Manipal University Jaipur

B.Tech. Program Second Year

PROJECT BASED LAB

COURSE CODE: EC2270

[E.C.E]

PROJECT REPORT ON

"SPY BOT CAR"

UNDER TAKEN BY

ADITYA SHARMA (209202224)

SIMRAN KUMARI (209202222)

HARSHIL SUTHAR (209202093)

UNDER THE GUIDANCE OF

Mr. MOHIT KUMAR SHARMA

Faculty of Engineering, Manipal University Jaipur

(Rajasthan)

Department of Engineering School of Electronics & Communication, Manipal University Jaipur, India

MAY, 2022

Manipal University Jaipur

Certificate

This is to certify that the project titled “SPY BOT CAR” is a record of the bona fide work done by “Aditya Sharma” (Registration - 209202224), Simran Kumari (Registration – 209202222), Harshil Suthar (Registration -209202093) submitted for the partial fulfilment of the requirements for the completion of the Project Based Lab (EC2270) course in the Department of Engineering of Electronics & Communication Engineering in Manipal University Jaipur, during the academic session Feb-June 2022.

Signature of the mentor

Name of the Mentor: Mr. Mohit Kumar Sharma

Department of Electronics & Communication Engineering

Signature of the HOD

Name of the HOD: Dr. Manish Tiwari

Department of Electronics & Communication Engineering

Manipal University
Jaipur

ACKNOWLEDGEMENT

It is a great privilege for us to express our profound gratitude to our respected teacher and our guide MR, Mohit Kumar Sharma, DEPT. OF ELECTRONICS & COMMUNICATION ENGINEERING, MANIPAL UNIVERSITY, JAIPUR for his constant guidance, valuable suggestions, supervision and inspiration throughout the course work without which it would have been difficult to complete the work within scheduled time. We would like to express our gratitude towards our other faculty members for his/her kind cooperation and encouragement which helped me in completion of this project. We are also indebted to the Head of the Department, Electronics & Communication Engineering, MANIPAL UNIVERSITY for permitting us to pursue the project. We would like to take this opportunity to thank all the respected teachers of this department for being a perennial source of inspiration and showing the right path at the time of necessity.

-ADITYA SHARMA (209202224)

-HARSHIL SUTHAR(209202093)

-SIMRAN KUMARI(209202222)

CONTENT

ABSTRACT:

Our aim is to design a SPY BOT CAR. As, a way to, develop a car using ESP32 module for surveillance and spying purposes.

This paper is focused in presenting the embedded into a surveillance bot car.

KEYWORDS:

Arduino UNO, ESP32 Bluetooth module with wi-fi, DC Motors, Wires, 5-12 V DC Battery, 4 Wheels, L289 Motor Driver, Bread-Board.

WORKING PRINCIPLE:

The Concept behind the "**Surveillance Bot Car"** is very simple. We will use a "**Arduino Uno**" to upload the codes into ESP32 module. Four DC Motor will be used to convert the direct current into mechanical energy for the wheels to rotate. L289 motor driver is used to control the dc motors. The ESP32 is used for the Video Surveillance and Wi-fi Connectivity.

INTRODUCTION:

This project titled “Surveillance Bot Car” is a modern approach to the conventional water dispenser that we use in our daily life. Water pumps are very useful in our daily life. Automatic Water Dispenser is containing a series of many functions like controlling the water level, showing the value of TDS and automatic ejection of water.

In day to day life intelligent systems are used in a wide range and these are embedded in design. We use some physical elements to perform certain tasks in our daily life. This paper is focused in presenting the embedded into an Automatic Water Dispenser. The thing by which we get motivated is the wastage of water and the impurity of water. We also know that it will help the environment and water cycle by which we can save water for our future.

So, in this project we will build a Automatic Water Dispenser using Atmel 328 that can automatically give you water when a glass is placed near it.

