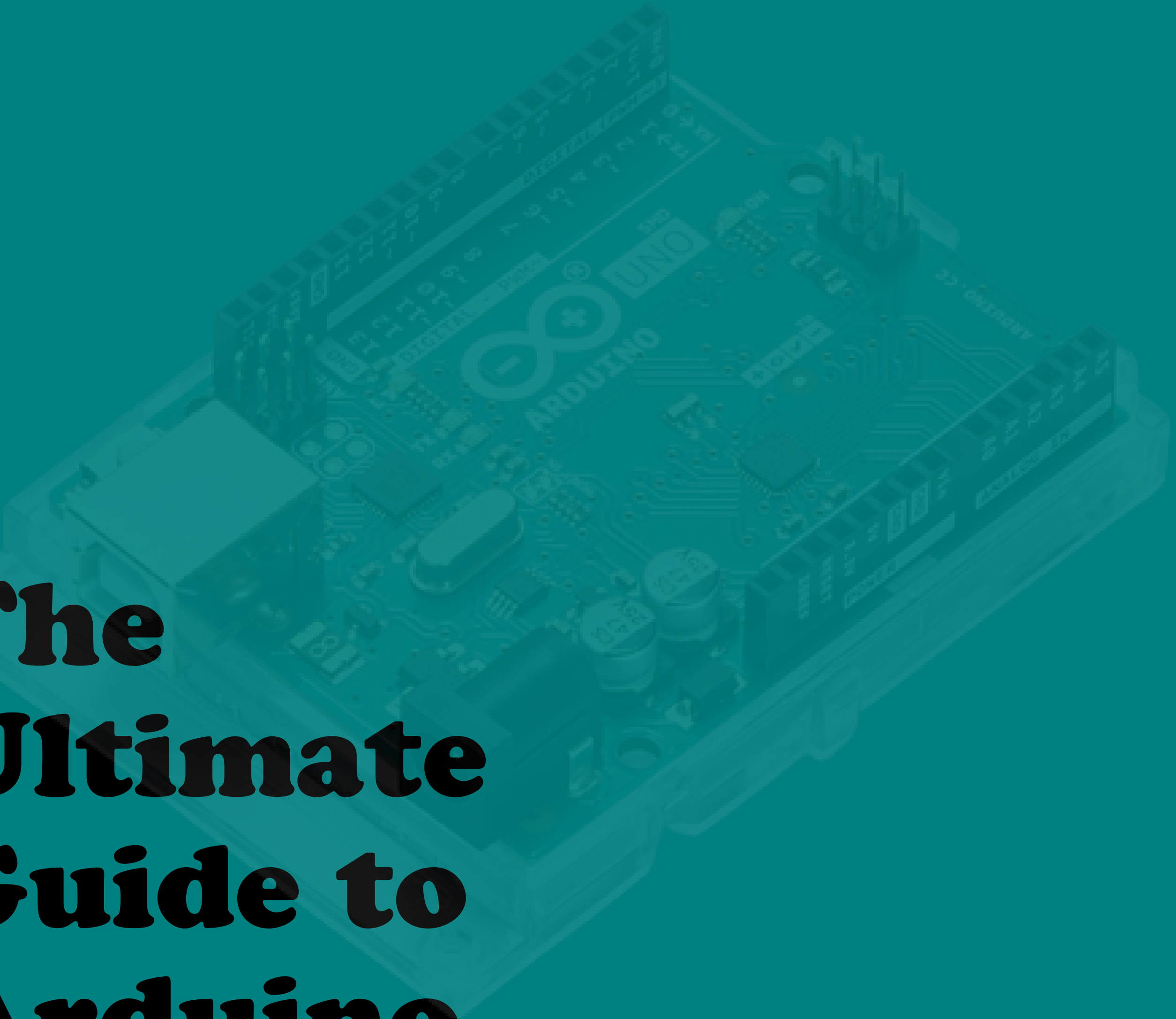
The Ultimate Guide to Arduino and ESP32 Automation



