

# **VIRTUAL TELEPRESENCE ROBOT**

*A Project report submitted in partial fulfillment of the requirements for  
the award of the degree of*

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

*Submitted by*

G Sanjay Bhargav (317126512079)

K Vijaya Tej(3181265120L24)

P Jaswanth (3181265120L21)

M Naga Pujitha (317126512088)

**Under the guidance of**

**Nitin Kumar**

**Asst. professor**



**ANITS**

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES  
(UGC AUTONOMOUS)**

*(Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade)*  
Sangivalasa, bheemili mandal, visakhapatnam dist.(A.P)  
2020-2021

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



**CERTIFICATE**

*This is to certify that the project report entitled “VIRTUAL TELEPRESENCE ROBOT” submitted by G SanjayBhargav (317126512079), K Vijaya Tej (3181265120L24), P Jaswanth (3181265120L21), M Naga Pujitha (317126512088) in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Electronics & Communication Engineering** of Andhra University, Visakhapatnam is a record of bonafide work carried out under my guidance and supervision.*

**Project Guide**  
**Nitin Kumar**

Asst. Professor  
Department of E.C.E  
ANITS

**Head of the Department**  
**Dr. V.Rajyalakshmi**  
**Professor&HOD**  
Department of E.C.E  
ANITS

## **ACKNOWLEDGEMENT**

We would like to express our deep gratitude to our project guide **Nitin Kumar** Asst. professor, Department of Electronics and Communication Engineering, ANITS, for his/her guidance with unsurpassed knowledge and immense encouragement. We are grateful to **Dr. V. Rajyalakshmi**, Head of the Department, Electronics and Communication Engineering, for providing us with the required facilities for the completion of the project work.

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

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

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

## **PROJECT STUDENTS**

G Sanjay Bhargav (3171261512079),

K Vijaya Tej(3181265120L24),

P Jaswanth(3181265120L21),

M Naga Pujitha(317126512088)

## **ABSTRACT**

A Telepresence robot is a remote-controlled, wheeled device with a display to enable video streaming which allows the participants to view remote locations, as if they were there. The project consists of a VR headset, with a smartphone in dual screen to experience virtual reality and 4 wheeled robotic vehicles. The movement of the Robot can be controlled by using an android app. The motion of camera of the robot is controlled by the accelerometer and magnetometer data processed by the Raspberry Pi. Video streamed is received by the smartphone using the IP address specified by the Raspberry Pi. This robot with a camera is placed in a remote location to capture the environment in visual form using Raspberry Pi. The captured visuals are displayed on the users virtual reality (VR) headset. An added feature allows the camera to move in the direction of the users head movements. This gives the user a real-time experience as if he is present where the robot is located. The robot can also be moved in any direction through an app installed in the users smartphone.

The cost is efficient for tele-presence robotic platform for complete and immersive remote operation, with raspberry Pi and suggestive feedback and real time feeling for the user in indoor environments such as hospitals, museums. Providing accessibility to live streams in harsh environments such as war prone areas and dust affected regions, nuclear blasts or industrially radioactive zones using remote links. Addressing the major problem of lack of immersion in a Telepresence Robotic system by integrating Virtual Reality headsets (such as Google Cardboard) and online real-time head movement control and stereoscopic stream to traditional telepresence platforms controlled remotely and wirelessly via the Internet.

# CONTENTS

<b>LIST OF FIGURES</b>	<b>7</b>
<b>CHAPTER 1 INTRODUCTION</b>	<b>8</b>
1.1 Project Objective	8
1.2 Project Outline	8
<b>CHAPTER 2</b>	<b>9</b>
2.1 Introduction	9
2.2 Block Diagram	9
2.3 Methodology	10
2.4 Virtual Reality	10
2.5 Telepresence	10
2.6 Component/software used	11
2.6.1 Hardware components	11
2.6.2 Software used	11
<b>CHAPTER 3 HARDWARE COMPONENTS</b>	<b>12</b>
3.1 Raspberry Pi	12
3.1.1 About Rpi	12
3.1.2 Rpi components	13
3.1.3 Raspberry Pi Ports	16
3.1.4 Specifications	22
3.1.5 Reasons for choosing Rpi	23
3.1.6 Rpi and Video Broadcasting	23
3.2 Node_Mcu	23
3.2.1 About Node_MCU	23
3.2.2 Specifications and Features	25
3.3 Power supply	25
3.4 Motor Driver(L293D)	26
3.4.1 About L293D	26

3.4.2 L293D Features	28
3.5 Geared Motor	28
3.6 VR Headset	29
3.7 Servo Motor	30
3.7.1 About servo motor	30
3.7.2 control of camera using servo motors	30
3.7.3 Flow chart for servo initializing	31
3.7.4 Servo mechanism	32
<b>CHAPTER 4 SOFTWARES USED</b>	<b>33</b>
4.1 Python	33
4.2 Raspbian OS	34
4.2.1 About Raspbian OS	34
4.2.2 Installing Raspbian OS in Raspberry pi	35
4.2.3 Setting up the software	36
4.3 Arduino sketch for Node_MCU	39
4.3.1 About Arduino IDE	39
4.3.2 Installing Arduino IDE in windows10	39
4.3.3 How to program Node_MCU using Arduino IDE	43
4.4 Application required for movement of camera	45
4.5 Application required for movement of Robot	46
<b>CHAPTER 5 LIVE STREAMING</b>	<b>47</b>
5.1 About streaming	47
5.2 Implementing Streaming With Flask	47
5.3 Building a Live Video Streaming Server	47
5.4 Obtaining Frames from a Video Camera	49
5.5 Limitations	49
<b>CHAPTER 6 WORKING OF THE PROJECT</b>	<b>51</b>
<b>CHAPTER 7 RESULTS AND DISCUSSIONS</b>	<b>53</b>
<b>CONCLUSIONS</b>	<b>54</b>
<b>REFERENCES</b>	<b>55</b>
<b>PAPER PUBLICATION DETAILS</b>	

## LIST OF FIGURES

<b>Figure no</b>	<b>Title</b>	<b>Page no</b>
Fig 2.1	Block Diagram	9
Fig 3.1	Raspberry Pi	12
Fig 3.2	The Raspberry Pi's system-on-chip	13
Fig 3.3	The Raspberry Pi's random access memory	14
Fig 3.4	The Raspberry Pi's radio module	15
Fig. 3.5	The Raspberry Pi's power management integrated circuit	16
Fig 3.6	The Raspberry Pi's USB ports	17
Fig 3.7	The Raspberry Pi's Ethernet port	18
Fig 3.8	The Raspberry Pi's 3.5 mm AV jack	19
Fig. 3.9	The Raspberry Pi's camera connector	19
Fig 3.10	The Raspberry Pi's HDMI port	20
Fig. 3.11	The Raspberry Pi's micro USB power port	21
Fig 3.12	The Raspberry Pi's display connector (DSI)	21
Fig 3.13	The Raspberry Pi's microSD card connector	22
Fig 3.14	Node MCU ESP8266	23
Fig 3.15	Node MCU Pin diagram	24
Fig 3.16	Power bank	25
Fig 3.17	MOTOR DRIVER	26
Fig 3.18	Motor Driver(L293D) Pin Diagram	27
Fig 3.19	Geared motor	28
Fig 3.20	VR headset	29
Fig 3.21	Servo motor	30
Fig 3.22	Flowchart for servo motor initializing	31
Fig 4.1	Thonny python IDE	33
Fig 4.2	The NOOBS menu without any operating systems installed	36
Fig 4.3	The NOOBS menu with Raspbian operating systems installed	37
Fig 4.4	Installing Raspbian OS	38
Fig 4.5	Raspbian Desktop	38
Fig 4.6	Arduino Installation option	39
Fig 4.7	Arduino Installation Folder	40
Fig 4.8	Arduino Installation process window	41
Fig 4.9	Arduino Installation complete window	41
Fig 4.10	Arduino Icon	42
Fig 4.11	Arduino IDE software	42
Fig 4.12	Choosing preferences for ESP8266	43
Fig 4.13	Wireless IMU App	45
Fig 4.14	Application for movement of robot	46

## **CHAPTER 1**

### **INTRODUCTION**

#### **1.1 Project Objective:**

A Telepresence robot is a remote-controlled, wheeled device with a display to enable video streaming which allows the participants to view remote locations in virtual reality , as if they were there at that place.

#### **Motivation:**

- 1.Real time feeling for the user in indoor environments such as hospitals, museums.
- 2.Providing accessibility to live streams in harsh environments such as war prone areas and dust affected regions, nuclear blasts or industrially radioactive zones using remote links.

#### **1.2 Project Outline:**

A Telepresence robot is a remote-controlled, wheeled device with a display to enable video streaming which allows the participants to view remote locations, as if they were there. The project consists of a VR headset, with a smartphone in dual screen to experience virtual reality and 4 wheeled robotic vehicles. The movement of the Robot can be controlled by using an android app. The motion of camera of the robot is controlled by the accelerometer and magnetometer data processed by the Raspberry Pi. Video streamed is received by the smartphone using the IP address specified by the Raspberry Pi. This robot with a camera is placed in a remote location to capture the environment in visual form using Raspberry Pi. The captured visuals are displayed on the users virtual reality (VR) headset. An added feature allows the camera to move in the direction of the users head movements. This gives the user a real-time experience as if he is present where the robot is located. The robot can also be moved in any direction through an app installed in the users smartphone



## CHAPTER 2

### 2.1 INTRODUCTION:

A telepresence robot is a remote-controlled, wheeled device that has wireless internet connectivity (wifi technology). Typically, the robot uses a smartphone to provide video streaming. Gives information about surroundings at where the robot is placed. Virtual presence can be considered as a special case of telepresence where the remote environment is artificially created. Camera is interfaced with Raspberry Pi which allows it to live stream and record the video. Also Node MCU plays the role of driving a robot through a motor driver. Robot is also controlled using a mobile application which allows us to send any direction command to the robot. However, operating and navigating these robots by individuals who have little knowledge and map of the remote environment is challenging.. Typically, the robot uses a smartphone to provide video and audio capabilities. Gives information about surrounding. Virtual presence can be considered as a special case of telepresence where the remote environment is artificially created. Camera is interfaced with Raspberry Pi which allows it to live stream and record the video. Also Node MCU plays the role of driving a robot through a motor driver. Robot is also controlled using a mobile application which allows us to send any direction command to the robot.

### 2.2 BLOCK DIAGRAM:

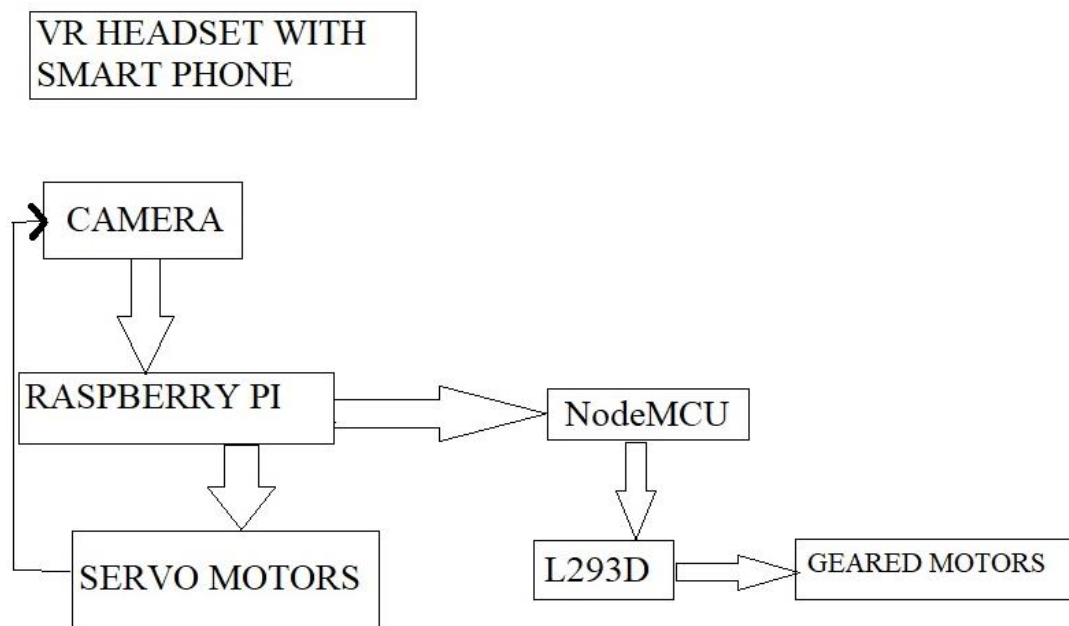


Fig 2.1 Block Diagram

## 2.3 METHODOLOGY:

The video captured by RPi camera can be viewed on a smartphone placed in the virtual reality (VR) headset. This project lets the user experience virtual reality through VR headset. Dual-screen mode is enabled in the smartphone for this purpose. The block diagram of virtual telepresence robot is shown in Fig. 1.

RPi is the brain of the system. It receives input from the smartphone via Wi-Fi, and then sends controlling pulse to the servo motors to move the RPi camera. The smartphone reads the accelerometer and magnetometer data of the direction in which the user turns his head, say, right or left. This data is sent to the modem over Wi-Fi and to the RPi board, which, in turn, provides these values as inputs to the servo motors. Two servo motors are used to move the camera—one for the vertical movement and the other for the horizontal movement. So when we turn our head along with VR headset to the right side, the RPi camera will also turn to the right direction.

## 2.4 Virtual Reality:

Virtual Reality (VR) is the use of computer technology to create a simulated environment. Unlike traditional user interfaces, VR places the user inside an experience. Instead of viewing a screen in front of them, users are immersed and able to interact with 3D worlds. By simulating as many senses as possible, such as vision, hearing, touch, even smell, the computer is transformed into a gatekeeper to this artificial world. The only limits to near-real VR experiences are the availability of content and cheap computing power.

## 2.5 Telepresence :

Telepresence utilizes technology closely related to that of VR. Telepresence is a medium in which transducers, such as video cameras and microphones, substitute for the corresponding senses of the participant. The participant is able to remotely see and hear from the first person POV with the aid of sensing devices in a remote location. The user can interact and affect the remote environment by actions reproduced on the remote end. Telepresence differs from the general case of VR by representing the physical world as opposed to representing a world that is entirely computer generated.

