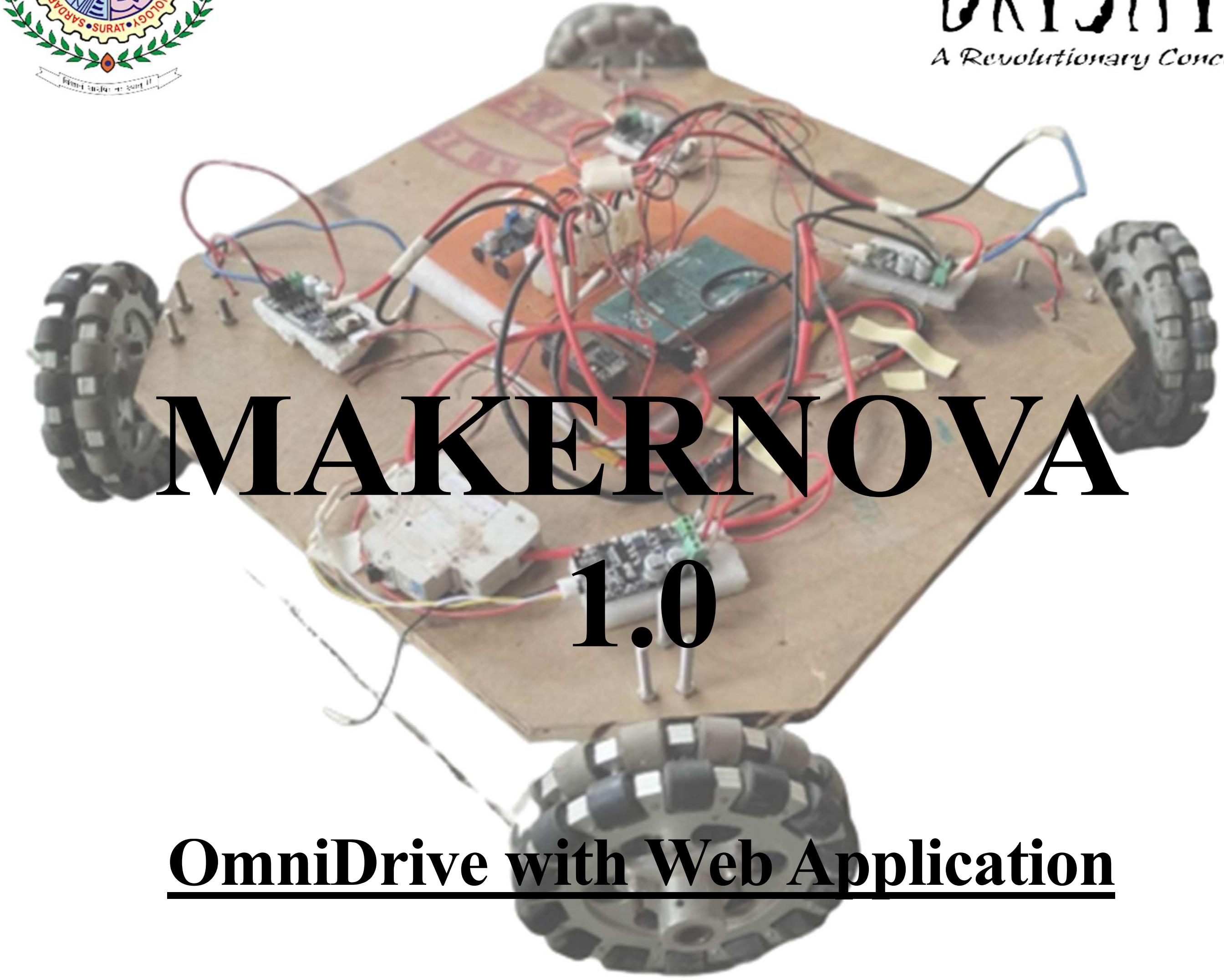


# MAKERNOVA 1.0

## OmniDrive with Web Application

# Project Documentation

## Problem Statement

The project creates a web-controlled 4-wheel omnidirectional drive using Omni Wheels, enabling seamless multidirectional mobility in confined spaces.

| TEAM MEMBERS | TEAM MENTORS |
|---|---|
| Aadil Siddiqui | Rohit Sultania |
| Ambar Gupta | Aditi Tapariya |
| Deshdeepak Verma | Harshal Patel |
| Dhruv Gandhi | Tarang Chaudhari |
| Dhruv Shah | |
| Neha Dhok | |
| Om Ramanuj | |
| Saiyam Kotadiya | |
| Saksham Samarth | |
| Seshunadh Tanuj | |
| Kashyap Mohanty | |
| Vaishnavi Reddy | |

# TABLE OF CONTENTS

**ABSTRACT:** ........ **4**

**INTRODUCTION:** ........ **5**

**SOFTWARE USED:** ........ **6**

PROTEUS SOFTWARE: ........ 6

ARDUINO IDE: ........ 6

VS CODE: ........ 7

WOKWI: ........ 7

**HARDWARE COMPONENTS:** ........ **8**

ARDUINO: ........ 8

ARDUINO UNO: ........ 8

ESP32 ........ 9

L293D MOTOR DRIVER ........ 10

MD13s MOTOR DRIVER ........ 11

**WEB SERVER** ........ **16**

CSS 3 ........ 17

JAVA SCRIPT: ........ 17

Why WebSocket over HTTP? ........ 17

Knowledge on various port numbers used throughout the data transaction ........ 18

mDNS: ........ 19

1. Synchronous Web Server: ........ 19

2. Asynchronous Web Server: ........ 19

BASIC COMMUNICATION PROTOCOL ........ 20

UART PROTOCOL ........ 20

**KINEMATICS** ........ **22**

Kinematics of Three-wheel omni-directional bot ........ 22

Four wheeled omnidirectional bot ........ 23

Inverse kinematics ........ 23

**ERRORS & CORRECTIONS** ........ **25**

**WORKFLOW** ........ **28**

**OUR CODES** ........ **29**

**REFERENCES** ........ **30**

# ABSTRACT:

This project introduces the concept of an “Omni Drive,” a novel approach to vehicular mobility that combines cutting-edge driving capabilities with seamless integration into web applications. The integration of web applications into the Omni Drive ecosystem further enhances its functionality. Through a user-friendly web interface, individuals can remotely monitor and control their vehicles, access real-time diagnostics, and plan routes efficiently. ESP32 comes with built-in Wi-Fi and Bluetooth capabilities which can directly connect the ESP32 to a wireless network and communicate with it over the internet, eliminating the need for additional shields or modules. The bot's omnidirectional movement capabilities, enabled by specialized omni wheels, allow for seamless navigation without changing orientation. The system’s versatility and user-friendly interface open doors to applications in fields like remote surveillance, education, and collaborative research.

# INTRODUCTION:

An omni-drive or holonomic drive, is a type of robotic or mechanical system that allows movement in any direction without the need to change the orientation of the device. This means the device can move forward, backward, sideways, and even rotate while maintaining its original orientation. One of the interesting applications of omni-drives is in web-controlled systems, where users can control the movement of the device through a web interface. This interface can be accessed through a standard web browser on a computer, smartphone, or tablet. Users can remotely control the movement of an Omni drive robot using web applications. They can provide directional commands and adjust the speed of each wheel, enabling seamless navigation even from a far distance. Web-controlled omni-drives are commonly used in various fields, including robotics, industrial automation, entertainment, and education. These systems often consist of a platform or vehicle equipped with omnidirectional wheels or mecanum wheels. These wheels are equipped with multiple rollers or treads mounted at an angle around their circumference. By independently controlling the speed and direction of each omni wheel, the system can achieve intricate movements like lateral motion, diagonal motion, and rotation on the spot. It is also connected with sensors, controllers, and a connection to the internet. The components involved in it are ESP32, MD13S (Motor driver), LM2596 buck converter (Voltage regulator), and omni wheels. An ESP32 onboard the device processes the sensor data and generates control signals for the omnidirectional wheels. It calculates the necessary wheel speeds and directions to achieve the desired movement.

