INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, PUNE



**LFWC Robot**

(Light following & wireless controlled Robot)

**ELECTRONICS WORKSHOP 2018-19**

**BY:-**

Year & semester:- First Year

Branch & Batch:- ECE & B2

1. Raushan kumar (111816029)

2.Shreyansh Gupta (111816034)

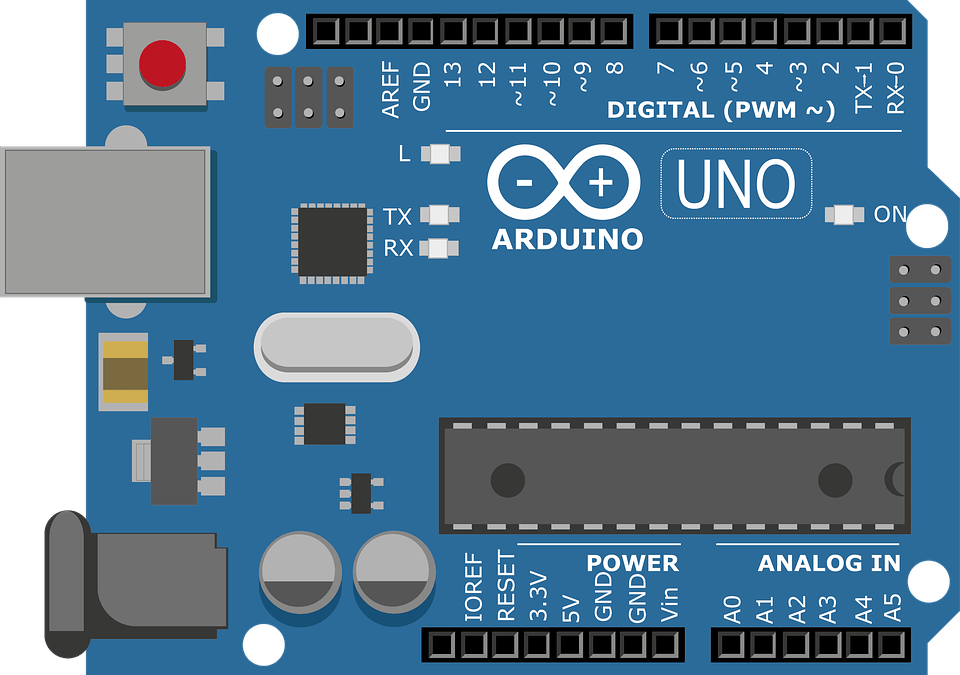
3.Suhani Maheshwari (111816039)

4.Ujjawal Dubey (111816041)

**Things our robot can do :-**

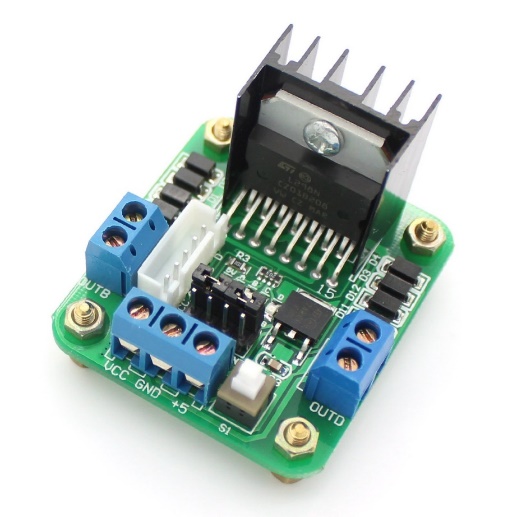
1. It will move according to our command by **Bluetooth**(in all directions eg. little right forwarded).
2. It will follow light and the speed of robot depends up on the intensity of light using **LDR**
3. It will stop before hitting any obstacle or tackle it in any accidental situation (in front) even if being forcibly made to strike.( **Ultrasonic Sensor**).
4. A **LCD** is fitted at the back which shows working condition of Robot car and displays the distance from the obstacle.

