**GOVERMANT POLYTECHNIC NAGPUR**

(AUTONOMOUS INSTITUTE OF MAHARASHTRA)

**Project Report**

Name of Project: **Obstacle Avoiding Robot**

**Submitted By**

Name of student: **1**. Ganesh Narayan Moroliya 1904038

**2**. Yugal Krushnaji Nasare 1904042

**3**. Atharva Ramesh Navghare 1904043

**4**. Chirag Hitesh Pabari 1904044

**5**. Pawan Dilip Sorte 1904058

Guided By : *Prof.* **Arshi Anjum**



**DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING**

(Academic Session 2021-22)

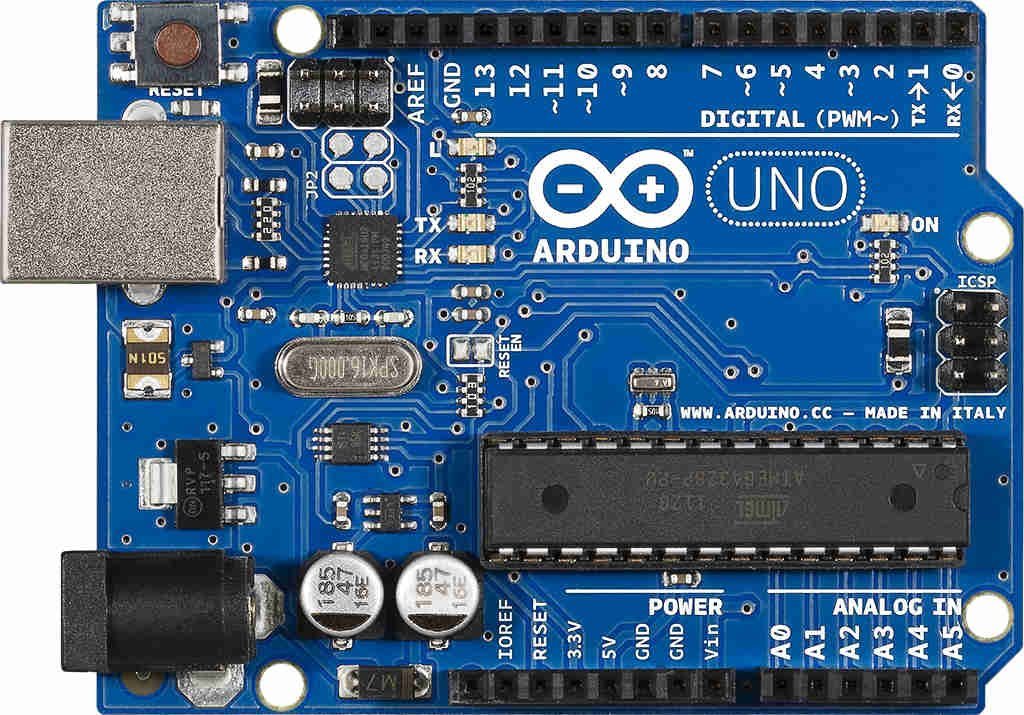
**Introduction of Obstacle Avoiding Robot**

1. Obstacle Avoiding Robot is an intelligent device that can automatically sense the obstacle in front of it and avoid them by turning itself in another direction.
2. This design allows the robot to navigate in an unknown environment by avoiding collisions, which is a primary requirement for any autonomous mobile robot.
3. Generally obstacle avoidance robot is design to allow robot to navigate in unknown environment by avoiding collisions.
4. And it’s senses obstacles in the path, avoids it and resumes its running.
5. Sometimes we have make use of sensors to achieve this objective.
6. The use of these methods front to classic methods (path planning) is a natural alternative when the scenario is dynamic with an unpredictable behaviour.
7. In these cases, the surroundings do not remain invariable, and thus the sensory information is used to detect the changes consequently adapting moving.

**Components Used**

1. Arduino UNO
2. BO Motor
3. Ultrasonic Sensor HC-SR04
4. L298N Motor Driver Module
5. Screw terminal connector

**Arduino UNO**



1. **The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.**
2. The board is equipped with sets of digital and analog input/output pins that may be interfaced to various expansion boards and other circuits.
3. The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable.
4. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 To 20 volts.
5. The ATmega328 on the board comes pre programmed with a bootloader that allows uploading new code to it without the use of an external hardware programmer.

* **GENERAL PIN FUNCTION**

1. **LED**: There is a built-in LED driven by digital pin 13.
2. **VIN**: The input voltage to the Arduino board when it is using an external power source.
3. **5V**: This pin outputs a regulated 5V from the regulator on the board.
4. **3V**: A 3volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
5. **GND**: Ground pins.
6. **IOREF**: This pin on the Arduino board provides the voltage reference with which the microcontroller operates.

