Faculty of Computers and Artificial Intelligence

Senior Project Documentation

Benha University

"CS13"

IOT Smart Wheelchair

A senior project submitted in partial fulfillment of the requirements for

The degree of Bachelor of Computers and Artificial Intelligence

Department of **Computer Science**

Project Team

- ✓ Ahmed Elsayed Sabra
- ✓ Ahmed Rafat AbdelWahab
- ✓ Ahmed Awny Ahmed
- ✓ Abdel Rahman Hemdan Abdel Rahman
- ✓ Marwa Ashraf Elshahat
- ✓ Marwa Mohamed Galal

Under Supervision of

Dr. Fatma Sakr

Declaration

We hereby certify that this material, which we now submit for assessment on the program of study leading to the award of Bachelor of Computers and Informatics in **Computer Science** is entirely our own work, that we have exercised reasonable care to ensure that the work is original, and does not to the best of our knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of our work.

Signed: _____

_____

_____

_____

_____

Date: Aug 18th, 2020

Acknowledgments

- The success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

- We'd also like to thank **Dr. Fatma Sakr**, for providing us with crucial guidance and mentorship to deal with challenges and accomplish our tasks and goals.

    - The supervision and support he gave truly help in the progression of our graduation project. The co-operation much indeed appreciated.
    - Without their contribution, this project has been possible.
    - Every project big or small is successful largely due to the effort of a number of wonderful people who have always given their valuable advice or lent a helping hand. We sincerely appreciate the inspiration, support and guidance of all those people who have been instrumental in making this project a success. We enjoyed the work very much; however, programming might have been hard work at some points.
    - We would also like to think all the faculty members of Benha University for their critical advice and guidance without which this project would not been possible.
    - We feel that we have really been lucky to be working with someone him.in addition, many thanks go to Eng. Aya Hatem for his inspirational thoughts, valuable guidance and assistance throughout this work and for their patience.

Finally, we are grateful to the many people who helped us with their contribution to complete this project.

Abstract

There are many paralysis patients who suffer from the difficulty of using the chair for the traditional wheelchair, in addition to the fact that many of them also suffer from loneliness and the absence of anyone to care for them, based on that the idea of our project came.

This project helps people with paralysis who have no one to look after them either.

The wheelchair is moved via a joystick or an Android app that the patient uses to give four way movement commands (left, right, forward, or backward).

The patient can also move the chair by voice commands using the same Android app.

The wheelchair is equipped with an ultrasound sensor,

If the chair falls in any direction, a message will be sent via the sensor to the hospital to obtain a rescue.

The Global Positioning System (GPS) is used to send the exact location of an infected patient so that the hospital sends someone to rescue him.

The wheelchair also has a sensor to sense the patient's body temperature.

All patient data read by the sensors is sent to the hospital in order to send patient rescue when needed.

CONTENTS

LIST OF FIGURES

List of Tables


Ch1
Chapter one
INTRODUCTION

Chapter One: An Introduction

1. *PROJECT OVERVIEW*

This project is a wheelchair equipped with some advanced features to help people with special needs, especially cases that have no one to take care of them permanently or even temporarily.

Also, this chair is distinguished because it makes the patient himself control the movement of the chair

The wheelchair is equipped with GPS and Wi-Fi to make the patient in constant contact with the hospital or the agency responsible for taking care of him and saving him.

The patient controls the movement of the wheelchair via a joystick or an Android application on his mobile phone or any device that calls Android.

1.1 PURPOSE

Our project aims to help people with special needs and specifically those who have no one to take care of them as these patients are not able to serve themselves.

Therefore, they are constantly at risk. Our project provides permanent communication between the patient and the agency responsible for remote care such as the hospital or others.

Where the wheelchair, through its advanced attachments, notifies that party responsible for the patient of any danger facing the patient such as the fall of the chair, for example, where the fall of the chair means that the patient himself has fallen and other characteristics that aim to protect the patient.

1.2 SCOPE

The Internet of things (IOT) is a system of interrelated computing devices, mechanical and digital machines provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

At the end of our project, we aim to make patients with special needs who suffer from loneliness and lack of those who care about them enjoy safety and ease of movement through our wheelchair that works with the technology of IOT.

Where the patient is the one who controls the chair through the joystick or Android application.

Our wheelchair will be a safe way to facilitate the patient's life in many aspects.

1.3 Motivation

The motivation that drove us to do this project is to help people with special needs as well as the elderly who have become unable to move. So we made this chair in order to make them able to move more easily. It also provides them with some services as it keeps the patient in constant contact with the hospital so that help is sent to him whenever he needs it.

The chair is attached to many sensors that transmit information about the patient to the hospital. It also contains GPS to make the hospital aware of the patient’s whereabouts, in order to send him help where he was.

1.4 Problem Statement

Now, we will talk about the state of the problem. Many patients who are unable to move, encounter major mobility problems and turn to the traditional wheelchair.

But there are many defects in this chair as it needs to make an effort on the part of the patient, and it also needs that the patient not be alone, as the patient may be exposed to the injury if the chair stumbles on anything on the ground and thus leads to the fall of the patient, which will harm him.

We have solved these problems as the chair moves easier by means of wheels and a motor

The movement of the chair is controlled by the patient either by joystick or mobile application

The patient uses the application to move the chair either by using voice commands or by using the directional buttons

We have also solved the problem of the patient's presence alone, as we have equipped sensors in the chair. If the chair falls, he is sent to the hospital, so rescue is sent to him.

1.5 Objective

Our project aims to help people with special needs who are unable to move, which facilitates movement for them, as well as communicating with the hospital or the rescue agency responsible for them.

The chair contains a set of sensors that take data about the patient, such as his temperature and location, and send it to the hospital via a Wi-Fi module and put it on the hospital's sign to be aware of the patient's condition always

The chair also contains an ultra-sonic sensor, the function of this sensor is that when the chair falls, it is sent to the hospital system, so the hospital sends a rescue to the patient

The chair uses a Bluetooth module that connects the chair with the Android application

The patient moves the chair using the application, either by using voice commands or by using the directional buttons

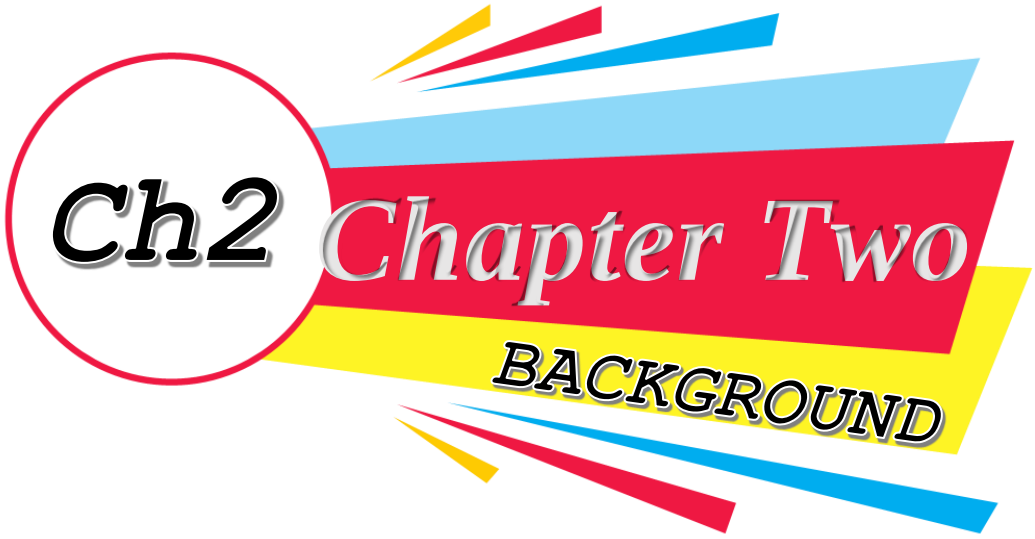
1.6 Documentation Organization:

In Chapter two: Background

In Chapter three: Methodology

In Chapter four: Conclusion

In Chapter Five: References


Ch2
Chapter Two
BACKGROUND

Chapter Two: BACKGROUND

2. IOT OVERVIEW

IOT systems allow users to achieve deeper automation, analysis, and integration within a system. They improve the reach of these areas and their accuracy. IOT utilizes existing and emerging technology for sensing, networking, and robotics.

IOT exploits recent advances in software, falling hardware prices, and modern attitudes towards technology. Its new and advanced elements bring major changes in the delivery of products, goods, and services; and the social, economic, and political impact of those changes. [1]

2.1 IOT – Advantages

The advantages of IOT span across every area of lifestyle and business. Here is a list of some of the advantages that IOT has to offer −

- Improved Customer Engagement − Current analytics suffer from blind-spots and significant flaws in accuracy; and as noted, engagement remains passive. IOT completely transforms this to achieve richer and more effective engagement with audiences.
- Technology Optimization − the same technologies and data which improve the customer experience also improve device use, and aid in more potent improvements to technology. IOT unlocks a world of critical functional and field data.
- Reduced Waste − IOT makes areas of improvement clear. Current analytics give us superficial insight, but IOT provides real-world information leading to more effective management of resources.
- Enhanced Data Collection − Modern data collection suffers from its limitations and its design for passive use. IOT breaks it out of those

spaces, and places it exactly where humans really want to go to analyze our world. It allows an accurate picture of everything. [1]

2.2 IOT – Disadvantages

Though IOT delivers an impressive set of benefits, it also presents a significant set of challenges. Here is a list of some its major issues −

- Security − IOT creates an ecosystem of constantly connected devices communicating over networks. The system offers little control despite any security measures. This leaves users exposed to various kinds of attackers.
- Privacy − the sophistication of IOT provides substantial personal data in extreme detail without the user's active participation.
- Complexity − Some find IOT systems complicated in terms of design, deployment, and maintenance given their use of multiple technologies and a large set of new enabling technologies.
- Flexibility − Many are concerned about the flexibility of an IOT system to integrate easily with another. They worry about finding themselves with several conflicting or locked systems.
- Compliance − IOT, like any other technology in the realm of business, must comply with regulations. Its complexity makes the issue of compliance seem incredibly challenging when many consider standard software compliance a battle. [1]

2.3 EMBEDDED SYSTEM OVERVIEW

An embedded system is a computer system with a dedicated function within a larger mechanical or electrical system.

Since the embedded system is dedicated to specific tasks, design engineers can optimize it to reduce the size and cost of the product and increase the reliability and performance.

Some embedded systems are mass-produced, benefiting from economies of scale.

Embedded systems are commonly found in consumer, cooking, industrial, automotive, medical, commercial and military applications

Embedded application is a software application that permanently resides in an industrial or consumer device, providing some type of control function and /or user interface the software is typically stored in non-volatile memory such as ROM or flash memory contrast with general-purpose computer that can be used to run all kinds of applications. **[2]**

2.3.1 Embedded system applications

Embedded systems find numerous applications in various fields such as digital electronics, telecommunications, computing network, smart cards, satellite systems, military defense system equipment, research system equipment, and so on. Let us discuss a few practical applications of embedded systems that are used in designing embedded projects as a part of engineering final year electronics projects

Some other applications:

- Latest Smart TVs
- GPS Navigation Systems
- Almost all Modern Day
- Phone Missile Guidance System
- Space Exploration
- (Rovers) Automobiles
- (ABS, Airbags) Industries
- (Assembly Robots)

Road Safety Systems (Traffic Monitoring and Collision Alert Systems) and many other. [2]

2.3.2 Embedded Systems Hardware:

Every electronic system consists of hardware circuitry, similarly, embedded system consists of hardware such as power supply kit, central processing unit, memory devices, timers, Output circuits, serial communication ports, and system application specific circuit components & circuits. [2]

2.4 Web service

The term Web service (WS) is either:

- a service offered by an electronic device to another electronic device, communicating with each other via the World Wide Web, or
- a server running on a computer device, listening for requests at a particular port over a network, serving web documents (HTML, JSON, XML, images), and creating web applications services, which serve in solving specific domain problems over the Web (WWW, Internet, HTTP).

In a Web service a Web technology such as HTTP is used for transferring machine-readable file formats such as XML and JSON.