**Hardware introduction & Materials Required :-**

1. Arduino :-

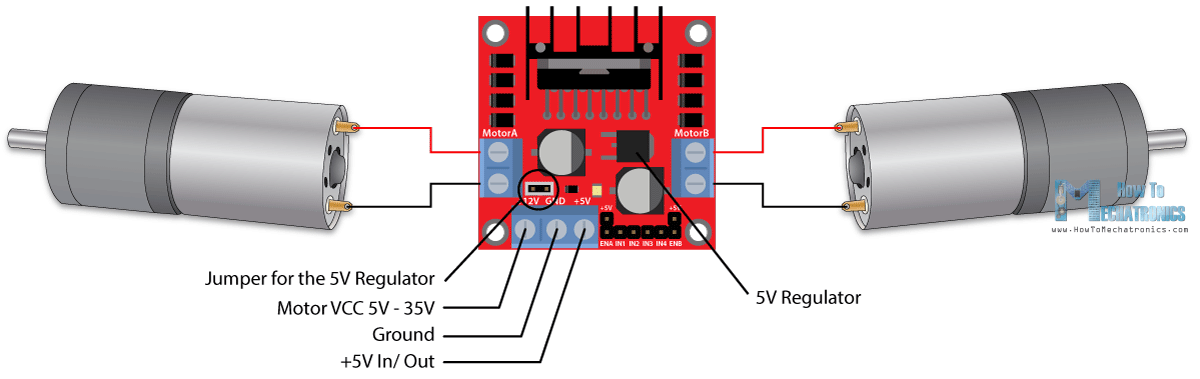
**Arduino** is an [open-source hardware](https://en.wikipedia.org/wiki/Open-source_hardware) and [software](https://en.wikipedia.org/wiki/Open-source_software) company, project and user community that designs and manufactures [single-board microcontrollers](https://en.wikipedia.org/wiki/Single-board_microcontroller) and [microcontroller](https://en.wikipedia.org/wiki/Microcontroller) kits for building digital devices and interactive objects that can sense and control both physically and digitally. Its products are licensed under the [GNU Lesser General Public License](https://en.wikipedia.org/wiki/GNU_Lesser_General_Public_License) (LGPL) or the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License) (GPL),[[1]](https://en.wikipedia.org/wiki/Arduino#cite_note-1) permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form or as [do-it-yourself](https://en.wikipedia.org/wiki/Do-it-yourself) (DIY) kits.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog [input/output](https://en.wikipedia.org/wiki/Input/output) (I/O) pins that may be interfaced to various expansion boards or [breadboards](https://en.wikipedia.org/wiki/Breadboards) (*shields*) and other circuits. The boards feature serial communications interfaces, including [Universal Serial Bus](https://en.wikipedia.org/wiki/Universal_Serial_Bus) (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B). In addition to using traditional [compiler](https://en.wikipedia.org/wiki/Compiler) [toolchains](https://en.wikipedia.org/wiki/Toolchains), the Arduino project provides an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) based on the [Processing](https://en.wikipedia.org/wiki/Processing_(programming_language)) language project.

