**Title: All Direction Robotic Car with L298N Motor Driver**

**INTRODUCTION:**

A Bluetooth controlled RC car is a car controlled by a remote control. The car is driven by an app imbedded in an Android mobile device. The subjected driven car is made up of the following components:

**COMPONENTS:**

We have used the following components to make All Direction Robotic Car:

1. Battery Operated (BO) Motors
2. Arduino Uno
3. L298N Motor Driver
4. 12V, 3.5A battery (Rechargeable)
5. Mecanum Wheels
6. HC – 05 Bluetooth Module

**BATTERY OPERATED (BO) MOTOR**

**BO (Battery Operated)** light weight DC geared motor which gives good torque and rpm at lower voltages. This motor can run at approximately 150 RPM when driven by a single Li-Ion cell. Great for battery operated light weight robots. A specific type of DC geared motors that can be operated through battery and that why known as Battery Operated (BO) motors. It is used for light weight applications mostly. Available in different torque and RPM

**Features**

* Input Voltage(V): 4.5 - 9 V
* Current rating: 0.07A (maximum on load)
* Speed (RPM): 300 RPM+-10%



Figure 1: BO Motor

**HC – 05 BLUETOOTH MODULE**

The HC-05 Bluetooth module is a module designed for wireless serial communication. It is a slave module meaning that it can receive serial data when serial data is sent out from a master Bluetooth device (device able to send serial data through the air: smart phones, PC).

**A picture containing text, electronics, circuit

Description automatically generated**

Figure 2: HC-05 Bluetooth Module

**ARDUINO UNO**

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. This board can be interfaced with other Arduino boards, Arduino shields, Raspberry Pi boards and can control relays, LEDs, servos, and motors as an output.

Arduino UNO features AVR microcontroller Atmega328, 6 analogue input pins, and 14 digital I/O pins out of which 6 are used as PWM output.

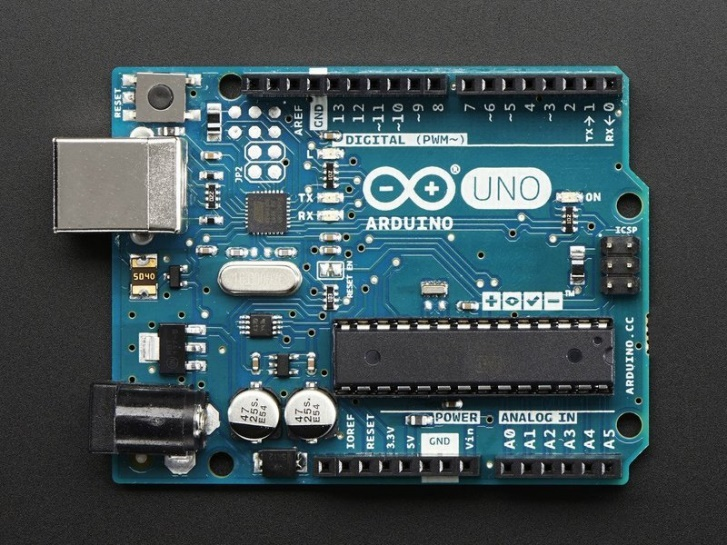


Figure 3: Arduino Uno

**L298N Motor Drivers**

L298N module is a high voltage, high current dual full-bridge motor driver module for controlling DC motor and stepper motor. It can control both the speed and rotation direction of two DC motors. This module consists of an L298 dual-channel H-Bridge motor driver IC. This module uses two techniques for the control speed and rotation direction of the DC motors.



