Bangladesh University of Engineering and Technology

Department of Electrical and Electronic Engineering

EEE 318 (January 2023)

Control Systems Laboratory

Final Project Report

Section: B1 Group: 02

Gesture Controlled Wheelchair for Disabled People

Course Instructors:

Shafin Bin Hamid, Lecturer

Mrinmoy Kundu, Part-Time Lecturer

Signature of Instructor: ______________________________

Academic Honesty Statement:

IMPORTANT! Please carefully read and sign the Academic Honesty Statement, below. Type the student ID and name, and put your signature. ***You will not receive credit for this project experiment unless this statement is signed in the presence of your lab instructor.***

"In signing this statement, We hereby certify that the work on this project is our own and that we have not copied the work of any other students (past or present), and cited all relevant sources while completing this project. We understand that if we fail to honor this agreement, We will each receive a score of ZERO for this project and be subject to failure of this course."

Signature: ________________________ **Full Name: Md. Hasnat Karim** **Student ID: 1906068**	**Signature:** ________________________ **Full Name: Navid Newaz** **Student ID: 1906073**
Signature: ________________________ **Full Name: Mehedi Hasan** **Student ID: 1906075**	**Signature:** ________________________ **Full Name: Asif Akhtab Ronggon** **Student ID: 1906078**
Signature: ________________________ **Full Name: Md. Faiyaz Abid** **Student ID: 1906079**	**Signature:** ________________________ **Full Name: Md. Shahin Ferdous** **Student ID: 1906088**

Table of Contents

1 Abstract

Our project gesture-controlled wheelchair aims to increase the independence and mobility of people with physical limitations. In this project we used microcontrollers, accelerometer and gyroscope for operating a wheelchair which will be driven by hand gesture.

People with physical disabilities struggle with traditional wheelchair control interfaces. The gesture-controlled wheelchair, however, provides a more practical and approachable solution. The system reads a user's hand and arm movements to convert them into wheelchair movement commands using a combination of gyroscope, accelerometer and Arduino Uno. The main goals of this project are to create a reliable and effective gesture detection system, to smoothly integrate it with wheelchairs, and to guarantee user safety throughout the control process. To improve the system's overall safety and usability, real-time gesture analysis, obstacle detection, and collision avoidance methods are used. The project's benefits go beyond better mobility; they also give people with physical limitations more autonomy and control over their daily lives. Additionally, because users can operate their wheelchairs more independently. So our project will be very helpful for the disable person for whom caregivers are not always available.

2 Introduction

Wheelchairs are necessary tools that give people with physical impairments or injuries independence and mobility. Some wheelchairs, however, need manual operation or outside help, which can be challenging or problematic for some users. Therefore, there is a need to create alternate, more adaptive, and user-friendly wheelchair control techniques.

One such technique is gesture control, which enables users to regulate the motion of the wheelchair with their hands or bodies. Compared to manual or voice control, gesture control is more natural, understandable, and dependable. Additionally, gesture control can make use of a variety of sensors and technologies, including cameras, infrared, gyroscopes, accelerometers, and webcams.

In this project, we created and put into use a gesture-controlled wheelchair for people with disabilities. The sensor is fastened to the user's hand and measures the hand's three-dimensional acceleration and orientation. The wheelchair can travel forward, backward, left, right, or stop depending on the sensor data. The major goal of this project is to offer disabled individuals a low-cost, high-quality wheelchair control system that will enhance their quality of life.

3 Design

3.1 Problem Formulation

People with physical disabilities like spinal cord injury, muscular injuries or poor nervous system face significant challenge for independent movement. These disabilities limits their free movement. So they have to rely on their caregiver. This may interrupt their privacy. Moreover caregivers are not always there to help them. Some people also depend on traditional wheelchair and joystick for movement. But this also has some limitations because people who are paralyzed or has a problem in upper body parts. This will put a physical pressure on them. So we want to design and implement a system which will control any wheelchair by hand gesture. Because traditional control system of wheelchair or joystick cannot ensure the proper safety for users. It is very important to design such a system which will ease their movement and quality of their life. As a result they might not avoid the collision with obstacle. So in our designed system proper control and safety issue will get the most priority. We will design a wheelchair control system which will move according to the hand gesture command of user and will stop when there is any obstacle in the direction of its movement. When the wheelchair turns about 90^0 from its current position the wheelchair will automatically stop.

3.1.1 Identification of Scope

The fundamental scope is to develop and integrate the modern gesture control technology to control it in a systematic way. In our project we used microcontroller, gyroscope, motor, motor driver, accelerometer, RF transmitter and receiver etc. Our project is user friendly and it can be customized for the people with variable degree of physical needs. We included feedback mechanism in our system which makes it easier for user. Using feedback also ensure a better control and better quality of system. As our system is not built up with very complex technology we hope it will be affordable for the mass user. Some development in hardware integration with software part we used can make it more reliable. However our safety management in this project is satisfactory. We added some features on obstacle detection, avoiding collisions and stopping the wheelchair when necessary. Our wheelchair system is very easy to use. User do not need any manuals or training to operate it safely. Only hand gesture is enough to move or stop it. Our system is responsive in a moderate manner. When a gesture command is given to the system, it responses quickly. It increases the degree of safety to the users. It would be better if we managed to design the system in such a manner that it would stop or move when there occurs any emergency.

3.1.2 Literature Review

The fundamental scope is to develop and integrate the modern gesture control technology to control it in a systematic way. In our project we used microcontroller, gyroscope, motor, motor driver, accelerometer, RF transmitter and receiver etc. Our project is user friendly and it can be customized for the people with variable degree of physical needs. We included feedback mechanism in our system which makes it easier for user. Using feedback also ensure a better control and better quality of system. As our system is not built up with very complex technology we hope it will be affordable for the mass user. Some development in hardware integration with software part we used can make it more reliable. However our safety management in this project is satisfactory. We added some features on obstacle detection, avoiding collisons and stopping the wheelchair when necessary. Our wheelchair system is very easy to use. User do not need any manuals or training to operate it safely. Only hand gesture is enough to move or stop it. Our system is responsive in a moderate manner. When a gesture command is given to the system, it responses quickly. It increases the degree of safety to the users. It would be better if we managed to design the system in such a manner that it would stop or move when there occurs any emergency.

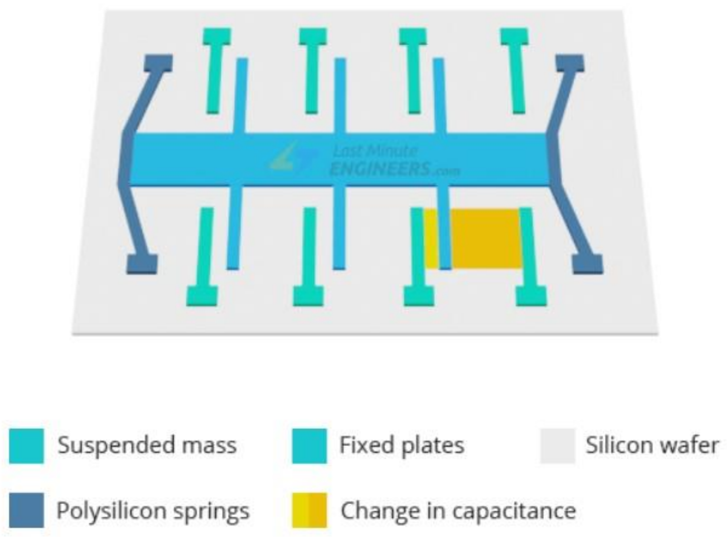
3.1.3 Formulation of Problem

In first place we need to detect the gesture command given by the user. Then we need to store the data and send it to the system so it can move according to the command. This command will drive the wheelchair. We added a feature about obstacle detectors. So we will need another sensor to detect whether there is a obstacle or not in the direction of movement of wheelchair. If there is any obstacle with whom the wheelchair may collide, then it will send a command to the system to stop immediately. So we need a feedback for this feature. We also added another feature that the wheelchair will stop moving if it rotates 90^0 from its current position. So we will need another feedback to detect whether its rotating angle is right angle or not. Because if we did not add this feature then the wheelchair would continue rotating when the command is to move left or right. That is, without using this feature we could not have a proper control on it and the motor would not stop rotating.