Telepresence is an application that uses VR-related technology to virtually place the user somewhere else in space whether in an adjoining room or on a neighbouring planet. Perhaps the primary reason the term *presence* is used by so many VR practitioners is because *telepresence* had been an established term in the related domain of remote control operations prior to the advent of VR.

*Tele* means distant and *present* is the state of being present or here. Examples of telepresence include remote manipulation of probes in the deep sea, working with dangerous chemicals, controlling operations on a space probe, or even manipulating surgical instruments just a few feet away.

## **2.6 Software Tools/Components used:**

### **2.6.1 Hardware Components:**

- a. Raspberry Pi board
- b. Node\_MCU
- c. Servo motors
- d. DC Geared motors
- e. Robot chasis
- f. VR Headset
- g. 12 Charger
- h. 8V power supply

### **2.6.2 Software used:**

- a. Application required for fetching of data from user and for controlling the bot
- b. Application required for movement of camera
- c. Python
- d. Arduino sketch for Node\_MCU.
- e. Raspbian operating system

## CHAPTER 3

### HARDWARE COMPONENTS

#### 3.1 Raspberry Pi :

##### 3.1.1 About Raspberry Pi:

The Raspberry pi is a single computer board with credit card size, that can be used for many tasks that your computer does, like games, word processing, spreadsheets and also to play HD video. It was established by the Raspberry pi foundation from the UK. It has been ready for public consumption since 2012 with the idea of making a low-cost educational microcomputer for students and children. The main purpose of designing the raspberry pi board is, to encourage learning, experimentation and innovation for school level students. The raspberry pi board is a portable and low cost. Maximum of the raspberry pi computers is used in mobile phones. In the 20th century, the growth of mobile computing technologies is very high, a huge segment of this being driven by the mobile industries. The 98% of the mobile phones were using ARM technology.

Raspberry Pi 4 Model B is the latest product in the popular Raspberry Pi range of computers. It offers ground-breaking increases in processor speed, multimedia performance, memory, and connectivity compared to the prior-generation Raspberry Pi 3 Model B+, while retaining backwards compatibility and similar power consumption. For the end user, Raspberry Pi 4 Model B provides desktop performance comparable to entry-level x86 PC systems. This product's key features include a high-performance 64-bit quad-core processor, dual-display support at resolutions up to 4K via a pair of micro-HDMI ports, hardware video decode at up to 4Kp60, up to 8GB of RAM, dual-band 2.4/5.0 GHz wireless LAN, Bluetooth 5.0, Gigabit Ethernet, USB 3.0, and PoE capability (via a separate PoE HAT add-on). The dual-band wireless LAN and Bluetooth have modular compliance certification, allowing the board to be designed into end products with significantly reduced compliance testing, improving both cost and time to market.

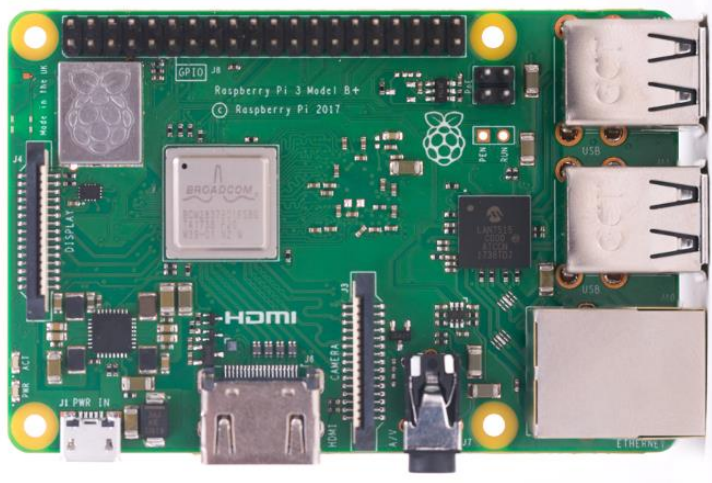


Fig 3.1 Raspberry Pi

### 3.1.2 RASPBERRY PI COMPONENTS :

Like any computer, the Pi is made up of various different components, each of which has a role to play in making it work. The first, and arguably most important, of these can be found just above the centre point on the top side of the board (**Figure 3-2**), covered in a metal cap: the *system-on-chip* (SoC).



Fig 3.2 The Raspberry Pi's system-on-chip (SoC)

The name system-on-chip is a great indicator of what you would find if you prised the metal cover off: a silicon chip, known as an *integrated circuit*, which contains the bulk of the Raspberry Pi's system. This includes the *central processing unit* (CPU), commonly thought of as the 'brain' of a computer, and the *graphics processing unit* (GPU), which handles the visual side of things. A brain is no good without memory, however, and on the underside of the Raspberry Pi you'll find exactly that: another chip, which looks like a small, black, plastic square (**Figure 3-3**). This is the Pi's *random access memory* (RAM).

When you're working on the Pi, it's the RAM that holds what you're doing; only when you save your work will it be written to the microSD card. Together, these components form the Pi's volatile and non-volatile memories: the volatile RAM loses its contents whenever the Pi is powered off, while the non-volatile microSD card keeps its contents.





Fig 3.3 The Raspberry Pi's random access memory (RAM)

Turning the board over again you'll find another metal lid to the upper-right, this one featuring an etched Raspberry Pi logo (**Figure 3-4**, overleaf). This covers the *radio*, the component which gives the Raspberry Pi the ability to communicate with devices wirelessly. The radio itself acts as two main components, in fact: a *WiFi radio*, for connecting to computer networks; and a *Bluetooth radio*, for connecting to peripherals like mice and for sending data to or receiving data from nearby smart devices like sensors or smartphones.

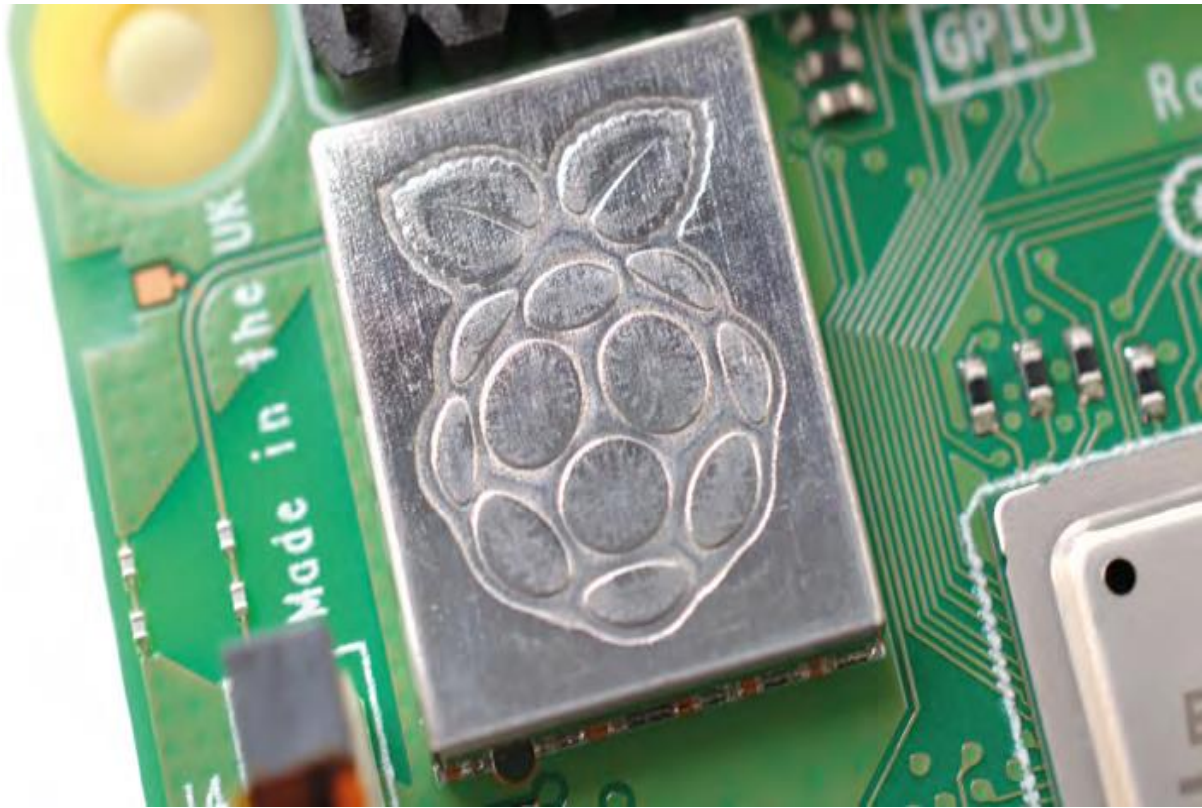


Fig 3.4 The Raspberry Pi's radio module

Another black, plastic-covered chip can be seen to the bottom edge of the board, just behind the middle set of USB ports. This is the *network and USB controller*, and is responsible for running the Ethernet port and the four USB ports. A final black chip, much smaller than the rest, can be found a little bit above the micro USB power connector to the upper-left of the board (**Figure 3-5**); this is known as a *power management integrated circuit (PMIC)*, and handles turning the power that comes in from the micro USB port into the power the Pi needs to run.



Fig. 3.5 The Raspberry Pi's power management integrated circuit (PMIC)

### 3.1.3 The Raspberry Pi's ports :

The Raspberry Pi has a range of ports, starting with four *Universal Serial Bus (USB) ports* (**Figure 3-6**) to the middle and right-hand side of the bottom edge. These ports let you connect any USB-compatible peripheral, from keyboards and mice to digital cameras and flash drives, to the Pi. Speaking technically, these are known as USB 2.0 ports, which means they are based on version two of the Universal Serial Bus standard.





Fig 3.6 The Raspberry Pi's USB ports

To the left of the USB ports is an *Ethernet port*, also known as a *network port* (**Figure 3-7**). You can use this port to connect the Raspberry Pi to a wired computer network using a cable with what is known as an RJ45 connector on its end. If you look closely at the Ethernet port, you'll see two light-emitting diodes (LEDs) at the bottom; these are status LEDs, and let you know that the connection is working.



Fig 3.7 The Raspberry Pi's Ethernet port

Just above the Ethernet port, on the left-hand edge of the Raspberry Pi, is a *3.5 mm audio-visual (AV) jack* (**Figure 3-8**). This is also known as the *headphone jack*, and it can be used for that exact purpose – though you'll get better sound connecting it to amplified speakers rather than headphones. It has a hidden, extra feature, though: as well as audio, the 3.5 mm AV jack carries a video signal which can be connected to TVs, projectors, and other displays that support a *composite video signal* using a special cable known as a *tip-ring-ring-sleeve (TRRS)* adapter.

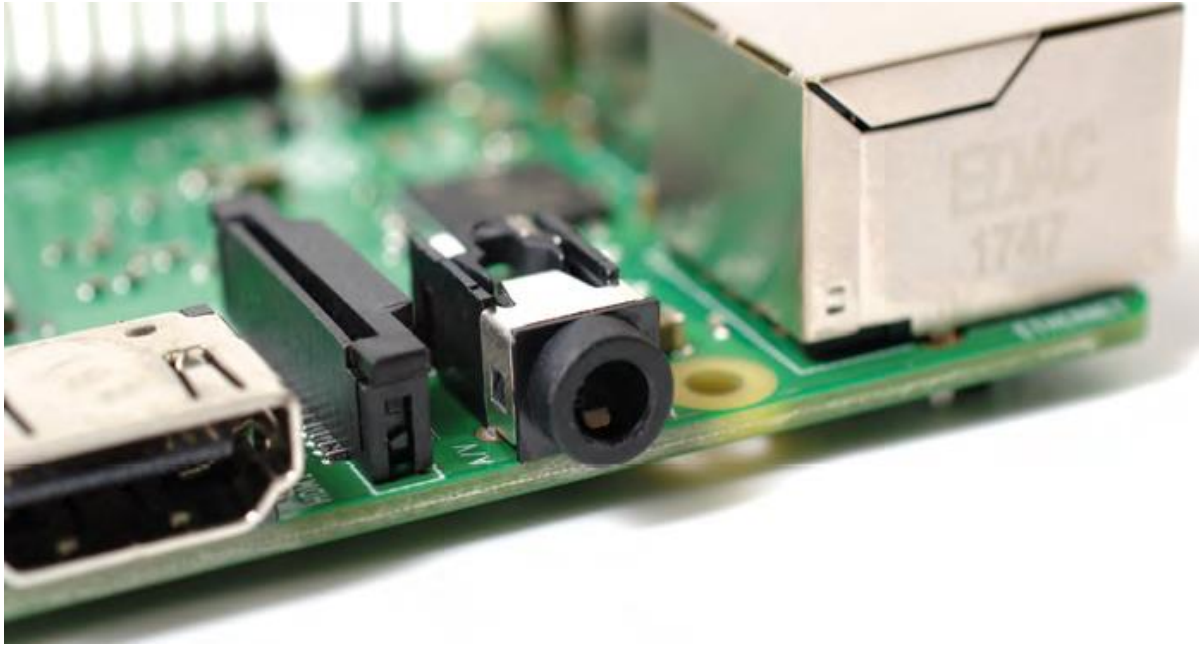


Fig 3.8 The Raspberry Pi's 3.5 mm AV jack

Directly above the 3.5 mm AV jack is a strange-looking connector with a plastic flap which can be pulled up; this is the *camera connector*, also known as the *Camera Serial Interface (CSI)* (**Figure 3-9**). This allows you to use the specially designed Raspberry Pi Camera Module.