Figure 4: L298N Motor Driver

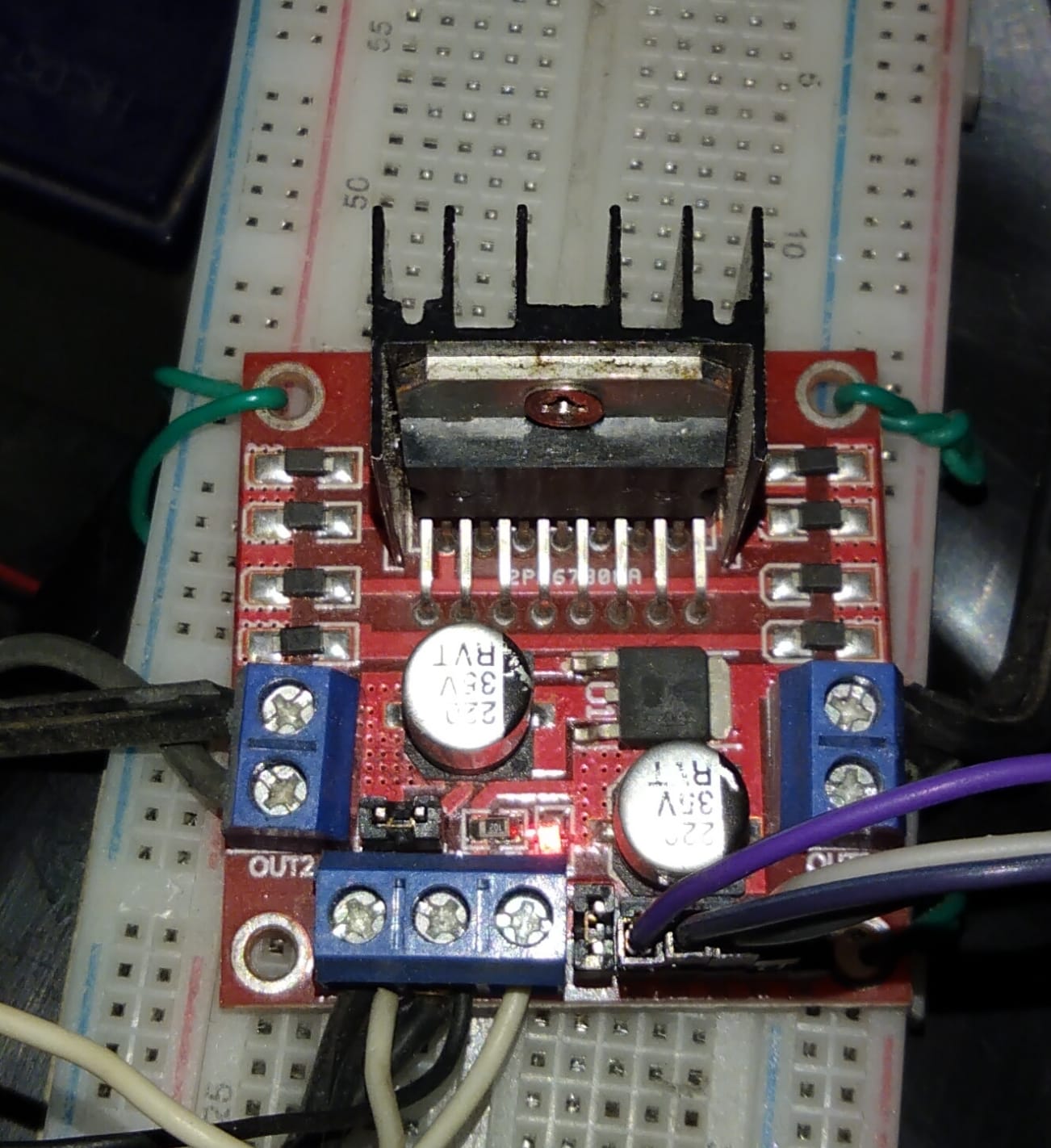


Figure 5: L298N Motor Driver Actual picture

**18650 CELL**

An 18650 is a lithium-ion rechargeable battery. Their proper name is “18650 cell”. The 18650 cell has voltage of 3.7v and has between 1800mAh and 3500mAh (mili-amp-hours).18650s may have a voltage range between 2.5 volts and 4.2 volts, or a charging voltage of 4.2 volts, but the nominal voltage of a standard 18650 is 3.7 volts.

* **Types**

There are two types: protected and unprotected. Protected cells include a protection circuit that stops the cell from being overcharged. Unprotected cells can be overcharged and burst or potentially cause a fire unless there are specific electronics to protect the battery. The popular LG HG2 and Samsung 25r are both UNPROTECTED batteries, only use them in a device designed to use unprotected 18650s

* **Charge time**

The average 18650 battery charge time is about 30 minutes. Charge time can vary with amperage and voltage of the charger and the battery type.

* **Other uses**

They are used in flashlights, electronics, laptops, vaping and even some electric vehicles use 18650s. The Tesla uses 7180 of these batteries. Many high lumen flashlights such as the [Thrunite TN14](https://www.amazon.com/ThruNite-TN12-2016-White-Flashlight/dp/B01EWW91S8/ref=as_li_ss_tl?ie=UTF8&linkCode=sl1&tag=commosensehom-20&linkId=d38e231c2b20d23869d552883dde92a2&language=en_US" \t "_blank) or [Fenix PD35](https://www.amazon.com/Fenix-Flashlights-FX-PD35TAC-Flashlight-Lumen/dp/B010ESCLHW/ref=as_li_ss_tl?ie=UTF8&qid=1541553694&sr=8-3&keywords=pd35+flashlight&linkCode=sl1&tag=commosensehom-20&linkId=14a1b2eeae42f924a388e1ae296944c8&language=en_US) use the 18650 or the even larger 21700. Laptops and other electronic devices use one or more 18650’s and have recharging electronics-built in. 18650's are also used in vaping (smoking) devices.