In practice, a Web service commonly provides an object-oriented Web-based interface to a database server, utilized for example by another Web server, or by a mobile app, that provides a user interface to the end-user. Many organizations that provide data in formatted HTML pages will also provide that data on their server as XML or JSON, often through a Web service to allow syndication, for example, Wikipedia's Export. Another application offered to the end-user may be a mash up, where a Web server consumes several Web services at different machines and compiles the content into one user interface. **[3]**

Web services that use markup languages

There are a number of Web services that use markup languages:

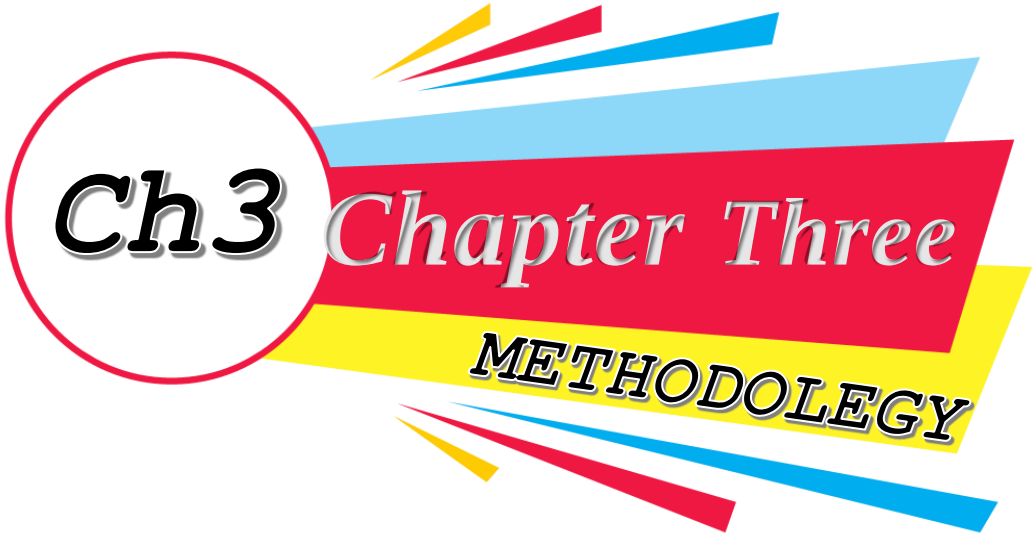
- JSON-RPC.
- JSON-WSP
- Representational state transfer (REST) versus remote procedure call (RPC)
- Web Services Conversation Language (WSCL)
- Web Services Description Language (WSDL), developed by the W3C
- Web Services Flow Language (WSFL), superseded by BPEL
- Web template
- WS-Metadata Exchange
- XML Interface for Network Services (XINS), provides a POX-style web service specification format. **[3]**

2.5 System Development Requirements we use in the Project:

- Web Service Tools:
    1) PHP/MySQL
    2) Java script
    3) Css
    4) Html

- Hardware Tools :
    1) Arduino. Uno
    2) Hc-05 Bluetooth module
    3) Esp. 32
    4) Neo 6 m v2 GPS
    5) DS18B20 sensor
    6) Ultrasonic sensor

- Software Tools :
    1) Android Studio
    2) Arduino ide
    3) Web host


Ch3
Chapter Three
METHODOLEGY

Chapter Three:

SYSTEM DESIGN & IMPLEMENTATON

3.1 Bluetooth Controlled Wheelchair

Step 1 : Components Required

1) Any car chassis kit
2) Arduino UNO
3) L298 motor drive
4) Hc-05 Bluetooth module
5) Two batteries
6) Jumper Wires
7) Mobile with Bluetooth

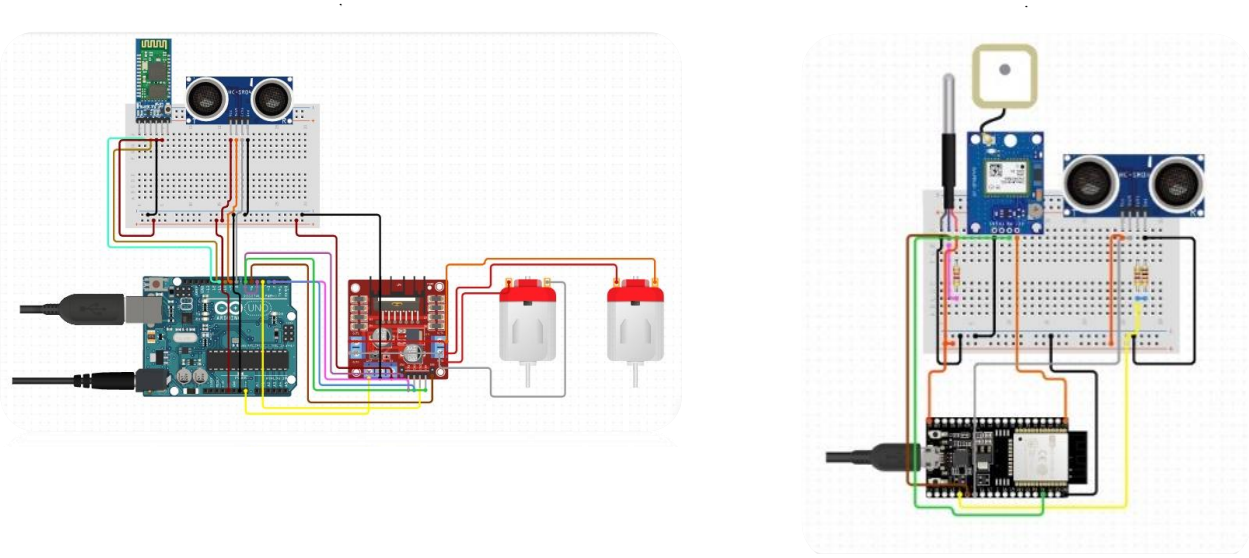


Fig 3.1: Connections& components required

Step 2: Join Bluetooth Module to Arduino

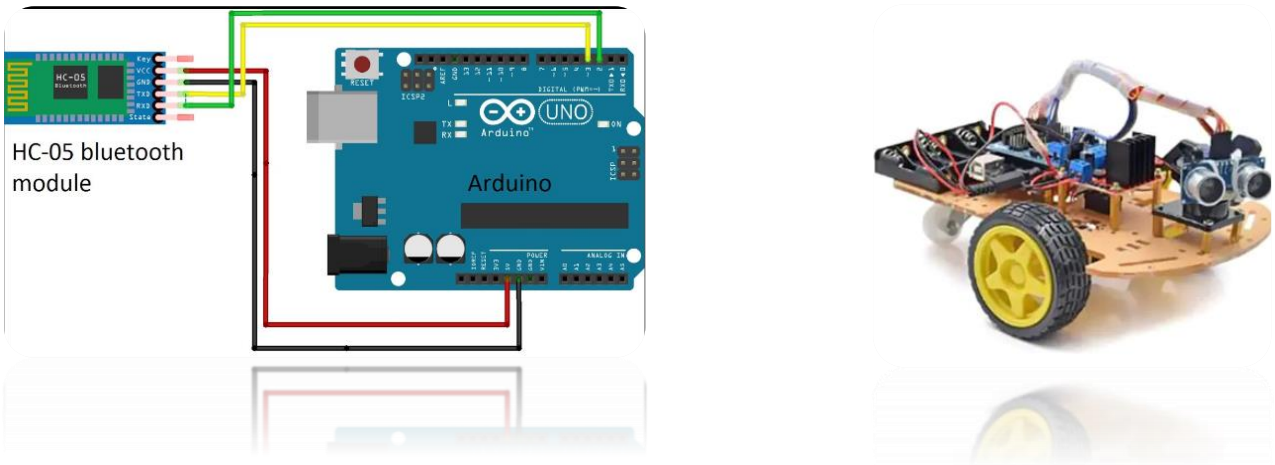


Fig 3.2: Bluetooth Module connection

Step 3: Connecting Motor Drive to Battery

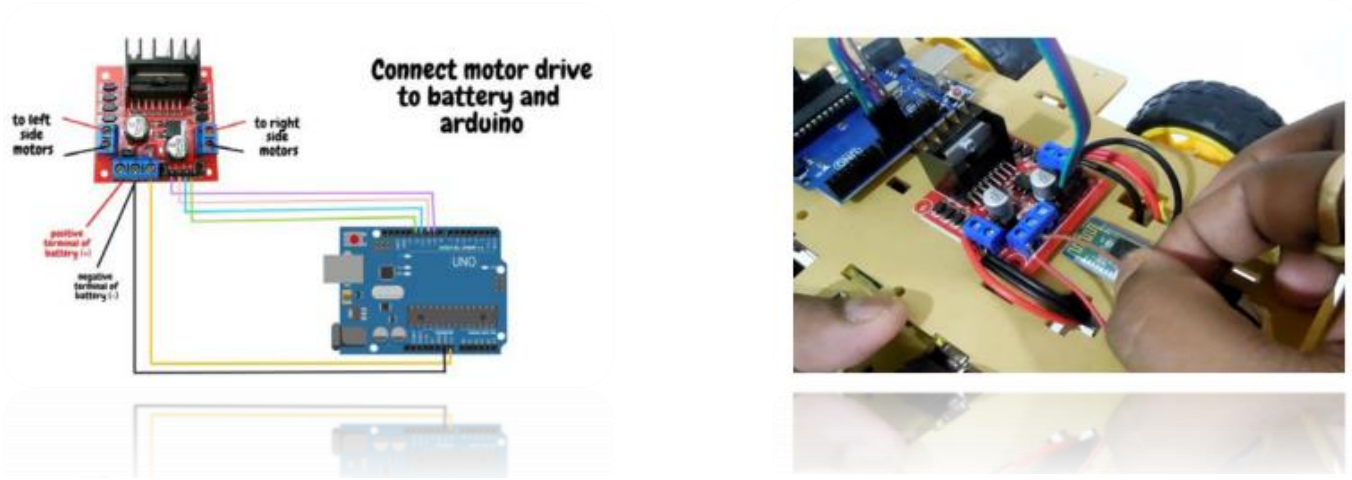


Fig 3.3: Connecting Motor Drive to Battery

Connect motor drive's power input socket, to positive and negative terminal of battery. Also connect the negative terminal of battery to GND of Arduino.

Finally connect 3rd terminal to Vin of Arduino.

Step 4:

After uploading, disconnect the Arduino from pc.

Now connect Rx of Hc-05 to TX of Arduino and TX of Hc-05 to Rx of Arduino.

(Do not connect these before uploading the code otherwise it may burn Your Arduino while uploading the code)

Step 5: **Pairing with Bluetooth Module**

Start the Wheelchair. Check that the LED of Bluetooth module is blinking fast without pairing.

Pair the HC-05 Bluetooth module with your smartphone. Enter password 1234. (If it not works try 0000)

After pairing open the app and choose HC-05 to pair with. Check the LED of Bluetooth module, its blinking rate would have been very slow now.

Step 6: Arduino code

```
1. char t;
2. void setup() {
3.                   pinMode(9,OUTPUT);   //left motors forward
4.                   pinMode(10,OUTPUT);   //left motors reverse
5.                   pinMode(11,OUTPUT);   //right motors forward
6.                   pinMode(12,OUTPUT);   //right motors reverse
7.                   Serial.begin(9600);   }
8.
9. void loop() {
10.      if(Serial.available()){
11.      t = Serial.read();
12.      Serial.println(t);
13.      }
14.
15.      if(t == '1'){            //move forward
16.                               digitalWrite(9,HIGH);
17.                               digitalWrite(10,LOW);
18.                               digitalWrite(11,HIGH);
19.                               digitalWrite(12,LOW);    }
20.
21.      else if(t == '2'){      //move reverse
22.                              digitalWrite(9,LOW);
23.                              digitalWrite(10,HIGH);
24.                              digitalWrite(11,LOW);
25.                              digitalWrite(12,HIGH);     }
26.
27.      else if(t == '3'){      //turn right
28.                              digitalWrite(9,LOW);
29.                              digitalWrite(10,LOW);
30.                              digitalWrite(11,HIGH);
31.                              digitalWrite(12,LOW);      }
32.
33.      else if(t == '4'){      //turn left
34.                              digitalWrite(9,HIGH);
35.                              digitalWrite(10,LOW);
36.                              digitalWrite(11,LOW);
37.                              digitalWrite(12,LOW);      }
38.
39.      else if(t == '5'){      //STOP (all motors stop)
40.                              digitalWrite(9,LOW);
41.                              digitalWrite(10,LOW);
42.                              digitalWrite(11,LOW);
43.                              digitalWrite(12,LOW);      }
44.      delay(100);
45.      }
```

3.2 Android Application

We've developed an Android app to provide an easier way to move for patients with paraplegia and moving a wheelchair.

