PROJECT REPORT

ON

HAND GESTURE CONTROLLED WHEELCHAIR WITH OBSTACLE DETECTION

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Bachelor of Engineering

In

Electronics and Telecommunication

By

Rutik Yogesh Andhalkar (Exam No - B190353001)
Aavesh Mansur Bagwan (Exam No - B190353006)
Dheeraj Dattatray Shinde (Exam No - B190353102)
Prathamesh Chandrakant Shinde (Exam No - B190353103)

Under the Guidance of

Mr. R. S. Piske

Department of Electronics and Telecommunication Engineering,

VIDYA PRATISHTHAN'S KAMALNAYAN BAJAJ INSTITUTE OF ENGINEERING & TECHNOLOGY, BARAMATI

2022-23

Affiliated to

Savitribai Phule Pune University

Department of Electronics and Telecommunication Engineering,

Vidya Pratishthan's Kamalnayan Bajaj Institute of Engineering & Technology, Baramati

CERTIFICATE

This is to Certify that the Project Report Entitled

"HAND GESTURE CONTROLLED WHEELCHAIR WITH OBSTACLE DETECTION"

Submitted By

Rutik Yogesh Andhalkar (Exam No - B190353001)

Aavesh Mansur Bagwan (Exam No - B190353006)

Dheeraj Dattatray Shinde (Exam No - B190353102)

Prathamesh Chandrakant Shinde (Exam No - B190353103)

is a bonafied work carried out satisfactorily by them under supervision and guidance and it is submitted towards the partial fulfilment of the requirements of Savitribai Phule Pune University, Pune for the award of degree, Bachelor of Engineering (Electronics and Telecommunication) during the academic year 2022-23.

Place: Baramati
Date:

Mr. R. S. Piske
Guide

Dr. B. H. Patil
Head of the Department

Dr. R. S. Bichkar
Principal
VPKBIET, Baramati.

ACKNOWLEDGMENT

In this project we have made a prototype for the hand gesture controlled wheelchair with obstacle detection capability, we would like to express our wholehearted gratitude to the Honourable Principal, **Dr.R.S.Bichkar** and Head of Eletronics and Telecommunication Department, **Dr.B.H.Patil** for there unwavering support in several ways in building this project. We would also like to thank our project guide **Mr.R.S.Piske** and project co-ordinator **Mr.S.D.Biradar** for there patience, helpful information, practical advice, valuable guidance and unconditional support in the process of developing project which helped us immensely throughout the journey.

We also want to express our gratitude towards our friends for being with us and assisting us in very minor to very major help in the developement phase of the wheelchair. We have been very thankful to our family members for keeping our moral high through there verbal support both family and friends plays an extremely essential role in the process.

Lastly, we would like to express my deepest gratitude to the entire engineering community for their continuous efforts in pushing the boundaries of knowledge and innovation. The collective knowledge and inspiration provided by researchers, scientists and engineers have paved the way for projects like ours and we are honored to be a part of this community.

This project has been an incredible learning experience and we are grateful to everyone who has contributed to its success. Their support and guidance have shaped not only the outcome of this project but also our personal growth as an engineer. Thank you all for being a part of this remarkable journey.

ABSTRACT

People with disabilities, especially those who have lost their mobility due to spinal cord injuries, cerebral palsy or muscular dystrophy often face difficulties in controlling their wheelchair independently. Currently, most wheelchairs are controlled using a joystick, which can be challenging for people with limited hand dexterity or hand-eye coordination. As a result, they may require assistance from a caregiver or may be unable to move around freely. Therefore, there is a need for an alternative wheelchair control system that is more intuitive and accessible for people with disabilities. This project demonstrate that how can we design and implement the hand gesture controlled wheelchair for the physically challenged person who cannot move by himself. The main focus of this project is to detect the motion of the wrist of the hand and control the motors accordingly to move in desired direction along with some additional features such as, obstacle detection for the help of operator/patient if needed. Previously we needed some other person to move the wheelchair in desired direction but later, we developed the "joystick controlled wheelchair" which are good in its own ways but most of the times it becomes difficult for the physically challenged person to use it efficiently to overcome this situations we have developed this "gesture controlled wheelchair with obstacle detection" and among all the available gestures we found that this hand gesture controlled wheelchair will be the best suited for the physically challenged person. For this demonstration of the project we have used ESP32, Accelerometer sensor, Motors and motor driver, Ultrasonic sensor for obstacle detection.

Contents

List of Figures

List of Tables

List of Abbreviations

	Abbreviation	Meaning
1.	SoC	System On Chip
2.	GPIO	General purpose Input Output
3.	IoT	Internet Of Things
4.	ADC	Analog to Digital Converter
5.	IDE	Integrated Developement Environment
6.	I2C	Inter-Integrated Circuit
7.	SPI	Serial Peripheral Interface
8.	RF	Radio Frequency
9.	SPST	Single Pole Single Throw
10.	PWM	Pulse Width Modulation
11.	IC	Integrated Circuits
12.	DIY	Do It Yourself
13.	ESP-NOW	ESP Wireless Over Network
14.	API	Application Programming Interface
15.	RPM	Revolutions Per Minute

1 INTRODUCTION

About 10% of the global population, i.e. about 650 million people, have disabilities and suffering from various challenges in the daily life, some of them are by birth due to certain neural issues are physically challenged and some of them are physically challenged due any accidents and due to increased rate of accidents there is high incremental rate of increase of physically challenged people. Hence it is extremely essential to build a system for them so that they can move by themselves without need of any other persons intervention and live more independently, flexibly and be self dependent for physical movement.

The need for a Hand Gesture Controlled Wheelchair with Obstacle Detection for physically challenged individuals arises from the desire to enhance their mobility, independence and overall quality of life. Traditional wheelchair controls often require manual dexterity and physical strength, which can be challenging or impossible for individuals with severe physical disabilities. By incorporating hand gesture recognition technology, this innovative wheelchair offers an intuitive and alternative control mechanism. It enables users to operate the wheelchair effortlessly by simply gesturing with their hands, eliminating the need for complex joystick controls or manual pushing. Moreover, obstacle detection plays a crucial role in ensuring the safety and well-being of wheelchair users. Navigating through crowded areas or maneuvering in tight spaces can be daunting tasks for individuals with limited mobility. By integrating obstacle detection sensors, the wheelchair can detect potential barriers in its path, such as furniture, uneven surfaces or unexpected obstacles. This real-time detection allows the wheelchair to autonomously adjust its course or provide feedback to the user, enabling them to avoid collisions and navigate with confidence. The Hand Gesture Controlled Wheelchair with Obstacle Detection addresses the pressing need for accessible and user-friendly mobility solutions for physically challenged individuals. By empowering users to control their wheelchair using natural hand gestures and incorporating obstacle detection capabilities, this technology aims to enhance their independence, promote inclusion and improve their overall well-being.

Previously a concept of controlling the wheelchair by the help of the motion or gesture of body developed during world war II to assist the injured veterans it was developed by the George Klein in 1953 but this has no any major security intelligent system which will help the operator. Since then many research has been done in this area. The concept of a hand gesture-controlled wheelchair with obstacle detection emerged as a response to the challenges faced by individuals with limited mobility. Engineers and researchers recognized the need for intuitive control interfaces that would empower users to navigate their surroundings independently. There are various motion controlled wheelchairs present such as eye movement controlled wheelchair, joystick controlled wheelchair, voice controlled wheel chair, head motion controlled wheelchair or even wheelchairs which are controlled by brain. But most of them have certain issues such as eye movement controlled wheelchair will require a screen in front of eyes of operator all the time which is cumbersome and also if operator is away from the wheelchair operator cannot use eye controlled wheelchair because he or she does not have that screen which detects eye motion in front of him, the voice controlled wheelchair has issue of understanding different voices around it which will cause a disturbance in an noisy environment and if operator is away from the wheelchair voice will also not be able to reach to the wheelchair. Joy stick controlled wheelchair is also not a proper solution because it is also not a convenient to move that joystick all the time and if operator is away from the wheelchair, operator does not have joystick to control it, even head motion controlled wheelchair become cumbersome most of the time its not a convenient to move head all the time, and brain reading wheelchairs are also under development but they are very expensive that's why most of the people simply cannot afford it.

To overcome all above mentioned important issues we found that using hand gesture controlled wheelchair can be very useful and convenient in all the condition for movement. The Hand Gesture-Controlled Wheelchair with Obstacle Detection project aims to address the challenges faced by individuals with physical disabilities, particularly those with limited upper body mobility. The hand gesture-controlled wheelchair system promotes user autonomy and reduces the reliance on caregivers or attendants for routine wheelchair operations. The hand gesture-controlled wheelchair system offers an alternative mode of control that does not require physical contact, eliminating the need for complex manual controls or joysticks. The system interprets specific hand gestures and translates them into corresponding wheelchair movements providing users with a natural and efficient means of controlling their mobility such by moving wrist of the hand of patient to the right, left, top, down, wheelchair can be moved to the right, left, backward and forward respectively. In this we have two circuits one which is gesture recognition circuit which is set on the hand of the operator which detects the gesture of the hand by accelerometer sensor and then transfers the data through wireless communication to the wheelchair moving circuit which is present at the wheelchair and then wheelchair moving circuit controls the motors using motor driver accordingly and for the obstacle detection we have used ultrasonic sensor at the wheelchair moving circuit which detects the obstacle and reports to the wheelchair moving circuit to stop the wheelchair. The Hand Gesture Controlled Wheelchair with Obstacle Detection marks a significant milestone in the realm of assistive technology, offering a cutting-edge solution for individuals with limited mobility. This innovative wheelchair system paves the way for enhanced independence and improved quality of life. With its robust obstacle detection capabilities and seamless integration of user-friendly controls, this project sets the stage for a future where individuals can navigate their surroundings effortlessly, granting them the freedom to explore the world around them with newfound confidence.

1.1 Motivation

The motivation for the project "Hand Gesture Controlled Wheelchair with Obstacle Detection" is driven by the desire to provide people with disabilities greater independence and mobility. The current joystick-based control system for wheelchairs can be difficult for individuals with limited hand dexterity or hand-eye coordination, and may require assistance from a caregiver.

By developing a hand gesture controlled wheelchair system, users can control the movement of their wheelchair more intuitively and independently. This system can provide a more accessible means of mobility for people with disabilities, allowing them to move around more freely and improve their overall quality of life.

In addition to providing greater independence, the proposed system includes obstacle detection capabilities, which can improve safety and reduce the risk of accidents. The system's sensors can detect obstacles in the user's path and alert them, giving them more time to react and avoid potential collisions. This obstacle detection capability can also reduce the burden on caregivers who may need to constantly monitor and assist wheelchair users in navigating through obstacles.

Furthermore, the development of a hand gesture controlled wheelchair system with obstacle detection has the potential to contribute to the broader field of assistive technology and robotics. This project can serve as a starting point for developing similar systems for other assistive devices, including robotic prosthetics, exoskeletons and other mobility aids. The motivation behind developing a gesture-controlled wheelchair with obstacle detection for our final year engineering project stems from the desire to empower individuals with mobility impairments. We believe that everyone deserves the freedom to move independently and engage with the world around them. By harnessing the power of intuitive hand gestures, we aim to create a revolutionary solution that not only provides seamless maneuverability but also ensures the safety of the user by incorporating advanced obstacle detection technology. Our project is driven by the passion to make a meaningful impact in the lives of those facing mobility challenges, allowing them to regain control, explore their surroundings and embrace a newfound sense of freedom. Overall, the motivations for the project "Hand Gesture Controlled Wheelchair with Obstacle Detection" are centered around improving the quality of life and independence of people with disabilities, increasing safety and contributing to the development of assistive technology and robotics.

1.2 Objective

Develop a hand gesture controlled wheelchair system that provides physically challenged individuals with a more independent means of mobility. Design a cost-efficient and user-friendly solution that can be easily adopted by people with disabilities, without requiring extensive training or assistance. Reduce the physical efforts required by disabled individuals in controlling their wheelchair, enabling them to move around more easily and comfortably. Incorporate obstacle detection capabilities into the system, enhancing safety and reducing the risk of accidents.

Implement sensors or wearable devices to capture and interpret hand gestures accurately. Train the system to recognize a set of predefined hand gestures that correspond to specific wheelchair commands. Integrate the hand gesture recognition system with the wheelchair control mechanism. Establish a communication interface between the gesture recognition system and the existing wheelchair control system. Map recognized hand gestures to appropriate wheelchair movements, such as forward, backward, turning and stopping. Ensure a seamless and responsive interaction between hand gestures and wheelchair control. Implement obstacle detection and collision avoidance capabilities. Integrate ultrasonic sensors to detect obstacles in the wheelchair's path.

Improve the overall quality of life for people with disabilities by providing them with a more accessible and intuitive means of mobility. By pursuing these objectives, the project on hand gesture-controlled wheelchair with obstacle detection aims to explore innovative approaches, improve system performance and address broader social and psychological aspects, resulting in a more comprehensive and impactful solution for individuals with physical disabilities.

1.3 Block Schematic of Proposed System

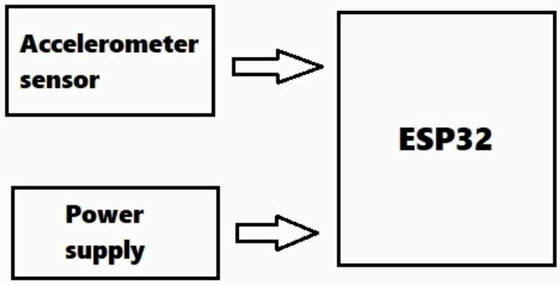


Figure 1: Block Diagram of Gesture Recognition Circuit

For the processing of the signals taken from the accelerometer sensor we need a microcontroller for that purpose in this project we have used ESP32 microcontroller board which is powered by the lithium ion battery at the transmitter end, we have accelerometer sensor which measures change in acceleration due to gravity this will helps in detecting the motion of the hand for gesture detection.

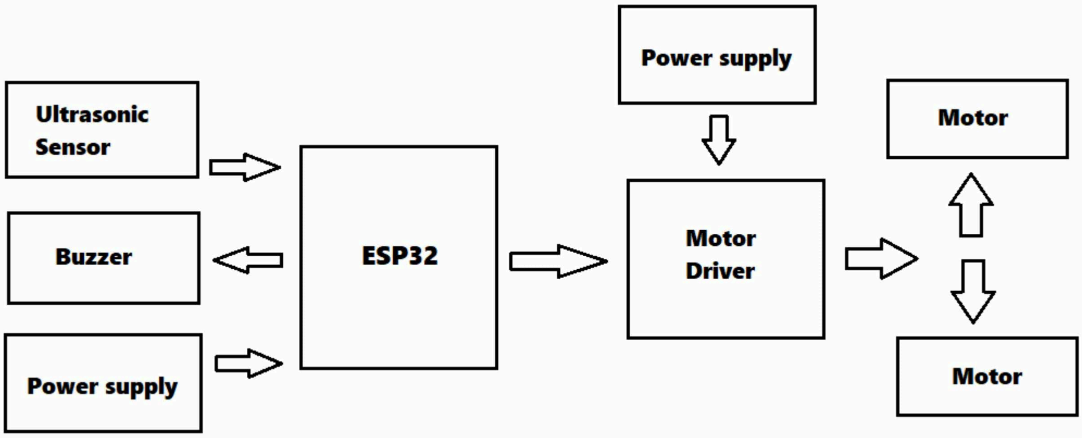


Figure 2: Block Diagram Of Wheelchair Moving Circuit

At here we have used same microcontroller board which is ESP32 for the processing of the signal here one signal is coming from the accelerometer sensor present at the transmitter section and the other signal is from ultrasonic sensor which is used for the obstacle detection, we have also used buzzer for the beeping when the obstacle is detected, we have the motors we need to move both of them backwards and the forwards but for that we are using H-bridge motor driver circuit as shown in the diagram, to supply power to all circuit we are using the lithium-ion batteries.