Figure 6: 18650 Battery

**MECANUM WHEELS (OR ILON WHEEL)**

The Mecanum wheel is an omnidirectional wheel design for a land-based vehicle to move in any direction. It is sometimes called the Swedish wheel or Ilon wheel after its inventor, ‘Bengt Erland Ilon’. The Mecanum wheel is based on a tireless wheel, with a series of rubberized external rollers obliquely attached to the whole circumference of its rim. These rollers typically each have an axis of rotation at 45° to the wheel plane and at 45° to the axle line. Each Mecanum wheel is an independent non-steering drive wheel with its own powertrain, and when spinning generates a propelling force perpendicular to the roller axle, which can be vectored into a longitudinal and a transverse component in relation to the vehicle.



Figure 7: Mecanum Wheels

A picture containing indoor, toy, LEGO

Description automatically generated

Figure 8: Mecanum Wheels actual picture

**SCHEMATIC DIAGRAM**

Diagram

Description automatically generated

Figure 9: Schematic Diagram

**PIN CONFIGRATION**

|  |  |  |
| --- | --- | --- |
| **Arduino UNO Pin** | **Bluetooth HC-05 Pin** | **L298N Motor Driver** |
| +5V | Vcc |  |
| GND | GND | GND |
| Rx | Tx |  |
| Tx | Rx |  |
| D2, D3 |  | Upper Right Motor (L298N #1 in-3&4) |
| D4, D5 |  | Upper Left Motor (L298N #1 in-1&2) |
| D8, D9 |  | Lower Right Motor (L298N #2 in-1&2) |
| D6, D7 |  | Lower Left Motor (L298N #2 in-3&4) |
| Vin |  | +5V |

**WORKING OF PROJECT**

The brain of this robot platform is an Arduino Uno board which controls each wheel individually. Each wheel is attached on a BO motor and knowing the fact that BO motors can be precisely controlled. We can wirelessly control the robot using the NRF24L01 radio transceiver modules, but in our case, we are trying to make it possible to be controlled using a smartphone via Bluetooth communication.

For powering the whole robot, we will use 12V power supply, and in our project, we are using 18650 cells which provides around 12V. For the Bluetooth communication we are using the HC-05 Bluetooth module. We also included a dedicated 5V voltage regulator which can provide around 3A of current. This is optional, like for driving Arduino board etc.

Diagram

Description automatically generated

Figure 10 Working Mechanism of Mecanum Wheel

Also, there is the design of the robot, we are using Mecanum wheels for achieving movement in all direction. So, by rotating the wheels in certain pattern, we utilize these diagonal forces and thus the robot can move in any direction.

We should also note here that we need two types of Mecanum wheels, often referred to as, left-handed, and right-handed Mecanum wheels. The difference between them is the orientation of the rollers and they must be installed in the robot in specific locations. The rotation axis of each wheel’s top roller should point to the centre of the robot.