As, it’s known about the Microcontroller (Atmega328p Arduino Uno Board) which is used as the controller to control the automatic operation of automated liquid dispenser machine. Microcontroller is selected as the controller because it is easier to implement and the compact size makes it easier to mount it on the system. The machine is also easy to operate and user friendly, where simple steps are needed to operate the machine. In this project microcontroller is used as the core of the system.[1]

TECHNOLOGIES USED:

- Arduino Uno r3 board

- L298 Motor Driver
- C++ Programming
- ESP32 CAM module

METHODOLOGY

The methodology of the project is-

- An Arduino Uno R3 is used as open-source platform for the use of microcontroller ATmega328P based on the AVR enhanced RISC architecture.
- Now, ESP32 CAM module is used for video streaming which we will connect through Arduino for the code transfer process.
- Now, we open the ARDUINO IDE software in the computer and write the necessary program in the simulator.
- Now we upload the program into Arduino and click on the "VERIFY" key of the simulator.
- Now, ESP32 module is connected with L289n motor driver which will control the movement of DC motors. Hence the Wheels.

CIRCUITS & DESCRIPTION-

In this project, the online platform "Tinkercad" will be used for the simulation of the dispenser model. It consists of the necessary components used in this project such as Arduino circuit, board, wires, sensors etc.

- **REQUIREMENTS-**

1. **Arduino UNO R3**

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board. The Arduino platform has become quite popular with people just starting out with electronics, and for good reason.

The Arduino is a microcontroller board based on the ATmega8. It has 14 digital -input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter. Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode. Revision of the board has the following new features:

• 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible with both the board that uses the AVR, which operates with 5V and with the Arduino Due that operates with 3.3V. The second one is a not connected pin, that is reserved for future purposes.[4]

- Stronger RESET circuit.
- ATmega 16U2 replace the 8U2

.

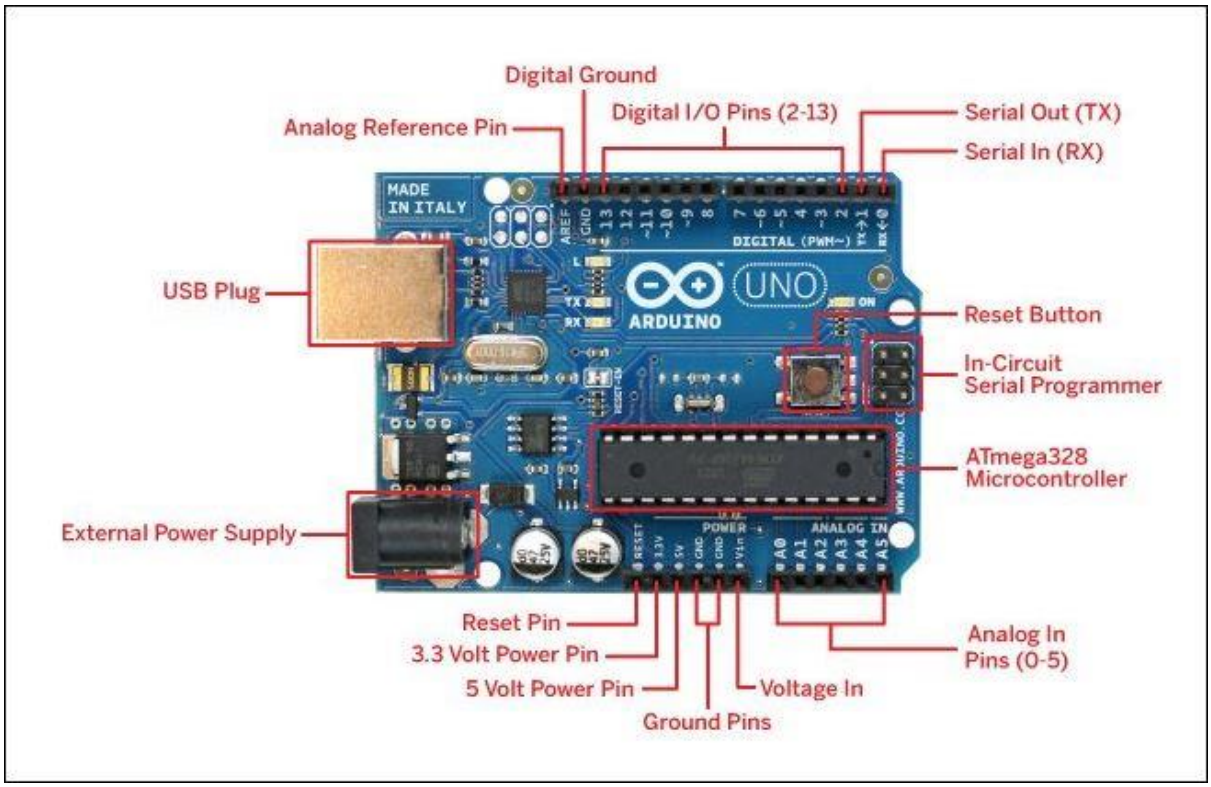


Fig. – Arduino UNO Board

Parameters of Arduino and the descriptions:

Microcontroller - ATmega328
Operating voltage - 5V.
Input voltage (recommended) - 7-12V.
Input voltage (limits) - 6-20V.
Digital I/O pins - 14 (D0-D13 are digital i/o; D3,D5,D6,D9,D10,D11 are PWM).
Analog I/P pins - 6.
Flash memory - 32KB (ATmega328).
SRAM - 2 KB (ATmega328).
EEPROM - 1KB (ATmega328).
Clock speed - 1.6 MHz

Length - 68.6 mm.
Width - 53.4 mm.
Weight - 25 g.

Power:

- The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.
- External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.
- The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.
- The power pins are as follows:
- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- **3V3.** A 3.3volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.[4]

INPUT & OUTPUT (Arduino)

Each of the 14 digital pins on the Uno can be used as an input or output, using pin Mode, digital Write, and digital Read functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 k Ohms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attach Interrupt function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analog Write () function.

- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the SPI library.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labelled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analog function. Additionally, some pins have specialized functionality:

- **TWI: A4 or SDA pin and A5 or SCL pin.** Support TWI communication using the Wire library.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analog reference.

- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the mapping between Arduino pins and ATmega328 ports. The mapping for the Atmega8, 168, and 328 is identical.[4]

2. **DC MOTOR**:

A DC motor (Direct Current motor) is the most common type of motor. DC motors normally have just two leads, one positive and one negative. If you connect these two leads directly to a battery, the motor will rotate. If you switch the leads, the motor will rotate in the opposite direction.

.

Fig. DC Motor

- **SOFTWARE USED-**

As explained, we have used online circuit designing platform “Tinkercad”, due to the limitations of unavailability of circuits on other platforms.

- C++ programming language is used for the working of Arduino & ESP32 Cam with the help of tinkercad.

3. **L298 Motor driver:**

   The L298N Motor Driver is a controller that uses an H-Bridge to easily control the direction and speed of up to 2 DC motors.

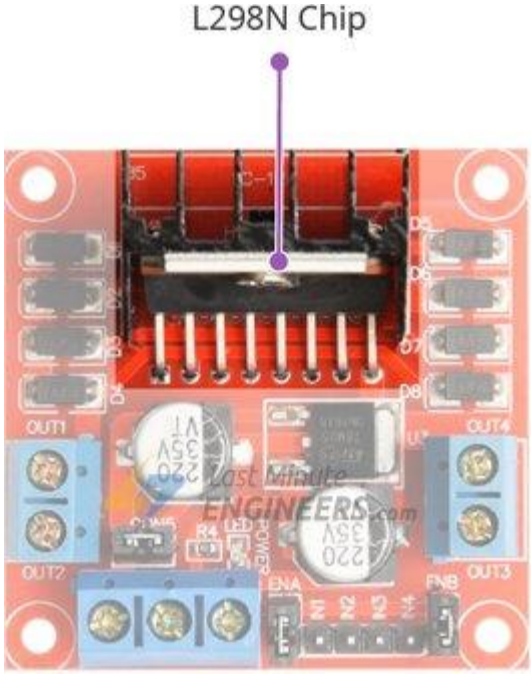
H-Bridge – For controlling rotation direction

The DC motor’s spinning direction can be controlled by changing polarity of its input voltage. A common technique for doing this is to use an H-Bridge.

An H-Bridge circuit contains four switches with the motor at the center forming an H-like arrangement.

Closing two particular switches at the same time reverses the polarity of the voltage applied to the motor. This causes change in spinning direction of the motor.