2 LITERATURE REVIEW

The literature review for hand gesture-controlled wheelchair reveals that there has been significant research conducted on this topic. The main objective of this research is to develop a user-friendly and efficient means of mobility for individuals with disabilities. The following review explores some of the key findings from various research papers on this topic.
Krunal Bansod, Kushal Asarkar, Mandar Topre, Vikrant Raj proposed a "Hand Gesture Controlled Wheelchair" and published paper at International Research Journal of Engineering and Technology Volume 07, Issue 07, July 2020. this uses arduino nano to process the signals produced by the accelerometer sensor and then used RF transmitter to transmit the signals then at the receiver they have used arduino uno for processing at the receiver section where RF receiver will receive the signals and command to the arduino for the desired movement, here they have used proximity sensor for obstacledetection and here they have used buzzer as well for beepsound to convey the presence of obstacle.
Mufrath Mahmood, Md. Fahim Rizwan, Masuma Sultana, Md.Habib, Mohammad Hasan Imam proposed "Design of a low cost Hand Gesture Controlled Automated Wheelchair" and published a conference paper at IEEE Region 10 Symposium in June 2020. Here they have developed a low cost hand gesture-controlled automated wheelchair. In this system Arduino nano is used to read data from the accelerometer sensor and then transmit it via RF transmitter, arduino mega is used to control the motors depending upon the signals coming from the transmitter and arduino mega along with the GSM module helps in emergency messaging system. Arduino nano and an ultrasonic sensor is used for the obstacle detection at receiver end and here they have implemented GPS location tracker using NodeMCU.
K.S Vairavel and R.Nevetha developed a "Hand Gesture Wheelchair Control Using Raspberry Pi" and published paper in the International Journal of Scientific and Technology Research, Volume 09, Issue 04, April 2020. here they presents a novel approach for controlling a wheelchair using hand gestures and Raspberry Pi. The authors address the limitations of conventional wheelchair control methods and propose a system that utilizes hand gestures for improved mobility. They use a Raspberry Pi microcontroller along with a camera to capture and interpret the hand gestures made by the user. The captured hand gestures are processed and mapped to specific commands for wheelchair movement. The paper discusses the hardware and software components used in the system, including the Raspberry Pi, camera module and image processing algorithms. The authors also present the methodology used for detecting and recognizing different hand gestures,which involves capturing

images, segmenting the hand region, and applying pattern recognition techniques. To validate their approach,the authors conducted experiments and evaluated the performance of the system. They measured the accuracy of gesture recognition and assessed the response time of the wheelchair to the detected gestures. The results demonstrate the feasibility and effectiveness of the proposed hand gesture control system.

Shayban Nasif and Muhammed Abdul Goffar Khan had developed a "Wireless Head Gesture Controlled Wheelchair for Disabled Persons" and published conference paper at IEEE region 10 Humanitarian technology conference in 2017. here a system is based on an Arduino microcontroller, and a RF transmitter and Receiver is used for wireless communication. The user's head gestures are captured by an accelerometer and gyroscope, and the wheelchair is controlled accordingly. The authors claimed that this system can be a cost-effective and efficient solution for individuals with physical disabilities.

Pushpendra Jha had developed a "Hand Gesture Controlled Wheelchair" and published paper at International Journal of Scientific & Technology Research, Volume 05, Issue 04, April 2016. where he had used an ADXL335 as accelerometer sensor which will be giving analog signal on moving it in X, Y, Z axis respectively. An LM324 operational amplifier is used as a comparator to convert the analog signal into the digital signal. Radio Frequency transmitter of 434 MHz frequency is used to transmit the signal wirelessly. Before sending, the data is encoded with an encoder IC HT12E. If the input data matches the preinstalled data then the signal is given to L293D IC on receiving the signal the L293D IC gives the signal to relays and then the wheelchair starts moving.

Revati A. Solanke and Arti R. Salunke had developed "Voice and Gesture Based Wheelchair for Physically Challenged Using AVR and Android" and published paper at the International Research Journal of Engineering and Technology, Volume 03, Issue 04, April 2016. here they presents a system that combines voice and gesture recognition for wheelchair control, using AVR microcontroller and Android platform. The authors address the need for an intuitive and user-friendly control mechanism for physically challenged individuals and propose a solution that allows wheelchair control through voice commands and hand gestures. They utilize an AVR microcontroller for processing the commands and an Android smartphone for capturing and interpreting voice and gesture inputs. The paper describes the hardware and software components used in the system, including the AVR microcontroller, Android smartphone, and relevant programming languages and libraries. The authors explain the integration of these components to establish communication between the wheelchair and the user's voice and gesture inputs. The methodology for voice and gesture recognition is discussed in detail. The authors explain the process of capturing voice commands through the android smartphone and using speech recognition algorithms to convert them into wheelchair movement commands. They also outline the process of capturing hand gestures using the smartphone's camera and utilizing image processing techniques for gesture recognition.

Prof. Vishal V. Pande, Nikita S. Ubale, Darshana P. Masurkar, Nikita R. Ingole, Pragati P. Mane had developed "Hand Gesture Based Wheelchair Movement Control for Disabled Person Using MEMS (Micro Electro-Mechanical Systems)" and published paper at International Journal of Engineering Research and Applications, Volume 04, Issue 04, April 2014. Here they had used ADXL202E for gesture detection at the transmitter, The ADXL202E can measure both dynamic acceleration (e.g., vibration) and static acceleration (e.g., gravity). for the wireless transmission they have used RF ASK Module It is an effective low cost solution to receiving data at 315/433MHz.

Overall, the literature review highlights that hand gesture controlled wheelchair technology has evolved significantly over the years. The systems proposed in various research papers have the potential to provide better mobility and user experience for individuals with disabilities. However, there is a need for further research to improve the accuracy and reliability of gesture recognition and to develop more affordable and user-friendly solutions.

3 SYSTEM DEVELOPMENT

For the implementation of the concept, which is hand gesture controlled wheelchair with obstacle detection we need to make two circuits one for the capturing gesture of the hand made by operator and other for the movement of the wheelchair according to that gestures and we need to establish a wireless communication between these two circuit for transferring signals from hand gesture capturing circuits to wheelchair moving circuit.

3.0.1 Gesture Recognition Circuit

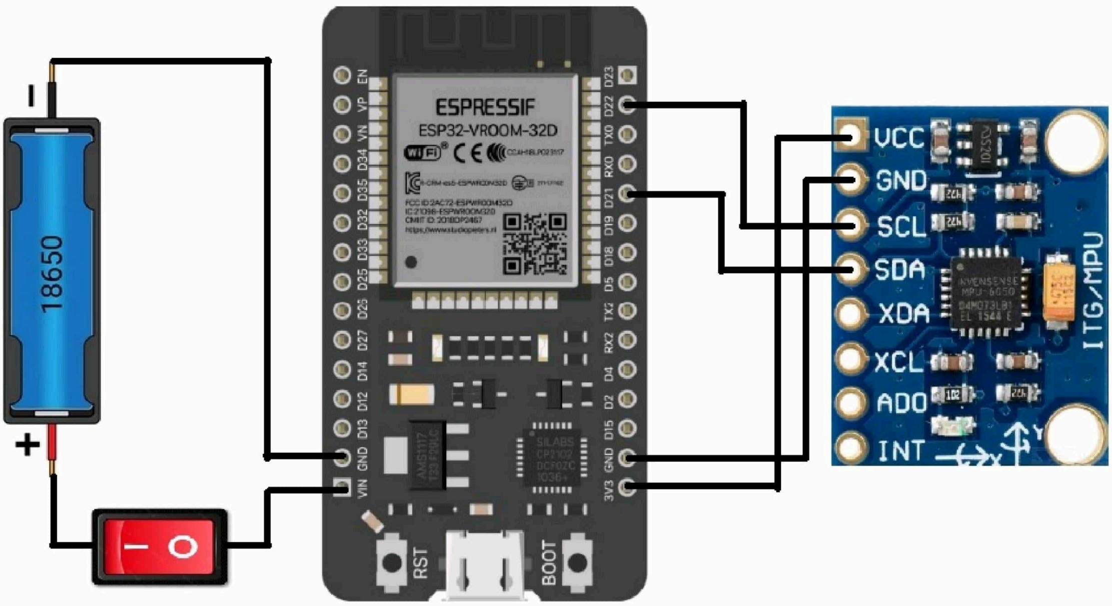


Figure 3: Circuit Diagram Of Gesture Recognition

This is a gesture recognition circuit which deals with the detecting gestures of the hand and then this circuit send commands to the wheelchair moving circuit, Here we need a sensor or a mechanism which can detect the gestures of the hand for that purpose we are using accelerometer sensor which deals with the measurements of the change in acceleration due to gravity done by itself. So if we put this on the hand of the operator we can easily detect the hand motion by measuring change in acceleration due to gravity made by itself.

Here we are using MPU6050 sensor as accelerometer sensor and we are communicating with this via inter-integrated communication protocol, then also we need a processing unit a microcontroller which can process the data and then transfer the data coming from sensor unit to the wheelchair moving circuit, for that purpose we use ESP WROOM 32 board

which contains 32bit ESP32 SoC and 30 IO pins which are used for interfering sensors with it. The microcontroller includes 520KB of SRAM for data storage and 4MB of Flash memory for program storage. The ESP32 supports multiple communication protocols such as UART,SPI,I2C,I2S,and CAN, enabling seamless integration with various peripherals and sensors. It offers Wi-Fi connectivity with support for 802.11 b/g/n standards, allowing for wire-less communication and Internet connectivity. The ESP32 also includes Bluetooth functionality, supporting both Bluetooth Classic and Bluetooth Low Energy (BLE) protocols. The ESP32 can be programmed using different programming languages, including C, C++ and MicroPython.
We are supplying power to the ESP32 via Vin pin using 3.7v lithium-ion battery and controlling it via switch.we are supplying power to the MPU6050 with the 3.3v pin of ESP32 board and for I2C communication with MPU6050. We are using D22 and D21 pins of the ESP32 and for the wireless communication we are using ESP-NOW protocol developed by espressif which is a fast communication protocol that can be used to exchange small messages (up to 250 bytes) between ESP32 boards. ESP-NOW is very versatile and we can have one-way or two-way communication in different setups.

3.0.2 Wheelchair Moving Circuit

This section will respond to the signals received from the gesture recognition circuit, Here for the processing of the data and controlling of the motors where we need to move wheelchair both forward, backward, right and left for that we need motor controlling operation and for that we are using ESP32 microcontroller board which contains 32bit ESP32 SoC and 30 IO pins which are used for interfering sensors with it. The microcontroller includes 520KB of SRAM for data storage and 4MB of Flash memory for program storage. The ESP32 supports multiple communication protocols such as UART, SPI, I2C, I2S,and CAN enabling seamless integration with various peripherals and sensors. It offers Wi-Fi connectivity with support for 802.11 b/g/n standards, allowing for wire-less communication and Internet connectivity. The ESP32 also includes Bluetooth functionality, supporting both Bluetooth Classic and Bluetooth Low Energy(BLE) protocols. The ESP32 can be programmed using different programming languages, including C, C++ and MicroPython.
We need a motor for movement of the wheelchair for this project purpose we are using dc geared motors this kind of motors have high torque and low RPM which make them suitable for this application, here we also need a driver circuit which helps us in controlling the

direction and speed of the rotation of motors for this we are using H bridge motor driver which is L298N motor driver which can supply maximum of 2 amps of current and 40v of the voltage at the output on each side.

This driver has four input pins IN1, IN2, IN3, IN4 which are used for the controlling the direction of motors and we have the EN1 and EN2 for controlling the speed or RPM of the motors, here at motor driver we have three power pins which are called 12v, Gnd, 5v pin where from 12v pins we are supplying power to the motors and form 5v pins we are supplying power to the motor driver itself but if input to the 12v pin is less than or equal to 12v then 5v input to the motor driver is not necessary we can use it as a 5v output pin and as our motor needs less than 12v we can use 5v to power our microcontroller board ESP32 via Vin power pin and same 5v pin of the motor driver is used to power the ultrasonic sensor as well.

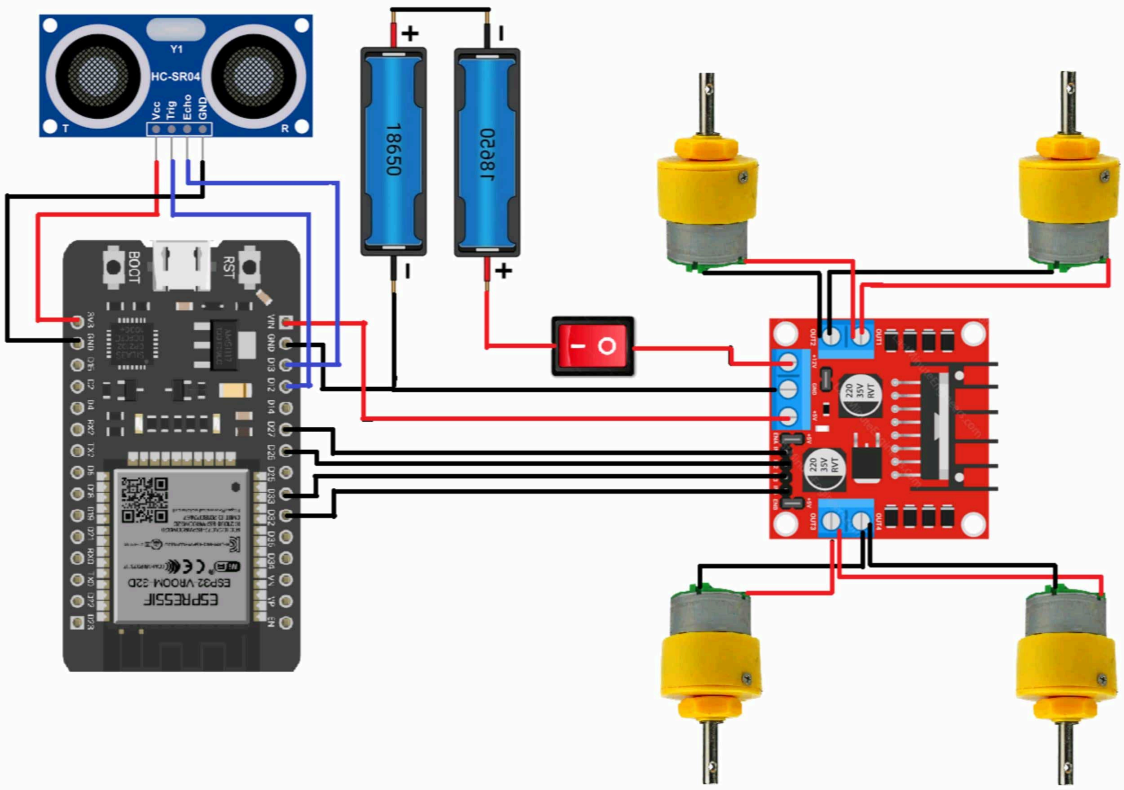


Figure 4: Circuit Diagram Of Wheelchair Movement

In order to move a chair we are using differential steering mechanism where turning of the chair can be accomplished by changing the direction of the rotation of the motors, if we need to move wheelchair forward then we are going to rotate both left side and right side motors in forward direction, if we need the move wheelchair backward then we will move both left side and right side motors to the backward, if we need to move the wheelchair to the right side then we are going to rotate left motors forward and right motors backward and if we need to move left side then we a going to move right motors to the forward and left motors to the backward.

Table 1: Motor Movement

Movement of Motors According to Hand Gesture		
Direction of Hand Gesture	**Movement of Left Motor**	**Movement of Right Motor**
Downward	Forward	Forward
Upward	Backward	Backward
Right	Forward	Backward
Left	Backward	Forward

In order to implement obstacle detection we are using ultrasonic sensor which uses ultrasonic waves for the detection of the obstacle, ultrasonic waves are a type of sound wave that has a frequency greater than the upper limit of human hearing, typically above 20,000 Hz and The approximate speed in air is 330 m/s. we are using HCSR04 ultrasonic sensor, it is an affordable and widely used ultrasonic distance sensor that allows non- contact measurement of distances in various applications. It consists of an ultrasonic transmitter and receiver,operating on the principle of echolocation. The sensor emits ultrasonic waves in the form of short pulses. These pulses travel through the air and bounce off objects in their path. The receiver then detects the reflected waves and by measuring the time taken for the waves to return, it calculates the distance to the object. The sensor can typically measure distances from 2 cm to 400 cm. The sensor typically provides a resolution of around 1 cm, meaning it can measure distances with a granularity of 1 cm. Here its trigPin is a ultrasonic wave transmitting pins and echoPin is ultrasonic wave receiving pin both are connected to the digital pin of the ESP32 and it has Vcc pin where we need to supply between 5v to 12v power supply in our case it is supplied through the 5v pin of the motor driver.

3.1 Hardware Design

In this section we will see the details about hardware which we have used for completion of the project and we will also see how they are useful for the completion of the project.