**CODE USED**

|  |
| --- |
| //MECANUM WHEELS ROBOT  //ROBOT LK  #define IN\_11 2 // L298N #1 in 4 motor Front Right  #define IN\_12 3 // L298N #1 in 3 motor Front Right  #define IN\_13 4 // L298N #1 in 2 motor Front Left  #define IN\_14 5 // L298N #1 in 1 motor Front Left  #define IN\_21 6 // L298N #2 in 4 motor Back Left  #define IN\_22 7 // L298N #2 in 3 motor Back Left  #define IN\_23 8 // L298N #2 in 2 motor Back Right  #define IN\_24 9 // L298N #2 in 1 motor Back Right  //ROBOT LK  int command; //Int to store app command state.  boolean state = 1;  void stopRobot() {  digitalWrite(IN\_11, LOW);  digitalWrite(IN\_12, LOW);  digitalWrite(IN\_13, LOW);  digitalWrite(IN\_14, LOW);    digitalWrite(IN\_24, LOW);  digitalWrite(IN\_23, LOW);    digitalWrite(IN\_22, LOW);  digitalWrite(IN\_21, LOW);    }  void back() {  digitalWrite(IN\_11, LOW);  digitalWrite(IN\_12, HIGH);  digitalWrite(IN\_13, LOW);  digitalWrite(IN\_14, HIGH);    digitalWrite(IN\_24, HIGH);  digitalWrite(IN\_23, LOW);    digitalWrite(IN\_22, HIGH);  digitalWrite(IN\_21, LOW);    }  void forward () {  digitalWrite(IN\_11, HIGH);  digitalWrite(IN\_12, LOW);  digitalWrite(IN\_13, HIGH);  digitalWrite(IN\_14, LOW);    digitalWrite(IN\_24, LOW);  digitalWrite(IN\_23, HIGH);    digitalWrite(IN\_22, LOW);  digitalWrite(IN\_21, HIGH);    }  void left() {  digitalWrite(IN\_11, HIGH);  digitalWrite(IN\_12, LOW);    digitalWrite(IN\_13, LOW);  digitalWrite(IN\_14, HIGH);  digitalWrite(IN\_24, LOW);  digitalWrite(IN\_23, HIGH);    digitalWrite(IN\_22, HIGH);  digitalWrite(IN\_21, LOW);    }  void right() {  digitalWrite(IN\_11, LOW);  digitalWrite(IN\_12, HIGH);    digitalWrite(IN\_13, HIGH);  digitalWrite(IN\_14, LOW);  digitalWrite(IN\_24, HIGH);  digitalWrite(IN\_23, LOW);    digitalWrite(IN\_22, LOW);  digitalWrite(IN\_21, HIGH);    }  void superleft () {  digitalWrite(IN\_11, LOW);  digitalWrite(IN\_12, HIGH);    digitalWrite(IN\_13, HIGH);  digitalWrite(IN\_14, LOW);  digitalWrite(IN\_24, LOW);  digitalWrite(IN\_23, HIGH);  digitalWrite(IN\_22, HIGH);  digitalWrite(IN\_21, LOW);  }  void superright () {  digitalWrite(IN\_11, HIGH);  digitalWrite(IN\_12, LOW);    digitalWrite(IN\_13, LOW);  digitalWrite(IN\_14, HIGH);    digitalWrite(IN\_24, HIGH);  digitalWrite(IN\_23, LOW);    digitalWrite(IN\_22, LOW);  digitalWrite(IN\_21, HIGH);  }  void forwardright () {  digitalWrite(IN\_11, LOW);  digitalWrite(IN\_12, LOW);  digitalWrite(IN\_13, HIGH);  digitalWrite(IN\_14, LOW);  digitalWrite(IN\_24, LOW);  digitalWrite(IN\_23, LOW);  digitalWrite(IN\_22, LOW);  digitalWrite(IN\_21, HIGH);    }  void forwardleft () {  digitalWrite(IN\_11, HIGH);  digitalWrite(IN\_12, LOW);    digitalWrite(IN\_13, LOW);  digitalWrite(IN\_14, LOW);    digitalWrite(IN\_24, LOW);  digitalWrite(IN\_23, HIGH);    digitalWrite(IN\_22, LOW);  digitalWrite(IN\_21, LOW);    }  void backleft () {  digitalWrite(IN\_11, LOW);  digitalWrite(IN\_12, HIGH);  digitalWrite(IN\_13, LOW);  digitalWrite(IN\_14, LOW);    digitalWrite(IN\_24, HIGH);  digitalWrite(IN\_23, LOW);    digitalWrite(IN\_22, LOW);  digitalWrite(IN\_21, LOW);    }  void backright () {  digitalWrite(IN\_11, LOW);  digitalWrite(IN\_12, LOW);    digitalWrite(IN\_13, LOW);  digitalWrite(IN\_14, HIGH);    digitalWrite(IN\_24, LOW);  digitalWrite(IN\_23, LOW);    digitalWrite(IN\_22, HIGH);  digitalWrite(IN\_21, LOW);    }  void setup() {  Serial.begin (9600);    pinMode(IN\_11, OUTPUT);  pinMode(IN\_12, OUTPUT);  pinMode(IN\_13, OUTPUT);  pinMode(IN\_14, OUTPUT);  pinMode(IN\_21, OUTPUT);  pinMode(IN\_22, OUTPUT);  pinMode(IN\_23, OUTPUT);  pinMode(IN\_24, OUTPUT);    }  //ROBOT LK  void loop() {  if (Serial.available()) {  command = Serial.read();  if (command == 'X') {  state = 1;  } else if (command == 'x') {  state = 0;  }  if (command == 'B') {  back();  } else if (command == 'F') {  forward();  } else if (command == 'R' && state == 1) {  superright ();  } else if (command == 'L' && state == 1) {  superleft ();  } else if (command == 'R' && state == 0) {  right ();  } else if (command == 'L' && state == 0) {  left ();  } else if (command == 'G') {  forwardleft ();  } else if (command == 'I') {  forwardright ();  } else if (command == 'H') {  backright ();  } else if (command == 'J') {  backleft ();  } else {  stopRobot();  }  }  } |

