PR1101

Automation Projects

PREPARED BY

Chinay Bohara (2020BTechCSE021)

Priyam Bhardwaj (2020BTechCSE060)

Risha Sharma (2020BTechCSE064)

Shubhankar Mishra (2020BTechCSE078)

FACULTY CO-ORDINATORS

Mr. Divanshu Jain

Mr. HP Agarwal

Final Report

Department of Computer Science Engineering

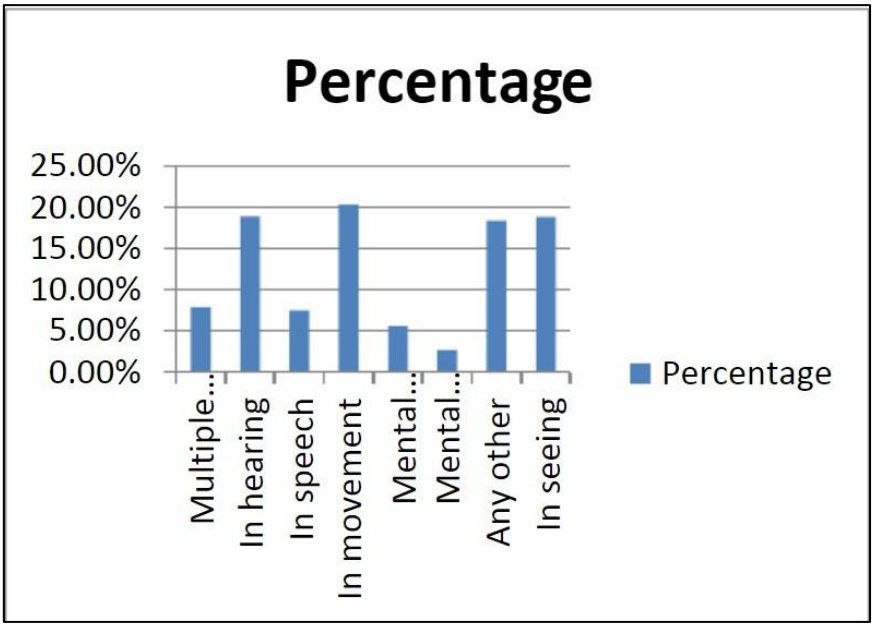
Institute of Engineering and Technology

JK Lakshmipat University, Jaipur

TABLE OF CONTENTS

ABSTRACT

Our project mainly focusses to the audience who are physically disabled or people under old age. The percent of disabled human beings has accelerated in each the agricultural and concrete elements of India. The incapacity will be through beginning or because of a few scientific or unintentional reason. This project pursuits to make a hand gesture-managed wheelchair to assist bodily disabled human beings, circulate from one region to some other through giving route from the hand. Today in India many humans are tormented by disability, there are humans whose decrease 1/2 of the frame is paralyzed. The Wheelchair will upload on to the comfort and make the existence of humans bit easier. Around 5436604 humans are affected from motion disability. Percentage of population which suffers from unique disabilities is proven in graph below. Out of overall disability maximum humans suffers from disability in motion.



TITLE OF PROJECT

Hand gesture controlled robotic wheelchair with fall detection

PROBLEM STATEMENT

The percentage of disabled people has increased in both rural and urban part of India. The disability could be by birth or due to some medical or accidental reason. However, its always not possible for someone else to be present with disabled person for their care. Hence, it creates a lot of chaos for them and make them feel helpless.

The aim of this project is to make a hand gesture-controlled wheelchair and using Accelerometer as sensor to help the physically disabled people in moving from one place to another just by giving direction from the hand.

While performing these gesture movement to move from one place to another, if accidently user get fall from the chair, then we have a sensor in built in the chair which can detect the motion of falling of the body and immediately inform someone known who is nearby through the message or fall alert alarm on their mobile phones.

METHODOLOGY

WIRELESS HAND GESTURE CONTROL: The system comprises of two main parts: Transmitter part and receiver part. In transmitter part the hand gesture is recognised by the sensor, digital output is transmitted to the controller and then transmitted to receiver side by the rf transmitter.

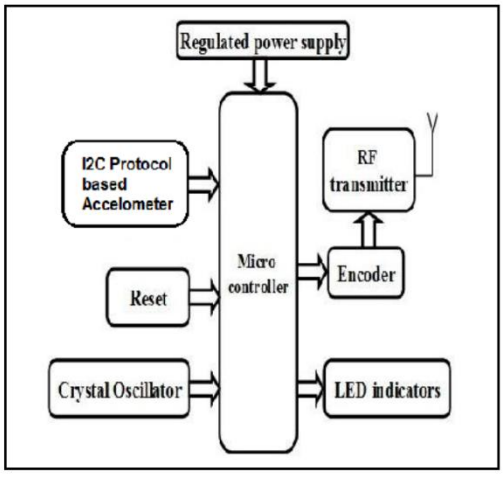


Fig. 1. Transmitter block diagram.