The purpose of the application is to control the chair by connecting to Bluetooth via the Bluetooth module HC 05

Installed in the Arduino device that controls the chair by application

The main features of the application evolve around control by choosing the appropriate control:

- ***Direction***
- ***Voice***

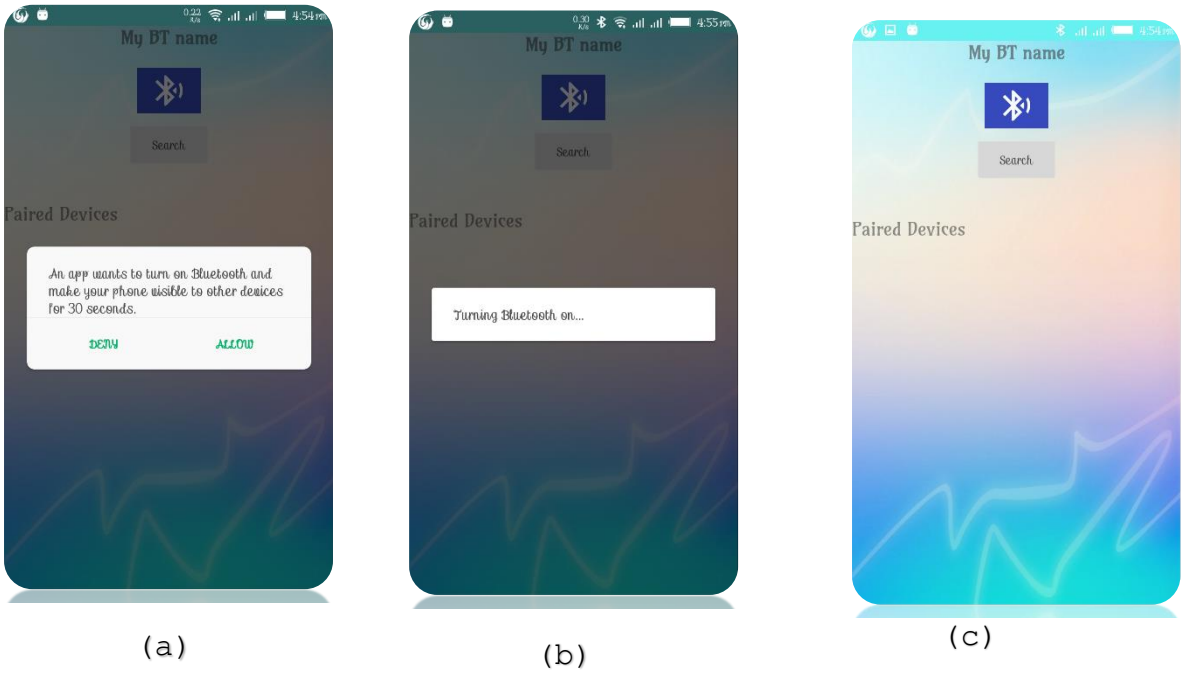
✓ The Splash page that is the first Page

In An Application that opening of

The Application .

Fig 3.4:
Splash Screen App

Step 1:

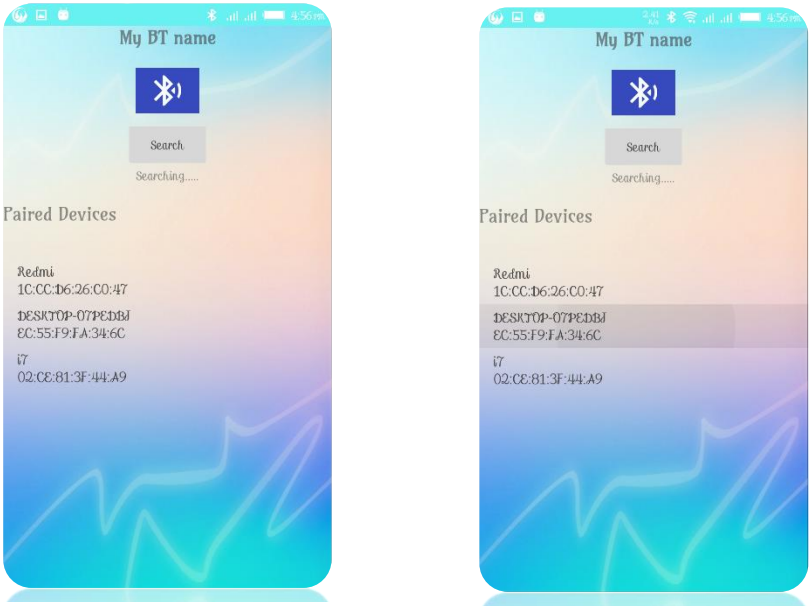


(a) (b) (c)

Fig 3.5: App Request Turn on Bluetooth

A Request for Permission indicates that the Bluetooth is turned on in the Android device, after the start of the operation, a Search for devices starts after what I do, and the paired devices appear.

Never connect to the Bluetooth Module in the Wheelchair.



(a) (b)

Fig 3.6: App Search & Connect

Step 2:

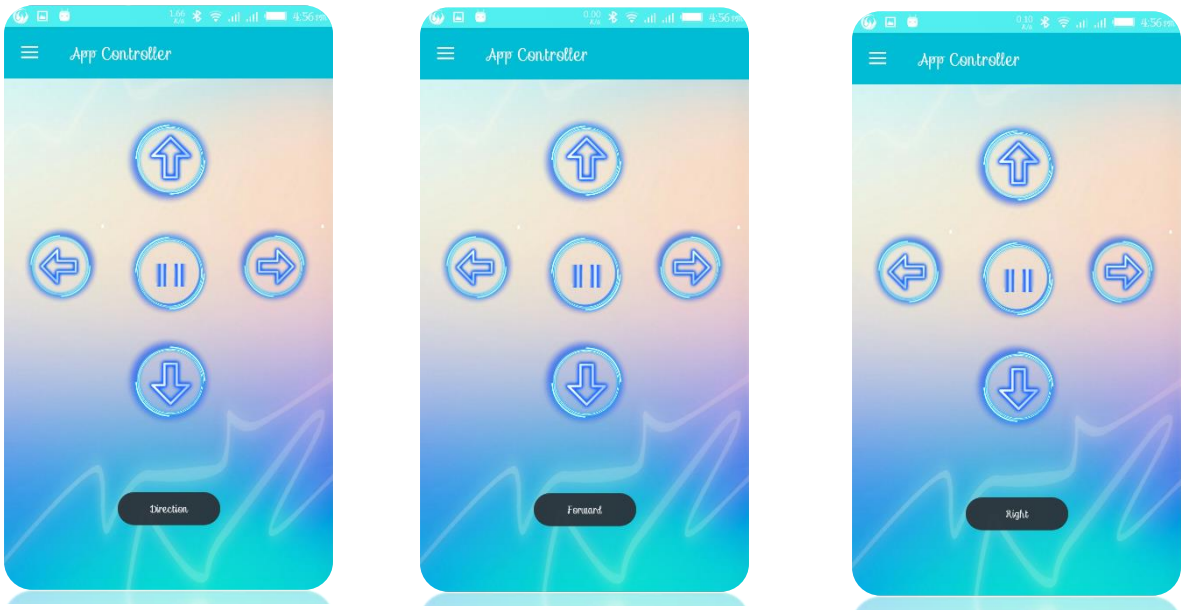
The address of the Bluetooth Module
Is followed by the application and a new page
Appears in which the Navigation Bar
Is able to choose the way to control
The Wheelchair.

(a)

Fig 3.7:

Selection Controlling

Step 3:



(a) (b) (c)

Fig 3.8
Direction Controls

The Direction page has 5 Button which are the 4 directions

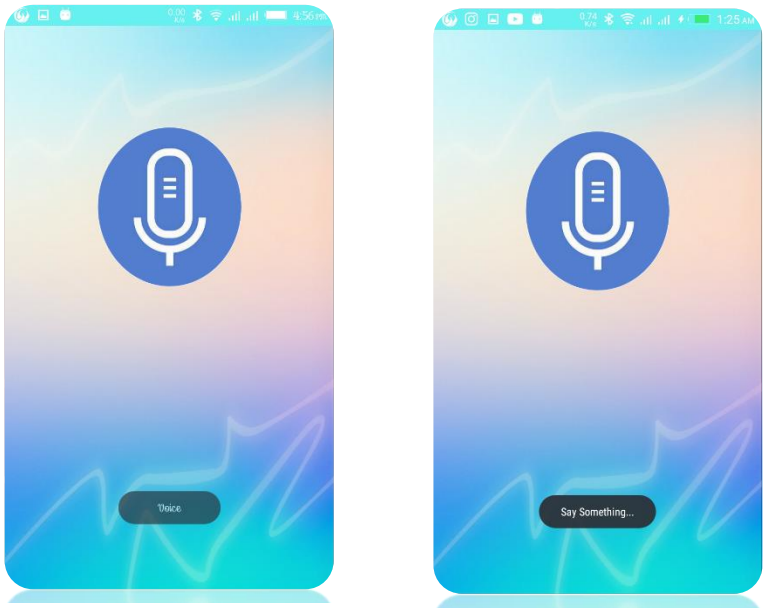
(Forward) - (Reserve) - (Right) - (Left)

In addition to the button for the stop of the chair

The chair will move via the command to send a message to the Bluetooth device, a message in it

- ❖ (Forward) Send Command (1)
- ❖ (Reverse) Send Command (2)
- ❖ (Right) Send Command (3)
- ❖ (Left) Send Command (4)
- ❖ (Stop) Send Command (5)

Step 4:



(a) (b)

Fig 3.9

Voice Controls

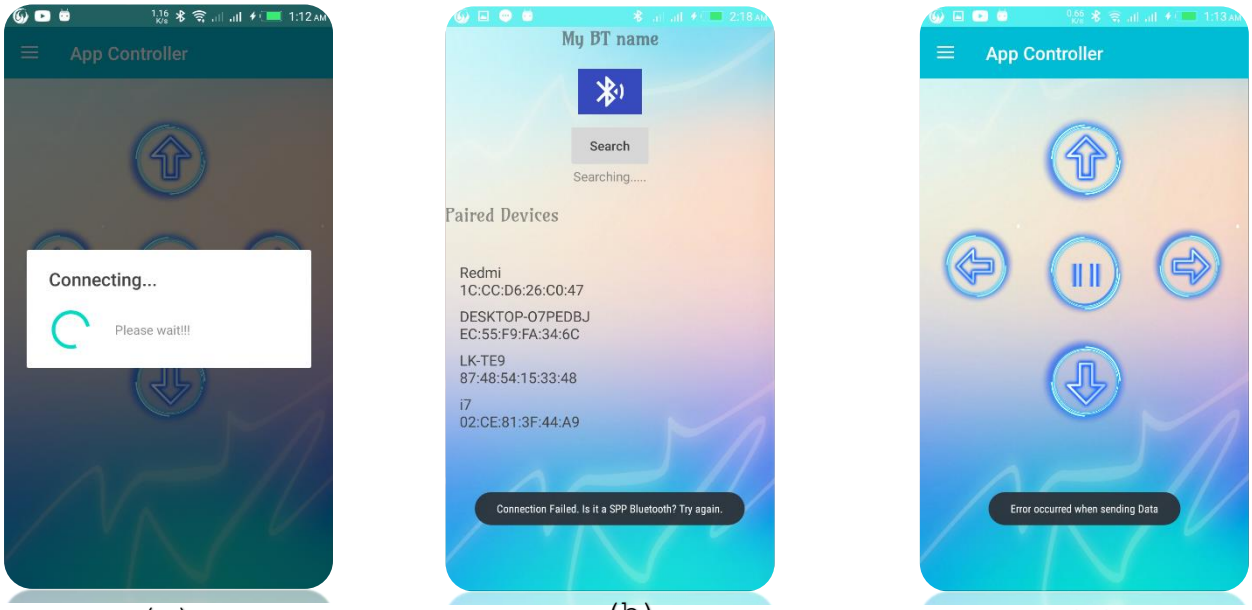
The next control

Page is Voice

Here is the idea in controlling Speech to Text, which is the conversion of voice to speech by controlling the command, and a message is sent to the Bluetooth device, a message in it when speaking a word from:

- ❖ (Forward) Send Command (1)
- ❖ (Reverse) Send Command (2)
- ❖ (Right) Send Command (3)
- ❖ (Left) Send Command (4)
- ❖ (Stop) Send Command (5)

Step 5:



(a) (b) (c)

Fig 3.10: Messages of Any Error Connection

The Check process occurs before any order is executed, for example:

- At the beginning of the activity after requesting the connection, if a problem occurs, then a message will be sent to the user if the Bluetooth is required

After making a connection to the device, if a Bluetooth Module disconnects, it is necessary to return to the first Activity. Call and control the chair in both ways

If you send a different address, it will ask me to call from the first

Each button when pressed will show a movement message.