# SOFTWARE USED:

## PROTEUS SOFTWARE:

The Proteus Design Suite is a proprietary software tool suite used primarily for electronic design automation. The software is used mainly by electronic design engineers and technicians to create schematics and electronic prints for manufacturing printed circuit boards. The microcontroller simulation in Proteus works by applying either a hex file or a debug file to the microcontroller part on the schematic. It is then co-simulated along with any analog and digital electronics connected to it. This enables its use in a broad spectrum of project prototyping in areas such as motor control, temperature control, and user interface design.

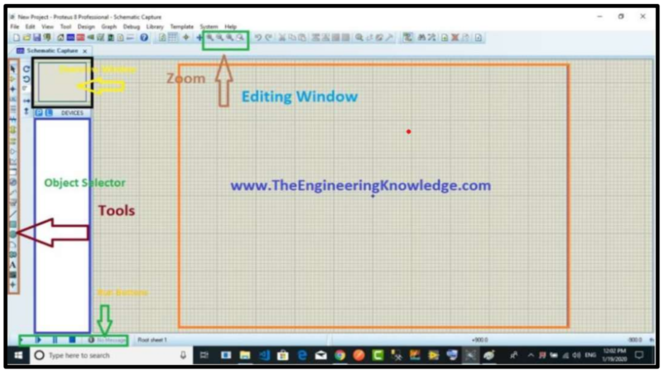
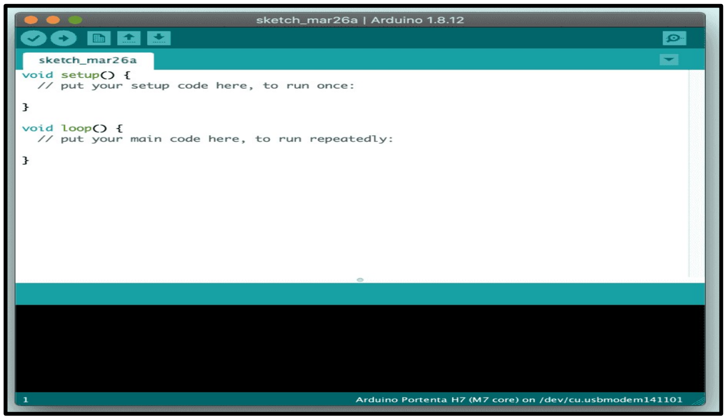

New Project - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
Overview Window
Zoom
Editing Window
DEVICES
Object Selector
Tools
www.TheEngineeringKnowledge.com
Run Buttons
No Message
Root sheet 1
Type here to search

**Fig 1: Interface of Proteus**

## ARDUINO IDE:

The Arduino software, commonly referred to as the Arduino IDE (Integrated Development Environment), serves as a vital platform for programming Arduino microcontrollers. One of its prominent features is the pinout diagram, which visually represents the layout of pins on the selected Arduino board, aiding users in correctly connecting sensors, actuators, and other components. This visual guide simplifies the hardware setup process and minimizes errors. Its accessibility, coupled with a wide array of libraries and community support, ensures that users can easily realize their creative ideas and innovations using Arduino. The libraries can be used to manage various boards such as esp32.


sketch_mar26a | Arduino 1.8.12
sketch_mar26a

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

1
Arduino Portenta H7 (M7 core) on /dev/cu.usbmodem141101

**Fig 2: Interface of Arduino IDE**

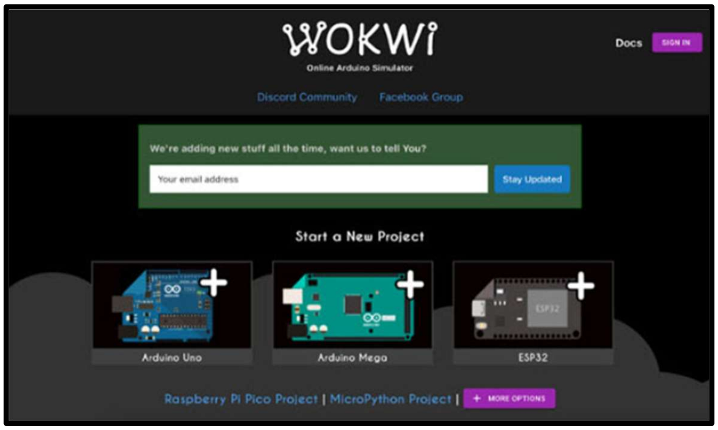
## VS CODE:

Visual Studio Code (VS Code) is a lightweight yet powerful code editor widely used for designing web pages and web applications. It offers a user-friendly interface, an extensive collection of extensions, and robust features that enhance web development. Developers can take advantage of features like syntax highlighting, intelligent code suggestions, and integrated version control. VS Code's real-time preview and debugging capabilities allow designers to instantly visualize the changes they make to their webpage's HTML, CSS, and JavaScript code. Furthermore, its seamless integration with various web technologies and frameworks makes it an ideal choice for creating responsive and visually appealing web content efficiently. Whether you're a beginner or an experienced developer, Visual Studio Code streamlines the webpage design process, contributing to a smoother and more productive development workflow.

**Fig 3: Visual Studio**

## WOKWI:

Wokwi Simulator is a web-based platform that revolutionizes electronics experimentation and learning. Through its interactive interface, users can design, simulate, and refine electronic circuits virtually, offering a comprehensive playground for hobbyists, students, and professionals alike. The platform boasts an extensive component library that includes microcontrollers, sensors, actuators, and more, enabling users to create intricate circuits with ease. This virtual environment not only accelerates prototyping and design iteration but also serves as an invaluable educational tool. With collaborative features, visual feedback, and a supportive community.



We're adding new stuff all the time, want us to tell You?

Start a New Project

**Fig 4: Interface of Wokwi online simulator**

# HARDWARE COMPONENTS:

## ARDUINO:

Arduino is an open-source electronics platform that consists of both hardware and software components, designed for building and prototyping interactive electronic projects.

### ARDUINO UNO:

Arduino Uno is one of the most popular and widely used microcontroller boards within the Arduino ecosystem. It is an entry-level board designed for beginners and experienced users alike, providing a straightforward and versatile platform for creating various electronic projects. The Arduino Uno is based on the Atmel ATmega328P microcontroller It is powered by the Atmel ATmega328P microcontroller, which is an 8-bit AVR microcontroller with 32 KB of flash memory for storing the program, 2KB of SRAM for data storage, and 1KB of EEPROM for non-volatile data storage.

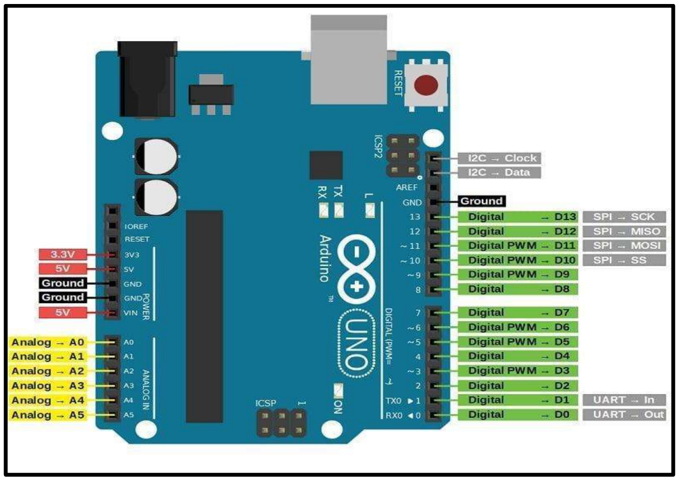


**Fig 5: Layout of Arduino UNO**

### Components of Arduino UNO:

**Digital I/O Pins:** The board has 14 digital input/output (I/O) pins, among which 6 pins can be used as PWM (Pulse Width Modulation) outputs to control devices like motors or LED brightness. There are 14 digital pins, of which pins 3,5,6,9,10, and 11 are PWM pins.

**Analog Inputs:** Arduino Uno has 6 analog input pins, which can be used to read analog voltage levels from sensors or other analog devices.

**USB Interface***:* Arduino Uno can be connected to a computer via a USB cable, which serves both as a power source and a means of programming the board.

**Power Jack:** It also has a power jack that allows you to power the board using an external power supply.

## ARDUINO DUE:

The Arduino Due is a microcontroller board based on the Atmel SAM3X8E ARM Cortex-M3 CPU. It is the first Arduino board based on a 32-bit ARM core microcontroller. It was introduced to provide more processing power and advanced capabilities compared to the traditional 8-bit Arduino boards.