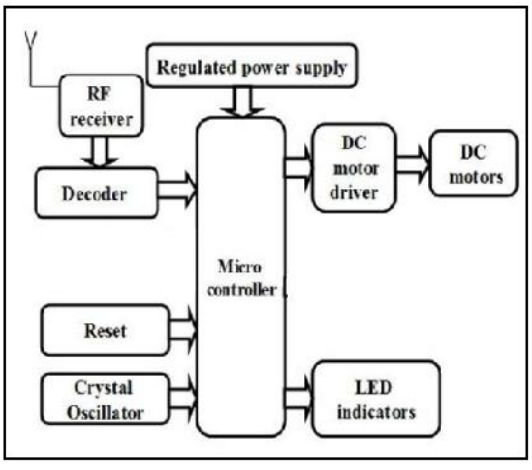


Fig. 2. Receiver block diagram.

In this project wheelchair is operated using hand gesture and to sense the hand gesture mems accelerometer is being used.

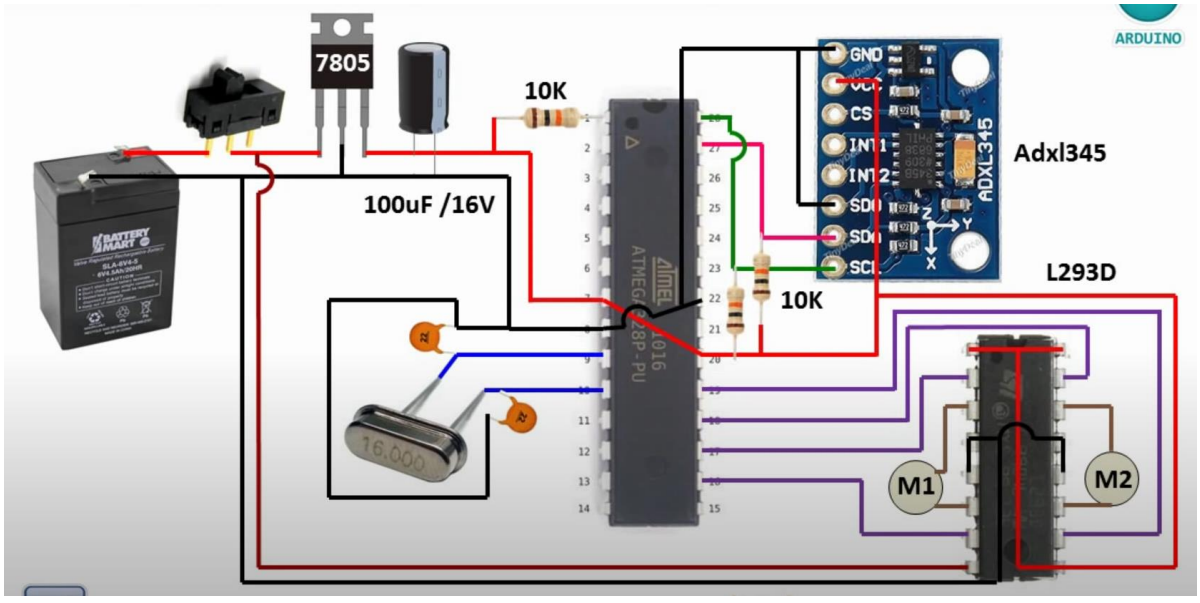
Transmission through RF is better than IR (infrared) because of many reasons. Firstly, signals through RF can travel through larger distances making it suitable for long range applications. Also, while IR mostly operates in line-of-sight mode, RF signals can travel even when there is an obstruction between transmitter & receiver. Next, RF transmission is stronger and more reliable than IR transmission. RF communication uses a specific frequency unlike IR signals which are affected by other IR emitting sources.

Also, the system, consists of switch which could enable the one of the 2 modes, hand gesture mode or remote-control mode. When we push the control, the transmitter sends a specific number of electrical pulses corresponding to that action through the air. The transmitter has its own power source, usually in the form of a 9-volt battery. Without the battery, the transmitter will not be able to send the radio waves to the receiver.

Once the wheelchair receives the radio waves, the motors kick into life to cause a specific action to occur. The power source sends power to all working parts, including the motor. The transmitter enables control through radio waves and the receiver activates the motors. When we press a button on the transmitter to make the wheelchair go forward or backward, a pair of electrical contact's touch. Receiver identifies signals, sends it to circuit.

WIRED HAND GESTURE CONTROL: All the above-mentioned work of transmitter and receiver could be done with the help of 1 Arduino UNO, just need to make all the connections of transmitter and receiver on one PCB circuit and with one microcontroller. No need of RF Transmitter to send signals.

Diagram:

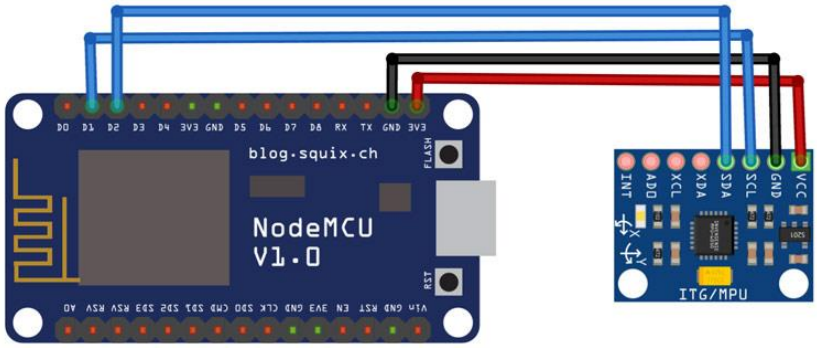


ATMega328p is an IC that will be programmed by Arduino UNO. Other than this L293D is a motor driver that will help the wheelchair to move according to the instruction signals received by it. Here, only one ADxl345 is needed, and it will work as a remote for the wheelchair.