3.1.1 ESP32 Board

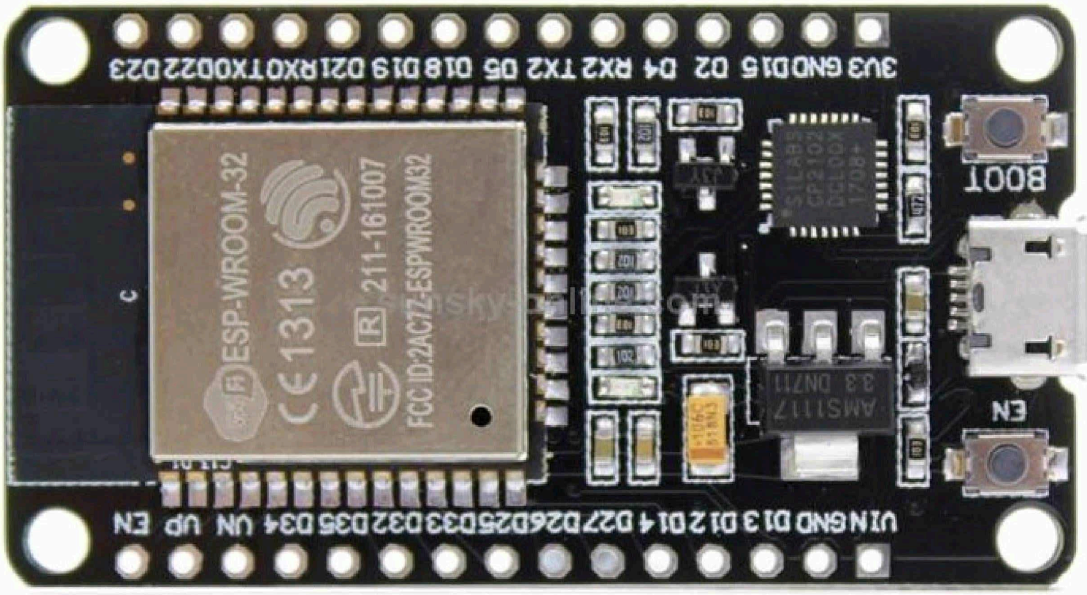


Figure 5: ESP32 WROOM Board

The ESP32 is a powerful microcontroller and Wi-Fi module that combines a high-performance processor, ample memory and built-in wireless connectivity. Developed by Espressif Systems, the ESP32 offers a wide range of features and capabilities,making it a popular choice for Internet of Things (IoT) projects and other embedded applications. In the following description, I will provide an overview of the ESP32's key features,components and functionalities. The ESP32 is based on the Xtensa LX6 dual-core processor, which operates at clock speeds up to 240 MHz. It features a scalable architecture with an adjustable CPU frequency and adjustable power modes to optimize performance and power consumption. The microcontroller includes 520KB of SRAM for data storage and 4MB of Flash memory for program storage. The ESP32 supports multiple communication protocols such as UART, SPI, I2C, I2S and Can enabling seamless integration with various peripherals and sensors.

It offers Wi-Fi connectivity with support for 802.11 b/g/n standards, allowing for wireless communication and Internet connectivity. The ESP32 also includes Bluetooth functionality, supporting both Bluetooth Classic and Bluetooth Low Energy (BLE) protocols. With dual-mode Bluetooth, The ESP32 can act as both a Bluetooth device and a Bluetooth host, enabling a wide range of wireless communication possibilities. The module features integrated Wi-Fi and Bluetooth antenna switches, power amplifier, low-noise receive amplifier, filters and power management modules. It supports Wi-Fi Direct which allows devices to connect to each other without the need for a traditional access point. The ESP32 provides secure Wi-Fi authentication and encryption protocols, including WPA/WPA2, WPA-PSK/WPA2-PSK and WEP. It supports IPv4 and IPv6 network protocols, allowing for compatibility with different network configurations.

The module features a rich set of GPIO (General Purpose Input/Output) pins that can be configured for various purposes. It includes numerous built-in sensors such as temperature, hall effect and capacitive touch sensors, simplifying the integration of sensing capabilities into projects. The ESP32 supports analog-to-digital conversion (ADC) with up to 18 channels, allowing for precise analog measurements. It includes a real-time clock (RTC) and supports timers and watchdogs, enabling accurate timekeeping and task scheduling. The module features a built-in temperature sensor that can be used for monitoring environmental conditions. It offers cryptographic hardware acceleration for secure communication and data encryption. The ESP32 supports over-the-air (OTA) updates, allowing firmware to be updated wirelessly, without physical access to the device. It provides a wide range of development tools and software libraries, including the Espressif IoT Development Framework (ESP-IDF)and the Arduino IDE. The ESP32 is compatible with various operating systems, including FreeRTOS, making it suitable for multi-threaded and real-time applications.

It supports deep sleep modes, consuming minimal power during idle periods and maximizing battery life in battery-powered applications. The module features an integrated power management unit (PMU) that optimizes power consumption and battery life. The ESP32 offers a rich ecosystem with a vibrant community, providing extensive documentation, tutorials and support. It supports different development boards and modules with varying form factors and pin configurations to accommodate diverse project requirements. The ESP32 can be programmed using different programming languages including C, C++ and MicroPython. It supports cloud-based platforms, such as Amazon Web Services (AWS) IoT Core and Google Cloud IoT facilitating seamless.

3.1.2 Accelerometer Sensor

We are using MPU6050 as a accelerometer sensor, The MPU6050 is manufactured by InvenSense and is commonly found in many electronic devices and development boards. The MPU6050 is a popular 6-axis motion tracking sensor module that combines a 3-axis gyroscope and a 3-axis accelerometer on a single chip. This module can be used to detect acceleration, tilt and vibration in various applications such as robotics, gaming and motion sensing.

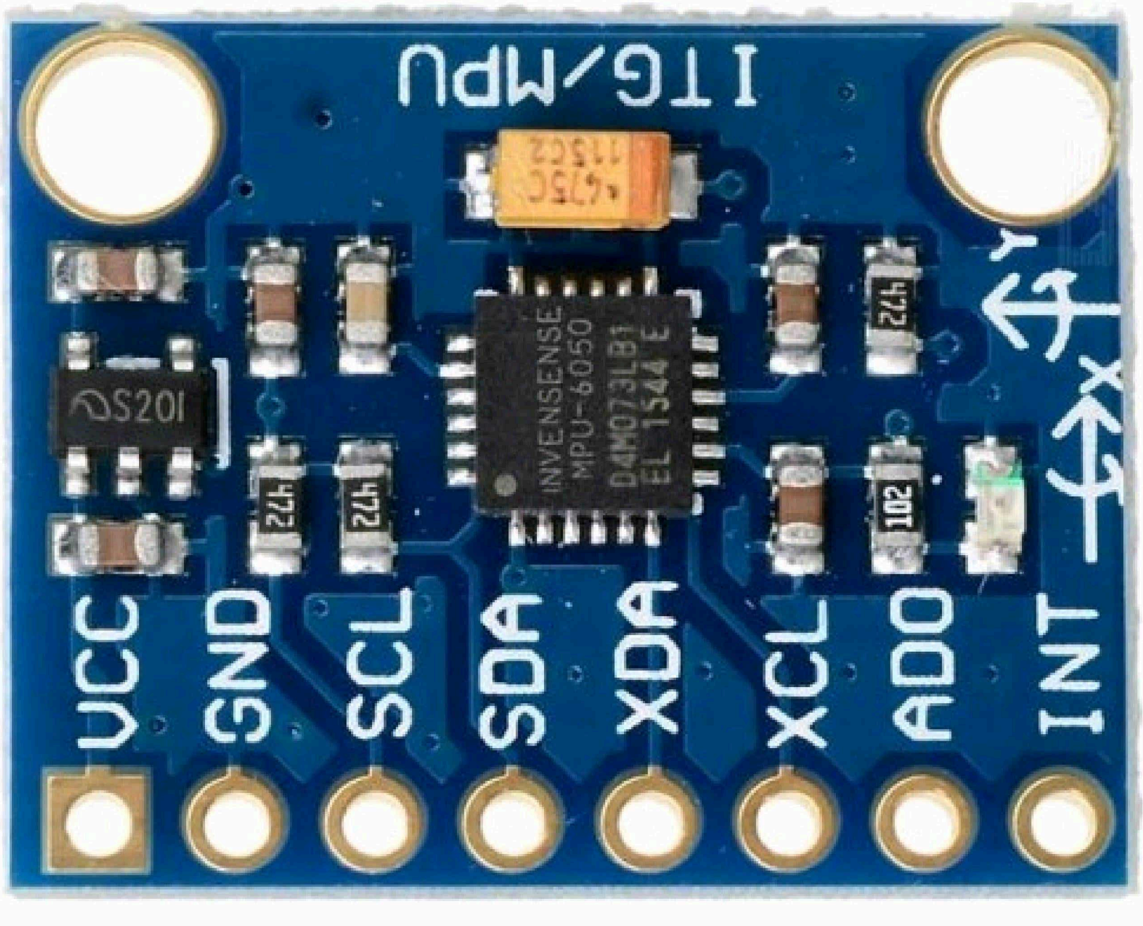


Figure 6: Accelerometer Sensor (MPU6050)

The accelerometer component of the MPU6050 measures acceleration in three dimensions: X, Y and Z. It does this by sensing changes in capacitance caused by the movement of a small mass inside the device. The gyroscope component measures rotational velocity in the same three dimensions using a microelectromechanical system (MEMS) to detect angular changes. It provides a measurement range of ±2g, ±4g, ±8g or ±16g depending on the

sensitivity setting configured by the user. The gyroscope component measures rotational velocity in degrees per second (dps) and offers a programmable range of ±250, ±500, ±1000 or ±2000 dps. Together the accelerometer and gyroscope can be used to provide accurate measurements of both linear and rotational motion. The module also includes on-board digital motion processing (DMP) that performs sensor fusion to combine the data from the accelerometer and gyroscope into a single output stream.
The MPU6050 can communicate with a microcontroller or other digital device via an I2C or SPI interface. The sensor operates at a maximum communication speed of 400 kHz for I2C and 1 MHz for SPI allowing for fast and efficient data transfer. It operates on a wide voltage range (2.375V-3.46V) and has low power consumption, making it suitable for battery-powered applications. The module also includes an on-board temperature sensor that can be used to compensate for changes in sensor output due to temperature variations. Overall, the MPU6050 is a versatile and widely used motion tracking sensor module that can be used in a variety of applications where accurate measurement of motion is required.

3.1.3 Ultrasonic Sensor

Figure 7: Ultrasonic Sensor (HCSR04)

The HCSR04 is an affordable and widely used ultrasonic distance sensor that allows non-contact measurement of distances in various applications. It consists of an ultrasonic transmitter and receiver, operating on the principle of echolocation. The sensor emits ultrasonic waves in the form of short pulses. These pulses travel through the air and bounce off objects in their path. The receiver then detects the reflected waves and by measuring the time taken for the waves to return, it calculates the distance to the object. The sensor can typically measure distances from 2 cm to 400 cm, although this range may vary depending on the specific model.

The HCSR04 provides good accuracy for most applications, with an error range of around ± 3 mm. It operates on the Time-of-Flight (ToF) principle, where the distance is calculated based on the time it takes for the ultrasonic waves to travel to the object and back.The sensor operates on a wide range of voltage levels, usually between 5V and 12V DC. The HCSR04 provides distance measurement output in either digital or analog format, depending on the model and configuration. The sensor requires two pins for operation -A trigger pin to initiate the measurement and an echo pin to receive the reflected waves. The sensor typically provides a resolution of around 1 cm, meaning it can measure distances with a granularity of 1 cm.

The HCSR04 offers a fast response time, allowing for quick and real-time distance measurements. Since it relies on ultrasonic waves. The sensor can measure distances without physical contact with the object making it suitable for various applications. The HCSR04 is commonly used in robotics, automation, security systems, object detection and proximity sensing applications. The sensor is usually mounted in a fixed position and directed towards the target area for accurate distance measurement. The HCSR04 performs well in most indoor environments. However, it may be affected by certain factors such as temperature, humidity and acoustic properties of the surroundings.

The sensor is easy to interface with microcontrollers and single-board computers using GPIO pins allowing for seamless integration into electronic projects. There are readily available libraries and code examples for various platforms that simplify the integration and usage of the HCSR04 sensor. The HCSR04 may encounter challenges when measuring the distance to certain objects, especially those with irregular shapes, soft or absorbent surfaces or extremely small targets.

3.1.4 Motors

We are using 12v 100RPM dc geared motor for our project. The 12V 100RPM DC geared motor is a compact and versatile motor widely used in various applications that require controlled speed and torque. It operates on a 12V DC power supply, making it compatible

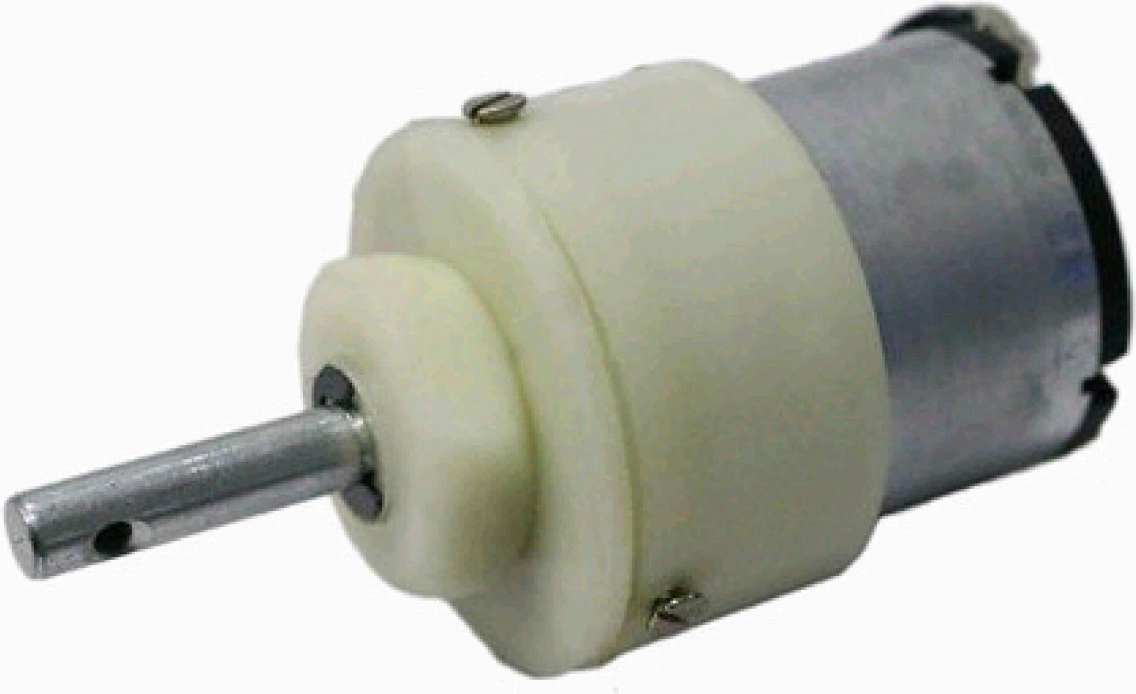


Figure 8: DC Geared Motor

with common power sources such as batteries and power adapters. The motor is designed to provide a rotational output of 100 revolutions per minute (RPM), which indicates the speed at which the motor shaft rotates under no load conditions.
This motor is equipped with a gearbox that reduces the output shaft speed while increasing the torque output. The specific gear ratio of the motor determines the reduction in speed and the increase in torque. In the case of a 100RPM motor, the gear ratio is optimized to provide a suitable balance between speed and torque.
The geared motor provides increased torque compared to a standard DC motor, making it suitable for applications that require more power and the ability to drive loads. The motor's direction of rotation can be easily controlled by changing the polarity of the applied voltage or by using an H-bridge motor driver circuit. The motor is typically designed for easy mounting, with options for mounting brackets or holes to secure it in place. The physical dimensions and weight of the motor vary depending on the specific model and manufacturer, but it is generally compact and lightweight.
The motor is equipped with an output shaft to which various mechanical components such as wheels, gears or pulleys can be attached for transmitting motion. The motor is designed to operate within a certain temperature range and may require additional cooling if used

continuously at high loads. The motor has a rated voltage of 12V, which should be strictly followed to prevent damage. Additionally, it has a current rating specified by the manufacturer, indicating the maximum current it can draw during operation.
The geared motor's efficiency refers to how effectively it converts electrical power into mechanical power. Higher efficiency motors generally consume less power and produce less heat. Some geared motors require periodic lubrication of the gearbox to ensure smooth and efficient operation. It is essential to follow the manufacturer's guidelines for maintenance. The 12V 100RPM DC geared motor finds applications in robotics, automation, automotive systems, electronic locks, conveyor systems, precision instruments and various DIY projects.The motor can be controlled using simple switches or by integrating it with microcontrollers and motor driver circuits for more precise speed and direction control.
It can be easily integrated with other electronic components and sensors to build complex systems or automation projects. The geared motor may produce some level of noise during operation, which can vary based on the gear design and overall quality. To reduce operational noise, it is possible to add vibration-damping materials or incorporate noise reduction techniques into the motor mounting system. The motor's lifespan depends on factors such as operating conditions, load, maintenance and build quality. Proper care and usage can significantly extend its operational life In order to prevent damage to the motor, it is advisable to incorporate overcurrent protection measures such as fuses or current-limiting circuits. The motor may draw a higher current at startup due to the initial surge required to overcome inertia. This should be considered when selecting power sources and motor drivers. To properly drive the geared motor, an appropriate motor driver circuit should be chosen based on the motor's voltage and current requirements.

3.1.5 Motor Driver

The L298N is a popular motor driver integrated circuit (IC) used to control and drive DC motors and stepper motors. It provides a convenient solution for controlling motors in a wide range of applications including robotics, automation and hobby projects. The L298N is capable of driving two DC motors or a single stepper motor simultaneously. The motor driver operates within a wide voltage range, typically between 5V and 35V DC. The L298N can handle a maximum current of up to 2A per channel or 4A in total making it suitable for driving medium-sized motors.
The motor driver utilizes an H-bridge configuration which allows for bidirectional control of the motors enabling forward, reverse and braking operations. The L298N includes built-in

protection features such as thermal shutdown and overcurrent protection to prevent damage to the IC and connected motors. The motor driver supports various control modes including direct input control, PWM (Pulse Width Modulation) control and enable/disable control. The L298N is compatible with both 5V and 3.3V logic systems making it easy to interface with microcontroller and other digital circuits.

