

HEALTH MONITORING AND ALERTING SMART WHEEL CHAIR FOR PHYSICALLY DISABLED

*A project report submitted in partial fulfilment of the requirements for the
award of the degree of*

**BACHELOR OF TECHNOLOGY IN
ELECTRONICS AND COMMUNICATION ENGINEERING**

Submitted by

P. DIVYA SRI (319126512171)

A. PAVAN SADHGUNA (319126512131)

S. AVINASH (319126512182)

RAVI KUMAR KANURI(319126512173)

Under the guidance of

Mr. N.RAM KUMAR

ASSISTANT PROFESSOR



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING
ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES
(UGC AUTONOMOUS)**
*(Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC With 'A'
Grade)*
Sangivalasa, bheemili mandal, Visakhapatnam dist. (A.P)
2022-2023

DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING
ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES
(UGC AUTONOMOUS)
(Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA &
NAAC With 'A' Grade)
Sangivalasa, bheemilimandal, visakhapatnam dist. (A.P)



This is to certify that the project report entitled “ Health Monitoring and Alerting Smart Wheelchair for Physically Disabled” submitted by PUNUMAJJI DIVYA SRI (319126512171), ATTADA PAVAN SADHGUNA (319126512131), SIRUKURI AVINASH (319126512182), RAVI KUMAR KANURI (319126512173) in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Electronics and Communication Engineering of Andhra University, Visakhapatnam is a record of bonafide work carried out under my guidance and supervision during the year 2022-2023.


Project Guide

Mr. N.Ram Kumar

Assistant Professor

Department of E.C.E

ANITS

Assistant Professor
Department of E.C.E

Anil Neerukonda

Institute of Technology & Sciences
Sangivalasa, Visakhapatnam-531 162


Head of the Department

Dr. B.Jagadeesh

Assistant professor & HOD

Department of E.C.E

ANITS

Head of the Department
Department of E C E

Anil Neerukonda Institute of Technology & Sciences
Sangivalasa - 531 162

ACKNOWLEDGMENT

We would like to express our deep gratitude to our project guide **Mr. N.Ram Kumar** Assistant Professor, Department of Electronics and Communication Engineering, ANITS, for his guidance with unsurpassed knowledge and immense encouragement. We are grateful to **Dr. B.Jagadeesh**, Head of the Department, Electronics and Communication Engineering, for providing us with the required facilities for the completion of the project work.

We are very much thankful to the **Principal and Management, ANITS, Sangivalasa**, for their encouragement and cooperation to carry out this work.

We express our thanks to all **teaching faculty** of Department of ECE, whose suggestions during reviews helped us in accomplishment of our project. We would like to thank all **non-teaching staff** of the Department of ECE, ANITS for providing great assistance in accomplishment of our project.

We would like to thank our parents, friends, and classmates for their encouragement throughout our project period. At last, but not the least, we thank everyone for supporting us directly or indirectly in completing this project successfully.

PROJECT STUDENTS

P. Divya Sri (318126512171)

A. Pavan Sadhguna (319126512131)

S. Avinash (319126512182)

Ravi Kumar Kanuri (319126512173)

ABSTRACT

This project is intended to help the elderly and those with physical limitations who are unable to move on their own due to frailty. The goal of this work is to design and implementation of health monitoring and alerting smart wheel chair for physically disabled people. The designed wheelchair can be operated by using voice commands via the provided input. The wheelchair that was designed includes an automatic obstacle detection system that uses ultrasonic sensors. If the person is in fire accident the sensor detects the fire using flame sensor module and send notifications to the guardians. This wheel chair will also monitor the health condition of the person like heart rate, internal body temperature and blood oxygen level. For the purpose of determining heart rate and blood oxygen levels, we incorporate a MAX30100 pulse oximeter heart rate sensor. The interior body temperature can be determined with an LM35 sensor. If the pulse and temperature are not in normal conditions it will send notifications and alerts their caretakers using blynk app. The person in the wheel chair can automatically control the room appliances by the commands provided by the user through voice recognition module by transmitter and receiver. Therefore, the evolved health monitoring and alerting smart wheelchair can help physical disabled and older people.

Keywords— LM35 sensor, Flame sensor, Ultrasonic sensor, Voice recognition module, Pulse Oximeter and Heartrate sensor

CONTENTS

LIST OF FIGURES	vii
LIST OF TABLES	viii
CHAPTER 1 INTRODUCTION	1
1.1 Introduction	2
1.2 Problem Identification	4
1.3 Objectives of the project	5
1.4 Literature review	5
CHAPTER 2 HARDWARE DESCRIPTION	7
2.11 Arduino board	8
2.12 Node MCU	9
2.13 Voice Recognition Module	10
2.14 Ultra Sonic Sensor	12
2.15 Bread board	13
2.16 LM35 Sensor	14
2.17 Flame Sensor	15
2.18 Pulse Oximeter Sensor (MAX30100)	16
2.19 Motor Driver(L298N)	17
2.20 DC Motors	18
2.21 RF Transmitter and Receiver	19
2.22 Jumper Wires	20

CHAPTER 3	METHODOLOGY	22
3.1	Block Diagram And Description	23
3.2	Flow Chart For Voice Control	26
3.3	Flow Chart For Health Monitoring	27
CHAPTER 4	SOFTWARE DESCRIPTION	29
4.1	Arduino IDE	30
4.2	Blynk App	33
4.3	Blynk Webpage Designing	34
CHAPTER 5	RESULT	38
5.1	Result	39
CHAPTER 6	CONCLUSION AND FUTURE SCOPE	42
6.1	Conclusion	43
6.2	Future Scope	43
REFERENCES		44
APPENDIX(CODE)		47
CONFERENCE		61

LIST OF FIGURES

Figure no.	Title	Page no.
Fig 2.11	Arduino Uno	8
Fig 2.12	Node MCU	10
Fig 2.13	Voice Recognition Module	11
Fig 2.14	Ultra Sonic Sensor	12
Fig 2.15	Bread Board	13
Fig 2.16	LM35 sensor	14
Fig 2.17	Flame Sensor	15
Fig 2.18	Max30100 Sensor	16
Fig 2.19	Motor Driver(L298N)	17
Fig 2.20	DC Motor	19
Fig 2.21	RF Transmitter and Receiver	20
Fig 2.22	Jumper Wires	21
Fig 5.1	Wheelchair prototype	39
Fig 5.2	Heartrate and spO2 readings	40
Fig 5.3	Wheelchair with labelled	40
Fig 5.4	LM35 & flame sensor readings in blynk app	41
Fig 5.5	Notification of flame sensor	41

LIST OF TABLES

Table no.	Title	Page No.
1	Commands used and directions for each motor either on clockwise or anticlockwise	40

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

A wheelchair is required by around 111 million individuals worldwide, according to a World Health Organization (WHO) estimate. Being disabled is a natural part of being human. About 2.21% (or 26.8 million) of Indians have a disability of some type. There are 1.18 crore disabled women and 1.5 crore disabled men in India, out of a total population of 62.32 crore and 58.78 crore, respectively.

In today's society, people are expected to be self-sufficient despite any inherent physical or mental difficulties. People with physical disabilities must rely on others to take care of their fundamental requirements. The likelihood of their venturing outside and interacting with the outside world is quite minimal unless they are given with contemporary movement tools like a wheel chair. Wheelchairs can be propelled manually or electrically, respectively. Only those with lower-limb disability have the first choice, and long-term use comes with additional health hazards. Moreover, manual-driven wheelchair efficiency is only 10% to 20%.

Today Due to recent improvements in the industrial population, wheelchairs are a crucial piece of equipment in the biomedical sector. The need for the elderly and physically challenged is constantly growing.

The information above suggests that people with impairments have a big need for wheelchairs. The need for wheelchairs is essential for their comfort and to make life simpler. We are recommending a smart wheelchair that monitors health and sends alerts in light of the issues experienced by people with disabilities.

This project's goal is to use voice control to steer a wheelchair. It allows people who are unable to move their hands or legs to get around on their own by using a voice recognition system. This project aims to use voice control to steer a wheelchair. It makes it feasible for people who are unable to move their hands or legs to move around on their own by employing voice recognition software coupled with motors to facilitate mobility as much as possible.

In contrast to manual wheelchairs, this one can be driven by the user via voice commands. Moreover, voice-controlled wheelchairs operate far more quickly than manual and joystick wheelchairs. Those who are unable to move their hands can utilise this wheelchair. The voice recognition system processes the received voice, extracting the voice command's feature, and compares it to a sample already present in the database. The module transmits control messages to the microcontroller after identifying the voice. Wheelchairs can move in several directions, including forward, back, left, and right.

Also, the wheel chair will be embedded with flame sensor which will detect fire accidents. The wheel chair also has health monitoring system which monitors the blood oxygen levels, heart rate, temperature. And the person in wheel chair can also control the appliances which are connected by giving voice commands to voice recognition module. Even it is a voice-controlled wheelchair when something suddenly comes across the wheelchair path the person may not speak suddenly so we are using obstacle detection with the help of ultrasonic sensor.

Until the user is capable of doing the task, the obstacle sensor will aid the rider navigate the wheelchair by avoiding object which are in the way of wheelchair. The speech command we give the voice recognition model is person-dependent, much like the voice command itself.

The system consists of transmitter and receiver. Function keys helps to store the voice commands in the data base. This wheelchair will automatically recognize the object which is in its way and it will automatically stop.

One of the drawbacks of the smart wheel chairs is they have highest price when comparing with the manual wheel chair.so we are commercializing the product with good price estimation.

1.2 PROBLEM IDENTIFICATION

For physically handicapped people, independent mobility is essential for the development of their social, cognitive, linguistic, and physical abilities. The exorbitant cost of electric wheelchairs. So, the goal of our project is to develop an advanced control system for an electric driven wheelchair.

The wheelchairs on the market are too pricey and out of the grasp of families from lower- and middle-class backgrounds. We are developing a wheelchair that will be inexpensive and versatile. This project intends to improve the lives of old and disabled persons who are unable to move properly and make life easier for them.

We concentrate on helping those who are incapable of moving even their hands. We therefore suggest a voice-activated smart wheelchair that can not only move in the direction that is commanded by the user but also avoid impediments in the way of the intended path.

It is hard for the disabled persons to control the electrical appliances as their hands are not functional, with the help of this smart wheel chair we are enabling a function where the disabled person can also control the appliances just with the help of voice.