FALL DETECTION:

It can notify the individual or family member when it detects a fall and reduces the risk of delay in medical attention. So, it leads to the development of various types of automatic fall detector systems. We are going to build an IoT Fall Detector Using MPU6050, NodeMCU ESP8266, and IFTTT Application.

MPU6050 sensor module features a gyroscope and an accelerometer. The gyroscope is used to determine the orientation and the accelerometer provides information about the angular parameter such as the three-axis data. To detect the fall, we will compare the acceleration magnitude with the threshold value. If the fall is detected, the device will send an SMS to the concerned person. NodeMCU is used here as a microcontroller and Wi-Fi module to connect with IFTTT to send SMS.



INNOVATION/DISTINCT FEATURES OF OUR PROJECT

Our project is different from regular robotic wheelchair as follows:

1. It has 2 operational features to control the wheelchair, one by using remote and other by using hand gestures. So, whenever the person gets tired of walking with person on wheelchair, he could operate the wheelchair nearby surroundings by sitting at one place.
2. If there is no one to operate the wheelchair, so the hand gesture system mode could be enabled, which allows the paralyzed or disabled person to use the hand gestures to control the chair.
3. A fall detection system would be introduced so that, an alarm would sound, and the message of alert would be send to person's knowns and relatives. Hence, easy identification of falling wheelchair.

Benefits to people who are:

a. Paralytic person.

b. Those who crawl.

c. Those who walk with the help of aid.

d. Those have acute and permanent problems of joints/muscles.

e. Those who have stiffness or tightness in movement or have loose, involuntary movements or tremors

of the body or have fragile bones.

f. Those who have difficulty balancing and coordinating body movement.

g. Those who have loss of sensation in body due to paralysis, Leprosy etc.

h. Those who have deformity of body like hunch back or are dwarf.

HARDWARE AND SOFTWARE REQUIREMENTS

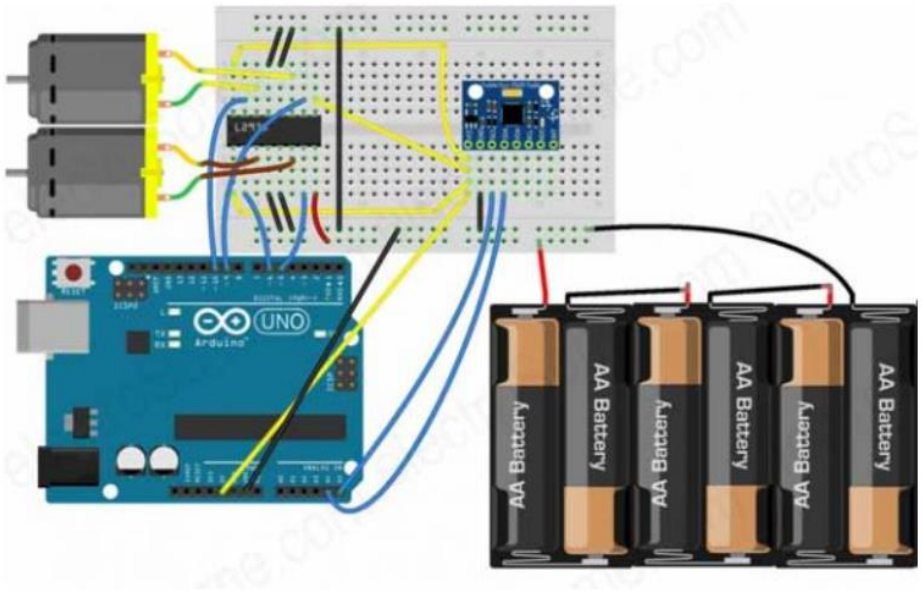
Hardware:

- → Atmega 328 IC/Arduino UNO -1
- → NodeMCU ESP8266 -1
- → Adxl345/MPU6050 -2
- → 16MHz -1
- → 22pf (2X) Capacitor -2
- → LM7805 -1
- → L293 Motor Driver -1
- → 10k Ohm Resister -3
- → 100Uf/16v Electrolytic Capacitor-1
- → Connecting Wires
- → Wheels -2
- → DC Motor -2
- → Battery
- → Slide Switch

Software:

- → Arduino
- → IFTTT

CIRCUIT DIAGRAM



Controlling of DC Motors using MPU-6050 Circuit Diagram

TESTING AND PARAMETERS

Parameters for gesture control are (Amplitude Vector and direction):

1. Ay <80 degree – move Reverse
2. Ay >145 degree – move Forward
3. Ax>155 degree – move right turn
4. Ax<80 degree – move left turn

Ax > 100 && Ax < 170 && Ay > 80 && Ay < 130 – stop the chair

```
sensor.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
ax = map(ax, -17000, 17000, 0, 255 ); //Send X axis data
ay = map(ay, -17000, 17000, 0, 255);  //Send Y axis data
```