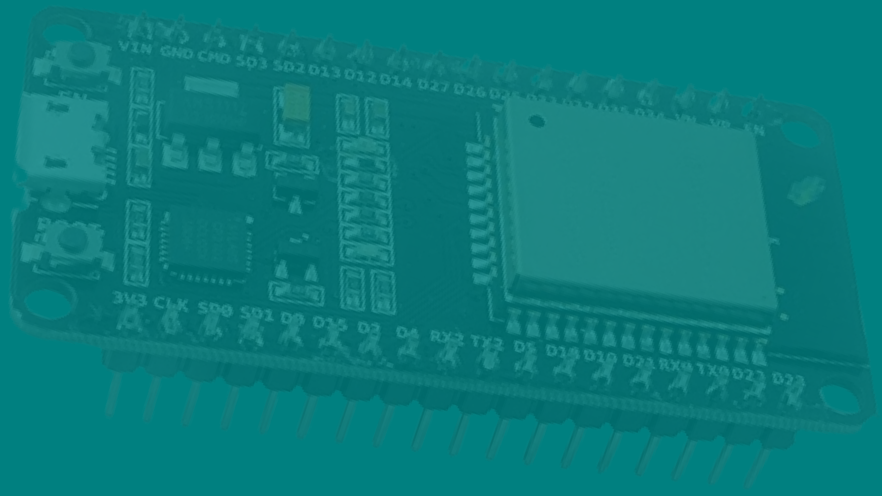

VIN GND CMD SD3 SD2 D13 D12 D14 D27 D26
3V3 CLK SD0 SD1 D15 D2 D4 RX2 TX2 D5 D18 D19 D21 RX0 TX0 D22 D23

TABLE OF CONTENTS

ARDUINO AUTOMATION

To get introduced and start to performing systems so at the end of this guide the user is able to perform their own projects and automation systems, this guide will show how the most used components works, by performing a test for each of them, and at the end, after learning all about these components, a final automation project will be done.

For each test, there will be a definition of what the test will do, then a list of the components will be shown (with the explanation for each one), the connection of the whole system with detailed explanation and diagram and finally finishes with the code of the system.

Besides that, each system has a specific code that may have different features, so to learn how to code in Arduino, it is important to read the comments about what each part of the code is doing (the comments will be in the code for each test).

DC MOTOR CONTROL

This test has the objective to control the rotational direction of a DC Motor, using a H-bridge. In the circuit, the motor will rotate to one direction, wait 3 seconds, rotate to the other direction and finally will stop for another 3 seconds, then it all repeats from the beginning.

- **Components Used**

Quantity	Components
1	Arduino Uno
1	DC Motor
1	H - Bridge L298n
6	Jumper Wires (4 male - male and 2 male - female)

Table 8: Components Used to Control a DC Motor

- **H-Bridge L298n Functionality**

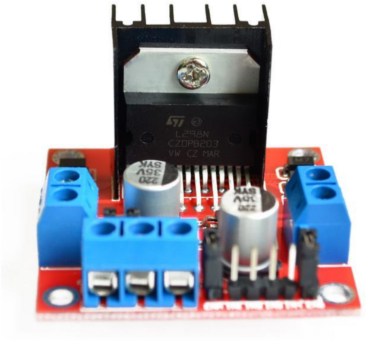


Image 6: H-Bridge L298n

The H-bridge shield is an integrated circuit board and this one specifically (L298n) was created for use in Arduino. It is capable of controlling tension, and makes the communication between the controller and the motor. So, it is basically a component that is used to control the rotational direction and speed of a DC motor.

A H-Bridge switches the polarity of the motor (this one is able to control 2 motors at the same time), and for each motor it should have 2 pins to control the output. If one pin is HIGH and the other one is LOW the motor should spin and by altering this order, the motor will change it's direction. Pulling both pins to LOW or HIGH, will stop the motor.

Observation: LOW and HIGH are part of the programming code for the system and will be seen in the next steps.

The diagram below is the representative sketch for each pin and port of the H-bridge that was explained. The pins might change for different H-bridges, but the principle is the same.