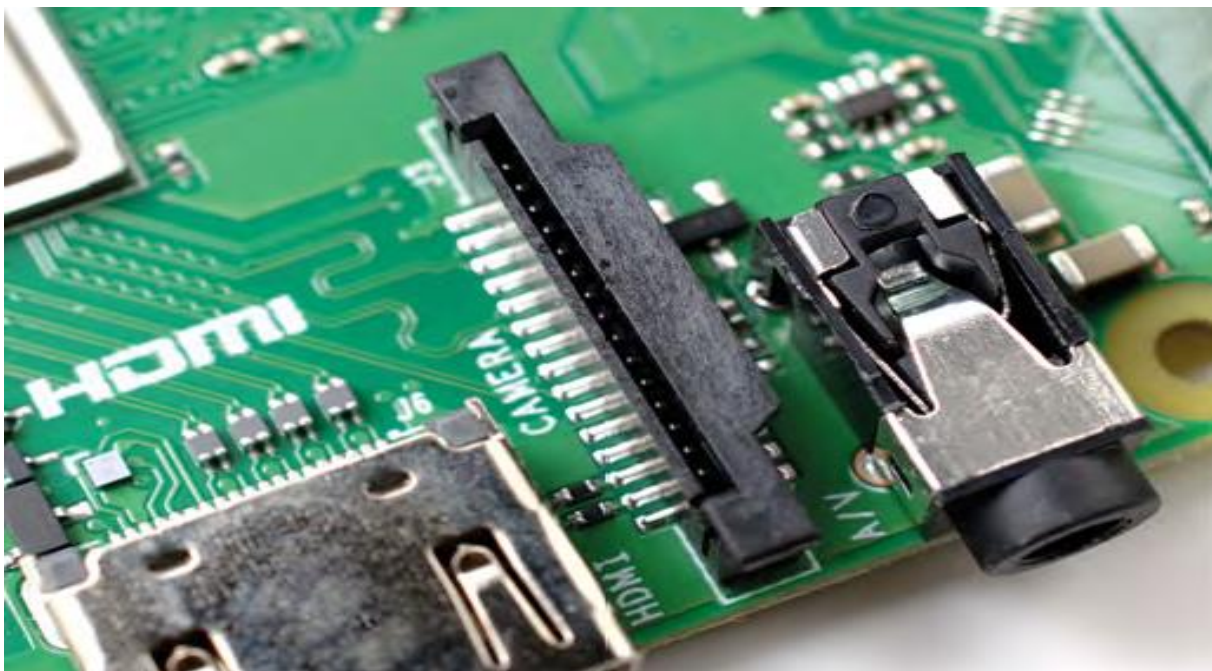


Fig 3-9 The Raspberry Pi's camera connector

Above that, still on the left-hand edge of the board, is the *High-Definition Multimedia Interface (HDMI) port* (**Figure 3-10**), which is the same type of connector you'll find on a games console, set-top box, and TV. The multimedia part of its name tells you that it carries both audio and video signals, while high-definition tells you that you can expect excellent quality. You'll use this to connect the Raspberry Pi to your display device, whether that's a computer monitor, TV, or projector.



Fig. 3.10 The Raspberry Pi's HDMI port

Above the HDMI port is a *micro USB power port* (**Figure 3-11**), which you'll use to connect the Raspberry Pi to a power source. The micro USB port is a common sight on smartphones, tablets, and other portable devices. So you could use a standard mobile charger to power the Pi, but for best results you should use the official Raspberry Pi USB Power Supply.





Fig 3.11 The Raspberry Pi's micro USB power port

At the top edge of the board is another strange-looking connector (**Figure 3-12**), which at first glance appears to be identical to the camera connector. This, though, is the exact opposite: a *display connector*, or *Display Serial Interface (DSI)*, designed for use with the Raspberry Pi Touch Display.



Fig 3.12 The Raspberry Pi's display connector (DSI)

There's one final port on the Raspberry Pi, but you won't see it on the top. Turn the board over and you'll find a microSD card connector on the opposite side of the board to the display connector (**Figure 1-15**). This is the Raspberry Pi's storage: the microSD card inserted in here contains all the files you save, all the software you install, and the operating system that makes the Raspberry Pi run.

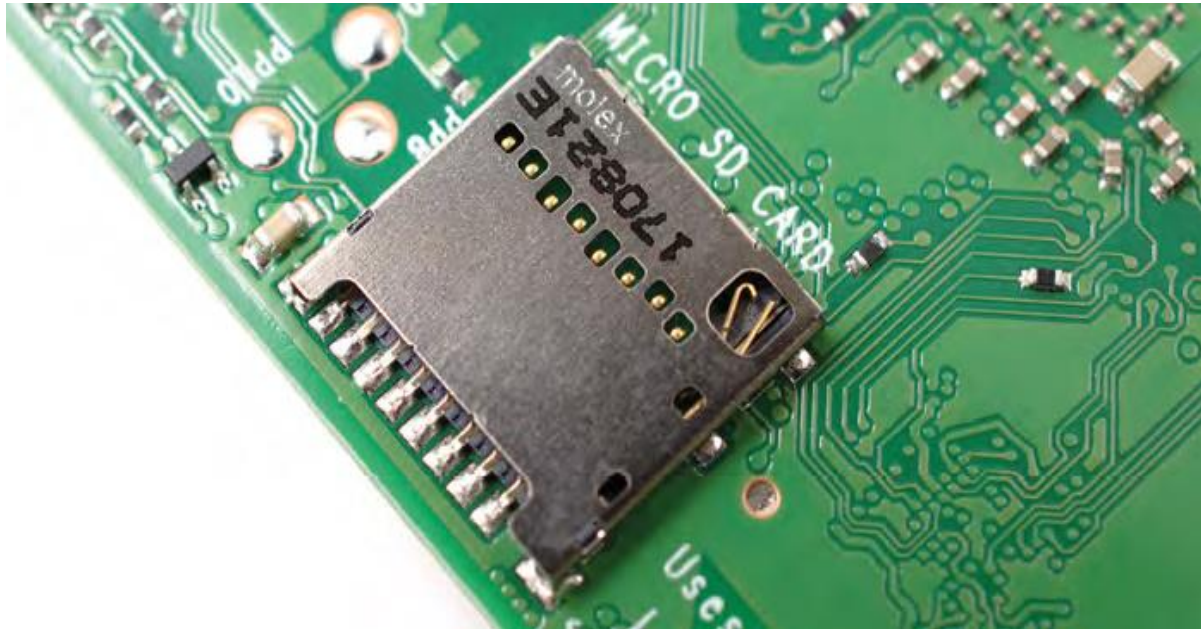


Fig. 3.13 The Raspberry Pi's microSD card connector

### 3.1.4 Specifications:

1. Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
2. 2GB, 4GB or 8GB LPDDR4-3200 SDRAM (depending on model)
3. 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
4. Gigabit Ethernet
5. 2 USB 3.0 ports; 2 USB 2.0 ports.
6. Raspberry Pi standard 40 pin GPIO header (fully backwards compatible with previous boards)
7. 2 × micro-HDMI ports (up to 4kp60 supported)
8. 2-lane MIPI DSI display port and 2-lane MIPI CSI camera port
9. 4-pole stereo audio and composite video port
10. H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)
11. Micro-SD card slot for loading operating system and data storage
12. 5V DC via USB-C connector (minimum 3A\*)
13. 5V DC via GPIO header (minimum 3A\*)

14. Power over Ethernet (PoE) enabled (requires separate PoE HAT)
15. Operating temperature: 0 – 50 degrees C ambient.

### 3.1.5 REASONS FOR CHOOSING RASPBERRY PI BOARD AS THE MAIN PROCESSING BOARD:

- 1 The Raspberry Pi consumes very little fuel. It can be charged by attaching it to a 5V power bank.
- 2 The price is poor in comparison to other production boards.
- 3 The Raspbian OS, which is open source, runs the Raspberry Pi. It has a Micro SD slot for saving data and installing operating systems.
- 4 It has a 40-pin extended GPIO to help with "real-world" projects.

### 3.1.6 Raspberry Pi and Video Broadcasting:

The Raspberry Pi camera module is a great accessory for the PI- its great quality, can capture still photos and record video in full HD (1080p). The camera module is high spec and much better quality than a basic USB webcam. Its feature packed firmware fully utilizes the power of the Video Core GPU in the Raspberry Pi 4, allowing recording 1080p video at 30fps, 720p at 60fps, and VGA resolution (640x480) at 90fps—perfect for slow-motion playback. RPi is used in this project because it allows easy video transmission over Wi-Fi. 1.1.3

## 3.2 NODE\_MCU:

### 3.2.1 About Node\_MCU:

Node MCU is an open-source Lua based firmware and **development board** specially targeted for IoT based Applications. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module.



Fig 3.14 Node\_MCU ESP8266

The **Node MCU ESP8266 development board** comes with the ESP-12E module containing ESP8266 chip having Tensilica Xtensa 32-bit LX106 RISC microprocessor. This microprocessor supports RTOS and operates at 80MHz to 160 MHz adjustable clock frequency.

NodeMCU has 128 KB RAM and 4MB of Flash memory to store data and programs. Its high processing power with in-built Wi-Fi / Bluetooth and Deep Sleep Operating features make it ideal for IoT projects. Node MCU can be powered using Micro USB jack and VIN pin (External Supply Pin). It supports UART, SPI, and I2C interface.

In fig.4 The ESP8266 is the low cost WiFi microchip this small module allows a microcontroller to connect to the WiFi network. Node\_MCU is an open source firmware for which we can program it using Arduino IDE so we have uploaded a program for motion control of our robot. Node\_MCU can be powered using Micro USB jack and VIN pin External Supply Pin. It supports UART, SPI, and I2C interface.

Node\_MCU Development Board can be programmed with an Arduino IDE since it is easy to use. . Programming Node MCU with the Arduino IDE is much easier. All you need is the Arduino IDE, a USB cable and the Node MCU board.

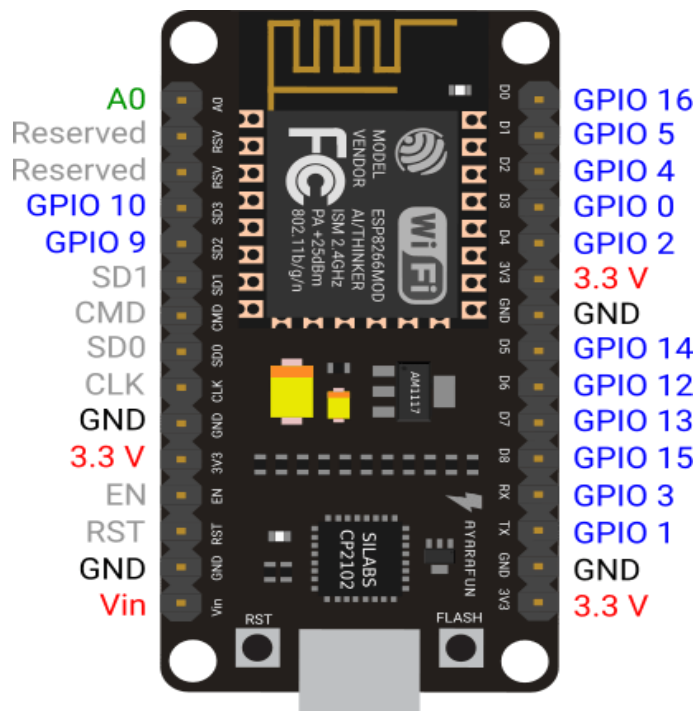


Fig 3.15 Node\_MCU Pin diagram



### 3.2.2 NodeMCU ESP8266 Specifications & Features:

1. Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
2. Operating Voltage: 3.3V
3. Input Voltage: 7-12V
4. Digital I/O Pins (DIO): 16
5. Analog Input Pins (ADC): 1
6. UARTs: 1
7. SPIs: 1
8. I2Cs: 1
9. Flash Memory: 4 MB
10. SRAM: 64 KB
11. Clock Speed: 80 MHz
12. USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
13. PCB Antenna
14. Small Sized module to fit smartly inside your IoT projects.

### 3.3 Power supply:



Fig 3.16 Power bank

The first stage of the circuit is a 12V charger for charging rechargeable batteries. Four lead-acid rechargeable batteries, each of 4V, 1Ah, are arranged in series and parallel combination to provide 8V power supply. If the batteries are connected in series, the final voltage gets added up and current remains the same. If the batteries are connected in parallel, current adds up and voltage remains the same. Thus, you get 8V, 2Ah power supply by connecting two each in series and then the two sets in parallel. Diode D1 is used to block discharging of the battery through LED1 and capacitor

C1. The diode also makes the circuit stable and controls temperature while charging. The 8V power supply is used to power the Node MCU, motor driver IC and other modules.

### 3.4 Motor Driver :

#### 3.4.1 About L293D:

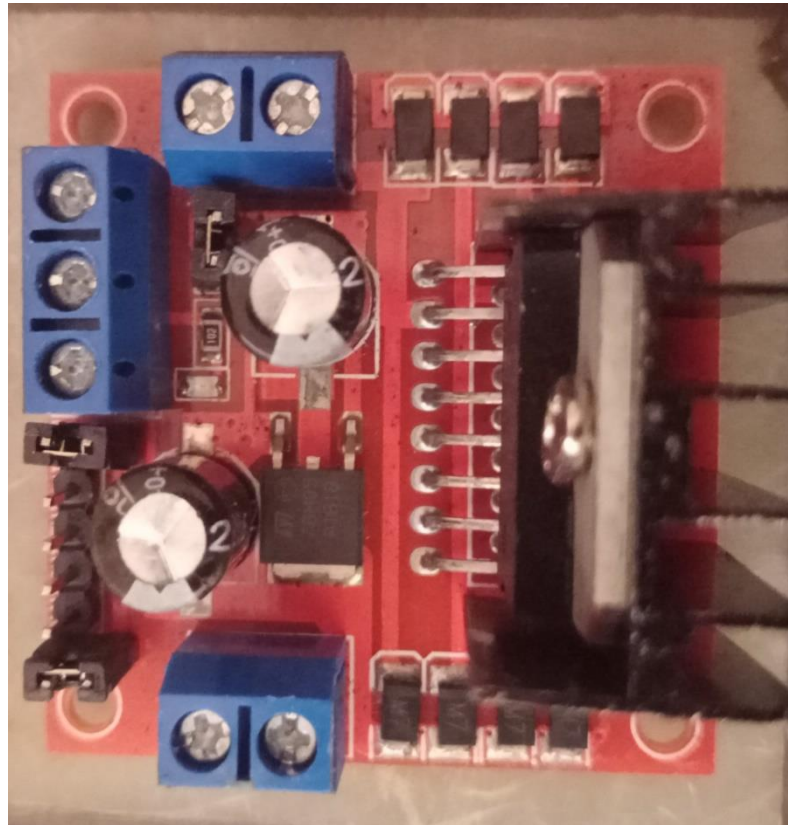


Fig 3.17 MOTOR DRIVER

The L293D is a popular 16-Pin **Motor Driver IC**. As the name suggests it is mainly used to drive motors. A single **L293D IC** is capable of running two DC Motors at the same time; also has operating voltage less than 36V and operating current less than 600mA, which are to be controlled by digital circuits like Op-Amp, 555 Timers, digital gates or even Microcontrollers like Arduino, PIC, ARM etc..