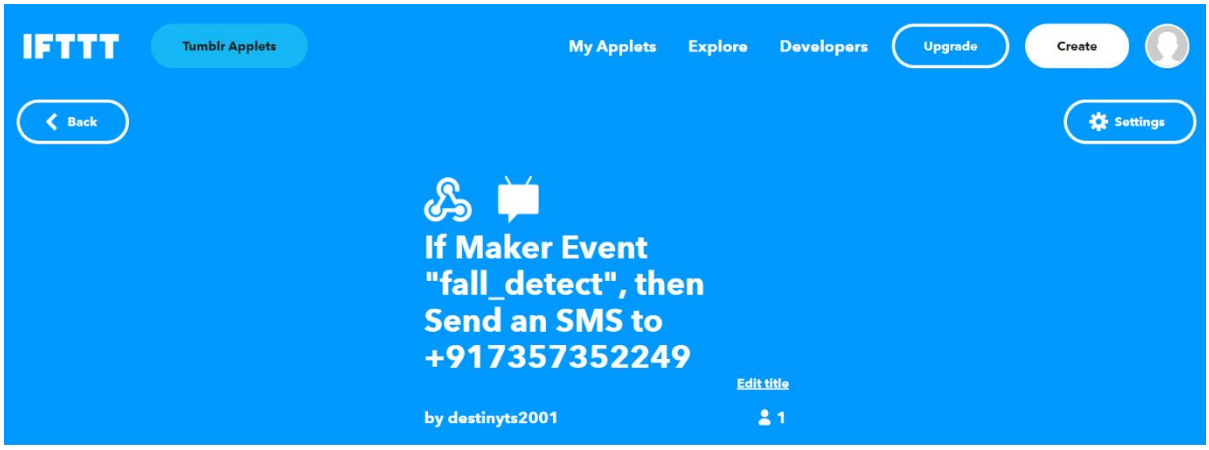
Parameters for Fall Detection (Amplitude Vector and Direction):

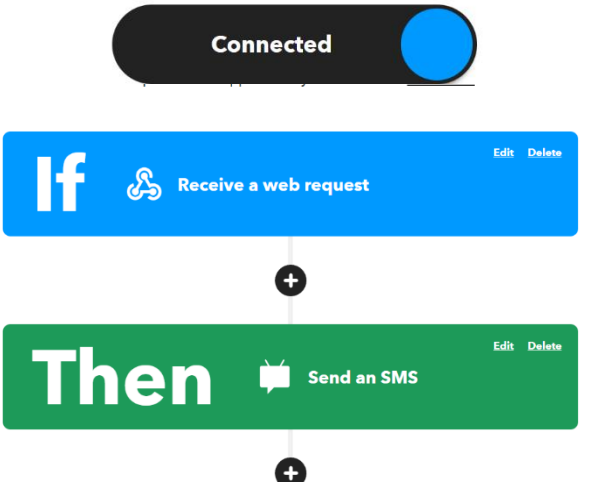
Works for 3 trigger points:

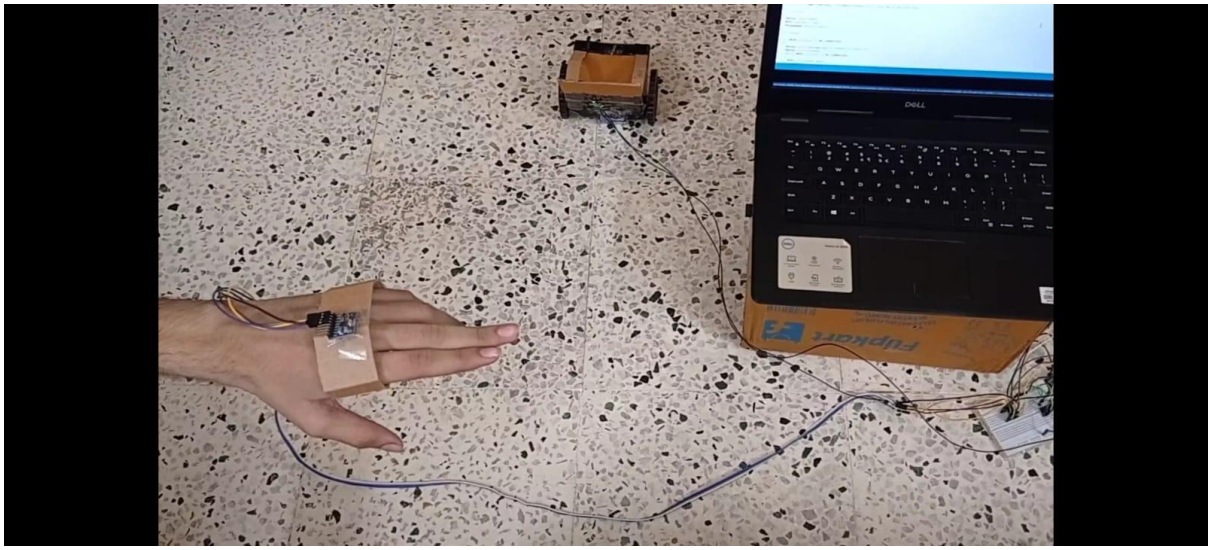
1. Amp<=2 – Trigger 1 Activated
2. Amp>=12 – Trigger 2 Activated
3. If large change in angle found after amplitude change – Trigger 3 Activated