We have incorporated a flame sensor so that it can detect whenever that person is caught under the circumstances of a fire since it can be tough for that person to move in the case of emergency fire mishaps. Along with that, we are also incorporating a healthcare system that monitors a person's body temperature, heart rate, and blood oxygen level while providing a daily update to the Guardian.

1.3 OBJECTIVES OF THE PROJECT

1. To develop a voice-controlled wheel chair system by using visual basic for wheel chair control.
- 2.To implement a wheel chair whose movement is controlled by the voice commands.
- 3.Providing the service by the help of this wheelchair for people who are unable to control their hands as well as legs.
- 4.Providing the facilities to control the room appliances by the disabled person.
- 5.This wheel chair will be provided with health monitoring system.
- 6.We are placing flame sensor so this wheel chair indicates when the person stuck in the fire accidents.

1.4 LITERATURE REVIEW

The aspects of this type of problem are incredibly complicated and change with time. We reviewed several related articles to find the most efficient and effective solution. The key things in study are summarized below.

Design and development of a smart wheelchair for the disabled people. It was proposed by Foez Ahmed et al [8]. This study presents a wheelchair that incorporates primarily two types of control systems: control by thumb and control by gesture. The user can select their preferred control system from the user menu. They can navigate the menu with their thumbs, but it is inaccessible to people who have lost their arms.

Gesture-based Smart Wheelchair for Assisting Physically Challenged People. It was proposed by Sarnali Basak et al. [2]. The research-based work presented in this project is a smart wheelchair system controlled by the gestures of user's hand that can acknowledge 3D movement of the hand recognition technology or movement of the hand using gesture.

Other features which will include the ability to communicate with anyone whose contact information is stored in the system, such as family, police, or healthcare facilities, by using the gesture sensor.

Design of An Intelligent Wheelchair for Handicap People Conducting by Body Movement. It was proposed by Mubiru Alam Sajid et al [6]. This wheelchair operates by detecting hand movement and helping the paralyzed patient with assistance. It has a feature that detects obstacles and avoids collisions. This wheelchair is based on gesture of user's hand and an android wheelchair control that any disabled person can use.

Smart Wheelchair- An Implementation of Voice and Android Controlled System. It was proposed by Muhammad Saad Amin et al [7]. As a result, any hand amputee can operate this smart wheel solely with their voice, with no physical human effort required for movement. Android application is also available for more advanced controls from a smartphone. It cannot monitor health conditions and therefore cannot control home appliances.

Next generation autonomous wheelchair control. It was proposed by John Benson et al [3]. Here they used joystick control. As joystick-controlled wheelchairs are widely used around the world, the problem is that handicapped people who have difficulty with finger movement are unable to control the joystick because it requires hand control.

Smart Wheel Chair. It was proposed by Sarmad Hameed et al [1]. In this model, the individual can operate the wheel chair with hand gestures via smartphone. However, if indeed the person is unable to move their hand, this method is ineffective.

CHAPTER 2
HARDWARE DESCRIPTION

2.1 ARDUINO

Microcontroller board, Arduino UNO with ATmega328P. It contains 6 analogue inputs, a 16 MHz ceramic resonator, 14 digital I/O pins (six of which can be used as PWM outputs), a USB port, a power jack, an ICSP header, and a reset button. It has every part required to support the microcontroller, so all you need to do to get started is plug it into a computer through USB, an AC-to-DC adapter, or a battery. Without worrying about making a mistake, you can experiment with your UNO; in the worst case, you can start over by replacing the chip for a few dollars.

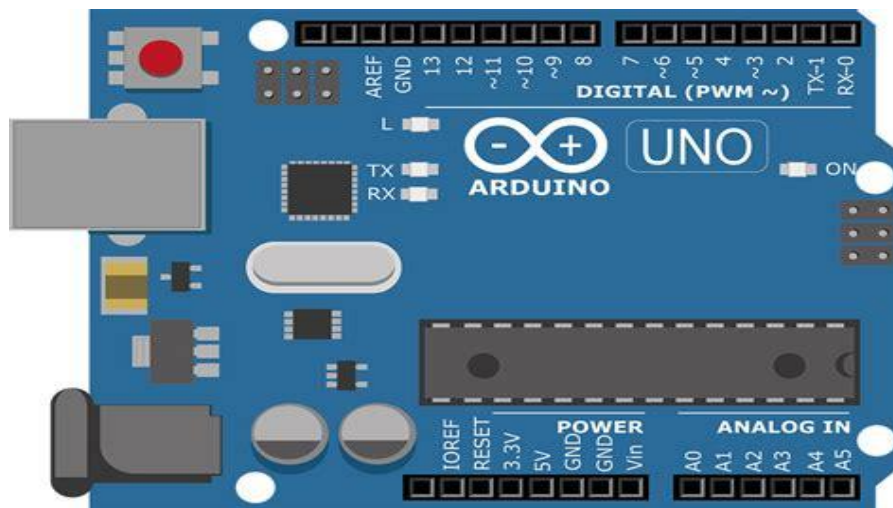


Fig 2.1 Arduino UNO

Arduino pins, inputs and outputs

- Analog Reference pin
- Digital Ground pin

- Digital Pins 2-13
- Reset Button - S1
- Analog in Pins 0-5
- Power and Ground Pins
- External Power Supply In (9-12VDC)
- USB (used for uploading sketches to the board and for serial communication between the board and the computer; can be used to power the board)

We used the following materials for our project:

1. The Arduino Uno.
2. A USB cable for programming.
3. External power supply or 9V battery (for stand-alone operation). Rather than being powered by a USB connection to a computer, the board is powered by a battery.
4. External circuits are built on a breadboard, and connections are made with solid wire.

2.2 NODE MCU

Based on the ESP8266, a low-cost System-on-a-Chip (SoC), the Node MCU (Node Micro Controller Unit) offers a completely accessible software and hardware development environment. All of a computer's fundamental parts are present in the Espressif Systems ESP8266, including the CPU, RAM, networking (WIFI), as well as a contemporary operating system and SDK. It is therefore a great option for all types of Internet of Things (IoT) projects.

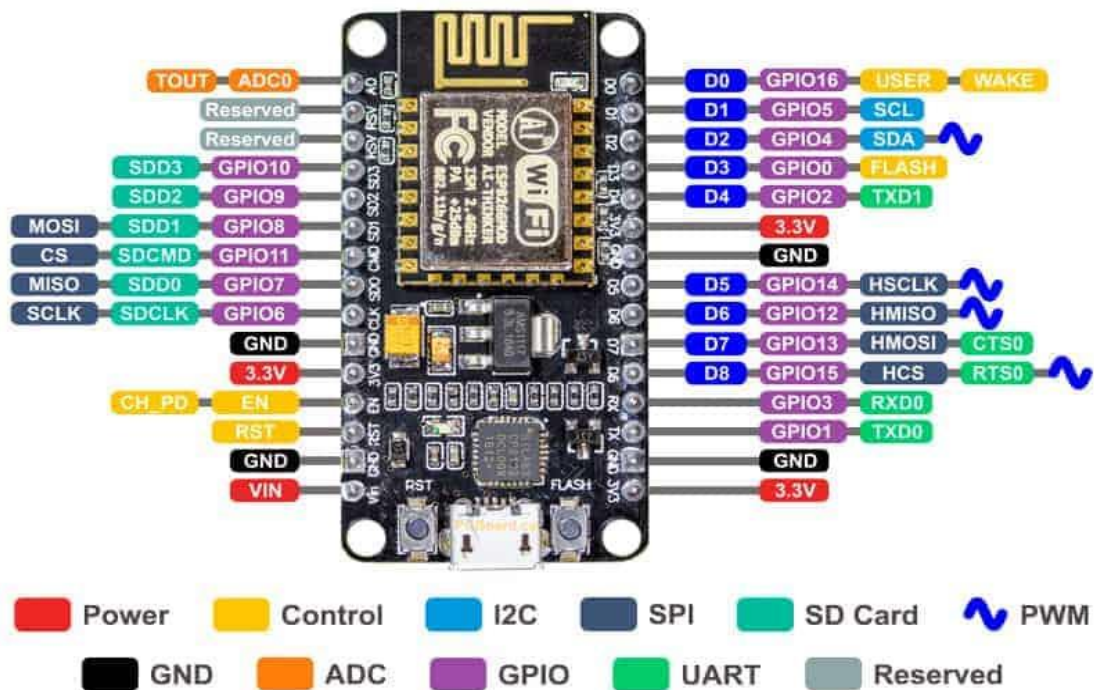


Fig 2.2 NODE MCU

2.3 VOICE RECOGNITION MODULE

This Voice Module v3 is a speaker-dependent voice recognition module. It can accommodate up to 80 voice commands altogether. There can be a maximum of seven voice commands active at once. Any noise can be made to serve as a command. Before allowing the module to identify any voice commands, users must train it. Both the Serial Port and the General Input Pins on this board have complete functionality. The board's General Output Pins could produce a variety of waves while the corresponding voice command was recognised.

On V3, voice commands are organised into one sizable group that resembles a library. Any of the seven voice commands available in the library might be loaded into recognizer. This indicates that seven commands are in use simultaneously.



Fig 2.3 Voice recognition Module

Parameters:

- Voltage: 4.5-5.5V
- Current: <40mA
- Digital Interface: 5V TTL level for UART interface and GPIO
- Analog Interface: 3.5mm mono-channel microphone connector + microphone pin interface
- Size: 31mm x 50mm
- Recognition accuracy: 99% (under ideal environment)

2.4 ULTRASONIC SENSOR

As their name suggests, ultrasonic sensors measure distance using ultrasonic waves. The sensor head emits an ultrasonic wave and then picks up the reflected wave from the target. Ultrasonic sensors determine the distance to the target by measuring the interval between the emission and reception of sound waves.

The following formula can be used to calculate the distance:

$$\text{Distance } L = 1/2 \times T \times C$$

where L stands for length, T for the interval between emission and reception, and C for the sound speed. T is the amount of time needed to travel the distance and return, hence the value is split by half.