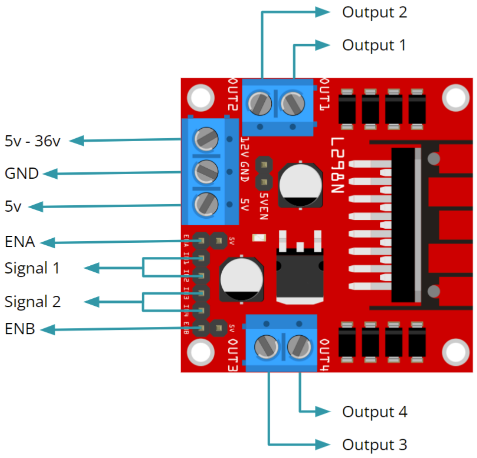


Diagram 10: H-Bridge L298n Functionality

H-Bridge Pins	Definition
Out 1 and Out 2	The are the pins that are connected to the motor and each pair is able to control 1 motor or both pairs can be used together to control 1 stepper motor
Out 3 and Out 4	
12v	Voltage pin the is able to support from 5v until 36v
5v	5 volts pin that usually is used to connect to the Arduino
GND	That is the ground pin of the H - Bridge
ENA	Pins used to control the speed of the motor, ENA is used to control the speed of Out 1 and Out 2; ENB is used to control speed of Out 3 and Out 4
ENB	
Signal 1 (IN1 + IN2)	Used to control the rotation direction of motor from the Out 1 and 2
Signal 2 (IN3 + IN4)	Used to control the rotation direction of motor from the Out 1 and 2

Table 9: Definition of Each H-Bridge L298n Pin

➢ **DC Motor Functionality**

Image 7: DC Motor 12v

A DC motor is a motor activated by a continuous power supply current that converts electrical energy into mechanical energy. This rotary electric motor is capable of varying its rotational speed, by altering its current intensity or by applying a variable power supply. Besides that, they also have the possibility to alternate the direction of the current (making it possible to choose the direction of rotation).

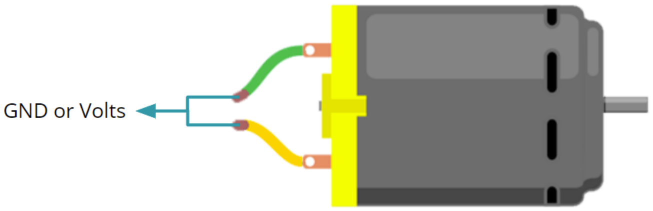


Diagram 11: DC Motor Functionality

Besides that, it doesn't need any specific connection, the pins to connect to the GND and Volts can be chosen, but depending on this connection, it will determine which is the direction that the motor will rotate.

- **Connection of the System**

H-Bridge Pins	Connection
12v	5v of Arduino
GND	GND of Arduino
IN1	Digital Pin 6 of Arduino
IN2	Digital Pin 7 of Arduino