3.1.4 Analysis

As we need to detect the command first we will need an accelerometer and Arduino. The accelerometer has x,y,z value depending on its movement. It will detect the change of its x,y,z value and by calculating these changes we will write an Arduino code for the movement of wheelchair, whether the command is left or right, forward or backward. These data will be sent from the RF transmitter module and data will be received in RF receiver module. Using the receiver module data the second microcontroller will drive the the motor driver which in turn will drive the motor. The gyroscope used here will calculate the angle of rotation. Calculating the z angle it will stop rotation if the rotation angle is 90^0.

Working Principle of Accelerometer: An accelerometer's main mode of operation is the transformation of mechanical energy into electrical energy. Piezoelectric materials are used for this purpose. When a mass is applied to the sensor, which functions exactly like a spring, it begins to descend. Since it is traveling downward, acceleration begins to occur. This acceleration is then transformed into an electric signal amount that is used to measure changes in the device's position. However in this project we used xyz type accelerometer. The XYZ-type accelerometer uses the gravitational force to compare the position of the Devices. It consists of three separate accelerometers that measure acceleration in the X, Y and Z planes.



.

Fig: Working principle of accelerometer

When acceleration is introduced in the system, the fixed parallel plates comes closer and far thus the value of accelerometer changes its reading according to the changing capacitance value.

Principle of Gyroscope: A gyroscope consists of four parts: Spin axis, Gimbal, Rotor, Gyroscope frame.

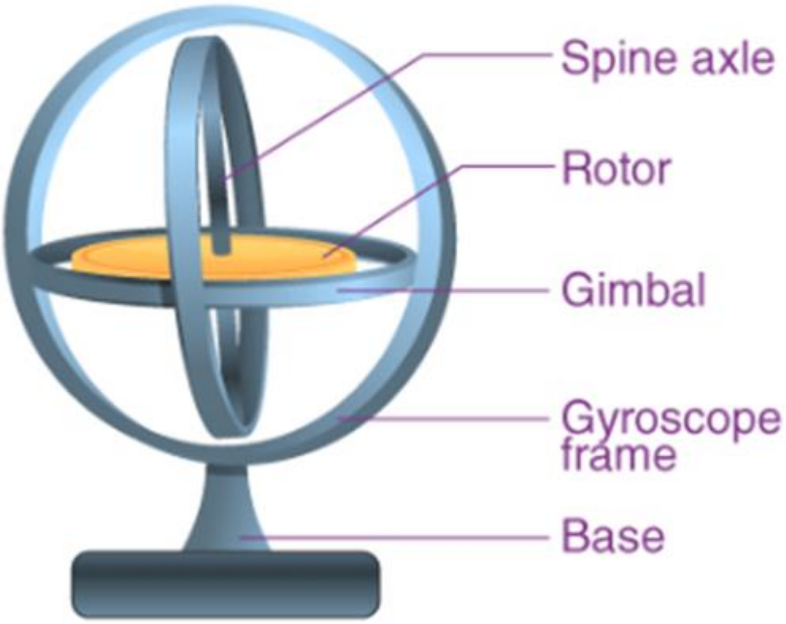


Fig: Parts of Gyroscope

A gyroscope is a large rotor that is fixed to supporting rings called gimbals. Through the use of frictionless bearings found in the gimbals, the core rotor is shielded from external torques. The axle of the rotating wheel establishes the spin axis. Due to its ability to keep the high-speed rotation axis near the center of the rotor, the rotor exhibits exceptional stability at high speeds. There are three rotational degrees of freedom for the rotor.

A gyroscope's operation is dependent on the law of gravity. A gyroscopic precession in the spinning wheel is explained as the result of angular momentum, which is experienced by the torque on a disc. Gyroscopic motion, also known as gyroscopic force, is the tendency of a rotating object to maintain its orientation. We are aware that the rotating item has angular momentum, and this momentum must be preserved. This is carried out because any change in the rotation's axis will result in an alteration of the orientation, which alters the angular momentum. As a result, it is clear that the gyroscope's operating principle is based on the conservation of angular momentum. It has four proof mass M1, M2, M3 and M4.

It has three mode of operation: roll mode, yaw and pitch.

Roll mode: When an angular displacement is introduced in the X-axis, M1 and M3 will move up and down out of the plane due to the Coriolis effect. This causes a change in the roll angle, hence the name Roll Mode.

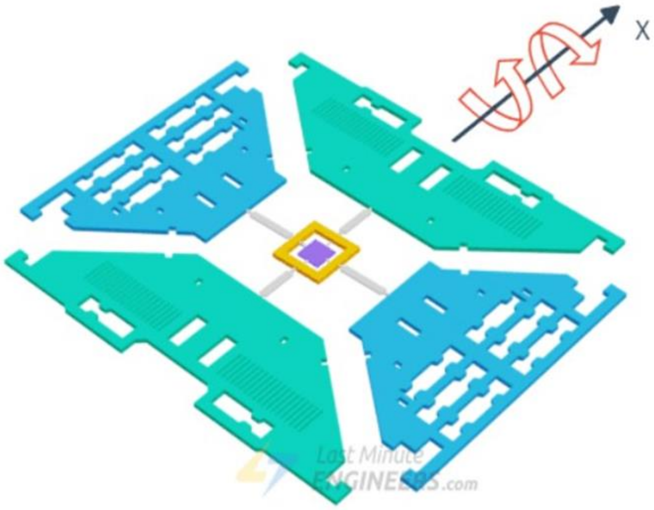


Fig: Roll Mode

Pitch mode: When an angular displacement is introduced in the Y-axis, M2 and M4 will move up and down out of the plane due to the Coriolis effect. This causes a change in the pitch angle, hence the name Pitch Mode.

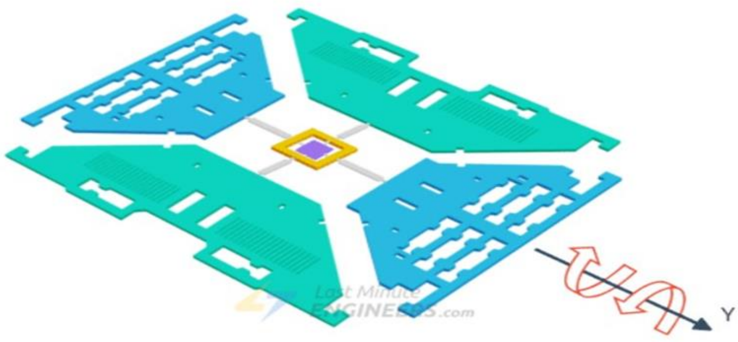


Fig: Pitch Mode

Yaw mode: When an angular displacement is introduced in the Z-axis, M2 and M4 will move horizontally due to the Coriolis effect. This causes a change in the yaw angle, hence the name

Yaw Mode.

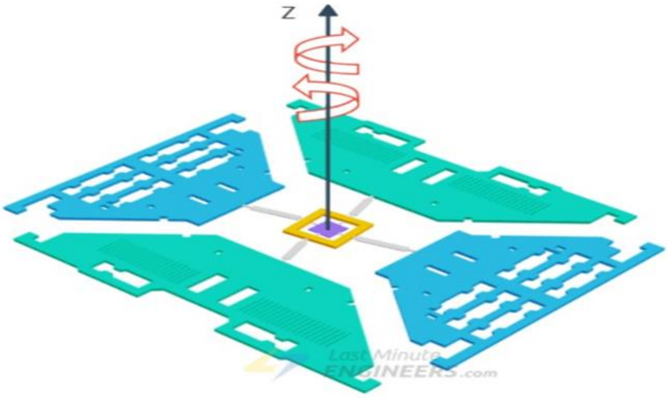


Fig: Yaw Mode

We used Yaw mode in this project.