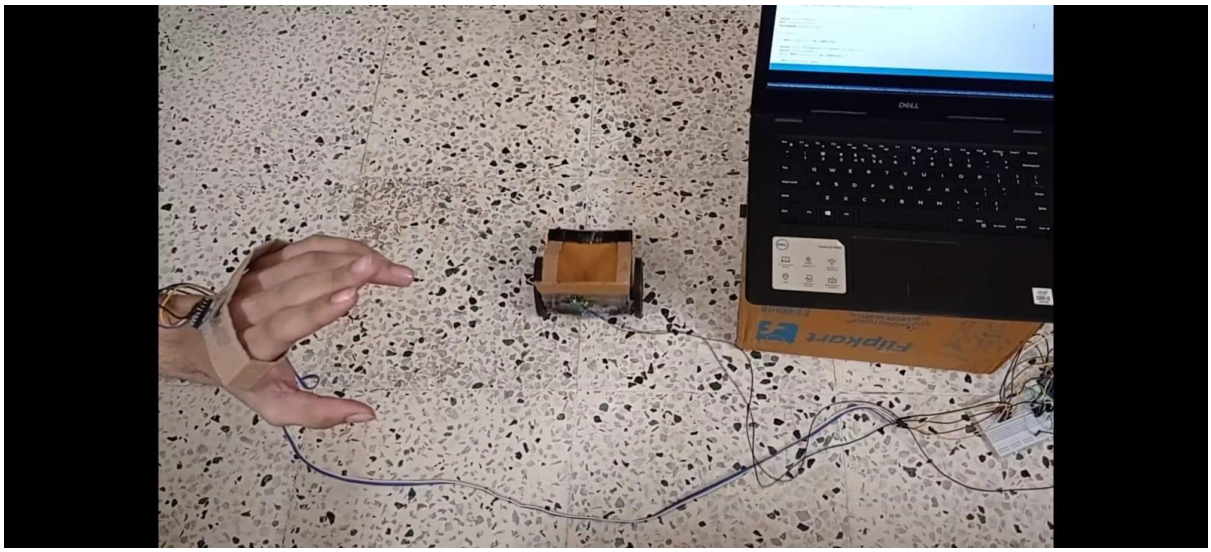
Hence, now the fall is detected.

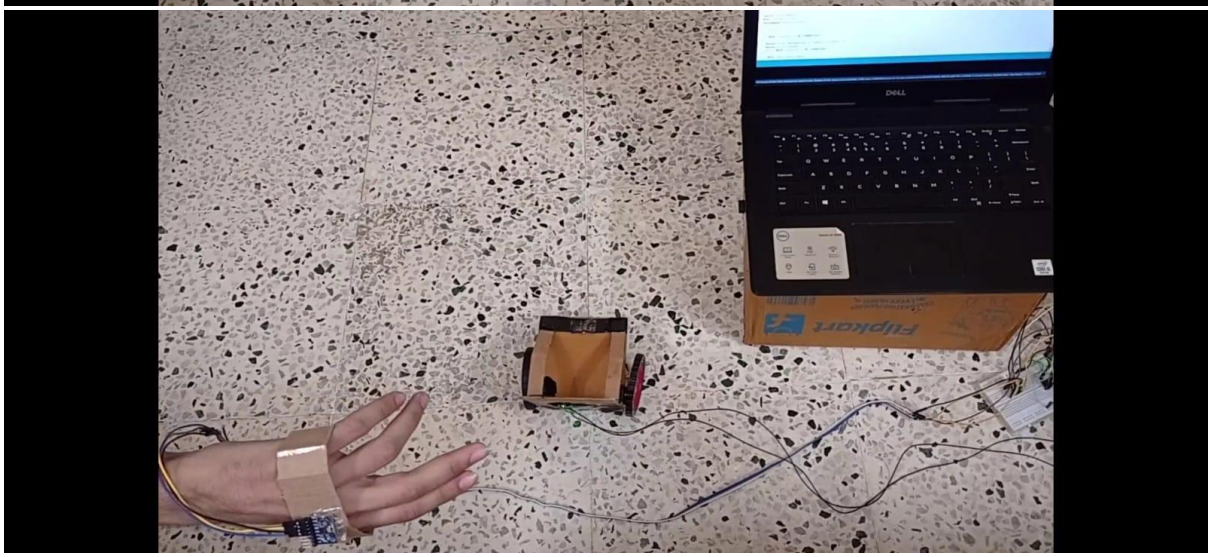
```
ax = (AcX-2050)/16384.00;
ay = (AcY-77)/16384.00;
az = (AcZ-1947)/16384.00;
```