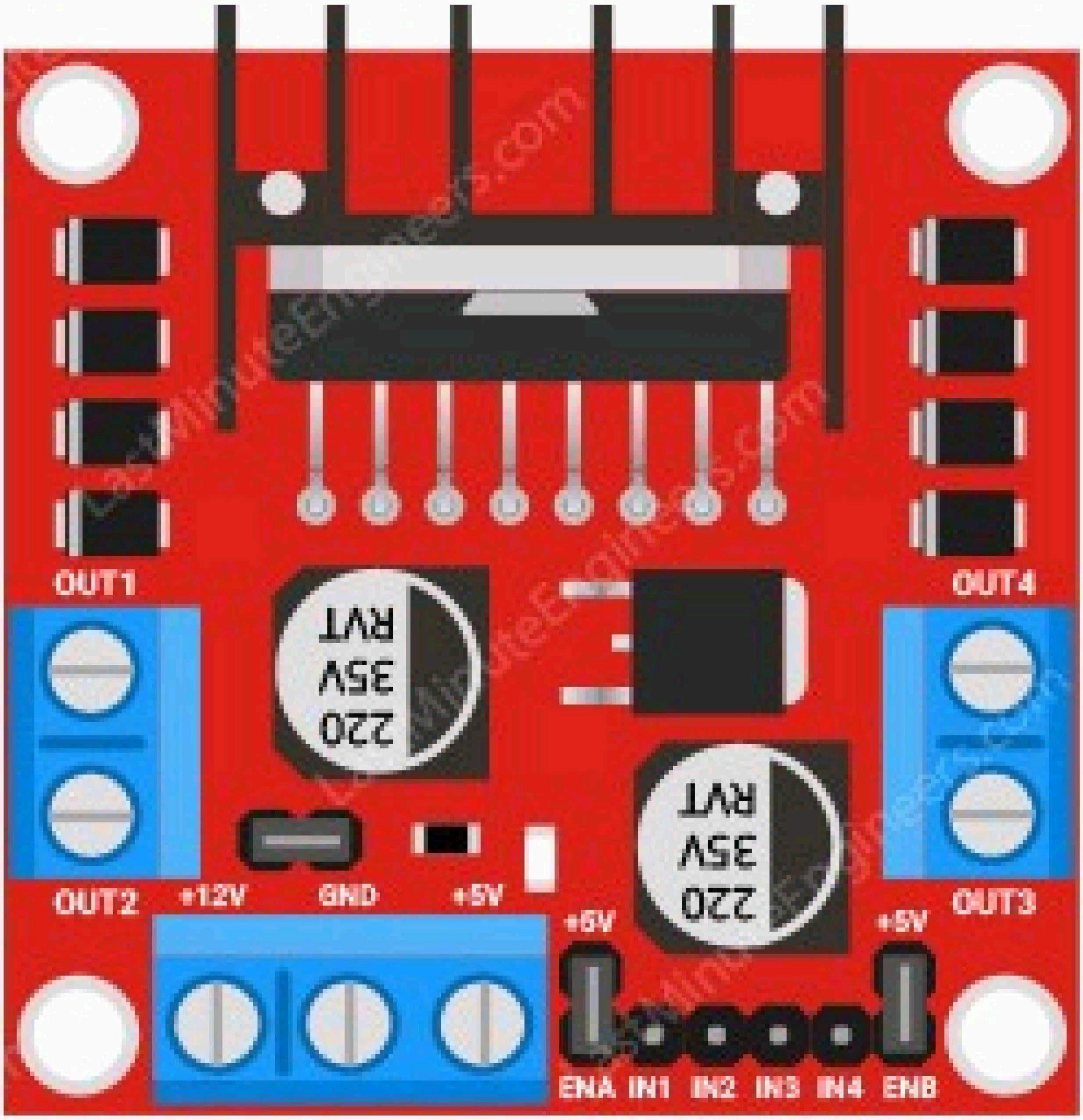


Figure 9: Motor Driver Module (L298N)

The L298N IC can generate heat during operation, So it is recommended to use heat sinks or cooling fans for efficient heat dissipation. The L298N has multiple pins for motor connections, power supply, control inputs and enable/disable functions making it easy to connect and control motors. The motor driver allows for speed control by varying the PWM signal or using analog voltage inputs for smooth speed modulation. The L298N supports both bipolar and unipolar stepper motors and provides step and direction control signals for precise positioning. The L298N motor driver can be used with a wide range of motors including brushed DC motors, gear motors and various types of stepper motors.

The IC incorporates built-in diodes (freewheeling diodes) to suppress voltage spikes that occur during motor operation and protect the circuit from reverse current flow. The L298N includes protection against overvoltage conditions, safeguarding the connected components. The L298N motor driver comes in a compact and easily mountable package, facilitating integration into small-scale projects. The L298N motor driver is widely used in applications such as robot platforms, CNC machines, 3D printers, camera sliders and motorized vehicles. The motor driver accepts control signals from microcontrollers, Arduino boards, Raspberry Pi, ESP and other digital devices allowing for flexible and precise motor control. The L298N motor driver offers a reliable and efficient solution for controlling and driving DC motors and stepper motors. Its versatility, protection features and ease of use make it a popular choice for motor control in a wide range of projects.

3.1.6 ESP-NOW Protocol

ESP-NOW stands for "ESP-Wireless Over Network." It is a lightweight communication protocol designed for IoT devices. It offers low-latency, secure and power-efficient data exchange, making it suitable for various IoT applications. ESP-NOW is a connection-less. Communication protocol developed by Espressif Systems for low-power, peer-to-peer communication between ESP8266 and ESP32 devices without the need for a WiFi network or Internet connection.

ESP-NOW operates in the 2.4 GHz frequency band similar to Wi-Fi, but with a different modulation scheme and lower power consumption. It uses the MAC layer of Wi-Fi to exchange data between devices but, it does not require an IP stack making it more efficient for simple device-to-device communication. The protocol has a small overhead making it suitable for resource-constrained devices with limited memory and processing power. ESP-NOW provides a simple way to discover nearby devices. It includes a mechanism for scanning and finding available devices in the vicinity. ESP-NOW can coexist with Wi-Fi on the same device. Enabling devices to communicate using both protocols simultaneously.

It offers power-saving features allowing devices to enter sleep mode when not actively transmitting or receiving data thus, conserving battery life.ESP-NOW can be used in scenarios where Wi-Fi connectivity is unreliable or not available, providing a robust alternative for local device communication. It uses a lightweight message format, strong security features. ESP-NOW supports Encrypted and unencrypted unicast communication, it can transfer data upto 250 bytes at a time. This protocol has a limitation it supports maximum of 20 encrypted and unencrypted peers.

The protocol uses a 6-byte MAC address to identify and address devices. Each ESP8266 or ESP32 module has a unique MAC address, which is used for device identification during communication.ESP-NOW provides a reliable delivery mechanism for data packets. When a packet is transmitted, the receiving device sends an acknowledgment (ACK) back to the sender to confirm successful reception. In case an ACK is not received within a specified time, the sender retransmits the packet to ensure its successful delivery. This mechanism helps maintain data integrity and reliability.

The protocol offers a simple pairing process for devices to establish a secure connection. Devices exchange a common key during the initial setup ensuring that only authorized devices can communicate. ESP-NOW can be used alongside other wireless protocols. Such as Wi-Fi and Bluetooth to extend the capabilities of IoT devices and enable different modes of communication. It provides a reliable means of sending commands and receiving responses, making it useful for remote device management and control scenarios. The protocol has a simple and intuitive API, making it easy to implement and integrate into existing projects. Libraries and examples are available for popular development platforms simplifying the development process. The ESP-NOW protocol has gained popularity in the maker and IoT communities due to its simplicity, efficiency and ability to establish direct communication between ESP8266 and ESP32 devices.

3.1.7 Lithium-Ion Battery

3.7v 18650 lithium-ion batteries are rechargeable batteries commonly used in a variety of electronic devices including laptops, flashlights and power tools. They are named after their dimensions with a length of 65 mm and a diameter of 18 mm. These batteries are popular due to their high energy density, which means they can store more energy than other types of batteries of similar size. They also have a longer lifespan and can withstand more charging and discharging cycles than other rechargeable batteries. The 18650 battery has a wide temperature range allowing it to operate in extreme environments. The battery can be used in series or parallel configurations to increase voltage or capacity respectively. The battery's capacity is measured in milliampere-hours (mAh) and indicates how much

Figure 10: Lithium-Ion Battery

charge it can store. Typical capacities range from around 2000mAh to 3500mAh. 18650 batteries use a lithium-ion chemistry that relies on the movement of lithium ions between an anode and a cathode to generate electrical current. This chemistry makes them more environmentally friendly than other types of batteries, as they do not contain heavy metals such as lead or cadmium.

The 18650 battery uses lithium-ion technology, which allows for the reversible movement of lithium ions between positive and negative electrodes during charging and discharging.It consists of several key components including the positive electrode (cathode), negative electrode (anode), separator, electrolyte and current collectors. The positive electrode is typically made of a lithium compound such as lithium cobalt oxide (LiCoO2), lithium manganese oxide (LiMn2O4) or lithium nickel manganese cobalt oxide (LiNiMnCoO2). The negative electrode is usually made of graphite, which can efficiently intercalate and deintercalate lithium ions during charge and discharge cycles.

The separator is a porous membrane that prevents direct contact between the positive and negative electrodes while allowing the passage of lithium ions. The electrolyte is a conductive solution containing lithium salts. Such as lithium hexafluorophosphate (LiPF6), which facilitates the movement of lithium ions within the battery. The current collectors are metal foils (usually made of copper for the positive electrode and aluminum for the negative electrode) that collect and distribute electrical current within the battery.

The 18650 battery operates based on the principle of redox reactions, where lithium ions are transferred between the electrodes through the electrolyte during charging and discharging. During charging, Lithium ions are extracted from the positive electrode and inserted into the negative electrode, while electrons flow through an external circuit creating a charging current. During discharging, the process is reversed.Lithium ions move from the negative electrode to the positive electrode providing a flow of electrons through the external circuit generating a discharge current.

The number of charge cycles the battery can withstand depends on factors such as temperature, charging rate and depth of discharge. To recharge an 18650 battery, it is connected to a suitable charger that provides the correct voltage and current for the battery's specifications. The charging process typically involves several stages,including bulk charging, absorption charging and trickle charging, to ensure optimal charging and prolong battery life. Safety features such as overcharge protection, over-discharge protection and short circuit protection are often incorporated into 18650 batteries or the devices that use them to prevent damage or accidents.

The 18650 battery's self-discharge rate is relatively low compared to other battery types. Meaning it retains its charge for longer periods when not in use. However, the battery should still be periodically recharged to maintain its optimal performance and prevent deep discharge which could lead to capacity loss.

The battery's energy density makes it suitable for use in portable devices, where space and weight are important factors. The 18650 battery has a relatively low internal resistance, which allows it to deliver high current output without significant voltage drop. The battery's low internal resistance also helps to minimize energy loss and maximize efficiency during charging and discharging. lithium-ion batteries are exceptional power storage solutions that combine high energy density, long cycle life, fast charging capabilities and a commitment to safety and environmental sustainability.

3.1.8 Charging Module

The TP4056 charging module is a popular and widely used charging board designed for charging single-cell lithium-ion or lithium-polymer batteries. The module utilizes the TP4056 chip, which is a dedicated linear charging controller IC. The TP4056 charging module is compact and easy to use, making it suitable for a wide range of applications including DIY projects, portable devices and power banks. The module supports 4.2V constant voltage charging, which is the standard charging voltage for most lithium-ion and lithium-polymer batteries.

The TP4056 charging module provides a simple and straightforward charging solution, requiring minimal external components for operation. The module features onboard protection

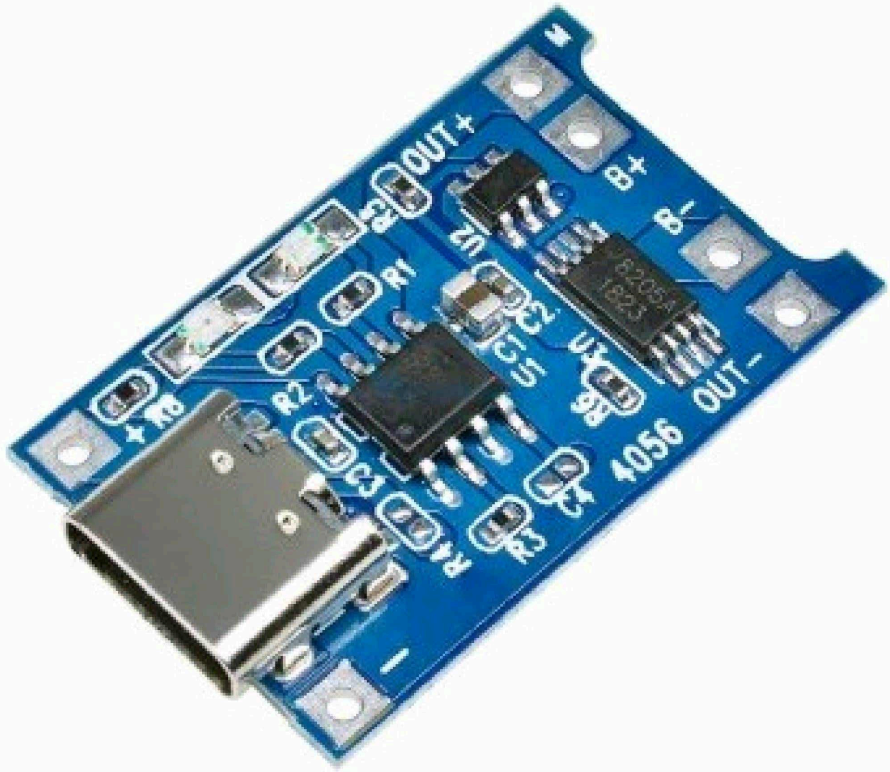


Figure 11: Charging Module (TP4056)

circuits to prevent overcharging, over-discharging and overcurrent situations, ensuring safe and reliable charging. The charging current can be set by adding a resistor to the module, allowing flexibility in charging different battery capacities. The TP4056 charging module typically provides a charging current range of 100mA to 1000mA depending on the resistor value used.

The module offers a built-in voltage regulation function that automatically terminates the charging process, when the battery reaches the preset voltage. The TP4056 charging module includes a charging status indicator, usually in the form of an LED to indicate the charging state of the battery. The module is designed to work with a 5V input power source, commonly provided by a USB connection or a 5V power supply.

The TP4056 charging module features a micro-USB or mini-USB connector for easy and convenient power input. The module is polarity protected, preventing damage to the circuit if the battery is connected with reversed polarity. The TP4056 charging module supports charging while the load is connected to the battery, enabling simultaneous charging and discharging. The module operates in a linear charging mode, which means it regulates the charging voltage and adjusts the charging current accordingly.

The TP4056 charging module has a low dropout voltage, ensuring efficient charging even when the input voltage is slightly higher than the battery voltage. The module incorporates thermal shutdown protection, which activates if the internal temperature exceeds a safe threshold preventing overheating and potential damage. The TP4056 charging module has a wide operating temperature range typically spanning from -40°C to 85°C, allowing for reliable charging in various environments. He module supports trickle charging, which provides a low charging current to top up the battery capacity and maintain its optimal performance.

The TP4056 charging module can be used with various types of lithium-ion and lithium-polymer batteries including popular sizes like 18650, 14500 and 10440. The module's compact size and lightweight design make it suitable for integration into small and portable devices, where space is limited. The TP4056 charging module is compatible with popular development platforms, such as Arduino and Raspberry Pi, allowing for easy integration into projects. The module can be used in conjunction with a battery protection circuit module to provide additional safety features, such as overcurrent and over-discharge protection.

The TP4056 charging module typically has several pins that serve specific functions. BAT+ (Battery Positive): This pin is the positive terminal of the battery being charged. It is connected to the positive terminal of the battery. BAT- (Battery Negative)This pin is the negative terminal of the battery being charged. It is connected to the negative terminal of the battery. IN+ (Input Positive): This pin is the positive input terminal for the charging module. It is connected to the positive terminal of the power source (usually 5V input). IN- (Input Negative): This pin is the negative input terminal for the charging module. It is connected to the negative terminal of the power source (usually ground). OUT+ (Output Positive): This pin is the positive output terminal of the charging module. It is connected to the positive terminal of the load (device being powered or charged). The TP4056 charging module can be customized or modified by adding additional components, such as capacitors or resistors, to adapt to specific charging requirements. The module is commonly used in power bank designs enabling users to charge their portable devices on the go. The TP4056 charging module offers a cost-effective charging solution for DIY enthusiasts, makers and hobbyists. The module's simplicity and ease of use make it suitable for beginners in electronics and battery charging applications, It's important to consult the datasheet or product documentation specific to your TP4056 module to ensure accurate pin assignments and functions, as there may be slight variations between different versions or manufacturers.