Calibration for gyroscope:

```
#include<Wire.h>


float z_angle=0;
 float RateRoll,RatePitch, RateYaw;
 float RateCallibrationRoll,RateCallibrationPitch,RateCallibrationYaw;
 int RateCallibrationNumber;
 void gyro_signals(void){
       //start I2C Communication with gyro
       Wire.beginTransmission(0x68);
       // switch on the low pass filter
       Wire.write(0x1A);
       Wire.write(0x05);
       Wire.endTransmission();


      //set the sensitivity scale factor
       Wire.beginTransmission(0x68);
       Wire.write(0x1B);
       Wire.write(0x8);
       Wire.endTransmission();
       // Access registors storing gyro measurements
       Wire.beginTransmission(0x68);
       Wire.write(0x43);
       Wire.endTransmission();
       Wire.requestFrom(0x68,6);
       int16_t GyroX=Wire.read()<<8 | Wire.read();
       int16_t GyroY=Wire.read()<<8 | Wire.read();
       int16_t GyroZ=Wire.read()<<8 | Wire.read();


      RateRoll=(float)GyroX/65.5;//data sheet a lsb sensitivoty diye vag
```

```
        RatePitch=(float)GyroY/65.5;
        RateYaw=(float)GyroZ/65.5;


}


void setup()
 {
   Serial.begin(9600);
   Wire.begin();
   delay(250);
   Wire.beginTransmission(0x68);
   Wire.write(0x6B);
   Wire.write(0x00);
   Wire.endTransmission();
   for(RateCallibrationNumber=0;RateCallibrationNumber<2000;RateCallibrationNumber++){
          gyro_signals();
          RateCallibrationRoll+=RateRoll;
          RateCallibrationPitch+=RatePitch;
          RateCallibrationYaw+=RateYaw;
          delay(1);
   }
   RateCallibrationRoll/=2000;
   RateCallibrationPitch/=2000;
   RateCallibrationYaw/=2000;
 }


void loop()
 {
   gyro_signals();


   RateRoll-=RateCallibrationRoll;
   RatePitch-=RateCallibrationPitch;
   RateYaw-=RateCallibrationYaw;
   Serial.print("Roll rate=");
 // Serial.print(RateRoll);


   Serial.print("Roll rate=");
   //Serial.print(RateRoll);


   Serial.print("Pitch rate=");
 // Serial.print(RatePitch);


   Serial.print("Yaw rate=");
   //Serial.print(RateYaw);


   static unsigned long prevTime = 0;
   unsigned long currentTime = millis();
   float dt = (float)(currentTime - prevTime) / 1000.0;
   prevTime = currentTime;
   z_angle = (z_angle + RateYaw*dt);
   Serial.print("Z_angle");
   Serial.print(z_angle);
```

```
  Serial.println(" deg");



  if(abs(z_angle) >= 90)
  {
    Serial.println("Yess!");
    z_angle = 0;
  }
}
```

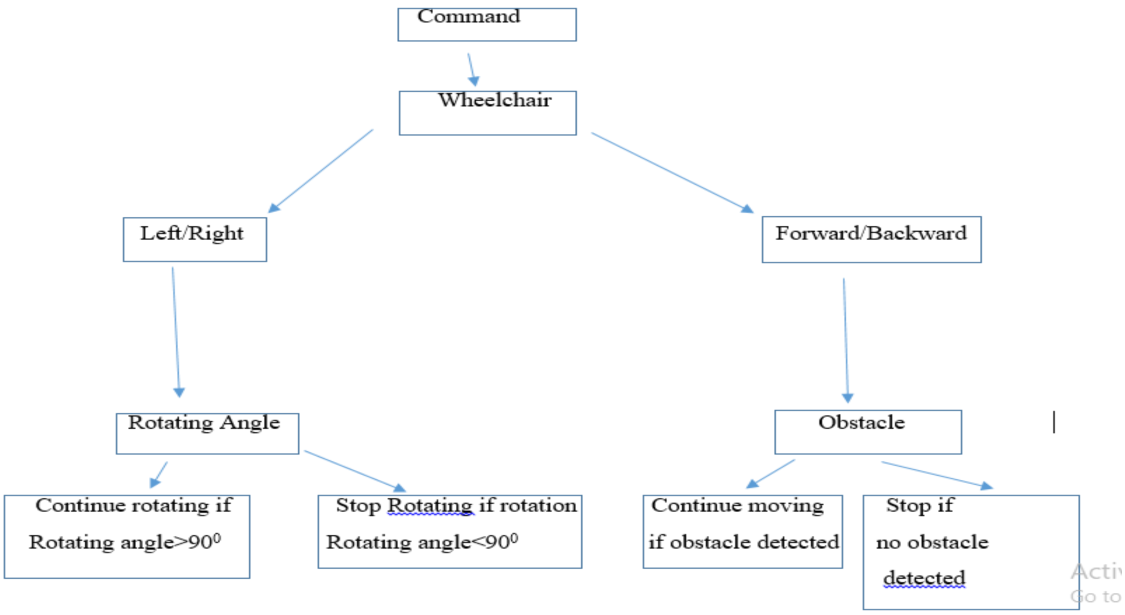
We need to callibrate the gyro sensor due to some offset value in the sensor.

Motor Driving Conditions:

	M1	M2	M3	M4
Right	High	Low	Low	Low
Left	Low	Low	High	Low
Forward	High	Low	High	Low
Backward	Low	High	Low	High
Stop	Low	Low	Low	Low

3.2 Design Method

We first estimated an approximate solution to this method. Our problem algorithm can be shown in the flow chart below:



3.3 **Circuit Diagram**

Here is our Circuit diagram for transmitter and reciever module

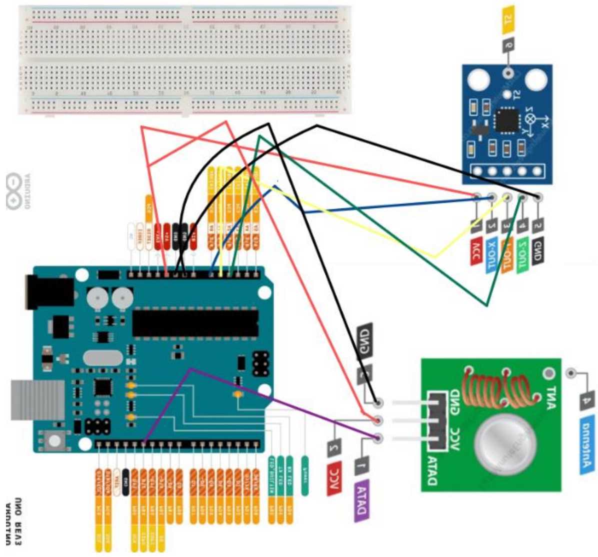


Fig: Circuit Diagram for Transmitter

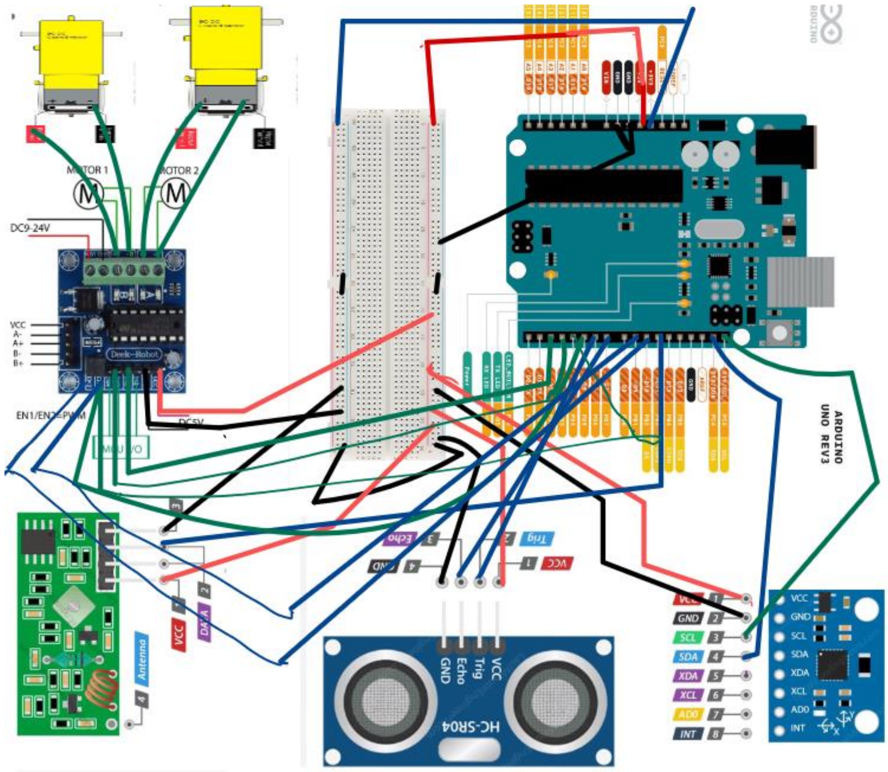


Fig: Circuit Diagram for reciever module

3.4Full Source Code of Firmware:

Table: Source Code for the main program

```
Transmitter Code:
#include <VirtualWire.h>
 #define x A0
 #define y A1
 #define z A2

char *data;

int x_val;
 int y_val;
 int z_val;

int x_val2;
 int y_val2;
 int z_val2;

void setup()
 {
   vw_set_tx_pin(12);
   vw_setup(2000);
   pinMode(x, INPUT);
   pinMode(y, INPUT);
   pinMode(z, INPUT);
   Serial.begin(9600);
   x_val2 = analogRead(x);
   y_val2 = analogRead(y);
   z_val2 = analogRead(z);
 }

void loop()
 {
   x_val = analogRead(x);
   y_val = analogRead(y);
   z_val = analogRead(z);

  int x_axis = x_val - x_val2;
   int y_axis = y_val - y_val2;
```

```
Receiver Code:

#include <VirtualWire.h>
#include<Wire.h>
#define m1 2
#define m2 3
#define m3 4
#define m4 5
const int trigger=6;
int echo=7;
float distance;
float dist_inches;
int EN1 = 9;
int EN2 = 10;
float z_angle=0;
float RateRoll,RatePitch, RateYaw;
float
RateCallibrationRoll,RateCallibrationPitch,RateCallibra
tionYaw;
int RateCallibrationNumber;


void gyro_signals(void){
      //start I2C Communication with gyro
      Wire.beginTransmission(0x68);
      // switch on the low pass filter
      Wire.write(0x1A);
      Wire.write(0x05);
      Wire.endTransmission();


      //set the sensitivity scale factor
      Wire.beginTransmission(0x68);
      Wire.write(0x1B);
      Wire.write(0x8);
      Wire.endTransmission();
      // Access registors storing gyro measurements
```

```
    int z_axis = z_val - z_val2;

    if(y_axis >= 30)
    {
      data="f";
      vw_send((uint8_t *)data, strlen(data));
      vw_wait_tx();
      delay(500);
      Serial.println("Backward");
    }
    else if(y_axis <= -30)
    {
      data="b";
      vw_send((uint8_t *)data, strlen(data));
      vw_wait_tx();
      delay(500);
      Serial.println("Forward");
    }
    else if(x_axis >= 30)
    {
      data="r";
      vw_send((uint8_t *)data, strlen(data));
      vw_wait_tx();
      delay(500);
      Serial.println("Right");
    }
    else if(x_axis <= -30)
    {
      data="l";
      vw_send((uint8_t *)data, strlen(data));
      vw_wait_tx();
      delay(500);
      Serial.println("Left");
    }
    else
    {
      data="s";
      vw_send((uint8_t *)data, strlen(data));
      vw_wait_tx();
      delay(500);
      Serial.println("Stop");
    }
  }
```

```
      Wire.beginTransmission(0x68);
      Wire.write(0x43);
      Wire.endTransmission();
      Wire.requestFrom(0x68,6);
      int16_t GyroX=Wire.read()<<8 | Wire.read();
      int16_t GyroY=Wire.read()<<8 | Wire.read();
      int16_t GyroZ=Wire.read()<<8 | Wire.read();


         RateRoll=(float)GyroX/65.5;//data  sheet  a  lsb
sensitivoty diye vag
      RatePitch=(float)GyroY/65.5;// raw data to convert
degree/second
      RateYaw=(float)GyroZ/65.5;


}



void setup()
{
    vw_set_rx_pin(11);
    vw_setup(2000);
    pinMode(m1, OUTPUT);
    pinMode(m2, OUTPUT);
    pinMode(m3, OUTPUT);
    pinMode(m4, OUTPUT);
    pinMode(EN1,OUTPUT);
    pinMode(EN2,OUTPUT);
    pinMode(trigger,OUTPUT);
    pinMode(echo, INPUT);

    vw_rx_start();
    Serial.begin(9600);
    Wire.begin();
    delay(250);
    Wire.beginTransmission(0x68);
    Wire.write(0x6B);
    Wire.write(0x00);
    Wire.endTransmission();
     for(RateCallibrationNumber=0;RateCallibrationNumber
<2000;RateCallibrationNumber++){
         gyro_signals();
         RateCallibrationRoll+=RateRoll;
         RateCallibrationPitch+=RatePitch;
         RateCallibrationYaw+=RateYaw;
         delay(1);
   }
   RateCallibrationRoll/=2000;
   RateCallibrationPitch/=2000;
   RateCallibrationYaw/=2000;
    //digitalWrite(EN1,HIGH);
    //digitalWrite(EN2,HIGH);
}


void loop()
{
    setspeed();
    uint8_t buf[VW_MAX_MESSAGE_LEN];
    uint8_t buflen = VW_MAX_MESSAGE_LEN;


    if (vw_get_message(buf, &buflen))
    {
      if(buf[0]=='f')
      {
        digitalWrite(trigger,LOW);
        delayMicroseconds(5);

    //Start Measurement
        digitalWrite(trigger,HIGH);
        delayMicroseconds(10);
        digitalWrite(trigger,LOW);

    // Acquire and convert to inches
         distance=pulseIn(echo,HIGH);
        distance=distance*0.0001657;
         dist_inches=distance*39.37;


    // print the value on the serial monitor
        Serial.print(dist_inches);
        Serial.println(" inches");
```

	<pre> if(dist_inches>=4)
	       {
	        forward();
	        Serial.println("Forward");
	       }
	      else if(dist_inches<4)
	      {
	        stop();
	        Serial.println("Rong");
	        delay(5000);
	      }
	
	    }
	    else if(buf[0]=='b')
	    {
	
	     digitalWrite(trigger,LOW);
	       delayMicroseconds(5);
	
	  //Start Measurement
	       digitalWrite(trigger,HIGH);
	       delayMicroseconds(10);
	       digitalWrite(trigger,LOW);
	
	  // Acquire and convert to inches
	       distance=pulseIn(echo,HIGH);
	       distance=distance*0.0001657;
	       dist_inches=distance*39.37;
	
	  // print the value on the serial monitor
	       Serial.print(dist_inches);
	       Serial.println(" inches");
	
	    if(dist_inches>=4){
	     backward();
	     Serial.println("Backward");
	    }
	    else if (dist_inches<4) {
	       stop();
	       Serial.println("Stop");
	    }
	
	    }
	    else if(buf[0]=='r')
	    {
	
	       digitalWrite(trigger,LOW);
	       delayMicroseconds(5);
	
	  //Start Measurement
	       digitalWrite(trigger,HIGH);
	       delayMicroseconds(10);
	       digitalWrite(trigger,LOW);
	
	  // Acquire and convert to inches
	       distance=pulseIn(echo,HIGH);
	       distance=distance*0.0001657;
	       dist_inches=distance*39.37;
	
	  // print the value on the serial monitor
	       //Serial.print(dist_inches);
	       //Serial.println(" inches");
	
	       gyro_signals();
	
	       RateRoll-=RateCallibrationRoll;
	       RatePitch-=RateCallibrationPitch;
	       RateYaw-=RateCallibrationYaw;
	       static unsigned long prevTime = 0;
	       unsigned long currentTime = millis();
	         float dt = (float)(currentTime - prevTime) /
	1000.0;
	       prevTime = currentTime;
	       z_angle = (z_angle + RateYaw*dt);
	       Serial.print("Z_angle");
	       Serial.print(z_angle);
	       Serial.println(" deg");
	
	       if(abs(z_angle)<=80 & dist_inches>4)
	       {
	
	        right();
	        Serial.println("Right");</pre>

	<pre> }
	        else if (abs(z_angle)<=80 & dist_inches<=4) {
	               stop();
	               Serial.println("Rong");
	               delay(5000);
	        }
	        else if(abs(z_angle)>80) {
	
	         stop();
	         z_angle=0;
	         Serial.println("Rong");
	         delay(5000);
	        }
	     }
	
	     else if(buf[0]=='l')
	     {
	        digitalWrite(trigger,LOW);
	        delayMicroseconds(5);
	
	   //Start Measurement
	        digitalWrite(trigger,HIGH);
	        delayMicroseconds(10);
	        digitalWrite(trigger,LOW);
	