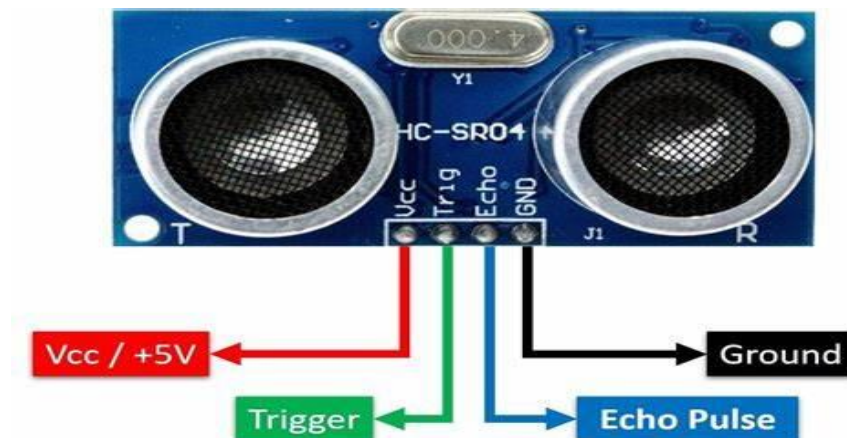


Fig 2.4 Ultrasonic Sensor

Because of their numerous advantages, ultrasonic sensors have established themselves as a go-to proximity sensor solution.

2.5 BREADBOARD

Undoubtedly, a breadboard may be used to prototype or construct circuits. It makes it possible to build circuits without the use of solder by allowing you to arrange parts and connections on the board. Your connections are handled by the holes on the breadboard, which physically hold onto the components or wires you insert into them and electrically connect them within the board. Simple circuits can be quickly prototyped thanks to their speed and simplicity. For more complicated and high frequency circuits, breadboarding is less suitable. Contrary to protoboard (perfboard) and PCB (printed circuit board) circuits, which are better suited for long-term usage, breadboard circuits don't require soldering and are less expensive to develop and produce.

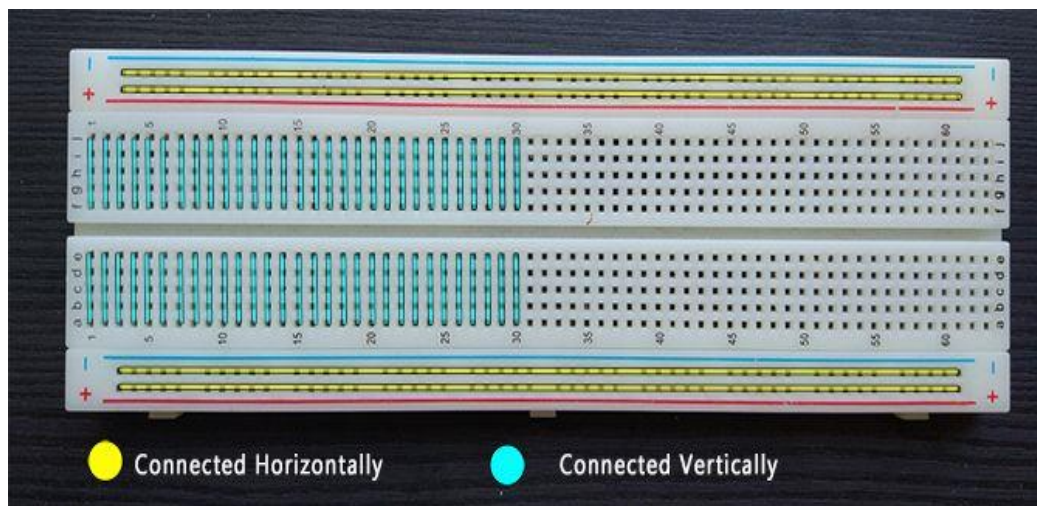


Fig 2.5 Bread Board

Usually, the board is connected to a power supply using the two bigger wires that run along either side. They are frequently referred to as power rails.

Circuit components are made from of smaller segments of wire that run perpendicularly across the board

2.6 LM35 SENSOR

A temperature sensor called the LM35 produces an analogue output proportional to the ambient temperature. It is simple to convert the output voltage to a temperature value in degrees Celsius. Compared to thermistors, the lm35 has the benefit of not requiring external calibration. Additionally, the covering prevents it from overheating.

The LM35 can monitor temperatures between -55 and 150 degrees Celsius. When used at the right temperature and humidity, the accuracy level is very high. The process of converting the output voltage to degrees Celsius is similarly easy and clear.

The input voltage range for the LM35 is +4 to 30 volts. It uses about 60 microamperes of current total.

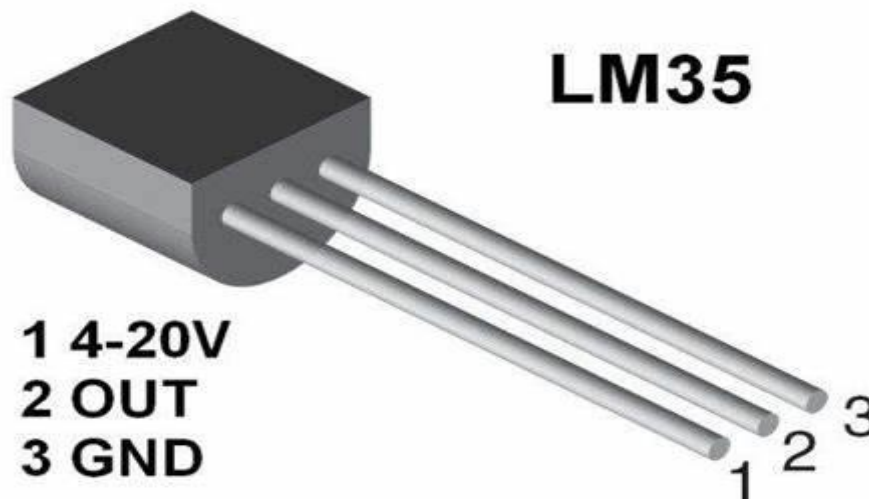


Fig 2.6 LM35 sensor

2.7 FLAME SENSOR

The most sensitive sensor to ambient light is a flame sensor. As a result, this sensor module is used in fire alarms. Flames and light from the light source with wavelengths between 760 and 1100 nm are picked up by this sensor. High temperatures can easily harm this sensor. This sensor can therefore be positioned at a particular distance from the flame. Flame detection is possible at a distance of 100 cm with a detection angle of 60°. This sensor can produce either an analogue or digital output. These sensors serve as a flame alert in firefighting robots.

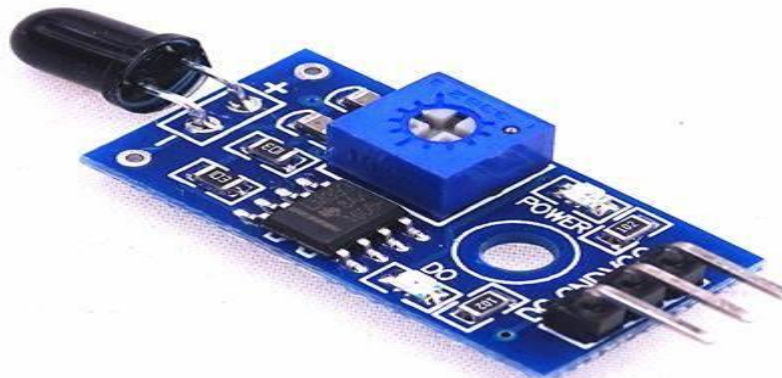


Fig 2.7 Flame sensor

The pin arrangement for this sensor is displayed below. These are the four pins that are included with it. The pins are connected to a microcontroller when this module is used with one.

- Pin1 (VCC pin): Voltage supply ranges from 3.3V to 5.3V
- Pin2 (GND): This is a ground pin
- Pin3 (AOUT): This is an analog output pin (MCU.IO)
- Pin4 (DOUT): This is a digital output pin (MCU.IO)

2.8 PULSE OXIMETER SENSOR(MAX30100)

The MAX30100 is a flexible sensor that can be used in a variety of applications. It combines a pulse oximeter and a heart rate monitor. In order to measure pulse oximetry and identify heart rate, the sensor has two Light Emitting Diodes, a photodetector, and a number of low noise signal processing components.

MAX30100 features:

Here are some of the MAX30100 Heart Rate Oxygen Pulse Sensor's features and specifications.

1. Operating Voltage - 1.8V to 3.3V
2. Input Current - 20mA
3. Integrated Ambient Light Cancellation
4. High Sample Rate Capability
5. Fast Data Output Capability

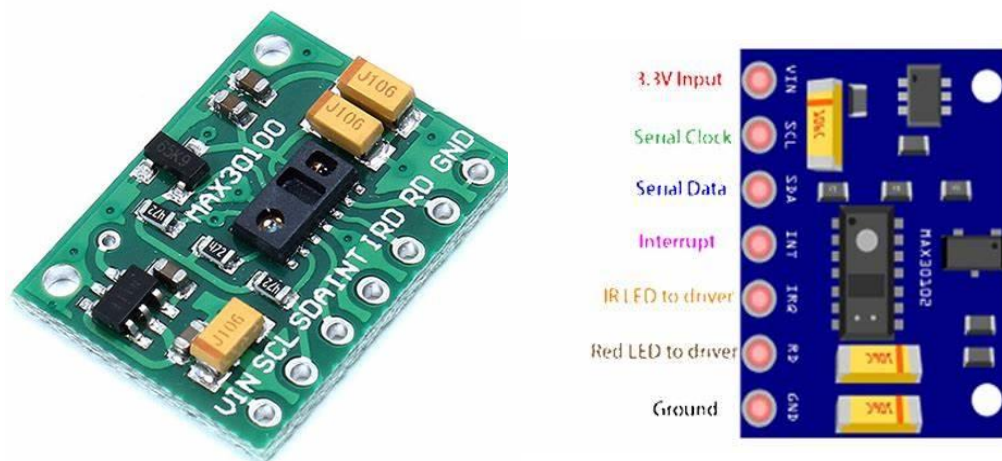


Fig 2.8 Max 30100

2.9 MOTOR DRIVER(L298N)

The mechanism that links the motor to the microcontroller is known as a motor driver. This is because the voltage levels at which the motor and microprocessor work differ. The microcontroller will use less power than the engine.

When connecting two devices that run at various current levels to a power supply voltage, a motor driver module is necessary. A motor serves as a third component in this scenario, ramping up or down the voltage supply.

Integrated circuits (ICs) make up the vast majority of motor drivers that are commercially available. Driver motors come in a variety of forms, each with a unique set of properties. The motor controller and motor driver ICs are connected by a H bridge circuit.