Key features of Arduino due:

- The Arduino Due provides 54 digital input/output pins, with 12 of them capable of analog input. Additionally, there are DAC (Digital to Analog Converter) pins that enable the generation of analog pins.

- The board offers multiple communication interfaces, including four UART (Universal Asynchronous Receiver-Transmitter) ports, two SPI (Serial Peripheral Interface) ports, two I2C (Inter-Integrated Circuit) ports, and a USB port. These interfaces allow communication with a wide range of devices and peripherals.

- The Arduino Due operates at 3.3V, which is different from the 5V operating voltage of many other Arduino boards. This means that peripherals designed for 5V operation might require level shifting or adaptation to work properly with the Due.

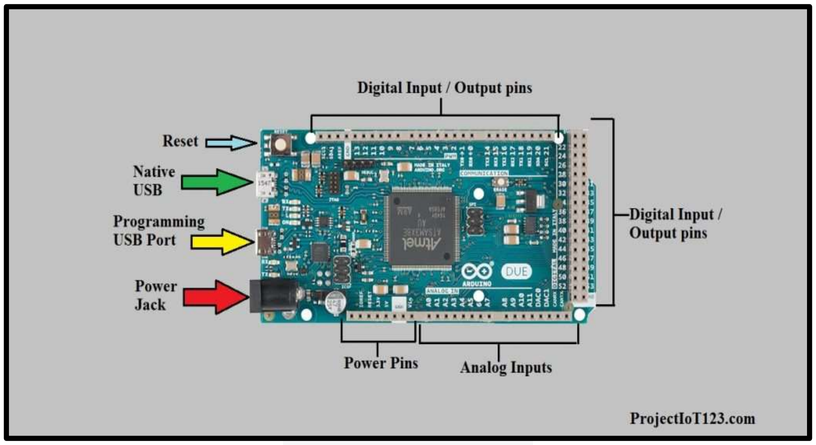


**Fig 6: Arduino Due**

## ESP32

Created by Espressif Systems, ESP32 is a low-cost, low-power system on a chip (SoC) series with Wi-Fi & dual-mode Bluetooth capabilities. At its heart, there's a dual-core or single-core Ten silica Xtensa LX6 microprocessor with a clock rate of up to 240MHz. ESP32 is highly integrated with built-in antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power management modules. Engineered for mobile devices, wearable electronics, and IoT applications, ESP32 achieves ultra-low power consumption through power-saving features including fine-resolution clock gating, multiple power modes, and dynamic power scaling.
The ESP32 peripherals include:

- 18 Analog-to-Digital Converter (ADC) channels
- 3 SPI interface
- 3 UART interfaces
- 2 I2C interfaces

- 16 PWM output channels
- 2 Digital-to-Analog Converters (DAC)
- 2 I2S interfaces
- 10 Capacitive sensing GPIOs

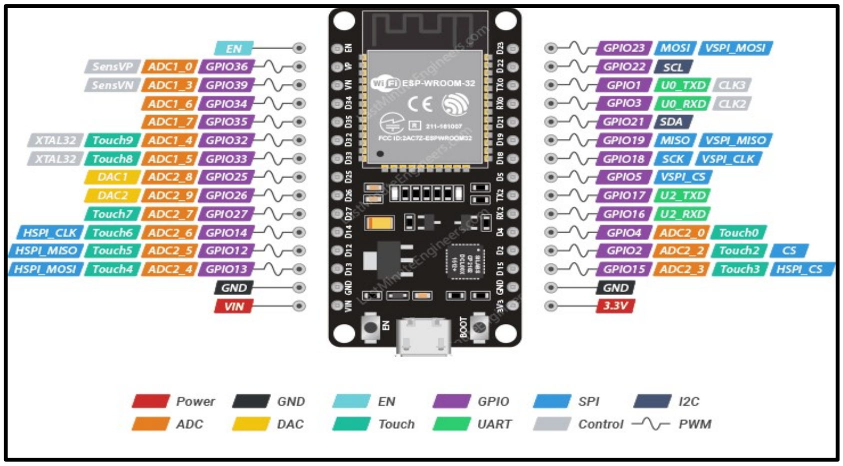


**Fig 7: Pin Layout of ESP32**

# L293D MOTOR DRIVER

## What is a Motor Driver?

A motor driver, also known as a motor controller, is an electronic device or circuit that is used to control the speed, direction, and torque of an electric motor. The main purpose of a motor driver is to take the low-power control signals from a microcontroller or other control unit and convert them into higher-power signals that can drive the motor. The motor driver ensures that the motor operates efficiently, safely, and according to the desired specifications.

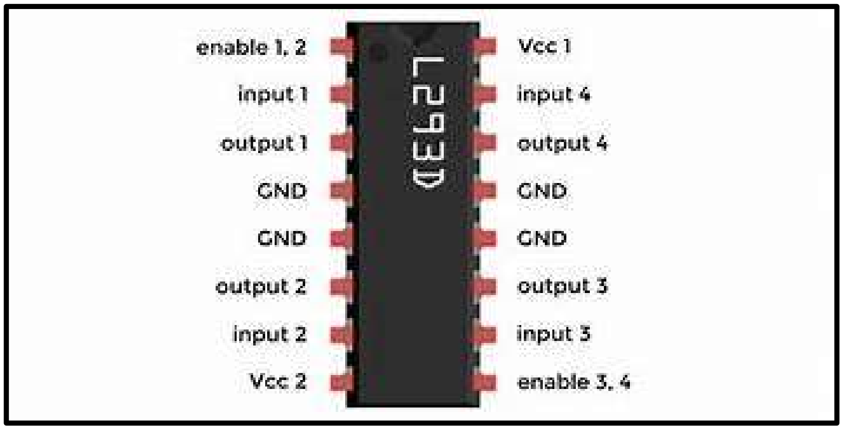


**Fig 8: Pin Diagram of Motor Driver (L293D)**

The L293D IC has two separate H-bridges, which allow it to control the direction of two DC motors independently or drive a single bipolar stepper motor i.e., the current will flow through output 1

.

- Pin 3 (Output 1): This pin is connected to one of the terminals of motor 1
- Pin 4, Pin 5 (GND): These pins should be connected to the circuit's ground. These also work as heat sinks.
- Pin 6 (Output 2): This pin is connected to one of the terminals of motor 1
- Pin 7 (Input 2): When this pin is given High or Logic 1, Output 2 becomes high i.e., the current will flow through Output 2
- Pin 8 (VCC2): This is the voltage required to run the motor. It can be greater than IC voltage Vcc 1. If we are driving 12 V DC motors then make sure that this pin is supplied with 12 V
- Pin 9 (Enable 3,4): When this pin is given High or logic 1, the right part of the IC will work, and when it is low the right part doesn't work. So, this pin is the Master Control pin for the right part of the IC
- Pin 10 (Input 3): When this pin is given High or Logic 1, Output 3 becomes high i.e., the current will flow through Output 3
- Pin 11 (Output 3): This pin is connected to one of the terminals of motor 2
- Pin 12,13 (GND): These pins should be connected to the circuit's ground
- Pin 14 (Output 4): This pin is connected to one of the terminals of the motor
- Pin 15 (Input 4): When this pin is given High or Logical 1, output 4 becomes High i.e., the current will flow through output 4
- Pin 16 (VCC1): This pin provides power to the IC. So, this pin should be supplied with a 5V. Used to give internal biasing of MOSFET.

Note: -

- All input pins are called direction-controlled pins.
- Input pins follow XOR gate logic.
- Enable pins can also be used as PWM pins.

## MD13s MOTOR DRIVER

MD13S has great improvement in current protection which MD10C does not have. With the new circuit and design, MD13S is able to output current to 30A max. The continuous current is at 13A (at room temperature of 25 degrees C). Some of the key features of the motor driver are given below.