	   // Acquire and convert to inches
	        distance=pulseIn(echo,HIGH);
	        distance=distance*0.0001657;
	        dist_inches=distance*39.37;
	
	   // print the value on the serial monitor
	        Serial.print(dist_inches);
	        Serial.println(" inches");
	        gyro_signals();
	
	        RateRoll-=RateCallibrationRoll;
	        RatePitch-=RateCallibrationPitch;
	        RateYaw-=RateCallibrationYaw;
	        static unsigned long prevTime = 0;
	        unsigned long currentTime = millis();
	          float dt = (float)(currentTime - prevTime) /
	1000.0;
	        prevTime = currentTime;
	        z_angle = (z_angle + RateYaw*dt);
	        Serial.print("Z_angle");
	        Serial.print(z_angle);
	        Serial.println(" deg");
	
	        if(abs(z_angle)<=80 && dist_inches>4)
	        {
	
	         left();
	         Serial.println("left");
	        }
	        else if (abs(z_angle)<=80 && dist_inches<=4) {
	           stop();
	           Serial.println("Rong");
	           delay(5000);
	        }
	        else if(abs(z_angle)>80) {
	
	         stop();
	         z_angle=0;
	         Serial.println("Rong");
	         delay(5000);
	        }
	     }
	
	     else if(buf[0]=='s')
	     {
	
	      stop();
	      Serial.print("Z_angle");
	      Serial.print(z_angle);
	      Serial.println("Stop");
	
	     }
	   }
	}</pre>

```
void forward()
{

        digitalWrite(m1,HIGH);
        digitalWrite(m2,LOW);
        digitalWrite(m3,HIGH);
        digitalWrite(m4,LOW);
        setspeed();
        //delay(100);
        //analogWrite(EN1,130);
        //analogWrite(EN2,130);
}


void stop()
{
        digitalWrite(m1,LOW);
        digitalWrite(m2,LOW);
        digitalWrite(m3,LOW);
        digitalWrite(m4,LOW);

       // setspeed();
       // analogWrite(EN1,0);
       // analogWrite(EN2,0);
}
void backward()
{


       digitalWrite(m1,LOW);
       digitalWrite(m2,HIGH);
       digitalWrite(m3,LOW);
       digitalWrite(m4,HIGH);
       setspeed();
       //delay(100);
      // analogWrite(EN1,130);
      // analogWrite(EN2,130);
}


void right()
{

        digitalWrite(m1,HIGH);
        digitalWrite(m2,LOW);
        digitalWrite(m3,LOW);
        digitalWrite(m4,LOW);
        setspeed();
        //delay(100);
        //analogWrite(EN1,130);
        //analogWrite(EN2,130);
}
void left()
{

        digitalWrite(m1,LOW);
        digitalWrite(m2,LOW);
        digitalWrite(m3,HIGH);
        digitalWrite(m4,LOW);
        setspeed();
        //delay(100);
        //digitalWrite(EN1,130);
        //digitalWrite(EN2,130);
        Serial.println("Left");
}


void setspeed()
{
     digitalWrite(EN1,50);
     digitalWrite(EN2,50);
}
```

4 Implementation

4.1 Description

There are mainly three parts in our project:

1. Transmitter
2. Receiver
3. Motor controller

The transmitter circuit will read the hand gestures and interpret into machine understandable characters via Arduino UNO which can be transmitted to the receiver. This circuit will sit on user's wrist. This circuit comprises an Arduino UNO, an RF Transmitter, an accelerometer (ADXL 335) and a mini breadboard. The ADXL 335 will get power up from 5 V output of Arduino. According to it's position and orientation, it will send (x,y,z) co-ordinates through the X-OUT, Y-OUT and Z-OUT. To avoid garbage values, the accelerometer needs to be calibrated first, which was done by code implementation in Arduino. Then, Arduino processes the co-ordinate data and converts it into characters. Then it sends the pulse code modulated output signal to RF transmitter's Data pin. The RF transmitter is also powered up by Arduino. The Arduino UNO is powered up by a 9V battery.

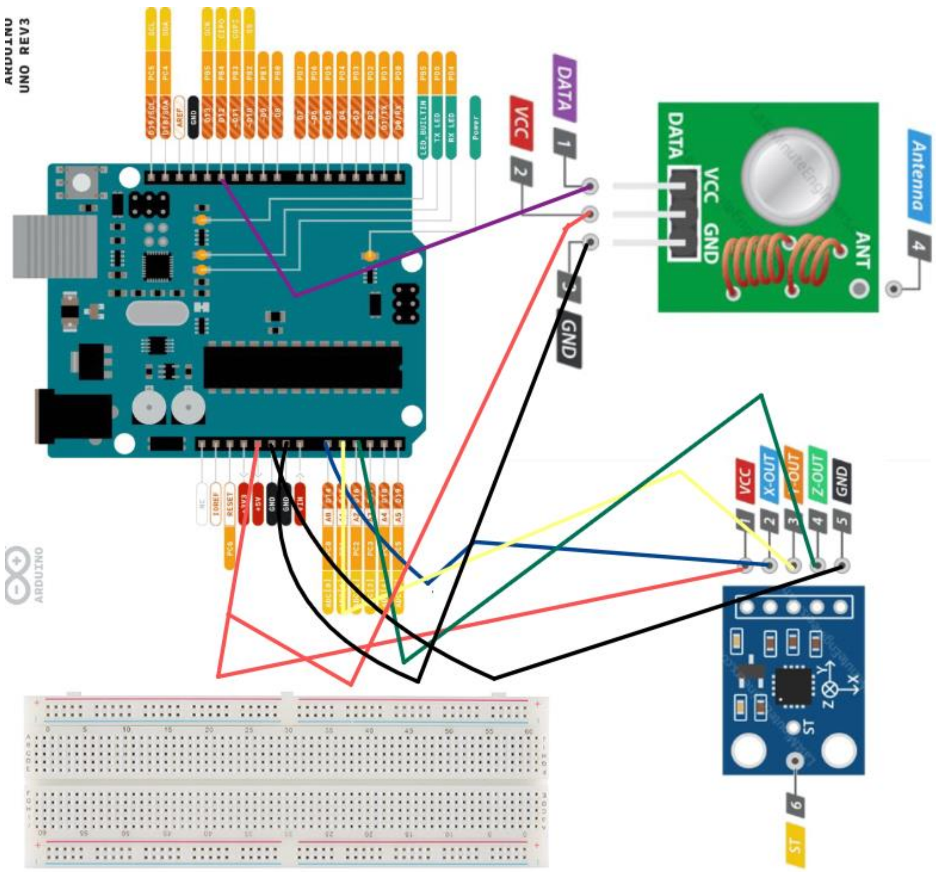


Figure1: Designing Transmitter Circuit

The receiver circuit is mounted on the chassis of the miniature vehicle. It consists an Arduino UNO, RF receiver, an HC SR04 ultrasonic sensor, a gyro sensor and a breadboard. From the

RF transmitter, the characters are received by RF receiver. Then, Arduino gets the data from RF receiver's data pin. For feedback system, an ultrasonic sensor is used so that it can detect obstacles and stop vehicle's motion. This sensor is also connected to Arduino's another input pin. A gyro sensor is also used for smooth turning of vehicle. The gyro sensor is calibrated via Arduino where a code is implemented to make a PI controller system. Arduino works as the brain of the whole system here. Everything is calibrated through it, powered up by it and it also generates output which is fed into motor driver.

The motor driver module is also powered up by Arduino's 5-volt output. According to the received character, motor driver runs the DC motors. It is also supplied to a 9V battery for additional power to drive the motors. The motor driver gives output voltage according to the input which is coming from the Arduino.

