VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

TITLE

Smart Gesture controlled Wheelchair

IoT Domain Analyst- Project

Submitted to: Prof Pradeep T

Members:

1. M.Achyuth 19BEC0321
2. Vipul Kumar 19BEC0332
3. Aditya Sagar 19BEC0391
4. B Pavan Sriram 19BEC0423

Abstract:

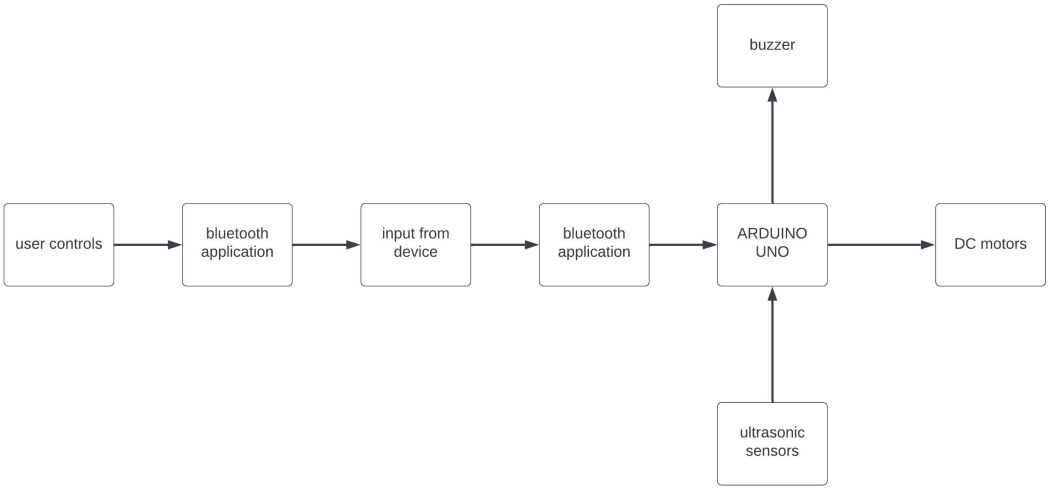
According to a research there are about 6 million populations in the world who are paralyzed and needs a wheelchair for their mobility. Earlier the wheel chairs had to be moved and be externally supported by any person. To help overcome this “joystick-controlled wheelchairs” are developed. But in regular use, these joystick-controlled wheelchairs became difficult to use. Especially in the case of paralyzed people, the use of joystick became more difficult due to the hard buttons and unidirectional use of the joysticks. To overcome these problems, we’ve tried to develop a “gesture-controlled wheelchair” which can be moved with a slight tilt of the hand. This can be used in both hands and can be controlled to come to the user from a distance. The current work is implemented with Arduino based devices such as Arduino NANO and UNO processors and programmed through Arduino IDE.

Introduction:

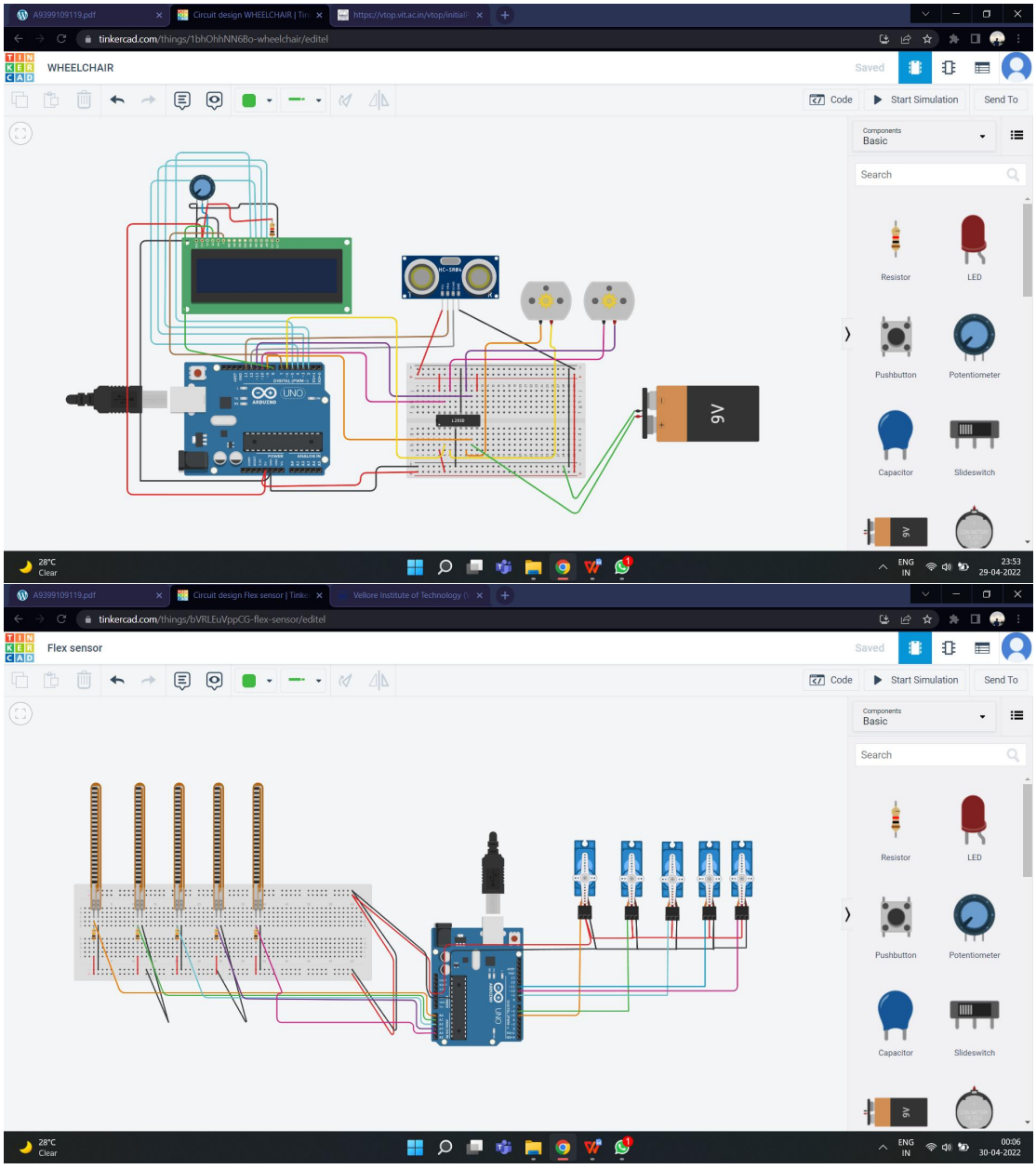
In present world around 650 million people are physically challenged. Hence it is highly essential to build a system in order to make their lives happy and more flexible. An advanced wireless wheel chair system is designed to make physically challenged person’s life more convenient and flexible. In the present project a hand gesture-based wheel chair system is developed for handicapped persons, those who are partially paralyzed. In this process the physical movement of the hand is sensed with sensors. The sensed signal is converted into electrical signal and then the signal is transmitted after converting analog signal into digital signal. The designed system best suitable who are paralyzed in accidents or paralyzed due to genetic order. The designed system is enabled and working based on user hand/finger movement commands. The low cost machine design is one of the main concerns to develop this machine.

The handicapped person can control the wheelchair and can send commands with their finger or arm or hand movement to wheelchair to call the wheel chair away from the person’s location. Hence the wireless hand gesture based wheelchair helps the physically challenged persons to move from one location to other location independently without second person intervention. The paralyzed persons those who cannot speak, walk or cannot move any body parts the existing cannot solve their problems. Just with the tilt of his/her hand they are able to move from one place to make them self-dependent. During the development of human machine interface for wheel chair control several designs are proposed based on input signal type.