* **SPECIAL PIN FUNCTION**

1. **Serial / UART**: pins 0 (RX) and 1 (TX), sed to receive (RX) and transmit (TX) TTL serial data.
2. **External interrupts**: pins 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.

1. **PWM (pulse-width modulation)**: pins 3, 5, 6, 9, 10, and 11, Can provide 8-bit PWM output with the analog write ().
2. **SPI (Serial Peripheral Interface)**: pins 10 (SS), 11 (MOSI), 12 (MISO), and 13 (SCK). These pins support SPI communication using the SPI library.
3. **TWI (two-wire interface)**: pin SDA (A4) and pin SCL (A5). Support TWI communication using the Wire library.
4. **AREF (analog reference)**: Reference voltage for the analog inputs

* **PROGRAMMING**

1. The Arduino Uno can be programmed with the (Arduino Software).
2. Select “Arduino Uno from the tools>Board menu (according to the microcontroller on your board).
3. The ATmega328 on the Arduino Uno comes programmed with a bootloader
4. that allows you to upload new code to it without the use of an external hardware programmer. It communicate using the original STK500 protocol.

**BO Motor**



1. Bo motor (Battery Operated) lightweight DC geared motor which gives good torque and rpm at lower voltages.
2. Here you can get BO motor with varying rated speed.
3. This motor can run at approximately 200 rpm when driven by a single Li-Ion cell.
4. It’s very Great for battery operated lightweight robots.

* **SPECIFICATIONS**
* Working Voltage 3-12V
* Load Speed: 200 rpm /- 10rpm
* Load Current: 125mA (max.170mA)

**Ultrasonic Sensor HC-SR04**

****

1. The HC-SR04 Ultrasonic Distance Sensor is a sensor used for detecting the distance to an object using sonar.
2. This economical sensor provides 2cm to 400cm of non-contact measurement functionality with a ranging accuracy that can reach up to 3mm.
3. Each HC-SR04 module includes an ultrasonic transmitter, a receiver and a control circuit.
4. There are only four pins that you need to worry about on the HC-SR04: VCC (Power), Trig (Trigger), Echo (Receive), and GND (Ground).
5. You will find this sensor very easy to set up and use for your next range-finding project.
6. This sensor has additional control circuitry that can prevent inconsistent "bouncy" data depending on the application.

**Specifications**

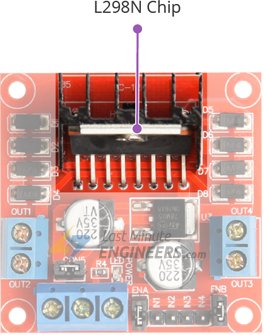
* Power Supply: DC 5V
* Working Current: 15mA
* Working Frequency: 40Hz
* Ranging Distance: 2cm – 400cm/4m
* Resolution: 0.3 cm
* Measuring Angle: 15 degree
* Trigger Input Pulse width: 10u

**L298N Motor Driver Module**



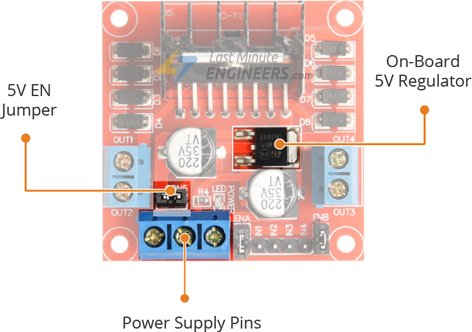
1. The L298N is a dual H-Bridge motor driver which allows speed and direction control of two DC motors at the same time.
2. The module can drive DC motors that have voltages between 5 and 35V, with a peak current up to 2A.

* **L298N Motor Driver IC:**



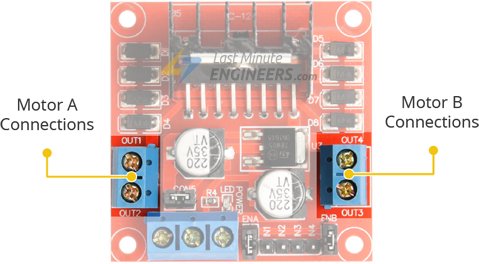
1. The L298N is a dual-channel H-Bridge motor driver capable of driving a pair of DC motors.
2. That means it can individually drive up to two motors making it ideal for building two-wheel robot platforms.