3.1.9 Buzzer

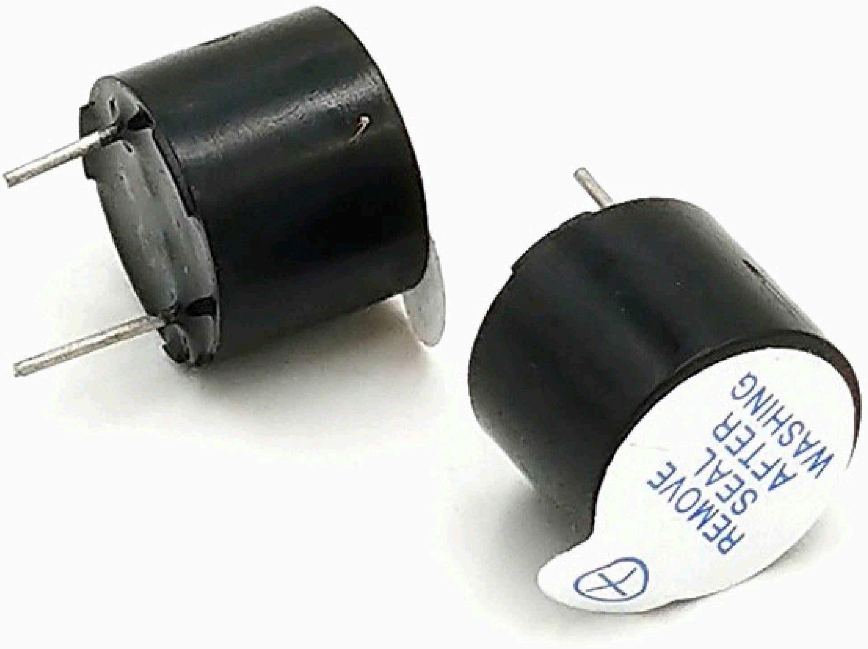


Figure 12: Buzzer

A buzzer is an electronic component that produces sound when an electrical signal is applied to it. Buzzer is an output device that is used in various electronic applications and devices to generate sound alerts, alarms or musical tones. Buzzer is a type of transducer that converts electrical energy into sound energy. The most common type of buzzer is a piezoelectric buzzer, which generates sound by vibrating a piezoelectric crystal or ceramic material. Piezoelectric buzzers are small, lightweight and energy-efficient, making them ideal for use in portable and battery-operated devices.
Buzzer is a simple and cost-effective way to add sound output to a project or device making it a popular component in electronic hobbyist projects. Buzzer is available in various sizes and shapes, from small surface-mount devices to large high-power units. The sound output of a buzzer can range from a simple beep to complex musical tones, depending on the type of buzzer and the control circuitry used. Buzzers can be driven by a wide range of electrical signals, including direct current (DC), alternating current (AC) and pulse width modulated (PWM) signals. Some buzzers have integrated control circuitry while others require an external driver circuit to generate the appropriate signals. Buzzer typically has two terminals, with one terminal connected to the positive voltage supply and the other connected to the

control circuitry or output device.

The control circuitry or output device applies an electrical signal to the buzzer, causing it to vibrate and generate sound. The frequency of the sound output is determined by the physical characteristics of the piezoelectric material and the electrical signal applied to the buzzer. Buzzer can be used in a wide range of applications, including alarms, doorbells, game consoles, musical instruments and electronic toys. Some buzzers have additional features, such as adjustable volume, multiple tones and LED indicators. Buzzer is commonly used in combination with other components, such as microcontrollers, sensors and switches, to create complex electronic systems. Piezoelectric buzzers have a number of advantages over other types of sound output devices, including low power consumption, high efficiency and fast response times.

Some buzzers are weather-resistant and can be used in outdoor applications, such as automotive and marine environments. Some buzzers have built-in protection circuitry to prevent damage from overvoltage, overcurrent and other electrical faults. Buzzer is a reliable to add sound output to a wide range of electronic devices and systems.

3.1.10 SPST Switch

Figure 13: SPST Switch

A Single Pole Single Throw (SPST) switch is a basic type of switch that consists of a single set of contacts, allowing for the control of a single circuit. It is commonly used in electronic circuits and electrical systems to turn devices on or off. An SPST switch is the simplest form of a switch, comprising a single input terminal, a single output terminal and a lever or button that can be toggled between two positions. It is referred to as single pole because it has only one input or connection point for the circuit and single throw because it can only connect or disconnect that single input. When the switch is in the "on" or closed position, the contacts are brought into contact, allowing current to flow through the circuit. Conversely, when the switch is in the "off" or open position, the contacts are separated, interrupting the flow of current. The lever or button of the switch is typically made of an insulating material, such as plastic to prevent accidental contact with the conducting contacts. SPST switches are often used as power switches for appliances, lamps and electronic devices, allowing users to turn them on or off.

3.2 Software Design

We have used arduino IDE for the programming of the ESP32 where we have used the c++ programming language for the programming of the microcontroller below flowchart shows how a algorithm is going to work at gesture recognition circuit and at the wheelchair moving circuit. Here for transmission of signal from from gesture recognition circuit to wheelchair moving circuit, we have used ESP-NOW protocol which is specially designed for the ESP32 boards.

We need to follow certain steps to ensure the proper programming of the ESP32 with arduino IDE. To install the ESP32 board in your Arduino IDE, follow these instructions:

1. In your Arduino IDE, go to File then goto Preferences.

2. Enter the following into the "Additional Board Manager URLs" field, `'https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32 _index.json'`

3. Open the Boards Manager. Go to Tools then goto Board then goto Boards Manager.

4. Search for ESP32 and press install button for the "ESP32 by Espressif Systems".

following these steps will ensure installing of ESP32 board manager and that our arduino IDE is compatible with ESP32.

Now our arduino ide is ready to program ESP32 so now we need to follow certain steps to ensure the proper communication between gesture recognition circuit and the wheelchair moving circuit. At the ESP32 of gesture recognition circuit we need to follow below steps to ensure sending of the signals:

1. First of all we need to Initialize ESP-NOW in sender ESP32 for that we have a function `'esp_now_init()'` which is used for initialising ESP32. this function is present inside the ESP-NOW library which is installed as we install the ESP32 board.

2. Register a callback function upon sending data in sender ESP32 this function will be executed when a message is sent. This can tell us if the message was successfully delivered or not. for registering callback function we use `'esp_now_register_send_cb()'` function. this function is present inside the ESP-NOW library which is installed as we install the ESP32 board.

3. Add a peer device (the receiver). For this, you need to know the receiver MAC address. we use `'esp_now_add_peer()'` function to pair a device and pass as an argument the peer MAC address to it. this function is present inside the ESP-NOW library which is installed as we install the ESP32 board.

4. Send a message to the peer device. to send a message after establishing connection with the receiver we use '`esp_now_send()`' function, this function is present inside the ESP-NOW library which is installed as we install the ESP32 board.

At the ESP32 of wheelchair moving circuit we need to follow below steps to ensure receiving of the signals:

1. First of all we need to Initialize ESP-NOW in sender ESP32 for that we have a function '`esp_now_init()`' which is used for initialising ESP32. this function is present inside the ESP-NOW library which is installed as we install the ESP32 board.

2. Register a callback function upon sending data in sender ESP32 this function will be executed when a message is sent. This can tell us if the message was successfully delivered or not. for registering callback function we use '`esp_now_register_rcv_cb()`' function. this function is present inside the ESP-NOW library which is already installed as we install the ESP32 board.

3. Inside that callback function, save the message into a variable to execute any task with that information.

We are using MPU6050 for detection change in accelertaion due to gravity and to program this in ESP32 with using arduino IDE, we'll use the '`Adafruit_MPU6050`' library. To use this library you also need to install the '`Adafruit_Unifiedsensor`' library and the '`Adafruit_BusIO`' Library.You can create a variable of type '`Adafruit_MPU6050`' and and then use '`getEvent()`' function to get sensor readings, this sensor can give us readings of acceleration due to gravity, gyrometer and also can give us readings of temperature as well but, we will use accelerartion due to gravity only and we are going to use this inside the transmitter ESP32.

We are using HCSR04 for the obstacle detection at the receiver section and we are going to program this inside the ESP32 at the receiver section, for that we have two pins one trigpin and other is echo pin which plays vital role. First of all we need to give high signal to the trigPin for the 10 microsecond then sensor will emit the ultrasonic wave which then will be detected by the echo pin and it will be turned high,then we have the function called '`pulseIn()`' which will give us duration it took to get back from obstacle then by multiplying it with speed of sound we will get the distance from the sensor and the obstacle and based on this reading we will decide the the obstacle is detected or not.

At the gesture recognition circuit we have accelerometer sensor it will give us reading of X, Y, Z axis which help us in deciding whether sensor is moved up, down, left, right. We have

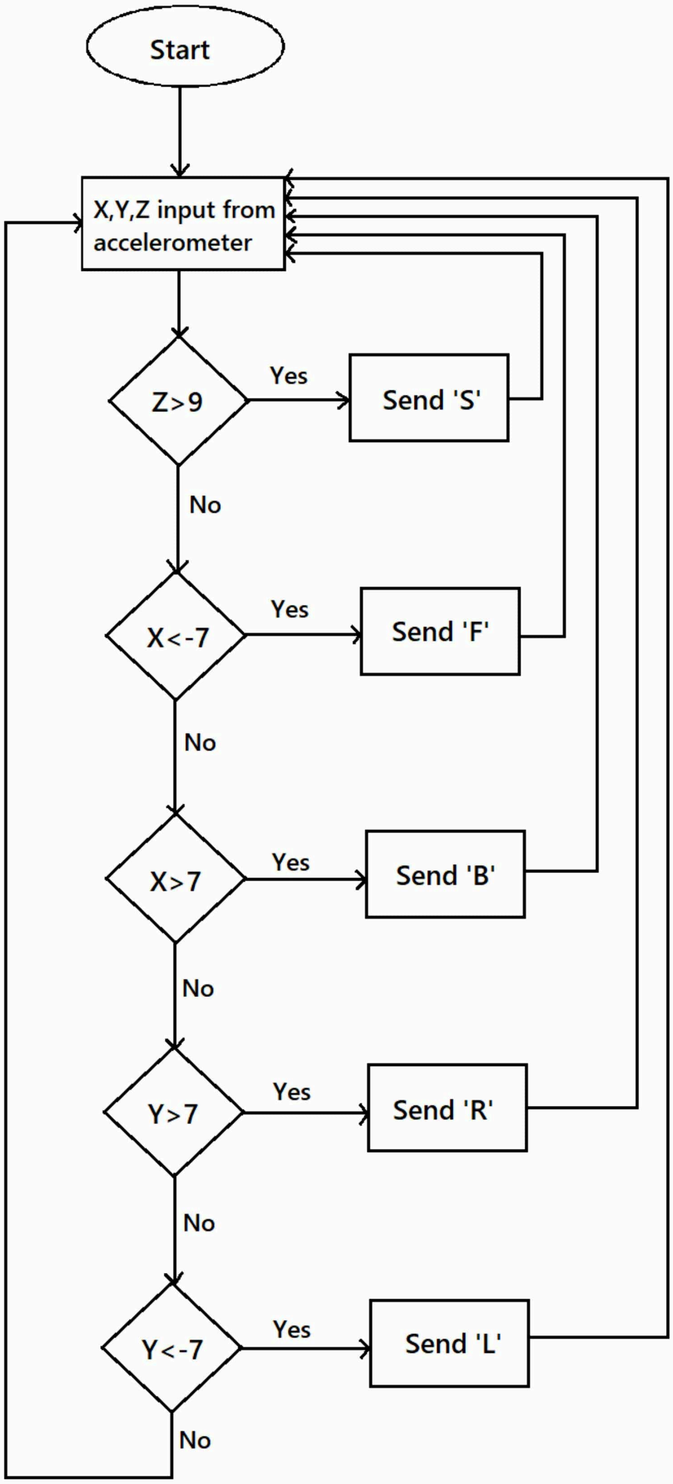


Figure 14: Flowchart of Working at Gesture Recognition Circuit

done the trial and error process and found some readings which we then used to program the ESP boards. If the z axis readings are greater than '9' then the sensor which we are mounting on the hand of the operator have not been moved yet or the operator wants to stop then the transmitter ESP32 will send the 'S' signal to the Receiver ESP32 and will look for next inputs, if the sensor readings are not greater than '9' then it will check next condition that is if the x axis readings are less than '-7' then sensor which is mounted on the hand of the operator is moved downward so now transmitter ESP32 will send the 'F' signal to the receiver ESP32 and will look for next inputs, after that we will check for the next condition which is if the x axis readings are greater than '7' then transmitter ESP32 will send the letter 'B' to the receiver ESP32 because it means that sensor which is on the hand of the operator has been moved upward and then will look for next inputs. if this condition also not satisfied then we will look for next condition, if the y axis readings are greater than '7' then it means sensor was moved towards right side and now transmitter ESP32 will send letter 'R' to the receiver ESP32 and will look for next inputs. if not then it will check the next condition, if the value of reading at the y axis is less than '-7' then it means sensor was moved towards the left and transmitter ESP32 now send the letter 'L' to the receiver ESP32 and then will look for next inputs. above flowchart shows the exact working of the code present at the gesture recognition circuit.

At the at wheelchair moving circuit, it will wait for the signals coming from the gesture recognition circuit and according the signals received it will perform its operation as soon as signal is received the callback function at receiver will execute and the letter, which is transmitted by the transmitter will be stored inside the variable, now if the signal received was letter 'F' then both motors of will start moving in forward direction and then again program will look for next inputs, if the signal received was letter 'B' then it will wait for the data which is coming form the ultrasonic sensor. If the distance from the sensor and the obstacle is greater than 30cm then both the motors of left and the right side will start the moving in opposite that is backward direction and then again program will look for next inputs but if the distance between the sensor and the obstacle is less than the 30cm then both the motors will stop moving the buzzer will start beeping as a symbol that the obstacle is detected and then again program will look for next inputs, if the signal received which is letter 'R' then left side motors will move in forward direction and the right side motor will move in the backward direction this will ensure that the chair is turned to the right side and then again program will look for next inputs.

If the signal received is letter ‘L’ then it means, we need to move the wheelchair to the

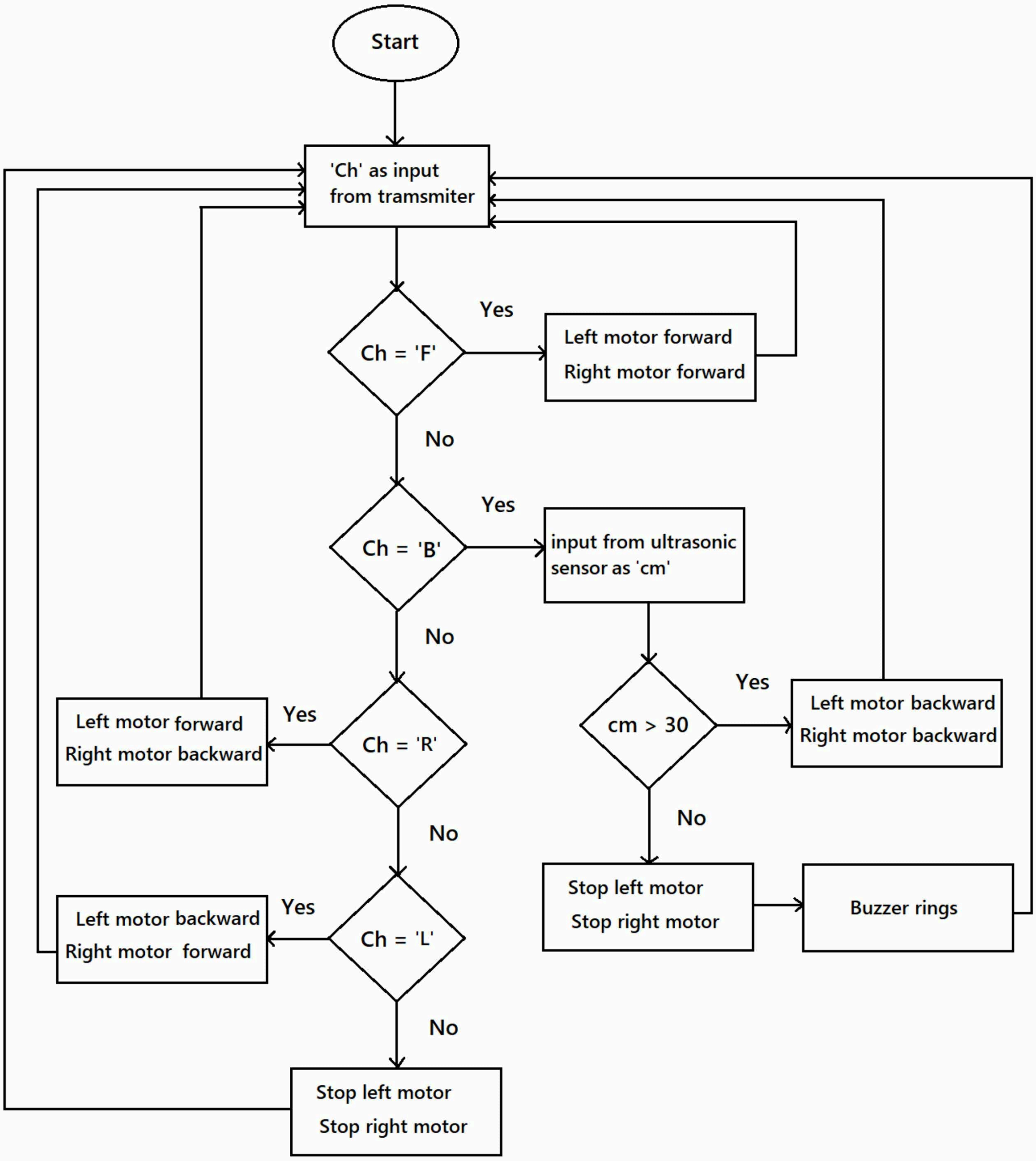


Figure 15: Flowchart of Working at Wheelchair Moving Circuit

left for that the right motor will move to the forward and the left motor will move to the backward direction this will make the chair to move in left direction and then again program will look for next inputs. If the signal received was the ‘S’ then it means the we need to stop the wheelchair chair and after that both motors will stop which make the wheelchair to stop and then again program will look for next inputs. above flowchart shows the working of the code present at the wheelchair moving circuit.