**Features:**

- Bi-directional control for 1 brushed DC motor.
- Support motor voltage ranges from 6V to 30V
- Maximum current up to 13A continuous and 30A peak (10 seconds).
- GROVE compatible
- 3.3V and 5V logic level input.
- Solid state components provide faster response time and eliminate the wear and tear of mechanical relays.
- Fully NMOS H-Bridge for better efficiency and no heat sink is required.
- Speed control PWM frequency up to 20KHz (Actual output frequency is same as input frequency).
- Support both locked-antiphase and sign-magnitude PWM operation, NOT RC (Radio Control) PWM.
- SMD compatible
- RoHS, FCC, and CE compliance products.
- Dimension: 61mm x 33mm

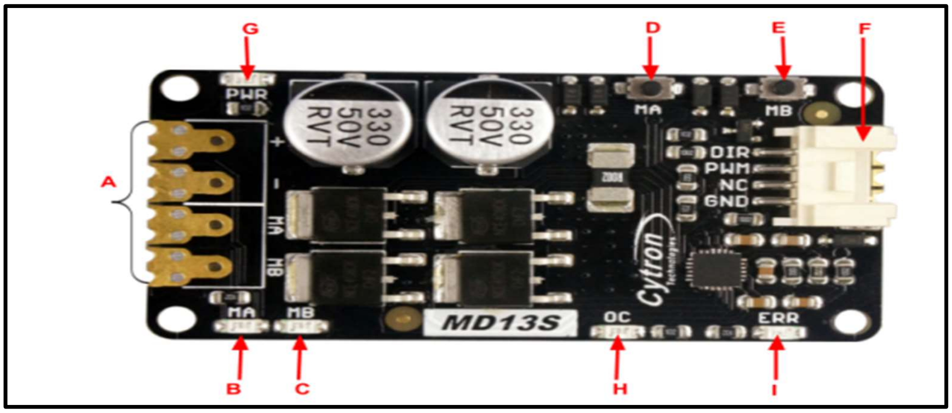


**Fig 9: MD13s Motor Drive**

| Label | Function |
|---|---|
| A | Terminal Block |
| B | MA LED Indicator |
| C | MB LED Indicator |
| D | MA Test Switch |
| E | MB Test Switch |
| F | Input pin |
| G | Power LED Indicator |
| H | OC LED Indicator |
| I | ERR LED Indicator |
| J | SMD Compatible |

**Table-1: Labels & its functions**

## BUCK CONVERTER

LM2596 DC-DC Buck Converter Adjustable Step-Down Power Supply Module is a buck converter step-down power module with a high-precision potentiometer, capable of driving a load up to 3A with high efficiency, which can work with free Duino, UNO, and other mainboards and basic modules. When the output current is greater than 2.5 A (or output power greater than 10W), add a heatsink to it.

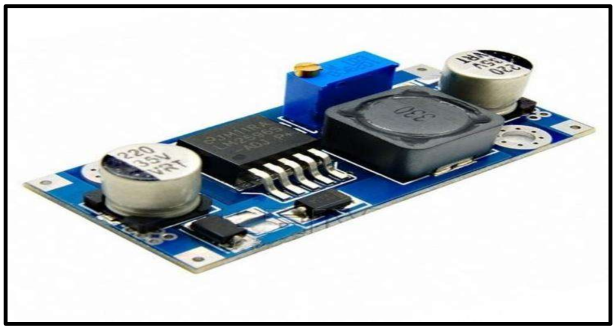


**Fig 10: Buck Convertor (LM2596)**

## Features:

- Input voltage:3-40V
- Output voltage:1.5-35V(Adjustable)
- Output current: Rated current is 2A, maximum 3A (Additional heatsink is required)
- Module Properties: non-isolated constant voltage module

## PS3:

The Play Station 3 (PS3) is a home video game console developed by Sony Computer Entertainment. The PS3 was a significant advancement in gaming technology, featuring powerful hardware capable of high-definition graphics, online connectivity, and a variety of multimedia capabilities. The primary purpose of the PS3 was gaming, and it had an extensive library of games across different genres, ranging from action and adventure to sports and simulation. The powerful hardware allowed for visually stunning and immersive gaming experiences.

**Fig 11. PlayStation 3**

## CIRCUIT DIAGRAM

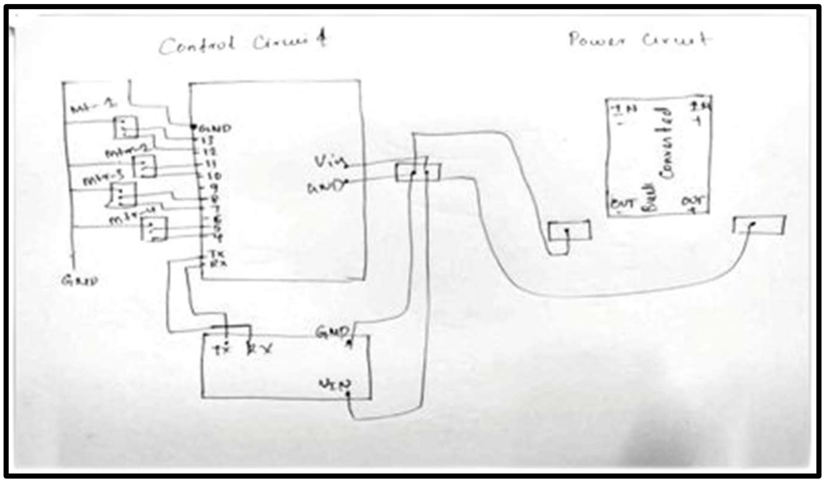


**Fig 12. Circuit diagram done on paper**

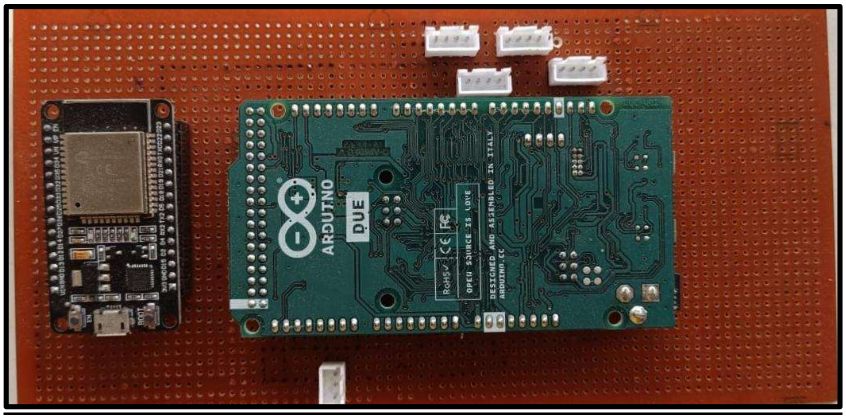


**Fig 13. Control Circuit implementation on GCB**

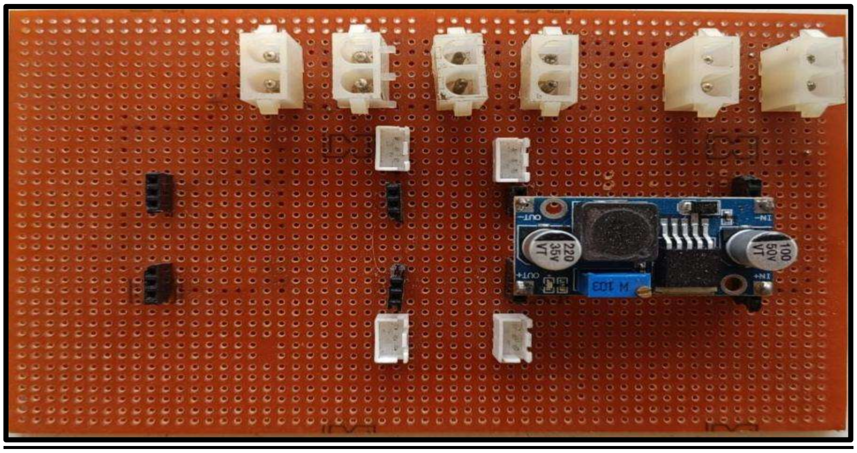


**Fig 14. Power Circuit implementation on GCB**

## SOLDERED CIRCUIT

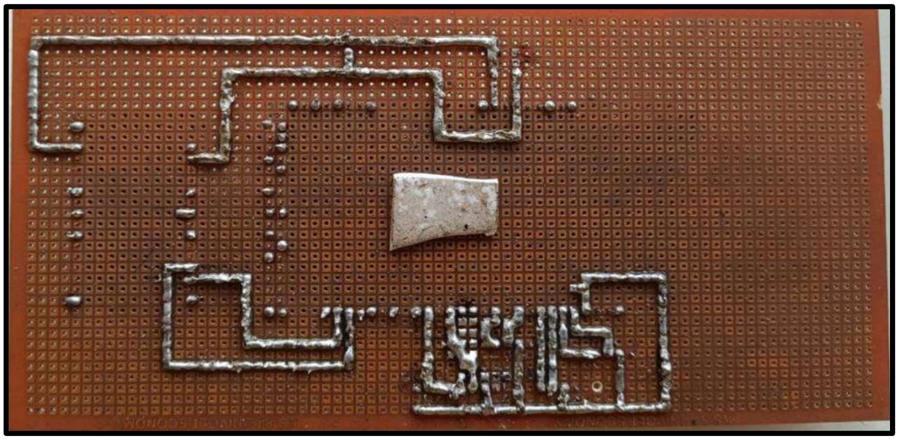


**Fig 15. Soldering of Control Circuit**

- Here we have used a total of 9 pins for controlling the motor driver (i.e. GND, 13, 12, 11, 10, 8, 7, 5, 4) with common ground.