**PROJECT PICTURES**

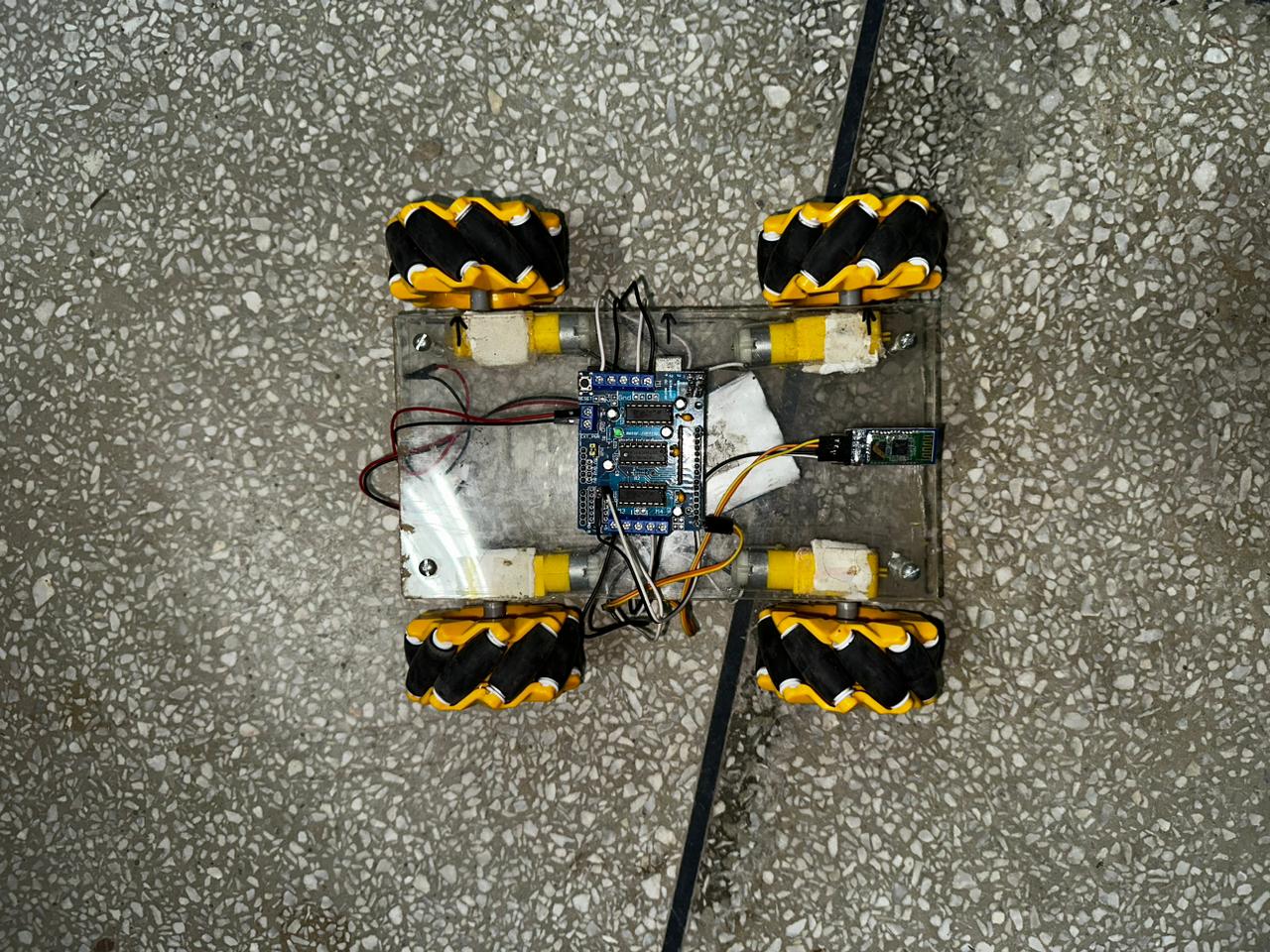


Figure 11: Upper view of vehicle

A screenshot of a video game

Description automatically generated with medium confidence

Figure 12: Full view of Mobile application

* **Buy Project Parts Here**

<https://hallroad.org/>

* **Wheel Buy link**

[Arduino UPS Generators Pakistan (arduinopak.com)](https://www.arduinopak.com/Search.aspx?Search_Term=mecanum%20wheel)

* **App link**

<https://play.google.com/store/apps/details?id=braulio.calle.bluetoothRCcontroller>