3.3 Joystick control

Overview:-

We can control the speed of the DC motor by simply controlling the input voltage to the motor and the most common method of doing that is by using PWM signal

- ### PWM DC Motor Control:-

PWM, or pulse width modulation is a technique which allows us to adjust the average value of the voltage that's going to the electronic device by turning on and off the power at a fast rate. The average voltage depends on the duty cycle, or the amount of time the signal is ON versus the amount of time the signal is OFF in a single period of time.

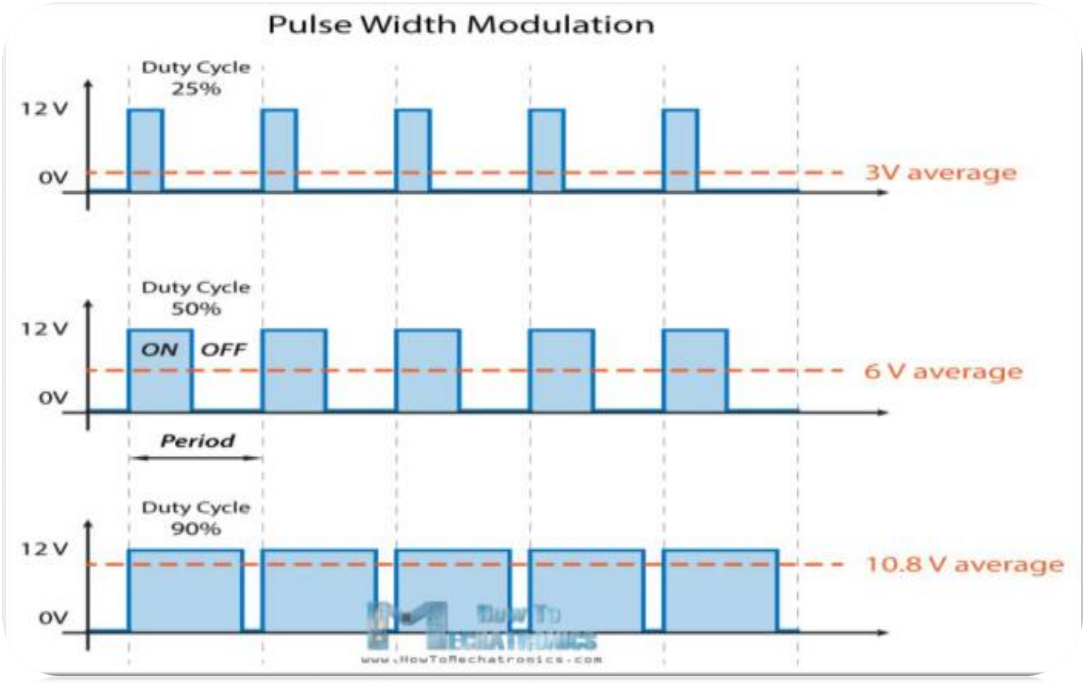


Fig 3.11: Pulse Width Modulation (PWM)

➢ *H-Bridge DC Motor Control direction:-*

On the other hand, for controlling the rotation direction, we just need to inverse the direction of the current flow through the motor, and the most common method of doing that is by using an H-Bridge. An H-Bridge circuit contains four switching elements, transistors or MOSFETs, with the motor at the center forming an H-like configuration.

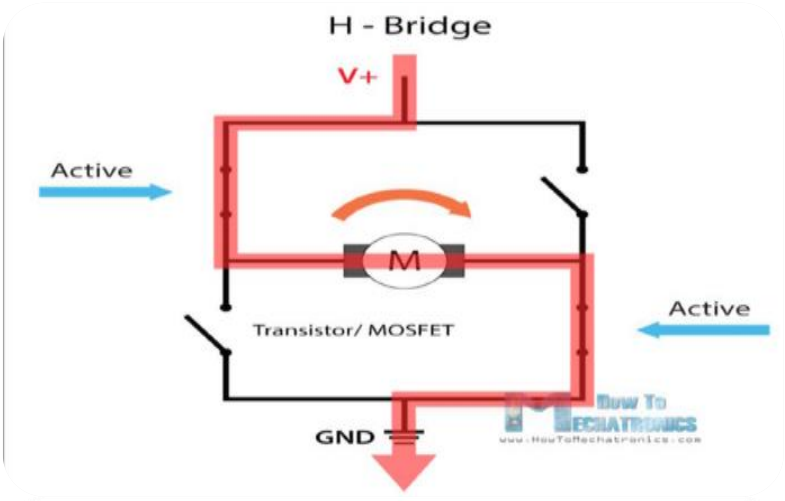


Fig 3.12: H-Bridge DC Motor

So if we combine these two methods, the PWM and the H-Bridge, we can have a complete control over the DC motor. There are many DC motor drivers that have these features and the L298N is one of them.

Now let's take a look at the Arduino code and see how it works.

After defining the pins, in the loop section, we start with reading the joystick X and

Y axis values. The joystick is actually made of two potentiometers which are connected to the analog inputs of the Arduino and they have values from 0 to 1023.

When the joystick stays in

Its center position the value of

Either potentiometers,

Or axes is around 512.

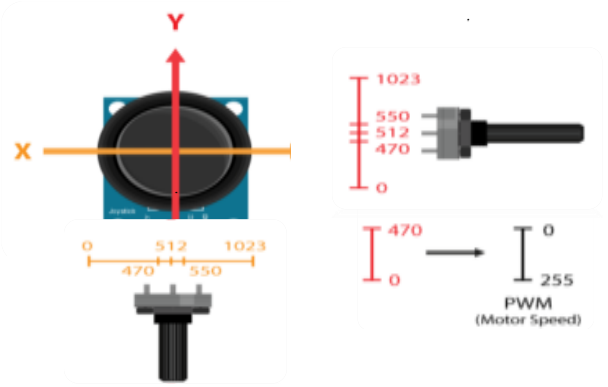


Fig 3.13

Joystick Coordinates

We will add a little tolerance and consider the values from 470 to 550 as center. So if we move the Y axis of joystick backward and the value goes below 470 we will set the two motors rotation direction to backward using the four input pins. Then, we will convert the declining values from 470 to 0 into increasing PWM values from 0 to 255 which is actually the speed of the motor.

Similar, if we move the Y axis of the joystick forward and the value goes above 550 we will set the motors to move forward and convert the readings from 550 to 1023 into PWM values from 0 to 255. If the joystick stays in its center the motors speed will be zero.

So again, first we need to convert the X axis readings into speed values from 0 to 255. For moving left, we use this value to decrease the left motor speed and increase the right motor speed. Here, because of the arithmetic functions we use two additional “if” statements to confine the range of the motor speed from 0 to 255 he same method is used for moving the car to the right.

Depending on the applied voltage and the motor itself, at lower speeds the motor is not able to start moving and it produces a buzzing sound. In my case, the motors were not able to move if the value of the PWM signal was below 70. Therefore using this two if statements I actually confined to speed range from 70 to 255. At the end we just send the final motor speeds or PWM signal to the enable pins of the L298N driver.

Joysticksone §

```
#define enA 9
#define in1 4
#define in2 5
#define enB 10
#define in3 6
#define in4 7
int motorSpeedA = 0;
int motorSpeedB = 0;
void setup() {
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
}
void loop() {
  int xAxis = analogRead(A0); // Read Joysticks X-axis
  int yAxis = analogRead(A1); // Read Joysticks Y-axis
  // Y-axis used for forward and backward control
  if (yAxis < 470) {
    forward();
    motorSpeedA = map(yAxis, 470, 0, 0, 255);
    motorSpeedB = map(yAxis, 470, 0, 0, 255);
  }
  else if (yAxis > 550) {
    reverse();
    motorSpeedA = map(yAxis, 550, 1023, 0, 255);
    motorSpeedB = map(yAxis, 550, 1023, 0, 255);
  }
  // If joystick stays in middle the motors are not moving
  else {
    motorSpeedA = 0;
    motorSpeedB = 0;
```

```
  // X-axis used for left and right control
  if (xAxis < 470) {
    // Convert the declining X-axis readings from 470 to 0 into increasing 0 to 255 value
    int xMapped = map(xAxis, 470, 0, 0, 255);
    // Move to left - decrease left motor speed, increase right motor speed
    motorSpeedA = motorSpeedA - xMapped;
    motorSpeedB = motorSpeedB + xMapped;
    // Confine the range from 0 to 255
    if (motorSpeedA < 0) {
      motorSpeedA = 0;
    }
    if (motorSpeedB > 255) {
      motorSpeedB = 255;
    }
  }
  if (xAxis > 550) {
    // Convert the increasing X-axis readings from 550 to 1023 into 0 to 255 value
    int xMapped = map(xAxis, 550, 1023, 0, 255);
    // Move right - decrease right motor speed, increase left motor speed
    motorSpeedA = motorSpeedA + xMapped;
    motorSpeedB = motorSpeedB - xMapped;
    // Confine the range from 0 to 255
    if (motorSpeedA > 255) {
      motorSpeedA = 255;
    }
    if (motorSpeedB < 0) {
      motorSpeedB = 0;
    }
  }
  if (motorSpeedA < 70) {
    motorSpeedA = 0;
  }
  if (motorSpeedB < 70) {
```

```
void forward()
{
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
}

void reverse()
{
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
}
```

3.4 IOT System Implementation

3.4.1 Data Transmission Hardware Device (H/W to Web Server):

Get starting with Esp. 32

The ESP32 is loaded with lots of new features. The most relevant: it combines Wi-Fi and Bluetooth wireless capabilities and its dual core.

We'll be using the ESP32 DEVKIT DOIT board as a reference

The type of esp.:-

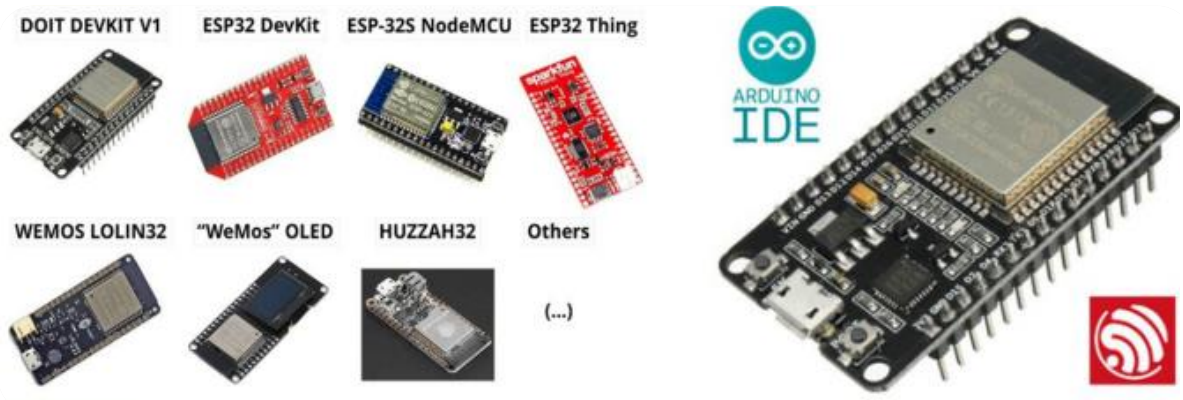


Fig 3.14: Types of ESP 32

Specifications:-

When it comes to the ESP32 chip specifications, you'll find that:

The ESP32 is dual core, this means it has 2 processors.