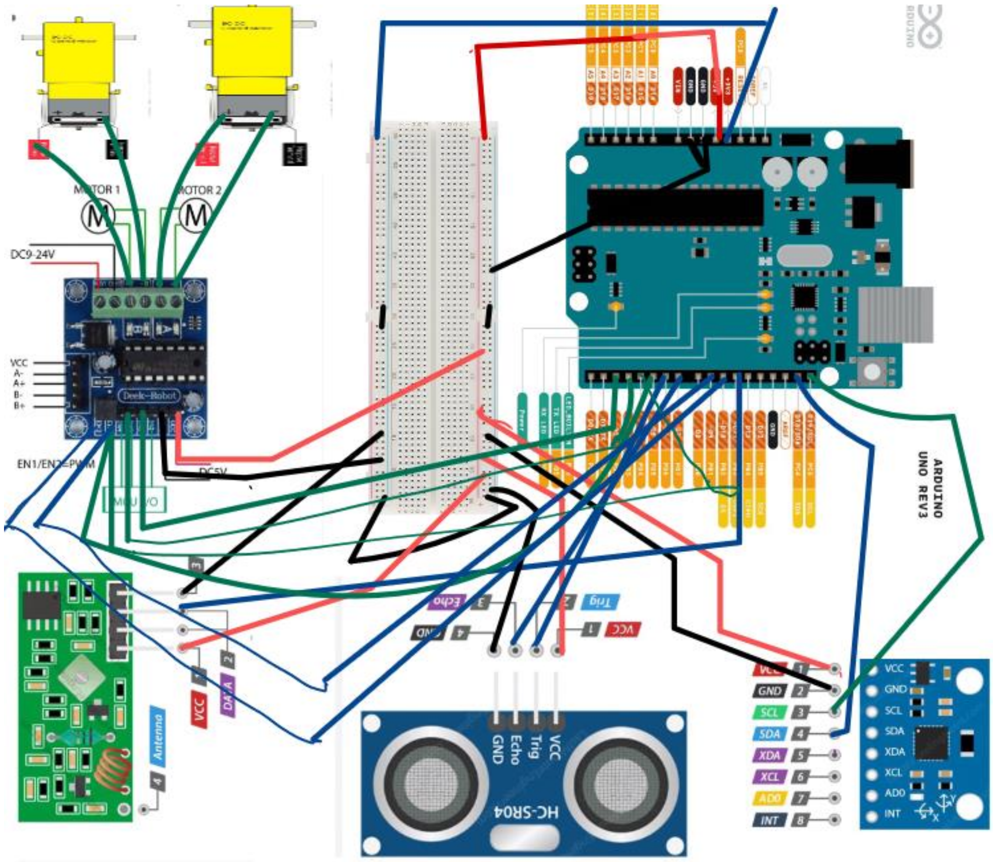


Figure 2: Designing Reciever Circuit

4.2 Experiment and Data Collection

We first tested our accelerometer which was situated in transmitter section. After setting it up with Arduino with necessary codes, it started to give the output which is (x,y,z) coordinates. But this raw data had to be converted into command so that it can be transferred easily via radio communication which was used in RF transmitter. To do this, we had to sample a lot of coordinate data while the accelerometer is in idle state. After sampling and averaging it, we used it as our idle coordinate which means "stop" or the car will not move in this state.

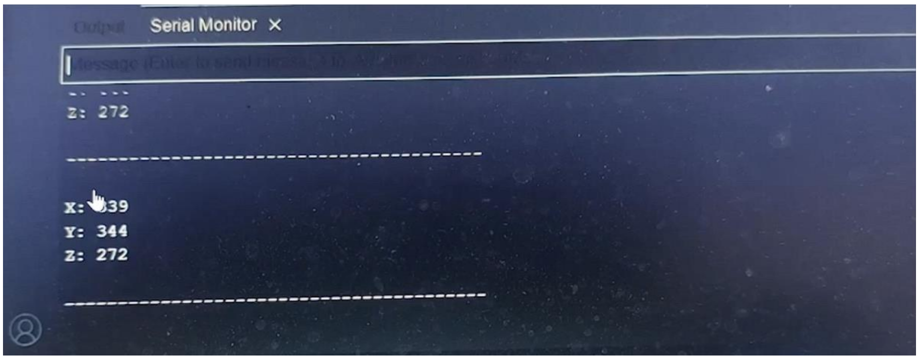


Figure 3: Accelerometer output observation

4.3 Data Analysis

After setting the idle state, we selected a particular threshold, such as 30 coordinate point. After that, we set up the movement characters according to the coordinates. For example, if we bend our hand backwards, Y increases, and we set the character "back" for this change of coordinate and vice versa. The same is for "Left" and "Right" character output. For increase in x value, it sends "Right" and vice versa. This was processed by Arduino and sent to the RF Transmitter. We observed these command changes which was received by RF Receiver.

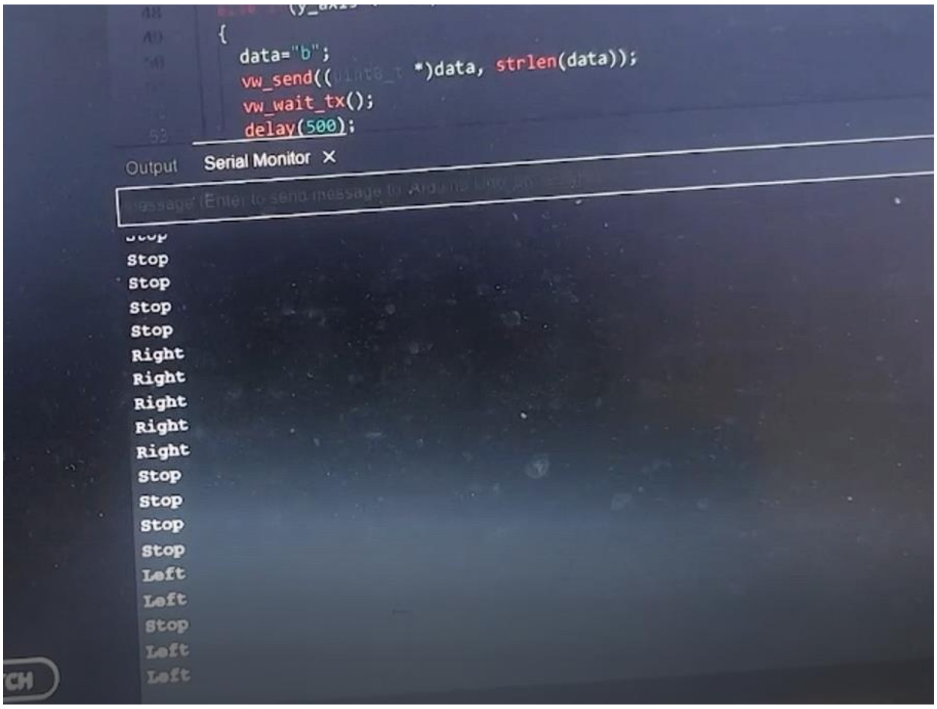


Figure 4: Receiver input observation

For the feedback system, we added gyro sensor for smooth turning and rotations. The implementation of gyro is incomplete. However, we also added an ultrasonic sensor which gives high or low output according to any obstacle it is facing. If ultrasonic sensor faces any obstacle, it gives high output and it goes to Arduino. By code manipulation, we then set the character "stop" as the state and the motor driver does not send any voltage to motors.

4.4 Results

We constructed our project very well except the gyro implementation part and our project runs successfully,

5 Design Analysis and Evaluation

5.1 Novelty

The novelty of our project lies in the following aspects:

- In contrast to other sensors and technologies, such as gyroscopes, cameras, or infrared, we utilized an accelerometer sensor to detect the user's hand motions because it is an easy and affordable solution. We are able to use a variety of gestures for varied wheelchair motions because of the accelerometer sensor's ability to measure the acceleration and hand direction in three dimensions.

- By eliminating the need for cables and enhancing the user's convenience and safety, we developed and implemented a wireless communication system between the sensor and the wheelchair. RF modules, batteries, and two Arduino microcontrollers make up the wireless communication system. The Arduino microcontrollers are set up to transmit and receive sensor data through RF, then use that information to drive the wheelchair motors.

5.2 Design Considerations

5.2.1 Considerations to public health and safety

The design of our gesture-controlled wheelchair for impaired people considers the public health and safety issues that may arise from its use. We considered the following aspects:

- The accelerometer detector is attached to the user's hand using a glove or a band, which ensures that the detector doesn't fall off or intrude with the user's hand movements. The detector is also calibrated so that it can avoid false or inaccurate readings that may result in unwanted wheelchair movements.
- The RF module that transmits the detector data to the wheelchair uses a low- power and low- frequency signal, which minimizes the interference with other signal and reduces the exposure to electromagnetic radiation. The RF module also has a built-in encryption and authentication system, which prevents unauthorized access or control of the wheelchair.
- The wheelchair has a speed limit and a braking system, which helps the wheelchair from moving too fast or colliding with obstacles. The wheelchair also has a homemade override switch, which allows the user or assistant to stop the wheelchair in case of an emergency or malfunction.