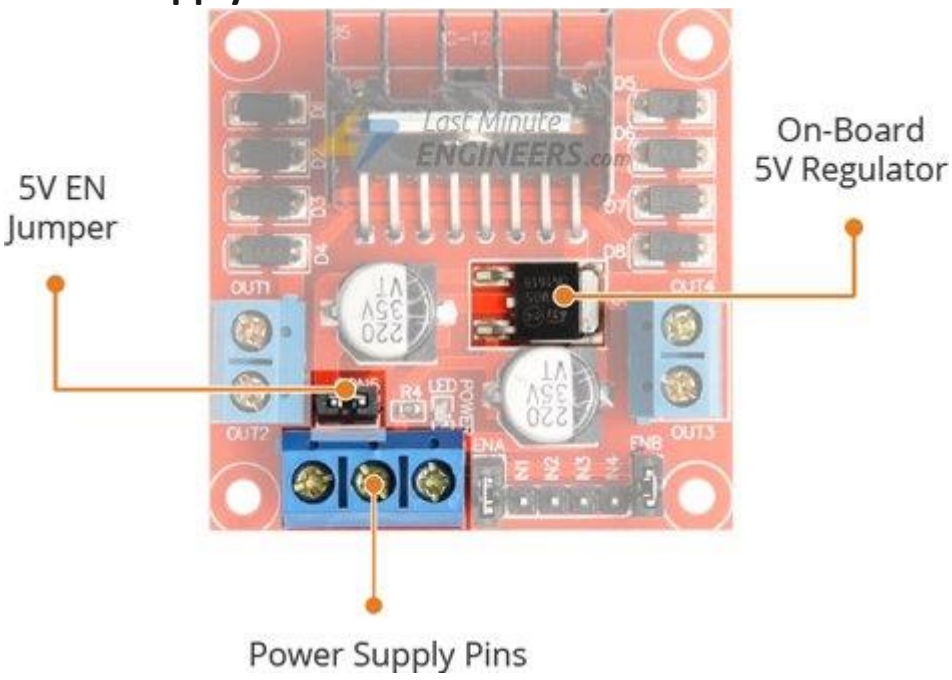
L298N Motor Driver IC



At the heart of the module is the big, black chip with chunky heat sink is an L298N.

The L298N is a dual-channel H-Bridge motor driver capable of driving a pair of DC motors. That means it can individually drive up to two motors making it ideal for building two-wheel robot platforms.

Power Supply



The L298N motor driver module is powered through 3-pin 3.5mm-pitch screw terminals. It consists of pins for motor power supply (Vss), ground and 5V logic power supply (Vss).

NOTE: - The L298N motor driver IC actually has two input power pins viz. 'Vss' and 'Vs'.From Vs pin the H-Bridge gets its power for driving the motors which can be 5 to 35V. Vss is used for driving the logic circuitry which can be 5 to 7V. And they both sink to a common ground named 'GND'.

4. ESP32

The development board equips the ESP-WROOM-32 module containing Tensilica Xtensa® Dual-Core 32-bit LX6 microprocessor. Operates at 80 to 240 MHz adjustable clock frequency and performs at up to 600 DMIPS (Dhrystone Million Instructions Per Second).

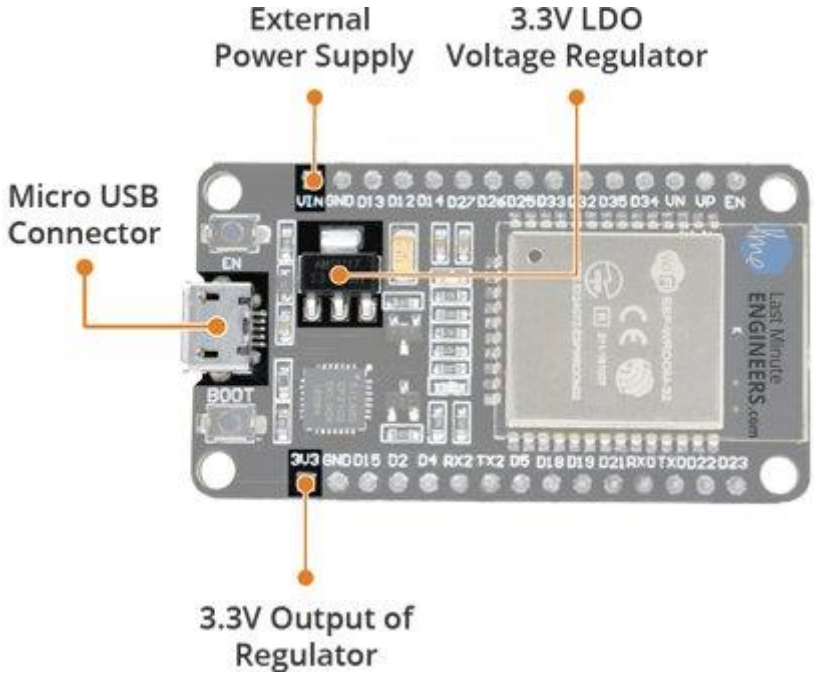
There's also 448 KB of ROM, 520 KB of SRAM and 4MB of Flash memory (for program and data storage) just enough to cope with the large strings that make up web pages, JSON/XML data, and everything we throw at IoT devices nowadays.