**L293D** is designed to drive a wide array of inductive loads such as relays, solenoids, DC and bipolar stepping motors, as well as other high-current and high-voltage loads. All inputs are TTL compatible and tolerant up to 7 V. Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an

enable input is high, the associated drivers are enabled, and their outputs are active and in phase with their inputs. When the enable input is low, those drivers are disabled, and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications.

The external high-speed output clamp diodes of the L293D are integrated to reduce system complexity and overall system size. A VCC1 terminal, separate from VCC2, is provided for the logic inputs to minimize device power dissipation. The L293D is characterized for operation from 0°C to 70°C.



Fig 3.18 Motor Driver(L293D) Pin Diagram

There are 4 input pins for direction control in L293d. Pin 2,7 (1A and 2A) on the left side and pin 15,10 (3A and 4A) on the right of the IC. The left side input pins regulate the rotation of the motor connected across the left end and the right-side input pins regulate the motor on the right-side. The motors are rotated based on the inputs provided across the input pins as HIGH or LOW signals.

L293D motor driver L293D is a typical motor driver IC that allows DC motor to drive in either direction. L293D is a 16-pin IC that can control a set of two DC motors simultaneously in any direction. It means you can control two DC motors with a single L293D IC

### 3.4.2 L293D Features:

1. Wide Supply-Voltage Range: 4.5 V to 36 V
2. Separate Input-Logic Supply
3. Internal ESD Protection
4. High-Noise-Immunity Inputs
5. Output Current 1 A Per Channel (600 mA for L293D)
6. Peak Output Current 2 A Per Channel (1.2 A for L293D)
7. Output Clamp Diodes for Inductive Transient Suppression (L293D)

### 3.5 Geared Motor:



Fig 3.19 Geared motor

A geared motor is a component whose mechanism adjusts the speed of the motor, leading them to operate at a certain speed. geared motor have the ability to deliver high torque at low speeds, as the gearhead functions as a torque multiplier and can allow small motors to generate higher speeds.

A geared motor can also be defined as a gear reducer because essentially, it is a combination of a speed reducer with a motor typically functioning as a gearbox, to reduce speed making more torque available.

A gearmotor delivers high torque at low horsepower or low speed. The speed specifications for these motors are normal speed and stall-speed torque. These motors use gears, typically assembled as a gearbox, to reduce speed, which makes more torque available. Gearmotors are most often used in applications that need a lot of force to move heavy objects. By and large, most industrial gearmotors use ac motors, typically fixed-speed motors. However, dc motors can also be used as gearmotors ... a lot of which are used in automotive applications.

Gearmotors have a number of advantages over other types of motor/gear combinations. Perhaps most importantly, gearmotors can simplify design and implementation by eliminating the step of separately designing and integrating the motors with the gears, thus reducing engineering costs.

Geared motors are a specific type of electrical motors that produce a high torque while maintaining a low- horsepower or low-speed motor output. These can be either AC (alternating current) or DC (direct current). They also have two different speed specifications normal speed and stall-speed torque. DC geared motors are primarily used to reduce speed in a series of gears, which, in turn, creates more torque. This is accomplished by series of gears or a gear box attached to the main motor rotor and shaft with the help of a second reduction shaft. The second shaft is then connected to the series of gears or gearbox to create what is known as a series of reduction gears. Two DC geared motors are used in this project.

### 3.6 Virtual Reality Headset:



Fig 3.20 VR headset

A **virtual reality headset** is a head-mounted device that provides virtual reality for the wearer. Virtual reality (VR) headsets are widely used with video games but they are also used in other applications, including simulators and trainers. They comprise a stereoscopic head-mounted display (providing separate images for each eye), stereo sound, and head motion tracking sensors<sup>1</sup> (which may include gyroscopes, accelerometers, magnetometers, structured light systems etc). Some VR headsets also have eye tracking sensors and gaming controllers.

The virtual reality headset allows the military personnel to interact with virtual reality people to make it feel real. They can talk to one another and do varying actions to make the virtual reality world feel like they are actually in that situation.

### **3.7 Servo motor:**

#### **3.7.1 About Servo motor:**

Servo motor is a rotatory actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with the servo motors. It is not a specific class of motor, although the term servomotor is often used to refer to a motor suitable for use in a closed loop control system. Two servo motors are used in this project.

Servo motor is a rotatory actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with the servo motors. It is not a specific class of motor, although the term servomotor is often used to refer to a motor suitable for use in a closed loop control system. Two servo motors are used in this project. 3.21 Camera module.

A camera module is an image sensor integrated with a lens, control electronics and an interface like CSI, Ethernet or plain raw low-voltage differential signalling. The Raspberry Pi camera module can be used to take high-definition videos as well as still photographs. The module has a five-megapixel fixed-focus camera that supports 1080p30, 720p60 and VGA90 video modes as well as stills capture. It attaches via a 15cm ribbon cable to the CSI port on the Raspberry Pi. It can be accessed through the MMAL and V4L APIs, and there are numerous third party libraries built for it, including the Pi camera Python library.



Fig 3.21 Servo motor

#### **3.7.2 Control of camera using servo motors:**

A servo motor is a motor with a built-in “servomechanism”. The servomechanism uses a sensor to monitor the motor shaft position and a controller to control the motor. It is fed a signal that indicates the position that the shaft should be set to. It then moves the motor into the required position. In the analog servo motors we will be working with the control signal which is a PWM signal whose pulse width determines the angle the motor shaft is to be positioned at. The motor itself is a simple DC motor with a lot of gearing to slow down its speed and to increase its torque. In order to function properly the servo motor needs a sensor that can accurately measure its shaft position. On some industrial and high-end hobby servos this is done using an optical interrupter disc, but in most standard hobby servo motors the sensor is a potentiometer. This works well as these servos typically travel 180 to 270 degrees, well within the range of a potentiometer.

In the robot, 2 servo motors are used to control the movement of the camera in both the axes namely horizontal and vertical. One servo controls the up and down motion while the other helps in sidewise motion. When both are operated simultaneously, we can control the movement of the camera in space. So basically the data from accelerometer and magnetometer sensors inside the mobile phone kept on the VR headset is sent to raspberry pi. This data is then fed to servo motors which then orient the camera based on the code written.

### 3.7.3 Flowchart for servo initializing:

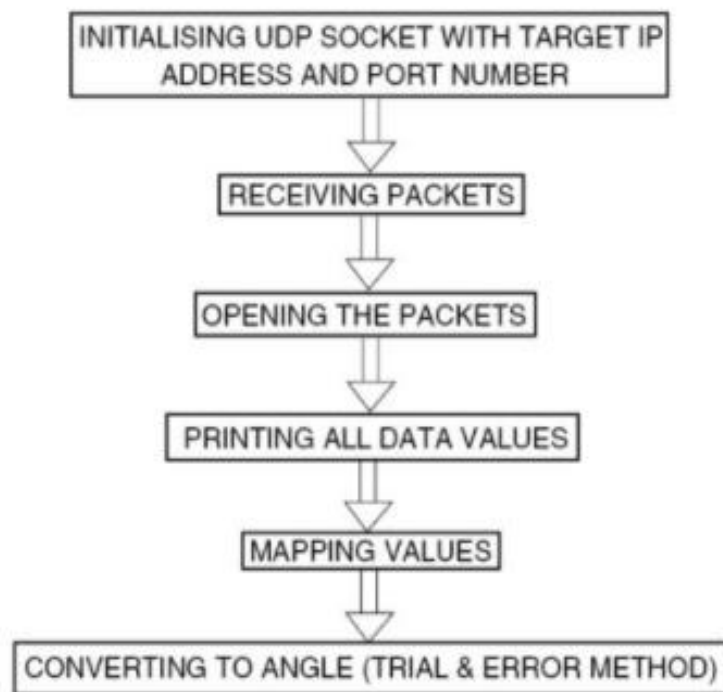


Fig 3.22 Flowchart for servo motor initializing

### **3.7.4 SERVO MECHANISM :**

Capturing live video, Servo motors are connected to RPi. In order to control the position of the camera, two servo motors are used to move the camera in X and Y directions. Download the Wireless IMU app from Play Store and install in your smartphone. The app supports accelerometer, gyroscope and magnetometer, and can be made to run in the background. The values from this app are sent to the RPi through UDP protocol. The magnetometer values make the servos move in either left or right direction. The accelerometer values make the servos move up and down. Thus, the camera can capture live video at different angles.



## CHAPTER 4

### SOFTWARES USED

#### 4.1 PYTHON:

Python is the best programming language to use in this project to communicate with the user on the Raspberry Pi. Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL).

Python is a widely used high-level programming language for general-purpose programming. Python programs don't need to be compiled before running them. However, the Python interpreter must be installed on the computer to run them. The Python interpreter is a program that reads Python files and executes the code. The `python_camera.py` code is used in the RPi board to control the servo motors.

In this project we were using Thonny Python IDE

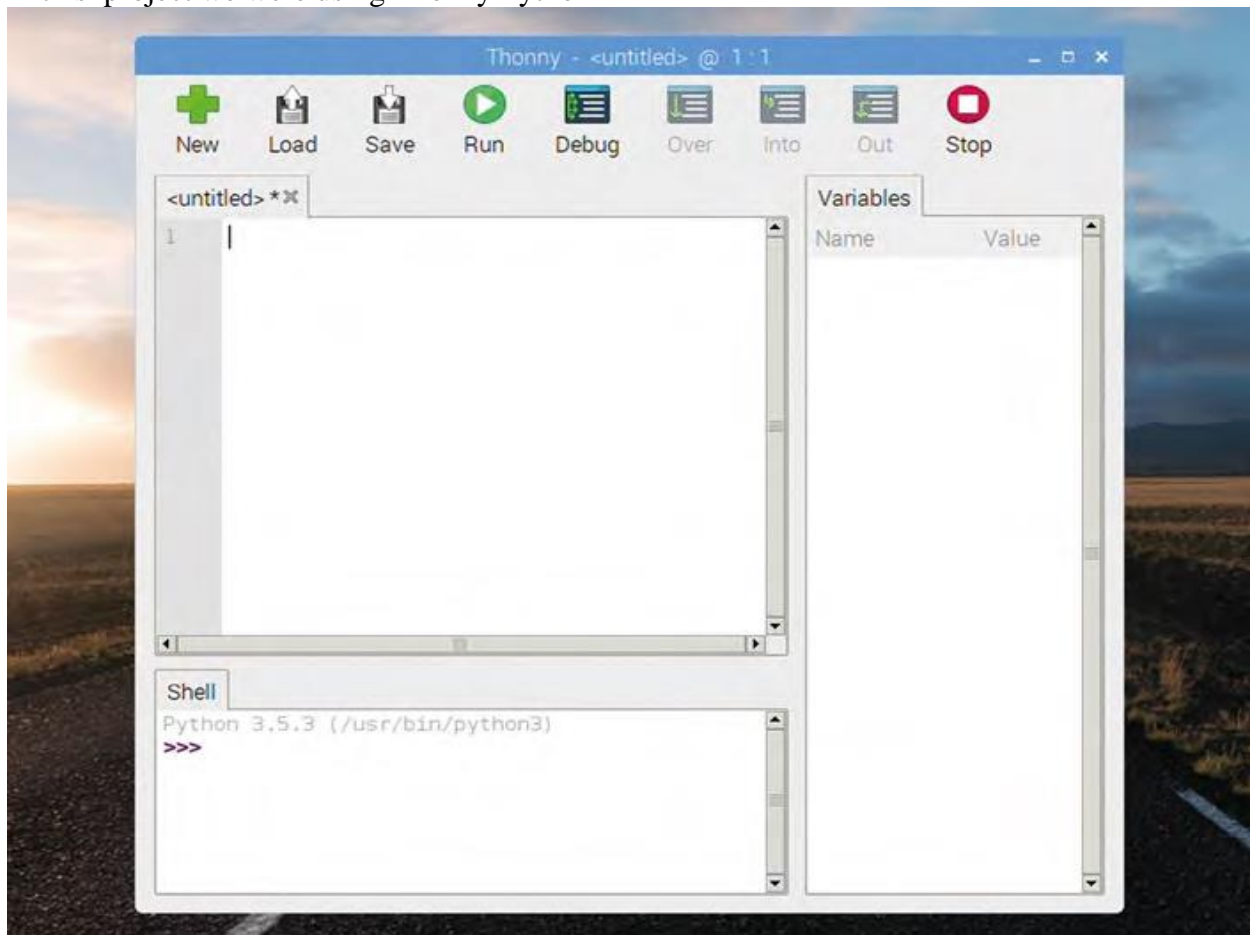


Fig 4.1 Thonny python IDE

## 4.2 RASPBIAN OS:

### 4.2.1 ABOUT RASPBIAN OS:

**Raspberry Pi OS** (formerly **Raspbian**) is a Debian-based operating system for Raspberry Pi. Since 2015, it has been officially provided by the Raspberry Pi Foundation as the primary operating system for the Raspberry Pi family of compact single-board computers. The first version of Raspbian was created by Mike Thompson and Peter Green as an independent project. The initial build was completed in June 2012.

Raspberry Pi OS is highly optimized for the Raspberry Pi line of compact single-board computers with ARM CPUs. It runs on every Raspberry Pi except the Pico microcontroller. Raspberry Pi OS uses a modified LXDE as its desktop environment with the Open box stacking window manager, along with a unique theme.

#### VERSIONS:

Raspberry Pi OS has 3 installation options:

- Raspberry Pi OS Lite
- Raspberry Pi OS
- Raspberry Pi OS Full

Raspberry Pi OS Lite is the smallest version and doesn't include a Desktop Environment. Raspberry Pi OS includes the Pixel Desktop Environment. Raspberry Pi OS Full also comes pre-installed with additional productivity software.