2.Motor driver ic(**L298N Driver**):-

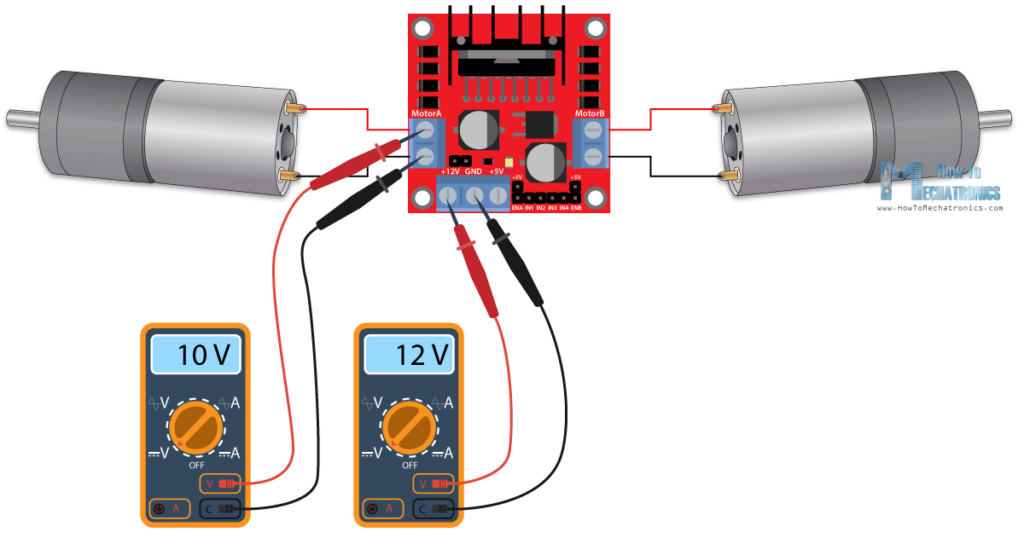
The L298N is a dual H-Bridge motor driver which allows speed and direction control of two DC motors at the same time. The module can drive DC motors that have voltages between 5 and 35V, with a peak current up to 2A.

Let’s take a closer look at the pinout of L298N module and explain how it works. The module has two screw terminal blocks for the motor A and B, and another screw terminal block for the Ground pin, the VCC for motor and a 5V pin which can either be an input or output.

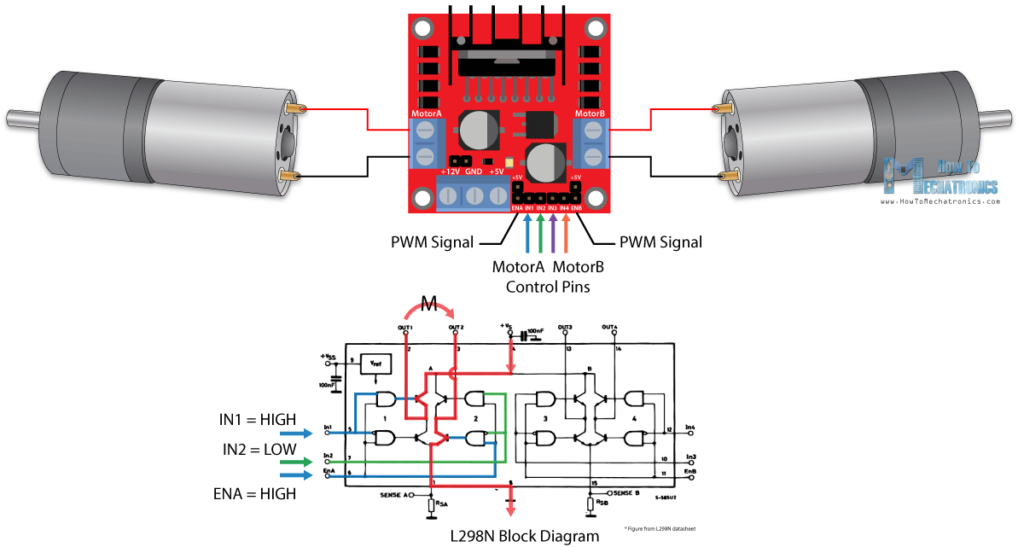


This depends on the voltage used at the motors VCC. The module have an onboard 5V regulator which is either enabled or disabled using a jumper. If the motor supply voltage is up to 12V we can enable the 5V regulator and the 5V pin can be used as output, for example for powering our Arduino board. But if the motor voltage is greater than 12V we must disconnect the jumper because those voltages will cause damage to the onboard 5V regulator. In this case the 5V pin will be used as input as we need connect it to a 5V power supply in order the IC to work properly.

We can note here that this IC makes a voltage drop of about 2V. So for example, if we use a 12V power supply, the voltage at motors terminals will be about 10V, which means that we won’t be able to get the maximum speed out of our 12V DC motor.