Out 1	DC Motor
Out 2	DC Motor

Table 10: Connection of Each H-Bridge Pin

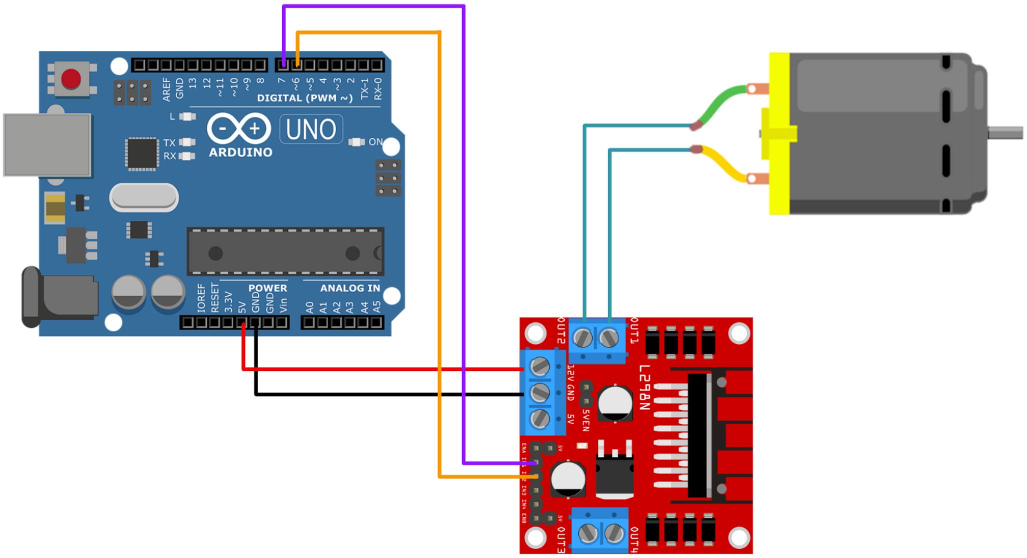


Diagram 12: System Connection For The Control of a DC Motor Using an H-Bridge

➢ **Code**

```
int IN1 = 6;  //define command port
int IN2 = 7;  //define command port

void setup(){
pinMode(IN1, OUTPUT); //set pin IN1 as Output
pinMode(IN2, OUTPUT); //set pin IN2 as Output
}

void loop(){
  //make the motor rotates to one direction
digitalWrite(IN1, LOW);
digitalWrite(IN2, HIGH);
delay(3000); //waits 3 seconds

 //change the direction of the motor
digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW);
delay(3000); //waits 3 seconds

 //stops the motor
 digitalWrite(IN1, LOW);
 digitalWrite(IN2, LOW);
 delay(3000);
}
```

ESP

➢ Description

The ESP board is one of the best options available to perform ioT projects, but what does this mean? Well, first of all, ioT must be explained and it is basically a network of physical objects that are incorporated to softwares or sensors, that interact with one another by the internet (in other words, it makes it possible to control devices with a smartphone, a computer, an ipad…).

The principle that was given in the Arduino tests, can be done very similarly with the ESP board, but with this board, it opens doors to improve the system that was previously made. In this way, it can be controlled by a certain distance using the principle of ioT, because these boards already have WiFi and bluetooth sensors integrated to it, so it can be connected to the network and be used to make that place smarter and autonomous.

➢ Operation (ESP 32)

The ESP32 is an improvement of the ESP8266, because it is integrated with a new system that makes it able to perform the control of devices (ioT way) easier, compact and more precise.

In total, the ESP 32 has 38 ports, that includes the power pins, analog and serial.

From the 38 ports, 22 can be used as it is an Arduino port that can work as a digital pin, an analog pin or be used for both (GPIO pins) that means they are general purpose input output (can be used as input or output pins).

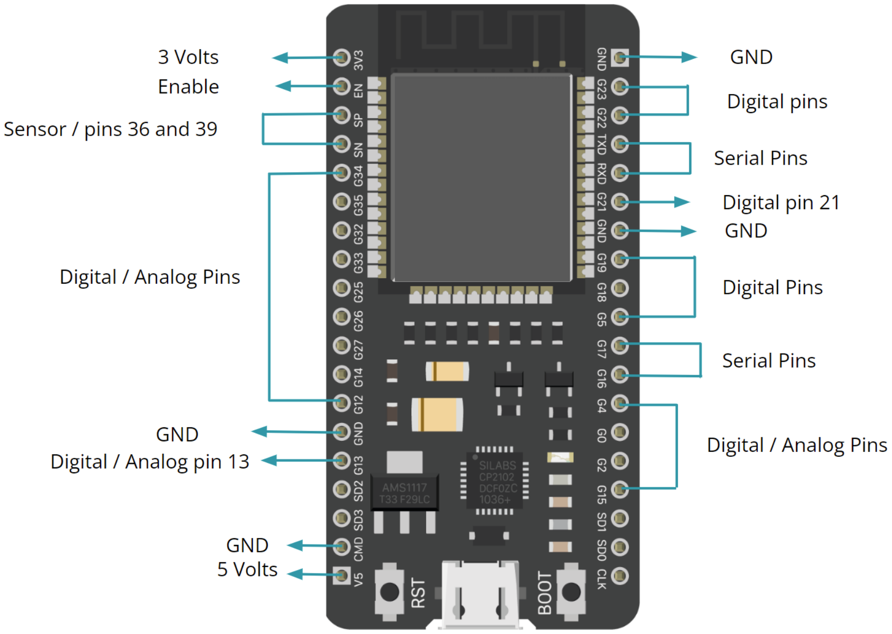


Diagram 43: ESP32 Functionality

Observation: Usually when working with ESP32, some errors might occur when uploading the code to the board and the usual ones are that a data cable is necessary (the same way as the Arduino) to connect the esp to the computer.

Another one, is that before uploading the code, the button BOOT needs to be pressed and released when the message “upload done” appears in the Arduino IDE, or always when uploading a code, some error might occur. Another way to fix this problem and not have to hold the button BOOT, is to put a 10uF capacitor between the ports EN and GND of the Esp (it can be done by soldering the capacitor between these two ports or using a protoboard).