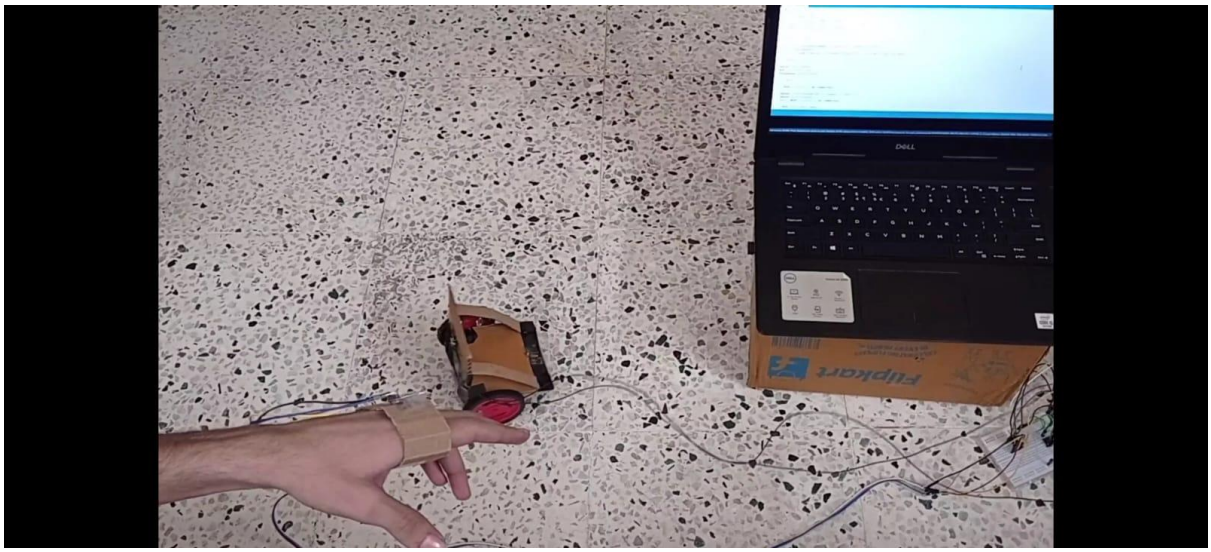


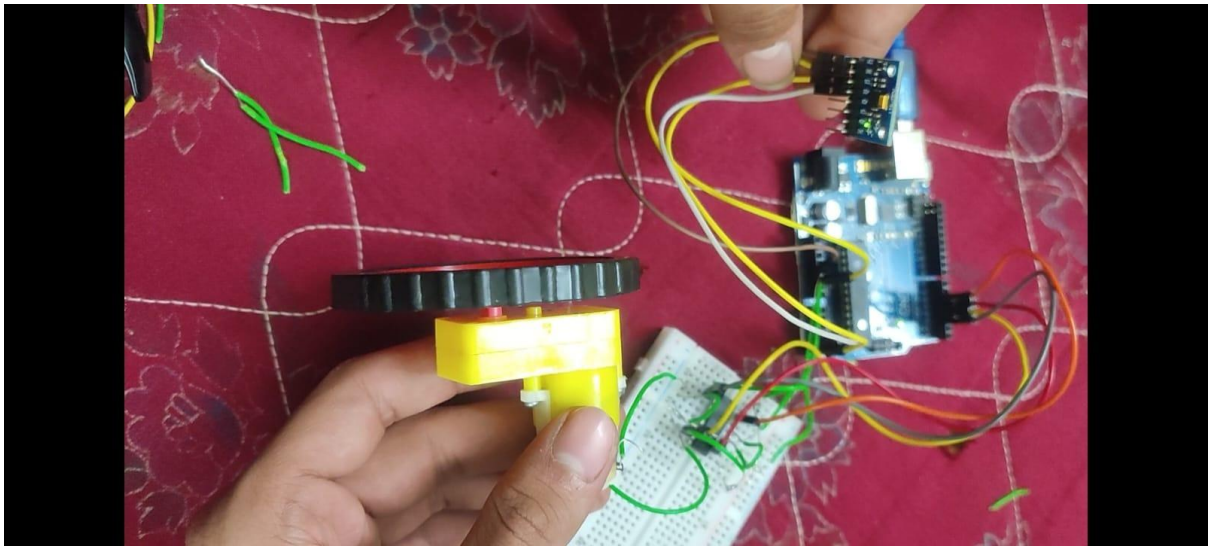

DELL
Flipkart


DELL
Flipkart


DELL
Flipkart


DELL
Flipkart


DELL
Flipkart



CONCLUSION

The wheelchair is fully capable of understanding the hand gestures in accordance with the gesture given by the person who will be using the wheelchair. Certain improvisation and improvement can be done to make the wheelchair more reachable to those whose whole body is paralyzed. Certain eyes gesture or brain signals reader can be imparted on the wheelchair system to make it better.

FUTURE SCOPE

The hand gesture wheelchair could bridge the gap between man and machine. Further this hand gesture can be changed to speech and brain signal recognition which will be a battle winning factor for all those people whose whole body is paralyzed. We can further improve wheelchairs by making it with low cost and high accuracy which are operating by a wireless remote with various sensors. An array of sensors can be used and integrating the inputs of multiple sensors and then processing them.