The ESP32 Integrates 802.11b/g/n HT40 Wi-Fi transceiver, so it can not only connect to a WiFi network and interact with the Internet, but it can also set up a network of its own, allowing other devices to connect directly to it. The ESP32 supports WiFi Direct as well, which is a good option for peer-to-peer connection without the need of an access point. The WiFi Direct is easier to setup and the data transfer speeds are much better than Bluetooth.

The chip also has dual mode Bluetooth capabilities, meaning it supports both Bluetooth 4.0 (BLE/Bluetooth Smart) and Bluetooth Classic (BT), making it even more versatile.

Power Requirement

As the operating voltage range of ESP32 is 2.2V to 3.6V, the board comes with a LDO voltage regulator to keep the voltage steady at 3.3V. It can reliably supply up to 600mA, which should be more than enough when ESP32 pulls as much as 250mA during RF transmissions. The output of the regulator is also broken out to one of the sides of the board and labeled as 3V3. This pin can be used to supply power to external components.



Power to the ESP32 development board is supplied via the on-board MicroB USB connector. Alternatively, if you have a regulated 5V voltage source, the VIN pin can be used to directly supply the ESP32 and its peripherals.

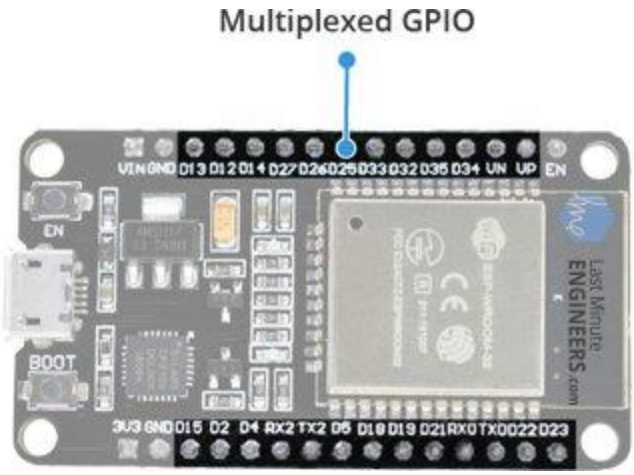
Also the sleep current of the ESP32 chip is less than 5 µA, making it suitable for battery powered and wearable electronics applications.

Peripherals and I/O

Although the ESP32 has total 48 GPIO pins, only 25 of them are broken out to the pin headers on both sides of the development

board. These pins can be assigned to all sorts of peripheral duties, including:

- 15 ADC channels – 15 channels of 12-bit SAR ADC's. The ADC range can be set, in firmware, to either 0-1V, 0-1.4V, 0-2V, or 0-4V
- 2 UART interfaces – 2 UART interfaces. One is used to load code serially. They feature flow control, and support IrDA too!
- 25 PWM outputs – 25 channels of PWM pins for dimming LEDs or controlling motors.
- 2 DAC channels – 8-bit DACs to produce true analog voltages.
- SPI, I2C & I2S interface – There are 3 SPI and 1 I2C interfaces to hook up all sorts of sensors and peripherals, plus two I2S interfaces if you want to add sound to your project.
- 9 Touch Pads – 9 GPIOs feature capacitive touch sensing.