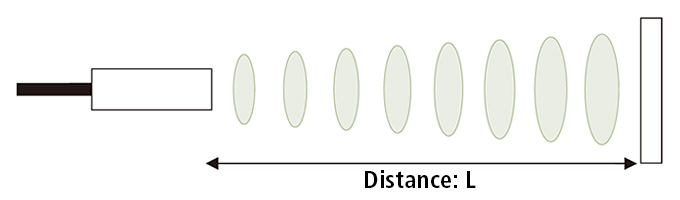
Next are the logic control inputs. The Enable A and Enable B pins are used for enabling and controlling the speed of the motor. If a jumper is present on this pin, the motor will be enabled and work at maximum speed, and if we remove the jumper we can connect a PWM input to this pin and in that way control the speed of the motor. If we connect this pin to a Ground the motor will be disabled.

[](https://howtomechatronics.com/wp-content/uploads/2017/08/L298N-Block-Diagram-Current-Flow-How-It-Works.png)

Next, the Input 1 and Input 2 pins are used for controlling the rotation direction of the motor A, and the inputs 3 and 4 for the motor B. Using these pins we actually control the switches of the H-Bridge inside the L298N IC. If input 1 is LOW and input 2 is HIGH the motor will move forward, and vice versa, if input 1 is HIGH and input 2 is LOW the motor will move backward. In case both inputs are same, either LOW or HIGH the motor will stop. The same applies for the inputs 3 and 4 and the motor B.

3.Ultra sonic sensor:-

As the name indicates, ultrasonic sensors measure distance by using ultrasonic waves.  
The sensor head emits an ultrasonic wave and receives the wave reflected back from the target. Ultrasonic Sensors measure the distance to the target by measuring the time between the emission and reception.



An optical sensor has a transmitter and receiver, whereas an ultrasonic sensor uses a single ultrasonic element for both emission and reception. In a reflective model ultrasonic sensor, a single oscillator emits and receives ultrasonic waves alternately. This enables miniaturization of the sensor head.

### Distance calculation

The distance can be calculated with the following formula:

### Distance L = 1/2 × T × C

where L is the distance, T is the time between the emission and reception, and C is the sonic speed. (The value is multiplied by 1/2 because T is the time for go-and-return distance.).

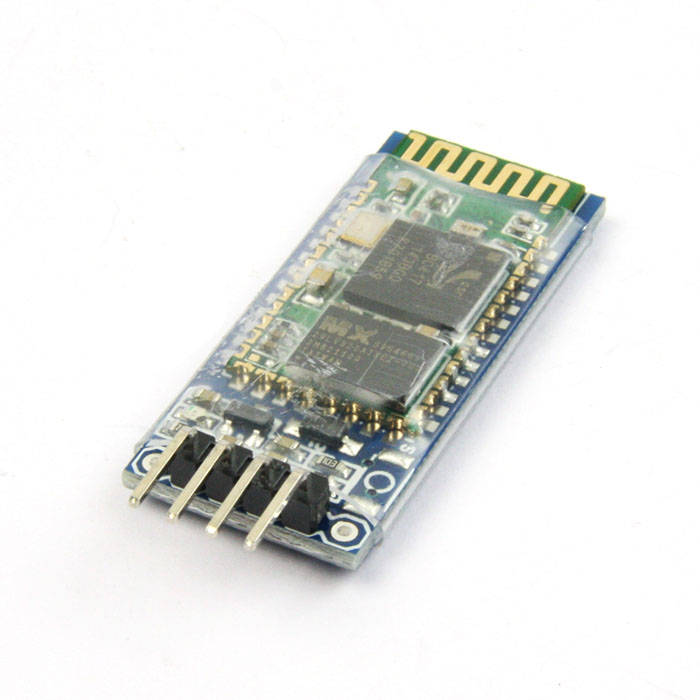
4. LDR(Light dependent resistor):-

A **photoresistor** (or **light-dependent resistor**, **LDR**, or **photo-conductive cell**) is a light-controlled variable [resistor](https://en.m.wikipedia.org/wiki/Resistor). The [resistance](https://en.m.wikipedia.org/wiki/Electrical_resistance) of a photoresistor decreases with increasing incident light intensity; in other words, it exhibits [photoconductivity](https://en.m.wikipedia.org/wiki/Photoconductivity). A photoresistor can be applied in light-sensitive detector circuits, and light-activated and dark-activated switching circuits.