5.2.2 Considerations to environment

The design of our gesture-controlled wheelchair for impaired people considers the environmental impact that may arise from its use. We considered the following aspects

- Rechargeable batteries are used to power the accelerometer sensor and the RF module, which lowers the need for disposable batteries and the production of electronic waste. A case that surrounds the batteries further safeguards them from leakage or explosion that could endanger the user or the environment.
- To save weight and fuel consumption, the wheelchair is constructed of strong, lightweight materials like aluminum and plastic. The wheelchair has a small environmental impact thanks to the recyclable materials.
- The wheelchair is equipped with a noise-cancelling technology that reduces the sound of the wheelchair's motors and wheels. Rubber tires, shock absorbers, and mufflers make up the noise-cancelling system. The noise reduction device increases user comfort and safety while lowering environmental noise pollution.

5.2.3 Considerations to cultural and societal needs

The design of our gesture- controlled wheelchair for impaired people considers the cultural and societal needs that may arise from its use. We considered the following aspects

- The gesture control system is made to work with a variety of hand motions that may differ across cultures and geographical areas. The user can alter the gestures for various wheelchair motions, such as forward, backward, left, right, or stop, using the system. Additionally, the system gives the user feedback and instructions on how to consistently and correctly execute the motions.
- The wheelchair with gesture control is made to be accessible and supportive of people with disabilities. The user of the wheelchair is not required to wear any particular equipment or accessories, which could be uncomfortable or cause social stigma. Additionally, the wheelchair does not rely on vocal commands, which for some users with speech impairments or language difficulties may be challenging or impossible.
- The gesture-controlled wheelchair is made to be helpful to disabled persons who use it and socially acceptable. The wheelchair increases the user's independence and mobility, which may enhance their wellbeing and sense of self. The wheelchair also makes it easier for the user to engage and communicate with others, perhaps increasing their social involvement and participation.

5.3 Investigations

5.3.1 Literature Review

Dr. S.P.S. Saini and Diksha Goyal: Described the work in gesture rearrangement use as application as a wheelchair in the presentation "Accelerometer based hand gesture controlled wheelchair". In this instance, a gesture was identified using a three-axis accelerometer sensor. A system is made up of sensors that are used to identify hand or gesture movements. The MEMS accelerometer sensor (Micro Electro Mechanical device) in this device recognizes gesture. A gadget called an accelerometer uses electromechanical principles to measure acceleration forces. This accelerometer sensor has three axes and will be connected to the back of the hand and fingertips. It is a moving object. When it moves, the gesture is recognized, and the wheelchair will function in accordance with the sensor's movement. One learns the system is entirely dependent on sensors after analyzing the design of a "accelerometer-based hand gesture controlled wheelchair". This method is not very user-friendly because it cannot function if the sensor cannot move in an inclined position or direction. This method does not offer trustworthy assistance to people who are handicapped or impaired.

Chin-Ming Fu, Chung-Lin Huang, and Feng-sheng Chen: presented a paper titled "Hand Gesture Reorganization using a real time tracking method and Hidden Markov Models" that outlines the development of a system for hand gesture recognition against stationary backgrounds. The item's motion in this system provides crucial and practical information for object localization and extraction. The overall system consists of four modules: real-time tracking, extraction, feature extraction, and training for hidden Markov models (HMM). when using the real-time hand tracking and extraction technique, to monitor the moving hand and extract the hand region. Use a Fourier Descriptor (FD) to describe the spatial feature and motion analysis to describe the temporal feature. As our feature vector, combine the spatial and temporal information from the input image sequences. Next, apply the HMM model to identify the input gesture. After looking at the "Hand Gesture Reorganization using a real time tracking method and Hidden Markov Models" architecture, we can see that this system depends on the HMM model to identify the gesture. Because it is more difficult and less accurate to recognize gestures, it is neither useful nor user-compatible.

Anooda GeethuMohan, SmithaPaulose, M.P. Fathima, M.S. Sajana, and K.A. Anupama: Using a MEMS Accelerometer sensor, the author demonstrated "Automatic Wheelchair Using Gesture Reorganization Along With Room Automation." This system takes a real-time approach to hand tracking, identification, and restructuring for use in interacting with intelligent wheelchairs and human robots. This device was designed to use the accelerometer sensor to translate hand and finger motions into signals that a computer could understand. In order to detect gestures or finger restructuring, the accelerometer data is calibrated and filtered. We employ a three-axis accelerometer for wheelchair control, which successfully converts hand and finger motions into signals that a computer can understand. This technology is not user-friendly because it is too expensive for patients to handle.

Varalakshmi B.D. and Devikarani Patil: The "Hand Gesture Recognition for MP3player Using Image Processing Technique and PIC16F8779" presented discusses the gesture recognition procedure that takes place when using a web camera. This solution suggested that the gesture image be captured using a web camera and processed using a remote interface and a MATLAB controller. The (x, y, and z) readings of certain objects are used to connect the database when a captured image is sent to MATLAB. When an object moves in any direction, the accelerometer records the readings. When the accelerometer moves in a certain set of directions, it can identify gestures or a specific way to use an application. Through image processing, this system recognized purpose using the K-L Transform.

5.4 Limitations of Tools

- In this project we have used accelerometer which has some limitations. This sensor only measures the change in speed. It cannot measure the constant speed, or rotation along its own axis. Moreover, it is affected by sampling time uncertainty problem.
- We have also used Rf Module. Which has problem of interferences from other wireless devices, noises in the communication. Power consumption by transmitter and receiver.
- We have used battery holder which at times does not work though the battery is still good enough voltage do not come across the holder pins.
- Jumper wires and mini breadboard wire sometimes were defected.
- These all limitations can make the accuracy of the project less than 90%.

5.5 Impact Assessment

5.5.1 Assessment of Societal and Cultural Issues

Our project is determined to ease the life of disable person. It will be useful for their betterment of lifestyle. Their life will be easier and they will have the courage to engage more and more in social and cultural life. It will reduce their struggle partially to pursue their education. They will get a lot of advantages in academic and day to day life.

5.5.2 Assessment of Health and Safety Issues

Our project is very useful for the people with disability specially in the upper part of body. This reduces their physical stress. Because our designed wheelchair is not driven manually rather we implemented a control system by which only gesture command from the user can run and drive the wheelchair in a specific direction. This will improve their quality of life. Safety issue of users was also considered while implementing the hardware setup. We used a gyroscope and ultrasonic sensor which will give a feedback to the user by detecting any obstacle successfully. If any obstacle is detected properly it will immediately stop the system to stop collision or any kind of accident. So the wheelchair control model is safe for users as safety is prioritized while designing.

5.6 Sustainability and Environmental Impact Evaluation

In this project we used eco-friendly materials. So the equipments of our project do not seem hazardous to environment. Though it ensures safety for user and environment our project is not so sustainable as expcted. We have added many useful features like obstacle detection and speed control which increases our project more sustainable. But due to hardware limitations we somehow could not ensure that much sustainability as we expected.

5.7 Ethical Issues

The Project "Gesture-Controlled Wheelchair for Disabled People," involves several ethical considerations that should be carefully addressed throughout its development and deployment. Here are some ethical issues to be aware of:

Privacy and Data Security: If your wheelchair includes health monitoring features, it will collect sensitive data about users' health and location. Ensure robust data security measures are in place to protect this information from unauthorized access or breaches.

User Autonomy and Control: While gesture control can enhance user autonomy, it's essential to respect users' preferences. Ensure that the wheelchair allows users to switch between control methods and doesn't force them into a specific mode.

Safety and Reliability: The project must prioritize safety to prevent accidents or harm to users. Rigorous testing, safety mechanisms, and user training are essential to mitigate risks.

Equity and Accessibility: Make the project affordability and availability to all. Strive to make it accessible to a wide range of users, including those with limited financial means.

Sensitivity to User Needs: The project should consider the diverse needs and preferences of disabled individuals. Avoid making assumptions about what users want or need and actively involve them in the design and testing process.