* **Power Supply**



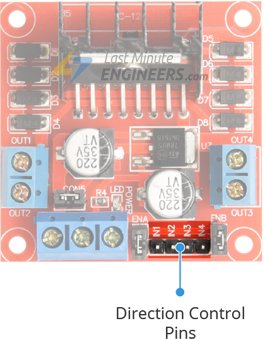
1. When this jumper is in place, the 5V regulator is enabled, supplying logic power supply (Vs) from the motor power supply (Vs).
2. In this case, 5V input terminal acts as an output pin and delivers 5V 0.5A.
3. You can use it to power up the Arduino or other circuitry that requires 5V power supply.

* **Output Pins**



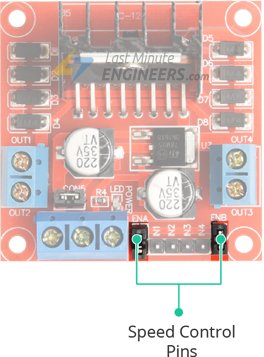
1. The L298N motor driver’s output channels for the motor A and B are broken out to the edge of the module with two 3.5mm-pitch screw terminals.
2. Each channel on the module can deliver up to 2A to the DC motor. However, the amount of current supplied to the motor depends on system’s power supply.

* **Direction Control Pins**



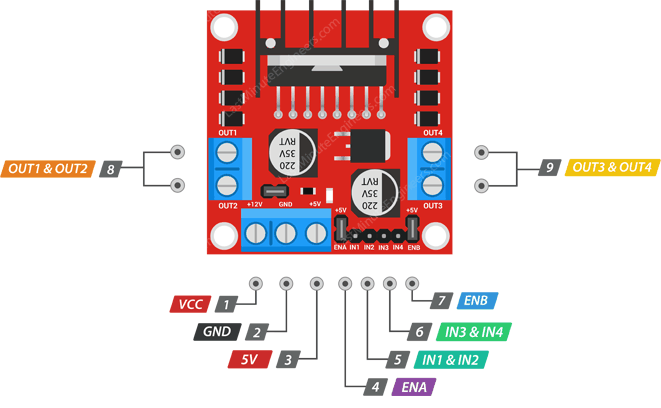
1. Using the direction control pins, we can control whether the motor spins forward or backward. These pins actually control the switches of the H-Bridge circuit inside L298N IC.
2. The module has two direction control pins for each channel. The **IN1** and **IN2** pins control the spinning direction of the motor A while **IN3** and **IN4** control motor B.

* **Speed Control Pins**



1. The speed control pins viz. **ENA** and **ENB** are used to turn the motors ON, OFF and control its speed.
2. The module usually comes with a jumper on these pins. When this jumper is in place, the motor is enabled and spins at maximum speed.
3. If you want to control the speed of motors programmatically, you need to remove the jumpers and connect them to PWM-enabled pins on Arduino.

**L298N Motor Driver Module Pinout**



VCC pin supplies power for the motor. It can be anywhere between 5 to 35V. Remember, if the 5V-EN jumper is in place, you need to supply 2 extra volts than motor’s actual voltage requirement, in order to get maximum speed out of your motor.

GND is a common ground pin.

5V pin supplies power for the switching logic circuitry inside L298N IC. If the 5V-EN jumper is in place, this pin acts as an output and can be used to power up your Arduino. If the 5V-EN jumper is removed, you need to connect it to the 5V pin on Arduino.

ENA pins are used to control speed of Motor A. Pulling this pin HIGH(Keeping the jumper in place) will make the Motor A spin, pulling it LOW will make the motor stop. Removing the jumper and connecting this pin to PWM input will let us control the speed of Motor A.

IN1 & IN2 pins are used to control spinning direction of Motor A. When one of them is HIGH and other is LOW, the Motor A will spin. If both the inputs are either HIGH or LOW the Motor A will stop.

IN3 & IN4 pins are used to control spinning direction of Motor B. When one of them is HIGH and other is LOW, the Motor B will spin. If both the inputs are either HIGH or LOW the Motor B will stop.

ENB pins are used to control speed of Motor B. Pulling this pin HIGH(Keeping the jumper in place) will make the Motor B spin, pulling it LOW will make the motor stop. Removing the jumper and connecting this pin to PWM input will let us control the speed of Motor B.

OUT1 & OUT2 pins are connected to Motor A.

OUT3 & OUT4 pins are connected to Motor B.