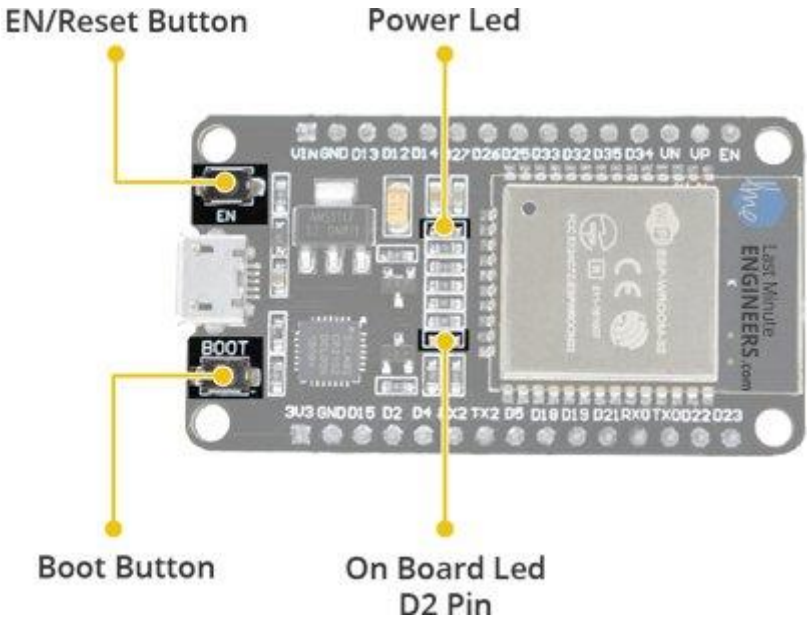


Thanks to the ESP32's pin multiplexing feature (Multiple peripherals multiplexed on a single GPIO pin). Meaning a single GPIO pin can act as an ADC input/DAC output/Touch pad.

On-board Switches & LED Indicators

The ESP32 development board features two buttons. One marked as EN located on the top left corner is the Reset button, used of

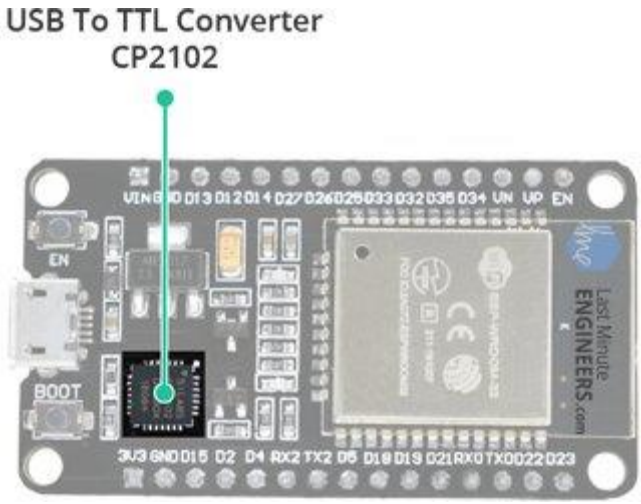
course to reset the ESP32 chip. The other Boot button on the bottom left corner is the download button used while downloading the new sketch/programs.



The board also has 2 LED indicators viz. Red LED & Blue LED. A Red LED indicates that the board is powered up and has 3.3V from the regulator. The Blue LED is user programmable and is connected to the D2 pin of the board.

Serial Communication

The board includes CP2102 USB-to-UART Bridge Controller from Silicon Labs, which converts USB signal to serial and allows your computer to program and communicate with the ESP32 chip.



ESP32 Development Board Pinout

The ESP32 development board has total 30 pins that interface it to the outside world. The connections are as follows:

There are two power pins viz. VIN pin & 3.3V pin. The VIN pin can be used to directly supply the ESP32 and its peripherals, if you have a regulated 5V voltage source. The 3.3V pin is the output of an on-board voltage regulator. This pin can be used to supply power to external components.

GROUND is a ground pin of ESP32 development board.

Arduino Pins are nothing but ESP32's hardware I2C and SPI pins to hook up all sorts of sensors and peripherals in your project.

GPIO Pins ESP32 development board has 25 GPIO pins which can be assigned to various functions programmatically. Each digital enabled GPIO can be configured to internal pull-up or pull-down, or set to high impedance. When configured as an input, it can also be set to edge-trigger or level-trigger to generate CPU interrupts.

ADC Channels The board integrates 12-bit SAR ADCs and supports measurements on 15 channels (analog enabled pins). Some of these pins can be used to build a programmable gain amplifier which is used for the measurement of small analog signals. The ESP32 is also designed to measure the voltages while operating in the sleep mode.

DAC Channels The board features two 8-bit DAC channels to convert digital signals into true analog voltages. This dual DAC can drive other circuits.

Touch Pads The board offers 9 capacitive sensing GPIOs which detect capacitive variations introduced by the GPIO's direct contact or close proximity with a finger or other objects.

UART Pins ESP32 development board has 2 UART interfaces, i.e. UART0 and UART2, which provide asynchronous communication (RS232 and RS485) and IrDA support, and communicate at up to 5 Mbps. UART provides hardware management of the CTS and RTS signals and software flow control (XON and XOFF) as well.