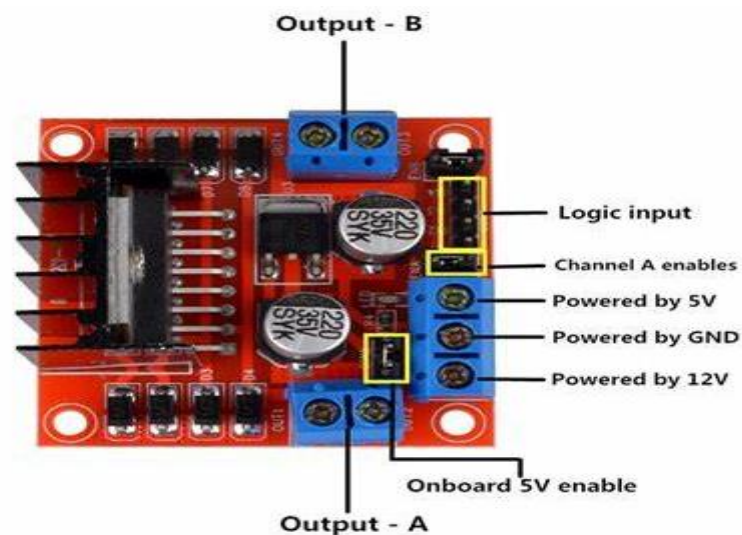


Fig 2.9 Motor driver

Input pins:

These are the input pins for Motor A: IN1, IN2. These are used to regulate the rotational direction of Motor A. Motor A starts to rotate in a particular direction when one of them is HIGH and the other is LOW. If both inputs appear to be HIGH or LOW, Motor A will stop

These are the input pins for Motor B: IN3, IN4. These are utilised to regulate the rotating axis of Motor A. Motor A starts to rotate in a particular direction when one of them is HIGH and the other is LOW. If both inputs are either HIGH or LOW, Motor A will stop.

Controlling the speed:

The L298n motor driver module uses the pulse width modulation technology to regulate the rotational speed of a DC motor. A DC motor's speed can be managed by adjusting its input voltage.

By sending a series of ON-OFF pulses, the Pulse Width Modulation (PWM) technique calculates the median value supplied by the input voltage. The Duty Cycle, often known as the relationship between average voltage and pulse width, is used to describe this relationship.

The average voltage is applied to the DC motor (High Speed) when the duty cycle is high, and the average voltage is applied to the DC motor (Low Speed) when the duty cycle is low.

2.10 DC MOTOR

A direct current motor, commonly referred to as a DC motor, is an electrical apparatus that attempts to create a magnetic field generated by direct current in order to convert electrical energy into mechanical energy. When a direct current motor is turned on, a magnetic field is produced in the stator. The magnetic field pulls and repels magnets on the rotor, which rotates as a result. The commutator, which is linked to brushes linked to the power source, provides current to the motor's wire windings to maintain the rotor's continuous rotation.



Fig 2.10 DC Motor

2.11 RF TRANSMITTER AND RECEIVER

An RF transmitter and an RF receiver are combined in this RF module. The transmitter/receiver (Tx/Rx) pair has a 433 MHz frequency.

The RF transmitter uses its RF antenna to wirelessly transmit serial data after receiving it. Between 1 and 10 Kbps are the available transmission rates. The RF receiver, which utilises the same frequency as the transmitter, receives the transmitted data.

Since RF modules have more uses than IR, we employ them for data transmission and reception. In order to send and receive data, the RF transceiver module will always operate in pairs, requiring a transmitter and a receiver. Data can only be sent from one end to the other, not the other way around, because a transmitter can only send data and a receiver can only receive it.

Features of RF Module:

- The receiver frequency is 433MHz
- The typical frequency is 105 Dbm
- The supply current is 3.5 mA.
- The operating voltage of the receiver is 5V
- The transmitter frequency range is 433.92MHz
- The supply voltage of the transmitter is between 3V and 6V
- The output power of the transmitter is between 4Dbm and 12Dbm.

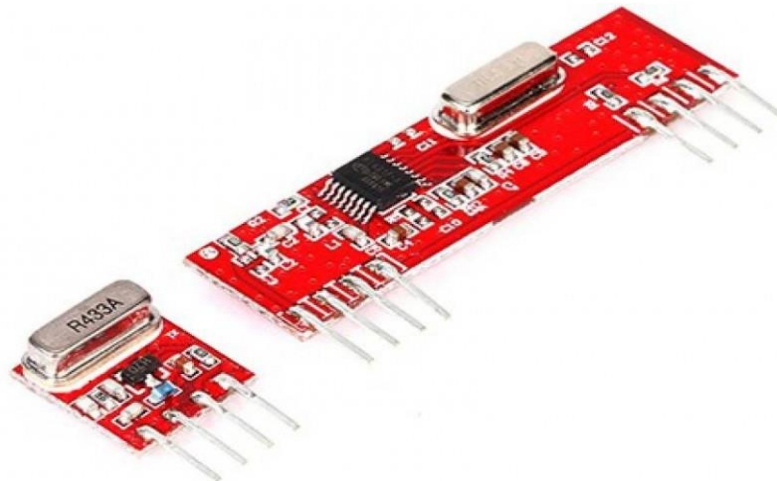


Fig 2.11 RF Transmitter and Receiver

2.12 JUMPER WIRES

In order to connect two sites without soldering, jumper wires are simple wires with connector pins on each end.

When using breadboards and other prototype tools, jumper wires are frequently utilised to make changing a circuit a breeze.

Typically, there are three types of jumper wires: male-to-male, male-to-female, and female-to-female. The wire's end point is where the difference exists.

Female ends are used to plug into items while male ends have a protruding pin and can plug into objects. Jumper wires that connect two males are the most prevalent and often utilised. Two breadboard ports must be connected using a male-to-male cable.



Fig 2.12 Jumper Wires

CHAPTER 3
METHODOLOGY

3.1 BLOCK DIAGRAM AND DESCRIPTION

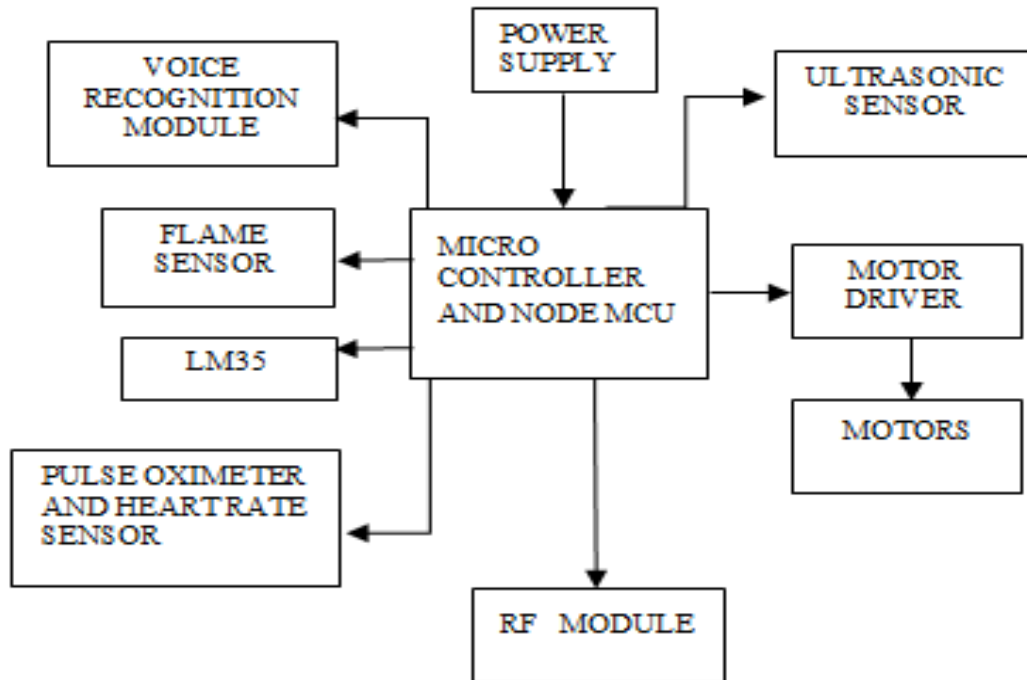


Fig 3.1 Flow chart

Methodology refers to the few steps included in the successful implementation of our project. The aim of our project is alerting the care taker in case of any misbalance in the health like body temperature, blood oxygen, heart rate of person in wheel chair. If he is stuck in any fire accident immediately sending an alert message to the nearby fire station or care taker. Voice commands are used for controlling the room's appliances.

Primary function of the project is to move the wheel chair using voice commands. These are the four main functionalities of our project. To attain all these,

we will be using some sensors and modules which you can see in the figure. Voice recognition module receives voice commands from the person in a wheelchair and forwards them to the microcontroller. Before using this voice recognition module, we have to train this with certain commands, other than these commands the module does not take other commands.

A wheel chair's surroundings are detected using an ultrasonic sensor. Without the user's command, the wheelchair will stop travelling if it detects any objects or obstructions while it is in motion in order to prevent a collision. The temperature of the individual seated in the wheelchair is measured using an LM35 sensor. With the aid of the Blynk app, it will immediately alert the guardian if the temperature rises above a certain level.

With voice commands he can control the room appliances like fan, light etc. Voice recognition module receives the commands and forwards those to microcontroller, it validates the commands and through RF Transmitter module it address those commands to RF Receiver module which is connected to the appliances. It receives the commands and act according to those commands.