**Screw terminal connector**

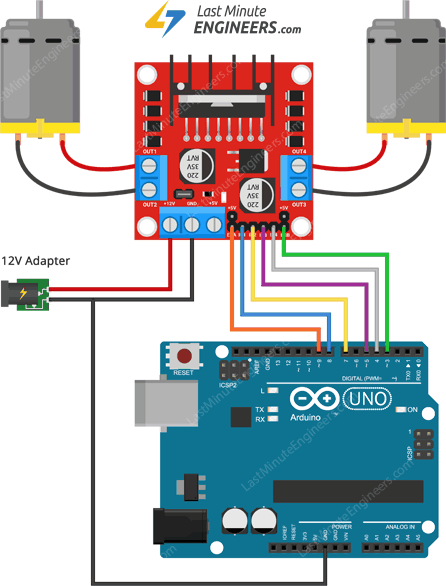


1. Screw terminal or screw type terminal blocks secure the wire against the conductor in the terminal block by tightening a screw which closes the clamp.
2. It is a type of electrical connection where a wire is held by the tightening of a screw.

**Specifications**

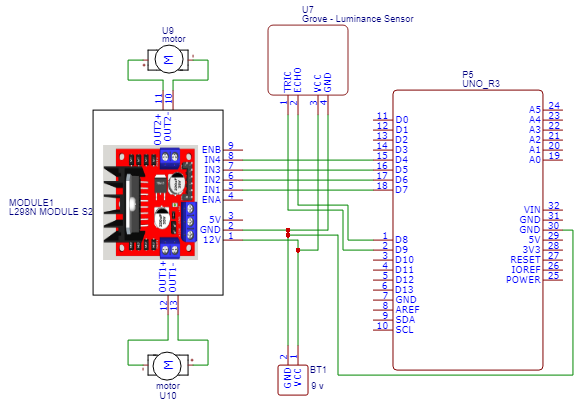
* Voltage rating: 125 V
* Contact resistance: 20 mΩ
* Wire strip length: 5 mm
* Spacing: 3.5

**Wiring L298N motor driver module To Arduino UNO**



1. Start by connecting power supply to the motors. In our experiment we are using DC Gearbox Motors (also known as ‘TT’ motors) that are usually found in two-wheel-drive robots.
2. They are rated for 3 to 12V. So, we will connect external 12V power supply to the VCC terminal.
3. Considering internal voltage drop of L298N IC, the motors will receive 10V and will spin at slightly lower RPM. But, that’s OK.
4. Next, we need to supply 5 Volts for the L298N’s logic circuitry. We will make use of the on-board 5V regulator and derive the 5 volts from the motor power supply so, keep the 5V-EN jumper in place.
5. Now, the input and enable pins (ENA, IN1, IN2, IN3, IN4 and ENB) of the L298N module are connected to six Arduino digital output pins(9, 8, 7, 5, 4 and 3). Note that the Arduino output pins 9 and 3 are both PWM-enabled.
6. Finally, connect one motor to terminal A(OUT1 & OUT2) and the other motor to terminal B(OUT3 & OUT4). You can interchange your motor’s connections, technically, there is no right or wrong way.

**Circuit Diagram and Working**

****

1. The obstacle avoidance robot uses ultrasonic sensors for its movements. A microcontroller is used to achieve the desired operation.
2. The motors are connected through the motor driver IC to the microcontroller and the ultrasonic sensor is attached in front of the robot.
3. Whenever the robot is going on the obstacle then ultrasonic sensor transmits the ultrasonic waves continuously from its sensor.
4. Then if obstacle comes ahead of it the ultrasonic waves are reflected from an object and that information is passed to the microcontroller.
5. The microcontroller controls the motors left, right, back, front, based on ultrasonic signals.

**Applications**

1. Obstacle avoiding robots can be used in almost all mobile robot navigation systems.
2. They can be used for household work like automatic vacuum cleaning.
3. They can also be used in dangerous environments, where human penetration could be fatal.

**Advantages and Disadvantage**

* **Advantages**

1. Whenever robot senses any obstacle automatically diverts its position to left or right and follows the path without human guidance.
2. The programming of the microcontroller is easy.
3. It is a low’ cost circuit.

* **Disadvantages**

1. It is time consuming project**.**
2. It is use for short distance only.
3. It is not recommended to keep the range very long because this would cause the ROBOT to keep moving forward and backward as it senses any obstacle, even far away from it.
4. It is not in human control.

**Result And Conclusion**

Thus we have successfully completed our mini project of Obstacle avoiding robot. It has 70% efficiency.

……….Our future work aims to acheive better efficiency.

**Reference**

<https://www.youtube.com/watch?v=KO1CaPIjt8M>

<https://easyeda.com/>

<https://www.electronicshub.org/obstacle-avoiding-robot-arduino/>