6 Reflection on Individual and Team work

6.1 Individual Contribution of Each Member

ID	Name	Contribution
1906068	Md. Hasnat Karim	Transmitter circuit construction, project shopping
1906073	Md. Navid Newaz	Receiver circuit construction
1906075	Md. Mehedi Hasan	Receiver Arduino code construction
1906078	Asif Aftab Ronggon	Gyro calibration, feedback addition, accelerometer calibration, ultrasonic sensor implementation, Code algorithm
1906079	Md. Faiyaz Abid	Transmitter Arduino code construction
1906088	Md. Shahin Ferdous	Motor driver circuit build up, necessary code construction

6.2 Mode of Teamwork

All of us have done the project work equally. We had to learn new things during the project construction. The coding part was supervised meticulously by Ronggon. Under his supervision, we have completed our assigned tasks.

6.3 Log Book of Project Implementation

Date	Milestone achieved	Individual Role	Team Role	Comments
20.6.23	Designing the project	Asif Aftab Ronggon		Successful
6.7.23	Simulation (Accelerometer, gyroscope)	Asif Aftab Ronggon		Successful
9.7.23	Project Shopping	Md. Hasnat Karim		Successful
15.7.23	Transmitter part completion	Md. Hasnat Karim, Md. Faiyaz Abid		Successful

24.7.23	Receiver part completion	Md. Navid Newaz,, Md. Mehedi Hasan, Md. Shahin Ferdous		Successful
5.8.23	Code implementation	Asif Aftab Ronggon, Md. Mehedi Hasan, Md. Shahin Ferdous, Md. Faiyaz Abid		Successful
17.8.23	Ultrasonic Sensor addition	Asif Aftab Ronggon		Successful
27.8.23	Gyro sensor addition	Asif Aftab Ronggon, Md. Shahin Ferdous		Partially Successful

7 Communication

7.1 Executive Summary

With the use of cutting-edge technology, our own creation, the "Miniature Gesture Controlled Wheelchair for Disabled People," successfully increases the mobility and independence of people with disabilities. We have developed a novel wheelchair control system using Arduino, an accelerometer for gesture interpretation, a gyro for feedback, RF transmitter and receiver modules, and motor drivers. Users of this system may easily maneuver the wheelchair using simple hand gestures, enabling an improvement in the quality of life for those who have mobility issues. Our initiative shows how technology has the power to significantly improve the lives of people with disabilities.

7.2 User Manual

7.2.1. Introduction

Purpose of the Manual

This user manual provides detailed instructions for operating and maintaining the "Miniature Gesture Controlled Wheelchair for Disabled People." It is designed to assist users and caregivers in understanding the system's functionality, safety guidelines, and troubleshooting procedures.

Target Audience

This manual is intended for individuals with disabilities who will use the gesture-controlled wheelchair, as well as caregivers and technicians responsible for its operation and maintenance.

7.2.2. System Overview

Components

The system comprises the following key components:

- Arduino Microcontroller
- Accelerometer for Gesture Interpretation
- Gyroscope for Feedback System
- RF Transmitter and Receiver Modules
- Motor Drivers
- DC Motors
- Chassis and Wheels
- Power Supply

Functional Description

The Arduino interprets hand gestures captured by the accelerometer, allowing users to control the wheelchair's movement. The gyroscope provides feedback to maintain stability. RF modules enable wireless communication, while motor drivers control the DC motors for propulsion.

7.2.3. Safety Guidelines

General Safety Precautions

- Always operate the wheelchair on smooth and level surfaces.
- Do not exceed the specified weight limit for the wheelchair.
- Keep hands and loose clothing away from moving parts.
- Ensure that the wheelchair is in good working condition before each use.

7.2.4. Assembly and Setup

Wheelchair Assembly

Follow the manufacturer's instructions for assembling the chassis and wheels. Make a model of wheelchair or buy it from elsewhere.

Electronic Component Installation

1. Mount the Arduino and sensor modules securely on the wheelchair frame.
2. Connect the RF modules and motor drivers to the Arduino as per the provided schematic.
3. Ensure all connections are secure and free from loose wires.

Powering on the System

Connect the power supply to the wheelchair and ensure that the voltage is within the specified range. Power on the system according to the provided instructions.

7.2.5. Using the Gesture-Controlled Wheelchair

Powering the Wheelchair

Press the power button to turn on the wheelchair. Wait for the system initialization to complete.

Gesture Interpretation

Follow the gesture interpretation guidelines provided in the project documentation to control the wheelchair's movement.

Navigating the Wheelchair

Use gestures to navigate forward, backward, left, and right. Practice in a safe environment until you are comfortable with the controls.

Feedback System

The gyroscope helps maintain stability. Pay attention to any feedback provided by the system to avoid tipping over or losing control.

8 Project Management and Cost Analysis

Project Schedule

The project’s duration is 8 weeks, from June 24, 2023 to September 11, 2023.
The project’s milestones are as follows:

Milestone	Description	Due Date
M1	Literature review completed	
M2	Prototype design completed	
M3	Prototype implementation completed	
M4	Prototype testing and evaluation completed	
M5	Project report completed	September 11, 2023

Project Risks

The project’s potential risks and mitigation strategies are as follows:

Risk ID	Risk Description	Risk Impact	Risk Probability	Mitigation Strategy
R1	Difficulty in finding relevant literature sources.	Low	Low	Use online databases and search engines to find academic papers and articles related to gesture recognition and wheelchair control.

R2	Difficulty in designing a user-friendly and ergonomic gesture interface.	Medium	Medium	Consult with experts and users on the best practices and preferences for gesture interaction. Use existing standards and guidelines for gesture design.
R3	Difficulty in implementing the gesture recognition algorithm.	High	High	Use existing libraries and frameworks for accelerometer data processing and gesture recognition. Test and debug the code frequently and thoroughly.
R4	Difficulty in integrating the hardware components.	High	Medium	Follow the wiring diagrams and instructions carefully. Use appropriate tools and equipment for soldering and connecting the components. Test and debug the hardware frequently and thoroughly.
R5	Difficulty in testing and evaluating the prototype.	Medium	Low	Define clear and measurable criteria and metrics for testing and evaluation. Use a controlled and safe environment for testing. Collect and analyze the data objectively and systematically.
R6	Difficulty in writing the project report.	Low	Low	Follow the report format and structure provided by the course instructor. Use clear and concise language and visual aids to present the information. Cite the sources properly and avoid plagiarism.

8.1 Bill of Materials

Tools name	**Number used**	**Price per**	**Total (Tk)**
Arduino Uno	2	1100	2200
Chassis	1	450	450
Bibo motor	2	1200	2400
Accelerometer	1	550	550

MPU6050	1	200	200
RF transmitter	2	330	660
Ultrasonic sensor	1	93	93
L29D motor driver	1	215	215
Battery 3.7 V	3	30	90
Holder	1	20	20
Mini bread board	2	35	70
Jumper wire	2 sets	80	160
		Grand Total:	7108

9 Future Work

Gesture-controlled wheelchair project can lead to several exciting future work possibilities. Here are some given below:

Advanced Gesture Recognition: Enhance the gesture recognition system by using more sophisticated sensors and machine learning algorithms. This can make the wheelchair more responsive and adaptable to a wider range of gestures and user preferences.

Customizable User Profiles: Develop a user profile system that allows individual users to customize their gesture commands and settings. This can enhance user comfort and usability.

Smartphone Integration: Create a smartphone app that connects to the wheelchair, providing users with a convenient interface for control and monitoring. This can also enable features like GPS navigation and remote assistance. It also give some information to mobile if the person feel uncomfortable

Voice Control: Implement voice recognition technology to allow users to control the wheelchair through voice commands, providing an alternative control method.

Battery Efficiency: Optimize power management and battery efficiency to extend the wheelchair's range and reduce the need for frequent recharging.

Health monitoring: We can develop a fall detection system that uses accelerometers and other sensors to identify sudden falls or accidents. This can trigger automatic alerts to caregivers or emergency services. We can implement posture monitoring to ensure users maintain a healthy and ergonomic sitting position to prevent pressure ulcers and musculoskeletal issues.

10 References

◦https://lastminuteengineers.com/mpu6050-accel-gyro-arduino-tutorial/

◦https://lastminuteengineers.com/adxl335-accelerometer-arduino-tutorial/

◦**https://www.researchgate.net/publication/316551968_Hand_Gesture_Controlled_Wheelchair**