|  |  |
| --- | --- |
|  | |
|  | |
|  |  |
|  |  |
|  | |
| The symbol for a photoresistor | |

[](https://en.m.wikipedia.org/wiki/File:Photoresistor_symbol.svg)A photoresistor is made of a high resistance [semiconductor](https://en.m.wikipedia.org/wiki/Semiconductor). In the dark, a photoresistor can have a resistance as high as several megohms (MΩ), while in the light, a photoresistor can have a resistance as low as a few hundred ohms. If incident light on a photoresistor exceeds a certain [frequency](https://en.m.wikipedia.org/wiki/Frequency), [photons](https://en.m.wikipedia.org/wiki/Photon) absorbed by the semiconductor give bound [electrons](https://en.m.wikipedia.org/wiki/Electron) enough energy to jump into the [conduction band](https://en.m.wikipedia.org/wiki/Conduction_band). The resulting free electrons (and their [hole](https://en.m.wikipedia.org/wiki/Electron_hole) partners) conduct electricity, thereby lowering [resistance](https://en.m.wikipedia.org/wiki/Electrical_resistance). The resistance range and sensitivity of a photoresistor can substantially differ among dissimilar devices. Moreover, unique photoresistors may react substantially differently to photons within certain wavelength bands.

A photoelectric device can be either intrinsic or extrinsic. An intrinsic semiconductor has its own [charge carriers](https://en.m.wikipedia.org/wiki/Charge_carrier) and is not an efficient semiconductor, for example, silicon. In intrinsic devices the only available electrons are in the [valence band](https://en.m.wikipedia.org/wiki/Valence_band), and hence the photon must have enough energy to excite the electron across the entire [bandgap](https://en.m.wikipedia.org/wiki/Bandgap). Extrinsic devices have impurities, also called [dopants](https://en.m.wikipedia.org/wiki/Dopants), added whose ground state energy is closer to the conduction band; since the electrons do not have as far to jump, lower energy photons (that is, longer wavelengths and lower frequencies) are sufficient to trigger the device. If a sample of silicon has some of its atoms replaced by phosphorus atoms (impurities), there will be extra electrons available for conduction. This is an example of an extrinsic semiconduct.

Bluetooth module (HC-06):-

The HC-06 is a class 2 slave Bluetooth module designed for transparent wireless serial communication. Once it is paired to a master Bluetooth device such as PC, smart phones and tablet, its operation becomes transparent to the user. All data received through the serial input is immediately transmitted over the air. When the module receives wireless data, it is sent out through the serial interface exactly at it is received. No user code specific to the Bluetooth module is needed at all in the user microcontroller program.

The HC-06 will work with supply voltage of 3.6VDC to 6VDC, however, the logic level of RXD pin is 3.3V and is not 5V tolerant. A [Logic Level Converter](http://www.sgbotic.com/index.php?dispatch=products.view&product_id=1613) is recommended to protect the sensor if connect it to a 5V device (e.g Arduino Uno and Mega).

**NOTE**: This module does not support AT command.

**Dimension**: 38(L) x 15(W) x 2.35(H)mm

**LCD (Liquid Crystal Display):-**

It has 16 pins and the first one from left to right is the Ground pin. The second pin is the VCC which we connect the 5 volts pin on the Arduino Board. Next is the Vo pin on which we can attach a potentiometer for controlling the contrast of the display.

Next, The RS pin or register select pin is used for selecting whether we will send commands or data to the LCD. For example if the RS pin is set on low state or zero volts, then we are sending commands to the LCD like: set the cursor to a specific location, clear the display, turn off the display and so on. And when RS pin is set on High state or 5 volts we are sending data or characters to the LCD.