The initial version of the Raspbian OS was made by Mike Thompson and Peter Green as an entirely independent endeavour. The Operating System was rooted in Debian, which is a kind of Linux operating system. This puts the Raspbian OS firmly into the UNIX-Like family of operating systems.

The initial versions of the Raspbian operating system were 32-bit and Debian-based. However, more recent editions of the OS have switched to being 64-bit, and have abandoned the use of Debian as their base.

The Raspberry Pi OS was made specifically with the Raspberry Pi in mind, and it'll run on every single kind of Raspberry Pi board, apart from the Pico edition, due to its far smaller size and computing power. The Raspberry Pi OS uses a modified version of the Lightweight X11 Desktop Environment (or LXDE) as its desktop environment. LXDE is a desktop environment specifically made for single-chip computers and those with low resources.

It uses an Openbox stacking window manager together with its own unique theme to bring a cohesive and unique user experience. Each distribution of the OS will come with an algebra

program called Wolfram Mathematica, as well as the Minecraft: Pi Edition game, in addition to a hyper-lightweight version of Chromium.

#### 4.2.2 INSTALLING RASPBIAN OS IN RASPBERRY PI:

##### Step 1 : Download the Required Software and Files

You need to download 2 software and 1 OS i.e. Raspbian for this complete process.

**1st software:** The first software is Win32 Disk Imager.

<https://sourceforge.net/projects/win32diskimager/>

**2nd software:** Second software is SD Card Formatter.

[https://www.sdcard.org/downloads/formatter\\_4/](https://www.sdcard.org/downloads/formatter_4/)

**Raspbian OS:** This is the Main operating system of the Pi.

<https://www.raspberrypi.org/downloads/raspbian/>

Extract all files to the desktop.

##### Step 2 : Get the SD Card and the Card Reader

Get a minimum **8GB class 10** SD card with a card reader. Insert that card into the card reader and plug that to the USB port.

##### Step 3: Check the Drive in Which the SD Card Is Mounted

Go to my computer or **My PC** and find the drive name where the SD card is **mounted**.

##### Step 4: Format the SD Card

Open SD Card Formatter and select the drive you noticed in the previous step.

Click on format and **don't alter any other options**.

When formatting is completed, click on OK.

##### Step 5: Write the OS on the SD Card

Open **win32diskimager**.

Browse the .img file of Raspbian OS that was extracted from the downloaded file.

Click on open and then click on Write. **If any warning pops up then ignore those by clicking OK.**

Wait for the write to be completed and it may take some minutes.

**Step 6:** Eject the SD Card

**Step7:** Keep the SD card into the SD card connector in the Raspberry Pi.

### 4.2.3 Setting up the software:

Before you can begin using the Raspberry Pi in earnest, you'll need to set up its software – in particular its operating system, the software which controls what the Pi can do. NOOBS, the New Out-Of-Box Software, is designed to make this as easy as possible, allowing you to choose from several different operating systems and install them automatically. Better still, you can do all this in little more than a few clicks of the mouse.

When the Pi is first switched on, or booted, with a fresh installation of NOOBS on its microSD card, you'll see a screen with the Raspberry Pi logo on it and a small progress window at the upper-left. After a short pause, which can take up to a minute the first time you use the NOOBS microSD card, you'll see the screen shown in **Fig 4.2**



Fig 4.2 The NOOBS menu without any operating systems installed

This is the NOOBS menu, a system which lets you choose the operating system to run on your Pi. Two operating systems are included with NOOBS as standard: Raspbian, a version of the Debian Linux operating system tailored specifically for the Raspberry Pi; and LibreELEC, a version of the Kodi Entertainment Centre software. If the Pi is connected to the network – either through a wired connection or using the ‘Wifi networks (w)’ option from the top bar of icons – you can also download and install other operating systems.

To begin installing an operating system, use the mouse to put a cross in the box to the left of Raspbian Full: point the cursor at the white box and click once with the left mouse button. When you’ve done so, you’ll see that the ‘Install (i)’ menu icon is no longer greyed-out; this lets you know that your operating system is ready to install



Fig 4.3 The NOOBS menu with Raspbian operating systems installed

Click the ‘Install (i)’ icon once with the left mouse button and you’ll see a warning message telling you that installing the operating system will overwrite any data currently stored on the microSD card – not counting NOOBS itself, which stays intact. Click ‘Yes’ and the installation process will begin (**Figure 4.4**).

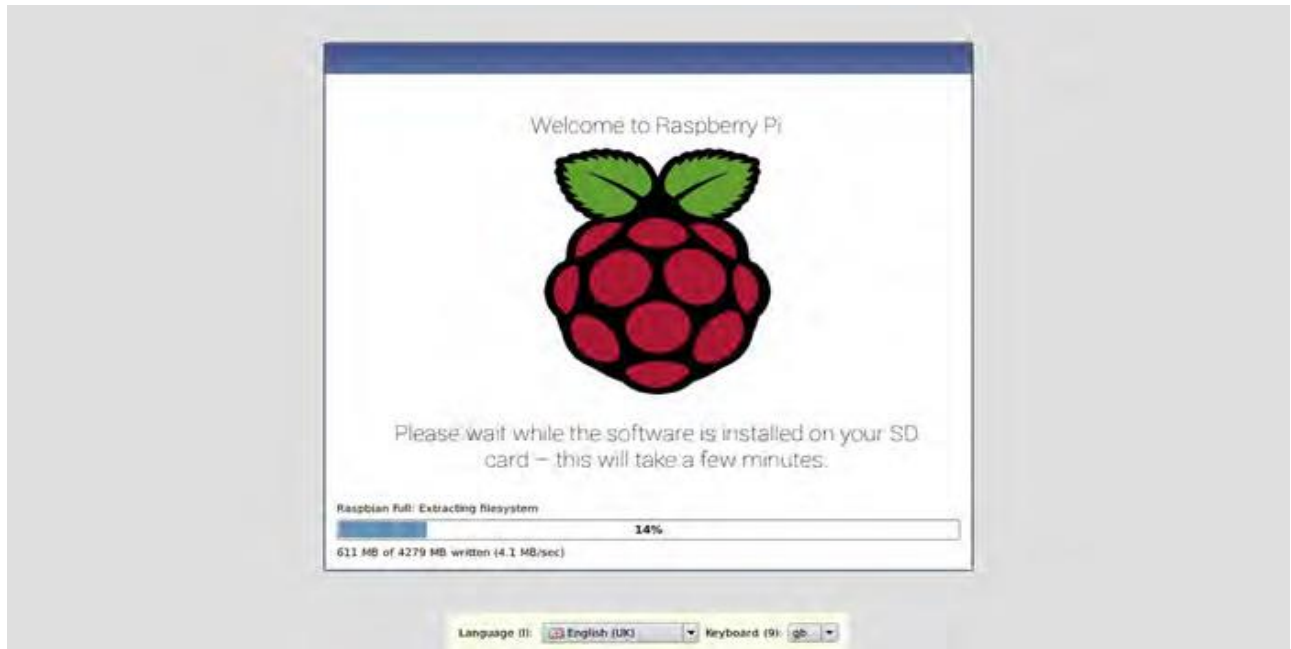


Fig 4.4 Installing Raspbian OS

After installing Finally, you'll briefly see a window with the Raspberry Pi logo on it before the Raspbian desktop and setup wizard appears, as in **Figure 4.5**. Your operating system is now fully installed and ready to be configured,  
Now your OS is installed on your Raspberry Pi.

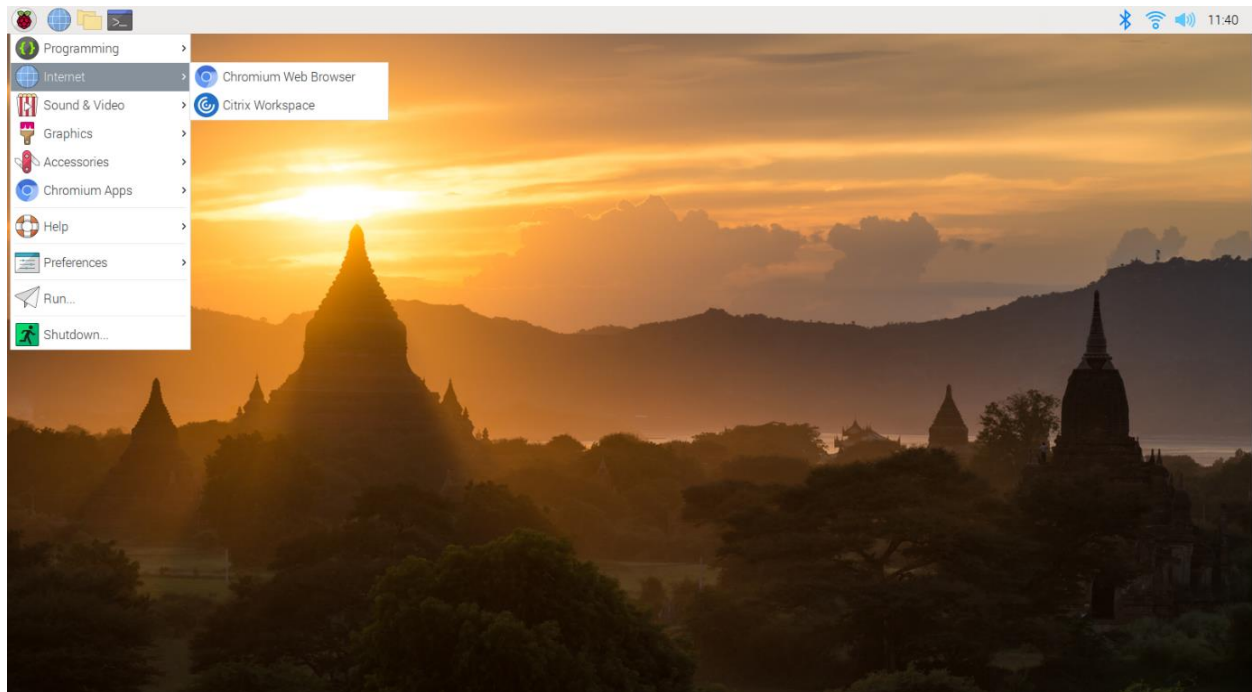


Fig 4.5 Raspbian Desktop

## 4.3 Arduino sketch for Node\_MCU:

### 4.3.1 About Arduino IDE:

**Arduino IDE** is software for programming Arduino Board / Node\_MCU. This software is used as a **text editor** to create, open, edit, and validate Code for Arduino. The Code or Program in Arduino is called "**sketch**".

### 4.3.2 INSTALLING ARDUINO IDE IN WINDOWS 10:

#### Step 1: Download File Arduino IDE

Click the link below to download the Software :<https://www.arduino.cc/en/software>

On that page, there are 3 download options for Windows.

1. **Windows Installer** : The software will be installed in Windows operating system and required admin access.
2. **Windows Zip file** : To make a portable installation.
3. **Windows App** : for Windows 8.1 or 10.

#### Step 2: Installation Option

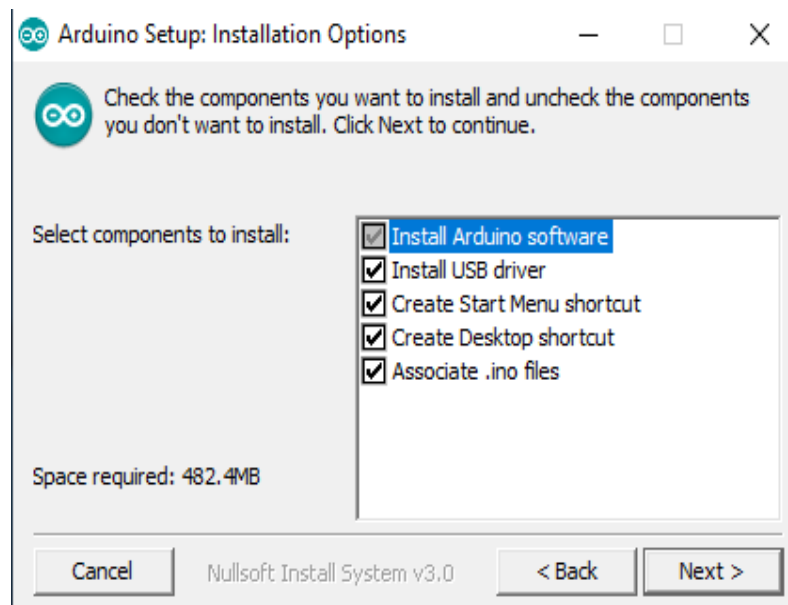


Fig 4.6 Arduino Installation option

### Step 3: Installation Folder

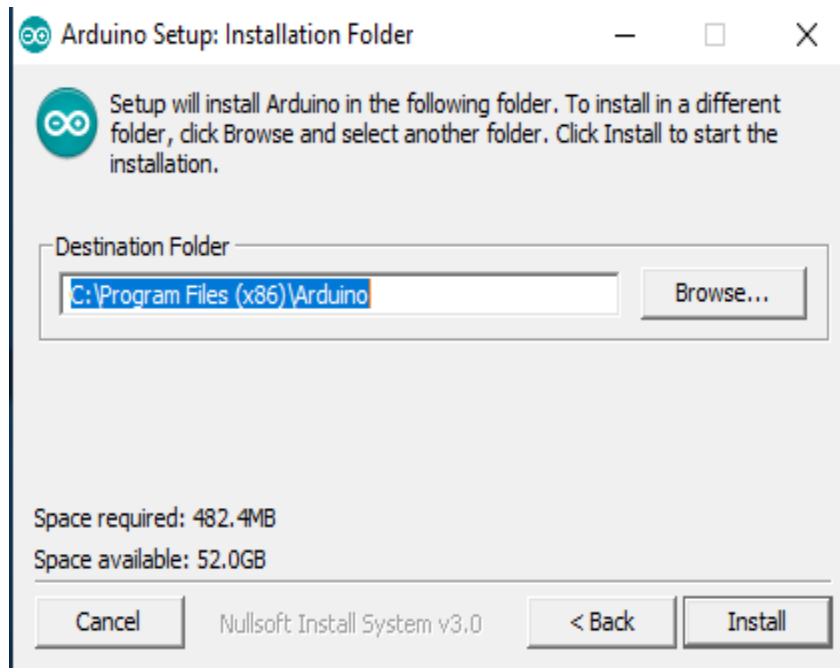


Fig 4.7 Arduino Installation Folder

1. Arduino will automatically be installed in "**C:\Program Files (x86)\Arduino**".
2. If you want to change the folder, click "*Browse*" and select the desired folder.
3. Click install to start the installation.