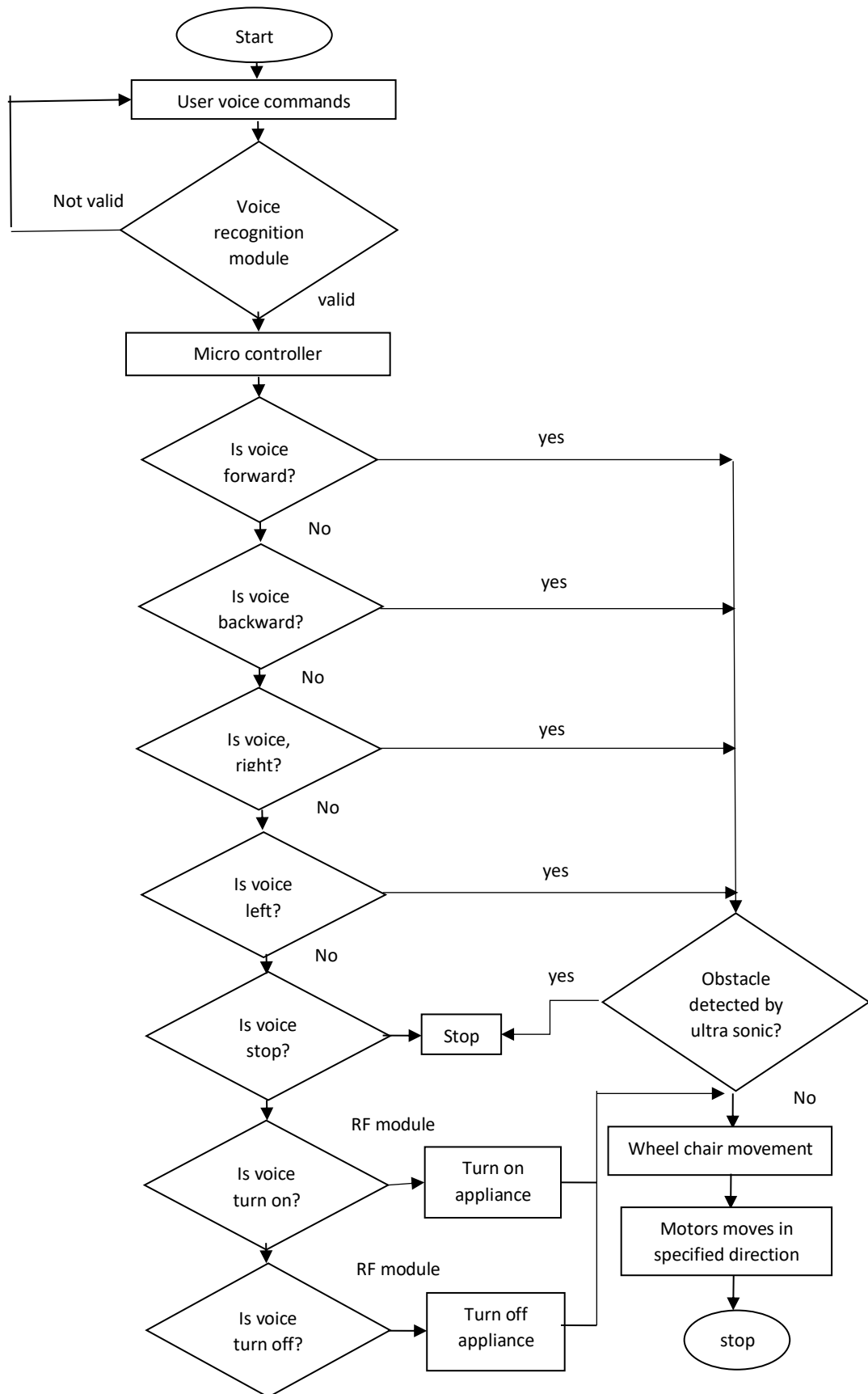
Motor driver is bridge of the motors which are connected to the wheel chair to control speed and direction. According to the user inputs the motors will rotate.

- Both motors will rotate in forward direction - **Forward movement**
- Both motors will rotate in backward direction - **Backward movement**
- Right motor will rotate in forward direction and left motor will rotate in backward direction - **Left movement**
- Left motor will rotate in forward direction and right motor will rotate in forward direction - **Right movement**

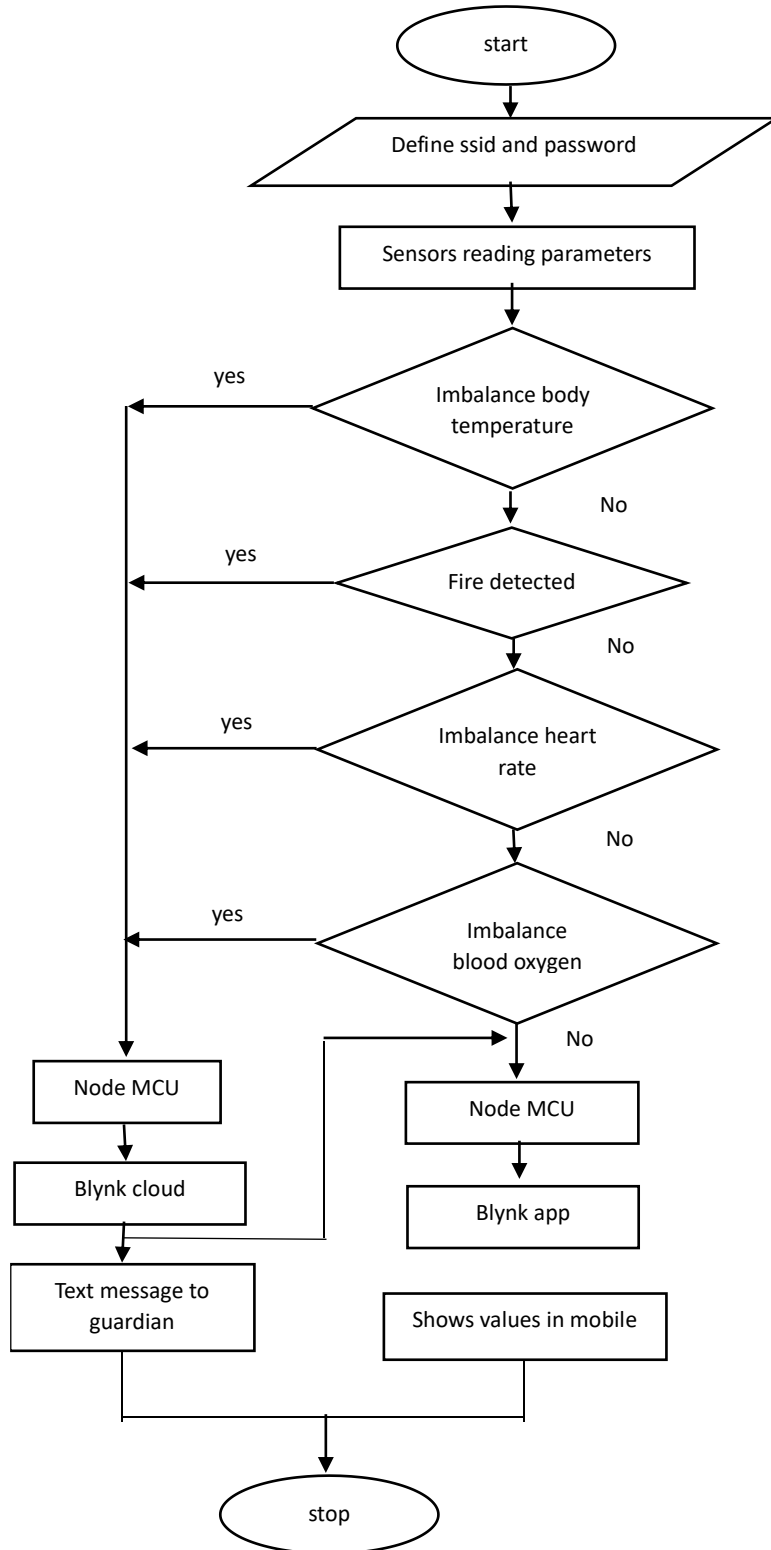
The L298n motor driver module uses the pulse width modulation method to regulate the DC motor rotational speed. Using this method, a DC motor's speed can be adjusted by changing the i/p voltage. By transmitting a series of ON-OFF pulses, the Pulse Width Modulation technology modifies the input voltage's average value. The Duty Cycle, also known as the width of the pulses, determines how much voltage is present on average. The average voltage is applied to the DC motor (High Speed) when the duty cycle is higher, and (Low Speed) when the duty cycle is lower.

Flame sensors are utilised to find the flames. In case of any fire accidents if it detects those flames, it automatically alerts the care taker or nearby fire station through a message. The user's body temperature is determined using the LM35 temperature sensor., and the pulse oximeter and heart rate sensor are used to measure the user's blood oxygen and heart rate. If any of the parameters increases their threshold value it alerts the care taker or nearby hospital with a message.

3.2 FLOW CHART FOR VOICE CONTROL



3.3 FLOW CHART FOR HEALTH MONITORING



Flow chart for voice control description:

- Step 1 : Start the device by powering it.
- Step 2 : Search for valid command from User.
- Step 3 : If commands are valid, it will inputs to microcontroller. Else it will search for right command.
- Step 4 : If the command is “Forward”, It will move forward.
- Step 5 : If the command is “Backward”, It will move backward.
- Step 6 : If the command is “Right”, It will move right.
- Step 7 : If the command is “Left”, It will move left.
- Step 8 : If the command is “stop”, It will stop.
- Step 9 : If Ultrasonic sensor detects any obstacle, it will stop.
- Step 10 : If the command is “Turn On”, it will turn on the room appliance.
- Step 11 : If the command is “Turn Off”, it will turn off the room appliance.
- Step 12 : Step 2 to Step 11 will repeat.

Flow chart for health monitoring description:

- Step 1 : Start the device by powering it.
- Step 2 : Sensor will read the body temperature, and update the values in mobile application. In case of any abnormalities, it will alert the attendant.
- Step 3 : In case of fire accident, sensor will detect and alert the attendant.
- Step 4 : Sensor will read the heart rate, and update the values in mobile application. In case of any abnormalities, it will alert the attendant.
- Step 5 : Sensor will read the blood oxygen level, , and update the values in mobile application. In case of any abnormalities, it will alert the attendant.
- Step 6 : Step 2 to Step 5 will repeat.

CHAPTER 4

SOFTWARE DESCRIPTION

4.1 ARDUINO IDE

The Arduino IDE, an open-source programme created by Arduino.cc, is a recognised Arduino software that makes code compilation simple enough that even a non-technical layperson may start learning. It operates on the Java Platform, which has built-in debugging, editing, and compiling capabilities, and is compatible with all operating systems, including MAC, Windows, and Linux.

The Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro, and many other modules are available. Each has a microcontroller on the board that accepts information in the form of code and has been programmed. Eventually, the primary code, also referred to as a sketch, created on the IDE platform will produce a Hex File, which is then transferred and uploaded into the board's controller.