4 CONCLUSION AND RESULTS

4.1 Conclusion

The hand gesture controlled wheelchair with obstacle detection is an innovative and practical solution for individuals with mobility impairments. The system provides a simple and intuitive way to control the movement of the wheelchair using hand gestures, while also detecting and avoiding obstacles in the environment. The main purpose of this paper is to demonstrate a a system which can show case the idea of the hand gesture controlled wheelchair with obstacle detection and this research paper successfully demonstrate the way of implementing a solution for above mentioned problem and the results show that the system is reliable, accurate and it is believed that the proposed system can reduce the complexities of other automated wheelchairs, further by improving the motors and motor driver we can make this capable of carrying a person and commercial production of the presented wheel chair could be a good replacement of imported one and could be a great help to the disabled patients in our country.

4.2 Results

Figure 16: Prototype Image-1

Figure 17: Prototype Image-2

References

[1] Krunal Bansod, kushal Asarkar, Mandar topre, vikrant raj "Hand Gesture Controlled Wheelchair" International Research Journal of Engineeirng and Technology Volume-07, Issue-07, July 2020.

[2] Mufrath Mahmood, Md. Fahim Rizwan, Masuma Sultana, Md. Habib, Mohammad Hasan Imam. "Design of a low cost Hand Gesture Controlled Automated Wheelchair" IEEE Region 10 Symposium, June 2020.

[3] K.S Vairavel, R.Nevetha, "Hand Gesture Wheelchair Control Using Raspberry pi". International Journal of Scientific & Technology Research Volume-09, Issue-04, April 2020.

[4] Shayban Nasif, Muhammed Abdul Goffar khan "Wireless Head Gesture Controlled Wheelchair for Disabled Person". IEEE humanitarian technology conference, December 2017.

[5] Pushpendra jha, "Hand Gesture Controlled Wheelchair". International Journal of Scientific and Technology Research Volume-05, Issue-04, April 2016.

[6] Revati A. Solanke1, Arti R. Salunke, "Voice and Gesture Based Wheelchair for Physically Challenged Using AVR and Android". International Research Journal of Engineering and Technology, Volume-03, Issue-04, April 2016.

[7] Prof. Vishal V. Pande, Nikita S. Ubale, Darshana P. Masurkar, Nikita R. Ingole, Pragati P. Mane. "Hand Gesture Based Wheelchair Movement Control for Disabled person Using MEMS". International Journal of Engineering Research and application Volume-04, issue-04. April 2014.

Appendix

Table 2: Bill Of Material

Bill Of Material				
Sr.no.	Component Name	Unit	Price per unit	Cost
01	ESP32 WROOM Board	2	Rs 600	Rs 1200
02	HCSR04 Sensor	1	Rs 200	Rs 200
03	MPU6050 Sensor	1	Rs 200	Rs 200
04	DC Geared Motor	4	Rs 180	Rs 720
05	L298N Motor Driver	1	Rs 300	Rs 300
06	18650 Batteries	4	Rs 90	Rs 360
07	TP4056 charging module	1	Rs45	Rs 45
08	Jumper Wires	20	Rs 3	Rs 60
09	Single Battery Holder	1	Rs 20	Rs 20
10	Dual Battery Holder	1	Rs 40	Rs 40
11	Wheels	4	Rs 40	Rs 160
12	Caster wheels	2	Rs 75	Rs 150
13	chair metal structure	1	Rs 600	Rs 600
14	Plywood	2	Rs 150	Rs 300
15	Small breadboard	1	Rs 65	Rs 65
16	Large breadboard	1	Rs 90	Rs 90
17	Buzzer	1	Rs 15	Rs 15
18	Others	-	Rs 400	Rs 575
Total Cost				Rs 5100

ESP32-WROOM-32 (ESP-WROOM-32)

Datasheet

Version 2.4

Espressif Systems

About This Guide

This document provides the specifications for the ESP32-WROOM-32(ESP-WROOM-32) module.

Revision History

For revision history of this document, please refer to the last page.

Documentation Change Notification

Espressif provides email notifications to keep customers updated on changes to technical documentation. Please subscribe here.

Certification

Download certificates for Espressif products from here.

Disclaimer and Copyright Notice

Contents

1. Overview

ESP32-WROOM-32 (ESP-WROOM-32) is a powerful, generic Wi-Fi+BT+BLE MCU module that targets a wide variety of applications, ranging from low-power sensor networks to the most demanding tasks, such as voice encoding, music streaming and MP3 decoding.

At the core of this module is the ESP32-D0WDQ6 chip*. The chip embedded is designed to be scalable and adaptive. There are two CPU cores that can be individually controlled, and the clock frequency is adjustable from 80 MHz to 240 MHz. The user may also power off the CPU and make use of the low-power co-processor to constantly monitor the peripherals for changes or crossing of thresholds. ESP32 integrates a rich set of peripherals, ranging from capacitive touch sensors, Hall sensors, SD card interface, Ethernet, high-speed SPI, UART, I2S and I2C.

Note:
* For details on the part number of the ESP32 series, please refer to the document ESP32 Datasheet.

The integration of Bluetooth, Bluetooth LE and Wi-Fi ensures that a wide range of applications can be targeted, and that the module is future proof: using Wi-Fi allows a large physical range and direct connection to the internet through a Wi-Fi router, while using Bluetooth allows the user to conveniently connect to the phone or broadcast low energy beacons for its detection. The sleep current of the ESP32 chip is less than 5 μA, making it suitable for battery powered and wearable electronics applications. ESP32 supports a data rate of up to 150 Mbps, and 20.5 dBm output power at the antenna to ensure the widest physical range. As such the chip does offer industry-leading specifications and the best performance for electronic integration, range, power consumption, and connectivity.

The operating system chosen for ESP32 is freeRTOS with LwIP; TLS 1.2 with hardware acceleration is built in as well. Secure (encrypted) over the air (OTA) upgrade is also supported, so that developers can continually upgrade their products even after their release.

Table 1 provides the specifications of ESP32-WROOM-32 (ESP-WROOM-32).

Table 1: ESP32-WROOM-32 (ESP-WROOM-32) Specifications

Categories	Items	Specifications
Certification	RF certification	FCC/CE/IC/TELEC/KCC/SRRC/NCC
	Wi-Fi certification	Wi-Fi Alliance
	Bluetooth certification	BQB
	Green certification	RoHS/REACH
Wi-Fi	Protocols	802.11 b/g/n (802.11n up to 150 Mbps)
		A-MPDU and A-MSDU aggregation and 0.4 μs guard interval support
	Frequency range	2.4 GHz ~ 2.5 GHz
Bluetooth	Protocols	Bluetooth v4.2 BR/EDR and BLE specification
	Radio	NZIF receiver with -97 dBm sensitivity
		Class-1, class-2 and class-3 transmitter
		AFH
	Audio	CVSD and SBC

Categories	Items	Specifications
Hardware	Module interface	SD card, UART, SPI, SDIO, I2C, LED PWM, Motor PWM, I2S, IR
		GPIO, capacitive touch sensor, ADC, DAC
	On-chip sensor	Hall sensor, temperature sensor
	On-board clock	40 MHz crystal
	Operating voltage/Power supply	2.7 ~ 3.6V
	Operating current	Average: 80 mA
	Minimum current delivered by power supply	500 mA
	Operating temperature range	-40°C ~ +85°C
	Ambient temperature range	Normal temperature
	Package size	18±0.2 mm x 25.5±0.2 mm x 3.1±0.15 mm
Software	Wi-Fi mode	Station/SoftAP/SoftAP+Station/P2P
	Wi-Fi Security	WPA/WPA2/WPA2-Enterprise/WPS
	Encryption	AES/RSA/ECC/SHA
	Firmware upgrade	UART Download / OTA (download and write firmware via network or host)
	Software development	Supports Cloud Server Development / SDK for custom firmware development
	Network protocols	IPv4, IPv6, SSL, TCP/UDP/HTTP/FTP/MQTT
	User configuration	AT instruction set, cloud server, Android/iOS app

2. Pin Definitions

2.1 Pin Layout

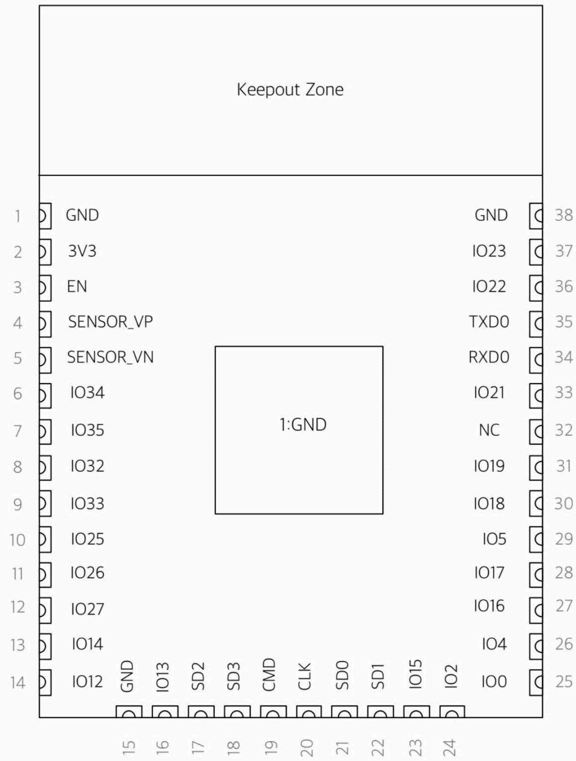


Figure 1: ESP32-WROOM-32 (ESP-WROOM-32) Pin layout

2.2 Pin Description

ESP32-WROOM-32 (ESP-WROOM-32) has 38 pins. See pin definitions in Table 2.

Table 2: Pin Definitions

Name	No.	Type	Function
GND	1	P	Ground
3V3	2	P	Power supply.
EN	3	I	Chip-enable signal. Active high.
SENSOR_VP	4	I	GPIO36, SENSOR_VP, ADC_H, ADC1_CH0, RTC_GPIO0
SENSOR_VN	5	I	GPIO39, SENSOR_VN, ADC1_CH3, ADC_H, RTC_GPIO3
IO34	6	I	GPIO34, ADC1_CH6, RTC_GPIO4
IO35	7	I	GPIO35, ADC1_CH7, RTC_GPIO5
IO32	8	I/O	GPIO32, XTAL_32K_P (32.768 kHz crystal oscillator input), ADC1_CH4, TOUCH9, RTC_GPIO9
IO33	9	I/O	GPIO33, XTAL_32K_N (32.768 kHz crystal oscillator output), ADC1_CH5, TOUCH8, RTC_GPIO8
IO25	10	I/O	GPIO25, DAC_1, ADC2_CH8, RTC_GPIO6, EMAC_RXD0
IO26	11	I/O	GPIO26, DAC_2, ADC2_CH9, RTC_GPIO7, EMAC_RXD1
IO27	12	I/O	GPIO27, ADC2_CH7, TOUCH7, RTC_GPIO17, EMAC_RX_DV

Name	No.	Type	Function
IO14	13	I/O	GPIO14, ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS, HSPICLK, HS2_CLK, SD_CLK, EMAC_TXD2
IO12	14	I/O	GPIO12, ADC2_CH5, TOUCH5, RTC_GPIO15, MTDI, HSPIQ, HS2_DATA2, SD_DATA2, EMAC_TXD3
GND	15	P	Ground
IO13	16	I/O	GPIO13, ADC2_CH4, TOUCH4, RTC_GPIO14, MTCK, HSPID, HS2_DATA3, SD_DATA3, EMAC_RX_ER
SHD/SD2*	17	I/O	GPIO9, SD_DATA2, SPIHD, HS1_DATA2, U1RXD
SWP/SD3*	18	I/O	GPIO10, SD_DATA3, SPIWP, HS1_DATA3, U1TXD
SCS/CMD*	19	I/O	GPIO11, SD_CMD, SPICS0, HS1_CMD, U1RTS
SCK/CLK*	20	I/O	GPIO6, SD_CLK, SPICLK, HS1_CLK, U1CTS
SDO/SD0*	21	I/O	GPIO7, SD_DATA0, SPIQ, HS1_DATA0, U2RTS
SDI/SD1*	22	I/O	GPIO8, SD_DATA1, SPID, HS1_DATA1, U2CTS
IO15	23	I/O	GPIO15, ADC2_CH3, TOUCH3, MTDO, HSPICS0, RTC_GPIO13, HS2_CMD, SD_CMD, EMAC_RXD3
IO2	24	I/O	GPIO2, ADC2_CH2, TOUCH2, RTC_GPIO12, HSPIWP, HS2_DATA0, SD_DATA0
IO0	25	I/O	GPIO0, ADC2_CH1, TOUCH1, RTC_GPIO11, CLK_OUT1, EMAC_TX_CLK
IO4	26	I/O	GPIO4, ADC2_CH0, TOUCH0, RTC_GPIO10, HSPIHD, HS2_DATA1, SD_DATA1, EMAC_TX_ER
IO16	27	I/O	GPIO16, HS1_DATA4, U2RXD, EMAC_CLK_OUT
IO17	28	I/O	GPIO17, HS1_DATA5, U2TXD, EMAC_CLK_OUT_180
IO5	29	I/O	GPIO5, VSPICS0, HS1_DATA6, EMAC_RX_CLK
IO18	30	I/O	GPIO18, VSPICLK, HS1_DATA7
IO19	31	I/O	GPIO19, VSPIQ, U0CTS, EMAC_TXD0
NC	32	-	-
IO21	33	I/O	GPIO21, VSPIHD, EMAC_TX_EN
RXD0	34	I/O	GPIO3, U0RXD, CLK_OUT2
TXD0	35	I/O	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
IO22	36	I/O	GPIO22, VSPIWP, U0RTS, EMAC_TXD1
IO23	37	I/O	GPIO23, VSPID, HS1_STROBE
GND	38	P	Ground

Note:

* Pins SCK/CLK, SDO/SD0, SDI/SD1, SHD/SD2, SWP/SD3 and SCS/CMD, namely, GPIO6 to GPIO11 are connected to the integrated SPI flash integrated on ESP32-WROOM-32 (ESP-WROOM-32) and are not recommended for other uses.

2.3 Strapping Pins

ESP32 has five strapping pins, which can be seen in Chapter 6 Schematics:

- MTDI
- GPIO0
- GPIO2
- MTDO
- GPIO5

Software can read the value of these five bits from the register "GPIO_STRAPPING".

During the chip's system reset (power-on reset, RTC watchdog reset and brownout reset), the latches of the strapping pins sample the voltage level as strapping bits of "0" or "1", and hold these bits until the chip is powered down or shut down. The strapping bits configure the device boot mode, the operating voltage of VDD_SDIO and other system initial settings.

Each strapping pin is connected with its internal pull-up/pull-down during the chip reset. Consequently, if a strapping pin is unconnected or the connected external circuit is high-impendence, the internal weak pull-up/pull-down will determine the default input level of the strapping pins.

To change the strapping bit values, users can apply the external pull-down/pull-up resistances, or apply the host MCU's GPIOs to control the voltage level of these pins when powering on ESP32.

After reset, the strapping pins work as the normal functions pins.

Refer to Table 3 for detailed boot modes' configuration by strapping pins.