### Step 4: Installing Process



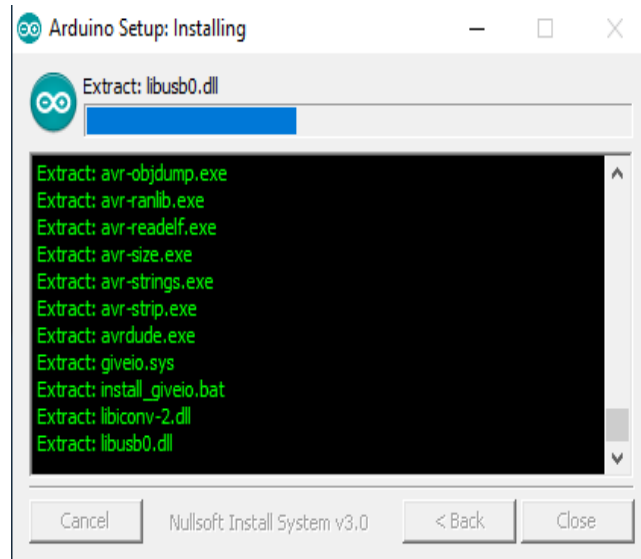


Fig 4.8 Arduino Installation process window

#### Step 5: Installation Complete

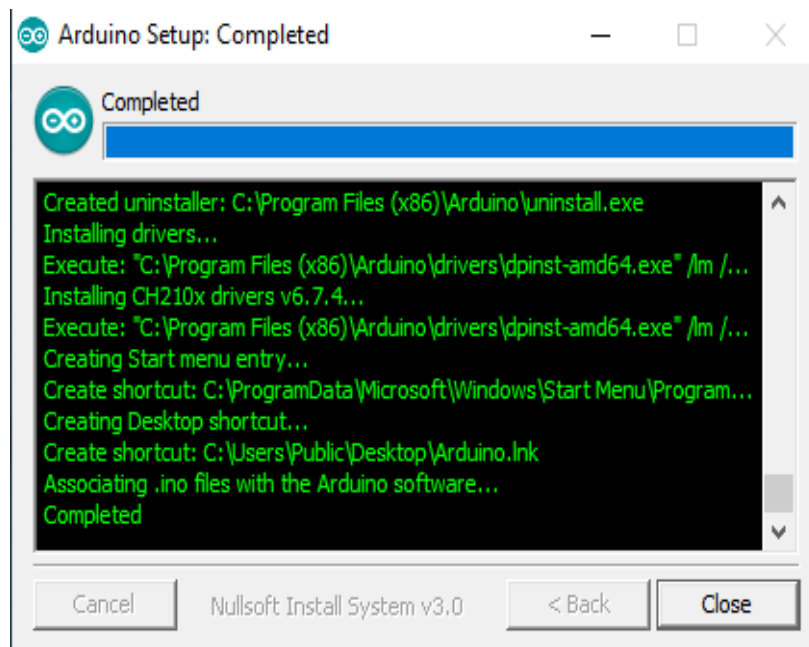


Fig 4.9 Arduino Installation complete window

If there is written "*complete*", it means that the installation process is complete. click "*Close*".

#### Step 6: Open Arduino IDE

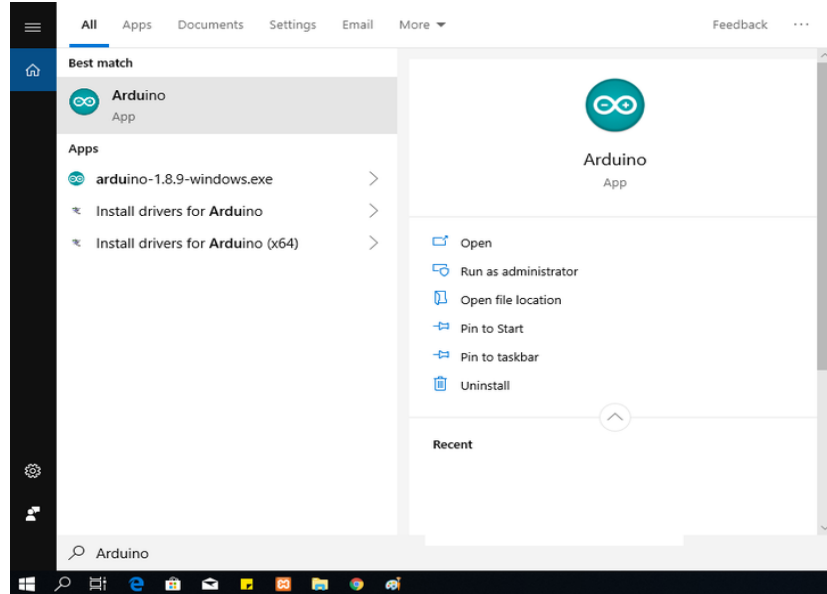


Fig 4.10 Arduino Icon

After the installation process is complete, there will be an Arduino icon on the Desktop. Or check on the search icon and write "arduino". If you have found the arduino icon, run the application.

### Step 7: Display Arduino IDE

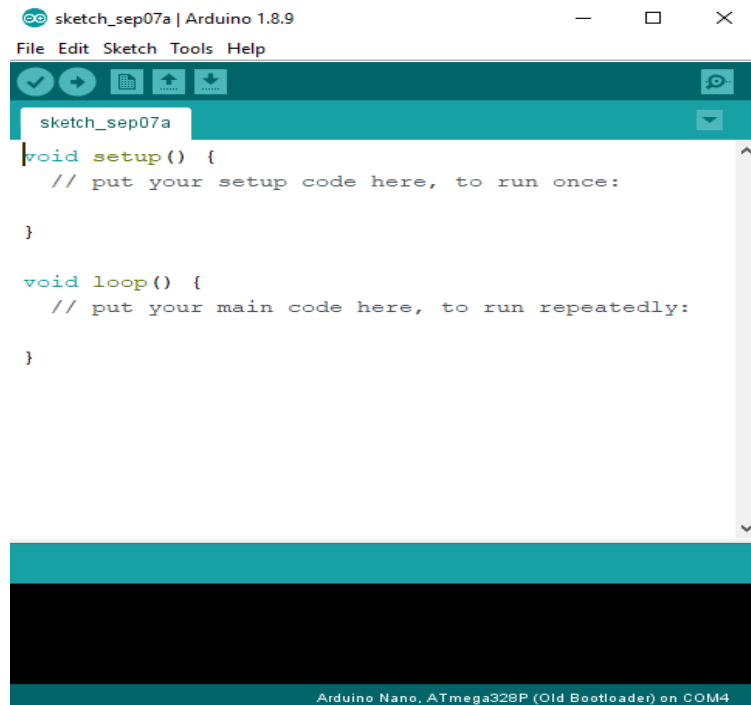


Fig 4.11 Arduino IDE software

This is a display of the Arduino IDE Software.

### 4.3.3 How to program NodeMCU using Arduino IDE

In order to use Arduino IDE to program the NodeMCU, you have to introduce it to the software at first.

To do this copy the following code and follow the steps below:

[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)

**Step1.** Choose Preferences in the File menu and enter the copied code in Additional Board Manager URLs part. Then press OK.

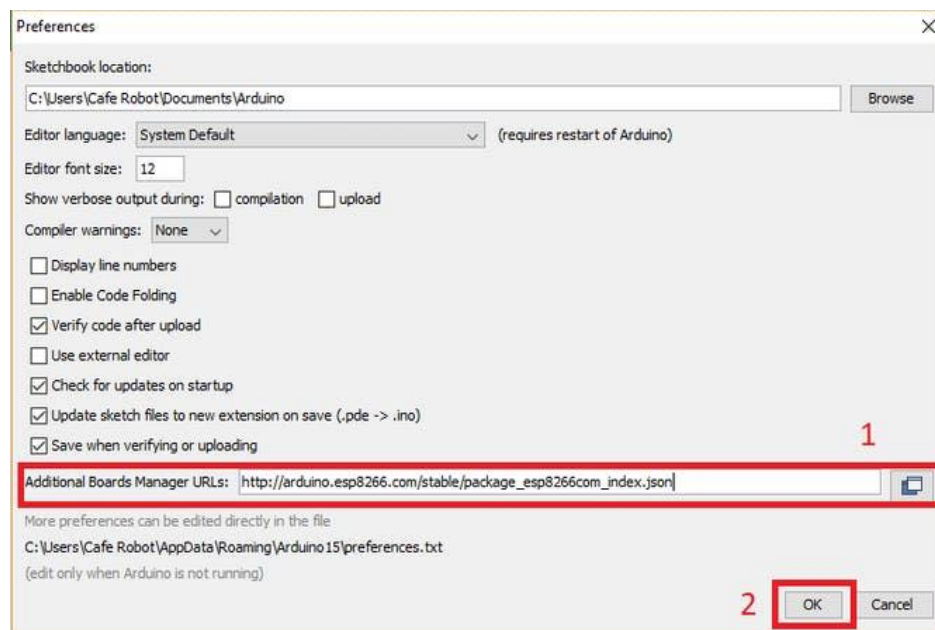
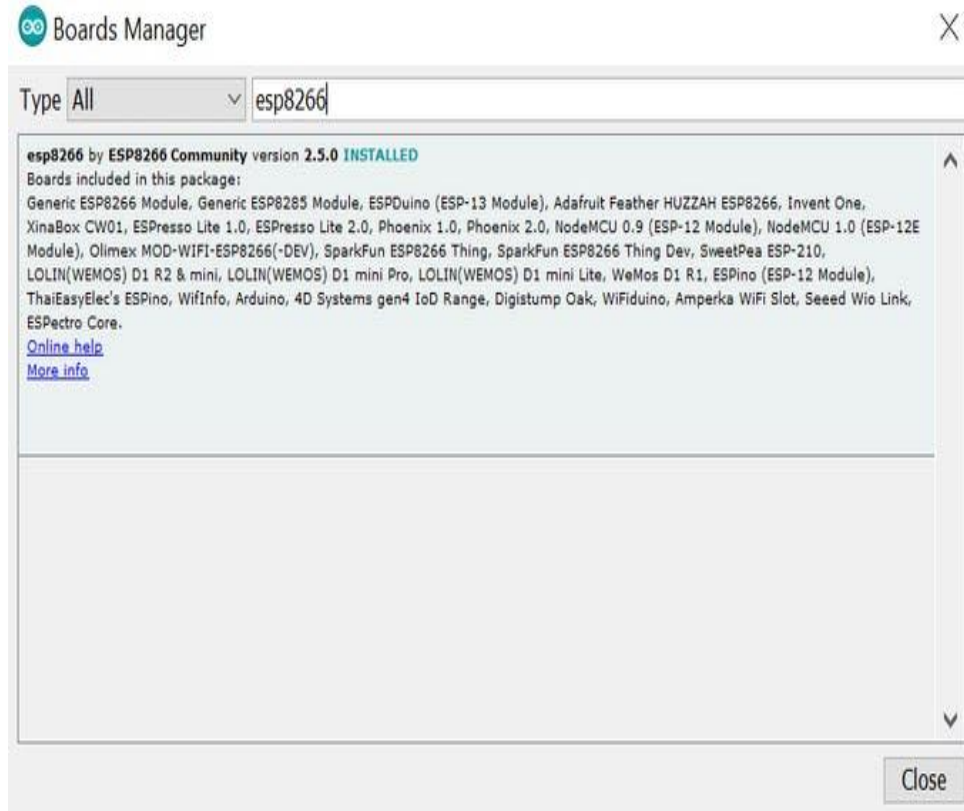


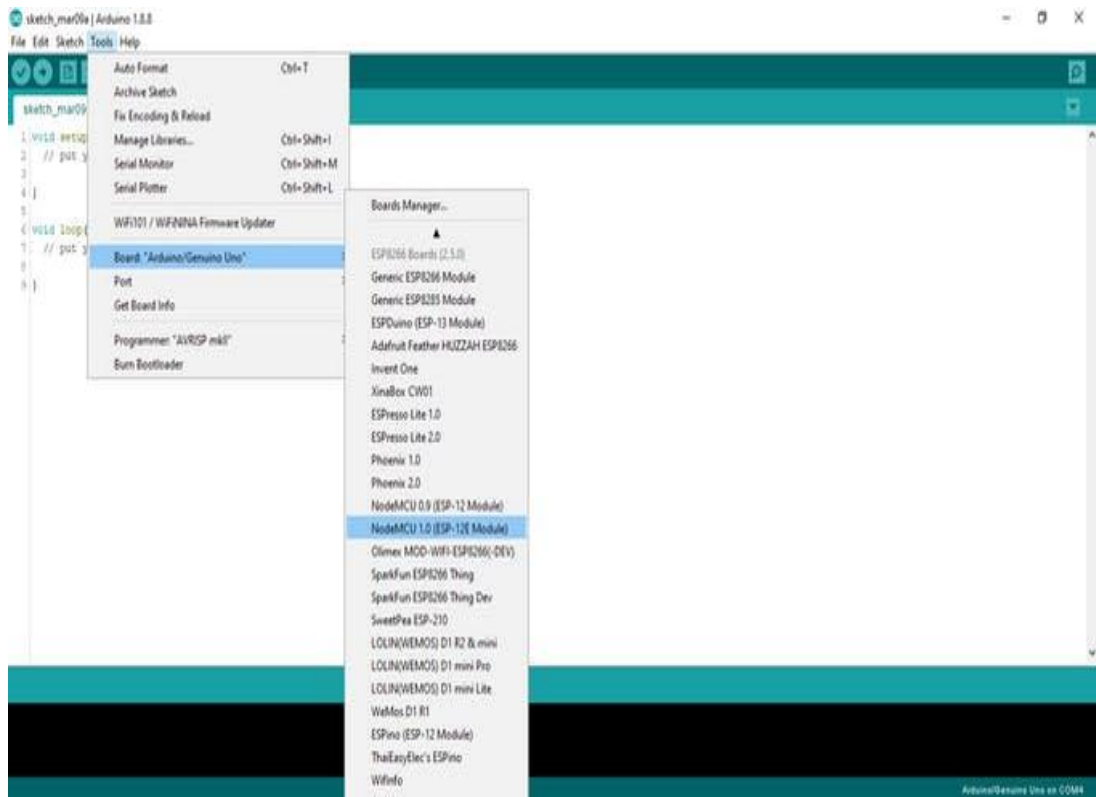
Fig 4.12 Choosing preferences for ESP8266

#### Step2:

1. Search the word ESP8266 in Boards>boards manager from Tools menu.
2. Then install ESP8266 boards.
3. After complete installation, you will see the INSTALLED label on ESP8266 boards.