Block Diagram:



Circuit:



Working principle

This wheel chair model can be considered into two types one is Bluetooth to Arduino part and the second is Arduino to DC motors.

Arduino of the motor is connected to the arduino in the flex sensor part and the flex sensor is connected to the motors of the wheel chair.

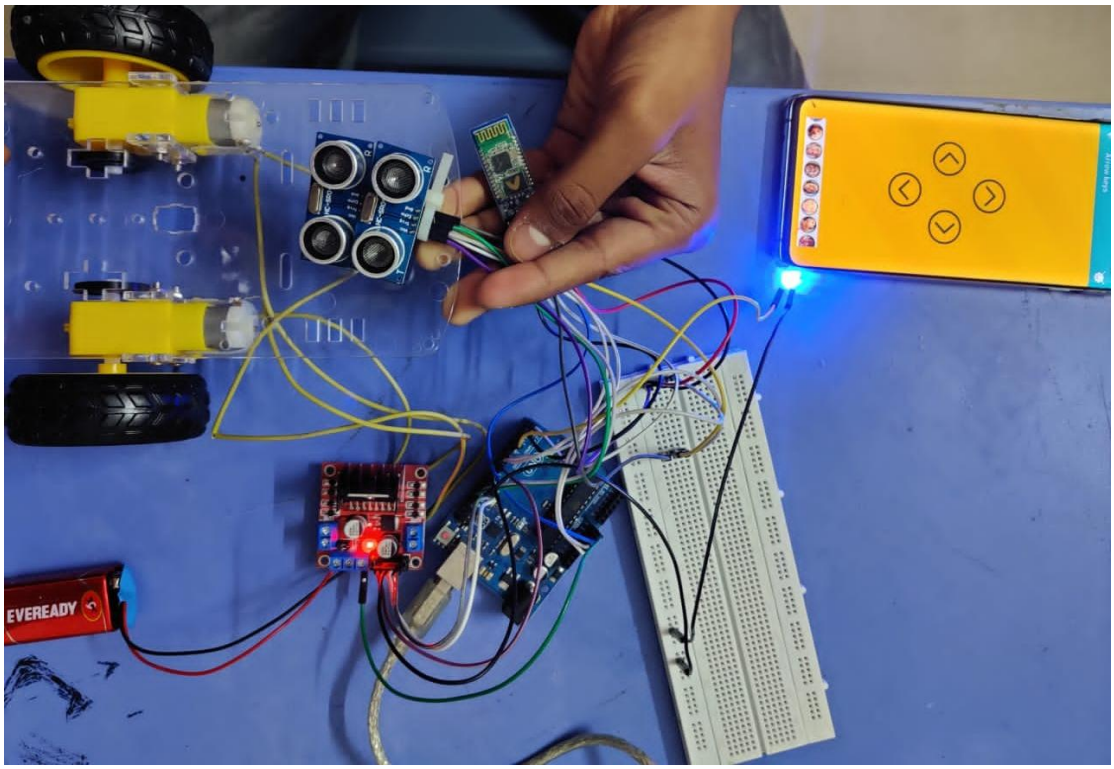
Then according to the pressure change of the flex sensors, and the threshold voltage set, we can make the motors move individually.

In the first part, the mobile app gestures are sent through the Bluetooth module to the micro-controller (using HC-05 module).

And the second part consists input from the ultrasonic sensor which helps in detection of the objection in the way, and it also consists of a panic alarm with a buzzer which is used as an indication when there is an emergency situation.

At last the input converted into voltage sent to the DC motors and the movement In the wheels are observed.

The motors also work through gyroscope present in mobile blue-tooth app, which can be used instead of accelerometer or joystick.
Everyone in the present day has a phone, so this is a very convenient way to make a smart wheelchair.