The IDE environment is primarily comprised of two basic components: the Editor and the Compiler, the previous head of which is used for writing the required code while the latter is responsible for compiling and uploading the code into the appropriate Arduino Module. While not connected to the internet, the Arduino Software (IDE) makes it simple to write code and upload it to the board. It is recommended for users with slow or no internet access. This software works with any Arduino board.

The Arduino IDE is currently available in two versions: IDE 1.x.x and IDE 2.x. The IDE 2.x is a significant improvement over the IDE 1.x.x in terms of speed and power. It includes advanced coding and debugging features, as well as a more modern editor and a more responsive interface.

The steps below will walk you through using the offline IDE (IDE 1.x.x or IDE 2.x):

1. Get the Arduino Software Development Kit (IDE) and install it:

- Arduino IDE 1.x.x (Mac OS, Windows, Linux, Portable IDE for Windows and Linux, chromeOS).
- Arduino IDE version 2.x

2. Connect your Arduino board to your device using the Arduino IDE version 2.x.

3. Open Arduino IDE (IDE).

The Arduino (IDE)Integrated Development Environment connects to the Arduino boards and allows you to communicate with them. Arduino Software (IDE) creates sketches which are programmes. These drawings are made in a text editor and saved as files with the extension. ino.

Making use of the offline IDE 1.x.x:

The editor is divided into four major sections:

1. A toolbar containing common function buttons as well as a series of menus. The toolbar buttons can be used to create sketches, open sketches, save sketches, verify programmes and upload programmes and launch the serial monitor.
2. The message area displays errors and provides feedback, during saving and exporting
3. The text editor in which your code will be written.
4. The text console displays error messages and other data from the Arduino Software (IDE).

In the bottom right-hand corner of the window, the configured board and serial port are shown

Let's try to make one of your boards blink once you've got everything set up!

5. Connect the computer to your Arduino board.
6. Select the appropriate core and board. To do so, navigate to Tools > Board > Arduino AVR Boards > Board. Make sure the board you're using is selected. If your board is missing, go to Tools > Board > Boards Manager and add it.
7. Now, select the port to ensure the computer recognises your board. This is simple to do by going to Tools > Port and selecting your board from the list.
8. Here is an example: select File > Examples > 01. Fundamentals > Blink.
9. By clicking the arrow in the top left corner it will be uploaded to your board. This process takes a few seconds, and the board must not be disconnected during this time. The message "Done uploading" will appear in the bottom output area, If the upload is successful
10. When the upload is finished, the yellow LED on your board with a L next to it should start blinking. By changing the delay number in the parenthesis to 100 and re-uploading the Blink sketch, you can change the blinking speed. The LED should now blink much more quickly.

Making use of the offline IDE 2.x:

The editor is divided into four major sections:

1. A toolbar containing common function buttons as well as a series of menus. You can verify programmes and upload programmes, create sketches, open sketches, and save sketches, select your board and port, and launch the serial monitor using the toolbar buttons.
2. Frequently used tools contained by sidebar. It allows you to quickly access board managers, libraries, board debugging, and a search and replacement tool.
3. The code will be written in text editor.

4. Console controls allow you to control the console's output.
5. The text output from the Arduino Software (IDE) will be displayed by text console, such as error messages and other data.

Serial port and configured board are shown in the bottom right-hand corner of the window.

Once you've completed your setup, let's attempt to make the board blink!

1. Connect your computer to your Arduino board.
2. Choose the correct board and port. This is done through the toolbar. Make sure the board you're using is selected. If you can't find your board, you can add it using the board manager in the sidebar.
3. Here's an example: go to File > Examples > 01. Basics > Blink.
4. Click the arrow in the top left corner to upload it to your message board. The board cannot be disconnected during this brief process, which takes place. The "Done uploading" message will show up in the bottom output area if the upload was successful
5. When the upload is completed, the yellow LED on your board with the letter L next to it should start blinking. By setting the parenthesis' delay value to 100 and uploading the Blink sketch again, you can change the blinking speed. The LED should now blink much more quickly.

4.2 BLYNK APP

Blynk can be used to develop mobile applications for smartphones that communicate with microcontrollers or even complete computers like the Raspberry Pi.

Blynk is a no-coding app builder that enables us to prototype, deploy, and manage integrated electronic equipment at any scale, from private work to millions of products used by clients.

The Blynk platform's main objective is to simplify the process of creating mobile applications. Dragging a widget and setting up a pin is all it takes to create a mobile app that can communicate with your Arduino, as you will see throughout this course.

With Blynk, you can operate a motor or LED from your phone without having to programme them. Both businesses and tech enthusiasts use the robust and scalable Blynk tool.

Soil humidity can be monitored by phone in your vegetable garden and control the irrigation or garage door. It can also be used to control smart furniture that learns from your habits, to incorporate IoT and AI into traditional industrial goods like boiler, and to improve the safety of the oilfields.

Blynk is free to use for individual use and prototyping. Selling subscriptions to businesses looking to publish Blynk-powered applications for their products or services generates revenue.

4.3 BLYNK WEBPAGE DESIGNING

In this project, we use the blynk app for monitoring the blood oxygen level, heart rate level and internal body temperature of the person sitting in the wheelchair. Along with this we are using fire sensor in front of wheel chair so we should know about the state of the fire sensor.

To stream all the things, we are designing the webpage according to the requirements.

- Begin by connecting all of these components to the Node MCU board.

We use,

1. MCU Node
2. Temperature sensor Lm35
3. Max30100(heart rate and pulse oximeter sensor)
4. Fire detector

- To begin, go to the Blynk website and sign up for a new account with your email address. Log in to your account if you have one.

- Create a new project with Node MCU as the device and WIFI as the connection type.

- Make a new template for this project and name it whatever you want.

- Now, select the data stream tab and, by clicking on new data streams, create data streams for this project.

- Navigate to new data streams, then virtual pins, and name that data stream before selecting a specific virtual pin after that give its ranging.

- LM35 temperature sensor

Virtual pin > Name-Temperature > pin-V0 > Data Type- integer > Min-0 > Max- 60

- For heart rate

Virtual pin > Name-Heart rate > pin-V0 > Data Type-Integer > Min-0 > Max- 150

- For blood oxygen levels

Virtual pin > Name-Blood Oxygen > pin-V1 > Data Type- integer > Min-0 > Max- 100

- For fire sensor

Virtual Pin > Name-Fire alarm> pin-V1 > Data Type- integer >Min-0>Max-1

- Here we used two node mcu's both the flame sensor and lm35 are connected to a single node mcu and max30100 is connected to another node mcu. So, here virtual pins are repeating.
- Next, select the web dash board tab. Then, on the dash board, drag and drop three gauge widgets and one LED widget.
- Temperature, spo2(blood oxygen level), and pulse rate are all gauged.
- The led is used for fire detection. If there is a fire, it will spread.
- Next, in the widgets, click on the setting icons one by one and select the data streams that we created earlier.
- Finally, arrange these widgets however you want.
- Now go to events, give the event a name like temperature, a type like warning, a description, and turn on "send event to notification tab."
- Add three more events for heart rate, blood oxygen levels, and fire.
- Then, click the search icon to add a new device. Choose the template you created earlier for this. Then we get the authentication code, which is used later in the code.

The Blynk Web Dash Board is now complete.

Next, let's create Blynk mobile dash board step by step.

First, download and install the Blynk app on your smartphone. Then, sign in to your account and select the template we previously created.

Then, on the dashboard, click the "Setup Dashboard" button and add the Three Gauge widgets and one LED widget. After that, arrange these widgets as desired.

Now, one by one, click on the widgets and select the data streams that we created in the web dashboard

The Blynk Mobile dash board is now available.

Now, upload the code to the node mcu and test all of these parameters (heart rate, temperature, spo2, and fire).

We will include some threshold values for all of these parameters in the code. If any of these parameters exceeds the threshold value, we will be notified via email.

CHAPTER 5

RESULT

5.1 RESULT

This is the prototype of our project “health monitoring and alerting smart wheel chair for physically disabled”.



Fig 5.1 Smart Wheelchair Prototype

The voice commands are key function of this wheelchair. we used four voice commands for the movement of wheelchair as shown in table 1 and a stop command to halt the wheelchair.

COMMAND	LEFT MOTOR	RIGHT MOTOR
Forward	clockwise	clockwise
Backward	Anti clockwise	Anti clockwise
Right	clockwise	Anti clockwise
Left	Anti clockwise	clockwise
Stop	No movement	No movement

Table 1: Commands used and directions for each motor either on clockwise or anticlockwise

we embedded health monitoring system to wheel chair which gives information about heart rate, blood oxygen levels(spO2) and body temperature. we can stream those values in mobile by blynk app.

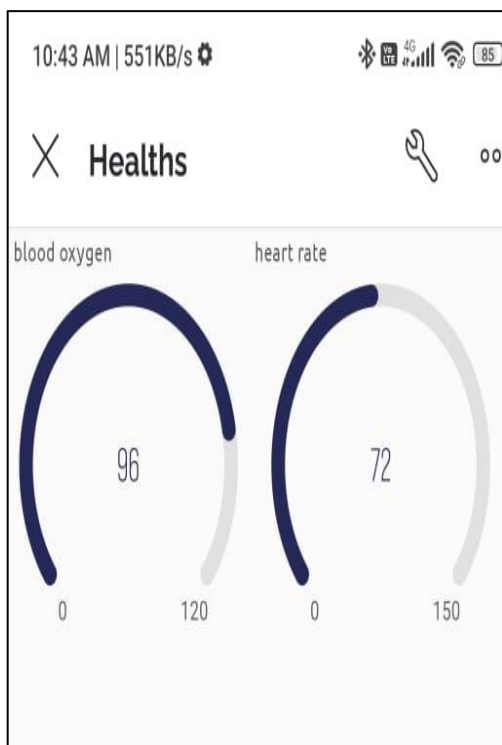


Fig 5.2 Blynk Application Interface

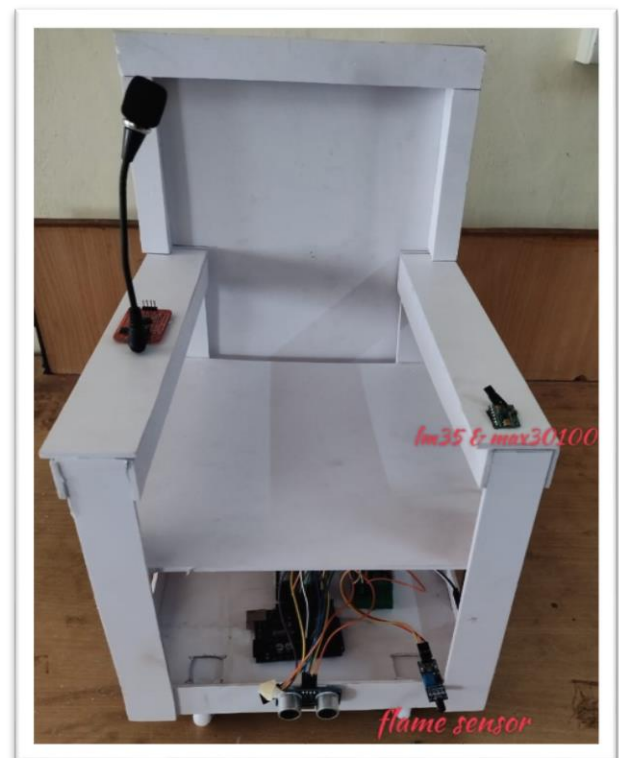


Fig 5.3 Wheelchair Prototype

We placed flame sensor in front of wheel chair and both the lm35 and flame sensor are connected to single node Mcu. If there are any abnormalities means if the readings of the sensors are above threshold value the notification will be sent to the guardian in the form of email.