After these two steps, you can see ESP8266 based boards such as NodeMCU in your Arduino IDE boards list, and you can choose your desired board to upload the code. In order to use digital pins, you should select GPIO numbers



#### 4.4 Application required for movement of camera :

This app measures and reports a body's specific force, angular rate and sometimes the magnetic field surrounding the body, using a combination of accelerometers and gyroscopes, and sometimes magnetometers. An inertial measurement unit (IMU) works by detecting the current rate of acceleration using one or more accelerometers, and detects changes in rotational attributes like pitch, roll and yaw using one or more gyroscopes. And some also include a magnetometer, mostly to assist calibration against orientation drift.

The accelerometers are placed such that their measuring axes are orthogonal to each other. Three gyroscopes are placed in a similar orthogonal pattern, measuring rotational position in reference to an arbitrarily chosen coordinate system. Magnetometers allow better performance for dynamic orientation calculation in attitude and heading reference systems.

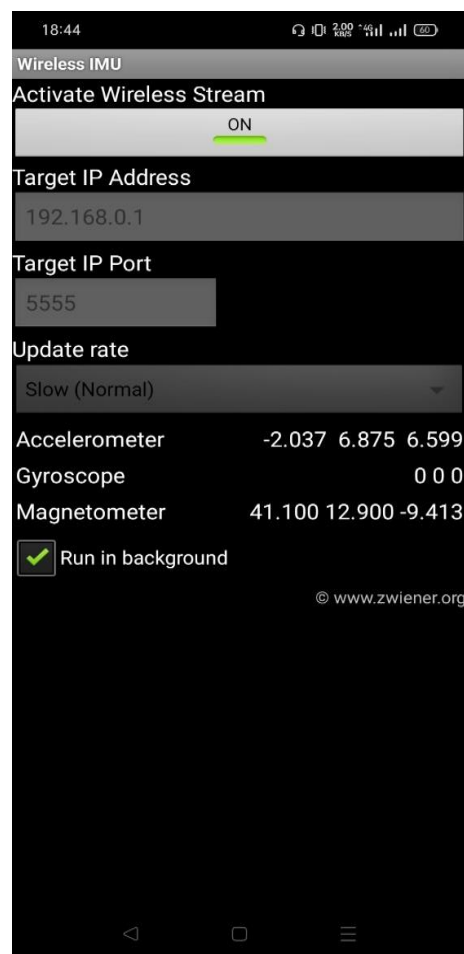


Fig 4.13 Wireless IMU Apps



#### 4.5 Application required for movement of Robot:

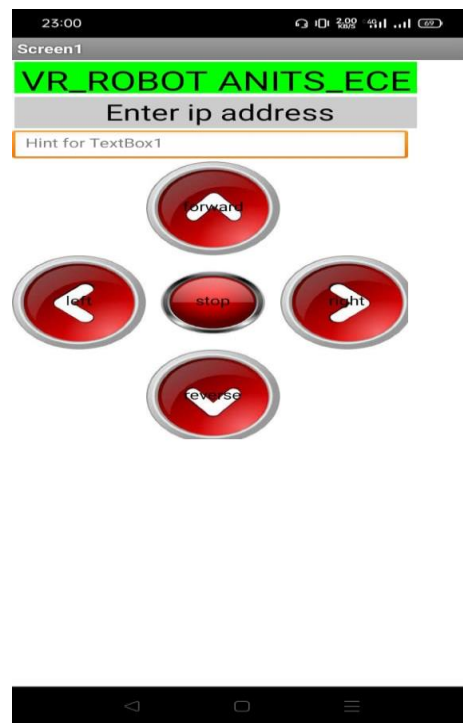


Fig 4.14 App for movement of robot

An application was designed in such a way that for movement of robot to forward, backward , left, right using those buttons in application we have to enter the ip address of ESP 8266 so that there establishes a connection between application and node MCU

## CHAPTER 5

### LIVE STREAMING

#### 5.1 Streaming :

Streaming is a technique in which the server provides the response to a request in chunks. I can think of a couple of reasons why this might be useful:

- **Very large responses.** Having to assemble a response in memory only to return it to the client can be inefficient for very large responses. An alternative would be to write the response to disk and then return the file with `flask.send_file()`, but that adds I/O to the mix. Providing the response in small portions is a much better solution, assuming the data can be generated in chunks.
- **Real time data.** For some applications a request may need to return data that comes from a real time source. A pretty good example of this is a real time video or audio feed. A lot of security cameras use this technique to stream video to web browsers.

#### 5.2 Implementing Streaming With Flask:

Flask provides native support for streaming responses through the use of generator functions. A generator is a special function that can be interrupted and resumed. A generator function can return multiple results in sequence. Flask uses this characteristic of generator functions to implement streaming. A route that returns a streamed response needs to return a `Response` object that is initialized with the generator function. Flask then takes care of invoking the generator and sending all the partial results as chunks to the client. In this way the flask works with the generator functions.

#### 5.3 Building a Live Video Streaming Server

There are many ways to stream video to browsers, and each method has its benefits and disadvantages. The method that works well with the streaming feature of Flask is to stream a sequence of independent JPEG pictures. This is called [Motion JPEG](#), and is used by many IP security cameras. This method has low latency, but quality is not the best, since JPEG compression is not very efficient for motion video.

This application imports a `Camera` class that is in charge of providing the sequence of frames. Putting the camera control portion in a separate module is a good idea in this case, this way the web application remains clean, simple and generic.

The application has two routes. The `/` route serves the main page, which is defined in the `index.html` template. Below you can see the contents of this template file:

```
<html>

<head>

  <title>Video Streaming Demonstration</title>

</head>

<body>

  <h1>Video Streaming Demonstration</h1>

</body>

</html>
```

This is a simple HTML page with just a heading and an image tag. Note that the image tag's `src` attribute points to the second route of this application, and this is where the magic happens.

The `/video_feed` route returns the streaming response. Because this stream returns the images that are to be displayed in the web page, the URL to this route is in the `src` attribute of the image tag. The browser will automatically keep the image element updated by displaying the stream of JPEG images in it, since multipart responses are supported in most/all browsers (let me know if you find a browser that doesn't like this).

The generator function used in the `/video_feed` route is called `gen()`, and takes as an argument an instance of the `Camera` class. The `mimetype` argument is set as shown above, with the `multipart/x-mixed-replace` content type and a boundary set to the string `"frame"`.

The `gen()` function enters a loop where it continuously returns frames from the camera as response chunks. The function asks the camera to provide a frame by calling the `camera.get_frame()` method, and then it yields with this frame formatted as a response chunk with a content type of `image/jpeg`, as shown above.

## 5.4 Obtaining Frames from a Video Camera:

Now all that is left is to implement the `Camera` class, which will have to connect to the camera hardware and download live video frames from it. The nice thing about encapsulating the hardware dependent part of this application in a class is that this class can have different implementations

for different people, but the rest of the application remains the same. You can think of this class as a device driver, which provides a uniform implementation regardless of the actual hardware device in use.

The other advantage of having the `Camera` class separated from the rest of the application is that it is easy to fool the application into thinking there is a camera when in reality there is not, since the camera class can be implemented to emulate a camera without real hardware.

Once you have the application installed, create a virtual environment and install Flask in it. Then you can run the application as follows:

```
$ python app.py
```

After the application starts enter `http://localhost:5000` in the web browser and we can observe the emulated video stream

Once everything starts working , fire up the Raspberry Pi with its camera module and implemented a new `Camera` class that converts the Pi into a video streaming server, using the `picamera` package to control the hardware.



## 5.5 Limitations of Streaming

When the Flask application serves regular requests the request cycle is short. The web worker receives the request, invokes the handler function and finally returns the response. Once the response is sent back to the client the worker is free and ready to take on another request.

When a request that uses streaming is received, the worker remains attached to the client for the duration of the stream. When working with long, never ending streams such as a video stream from a camera, a worker will stay locked to the client until the client disconnects. This effectively

means that unless specific measures are taken, the application can only serve as many clients as there are web workers. When working with the Flask application in debug mode that means just one, so you will not be able to connect a second browser window to watch the stream from two places at the same time.

There are ways to overcome this important limitation. The best solution in my opinion is to use a coroutine based web server such as [gevent](#), which Flask fully supports. With the use of coroutines gevent is able to handle multiple clients on a single worker thread, as gevent modifies the Python I/O functions to issue context switches as necessary.

## **CHAPTER 6**

### **WORKING OF THE PROJECT**

#### **Navigation**

The navigation circuit controls movement of the robot. It consists of Node MCU, L293D motor driver (IC1), and two DC geared motors M3 and M4. The smartphone reads the accelerometer and magnetometer data of the direction in which the user turns his head, say, right or left. This data is sent to the modem over Wi-Fi and to the RPi board, which, in turn, provides these values as inputs to the servo motors. Two servo motors are used to move the camera—one for the vertical movement and the other for the horizontal movement. So when we turn our head along with VR headset to the right side, the RPi camera will also turn to the right direction..

#### **Virtual reality and Servo Control**

Virtual reality (VR) is a computer-generated environment that simulates several senses and lets the user interact with the content in a manner that is similar to a physical place. Virtual reality can be described as a technology that combines 3D graphics with immersive display and tracking technology, in order to create a system where the displayed content matches the user's viewpoint. However Riva means that this description is too focused on hardware and would rather describe it as the inclusive relationship between the virtual content and the user. According to Riva, VR can be considered as a form of computer-mediated-communication in a multi-user scenario. Riva also believes that VR has the possibility to become the next dominant medium, outperforming television and telephones.( Riva is a famous VR enthusiast).





## Dual screen:

The DualScreen app allows the screen of the smartphone to be shown in two different parts. This is done to get accurate visuals through the VR lens. There are two eyepieces in VR headset, hence dual-screen mode is used.

## Construction:

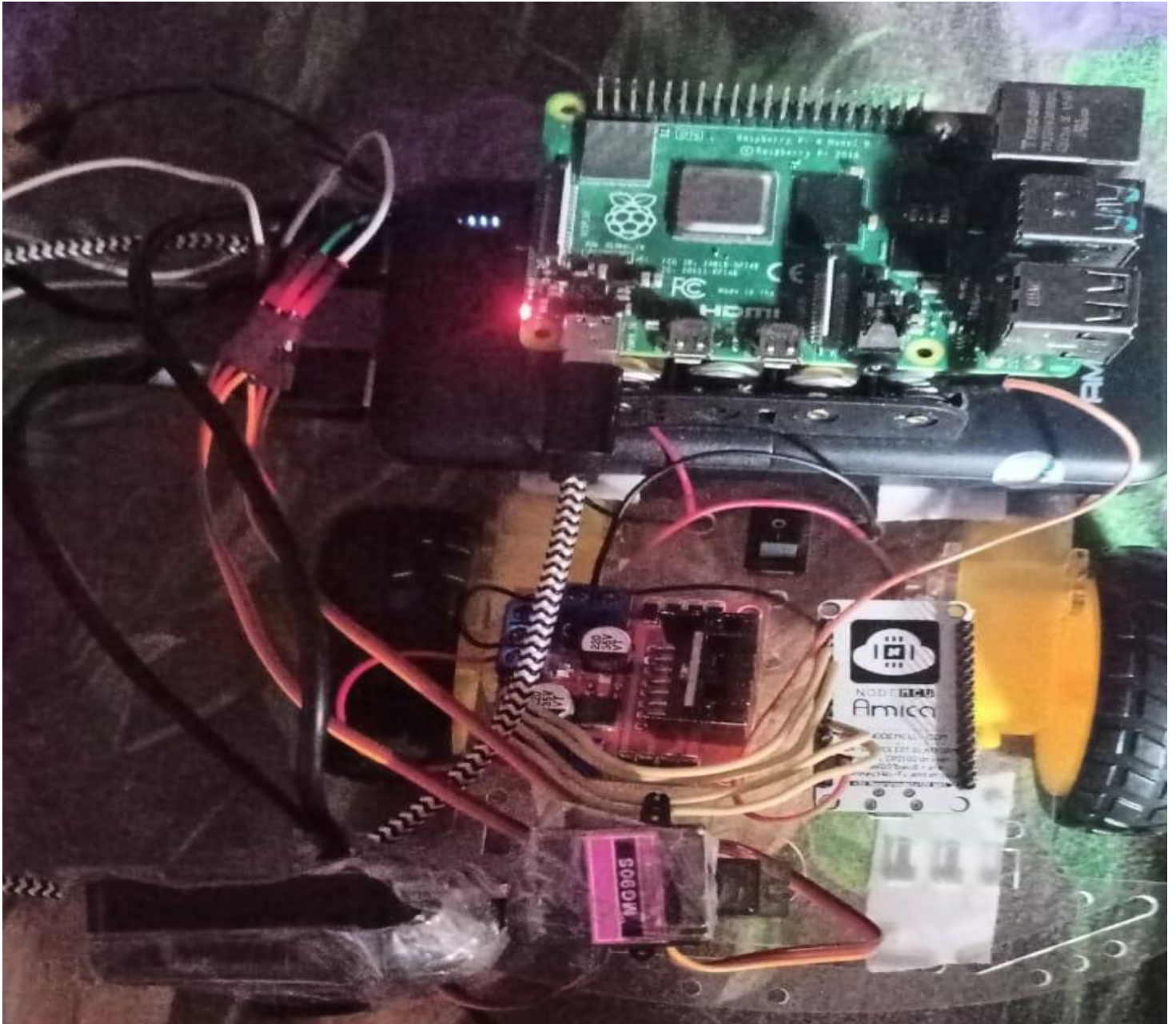
Use a suitable chassis for the virtual telepresence robot. Fix all the batteries underneath the chassis. Mount the PCB and boards on top of the chassis. Connect servo motors M1 and M2. Attach the RPi camera on these two motors for horizontal and vertical movements. Connect the two DC geared motors to the chassis of the virtual telepresence robot for moving forward, backward, left and right.