Table 3: Strapping Pins

<table>
<tr><th colspan="6">Voltage of Internal LDO (VDD_SDIO)</th></tr>
<tr><td>Pin</td><td>Default</td><td colspan="2">3.3V</td><td colspan="2">1.8V</td></tr>
<tr><td>MTDI</td><td>Pull-down</td><td colspan="2">0</td><td colspan="2">1</td></tr>
<tr><th colspan="6">Booting Mode</th></tr>
<tr><td>Pin</td><td>Default</td><td colspan="2">SPI Boot</td><td colspan="2">Download Boot</td></tr>
<tr><td>GPIO0</td><td>Pull-up</td><td colspan="2">1</td><td colspan="2">0</td></tr>
<tr><td>GPIO2</td><td>Pull-down</td><td colspan="2">Don't-care</td><td colspan="2">0</td></tr>
<tr><th colspan="6">Debugging Log Printed on U0TXD During Booting?</th></tr>
<tr><td>Pin</td><td>Default</td><td colspan="2">U0TXD Toggling</td><td colspan="2">U0TXD Silent</td></tr>
<tr><td>MTDO</td><td>Pull-up</td><td colspan="2">1</td><td colspan="2">0</td></tr>
<tr><th colspan="6">Timing of SDIO Slave</th></tr>
<tr><td>Pin</td><td>Default</td><td>Falling-edge Input
Falling-edge Output</td><td>Falling-edge Input
Rising-edge Output</td><td>Rising-edge Input
Falling-edge Output</td><td>Rising-edge Input
Rising-edge Output</td></tr>
<tr><td>MTDO</td><td>Pull-up</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>GPIO5</td><td>Pull-up</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
</table>

Note:
Firmware can configure register bits to change the settings of "Voltage of Internal LDO (VDD_SDIO)" and "Timing of SDIO Slave" after booting.

3. Functional Description

This chapter describes the modules and functions integrated in ESP32-WROOM-32 (ESP-WROOM-32).

3.1 CPU and Internal Memory

ESP32-D0WDQ6 contains two low-power Xtensa® 32-bit LX6 microprocessors. The internal memory includes:

- 448 kB of ROM for booting and core functions.
- 520 kB (8 kB RTC FAST Memory included) of on-chip SRAM for data and instruction.
  - 8 kB of SRAM in RTC, which is called RTC FAST Memory and can be used for data storage; it is accessed by the main CPU during RTC Boot from the Deep-sleep mode.
- 8 kB of SRAM in RTC, which is called RTC SLOW Memory and can be accessed by the co-processor during the Deep-sleep mode.
- 1 kbit of eFuse, of which 320 bits are used for the system (MAC address and chip configuration) and the remaining 704 bits are reserved for customer applications, including Flash-Encryption and Chip-ID.

3.2 External Flash and SRAM

ESP32 supports up to four 16-MB of external QSPI flash and SRAM with hardware encryption based on AES to protect developers' programs and data.

ESP32 can access the external QSPI flash and SRAM through high-speed caches.

- Up to 16 MB of external flash are memory-mapped onto the CPU code space, supporting 8, 16 and 32-bit access. Code execution is supported.
- Up to 8 MB of external flash/SRAM are memory-mapped onto the CPU data space, supporting 8, 16 and 32-bit access. Data-read is supported on the flash and SRAM. Data-write is supported on the SRAM.

ESP32-WROOM-32 (ESP-WROOM-32) integrates 4 MB of external SPI flash. The 4-MB SPI flash can be memory-mapped onto the CPU code space, supporting 8, 16 and 32-bit access. Code execution is supported. The integrated SPI flash is connected to GPIO6, GPIO7, GPIO8, GPIO9, GPIO10 and GPIO11. These six pins cannot be used as regular GPIO.

3.3 Crystal Oscillators

The ESP32 Wi-Fi/BT firmware can only support 40 MHz crystal oscillator for now.

3.4 RTC and Low-Power Management

With the use of advanced power management technologies, ESP32 can switch between different power modes.

- Power modes
  - Active mode: The chip radio is powered on. The chip can receive, transmit, or listen.
  - Modem-sleep mode: The CPU is operational and the clock is configurable. The Wi-Fi/Bluetooth baseband and radio are disabled.
  - Light-sleep mode: The CPU is paused. The RTC memory and RTC peripherals, as well as the ULP co-processor are running. Any wake-up events (MAC, host, RTC timer, or external interrupts) will wake up the chip.
  - Deep-sleep mode: Only the RTC memory and RTC peripherals are powered on. Wi-Fi and Bluetooth connection data are stored in the RTC memory. The ULP co-processor can work.
  - Hibernation mode: The internal 8-MHz oscillator and ULP co-processor are disabled. The RTC recovery memory is powered down. Only one RTC timer on the slow clock and some RTC GPIOs are active. The RTC timer or the RTC GPIOs can wake up the chip from the Hibernation mode.

The power consumption varies with different power modes/sleep patterns and work statuses of functional modules. Please see Table 4 for details.

Table 4: Power Consumption by Power Modes

Power mode	Description	Power consumption
Active (RF working)	Wi-Fi TX packet 14 dBm ~ 19.5 dBm	Please refer to ESP32 Datasheet.
	Wi-Fi / BT TX packet 0 dBm	
	Wi-Fi / BT RX and listening	
	Association sleep pattern (by Light-sleep)	1 mA ~ 4 mA @DTIM3
Modem-sleep	The CPU is powered on.	Max speed 240 MHz: 30 mA ~ 50 mA
		Normal speed 80 MHz: 20 mA ~ 25 mA
		Slow speed 2 MHz: 2 mA ~ 4 mA
Light-sleep	-	0.8 mA
Deep-sleep	The ULP co-processor is powered on.	150 μA
	ULP sensor-monitored pattern	100 μA @1% duty
	RTC timer + RTC memory	10 μA
Hibernation	RTC timer only	5 μA
Power off	CHIP_PU is set to low level, the chip is powered off	0.1 μA

Note:

- When Wi-Fi is enabled, the chip switches between Active and Modem-sleep mode. Therefore, power consumption changes accordingly.
- In Modem-sleep mode, the CPU frequency changes automatically. The frequency depends on the CPU load and the peripherals used.
- During Deep-sleep, when the ULP co-processor is powered on, peripherals such as GPIO and I2C are able to work.
- When the system works in the ULP sensor-monitored pattern, the ULP co-processor works with the ULP sensor periodically; ADC works with a duty cycle of 1%, so the power consumption is 100 μA.

4. Peripherals and Sensors

Please refer to Section 4 Peripherals and Sensors in ESP32 Datasheet.

Note:
External connections can be made to any GPIO except for GPIOs in the range 6-11. These six GPIOs are connected to the module's integrated SPI flash. For details, please see Section 6 Schematics.

5. Electrical Characteristics

Note:
The specifications in this chapter have been tested under the following general condition: VDD = 3.3V, T_A = 27°C, unless otherwise specified.

5.1 Absolute Maximum Ratings

Table 5: Absolute Maximum Ratings

Parameter	Symbol	Min	Typ	Max	Unit
Power supply	VDD	2.7	3.3	3.6	V
Minimum current delivered by power supply	I_{VDD}	0.5	-	-	A
Input low voltage	V_{IL}	-0.3	-	$0.25 \times V_{IO}$[1]	V
Input high voltage	V_{IH}	$0.75 \times V_{IO}$[1]	-	V_{IO}[1]+0.3	V
Input leakage current	I_{IL}	-	-	50	nA
Input pin capacitance	C_{pad}	-	-	2	pF
Output low voltage	V_{OL}	-	-	$0.1 \times V_{IO}$[1]	V
Output high voltage	V_{OH}	$0.8 \times V_{IO}$[1]	-	-	V
Maximum output drive capability	I_{MAX}	-	-	40	mA
Storage temperature range	T_{STR}	-40	-	85	°C
Operating temperature range	T_{OPR}	-40	-	85	°C

1. V_{IO} is the power supply for a specific pad. More details can be found in the ESP32 Datasheet, Appendix IO_MUX. For example, the power supply for SD_CLK is the VDD_SDIO.

5.2 Wi-Fi Radio

Table 6: Wi-Fi Radio Characteristics

Description	Min	Typical	Max	Unit
Input frequency	2412	-	2484	MHz
Input reflection	-	-	-10	dB
Tx power				
Output power of PA for 72.2 Mbps	13	14	15	dBm
Output power of PA for 11b mode	19.5	20	20.5	dBm
Sensitivity				
DSSS, 1 Mbps	-	-98	-	dBm
CCK, 11 Mbps	-	-91	-	dBm
OFDM, 6 Mbps	-	-93	-	dBm
OFDM, 54 Mbps	-	-75	-	dBm
HT20, MCS0	-	-93	-	dBm
HT20, MCS7	-	-73	-	dBm

Description	Min	Typical	Max	Unit
HT40, MCS0	-	-90	-	dBm
HT40, MCS7	-	-70	-	dBm
MCS32	-	-89	-	dBm
Adjacent channel rejection				
OFDM, 6 Mbps	-	37	-	dB
OFDM, 54 Mbps	-	21	-	dB
HT20, MCS0	-	37	-	dB
HT20, MCS7	-	20	-	dB

5.3 BLE Radio

5.3.1 Receiver

Table 7: Receiver Characteristics — BLE

Parameter	Conditions	Min	Typ	Max	Unit
Sensitivity @30.8% PER	-	-	-97	-	dBm
Maximum received signal @30.8% PER	-	0	-	-	dBm
Co-channel C/I	-	-	+10	-	dB
Adjacent channel selectivity C/I	F = F0 + 1 MHz	-	-5	-	dB
	F = F0 - 1 MHz	-	-5	-	dB
	F = F0 + 2 MHz	-	-25	-	dB
	F = F0 - 2 MHz	-	-35	-	dB
	F = F0 + 3 MHz	-	-25	-	dB
	F = F0 - 3 MHz	-	-45	-	dB
Out-of-band blocking performance	30 MHz ~ 2000 MHz	-10	-	-	dBm
	2000 MHz ~ 2400 MHz	-27	-	-	dBm
	2500 MHz ~ 3000 MHz	-27	-	-	dBm
	3000 MHz ~ 12.5 GHz	-10	-	-	dBm
Intermodulation	-	-36	-	-	dBm

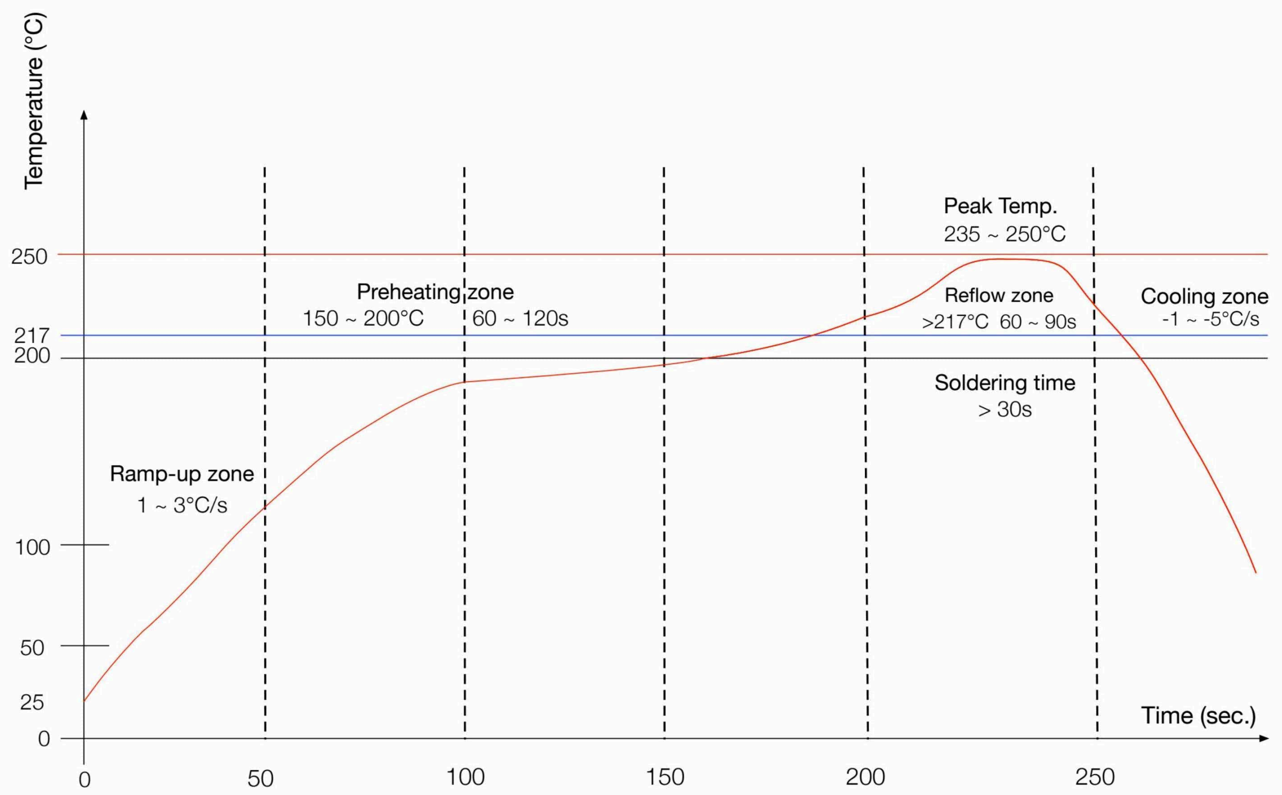
5.3.2 Transmitter

Table 8: Transmitter Characteristics — BLE

Parameter	Conditions	Min	Typ	Max	Unit
RF transmit power	-	-	0	-	dBm
Gain control step	-	-	±3	-	dBm
RF power control range	-	-12	-	+12	dBm

Parameter	Conditions	Min	Typ	Max	Unit
Adjacent channel transmit power	F = F0 + 1 MHz	-	-14.6	-	dBm
	F = F0 - 1 MHz	-	-12.7	-	dBm
	F = F0 + 2 MHz	-	-44.3	-	dBm
	F = F0 - 2 MHz	-	-38.7	-	dBm
	F = F0 + 3 MHz	-	-49.2	-	dBm
	F = F0 - 3 MHz	-	-44.7	-	dBm
	F = F0 + > 3 MHz	-	-50	-	dBm
	F = F0 - > 3 MHz	-	-50	-	dBm
$\Delta f1_{avg}$	-	-	-	265	kHz
$\Delta f2_{max}$	-	247	-	-	kHz
$\Delta f2_{avg}/\Delta f1_{avg}$	-	-	-0.92	-	-
ICFT	-	-	-10	-	kHz
Drift rate	-	-	0.7	-	kHz/50 μs
Drift	-	-	2	-	kHz

5.4 Reflow Profile



Ramp-up zone — Temp.: <150°C Time: 60 ~ 90s Ramp-up rate: 1 ~ 3°C/s
Preheating zone — Temp.: 150 ~ 200°C Time: 60 ~ 120s Ramp-up rate: 0.3 ~ 0.8°C/s
Reflow zone — Temp.: >217°C Time: 60 ~ 90s; Peak Temp.: 235 ~ 250°C (<245°C recommended) Time: 30 ~ 70s
Cooling zone — Peak Temp. ~ 180°C Ramp-down rate: -1 ~ -5°C/s
Solder — Sn&Ag&Cu Lead-free solder (SAC305)