It has Wi-Fi and Bluetooth built-in.

It runs 32 bit programs.

The clock frequency can go up to 240MHz and it has a 512 kB RAM.

This particular board has 30 or 36 pins, 15 in each row.

It also has wide variety of peripherals available, like: capacitive touch, ADCs, DACs,

UART, SPI, I2C and much more.

It comes with built-in Hall Effect sensor and built-in temperature sensor.

Table 1: ESP 32 Specification

Number of cores	**2 (dual core)**
Wi-Fi	**2.4 GHz up to 150 M bits/s**
Bluetooth	**BLE (Bluetooth Low Energy) and legacy Bluetooth**
Architecture	**32 bits**
Clock frequency	**Up to 240 MHz**
RAM	**512 KB**
Pins	**30 or 36 (depends on the model)**
Peripherals	**Capacitive touch, ADC (analog to digital converter), DAC (digital to analog converter), I2C (Inter-Integrated Circuit), UART (universal asynchronous receiver/transmitter), CAN 2.0 (Controller Area Network), SPI (Serial Peripheral Interface), I2S (Integrated Inter-IC Sound), RMII (Reduced Media-Independent Interface), PWM (pulse width modulation), and more.**

➢ *Programming Environments:-*

The ESP32 can be programmed in different programming environments. You can use:

- ❖ Arduino IDE
- ❖ Espressif IDF (IOT Development Framework)
- ❖ Micro python

ESP32 Pin out Guide

The ESP32 has more GPIOs with more functionalities compared to the ESP826.

With the ESP32 you can decide

Which pins are

UART, I2C, or SPI - you just

Need to set that on the code.

This is possible due to the

ESP32 chip's multiplexing feature

That allows to assign multiple

Functions to the same pin.

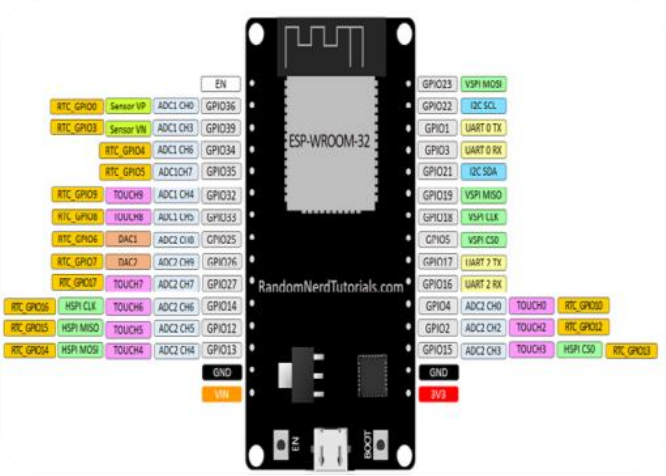


Fig 3.15 Configuration Esp. 32

If you don't set them on the code, the pins will be used as default - as shown in the figure below (the pin location can change depending on the manufacturer).

Types of Arduino Boards:-

Fig 3.16: Types of Arduino Board

Arduino Uno:-

One of the most popular Arduino boards out there is the Arduino Uno. While it was not actually the first board to be released, it remains to be the most actively used and most widely documented on the market.

Because of its extreme popularity, the Arduino Uno has a ton of project tutorials and forums around the web that can help you get started or out of a jam. We're big fans of the

Uno because of its great features and ease of use

Fig 3.17: Arduino Uno Pins

Board Breakdown:-

Here are the components that make up an Arduino board

1) ***Reset Button -*** This will restart any code that is loaded to the Arduino board
2) ***AREF -*** Stands for "Analog Reference" and is used to set an external reference voltage
3) ***Ground Pin -*** There are a few ground pins on the Arduino and they all work the same
4) ***Digital Input/output -*** Pins 0-13 can be used for digital input or output
5) ***PWM -*** The pins marked with the (~) symbol can simulate analog output
6) ***USB Connection -*** Used for powering up your Arduino and uploading sketches
7) ***TX/RX -*** Transmit and receive data indication LEDs
8) ***AT mega Microcontroller -*** This is the brains and is where the programs are stored
9) ***Power LED Indicator -*** This LED lights up anytime the board is plugged in a power source
10) ***Voltage Regulator -*** This controls the amount of voltage going into the Arduino board
11) ***DC Power Barrel Jack -*** This is used for powering your Arduino with a power supply
12) ***3.3V Pin -*** This pin supplies 3.3 volts of power to your projects

13) ***5V Pin -*** This pin supplies 5 volts of power to your projects

14) ***Ground Pins -*** There are a few ground pins on the Arduino and they all work the same

15) ***Analog Pins -*** These pins can read the signal from an analog sensor and convert it to digital.

How to Program Arduino:-

Once the circuit has been created on the breadboard, you'll need to upload the program (known as a sketch) to the Arduino. The sketch is a set of instructions that tells the board what functions it needs to perform. An Arduino board can only hold and perform one sketch at a time. The software used to create Arduino sketches is called the IDE which stands for Integrated Development Environment

➢ *Ultrasonic Sensor HC-SR04*

The HC-SR04 ultrasonic sensor uses sonar to determine distance to an object like bats do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package. It comes complete with ultrasonic transmitter and receiver modules.

Features:-

- ❖ Power Supply: +5V DC
- ❖ Quiescent Current: <2mA
- ❖ Working Current: 15mA
- ❖ Effectual Angle: <15°
- ❖ Ranging Distance: 2cm - 400 cm/1″ - 13ft
- ❖ Resolution: 0.3 cm
- ❖ Measuring Angle: 30 degree

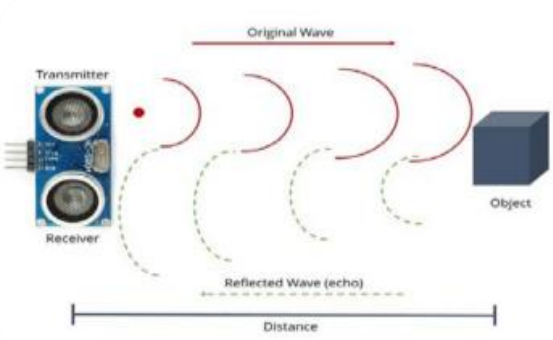


Fig 3.18

Ultrasonic Sensor

- ❖ Trigger Input Pulse width: 10uS
- ❖ Dimension: 45mm x 20mm x 15mm

How does it Work?

The ultrasonic sensor uses sonar to

Determine the distance to an object.

Here's what happens:

1) The transmitter (trig pin) sends a signal: a high- frequency sound.

2) When the signal finds an object, it is reflected and…

3) … the transmitter (echo pin) receives it.

HC-SR04 Ultrasonic Sensor Pin out:-

Pins:-

- ❖ VCC: +5VDC
- ❖ Trig : Trigger (INPUT)
- ❖ Echo: Echo (OUTPUT)
- ❖ GND: GND

Fig 3.19

Ultrasonic Sensor Pins

➢ *Dual H-bridge motor driver using L298N:-*

L298N Dual H Bridge DC Stepper Motor Drive Controller Board Module for Arduino

The L298 Stepper Controller makes it easy to drive either two DC motors or a bipolar stepper motor. This is a very high quality board and is very compact for designs where space really matters.

Fig 3.20:
Dual H-bridge motor driver

Features:

- ❖ Double H bridge drive
- ❖ Chip L298N (ST NEW)
- ❖ Logical voltage 5V
- ❖ Drive voltage 5V-35V
- ❖ Logic current 0mA-36mA
- ❖ Drive current 2A per channel (MAX single bridge)
- ❖ Max power 25W
- ❖ Weight 30g
- ❖ Small size 43*43*27mm (approx. 1.75" x 1.75" x 1")
- ❖ Compatible with L297/L298 driver

3.4.1 Temperature Measuring Sensor

DS18B20 Temperature Sensor

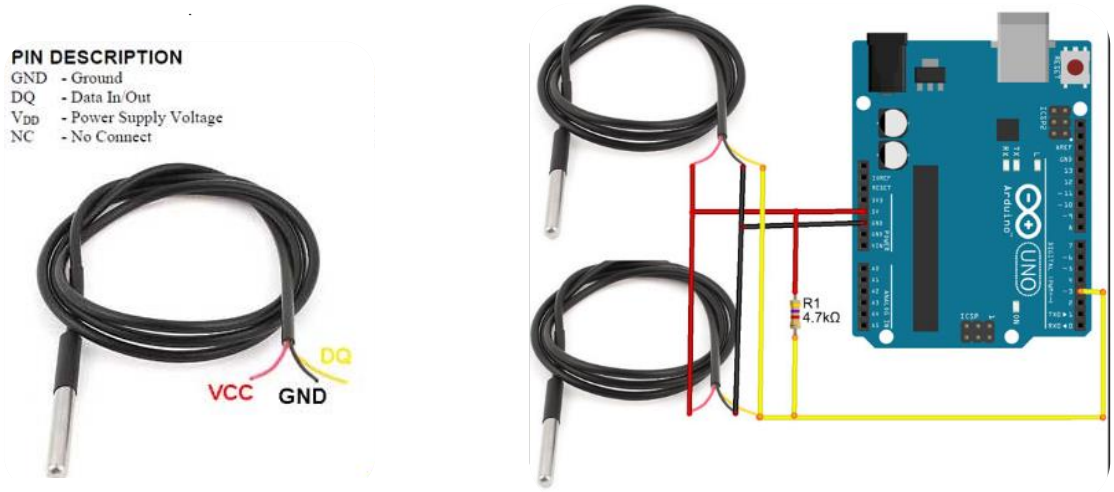


Fig 3.21: DS18B20 Temperature Sensor

The following table shows how you should wire the DS18B20 sensor to your board:

Table 2: DS18B20 Temperature Sensor Specification

DS18B20	Arduino
GND	GND
DQ	Any digital pin (with 4.7k Ohm pull-up resistor)
VDD	5V (normal mode) or GND (parasite mode)

DS18B20 Sensor Specifications

- Programmable Digital Temperature Sensor
- Communicates using 1-Wire method
- Operating voltage: 3V to 5V
- Temperature Range: -55°C to +125°C
- Accuracy: ±0.5°C
- Output Resolution: 9-bit to 12-bit (programmable)
- Unique 64-bit address enables multiplexing
- Conversion time: 750ms at 12-bit
- Programmable alarm options
- Available as To-92, SOP and even as a waterproof sensor

Description:

Pluggable terminal waterproof DS18B20 temperature sensor can be used in many places, such as soil temperature detection, hot water tank temperature control, waterproof DS18B20 temperature sensor must also be connected to a pull-up resistor can be used, for which we designed the converter to send use.

Product Specifications:

Temperature sensor supply voltage: 3.0V ~ 5.5V

Temperature sensor resolution: 9 to 12 adjustable resolution

Temperature range: -55 ~ +125 ° (lead can only withstand the highest temperature of 85 degrees)

Temperature Sensor Output Lead: Yellow (DATA) Red (VCC) and Black (GND)

Adapter Cables: DATA, VCC, BLK,

Suitable platform: for Arduino and Raspberry Pi.

Package Included:

- 1PC * DS18B20 temperature sensor
- 1PC * pluggable terminal adapter
- 1PC * 3pin cable

3.4.3 Positioning Determination (GPS Module)

Gps neo 6 mv2

- This module has an external antenna and built-in EEPROM.
- Interface: RS232 TTL
- Power supply: 3V to 5V
- Default baud rate: 9600 bps
- Works with standard NMEA sentences

The NEO-6M GPS module is also compatible with other Microcontroller boards. To learn how to use the NEO-6M GPS module with the Raspberry Pi, you can read: Email Alert System on Location Change with Raspberry Pi and GPS Module.