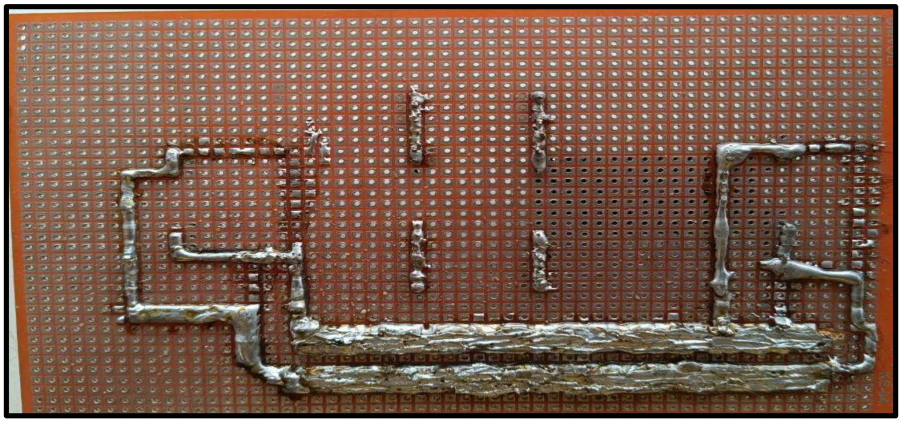


**Fig 16. Soldering of Control Circuit**

- As we are giving 12V from the battery here we have done 3 layers of soldering for input and ground.

# WEB SERVER

The web server at //esp.Local/

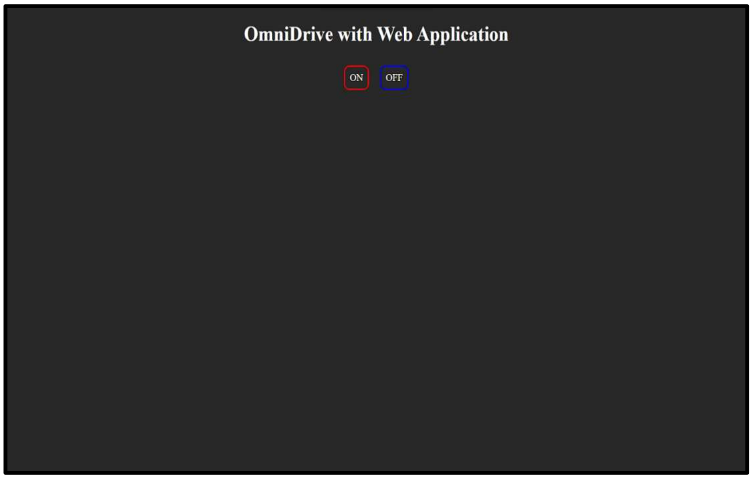

OmniDrive with Web Application

**Fig 17. Example Website which is used to turn on/off LED on ESP32**

**Technology stack used**

- HTML 5
- CSS 3
- JS

## HTML:

HTML (Hyper Text Markup Language) is the fundamental building block of websites. It is used to **form the layout of our website**

**Cross-Platform Compatibility:** HTML is supported by all major web browsers and can be rendered on various devices, including desktops, laptops, tablets, smartphones, and even smart TVs until it has a web browser. This cross-platform compatibility ensures that web pages can reach a wide audience regardless of the device or operating system being used.

**Ease of Maintenance***:* HTML separates the structure and content from the presentation (handled by CSS) and behavior (handled by JavaScript). This separation makes it easier to maintain and update websites since changes can be made to specific components without affecting the entire site.

**Progressive Enhancement:** HTML supports the principle of progressive enhancement, where basic content is accessible to all users, and additional features are provided for users with modern browsers or capabilities. This approach ensures a better user experience for all users regardless of their device or browser limitations.

## CSS 3

CSS (Cascading Style Sheets) is a fundamental technology used in web development to control the presentation and layout of web pages. Here used between <style> tags are responsible for styling the HTML content, determining how the elements should appear on the screen, and creating visually appealing and consistent designs. It is an essential tool for web development, enabling developers to create visually appealing and responsive websites. It works in conjunction with HTML and JavaScript to build interactive and dynamic web applications.

### Key Points about CSS:

**Styling Language**: CSS is a stylesheet language that complements HTML. While HTML defines the structure and content of a web page, CSS is used to define the appearance, colors, fonts, margins, padding, and other visual properties.

**Responsive Design**: CSS facilitates responsive web design, where websites adapt their layout and appearance based on the user's device and screen size. Media queries in CSS allow developers to apply different styles based on specific conditions, ensuring an optimal user experience on various devices.

**Browser Compatibility:** CSS helps normalize the appearance of web pages across different web browsers. Each browser may interpret HTML and CSS slightly differently, but CSS allows developers to work around these differences and create consistent designs.

## JAVA SCRIPT:

JavaScript (JS) is a powerful and essential technology used in web servers for several key reasons:

**Client-Side Interactivity:** JavaScript enables interactivity and dynamic behavior on the client side, meaning it runs directly in the user's web browser. It allows developers to create responsive and interactive web pages where elements can change, update, or react to user actions without the need for a page reload.

**Event Handling:** JavaScript allows developers to attach event handlers to HTML elements, enabling them to respond to various user interactions such as clicks, keypresses, form submissions, and more. This enables the creation of user-friendly and interactive web interfaces.

**Animations and Effects:** JavaScript enables the creation of animations and visual effects, making websites more engaging and aesthetically appealing. Libraries like jQuery and modern CSS transitions, coupled with JS, allow developers to create stunning animations.

We used a WebSocket server to communicate with ESP32 instead of HTTP.

## Why WebSocket over HTTP?

**HTTP** and **WebSocket** are just a set of rules and methods set to help in the communication of 2 computers, where one asks for information and the other sends it. For the tech-savvy, both are basically communication protocols for client-server communication. The difference mainly is that WebSocket allows for two-way communication, called full duplex whereas HTTP works for one way only at a time, called half duplex. Thus, WebSocket allows the server to push information to the client.

Other technical differences are listed below: -

## Difference Between HTTP and WEB SOCKET

| HTTP | WEB SOCKET |
|---|---|
| **Duplex** | |
| Half | Full |
| **Messaging Pattern** | |
| Request-response | Bi-directional |
| **Service Push** | |
| Not natively supported. Client polling or streaming download techniques used. | Core feature |
| **Overhead** | |
| Moderate overhead per request/connection. | Moderate overhead to establish and maintain the connection, then minimal overhead per message. |
| **Intermediary/Edge Caching** | |
| Core feature | Not possible |
| **Supported Clients** | |
| Broad Support | Modern languages & clients |

**Table-2: Http vs. WebSocket**

## Knowledge on various port numbers used throughout the data transaction

A port number is a way to identify a specific process to which an internet or other network message is to be forwarded when it arrives at a server. A port number just identifies the way in which the message sent is to be sent or transmitted.