Next comes the R / W pin which selects the mode whether we will read or write to the LCD. Here the write mode is obvious and it is used for writing or sending commands and data to the LCD. The read mode is used by the LCD itself when executing the program which we don’t have a need to discuss about it in this tutorial.

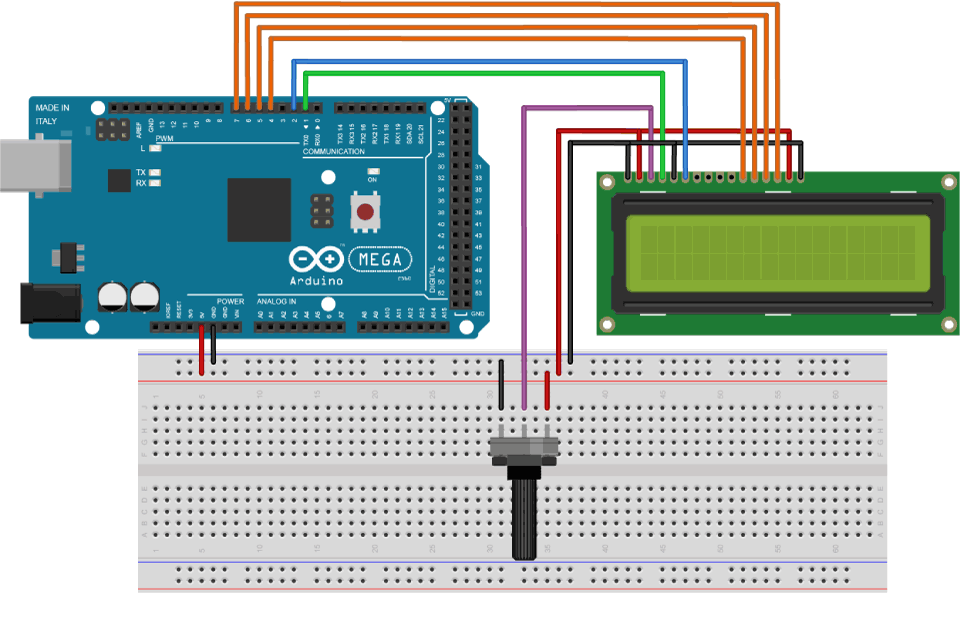
Next is the E pin which enables the writing to the registers, or the next 8 data pins from D0 to D7. So through this pins we are sending the 8 bits data when we are writing to the registers or for example if we want to see the latter uppercase A on the display we will send 0100 0001 to the registers according to the ASCII table.

And the last two pins A and K, or anode and cathode are for the LED back light.

After all we don’t have to worry much about how the LCD works, as the Liquid Crystal Library takes care for almost everything. From the Arduino’s official website you can find and see the functions of the library which enable easy use of the LCD. We can use the Library in 4 or 8 bit mode. In this tutorial we will use it in 4 bit mode, or we will just use 4 of the 8 data pins.

## Circuit Schematic

We will use just 6 digital input pins from the Arduino Board. The LCD’s registers from D4 to D7 will be connected to Arduino’s digital pins from 4 to 7. The Enable pin will be connected to pin number 2 and the RS pin will be connected to pin number 1. The R/W pin will be connected to Ground and the Vo pin will be connected to the potentiometer.