Result:

The ultrasonic sensor detects the surroundings and senses if there is any obstacle around and gives signals through the LED light.
Then when there is no obstacle around, we can send data to the arduino to move it forward, reverse, right or left.
We can send the information in three modes, our main focus is on sending information through flex sensors, then the other modes are through phone tilt (gyroscope) and arrow keys on our phone which all are connected through blue-tooth.
The ultrasonic sensors keep scanning an area of 120 degrees, and stops the movement of the wheelchair if there's any obstacle within the threshold distance set.

Code:
Flex sensor code:

```
#include <Servo.h>

Servo servo_1;
Servo servo_2;
Servo servo_3;
Servo servo_4;
Servo servo_5;

int flex_1 = A0;
int flex_2 = A1;
int flex_3 = A2;
int flex_4 = A3;
int flex_5 = A4;
;
void setup()
{

  servo_1.attach(5);
  servo_2.attach(6);
  servo_3.attach(9);
  servo_5.attach(10);
  servo_4.attach(11);
  Serial.begin(9600);

}


void loop()
{
  int flex_1_pos;
  int servo_1_pos;
  flex_1_pos = analogRead(flex_1);
  servo_1_pos = map(flex_1_pos, 770, 950, 0, 180);
  servo_1_pos = constrain(servo_1_pos, 0, 180);
  servo_1.write(servo_1_pos);

  int flex_2_pos;
  int servo_2_pos;
  flex_2_pos = analogRead(flex_2);
  servo_2_pos = map(flex_2_pos, 770, 950, 0, 180);
  servo_2_pos = constrain(servo_2_pos, 0, 180);
  servo_2.write(servo_2_pos);

  int flex_3_pos;
  int servo_3_pos;
  flex_3_pos = analogRead(flex_3);
  servo_3_pos = map(flex_3_pos, 770, 950, 0, 180);
  servo_3_pos = constrain(servo_3_pos, 0, 180);
  servo_3.write(servo_3_pos);

  int flex_5_pos;
  int servo_5_pos;
  flex_5_pos = analogRead(flex_5);
  servo_5_pos = map(flex_5_pos, 770, 950, 0, 180);
  servo_5_pos = constrain(servo_5_pos, 0, 180);
  servo_5.write(servo_5_pos);

  int flex_4_pos;
  int servo_4_pos;
  flex_4_pos = analogRead(flex_4);
  servo_4_pos = map(flex_4_pos, 770, 950, 0, 180);
  servo_4_pos = constrain(servo_4_pos, 0, 180);
  servo_4.write(servo_4_pos);
```

```
 Serial.print("thumb =");
 Serial.println(flex_1_pos);

 Serial.print("index =");
 Serial.println(flex_2_pos);

 Serial.print("middle =");
 Serial.println(flex_3_pos);

 Serial.print("ring =");
 Serial.println(flex_4_pos);

 Serial.print("pinkie =");
 Serial.println(flex_5_pos);

 Serial.println("-------------------");

 delay(1000);


}
```