When hardware assembly is done, install all the required apps in your smartphone as mentioned earlier. Enable dual screen mode. Place your smartphone on the VR headset such as a VR Box. Now, put the VR headset on your head. Adjust the right and left lenses of the VR glasses to make sure that the image is being displayed properly. If everything is fine, turn your head left and right, the camera fitted on the virtual telepresence robot will also move in the same direction accordingly



## RESULTS AND DISCUSSIONS

### Final prototype:







## **CONCLUSIONS**

### **Limitations and solutions:**

1. In order for video transmission to happen a Wi-Fi connection is required. Over a LAN the transmission is almost instantaneous but it keeps varying over slower Wi-Fi. High speed internet connectivity can solve this problem.
2. The directional data are transferred as the head moves. In the case when the head moves very fast the data transfer rate is high. This in turn requires the Raspberry Pi to process the data very quickly. The app should be programmed in such a way that the Raspberry Pi board gets enough time to process the data and provide it as input to the servos.
3. The currently designed robot is suitable to run over smooth surfaces. Stronger body of the robot allows it to run over rougher terrains.

### **Applications**

1. This robot can be used as a surveillance robot. In situations where the house owner needs to keep an eye of what the servant is up to at home.
2. Other applications include: In the case of military, instead of sending a soldier to a sight for monitoring the robot can be sent. In this way, even if there is an unexpected attack no life is lost only the robot is under damage.
3. In the Medical case, at times when the doctor is not able to go on rounds, the robot can be used to check the state of the patient.
4. If the robot is made fire proof, then the robot can be used in fire and rescue operations.
5. On further expansion of the robot and increasing its connectivity, the robot can be used in space research.

## REFERENCES:

- [1] Kwon, O.H., Koo, S.Y., Kim, Y.G. and Kwon, D.S., 2010, October. Telepresence robot system for English tutoring. In 2010 IEEE workshop on advanced robotics and its social impacts (pp. 152-155). IEEE.
- [2] Documentation by <https://www.raspberrypi.org>
- [3] Stack overflow website for programming concepts
- [4] Github –Flask video streaming
- [5] Hossain, Mohammad Tanzir Kabir, “A Real-time Surveillance Nazmul Mini-rover Based on OpenCV-Python J AVA Using Raspberry Pi 2”, 2015 IEEE International Conference on Control System, Computing and Engineering, 27 - 29 November 2015, Penang, Malaysia

## PAPER PUBLICATIONS DETAILS:

### VIRTUAL TELEPRESENCE ROBOT

G Sanjay Bhargav<sup>1</sup>, K Vijaya Tej<sup>2</sup>, P Jaswanth<sup>3</sup>, M Naga Pujitha<sup>4</sup>, Mr. Nitin Kumar<sup>5</sup>

<sup>1</sup>Department of ECE, Anil Neerukonda Institute of Technology and Sciences, INDIA  
sanjaybhargav.2017.ece@gmail.com

<sup>2</sup>Department of ECE, Anil Neerukonda Institute of Technology and Sciences, INDIA  
vijayatej.2017.lece@anits.edu.in

<sup>3</sup>Department of ECE, Anil Neerukonda Institute of Technology and Sciences, INDIA  
nagapujitha.2017.ece@anits.edu.in

<sup>4</sup>Department of ECE, Anil Neerukonda Institute of Technology and Sciences, INDIA  
vijayatej.2017.lece@anits.edu.in

<sup>5</sup>Department of ECE, Anil Neerukonda Institute of Technology and Sciences, INDIA  
nitinkumar.ece@anits.edu.in

**Abstract** - The Telepresence robot is a remote-controlled, two-wheeled device with a screen to give video streaming it allows the person to view remote locations in virtual reality as the user feels that he was present at that place where the robot is situated. This project consists of a Virtual Reality headset, with a mobile in dual screen mode to come across virtual reality and a two-wheeled robot. The working of the Robot is control by a remote controlled application on the smartphone. The movement of the camera is attached to a robot is controlled by two servomotors one is to move left or right, another is to up or down, and Raspberry Pi. Video streaming is received by the smartphone using the IP address of the Raspberry Pi.

**Key Words:** Remote controlled robot, VR headset, Video streaming, Raspberry pi, Wireless application.

#### The objective of the project:

The main objective is to create a virtual reality robot that gives the user a real-time experience as if he is present where the robot is located.

## 1.METHODOLOGY

The below Figure depicts a block diagram of a virtual telepresence robot. The RPi serves as the system's central processing unit (CPU). It receives feedback from the smartphone via Wi-Fi and then sends a controlling pulse to the servo motors, causing the RPi camera to turn. The smartphone also sends data to the node MCU for the virtual telepresence robot to move front, back, left, right, and to stop. The motor driver IC L293D and geared motors are connected with node MCU for movement of the robot. The smartphone can send commands to the virtual telepresence robot via Bluetooth/wifi or an RF module. In our case, we use Wifi.

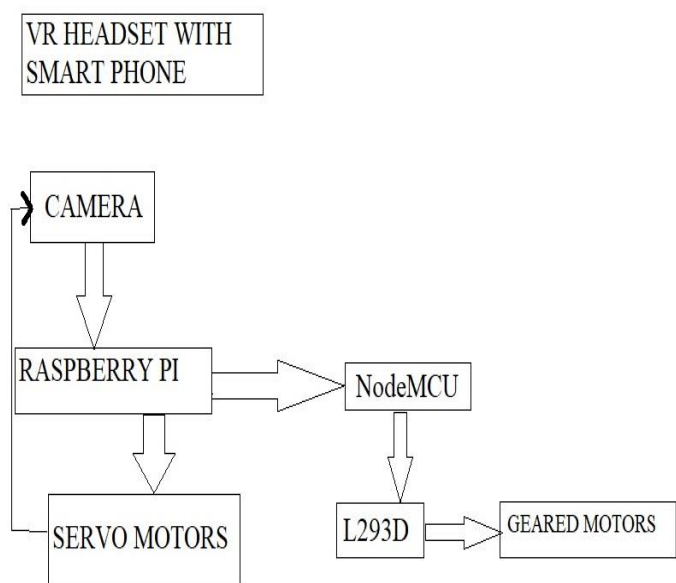


Fig-1: Block Diagram

## 2. COMPONENTS REQUIRED

**Hardware:** Raspberry Pi, node MCU, servo motors, DC geared motors, Power supply 12v, USB camera.

**Software:** Applications required for fetching data from users and for controlling the bot, Python for processing the images, and Arduino sketch for node MCU

### 2.1 HARDWARE COMPONENTS

The hardware components used in this project is inexpensive and easily accessible, both online and in the local market. Every aspect of the project was created from the ground up.

**The following is a brief summary and list of the hardware used in the project:**

#### Raspberry 4 model B:

The Raspberry Pi was chosen because it allows for simple video transmission through Wi-Fi. The Raspberry Pi serves as the robot's central processing unit (CPU). The Raspberry Pi 4 Model B is the fourth generation Raspberry Pi. This versatile



credit-card-sized single board machine can be used for a variety of tasks.



**Fig-2:** Raspberypi

The following are the reasons for choosing the RPi as the main processing board:

1. The Raspberry Pi consumes very little fuel. It can be charged by attaching it to a power bank.
2. The price is poor in comparison to other production boards.
3. The Raspbian OS, which is open source, runs the Raspberry Pi. It consists a SD slot for saving data and installing Raspbian OS.
5. It has a 40-pin extended GPIO to help with "real-world" projects.

### Motor Driver IC L239D:

The DC motor can be driven in any direction thanks to the L293D. The L293D is a 16-pin integrated circuit that can power two DC motors at the same time. The Arduino instructs the gear motors to turn in the direction defined. Since the Arduino's strength is inadequate to drive the gears, we use a motor driver IC.

### Node MCU:

The Node MCU is a development board that includes the esp8266 and the same-named firmware. The esp8266 receives its 3.3 V power from the 5 V supplied by the USB. The node MCU firmware is a Lua language interpreter that has little to do with Arduino; however, the node MCU development board can be used in the same way that any other esp8266 development board can. Here the Node MCU board serves as the robot's directional processor. The node MCU processes the commands to push the robot in the designated direction and sends them to a driver IC, which makes the motors to work as the user required.



**Fig-3:** Node MCU

### Power bank:

The Raspberry Pi on the robot is powered by a 10000mah Ambrane power bank.



**Fig-4:** Power bank

### Virtual Reality Headset:

A VR headset helps the user to immerse themselves in virtual reality with the use of visuals.



**Fig-5:** VR headset

### USB CAMERA:



**Fig-6:** usb camera

We are using a USB camera for video capturing and this camera is fixed to servo motors such that movement of the servo makes the camera move in that required direction.

### 2.2 Software:

The following are the primary programming languages and applications used:

#### Raspbian OS:

The Raspbian operating system is mounted on a 32GB memory card. This is then loaded into the Raspberry Pi board's memory card slot. The power bank is linked to the board and provides the necessary power required to on. By entering the username and password in the remote desktop, you can access the Raspberry Pi kernel.

#### Python:

Python is latest programming language to use in this project to communicate with the user on the Raspberry Pi.

#### Arduino IDE (Arduino Development Environment):

The node MCU is programmed in the C/C++ programming language. It is used for movement of robot in this project

### Application required for movement of camera:

For the movement of the camera, we are using a wireless IMU application. We have typed the IP address of raspberry pi in the application.

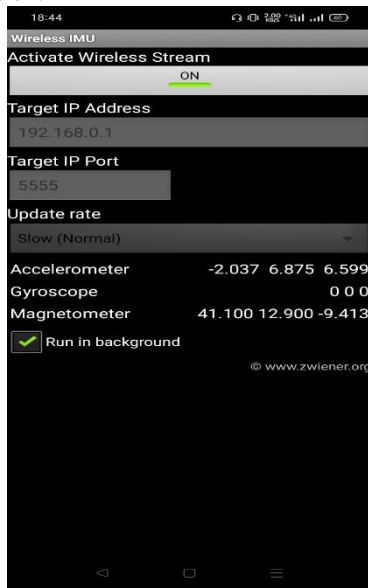


Fig-6 : Wireless IMU app

### Application required for movement of Robot:

An application was created to control the robot



Fig-7: Application for movement of robot

### Streaming:

Streaming is a technique in which the server delivers the response to a piece request. There are a couple of reasons why it would be helpful:

1. For very large responses, having to compile an answer in memory and then return it to the client can be inefficient. Another option is to write the answer to disc and then use `flask.send_file()` to return the file, but this adds I/O to the equation. If the data can be provided in chunks, providing the answer in small chunks is a much better solution.

2. Real-time data, A request for certain applications which need to return data from a real-time source. A real-time video or audio stream is an excellent example of this. This method is used by a lot of surveillance cameras to stream video to web browsers

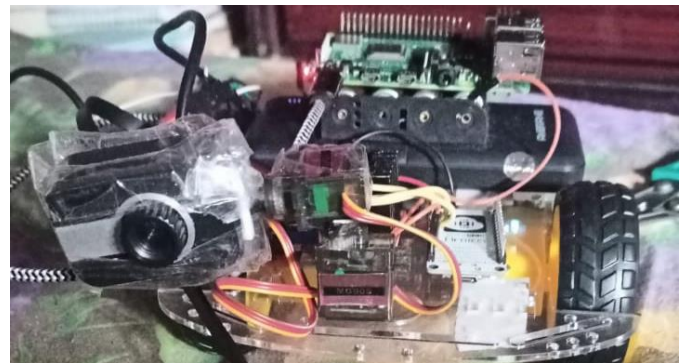
**Implementing Streaming With Flask:** Flask supports streaming responses natively through generator functions. A generator is a special feature that can be started and stopped. Flask's approach to generator functions is that A streamed response route must return a Response object that has been initialized with the generator feature. Flask then handles calling the generator and submitting all partial results to the client as chunks.

There are numerous methods for streaming video to browsers, each with advantages and disadvantages. The approach that works well with Flask's streaming function is to stream a series of independent JPEG images. This is known as Motion JPEG, and it is used by a large number of IP surveillance cameras. This approach has low latency but poor quality because JPEG compression is inefficient for motion.

### RESULTS:

After arranging all the hardware properly in the required manner and install and run the software, we successfully completed the telepresence robot and user can be able to experience virtual reality.

### FINAL PROTOTYPE



### 3. CONCLUSIONS

1. This vr robot used for protecting households. In the situation when the house owner needs to protect his/her house he/she needs a servant so in that case, this robot is helpful.
2. If the robot is manufactured by fire resistant, So this robot is used in retrieve operations.
3. In the military, instead of sending a soldier to sight for monitoring this robot can be sent. if any unforeseen attack occurred losing of life is less, only the robot will be damaged so we can protect the life of a soldier.
4. In Medical cases when the doctor is unable to go near the patient, this robot can be used to check the situation of the patient.

### REFERENCES

- [1] Kwon, O.H., Koo, S.Y., Kim, Y.G. and Kwon, D.S., 2010, October. Telepresence robot system for English tutoring. In 2010 IEEE workshop on advanced robotics and its social impacts (pp. 152-155). IEEE.
- [2] Documentation by <https://www.raspberrypi.org>
- [3] Stack overflow website for programming concepts
- [4] Github –Flask video streaming

- [5] Hossain, Mohammad Tanzir Kabir, "A Real-time Surveillance Nazmul Mini-rover Based on OpenCV-Python J AVA Using Raspberry Pi 2", 2015 IEEE International Conference on Control System, Computing and Engineering, 27 - 29 November 2015, Penang, Malaysia.