REFERENCES

- Prof. Vishal V. Pande, "Hand Gesture Based Wheelchair Movement Control for Disabled person Using MEMS" et al Int. Journal of Engineering Research and Applications Vol. 4, Issue 4(Version 4), April 2014, pp.152-158
- https://www.researchgate.net/publication/316551968_Hand_Gesture_Controlled_Wheelchair
- https://ieeexplore.ieee.org/document/9230849
- https://www.eetimes.eu/sensors-for-fall-detection/
- https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.669.9587&rep=rep1&type=pdf
- https://www.hackster.io/embeddedlab786/joystick-controlled-wheelchair-using-arduino-and-l298-motor-35813d
- https://ijcrt.org/papers/IJCRT1813627.pdf
- https://iotdesignpro.com/projects/iot-based-fall-detection-system-using-nodemcu-esp8266-and-accelerometer-mpu6050
- https://www.youtube.com/watch?v=TNpaePxo_fI&t=3s
- https://www.ijert.org/research/hand-gesture-controlled-robot-IJERTV9IS110014.pdf
- https://scholar.google.co.in/scholar_url?url=https://iopscience.iop.org/article/10.1088/1742-6596/1380/1/012060/pdf&hl=en&sa=X&ei=ZV6GY5LGLY70mgG3m7KgDQ&scisig=AAGBfm1j9m4ZtvbrjaaL_OeAaJ_2NHEB6Q&oi=scholarr
- https://www.sciencedirect.com/science/article/abs/pii/S2542660519301623

APPENDIX

1. Fall Detection Code

```
#include <Wire.h>
#include <ESP8266WiFi.h>
const int MPU_addr=0x68;  // I2C address of the MPU-6050
int16_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;
float ax=0, ay=0, az=0, gx=0, gy=0, gz=0;
boolean fall = false; //stores if a fall has occurred
boolean trigger1=false; //stores if first trigger (lower threshold) has occurred
boolean trigger2=false; //stores if second trigger (upper threshold) has occurred
boolean trigger3=false; //stores if third trigger (orientation change) has occurred
byte trigger1count=0; //stores the counts past since trigger 1 was set true
byte trigger2count=0; //stores the counts past since trigger 2 was set true
byte trigger3count=0; //stores the counts past since trigger 3 was set true
int angleChange=0;
// WiFi network info.
const char *ssid = "Mi 10i"; // Enter your Wi-Fi Name
const char *pass = "0987654321"; // Enter your Wi-Fi Password
void send_event(const char *event);
const char *host = "maker.ifttt.com";
const char *privateKey = "k4FhW7DduKW2akgrEOo4aUfNTapzD994O4HBkc1i2yw";
void setup(){
 Serial.begin(115200);
 Wire.begin();
 Wire.beginTransmission(MPU_addr);
 Wire.write(0x6B);  // PWR_MGMT_1 register
 Wire.write(0);     // set to zero (wakes up the MPU-6050)
 Wire.endTransmission(true);
 Serial.println("Wrote to IMU");
  Serial.println("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");              // print ... till not connected
  }
  Serial.println("");
  Serial.println("WiFi connected");
```

```
}
void loop(){
 mpu_read();
 ax = (AcX-2050)/16384.00;
 ay = (AcY-77)/16384.00;
 az = (AcZ-1947)/16384.00;
 gx = (GyX+270)/131.07;
 gy = (GyY-351)/131.07;
 gz = (GyZ+136)/131.07;
 // calculating Amplitute vactor for 3 axis
 float Raw_Amp = pow(pow(ax,2)+pow(ay,2)+pow(az,2),0.5);
 int Amp = Raw_Amp * 10;  // Mulitiplied by 10 bcz values are between 0 to 1
 Serial.println(Amp);
 if (Amp<=2 && trigger2==false){ //if AM breaks lower threshold (0.4g)
   trigger1=true;
   Serial.println("TRIGGER 1 ACTIVATED");
   }
 if (trigger1==true){
   trigger1count++;
   if (Amp>=12){ //if AM breaks upper threshold (3g)
     trigger2=true;

     Serial.println("TRIGGER 2 ACTIVATED");
     trigger1=false; trigger1count=0;
     }
 }
 if (trigger2==true){
   trigger2count++;
   angleChange = pow(pow(gx,2)+pow(gy,2)+pow(gz,2),0.5); Serial.println(angleChange);
   if (angleChange>=30 && angleChange<=400){ //if orientation changes by between 80-100 degrees
     trigger3=true; trigger2=false; trigger2count=0;
     Serial.println(angleChange);
     Serial.println("TRIGGER 3 ACTIVATED");
       }
   }
 if (trigger3==true){
    trigger3count++;
    if (trigger3count>=10){
       angleChange = pow(pow(gx,2)+pow(gy,2)+pow(gz,2),0.5);
       //delay(10);
       Serial.println(angleChange);
       if ((angleChange>=0) && (angleChange<=10)){ //if orientation changes remains between 0-10 degrees
           fall=true; trigger3=false; trigger3count=0;
           Serial.println(angleChange);
             }
       else{ //user regained normal orientation
          trigger3=false; trigger3count=0;
          Serial.println("TRIGGER 3 DEACTIVATED");
       }
     }
  }
 if (fall==true){ //in event of a fall detection
   Serial.println("fall_detect");
   send_event("fall_detect");
   fall=false;
   }
 if (trigger2count>=6){ //allow 0.5s for orientation change
   trigger2=false; trigger2count=0;
   Serial.println("TRIGGER 2 DECACTIVATED");
   }
 if (trigger1count>=6){ //allow 0.5s for AM to break upper threshold
   trigger1=false; trigger1count=0;
   Serial.println("TRIGGER 1 DECACTIVATED");
   }
  delay(100);
   }
```

```
void mpu_read(){
 Wire.beginTransmission(MPU_addr);
 Wire.write(0x3B);  // starting with register 0x3B (ACCEL_XOUT_H)
 Wire.endTransmission(false);
 Wire.requestFrom(MPU_addr,14,true);  // request a total of 14 registers
 AcX=Wire.read()<<8|Wire.read();  // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
 AcY=Wire.read()<<8|Wire.read();  // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
 AcZ=Wire.read()<<8|Wire.read();  // 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)
 Tmp=Wire.read()<<8|Wire.read();  // 0x41 (TEMP_OUT_H) & 0x42 (TEMP_OUT_L)
 GyX=Wire.read()<<8|Wire.read();  // 0x43 (GYRO_XOUT_H) & 0x44 (GYRO_XOUT_L)
 GyY=Wire.read()<<8|Wire.read();  // 0x45 (GYRO_YOUT_H) & 0x46 (GYRO_YOUT_L)
 GyZ=Wire.read()<<8|Wire.read();  // 0x47 (GYRO_ZOUT_H) & 0x48 (GYRO_ZOUT_L)
 }
void send_event(const char *event)
{
  Serial.print("Connecting to ");
  Serial.println(host);
    // Use WiFiClient class to create TCP connections
  WiFiClient client;
  const int httpPort = 80;
  if (!client.connect(host, httpPort)) {
    Serial.println("Connection failed");
    return;
  }
    // We now create a URI for the request
  String url = "/trigger/";
  url += event;
  url += "/with/key/";
  url += privateKey;
  Serial.print("Requesting URL: ");
  Serial.println(url);
  // This will send the request to the server
  client.print(String("GET ") + url + " HTTP/1.1\r\n" + "Host: " + host + "\r\n" +
               "Connection: close\r\n\r\n");
               "Connection: close\r\n\r\n");
  while(client.connected())
  {
    if(client.available())
    {
      String line = client.readStringUntil('\r');
      Serial.print(line);
    } else {
      // No data yet, wait a bit
      delay(50);
    };
  }
  Serial.println();
  Serial.println("closing connection");
  client.stop();
}
```