**WORKING PRINCIPLE:-**

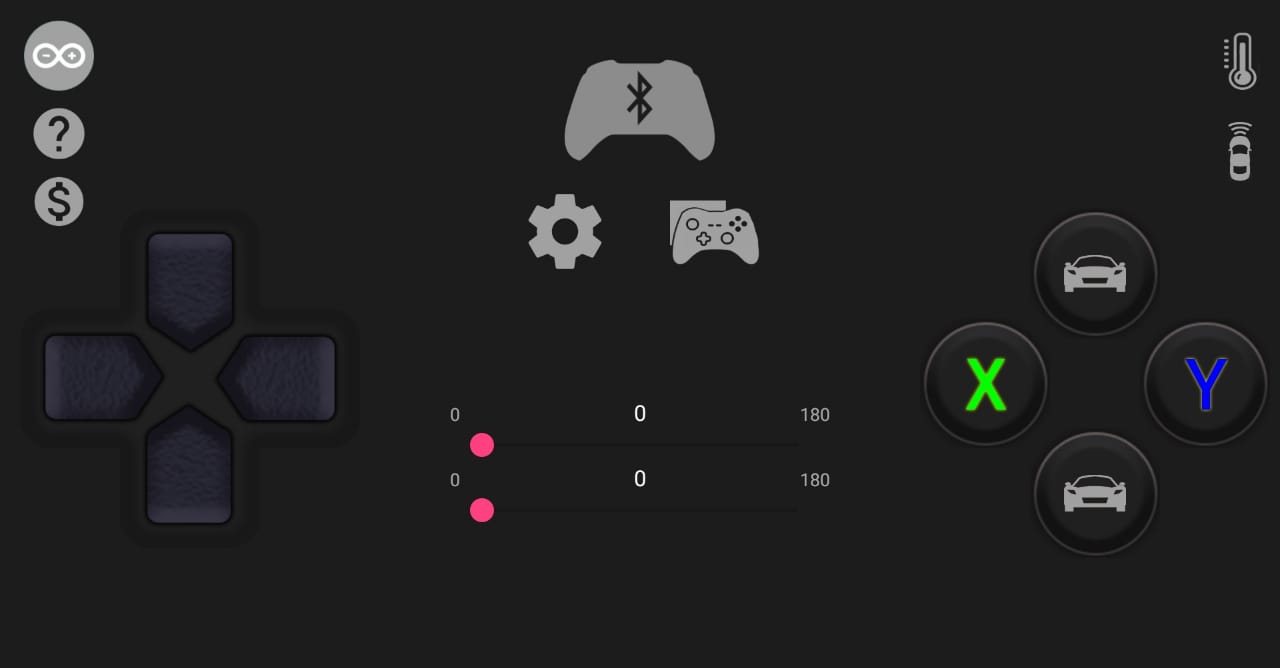
**REMOTE CONTROL:-**

After assembling the robot,we make the necessary connections and uploaded the code to Arduino. We understood the HC-05 Bluetooth Module, then implementing the Bluetooth Controlled Robot project was not tough.

First, in the Android App, We have mainly used 4 keys as Forward, Reverse, Left, Right. The corresponding data associated with each key is as follows:

* Forward – “F”
* Reverse – “G”
* Left – “L”
* Right – “R”
* Stop –“S” When none of the above inputs was being given,
* Many more inputs were being edited in the source code for Arduino.

When a key is pressed, the corresponding data is transmitted to the Bluetooth Module from the Phone over Bluetooth Communication.

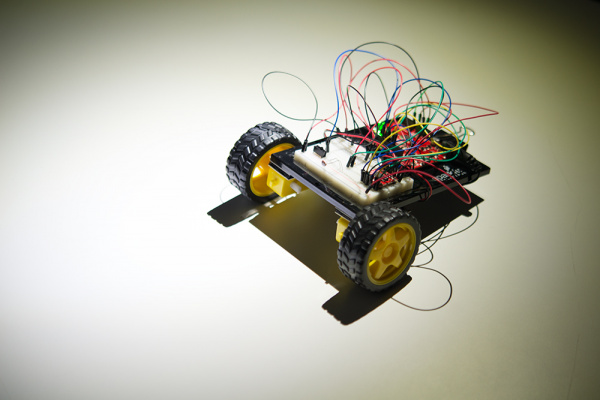


In the Arduino code, the Arduino UNO receives any of this data from the Bluetooth Module (as per the key pressed) and performs a simple switch case operation, where each case associated with appropriate instructions to the Motor Driver Input Pins.