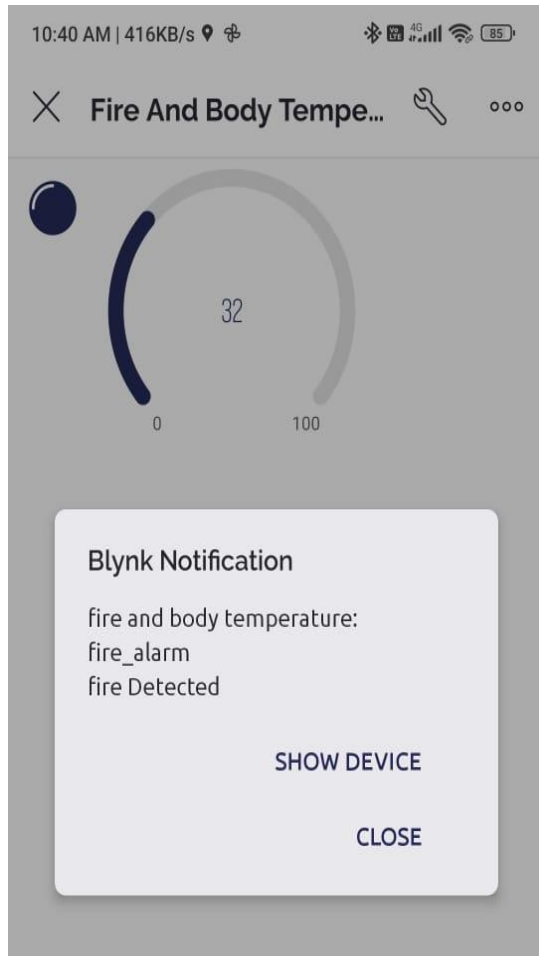


Fig 5.4

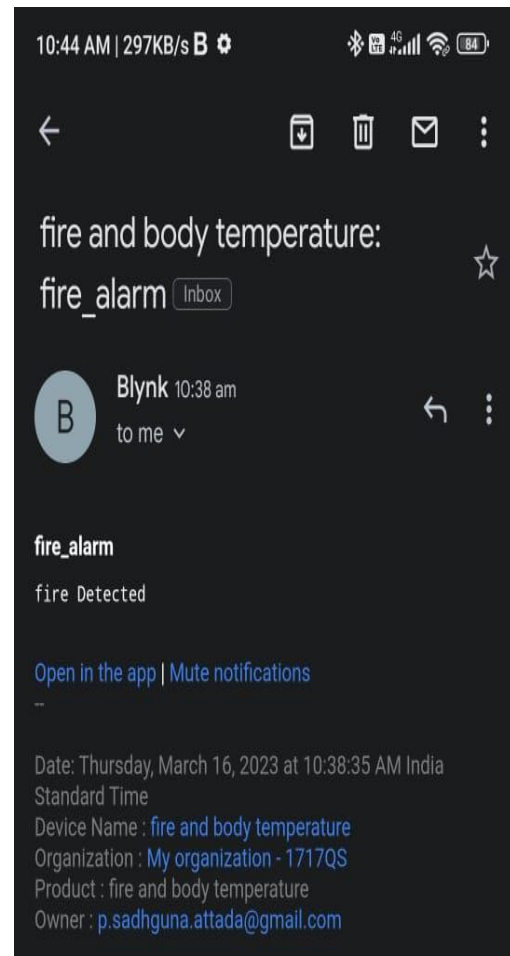


Fig 5.5

From the above figures we can see that fire is detected near wheel chair so led is on as shown in fig 6.3 and notification is sent to guardian by blynk as shown in figure 6.4.

Similarly, it will send notifications if there are any abnormalities in person body like heart rate, blood oxygen levels and temperature.

CHAPTER 6
CONCLUSION AND FUTURESCOPE

6.1 CONCLUSION

We target on the people who are unable to move even their hands. So, we are proposing a voice controlled smart wheel chair which can not only move in a certain instructed direction with the help of the voice but also can avoid obstacles in between the original path. With the help of this smart wheel chair, we are enabling a function where the disabled person can also control the appliances just with the help of voice. In case of emergency fire accidents, it's difficult for that person to move so we have included flame sensor so that it can detect whenever that person is caught under the circumstances of fire. With that we are also including Health care system which keeps track of the person's blood oxygen level, body temperature, Heart rate and gives update to the Guardian daily. All of this should be accomplished at a cost that people and organizations supporting them could still afford. We designed a health monitoring and alerting smart wheel chair with all of these features in mind.

6.2 FUTURE SCOPE

In future, we can add a feature which makes wheelchair to rotate specific angle using voice command like turn left 45 degree, turn right 90 degree to be precise etc. For the convenience of the patient the chair can be modified and can be implemented as a chair cum bed. We can add brakes to the wheel chair, so that even he is any slope area he can easily stop the wheel chair.

REFERENCES

REFERENCES

- [1] Sarmad Hameed, Muhammad Hamza Shaukat, Abdur Rafay Khan, khwaja Mobeenharoon,” SMART WHEEL CHAIR”, Department of Mechatronics Engineering, Shaheed Zulfiqar Ali Bhutt Institute of Science and Technology (SZABIST), University in Karachi, Pakistan
- [2] Sarnali Basak, Fariha Faruque Nandiny, S. M. Mazharul Hoque Chowdhury, Al Amin Biswas, “GUESTURE-BASED SMART WHEEL CHAIR FOR ASSISTING PHYSICALLY CHALLAENGED PEOPLE” Department of Computer Science and Engineering Jahangirnagar University Dhaka, Bangladesh, (ICCCI -2021), Jan. 27 – 29, 2021, Coimbatore, INDIA.
- [3] John Benson, Steven Barrett,” Next generation autonomous wheelchair control”. Electrical and Computer Engineering Department, University of Wyoming, 1000 E. University Avenue, Laramie, WY 82071, USA. prince@uwyo.edu
- [4] A. Murali, M. Mizuguchi, M. Nishikori, T. Saito, T. Osaki and R. Kenisha, "Voice Activated Wheelchair with Collision Avoidance Using Sensor Information ", ICROS-SICE International Joint Conference Fukuoka International Congress Centre, pp. 4232-4237, August 2009.
- [5] M. V. Bramhe, N. Vijay, K. B. Rao, P. Bisen, R. Navsalkar and T. Bajganiya, "Voice Controlled Wheelchair for physically Disabled Person “, International Journal of Advanced Research in Electrical Electronics and Instrumentation Engineering, vol. 6, no. 2, pp. 940-948, 2017.
- [6] Mubdi-UI alam sajid, Md Feroz Mahmud, Imteaz Rahman, Saquib shahriar, Mim Naz Rahman implemented “DESIGN OF INTELLIGENT WHEEEL CHAIR FOR HADICAPPED PEOPLE CONDUCTING BY BODY MOVEMENT”. Department of Mechatronics, RUET rajshahi University of Engineering and Technology, 2020.
- [7] Mohammad Saad Amin, Syed Tahir Hussain Rizvi, Zaid Bin Faheem, Ammar Liaqat, Sameer Malik implemented “SMART WHEEL CHAIR AN IMPLEMENTATION OF VOICE AND ANDROID CONTROLLED SYSTEM”,

Department of Computer Engineering the University of Lahore, Lahore, Pakistan, 2021.

[8] Foez Ahmed, Robi Paul, Md. Mufassal Ahmad, Arif Ahammad, Showmik Singha implemented “DESIGN AND DEVELOPMENT OF A SMART WHEEL CHAIR FOR THE DISABLED PEOPLE” Department of Electrical & Electronic Engineering, Shahjalal University of Science and Technology, Bangladesh, 2021.

[9] Krishna pal Tiwari, Mr.Kranti Kumar Dewangan, “Voice Controlled Autonomous Wheel Chair” , Department Of Computer Science And Engineering,ITM – University, School of Engineering and research, uparwara New Raipur, Raipur 492002 Chattisgarh, India

[10]Rajesh Kannan Megalingam, Ramesh Nammily Nair, Sai Manoj Prakhya,” Automated Voice based Home Navigation System for the Elderly and the Physically Challenged”proc.ICACT2011P.603-08,Feb.2011.

[11] R.Posada Gomez, L.H.Sanchez Medel, “A Hand Gesture System Of Control For An Intelligent Wheelchair”,Proc.ICEEE,P.68-71,5-7,sept.2007.

[12] Y. Asai; R. Enomoto, Y. Ueda, D. Iwai, K. Sato, “Virtual Hand Representation and Motion Control for Smart Wheelchair with TouchBased Extended Hand Projection,” In IEEJ Transactions on Electronics, Information and Systems, Vol. 139, No. 5, pp. 662-669, 2019.

[13] F. F. Alkhalid and B. K. Oleiwi, “Smart Autonomous Wheelchair Controlled by Voice Commands-Aided by Tracking System,” In IraqiJournal of Computers, Communication and Control & Systems Engineering, Vol. 19, No. 1, pp. 82-87, 2019.