2. Gesture Control Code

```
#include <Wire.h>
#include <MPU6050.h>

#define first_motor_pin1 5
#define first_motor_pin2 6
#define second_motor_pin1 9
#define second_motor_pin2 10

MPU6050 sensor;
int16_t ax, ay, az;
int16_t gx, gy, gz;

int first_motor_speed;
int second_motor_speed;

void setup ( )
{
  Wire.begin( );
  Serial.begin (9600);
  Serial.println ("Initializing the sensor");
  sensor.initialize ( );
  Serial.println(sensor.testConnection( ) ? "Successfully Connected" : "Connection failed");
  delay(1000);
  Serial.println("Taking Values from the sensor");
  delay(1000);
}

void loop ( )
{
  sensor.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
  ax = map(ax, -17000, 17000, 0, 255 ); //Send X axis data
  ay = map(ay, -17000, 17000, 0, 255);  //Send Y axis data
  first_motor_speed = 255-ax;
  second_motor_speed = 255+ax;
  Serial.print("X: ");
  Serial.print(ax);
  Serial.print("    ");
```

```
    Serial.print("    ");
    Serial.print("Y: ");
    Serial.print(ay);
    Serial.print("\n");

    //analogWrite (first_motor_pin2, first_motor_speed);
    //analogWrite (second_motor_pin2, second_motor_speed);
    //delay (200);
    while(true){
      if (ay < 80) { //Reverse
        digitalWrite(first_motor_pin1, HIGH);
        digitalWrite(first_motor_pin2, LOW);
        digitalWrite(second_motor_pin1, LOW);
        digitalWrite(second_motor_pin2, HIGH);

      }

      if (ay > 145) {//forward
        digitalWrite(first_motor_pin1, LOW);
        digitalWrite(first_motor_pin2, HIGH);
        digitalWrite(second_motor_pin1, HIGH);
        digitalWrite(second_motor_pin2, LOW);
      }

      if (ax > 155) {//right turn
        digitalWrite(first_motor_pin1, LOW);
        digitalWrite(first_motor_pin2, HIGH);
        digitalWrite(second_motor_pin1, LOW);
        digitalWrite(second_motor_pin2, HIGH);
      }

      if (ax < 80) {//left turn
        digitalWrite(first_motor_pin1, HIGH);
        digitalWrite(first_motor_pin2, LOW);
        digitalWrite(second_motor_pin1, HIGH);
        digitalWrite(second_motor_pin2, LOW);
      }

     if (ax > 100 && ax < 170 && ay > 80 && ay < 130) { //stop
        digitalWrite(first_motor_pin1, LOW);
        digitalWrite(first_motor_pin2, LOW);
        digitalWrite(second_motor_pin1, LOW);
        digitalWrite(second_motor_pin2, LOW);
      }
    }
}
```