Pin out:

1) RX ==>> TX of ESP32 microcontroller
2) TX ==>> RX of ESP32 microcontroller
3) VCC 3.3 ~ 5 V
4) Gnd

Fig 3.22: Gps neo 6 mv2

➢ *Joystick*

Features

- ❖ Two independent Potentiometer: one for each axis (X and Y)
- ❖ Auto return to center position
- ❖ Low weight
- ❖ Cup-type Knob
- ❖ Compatible to interface with Arduino or with most microcontroller

Technical Specifications

- ❖ Operating Voltage: 5V
- ❖ Internal Potentiometer value: 10k
- ❖ 2.54mm pin interface leads
- ❖ Dimensions: 1.57 in x 1.02 in x 1.26 in (4.0 cm x 2.6 cm x 3.2 cm)
- ❖ Operating temperature: 0 to 70 °C

Pin out:

1. VCC: 5V
2. Gnd
3. VR X:
4. VR Y:

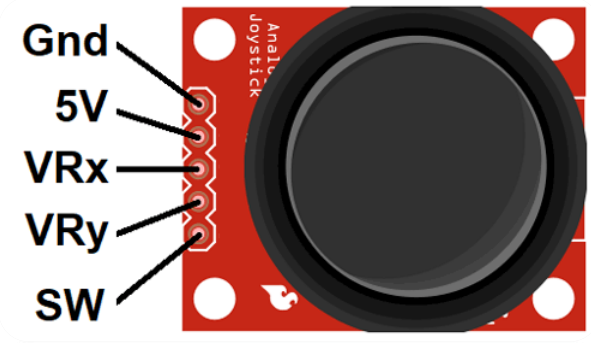


Fig 3.23: Joystick

➢ *Bluetooth hc-05*

HC-05 Technical Specifications

- ❖ Serial Bluetooth module for Arduino and other microcontrollers
- ❖ Operating Voltage: 4V to 6V (Typically +5V)
- ❖ Operating Current: 30mA
- ❖ Range: <100m
- ❖ Works with Serial communication (USART) and TTL compatible
- ❖ Follows IEEE 802.15.1 standardized protocol
- ❖ Uses Frequency-Hopping Spread spectrum (FHSS)
- ❖ Can operate in Master, Slave or Master/Slave mode
- ❖ Can be easily interfaced with Laptop or Mobile phones with Bluetooth
- ❖ Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.

HC-05 Equivalent Bluetooth Module

HC-02

Fig 3.24: Bluetooth hc-05

Pin Configuration

Table 3: Bluetooth hc-05 Pins Configuration

Pin Number	Pin Name	Description
1	Enable/ key	This pin is used to toggle between Data Mode (set low) and AT command mode (set high). By default it is in Data mode
2	Vcc	Powers the module. Connect to +5V Supply voltage
3	Ground	Ground pin of module, connect to system ground.
4	TX - Transmitter	Transmits Serial Data. Everything received via Bluetooth will be given out by this pin as serial data.
5	RX - Receiver	Receive Serial Data. Every serial data given to this pin will be broadcasted via Bluetooth
6	State	The state pin is connected to on board LED, it can be used as a feedback to check if Bluetooth is working properly.
7	LED	Indicates the status of Module ❖ Blink once in 2 sec: Module has entered Command Mode ❖ Repeated Blinking: Waiting for connection in Data Mode ❖ Blink twice in 1 sec: Connection successful in Data Mode
8	Button	Used to control the Key/Enable pin to toggle between Data and command Mode

➢ *Wheelchair*

❖ *Specifications:*

❖ *Motors:*

- ❖ Suggested Voltage: 4.5V DC (work well from 3-6V)
- ❖ No load Speed: 90±10rpm (speed depend on input voltage)
- ❖ No Load Current: 190mA (max.250mA)
- ❖ Torque: 800gf.cm min
- ❖ Stall current approx. 1A

❖ *Wheels:*

- ❖ 65mm diameter, 30mm width
- ❖ Plastic rims with solid rubber tires
- ❖ Caster Wheel in the front

❖ *Chassis:*

- ❖ Laser cut acrylic - 2 floor
- ❖ Metal standoffs caster wheel

Fig 3.25: Wheelchair

3.4.4 Monitoring Patient health Status, The Wheelchair balance and sending an Alert message in case of failure :

1-Establishing wireless network connection

- Set network ssid and password.
- Trying to connect to network.

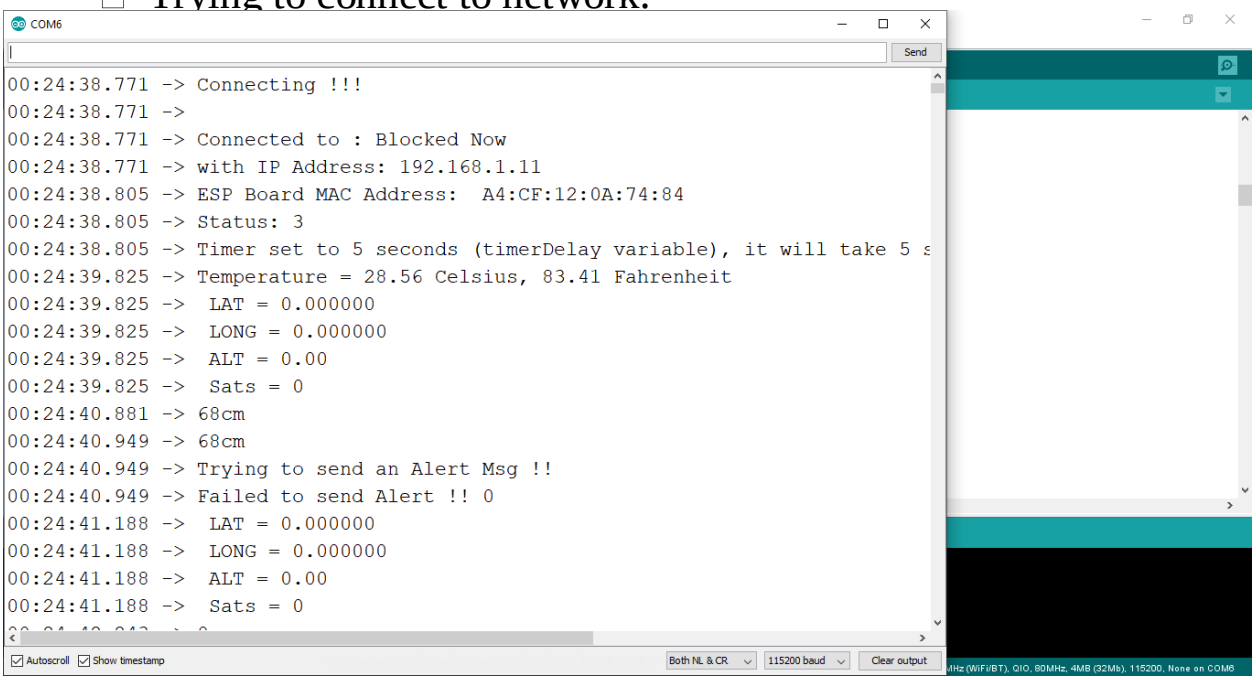


Fig 3.26: Connection ESP to Server

2-Reading data coming from sensors connected to the wheel chair:

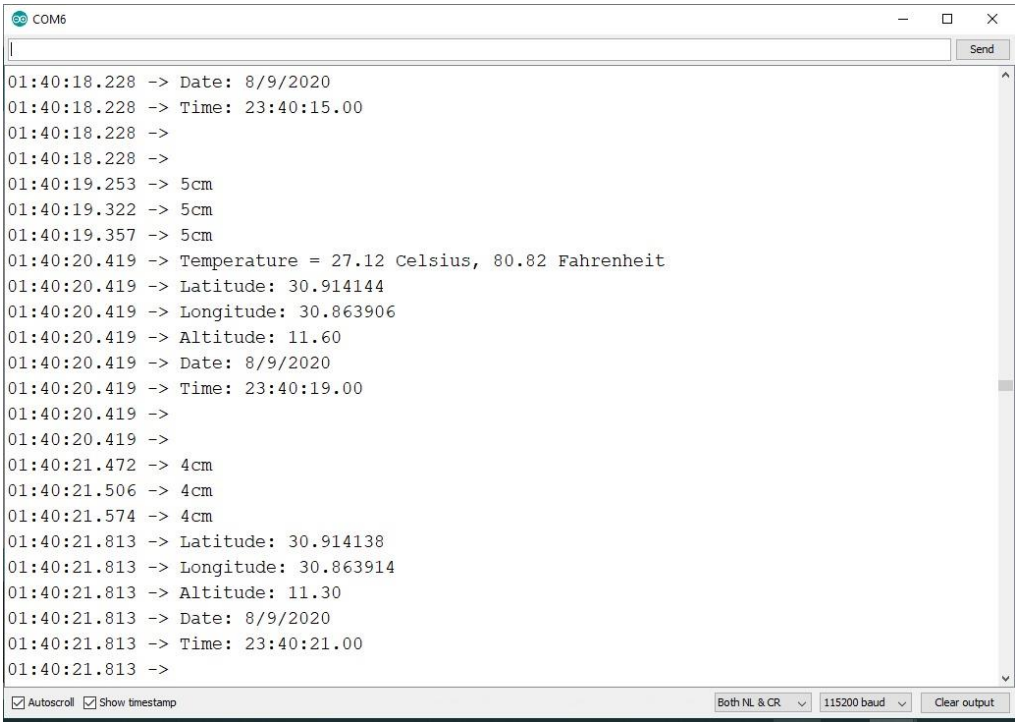


Fig 3.27: Reading data

Second case reading distance from ultrasonic

- Reading distance from ultrasonic sensor to check the balance of the wheel chair. We have two cases could happen :

  1- First case: if distance measured by the ultrasonic is less than or equal 10 cm and not equal zero in this Case the wheel chair is balanced, no error then Continue reading distance and not sanding any alert Message.

2- Second case: if distance measured by the ultrasonic is more than 10 cm or equal zero in this case the wheel chair is not balanced, an error happened the we need to send an alert message to a hospital which contains patient's location, temperature, time. And the wait 10 seconds to get a response from the hospital that received the alert message to stop sending an alert message again.

- Reading patient temperature at the same time of checking the wheel chair balance and also here we have to cases :

  1- First case if the temperature measured is more than 39℃ then the patient temperature is high so we need to send an alert message to a hospital to rescues him/her which contain it location ,temperature ,time . And the wait 10 seconds to get a response from the hospital that received the alert message to stop sending an alert message again.

  2- Second case if the patient's temperature measured is less than 38. In this case we don't need to send and alert message to a hospital so that sensor will continue in measuring the temperature.

Defining the global variables and set initial value to them

```
ESP32_-_endPoint_GetSensorsDataUpload
16 TinyGPSPlus gps;                                    // The TinyGPS++ object.
17 HardwareSerial SerialGPS ( 1 );
18 int distance = 0;
19 float lat= 0, lng = 0;
20 float celsius = 0 , fahrenheit = 0;
21 int ResponseCode_1 = 0 , ResponseCode_2 = 0;
22 unsigned long lastTime = 0;
23 unsigned long timerDelay = 5000;                    // Set timer to 5 seconds (5000)
24 float sensorLocData [2];
25 float sensorTempData[2];
26 const char* serverName_1 = "http://192.168.1.4:1880/update-sensor";
27 const char* serverName_2 = "https://healthcaare.000webhostapp.com/new.php";
28
29 OneWire  ds(2);
```

Fig 3.28: Defining variables

Functions headers and set up the bound rate of esp. 32 to 9600 and also bound rate of neo 6 m v2 GPS module , set up the wireless network connection .

```
ESP32_-_endPoint_GetSensorsDataUpload
29 OneWire  ds(2);
30 void initNet();
31 int sendSensorsData(String httpRequestData ,String serverName);
32 int sendData(String httpRequestData , String serverName);
33 void getLoc();
34 void getTempC_F();
35 int getDistance();
36 NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
37
38 void setup() {
39   Serial.begin(115200);
40   SerialGPS. begin ( 9600 , SERIAL_8N1, 16 , 17 );
41   initNet();
42 }
```

Fig 3.29: set up the bound rate of esp. 32 & neo 6 m v2 GPS module

Prepare data before sending in case of in balance of wheel chair of high temperature measured from the patient.


```
void loop() {
    getTempC_F();
    getLoc();
    distance = getDistance();

    sensorLocData [0]= lng;
    sensorLocData [1]= lat;
    sensorTempData[0]= celsius;
    sensorTempData[1]= fahrenheit;

    String httpRequestData = "sensor_1=Neo6-MV2GPS&lang="+String(sensorLocData[0],5)+
                             "&late="+String(sensorLocData[1],5)+"&sensor_2=DS18B20&tempC="+
                             String(sensorTempData[0],5)+"&tempF="+String(sensorTempData[1],5);
```

Fig 3.30: Sending Data high temperature to Server

Establishing a wireless Network connection

```
void initNet(){
  WiFi.begin(ssid, password);
  Serial.println("Connecting");
  while(WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.println("Connecting !!!");
  }
  Serial.println("");
  Serial.print("Connected to : ");
  Serial.println(WiFi.SSID());
  Serial.print ("with IP Address: ");
  Serial.println(WiFi.localIP());
  Serial.print("ESP Board MAC Address:  ");
  Serial.println(WiFi.macAddress());
```

Fig 3.31: Network connection

Sending patient data which contains last temperature measured and location, time date. Ensuring that the sent to the hospital successfully

```
//  Human_temp=40&latitude=30.784451&longitude=30.997881&time=1%3A30&date=2020-8-5
    String reqData = "Human_temp="+ String(celsius,4)+"&latitude="+String(lat,6)+"&longitude="+Stri

    if (getDistance()> 10 || getDistance()== 0 || celsius > 32 ){
        Serial.println ("Trying to send an Alert Msg !!");
        ResponseCode_1 = sendSensorsData(httpRequestData , serverName_1);
        ResponseCode_2 = sendData(reqData , serverName_2);
        if (ResponseCode_1 == 200){
        Serial.print("Alert Sent Successfully !!  ");
        Serial.print(ResponseCode_1 );                  //+ " , " + ResponseCode_2);
        Serial.println(ResponseCode_2 );
      }
        else if (ResponseCode_1 != 200){
        Serial.print("Failed to send Alert !! ");
```

Fig 3.32: Sending patient data

"Web service"

Web service is a standardized medium to propagate communication between the client and server applications on the World Wide Web. A web service is a software module that is designed to perform a certain set of tasks. **[4] [5]**

We use the web service to send patient data

(Human temperature, longitude, latitude... etc.) From the patient to the hospital server, show it on an interface and store it into database table.