APPENDIX(CODE)

CODE USED IN TRANSMITTER SIDE OF ARDUINO

```
#include <SoftwareSerial.h>
#include "VoiceRecognitionV3.h"
#include <VirtualWire.h>
char *data;
int LeftMotorForward = 5;
int RightMotorForward = 7;
int LeftMotorBackward = 6;
int RightMotorBackward = 8;
int speed1;
const int trigPin = 10;
const int echoPin = 11;
long duration;
int distance;
/
  Connection
  Arduino  VoiceRecognitionModule
  2 ----->  TX
  3 ----->  RX
*/
VR myVR(2,3); // 2:TX 3:RX, you can choose your favourite pins.

uint8_t records[7]; // save record
```



```

uint8_t buf[64];

#define forward (0)
#define backward (1)
#define right (2)
#define left (3)
#define stopp (4)
#define turnon (5)
#define off (6)

void printSignature(uint8_t *buf, int len)
{
    int i;
    for(i=0; i<len; i++){
        if(buf[i]>0x19 && buf[i]<0x7F){
            Serial.write(buf[i]);
        }
        else{
            Serial.print("[");
            Serial.print(buf[i], HEX);
            Serial.print("]");
        }
    }
}

void printVR(uint8_t *buf)
{
    Serial.println("VR Index\tGroup\tRecordNum\tSignature");
}

```

```

Serial.print(buf[2], DEC);
Serial.print("\t\t");
if(buf[0] == 0xFF){
  Serial.print("NONE");
}
else if(buf[0]&0x80){
  Serial.print("UG ");
  Serial.print(buf[0]&(~0x80), DEC);
}
else{
  Serial.print("SG ");
  Serial.print(buf[0], DEC);
}
Serial.print("\t");
Serial.print(buf[1], DEC);
Serial.print("\t\t");
if(buf[3]>0){
  printSignature(buf+4, buf[3]);
}
else{
  Serial.print("NONE");
}
Serial.println("\r\n");
}
void setup()

```

```

{
  / initialize */
  myVR.begin(9600);
  vw_set_tx_pin(12);
  vw_setup(2000);
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT);
  pinMode(LeftMotorForward, OUTPUT);
  pinMode(RightMotorForward, OUTPUT);
  pinMode(LeftMotorBackward, OUTPUT);
  pinMode(RightMotorBackward, OUTPUT);
  digitalWrite(LeftMotorBackward, LOW);
  digitalWrite(RightMotorBackward, LOW);
  digitalWrite(LeftMotorForward, LOW);
  digitalWrite(RightMotorForward, LOW);
  Serial.begin(115200);
  Serial.println("Elechouse Voice Recognition V3 Module\r\nControl LED sample");
  if(myVR.clear() == 0){
    Serial.println("Recognizer cleared.");
  }
  else{
    Serial.println("Not find VoiceRecognitionModule.");
    Serial.println("Please check connection and restart Arduino.");
    while(1);
  }
  if(myVR.load((uint8_t)forward) >= 0){
    Serial.println("forward loaded");
  }
}

```

```

if(myVR.load((uint8_t)backward) >= 0){
    Serial.println("backward loaded");
}
if(myVR.load((uint8_t)right) >= 0){
    Serial.println("right loaded");
}
if(myVR.load((uint8_t)left) >= 0){
    Serial.println("left loaded");
}
if(myVR.load((uint8_t)stopp) >= 0){
    Serial.println("stopp loaded");
}
if(myVR.load((uint8_t)turnon) >= 0){
    Serial.println("turnon loaded");
}
if(myVR.load((uint8_t)off) >= 0){
    Serial.println("off loaded");
}
}
void loop()
{
    int ret;
    ret = myVR.recognize(buf, 50);

```

```

digitalWrite(trigPin, LOW);
delayMicroseconds(2);
// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
// Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);
// Calculating the distance
distance = duration * 0.034 / 2;
if(distance <12){
    digitalWrite(LeftMotorBackward, LOW);
    digitalWrite(RightMotorBackward, LOW);
    digitalWrite(LeftMotorForward, LOW);
    digitalWrite(RightMotorForward, LOW);
}
if(ret>0){
switch(buf[1]){
case forward:
digitalWrite(RightMotorForward, HIGH);
digitalWrite(RightMotorBackward, LOW);
digitalWrite(LeftMotorForward, HIGH);
digitalWrite(LeftMotorBackward, LOW);

```

```
break;
case backward:
digitalWrite(RightMotorForward, LOW);
digitalWrite(RightMotorBackward, HIGH);
digitalWrite(LeftMotorForward, LOW);
digitalWrite(LeftMotorBackward, HIGH);
break;
case right:
digitalWrite(RightMotorForward, LOW);
digitalWrite(RightMotorBackward, HIGH);
digitalWrite(LeftMotorForward, HIGH);
digitalWrite(LeftMotorBackward, LOW);
break;
case left:
digitalWrite(RightMotorForward, HIGH);
digitalWrite(RightMotorBackward, LOW);
digitalWrite(LeftMotorForward, LOW);
digitalWrite(LeftMotorBackward, HIGH);
break;
case stopp:
digitalWrite(LeftMotorBackward, LOW);
digitalWrite(RightMotorBackward, LOW);
digitalWrite(LeftMotorForward, LOW);
digitalWrite(RightMotorForward, LOW);
break;
case turnon:
```

```
data="a";
vw_send((uint8_t *)data, strlen(data));
vw_wait_tx();
delay(500);
break;
case off:
data="b";
vw_send((uint8_t *)data, strlen(data));
vw_wait_tx();
delay(500);
break;
default:
Serial.println("Record function undefined");
break;
}
/** voice recognized */
printVR(buf);
}
}
```

CODE USED IN RECEIVING SIDE OF ARDUINO

```
#include <VirtualWire.h>

#define ledPin 6

void setup()
{
    vw_set_rx_pin(11);
    vw_setup(2000);
    pinMode(ledPin, OUTPUT);
    vw_rx_start();
}

void loop()
{
    uint8_t buf[VW_MAX_MESSAGE_LEN];
    uint8_t buflen = VW_MAX_MESSAGE_LEN;
    if (vw_get_message(buf, &buflen)){
        if(buf[0]=='a')
        {
            digitalWrite(ledPin,HIGH);
        }
        else if(buf[0]=='b')
        {
            digitalWrite(ledPin,LOW);
        }
    }
}
```


CODE USED IN NODE MCU (LM35 AND FLAME SENSOR)

```
#define BLYNK_TEMPLATE_ID "TMPLHqYkB4Mk"
#define BLYNK_TEMPLATE_NAME "fire and body temperature"
#define BLYNK_AUTH_TOKEN "vW6Op23uJ5lFAqeiZtKx9Q-9ZYnQAcvn"
#include <ESP8266WiFi.h>
#define BLYNK_PRINT Serial
#include <BlynkSimpleEsp8266.h>
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "pavan"; // type your wifi name
char pass[] = "12345678";
float temp_val;
const int flamepin = D0;
int flame = HIGH;
WidgetLED led1(V1);
void setup() {
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);
  pinMode(flamepin, INPUT);
}

void loop() {
  temp_val = (analogRead(A0) * (3.3/1023) * 80); /* Convert adc value to
equivalent voltage */
  Serial.print("Temperature = ");
  Serial.print(temp_val);
  Blynk.virtualWrite(V0,temp_val);
  Serial.print(" Degree Celsius\n");
}
```

```
if(temp_val > 29){
  Blynk.email("test@gmail.com", "Alert", "temperature Detected!");
  Blynk.logEvent("body_temperature", "temperature Detected");
}
flame = digitalRead(flamepin); // read FLAME sensor
if(flame == LOW)
{
  Serial.println("Fire! Fire!");
  Blynk.email("test@gmail.com", "Alert", "fire Detected!");
  Blynk.logEvent("fire_alarm", "fire Detected");
  led1.on();
}
else{
  Serial.println("Peace");
  led1.off();
}
delay(2000);
Blynk.run();
}
```

CODE USED IN NODE MCU (MAX30100)

```
#define BLYNK_TEMPLATE_ID "TMPLaETIbBUq"
#define BLYNK_TEMPLATE_NAME "healths"
#define BLYNK_AUTH_TOKEN "FMwRwATmM7jR8J80wF8zqV20Si3OT5Ng"
#include <Wire.h>
#include "MAX30100_PulseOximeter.h"
#include <ESP8266WiFi.h>
#define BLYNK_PRINT Serial
#include <BlynkSimpleEsp8266.h>

#define REPORTING_PERIOD_MS 1000
PulseOximeter pox;
uint32_t tsLastReport = 0;
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "pavan"; // type your wifi name
char pass[] = "12345678";
void onBeatDetected() {
  Serial.println("Beat!");
}
void setup() {
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);
  Serial.println("Initializing pulse oximeter..");
  if (!pox.begin()) {
    Serial.println("FAILED");
  } else {
    Serial.println("SUCCESS");
  }
}
```

```

pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA);
pox.setOnBeatDetectedCallback(onBeatDetected);
}
void loop() {
  pox.update();
  if (millis() - tsLastReport > REPORTING_PERIOD_MS)
  {
    int h = pox.getHeartRate();
    Serial.print("Heart rate:");
    Serial.print(h);
    int b =pox.getSpO2();
    Serial.print("bpm / SpO2:");
    Serial.print(b);
    Serial.println("%");
    Blynk.virtualWrite(V0,h);
    if(pox.getHeartRate()> 90){
      Blynk.email("test@gmail.com", "Alert", "heart rate Detected!");
      Blynk.logEvent("heart_rate","heart rate Detected");
    }
    Blynk.virtualWrite(V1, b);
    if(pox.getSpO2()> 90){
      Blynk.email("test@gmail.com", "Alert", "blood oxygen Detected!");
      Blynk.logEvent("blood_oxygen","blood oxygen Detected");
    }
    Blynk.run();
    tsLastReport = millis();
  }
}

```

CONFERENCE

Punumajji Divya Sri, Attada Pavan Sadhguna, Sirukuri Avinash, Ravi kumar Kanuri, “**VOICE CONTROLLED SMART WHEEL CHAIR FOR PHYSICALLY DISABLED PEOPLE** “, International Conference on Advancement in Electronic Systems and Communication Technologies(ICAESCT 2022), Department of Electronics and Communication Engineering, Anil Neerukonda Institution of Technology and Sciences(ANITS), Visakhapatnam, INDIA.