Different port numbers used in this are:

**80** – HTTP (Hypertext Transfer Protocol)

- used for transferring web pages

**81** - Torpark - Onion routing OR port

– commonly used for a proxy server (a gateway server between the client and the internet for security, and privacy)

– Used here for web socket server which is intermediate between client and internet

## mDNS:

On the interconnected network of computers called the internet, each computer has a unique ID called IP Address. Similarly, each webpage has its own unique IP Address. Though the computer may love numbers for such a task (E.g.: 172.168.1.1), We humans hate them so for our convenience we use the alphabet. So, we use a program called Domain Name System (DNS) to bind these unique combinations of alphabets and words with these IP Addresses.

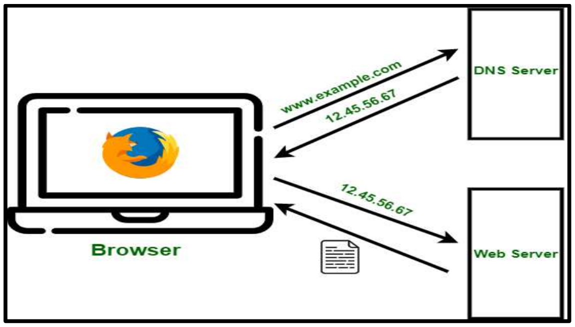


**Fig 18. Communication between Browser and Server**

## Why is Asynchronous Web Server used in our project? Difference between Synchronous and Asynchronous Web Servers?

There are 2 types of servers based on how they respond to a query asked to it. They are:

### 1. Synchronous Web Server:

A synchronous web server is a type of web server architecture where incoming client requests are handled one at a time, in a sequential manner. In this setup, each request is processed before the server moves on to the next one. This means that if one request takes a significant amount of time to process, it can block the server from serving other requests in the meantime.

### 2. Asynchronous Web Server:

The server immediately responds to the sent query even if there is already a query it is working on. Here, we needed a quick and responsive server that processes information and gives answers as soon as we give the input or change it. Also, the input given is pretty simple to process, so there is no worry of a query not being processed at all.

# BASIC COMMUNICATION PROTOCOL

**TX:** The data-sending end, generally plays the role of the transmitter, normally the TXD pin must be connected to the RXD Pin of other devices.

**RX**: The data-receiving end, generally plays the role of the receiver, normally The RXD pin must be connected to the TXD Pin of other devices

## UART PROTOCOL

UART stands for **Universal Asynchronous Receiver/Transmitter**. It's not a communication protocol like SPI and I2C, but a physical circuit in a microcontroller, or a stand-alone IC. A UART's main purpose is to transmit and receive serial data.

In UART communication, two UARTs communicate directly with each other. The transmitting UART converts parallel data from a controlling device like a CPU into serial form and transmits it in serial to the receiving UART, which then converts the serial data back into parallel data for the receiving device. Only two wires are needed to transmit data between two UARTs. Data flows from the Tx pin of the transmitting UART to the Rx pin of the receiving UART.

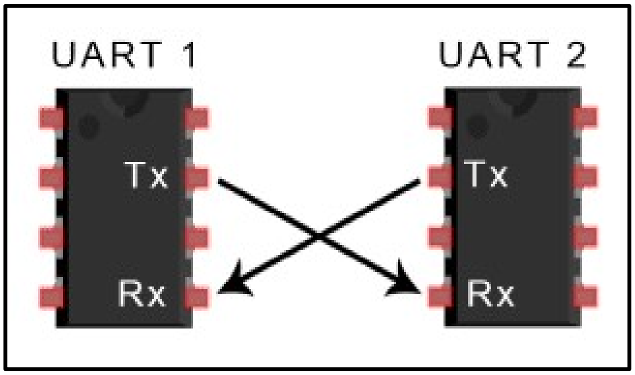


**Fig 19: UART between two microcontrollers (ESP32 and ARDUINO DUE)**

UARTs transmit data asynchronously, which means there is no clock signal to synchronize the output of bits from the transmitting UART to the sampling of bits by the receiving UART. Instead of a clock signal, the transmitting UART adds start and stop bits to the data packet being transferred. These bits define the beginning and end of the data packet so the receiving UART knows when to start reading the bits. When the receiving.

UART detects a start bit, it starts to read the incoming bits at a specific frequency known as the baud rate. Baud rate is a measure of the speed of data transfer, expressed in bits per second (bps). Both UARTs must operate at about the same baud rate. The baud rate between the transmitting and receiving UARTs can only differ by about 10% before the timing of bits gets too far off.

### Why are we using the UART protocol??

The UART protocol is commonly used for communication in omni drive systems. We also need simultaneous communication between receiver and sender and even URAT uses only 2 lines for this type of communication so we don't need to worry about the collision of messages as in the I2C protocol. These attributes make UART suitable for facilitating communication between a central control unit and the omni-drive system, enabling precise control, feedback, and coordination of movement in any direction.

### ADVANTAGES

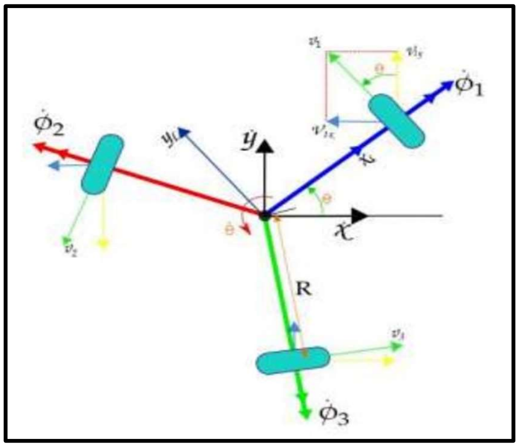
- Only uses two wires.
- No clock signal is necessary.
- Has a parity bit to allow for error checking.
- The structure of the data packet can be changed as long as both sides are set up for it.

### DISADVANTAGES

- The size of the data frame is limited to a maximum of 9 bits.
- Doesn't support multiple slave or multiple master systems.
- The baud rates of each UART must be within 10% of each other.

# KINEMATICS

## Kinematics of Three-wheel omni-directional bot



Wheel velocities of 3-wheel omni drive

The Omni Wheel are mounted in a 45-degree angle to the chassis.

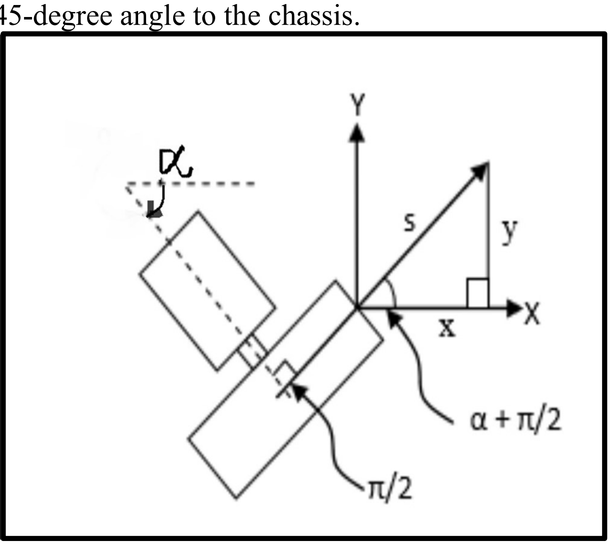


**Fig 20: Omni Wheel** **Fig 21: Kinematics of OMNI Wheels**

The robot has four Mecanum wheels, each at a 45-degree angle with respect to the robot's body frame
By trigonometry,
The motion of the robot can be decomposed into two components:

- Translation along the x-axis (forward and backward movement).
- Rotation around the z-axis (yaw or turning).

The following equations can be used to calculate the individual wheel velocities (Vi) required to achieve the desired translational velocity (x, y) and rotational velocity (W) of the robot:
For forward and backward movement (x):
X is the x component in the x direction and y is the component in y direction.

## Rotation

Robot rotation is just a simple sum of each motor speed.

