Cost of Project

APPENDICES

Program Code:

For Hand:

```
#include <SoftwareSerial.h>
SoftwareSerial BT_Serial(2, 3); // RX, TX

#include <Wire.h> // I2C communication library

const int MPU = 0x68; // I2C address of the MPU6050 accelerometer
int16_t AcX, AcY, AcZ;
int flag=0;

void setup()

{
	// put your setup code here, to run once

	Serial.begin(9600); // start serial communication at 9600
	bps BT_Serial.begin(9600);

	// Initialize interface to the MPU6050
	Wire.begin();
	Wire.beginTransmission(MPU);
	Wire.write(0x6B);
	Wire.write(0);
	Wire.endTransmission(true);

	delay(500);
}

void loop ()
{
	Read_accelerometer(); // Read MPU6050 accelerometer

	if(AcX<60 && flag==0){flag=1; BT_Serial.write('f');}
	if(AcX>130 && flag==0){flag=1; BT_Serial.write('b');}

	if(AcY<60 && flag==0){flag=1; BT_Serial.write('l'); }
	if(AcY>130 && flag==0){flag=1; BT_Serial.write('r');}

	if((AcX>70)&&(AcX<120)&&(AcY>70)&&(AcY<120)&&(flag==1))
	{
		flag=0;
		BT_Serial.write('s');
	}

	delay(100);
}
```

```
void Read_accelerometer()
{
      // Read the accelerometer data
       Wire.beginTransmission(MPU);
       Wire.write(0x3B); // Start with register 0x3B (ACCEL_XOUT_H)
       Wire.endTransmission(false);
       Wire.requestFrom(MPU, 6, true); // Read 6 registers total, each axis value
       is stored in 2 registers

       AcX = Wire.read() << 8 | Wire.read(); // X-axis value
       AcY = Wire.read() << 8 | Wire.read(); // Y-axis value
       AcZ = Wire.read() << 8 | Wire.read(); // Z-axis value

       AcX = map(AcX, -17000, 17000, 0, 180);
       AcY = map(AcY, -17000, 17000, 0, 180);
       AcZ = map(AcZ, -17000, 17000, 0, 180);

       Serial.print(AcX);
       Serial.print("\t");
       Serial.print(AcY);
       Serial.print("\t");
       Serial.println(AcZ);
}
```

For Robot:

```
#include <SoftwareSerial.h>

SoftwareSerial BT_Serial(2, 3); // RX, TX

#define enA 10//Enable1 L298 Pin enA
#define in1 9 //Motor1  L298 Pin in1
#define in2 8 //Motor1  L298 Pin in1
#define in3 7 //Motor2  L298 Pin in1
#define in4 6 //Motor2  L298 Pin in1
#define enB 5 //Enable2 L298 Pin enB

char bt_data; // variable to receive data from the serial port
int Speed = 150; //Write The Duty Cycle 0 to 255 Enable Pins for Motor Speed

void setup() { // put your setup code here, to run once

Serial.begin(9600); // start serial communication at 9600bps
BT_Serial.begin(9600);

pinMode(enA, OUTPUT); // declare as output for L298 Pin enA
pinMode(in1, OUTPUT); // declare as output for L298 Pin in1
pinMode(in2, OUTPUT); // declare as output for L298 Pin in2
pinMode(in3, OUTPUT); // declare as output for L298 Pin in3
pinMode(in4, OUTPUT); // declare as output for L298 Pin in4
pinMode(enB, OUTPUT); // declare as output for L298 Pin enB

delay(200);
}
```

```
void loop(){
if(BT_Serial.available() > 0){  //if some date is sent, reads it and saves in
state
bt_data = BT_Serial.read();
Serial.println(bt_data);
}
  //Serial.println("Received data");
if(bt_data == 'f'){forword();  Speed=180;}  // if the bt_data is 'f' the DC motor
will go forward
else if(bt_data == 'b'){backword(); Speed=180;}  // if the bt_data is 'b' the
motor will Reverse
else if(bt_data == 'l'){turnLeft(); Speed=150;}  // if the bt_data is 'l' the
motor will turn left
else if(bt_data == 'r'){turnRight();Speed=150;} // if the bt_data is 'r' the motor
will turn right
else if(bt_data == 's'){Stop(); }     // if the bt_data 's' the motor will Stop

analogWrite(enA, Speed); // Write The Duty Cycle 0 to 255 Enable Pin A for Motor1
Speed
analogWrite(enB, Speed); // Write The Duty Cycle 0 to 255 Enable Pin B for Motor2
Speed

delay(50);
}

void forword(){
   //forword
digitalWrite(in1, HIGH); //Right Motor forword Pin
digitalWrite(in2, LOW);  //Right Motor backword Pin
digitalWrite(in3, LOW);  //Left Motor backword Pin
digitalWrite(in4, HIGH); //Left Motor forword Pin
}

void backword(){ //backword
digitalWrite(in1, LOW);  //Right Motor forword Pin
digitalWrite(in2, HIGH); //Right Motor backword Pin
digitalWrite(in3, HIGH); //Left Motor backword Pin
digitalWrite(in4, LOW);  //Left Motor forword Pin
}

void turnRight(){ //turnRight
digitalWrite(in1, LOW);  //Right Motor forword Pin
digitalWrite(in2, LOW); //Right Motor backword Pin
digitalWrite(in3, LOW);  //Left Motor backword Pin
digitalWrite(in4, HIGH); //Left Motor forword Pin
}

void turnLeft(){ //turnLeft
digitalWrite(in1, HIGH); //Right Motor forword Pin
digitalWrite(in2, LOW);  //Right Motor backword Pin
digitalWrite(in3, LOW); //Left Motor backword Pin
digitalWrite(in4, LOW);  //Left Motor forword Pin
}

void Stop(){ //stop
digitalWrite(in1, LOW); //Right Motor forword Pin
```

```
digitalWrite(in2, LOW); //Right Motor backword Pin
digitalWrite(in3, LOW); //Left Motor backword Pin
digitalWrite(in4, LOW); //Left Motor forword Pin
```

References

1. "Hand Gesture Controlled Robot using Arduino and MPU6050 GyroscopeAccelerometer" by J.S. Kumar and P. Vinayakumar, International Journal of Engineering Research & Technology (IJERT), vol. 6, no. 2, pp. 148-152, 2017.
2. "Hand Gesture Recognition-Based Wireless Robot Control Using Arduino" by R. K.Sahoo, P. Kumar, and S. K. Mahapatra, International Journal of Control Theory and Applications, vol. 10, no. 43, pp. 277-284, 2017.
3. "Real-Time Hand Gesture Recognition for Controlling a Robot Using Arduino" by M. G. Nadeem and N. M. Khan, International Journal of Computer Applications, vol. 179, no. 11, pp. 31-36, 2018.