# ESP32 Dual Motor Interface using TB6612FNG

This document gives a connection diagram of ESP32 with two motors via TB6612FNG chip (a driver IC for DC motors) and ESP32 sample code. Two input signals, IN1 and IN2, choose one of four modes: CW, CCW, short-brake, and stop mode for each motor. In addition, the speed of each motor can be controlled via separate PWM pins. Of course, you can write your code from scratch to control the speed and direction of motors. However, a worked-out example has been provided in this document that uses a library to do so. This example is for Arduino IDE, but it can also work on Visual Studio Code.

## Installing Required Library

In the example below, a library named Robojax ESP32 DC Motor has been added to IDE. It can be done in the following steps:

1. Download the zip file from the [link](http://robojax.com/sites/default/files/zip-files/robojax_ESP-L298N-DC-Motor_library.zip).
2. Add this zip file by clicking Sketch > Include Library > Add .ZIP Library and choosing the downloaded zip file, as shown below:

Graphical user interface, text, application

Description automatically generated

1. Restart the IDE.
2. Make sure that the Library is correctly added by checking the availability of the following examples:

Graphical user interface

Description automatically generated

## Connections and Pin Diagram

You need only six control signals from ESP32 to TB6612FNG Chip; three for each motor. Moreover, power and ground signals are also provided from the power rail of the ESP32. However, you can always separate the source of to establish isolation between motors and remaining circuitry. The standby is an active low input, i.e., to take your chip to low current standby mode, you need to provide it 0 V.

The detailed specification and functionality of each pin of TB6612FNG Chip can be accessed [here](https://www.sparkfun.com/datasheets/Robotics/TB6612FNG.pdf).

For ESP32, make sure you identify the correct pins

Diagram, schematic

Description automatically generatedNEW PinAssign

PWMA ->

Pin D8 = GPIO17

AIN2 ->

Pin D9 = GPIO16

AIN1 ->

Pin D10 = GPIO14

BIN1 ->

Pin D11 = GPIO4

BIN2 ->

Pin D12 = GPIO15

PWMB –>

Pin D13 = GPIO2

To set bootloader mode to “download”, force

Force GPI2 (Pin 13) low ??

## Sample Code

The following code would move motors for three seconds in different modes as mentioned in the comments of the code given below, given that motors are correctly interfaced with the motor driver IC.

#include <Robojax\_L298N\_DC\_motor.h>

// motor 1 settings

#define CHA 0

#define ENA 19 // this pin must be PWM enabled pin if Arduino board is used

#define IN1 18

#define IN2 5

// motor 2 settings

#define IN3 17

#define IN4 16

#define ENB 4// this pin must be PWM enabled pin if Arduino board is used

#define CHB 1

const int CCW = 2; // do not change

const int CW = 1; // do not change

#define motor1 1 // do not change

#define motor2 2 // do not change

// for two motors without debug information // Watch video instruciton for this line: https://youtu.be/2JTMqURJTwg

Robojax\_L298N\_DC\_motor robot(IN1, IN2, ENA, CHA, IN3, IN4, ENB, CHB);

// for two motors with debug information

//Robojax\_L298N\_DC\_motor robot(IN1, IN2, ENA, CHA, IN3, IN4, ENB, CHB, true);

void setup() {

Serial.begin(115200);

robot.begin();

//L298N DC Motor by Robojax.com

}

void loop() {

// move straight for 3 sec

robot.rotate(motor1, 70, CW);//run motor1 at 60% speed in CW direction

robot.rotate(motor2, 70, CCW);//run motor1 at 60% speed in CW direction

delay(3000);

robot.brake(1);

robot.brake(2);

delay(3000);

// rotate for 3 sec

robot.rotate(motor1, 100, CW);//run motor1 at 60% speed in CW direction

robot.rotate(motor2, 100, CW);//run motor1 at 60% speed in CW direction

delay(3000);

robot.brake(1);

robot.brake(2);

delay(3000);

// move straight for 3 sec

robot.rotate(motor1, 70, CW);//run motor1 at 60% speed in CW direction

robot.rotate(motor2, 70, CCW);//run motor1 at 60% speed in CW direction

delay(3000);

robot.brake(1);

robot.brake(2);

delay(3000);

// rotate for 3 sec in opposite ditection

robot.rotate(motor1, 100, CCW);//run motor1 at 60% speed in CW direction

robot.rotate(motor2, 100, CCW);//run motor1 at 60% speed in CW direction

delay(3000);

robot.brake(1);

robot.brake(2);

delay(2000);

for(int i=0; i<=100; i++)

{

robot.rotate(motor1, i, CW);// turn motor1 with i% speed in CW direction (whatever is i)

robot.rotate(motor2, i, CCW);// turn motor1 with i% speed in CW direction (whatever is i)

delay(100);

}

delay(2000);

robot.brake(1);

robot.brake(2);

delay(2000);

}