$\omega = s_1 + s_2 + s_3$

Here, $S_i$ velocities of motors (i =1, 2, 3)

Inverse kinematics for measuring wheel angular velocity. In this study, the methods of inverse kinematics were used to determine the DC motor speed. The inverse kinematic formula for a robot with three wheels is defined as in Equation,

$$\begin{pmatrix} x \\ y \\ \omega \end{pmatrix} = \begin{pmatrix} \cos\left(\alpha_1 + \frac{\pi}{2}\right) & \cos\left(\alpha_2 + \frac{\pi}{2}\right) & \cos\left(\alpha_3 + \frac{\pi}{2}\right) \\ \sin\left(\alpha_1 + \frac{\pi}{2}\right) & \sin\left(\alpha_1 + \frac{\pi}{2}\right) & \sin\left(\alpha_1 + \frac{\pi}{2}\right) \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ s_3 \end{pmatrix}$$

By giving the values and inputting x, y, and w values, we get the velocities of the motors.

$$\begin{pmatrix} s_1 \\ s_2 \\ s_3 \end{pmatrix} = \begin{pmatrix} -0.33 & 0.58 & 0.33 \\ -0.33 & -0.58 & 0.33 \\ 0.67 & 0 & 0.33 \end{pmatrix} \begin{pmatrix} x \\ y \\ \omega \end{pmatrix}$$

$\alpha_i$ = angle of axel of wheel w.r.t x-axis

Similarly,

## Four wheeled omnidirectional bot

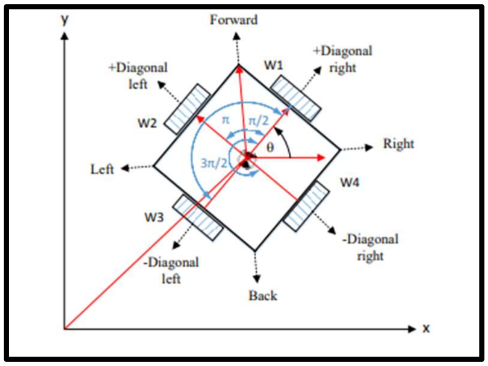


**Fig 22: Velocity vectors for 4-wheel omni drive**

## Inverse kinematics

To inverse will obtain equation, which provides the robot's velocities:

$$V_w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} -sin\left(\theta + \frac{\pi}{4}\right) & cos\left(\theta + \frac{\pi}{4}\right) & R \\ -sin\left(\theta + \frac{3\pi}{4}\right) & cos\left(\theta + \frac{3\pi}{4}\right) & R \\ -sin\left(\theta + \frac{5\pi}{4}\right) & cos\left(\theta + \frac{5\pi}{4}\right) & R \\ -sin\left(\theta + \frac{7\pi}{4}\right) & cos\left(\theta + \frac{7\pi}{4}\right) & R \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ \dot{\theta} \end{bmatrix}$$

Eight directions of the robot's motion relate to the four-wheel rotation direction

| DIRECTION | W1 | W2 | W3 | W4 |
|---|---|---|---|---|
| Forward | CW | CCW | CCW | CW |
| Left | CW | CW | CCW | CCW |
| Right | CCW | CCW | CW | CW |
| Back | CCW | CW | CW | CCW |
| +Diagonal Right | 0 | CCW | 0 | CW |
| -Diagonal Right | 0 | CW | 0 | CCW |
| +Diagonal left | CW | 0 | CCW | 0 |
| -Diagonal left | CCW | 0 | CW | 0 |

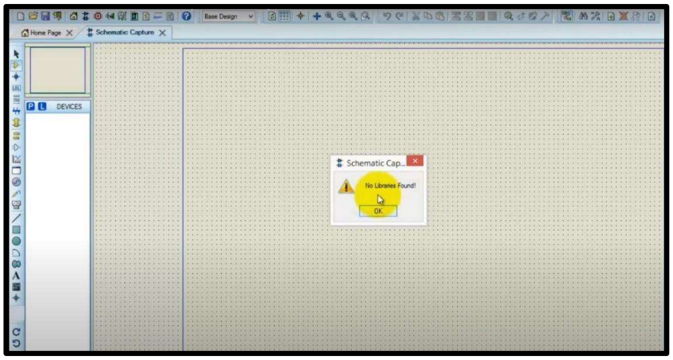
**Table 3: Directions of wheels**

# ERRORS & CORRECTIONS

**Error 1** - Proteus no libraries found for ESP 32

SOLUTION -

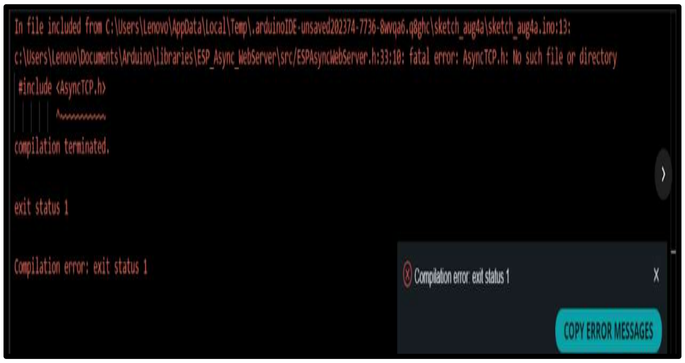
Step 1 - Copy the JSON link from GitHub then paste it in the Preferences section in the File menu.
Step 2 - Include a library named ESP 32 Espressif from the include library section in the Sketch menu.



No Libraries Found!

**Fig 23: Error-1**

**Error 2** - Compilation occurred in Arduino IDE due to using trouble-free multi-connection network code.



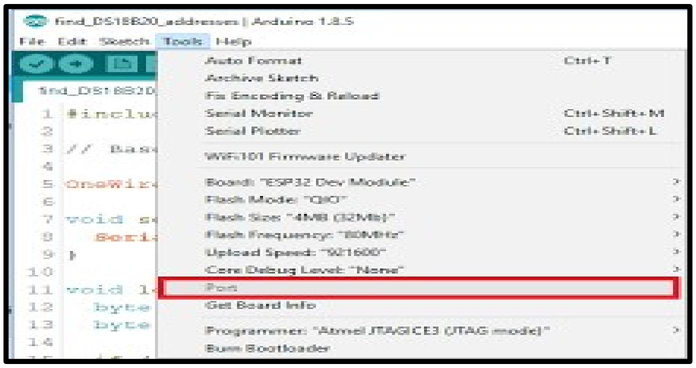
```
In file included from C:\Users\Lenovo\AppData\Local\Temp\.arduinoIDE-unsaved202374-7736-8wvqa6.q8ghc\sketch_aug4a\sketch_aug4a.ino:13:
c:\Users\Lenovo\Documents\Arduino\libraries\ESP_Async_WebServer\src/ESPAsyncWebServer.h:33:10: fatal error: AsyncTCP.h: No such file or directory
 #include <AsyncTCP.h>
          ^~~~~~~~~~~~
compilation terminated.

exit status 1

Compilation error: exit status 1
```

Compilation error: exit status 1

**Fig 24: Error-2**

SOLUTION -

Therefore, we installed a library named Async TCP library from GitHub.


find_DS18B20_addresses | Arduino 1.8.5
File Edit Sketch Tools Help
find_DS18B20
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater

Board: "ESP32 Dev Module"
Flash Mode: "QIO"
Flash Size: "4MB (32Mb)"
Flash Frequency: "80MHz"
Upload Speed: "921600"
Core Debug Level: "None"

Programmer: "Atmel JTAGICE3 (JTAG mode)"

```
#inclu
// Bas
OneWir
void se
  Seri
}
void l
  byte
  byte
```

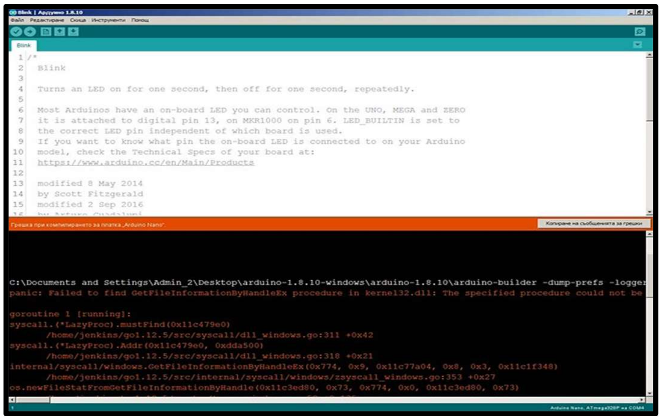
**Fig 25: location of ports**

**Error 3**- The Port for ESP 32 is not working in Arduino IDE.

SOLUTION -

We installed the cp210x_universal windows driver from Google as the cp210x driver is used for Virtual COM Port to facilitate host communication with CP210x products.