SPI Pins SPI Pins ESP32 features three SPIs (SPI, HSPI and VSPI) in slave and master modes. These SPIs also support the following general-purpose SPI features:

- 4 timing modes of the SPI format transfer
- Up to 80 MHz and the divided clocks of 80 MHz
- Up to 64-Byte FIFO

All SPIs can also be used to connect to the external Flash/SRAM and LCD.

~ PWM Pins The board has 25 channels (Nearly All GPIO pins) of PWM pins controlled by Pulse Width Modulation (PWM) controller. The PWM output can be used for driving digital motors and LEDs. The controller consists of PWM timers and the PWM operator. Each

timer provides timing in synchronous or independent form, and each PWM operator generates the waveform for one PWM channel.

EN Pin is used to enable ESP32. The chip is enabled when pulled HIGH. When pulled LOW the chip works at minimum power.

SOFTWARE USED-

As explained, we have used online circuit designing platform "Tinkercad", due to the limitations of unavailability of circuits on other platforms.

- C++ programming language is used for the working of Arduino & ESP32 Cam with the help of tinkercad.

CIRCUIT DIAGRAM-

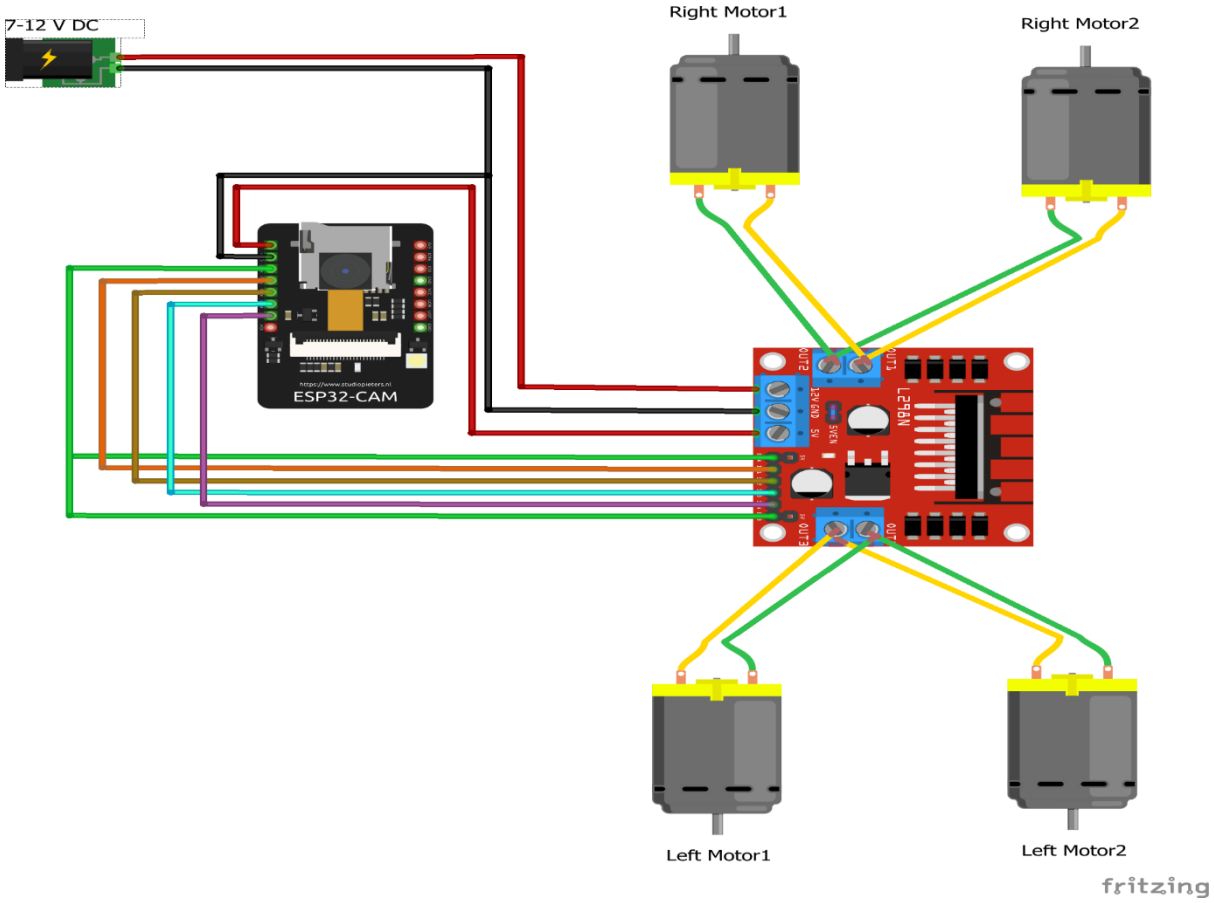
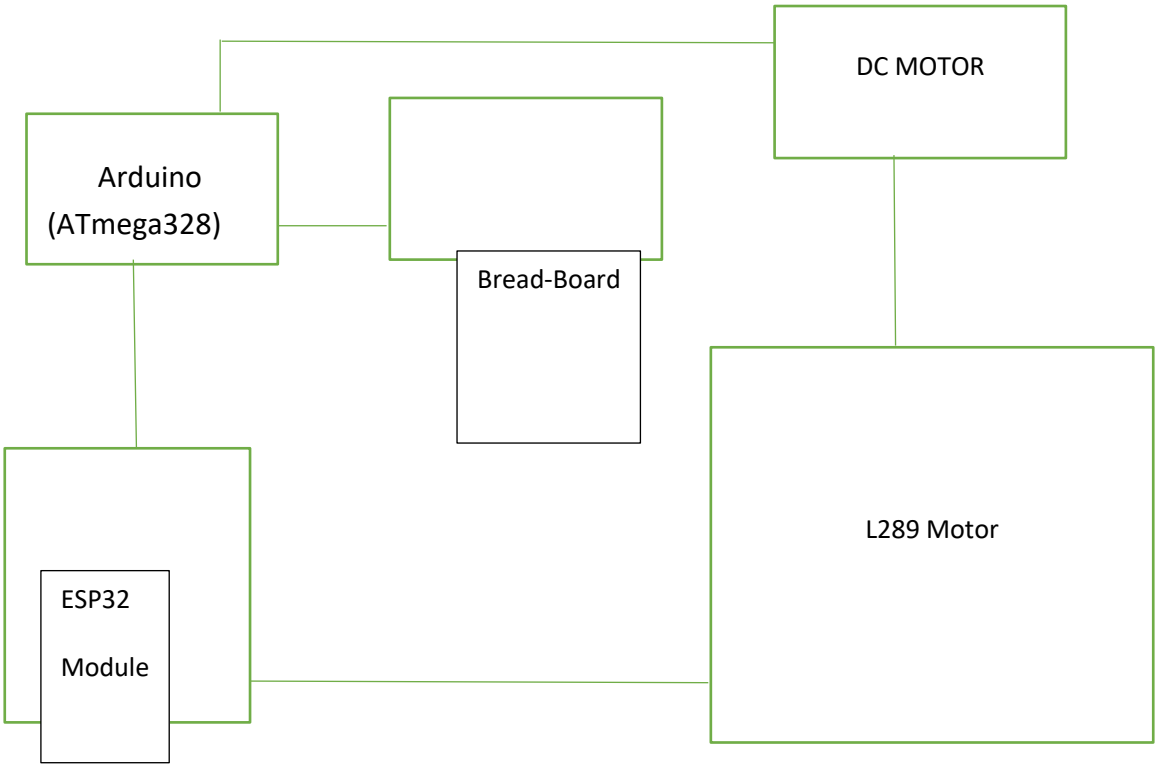


Fig. Circuit Diagram of Bot Car

IMPLEMENTATION



RESULTS & CONCLUSIONS:

FUTURE SCOPES:

There is no limitation for improvisation, different ideas can reflect the changes in the way it is implemented, and collaboration of technology at interdisciplinary level of engineering improves the quality of the product potentially. The technologies today are more powerful, and to make this module much more qualified it is built even to support any further expansion and to make it work more efficiently. Servers which can control multiple Bot/drone systems can improve the reliability of this prototype.

LIMITATIONS:

Regarding this project, the implementation of the circuit has no limitations. The only limitation may occur when we implement it on a large scale.

REFERENCES:

[1] https://docs.arduino.cc/hardware/uno-rev3/

[2] https://components101.com/modules/l293n-motor-driver-module

[3] https://randomnerdtutorials.com/esp32-cam-video-streaming-face-recognition-arduino-ide/

[4] https://en.wikipedia.org/wiki/DC_motor