Figure 2: Reflow Profile

6. Schematics

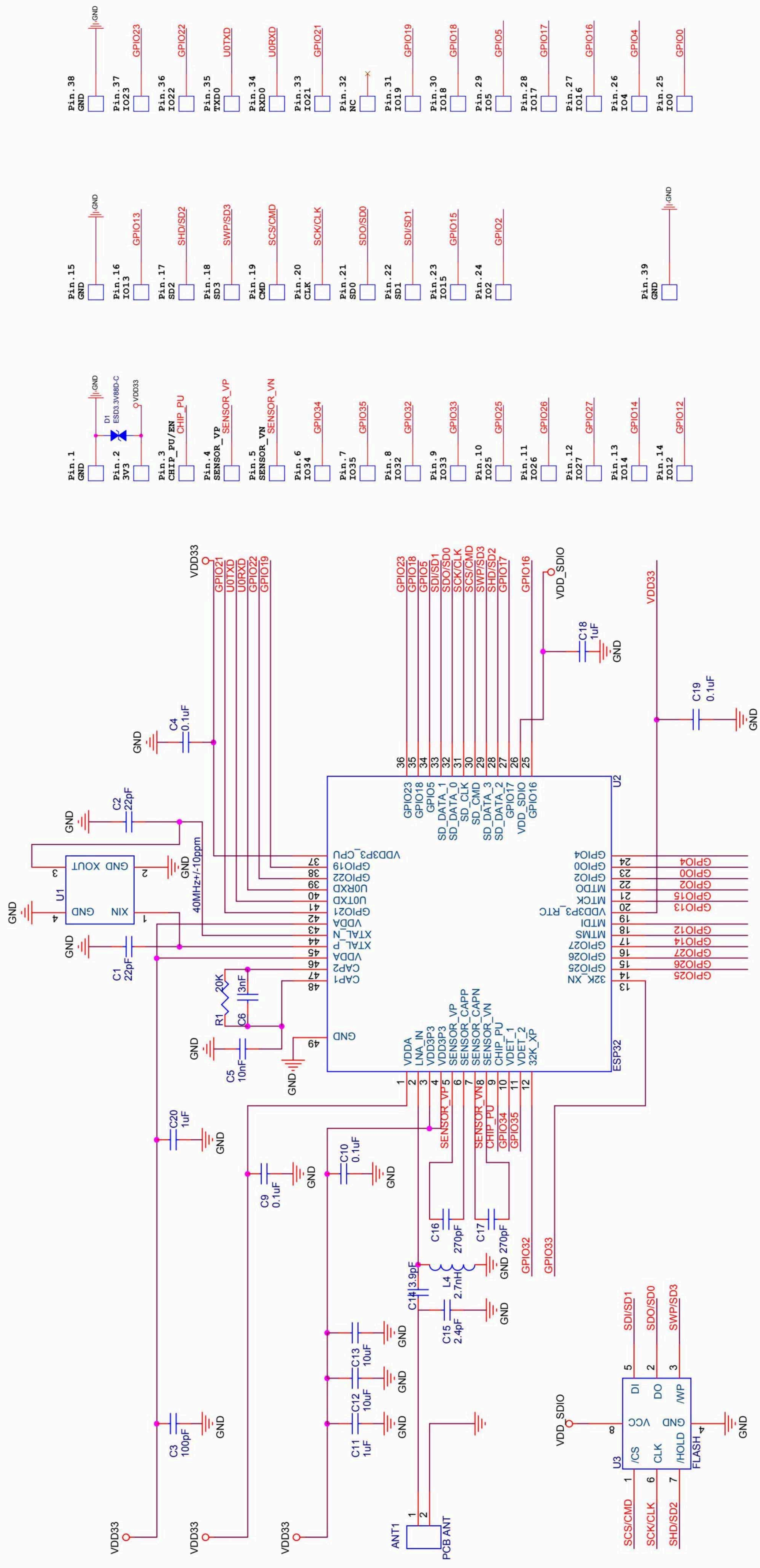


Figure 3: ESP32-WROOM-32 (ESP-WROOM-32) Schematics

7. Peripheral Schematics

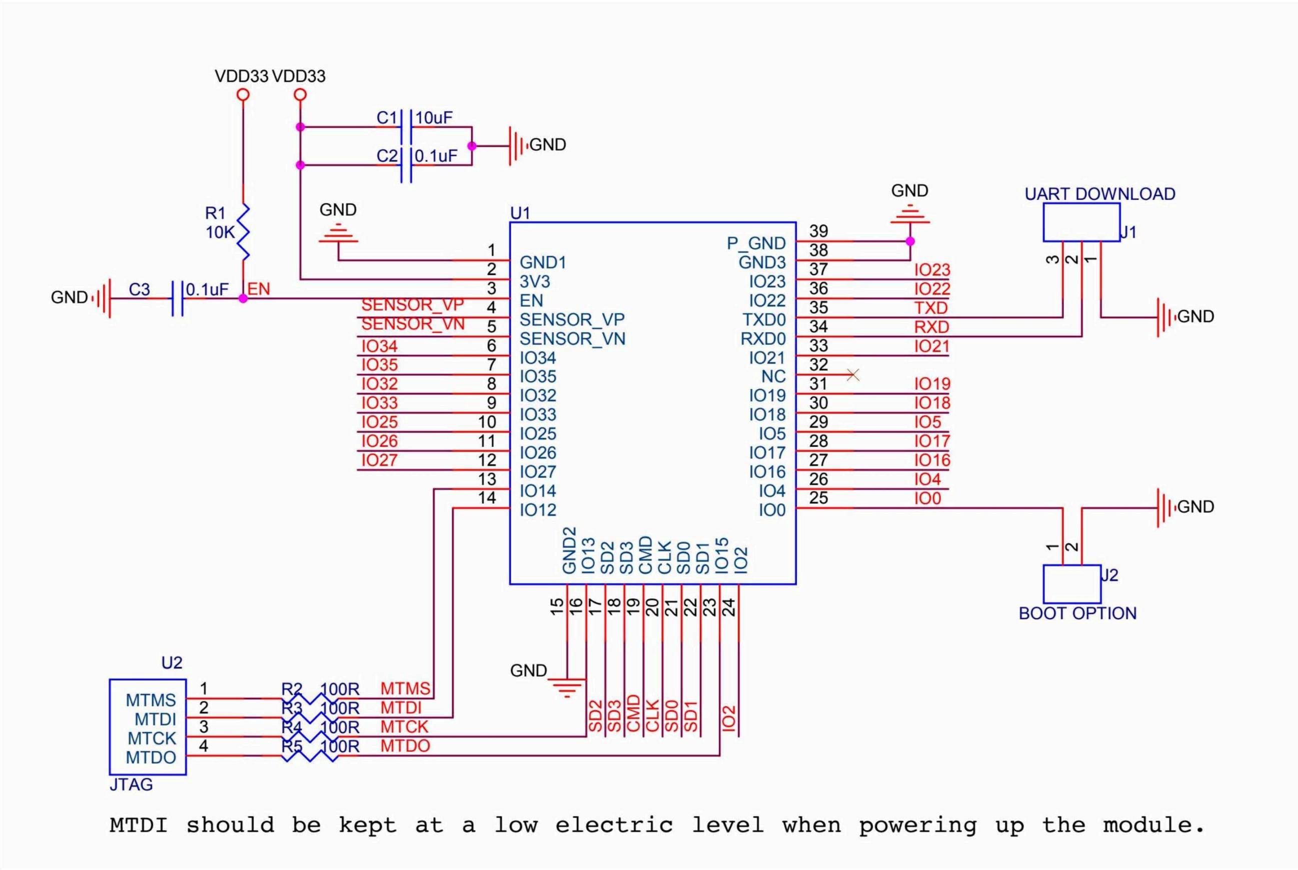


Figure 4: ESP32-WROOM-32 (ESP-WROOM-32) Peripheral Schematics

> **Note:**
> Soldering Pad 39 to the Ground of the base board is not necessary for a satisfactory thermal performance. If users do want to solder it, they need to ensure that the correct quantity of soldering paste is applied.

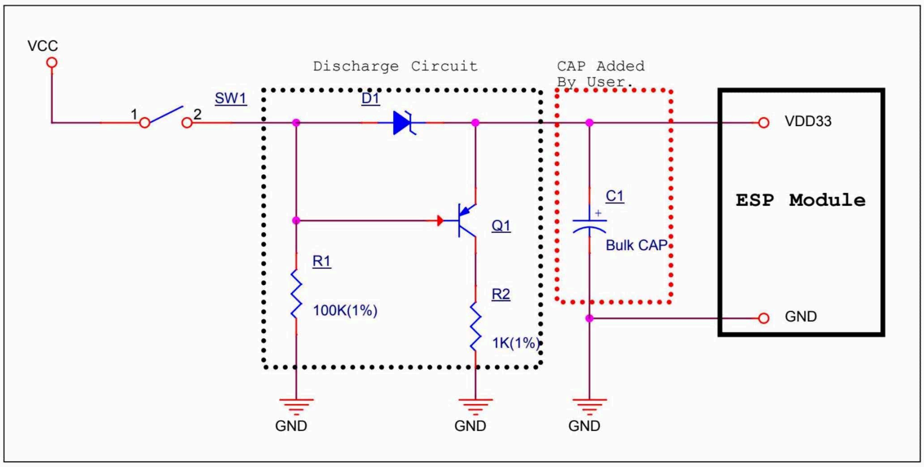


Figure 5: Discharge Circuit for VDD33 Rail

> **Note:**
> The discharge circuit can be applied in scenarios where ESP32 is powered on and off repeatedly by switching the power rails, and there is a large capacitor on the VDD33 rail. For details, please refer to Section **Power Scheme** in ESP32 Datasheet.

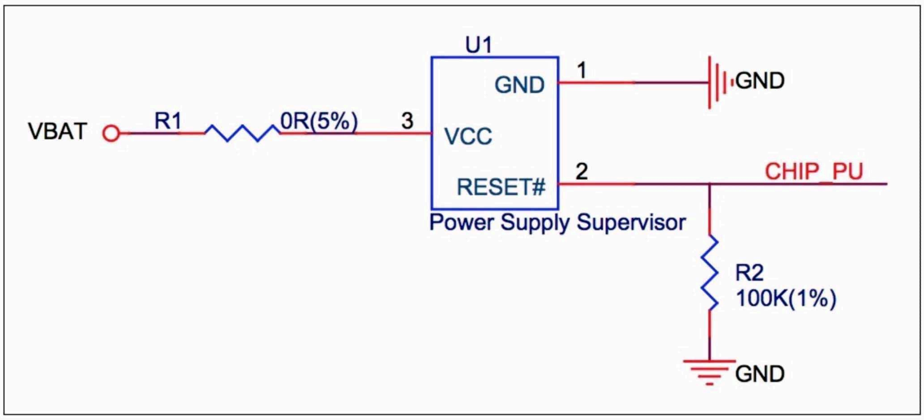


Figure 6: Reset Circuit

Note:

When battery is used as the power supply for ESP32 series of chips and modules, a supply voltage supervisor is recommended to avoid boot failure due to low voltage. Users are recommended to pull CHIP_PU low if the power supply for ESP32 is below 2.3V.

8. Dimensions

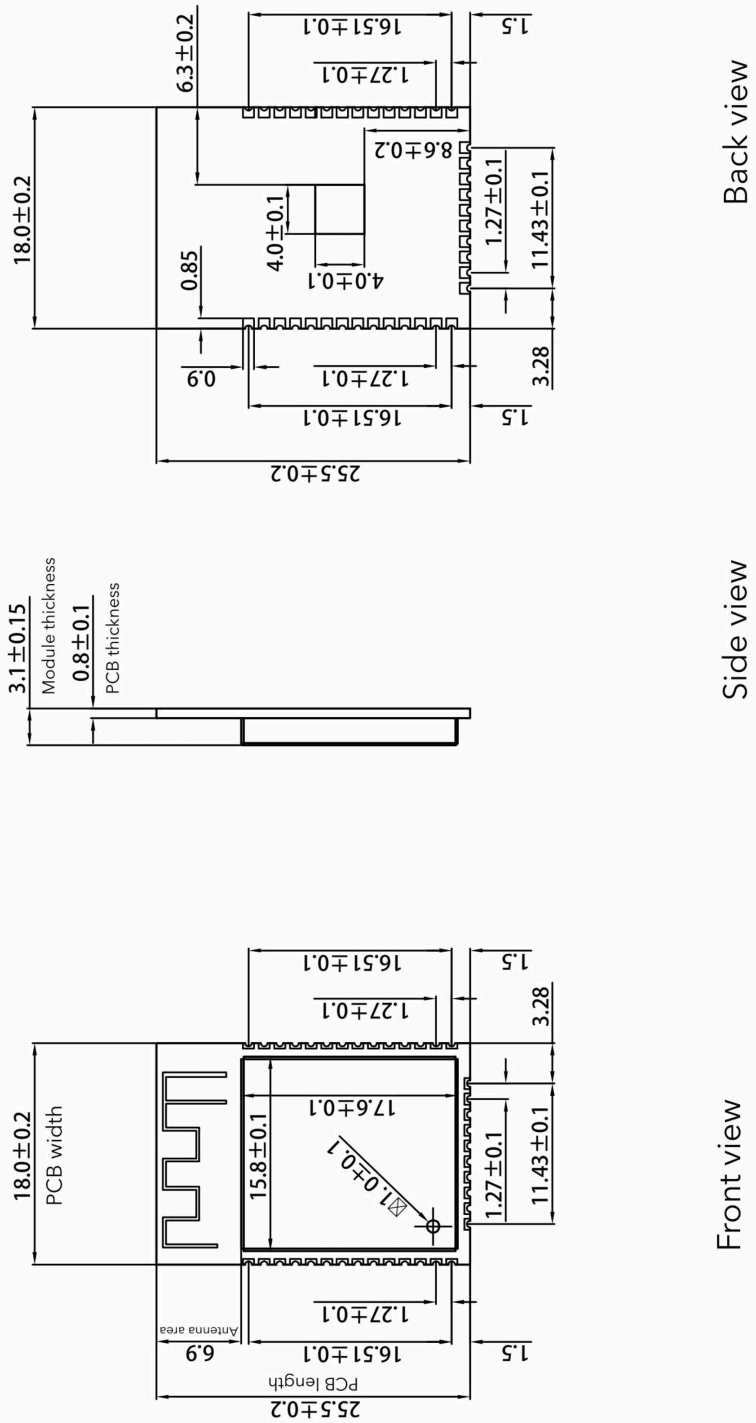


Figure 7: Dimensions of ESP32-WROOM-32 (ESP-WROOM-32)

Note:

All dimensions are in millimeters.

9. Learning Resources

9.1 Must-Read Documents

The following link provides documents related to ESP32.

- ESP32 Datasheet
  This document provides an introduction to the specifications of the ESP32 hardware, including overview, pin definitions, functional description, peripheral interface, electrical characteristics, etc.
- ESP-IDF Programming Guide
  It hosts extensive documentation for ESP-IDF ranging from hardware guides to API reference.
- ESP32 Technical Reference Manual
  The manual provides detailed information on how to use the ESP32 memory and peripherals.
- ESP32 Hardware Resources
  The zip files include the schematics, PCB layout, Gerber and BOM list of ESP32 modules and development boards.
- ESP32 Hardware Design Guidelines
  The guidelines outline recommended design practices when developing standalone or add-on systems based on the ESP32 series of products, including ESP32, the ESP-WROOM-32 module, and ESP32-DevKitC—the development board.
- ESP32 AT Instruction Set and Examples
  This document introduces the ESP32 AT commands, explains how to use them, and provides examples of several common AT commands.
- Espressif Products Ordering Information

9.2 Must-Have Resources

Here are the ESP32-related must-have resources.

- ESP32 BBS
  This is an Engineer-to-Engineer (E2E) Community for ESP32 where you can post questions, share knowledge, explore ideas, and help solve problems with fellow engineers.
- ESP32 GitHub
  ESP32 development projects are freely distributed under Espressif's MIT license on GitHub. It is established to help developers get started with ESP32 and foster innovation and the growth of general knowledge about the hardware and software surrounding ESP32 devices.
- ESP32 Tools
  This is a webpage where users can download ESP32 Flash Download Tools and the zip file "ESP32 Certification and Test".
- ESP-IDF
  This webpage links users to the official IoT development framework for ESP32.
- ESP32 Resources
  This webpage provides the links to all available ESP32 documents, SDK and tools.

Revision History

Date	Version	Release notes
2018.03	V2.4	Updated Table 1 in Chapter 1.
2018.01	V2.3	Deleted information on LNA pre-amplifier; Updated section 3.4 RTC and Low-Power Management; Added reset circuit in Chapter 7 and a note to it.
2017.10	V2.2	Updated the description of the chip's system reset in Section 2.3 Strapping Pins; Deleted "Association sleep pattern" in Table 4 and added notes to Active sleep and Modem-sleep; Updated the note to Figure 4 Peripheral Schematics; Added discharge circuit for VDD33 rail in Chapter 7 and a note to it.
2017.09	V2.1	Updated operating voltage/power supply range updated to 2.7 ~ 3.6V; Updated Chapter 7.
2017.08	V2.0	Changed the sensitivity of NZIF receiver to -97 dBm in Table 1; Updated the dimensions of the module; Updated Table 4 Power Consumption by Power Modes, and added two notes to it; Updated Table 5, 6, 7, 8; Added Chapter 8; Added the link to certification download.
2017.06	V1.9	Added a note to Section 2.1 Pin Layout; Updated Section 3.3 Crystal Oscillators; Updated Figure 3 ESP-WROOM-32 Schematics; Added Documentation Change Notification.
2017.05	V1.8	Updated Figure 1 Top and Side View of ESP32-WROOM-32 (ESP-WROOM-32).
2017.04	V1.7	Added the module's dimensional tolerance; Changed the input impedance value of 50Ω in Table 6 Wi-Fi Radio Characteristics to output impedance value of 30+j10 Ω.
2017.04	V1.6	Added Figure 2 Reflow Profile.
2017.03	V1.5	Updated Section 2.2 Pin Description; Updated Section 3.2 External Flash and SRAM; Updated Section 4 Peripherals and Sensors Description.
2017.03	V1.4	Updated Chapter 1 Preface; Updated Chapter 2 Pin Definitions; Updated Chapter 3 Functional Description; Updated Table Recommended Operating Conditions; Updated Table 6 Wi-Fi Radio Characteristics; Updated Section 5.4 Reflow Profile; Added Chapter 9 Learning Resources.
2016.12	V1.3	Updated Section 2.1 Pin Layout.
2016.11	V1.2	Added Figure 7 Peripheral Schematics.
2016.11	V1.1	Updated Chapter 6 Schematics.
2016.08	V1.0	First release.