The way of receiving a request:

1- The patient send data to the server in a get request, and the service receives it.

Receiving data:

```php
<?php
include("connection.php");

    $Human_temp= $_GET["Human_temp"];
    $latitude = $_GET["latitude"];
    $longitude = $_GET["longitude"];

       $date= $_GET["date"] ;
       $time= $_GET["time"] ;
      $res_time=0;
     $responded=0;
```

Fig 3.33: Receiving patient data

- Longitude and latitude are converted to a physical address by using an API.

```
15
16      function getAddress($RG_Lat,$RG_Lon)
17
18  {
19      $json = "https://nominatim.openstreetmap.org/reverse?format=json&lat=".$RG_Lat."&lon=".$RG_Lon."&zoom
            =27&addressdetails=1";
20
21      $ch = curl_init($json);
22      curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
23      curl_setopt($ch, CURLOPT_USERAGENT, "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:59.0) Gecko/20100101 Firefox/59.0");
24      $jsonfile = curl_exec($ch);
25      curl_close($ch);
26
27      $RG_array = json_decode($jsonfile,true);
28
29      return $RG_array['display_name'];
30      // $RG_array['address']['city'];
31      // $RG_array['address']['country'];
32  }
33
```

Fig 3.34: Converted to a Physical address

- Sending response code if the request is sent successfully
- Then, Data is stored in a database table

Inserting Data to Patient Table

```
$sql = "INSERT INTO patients( Human_temp,longitude,latitude,location_details,time,date,responded, response_time) VALUES
    ( '$Human_temp','$longitude ','$latitude','$addr','$time','$date','$responded','$res_time')";
if(mysqli_query($conn , $sql)){


exit();


} else{
    echo "ERROR: Could not able to execute $sql. " . mysqli_error($conn );
}

?>
```

Fig 3.35: Inserting Data to Patient Table

Patient table

Table 4: Patient Table

id	Human_temp	longitude	latitude	location_details	time	date	responded	response_time
1	38	29.965355353	30.9987655	الإسكندرية, Egypt / مصر	12:45:00	2020-09-08	1	06:16:48
2	38	29.965355353	30.9987655	الإسكندرية, Egypt / مصر	12:45:00	2020-09-08	1	06:16:48
3	38	29.965355353	30.9987655	الإسكندرية, Egypt / مصر	12:45:00	2020-09-08	1	06:16:48
4	38	30.34569	30.9987652	عزبة كوم أبو خليفة, البحيرة, Egypt / مصر	12:54:00	2020-08-09	1	06:16:48
5	38	30.34569	30.9987652	عزبة كوم أبو خليفة, البحيرة, Egypt / مصر	12:54:00	2020-08-09	1	06:16:48
6	38	30.34569	30.9987652	عزبة كوم أبو خليفة, البحيرة, Egypt / مصر	12:54:00	2020-08-09	1	06:16:48
7	38	30.34569	30.9987652	عزبة كوم أبو خليفة, البحيرة, Egypt / مصر	12:54:00	2020-08-09	1	06:16:48
8	38	29.965355353	30.9987655	الإسكندرية, Egypt / مصر	12:45:00	2020-09-08	0	00:00:00
9	38	29.965355353	30.9987655	الإسكندرية, Egypt / مصر	12:45:00	2020-09-08	0	00:00:00
10	38	29.965355353	30.9987655	الإسكندرية, Egypt / مصر	12:45:00	2020-09-08	0	00:00:00
11	38	29.965355353	30.9987655	الإسكندرية, Egypt / مصر	12:45:00	2020-10-08	0	00:00:00

- Data is shown on an interface of the hospital system in a table, the interface data is taken from database patients table.

Dashboard (Interface):

First, User should register an account to enter and handle the dashboard, He register his username, email and password, and then his data is stored in a database

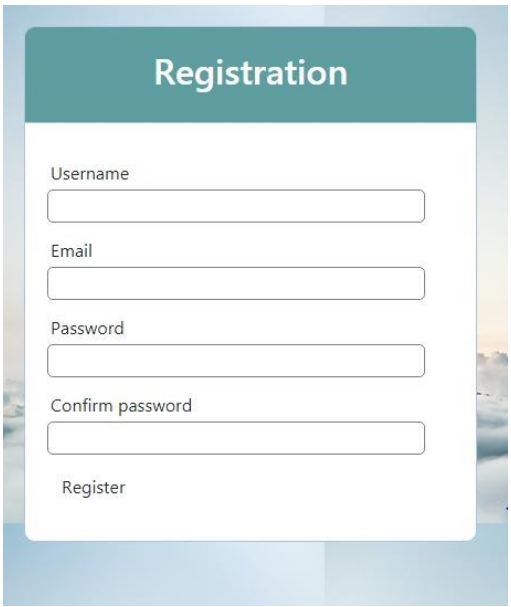


Fig 3.36: Registration form

Table of Database of Users

Table 5: Table of Database of Users

user_id	username	email	password
1	Ahmed	ahmedeldaly2110@gmail.com	81dc9bdb52d04dc20036dbd8313ed055
2	marwa	marwagalal6236@gmail.com	202cb962ac59075b964b07152d234b70
3	abdo	abdohamada1571998@gmail.com	f937b4bdf6f433022f9ceee12cf1a51c
4	mohamed	mohamed@gmail.com	827ccb0eea8a706c4c34a16891f84e7b

```
$username = "";
$email    = "";
$errors = array();
$usernameErr="";
$passwordErr="";

$passwordErr2="";


if(isset($_POST['reg_user']))
 {
  $username = ($_POST['username']);
  $email =  $_POST['email'];
  $password_1 = $_POST['password_1'];
  $password_2 = $_POST['password_2'];
  if (empty($username)) { array_push($errors, "Username is required");
  if (empty($email)) { array_push($errors, "Email is required"); }
  if (empty($password_1)) { array_push($errors, "Password is required"
  if ($password_1 != $password_2) {
    array_push($errors, "The two passwords do not match");
  }

  }else {
  $loginErr="Wrong username/password combination";

   }
  }
}
?>
```

Fig 3.37: Registration Code

Adding to Users Table

```
    if (count($errors) == 0) {
    $password = md5($password_1);//encrypt the password before saving in the database

    $query = "INSERT INTO users (username, email, password)
              VALUES('$username', '$email', '$password')";
    mysqli_query($conn, $query);
    $_SESSION['username'] = $username;
    $_SESSION['success'] = "Account is registered";
     echo "<script>
alert('registered successfully');
window.location.href='index.php';

</script>";
  }
 }
```

Fig 3.38: Adding to Users Table

Then, He should login with his account that He registered, the data is got from user's database table.

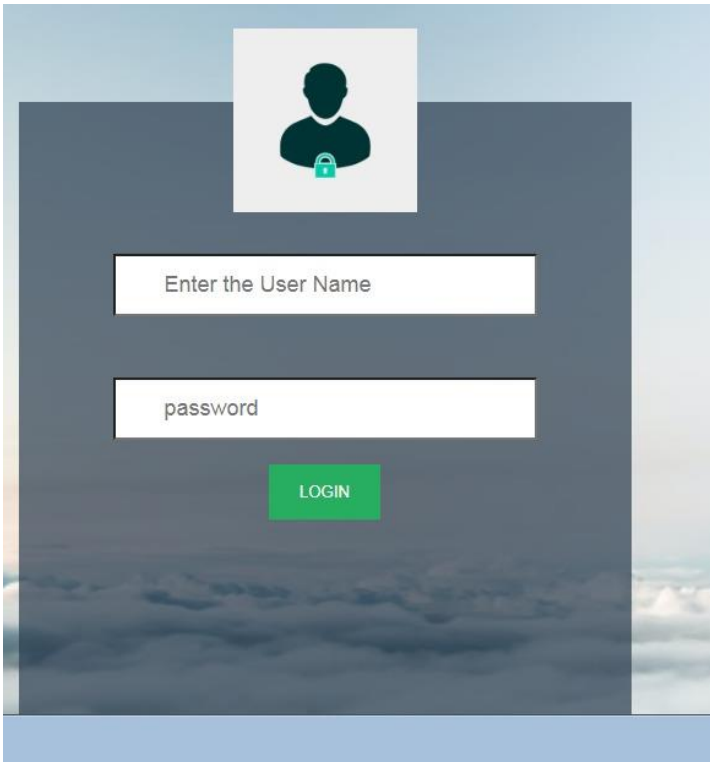


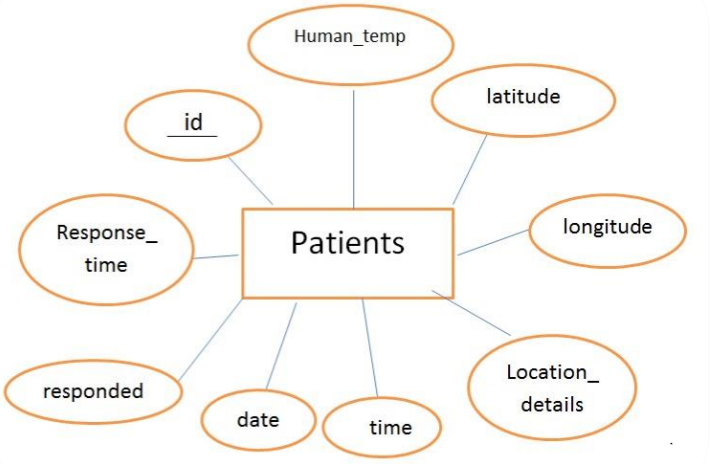
Fig 3.39: Login Form

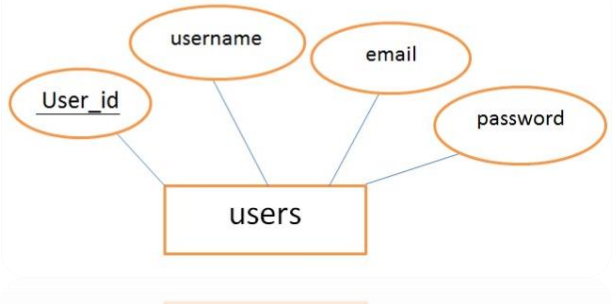
```php
if (isset($_POST['LOGIN'])) {
 $username = $_POST['username'];

  $_POST['password'];
 $password =$_POST['password'];

  if (empty($username)) {
        $usernameErr ="Username  is required";

  }
  if (empty($password)) {
     $passwordErr = "password   is required";
  }
```

Fig 3.40: Login Code

When users logs in interface He can see the table of patients data, When he clicks on a confirm button, responded column value in database updates from 0 to 1, that means he stored that the patient request is confirmed.