Ultrasonic and servo motor code:

```
int motorRightA = 8;    //Right Motor-clockwise
int motorRightB = 9;   //Right Motor-anticlockwise
int motorLeftA = 11;   //Left Motor-clockwise
int motorLeftB = 10;   //Left Motor-anticlockwise
int trigPin1 = 12;     // Trig Pin
int echoPin1 = 13;     // Echo Pin
int light = 5;
long duration1;
int distance1;
char bt = 0;            //Bluetooth Control
int trigPin2 = 7;       // Trig Pin
int echoPin2 = 6;       // Echo Pin
long duration2;
int distance2;
int led = 4;
int pushButton = 3;
void setup()
{
 pinMode(motorRightA, OUTPUT);
 pinMode(motorRightB, OUTPUT);
 pinMode(motorRightB, OUTPUT);
 pinMode(motorLeftB, OUTPUT);
 pinMode(trigPin1, OUTPUT);
 pinMode(echoPin1, INPUT);
 pinMode(trigPin2, OUTPUT);
 pinMode(echoPin2, INPUT);
 pinMode(light, OUTPUT);
 pinMode(led, OUTPUT);
 pinMode(pushButton, INPUT_PULLUP);
 Serial.begin(9600);
}
void loop()
{
 //Light On Off
 lightOnOff();
 //Panic Button
 panicSound();

 //  Right
 digitalWrite(trigPin1, LOW);
 delayMicroseconds(2);
 // Sets the trigPin on HIGH state for 10 micro seconds
 digitalWrite(trigPin1, HIGH);
```

```
  delayMicroseconds(1);
  digitalWrite(trigPin1, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration1 = pulseIn(echoPin1, HIGH);
  // Calculating the distance
  distance1 = duration1 * 0.034 / 2;
  // Prints the distance on the Serial Monitor
  Serial.print("Distance1: ");
  Serial.println(distance1);

  // Left
  digitalWrite(trigPin2, LOW);
  delayMicroseconds(2);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin2, HIGH);
  delayMicroseconds(1);
  digitalWrite(trigPin2, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration2 = pulseIn(echoPin2, HIGH);
  // Calculating the distance
  distance2 = duration2 * 0.034 / 2;
  // Prints the distance on the Serial Monitor
  Serial.print("Distance2: ");
  Serial.println(distance2);

  if (distance1 <= 20 || distance2 <= 20) {
    //Stop Wheel Chair
    digitalWrite(motorRightA, LOW);
    digitalWrite(motorRightB, LOW);
    digitalWrite(motorLeftA, LOW);
    digitalWrite(motorLeftB, LOW);
    control();
  }
  else {
    control(); // Call All the Control
  }
}

// All the Controls of the Wheel Chair
void control() {
  if (Serial.available() > 0)
  {
    bt = Serial.read();
    if (bt == 'F')       //move forwards
    {
      digitalWrite(motorRightA, HIGH);
      digitalWrite(motorLeftA,  HIGH);
    }
    else if (bt == 'B')       //move backwards
    {
      digitalWrite(motorRightB, HIGH);
      digitalWrite(motorLeftB, HIGH);
    }
    else if (bt == 'S')     //stop!
    {
      digitalWrite(motorRightA, LOW);
      digitalWrite(motorRightB, LOW);
      digitalWrite(motorLeftA, LOW);
      digitalWrite(motorLeftB, LOW);
    }
    else if (bt == 'R')    //right
    {
      digitalWrite(motorRightA, LOW);
      digitalWrite(motorRightB, LOW);
      digitalWrite(motorLeftA, HIGH);
      digitalWrite(motorLeftB, LOW);
    }
```

```
    else if (bt == 'L')     //left
    {
      digitalWrite(motorRightA, HIGH);
      digitalWrite(motorRightB, LOW);
      digitalWrite(motorLeftA, LOW);
      digitalWrite(motorLeftB, LOW);
    }
    else if (bt == 'I')    //forward right
    {
      digitalWrite(motorRightA, HIGH);
      digitalWrite(motorRightB, LOW);
      digitalWrite(motorLeftA, LOW);
      digitalWrite(motorLeftB, HIGH);
    }
    else if (bt == 'G')    //forward left
    {
      digitalWrite(motorRightA, LOW);
      digitalWrite(motorRightB, HIGH);
      digitalWrite(motorLeftA, HIGH);
      digitalWrite(motorLeftB, LOW);
    }
  }
}

void lightOnOff() {

  if (bt == 'O') {
    digitalWrite(light, HIGH);
  }
  else if (bt == 'o') {
    digitalWrite(light, LOW);
  }
}
void panicSound() {
//  int val = digitalRead(pushButton);
  if  ( distance1 < 60) {
    digitalWrite(led, LOW);
  } else {
    digitalWrite(led, HIGH);
  }
}
```