**Error 4** - mDNS server error during compiling code in Arduino ide


Blink | Ардуино 1.8.10
Файл Редактиране Скица Инструменти Помощ
Blink
Грешка при компилирането за платка „Arduino Nano".
Копиране на съобщенията за грешки
Arduino Nano, ATmega328P на COM4

```
/*
  Blink

  Turns an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
  the correct LED pin independent of which board is used.
  If you want to know what pin the on-board LED is connected to on your Arduino
  model, check the Technical Specs of your board at:
  https://www.arduino.cc/en/Main/Products

  modified 8 May 2014
  by Scott Fitzgerald
  modified 2 Sep 2016
```

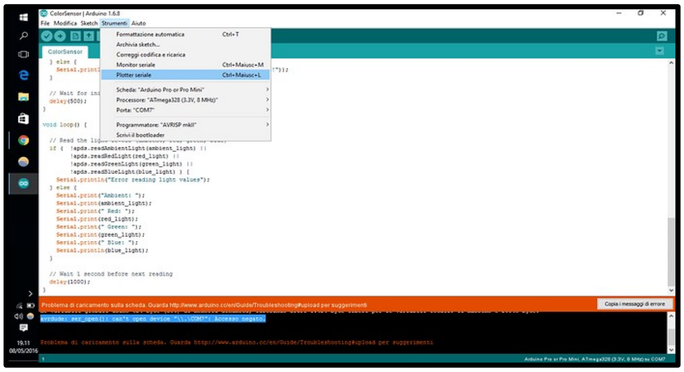
```
C:\Documents and Settings\Admin_2\Desktop\arduino-1.8.10-windows\arduino-1.8.10\arduino-builder -dump-prefs -logger
panic: Failed to find GetFileInformationByHandleEx procedure in kernel32.dll: The specified procedure could not be

goroutine 1 [running]:
syscall.(*LazyProc).mustFind(0x11c479e0)
        /home/jenkins/go1.12.5/src/syscall/dll_windows.go:311 +0x42
syscall.(*LazyProc).Addr(0x11c479e0, 0xdda500)
        /home/jenkins/go1.12.5/src/syscall/dll_windows.go:318 +0x21
internal/syscall/windows.GetFileInformationByHandleEx(0x774, 0x9, 0x11c77a04, 0x8, 0x3, 0x11c1f348)
        /home/jenkins/go1.12.5/src/internal/syscall/windows/zsyscall_windows.go:353 +0x27
os.newFileStatFromGetFileInformationByHandle(0x11c3ed80, 0x73, 0x774, 0x0, 0x11c3ed80, 0x73)
```

**Fig 26: Error-4**

SOLUTION -

Step 1 - Changing to a different board from the sketch menu.
Step 2 - Updated the library ESP 32 Espressif from the tool's menu.

**Error 5 -** Access denied error while compiling code in Arduino


ColorSensor | Arduino 1.6.8
File Modifica Sketch Strumenti Aiuto
ColorSensor
Formattazione automatica Ctrl+T
Archivia sketch...
Correggi codifica e ricarica
Monitor seriale Ctrl+Maiusc+M
Plotter seriale Ctrl+Maiusc+L
Programmatore: "AVRISP mkII"
Scrivi il bootloader
Copia i messaggi di errore
19.11
08/05/2016

Scheda: "Arduino Pro or Pro Mini"
Processore: "ATmega328 (3.3V, 8 MHz)"
Porta: "COM7"

```
  } else {
    Serial.print
  }

  // Wait for ini
  delay(500);
}

void loop() {

  // Read the ligh
  if (  !apds.readAmbientLight(ambient_light) ||
        !apds.readRedLight(red_light) ||
        !apds.readGreenLight(green_light) ||
        !apds.readBlueLight(blue_light) ) {
    Serial.println("Error reading light values");
  } else {
    Serial.print("Ambient: ");
    Serial.print(ambient_light);
    Serial.print(" Red: ");
    Serial.print(red_light);
    Serial.print(" Green: ");
    Serial.print(green_light);
    Serial.print(" Blue: ");
    Serial.println(blue_light);
  }

  // Wait 1 second before next reading
  delay(1000);
}
```

Problema di caricamento sulla scheda. Guarda http://www.arduino.cc/en/Guide/Troubleshooting#upload per suggerimenti

```
avrdude: ser_open(): can't open device "\\.\COM7": Accesso negato.


Problema di caricamento sulla scheda. Guarda http://www.arduino.cc/en/Guide/Troubleshooting#upload per suggerimenti
```

Arduino Pro or Pro Mini, ATmega328 (3.3V, 8 MHz) su COM7

**Fig 27: Error-5**

SOLUTION -

Reinstalled the different version (1.8.19) of Arduino from the google chrome.

**Error 6:** The font of the webpage wasn’t appearing as designed in VSCode.

SOLUTION-

Thus, the WIFI mode of the Esp was changed from “softAP " to “WIFI_STA ". Thus, the connection methodology was changed which fixed the issue due to internet connectivity being available.

**Error 7**: The Soldering line was thin due to which the circuit was not conducting properly.

SOLUTION-

Soldered again properly to ensure no gaps.

**Error 8:** In the motor driver, Ground, PWM & DIR pins got shorted.

SOLUTION-

Resoldering of all the pins in that motor driver.

# WORKFLOW

1. **10 JULY 2023** - Make Everyone familiar with the course of action through the online mode.
2. **16 JULY 2023** - Run and implement Arduino code using the simulation proteus.
3. **18 JULY 2023 -** Discussion about components to be used in the Omni web bot.
4. **19 JULY 2023 -** Brief Intro on ESP32 and L293D and integration of esp32 on Arduino IDE**.**
5. **31 JULY 2023 -** Mathematics behind the movement of 3 Wheels Omni-Drive, PWM Mapping concept.
6. **2 AUGUST 2023 -** Getting Started with HTML, CSS, JavaScript.
7. **5 AUGUST 2023** - Implementation of Web-Server on ESP32 & LED Blinking using Web Page.
8. **10 AUGUST 2023 -** Started hardware designing, Designing of further web design.
9. **11 AUGUST 2023 -** To complete soldering of control and power circuits, Web design for Joy-Stick, Learning to control the bot using PS3.
10. **12 AUGUST 2023 -** Understanding mathematics and control circuit, Complete basic website, PS3 integration with its calibration and solder power circuit.
11. **14 AUGUST 2023 -** Applying the code of Arduino on Bot**,** Introduction to Web-Sockets.
12. **16 AUGUST 2023** - Developing the UART Communication using TX RX pins of Arduino and ESP.
13. **17 AUGUST 2023 -** Connecting the PS3 to ESP and sending the data to Arduino**,** Finalizing the web page.
14. **18 AUGUST 2023 -** Implementation of Omni Maths in the Arduino code, Running of Bot using PS3.

# OUR CODES

https://github.com/Omramanuj/OmniDRIVE-with-WEB?fbclid=IwAR1xMqx1mDTF2AhdbKfOlFuccGD9CoutdSwfAD42wkzIK1AAkjM65IKbCts

# REFERENCES

- ESP32 PINOUT REFERENCE
- PROTEUS INSTALLATION
- HTML & CSS TUTORIAL
- ARDUINO SETTING UP TUTORIAL: Arduino Tutorial 1: Setting Up and Programming the Arduino for Absolute Beginners WORKING OF OMNI DRIVE: ESP32-based omnidirectional robot without a camera
  YOUTUBE CHANNEL: maker.moekoe
- INSTALLING ESP32 IN ARDUINO IDE
- ESP32 ASYNC WEB SERVER CONTROLS
- GITHUB ESPASYNC WEB SERVER LIBRARY
- GITHUB ESPASYNC TCP LIBRARY
- OMNIDIRECTIONAL WHEELED ROBOTICS LECTURE
- W3SCHOOLS
- SERIAL COMMUNICATION: Serial - Arduino Reference
- UART:  A Hardware Communication Protocol Understanding Universal Asynchronous Receiver/Transmitter | Analog Devices
- SERIAL COMMUNICATION BETWEEN TWO ESP'S: ESP32 UART Communication Pins Explained with Example (microcontrollerslab.com)
- SERIAL COMMUNICATION TWO ARDUINO: Serial communication between two Arduino boards (electroniclinic.com)