3.4.6 Analysis





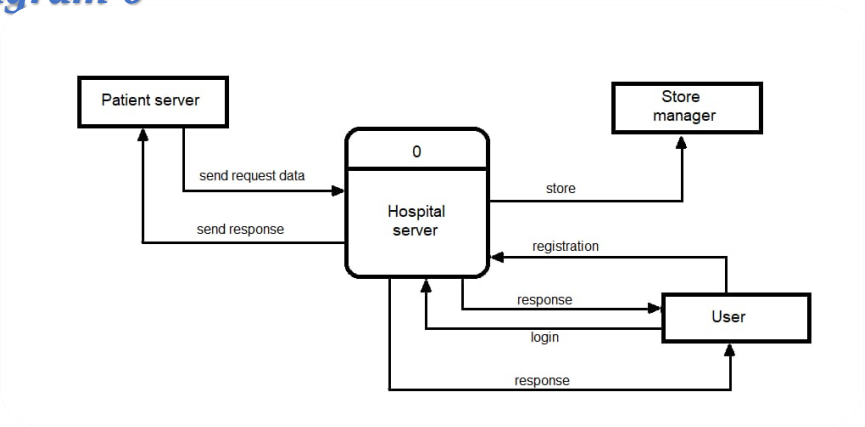
patients

id	Human_temp	latitude	longitude	location_details	time	date	responded	response_time

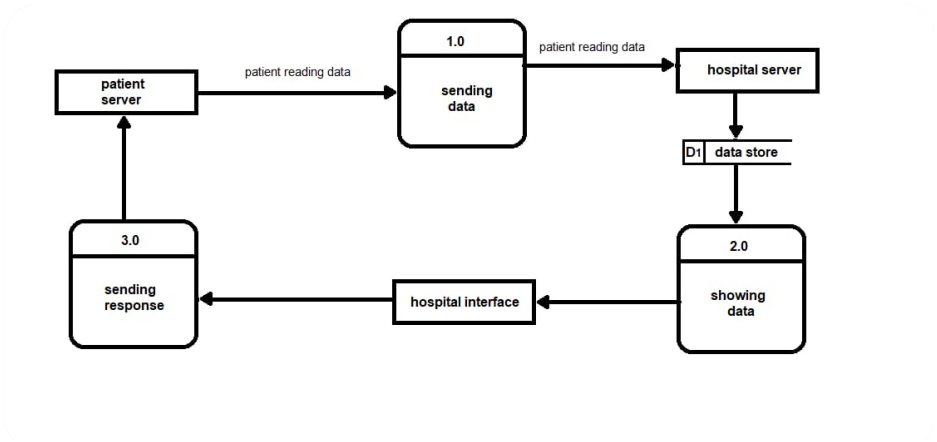
users

users_id	username	email	password

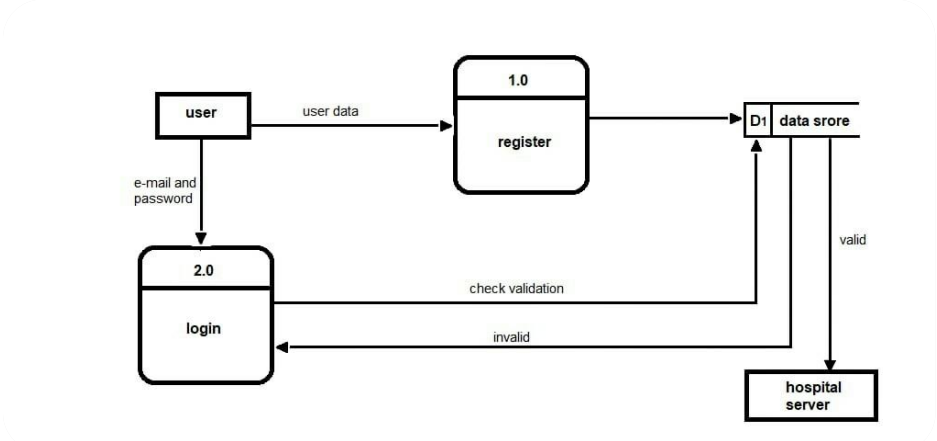
Context diagram 0

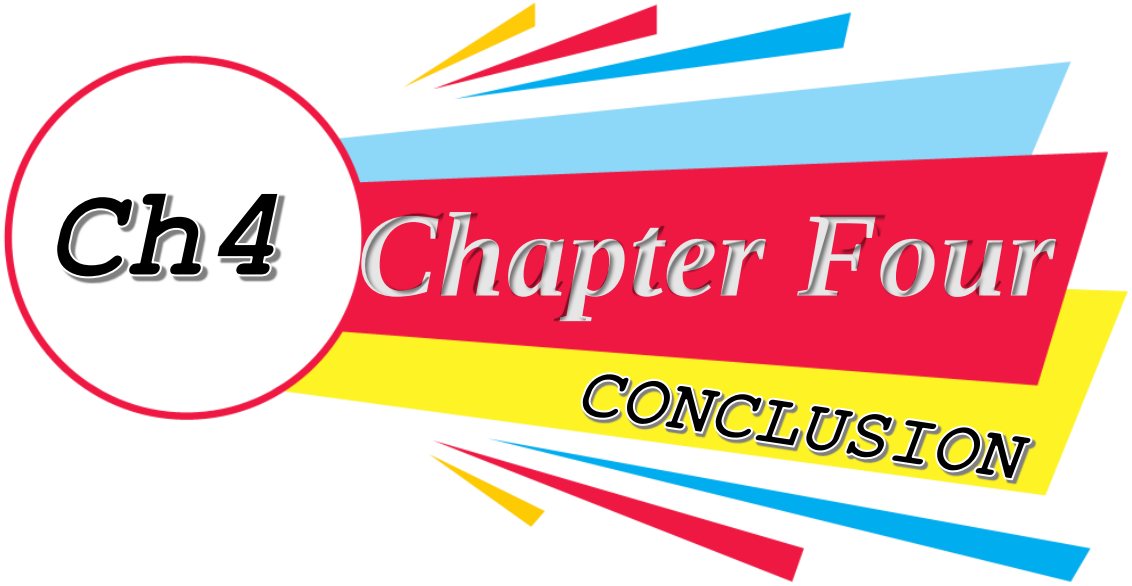


DataFlow diagram



DataFlow diagram(Login)




Ch4
Chapter Four
CONCLUSION

Chapter 4: CONCLUSION

Our project aims to help paralysis patients to move with high safety. we got them safety through using fall detection device by using ultra sonic sensor, Measuring his body temperature, and in case of measuring temperature exceeded 38°C , and also in case of falling, send to hospital and determine patient location so the hospital fastly sends response to the patient, and at the same time it sends rescue to him.

5.1 FUTURE WORK

We can add many extensions to our project in future.

- In the future, we could make the chair move by the patient's nerve signals.

- We can add a sensor to measure the heartbeat.

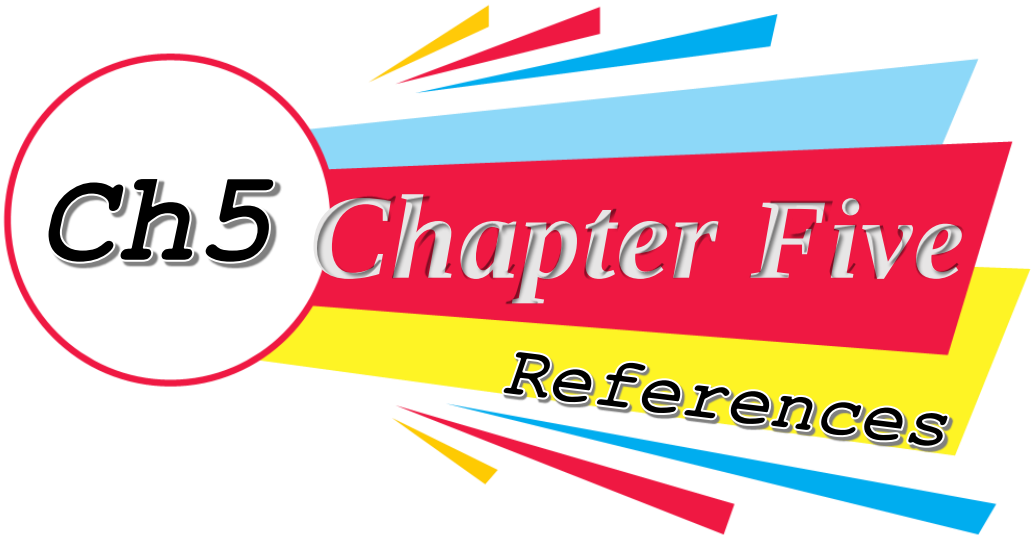
- We can add a sensor to measure the patient's blood pressure.

5.2 challenges

- One of the most difficult challenges we faced was the Corona virus and the neighborhood ban, which led to the difficulty of communicating with us as team members to complete the project.

- Voice commands are sent to the application via Google Voice and it is inaccurate, so wrong data is sent to the chair.

- Sometimes the GPS cannot analyze the coordinates of the location, so it sends the wrong location to the hospital.


Ch5
Chapter Five
References

Chapter 5: References

[1] https://en.m.wikipedia.org/wiki/Internet_of_things

[2] https://en.m.wikipedia.org/wiki/Embedded_system

[3] https://en.m.wikipedia.org/wiki/Web_service

[4] https://stackoverflow.com/

[5] https://www.geeksforgeeks.org/

- https://developer.android.com/
- https://www.arduino.cc/
- Android Programming: The Big Nerd Ranch Guide (Paperback) Book
- https://www.000webhost.com/
- PHP & MySQL Novice to Ninja
- Head First PHP & MySQL
- installing-the-esp32-board-in-arduino-ide
- esp32-pinout-reference-gpios
- https://www.arduino.cc/en/Main/Software
- https://ram-e-shop.com/product/kit-l298-red/
- https://randomnerdtutorials.com/guide-for-ds18b20-temperature-sensor-with-arduino/

ملخص المشروع

يوجد الكثير من مرضي الشلل الذين يعانون من صعوبة استخدام الكرسي لمتحرك التقليدي بالاضافه الي ان الكثير منهم ايضا يعانون من الوحده وعدم وجود من يعتني بهم، بناء على ذلك جاءت فكرة مشروعنا .

يساعد هذا المشروع مرضي الشلل الغير قادرين علي التحرك وكذلك لا يوجد لديهم من يرعاهم.

يتم تحريك الكرسي عبر عصا تحكم أو تطبيق Android **يستخدمه المريض لإعطاء أوامر حركة الاتجاهات الاربع (يسار أو يمين أو للأمام أو للخلف).**

يمكن للمريض أيضاً تحريك الكرسي عن طريق الأوامر الصوتية باستخدام نفس تطبيق Android.

الكرسي مزود بجهاز استشعار بالموجات فوق الصوتية يتم تثبيته تحت الكرسي ،

إذا سقط الكرسي في أي اتجاه ، فسيتم ارسال رسالة عبر المستشعر للمستشفي للحصول علي انقاذ.

يستخدم نظام تحديد المواقع العالمي (GPS) **لإرسال الموقع الدقيق للمريض المصاب حتى ترسل المستشفى شخصاً لإنقاذه.**

يحتوي الكرسي أيضاً على جهاز استشعار لاستشعار درجة حرارة جسم المريض.

يتم إرسال جميع بيانات المريض التي تقرأها أجهزة الاستشعار إلى المستشفى من أجل إرسال إنقاذ المريض عندما يحتاج إليه.

جامعة بنها **"CS13"** **كلية الحاسبات والذكاء الاصطناعي**

كرسي متحرك ذكي بتقنية انترنت الاشياء

مشروع التخرج المقدم استيفاءً جزئيًا لمتطلبات درجة بكالوريوس الحاسبات والذكاء الاصطناعي

قسم علوم الحاسب

فريق المشروع

- ✓ **أحمد السيد صبرة**
- ✓ **أحمد رأفت عبد الوهاب**
- ✓ **أحمد عوني أحمد الدالي**
- ✓ **عبدالرحمن حمدان عبدالرحمن علي**
- ✓ **مروة اشرف الشحات**
- ✓ **مروة محمد جلال**

تحت إشراف

د/فاطمة صقر