For example, if ‘Forward’ key is pressed in the Android Phone, then ‘F’ is transmitted. Arduino will then make IN1 and IN3 as HIGH and IN2 and IN4 as LOW to achieve a forward motion.

Similarly, other keys correspond to appropriate setting of IN1 – IN4 pins.

**LIGHT FOLLOWING:-**

**[](https://cdn.sparkfun.com/assets/learn_tutorials/7/1/3/SIKv4_Projects-12.jpg)**When the input command is OFF, the robot will not move or is at standby condition. When you turn the light fall on LDRs, the robot will turn left and right, taking light measurements at each extreme. It will also take a light measurement from the center.

The robot turns to the direction with the most intense light and moves forward a small amount. It then repeats the pattern of looking for light and moves toward the direction of brightest light.

Note that trying to direct the robot with a flashlight or other light source can be difficult. Reflected light from the wheels can sometimes be brighter, for instance, than reflected light on the ground. It can take some patience to get the robot to move the way you want. You can also try modifying the code to make it faster at taking three measurements (by spinning the wheels more quickly or not turning as far to measure) or to take more than three measurements at a time.

**ACCIDENT CONTROL:-**

Our robot has two sensor and performs a very simple search-decide-move loop. Now we wanted to make your mechanical and electrical "organism" more robust and smarter, so we used additional sensors, such as an [Ultrasonic Distance Sensor](https://www.sparkfun.com/products/13959) to determine if the robot is about to hit an object.

So, now whenever the distance from any obstacle(in front) is less than 20cm( as per our code), it stops and only the backward command works from the Bluetooth.

**Arduino Code :-**

**#include <LiquidCrystal.h>**

**// initialize the library with the numbers of the interface pins**

**LiquidCrystal lcd(A4, A5, 2, 3, 4, 5); /// REGISTER SELECT PIN,ENABLE PIN,D4 PIN,D5 PIN, D6 PIN, D7 PIN**

**char Incoming\_value ;//Variable for storing Incoming\_value**

**int ldr\_left,ldr\_right;**

**int n=0;**

**int m=0;**

**void setup()**

**{**

**Serial.begin(38400);**

**lcd.begin(16, 2); // set up the LCD's number of columns and rows:**

**pinMode(2,OUTPUT); // lcd dispaly of respectively from d4 to d7**

**pinMode(3,OUTPUT);**

**pinMode(4,OUTPUT);**

**pinMode(5,OUTPUT);**

**//A0 and A1 for ldr..**

**pinMode(A2,OUTPUT); //trigger for ultrasonic**

**pinMode(A3,INPUT); // reciever for ultrasonic**

**//A4 and A5 for RS(read serial) & enable...**

**pinMode(8,OUTPUT); //mtr 1 ground (right from back)**

**pinMode(9,OUTPUT); //mtr 1 vcc**

**pinMode(10,OUTPUT); //mtr 2 ground (left from back)**

**pinMode(11,OUTPUT); //mtr 2 vcc**

**pinMode(6,OUTPUT); //front light**

**pinMode(7,OUTPUT); //back light**

**pinMode(12,OUTPUT); //right light**

**pinMode(13,OUTPUT); //left light //universal ground for light is ground pin....**

**int i;**

**lcd.display(); //to activate the display..**

**lcd.print(" WEL-COME "); //print welcome..**

**lcd.setCursor(0, 1); // set the cursor to column 0, line 2**

**lcd.print(" LFWC Robot .."); //print name of project..**

**for(i=9 ;i>0;i--)**

**{**

**lcd.setCursor(13, 1); // setting cursor to 13 column in line 2**

**lcd.print(i); // countdown for starting...**

**delay(1000); // countdown will be follow the speed of second arm.....**

**}**

**lcd.setCursor(0, 0);**

**lcd.clear();**

**lcd.print(" STARTING ......");**

**delay(1000);**

**lcd.clear();**

**lcd.setCursor(0,0);**

**}**

**float readings() //for calculate the distance....by ultrasonic sensor...☺☻♥♥♥**

**{**

**float raw\_dist1,dist1;**

**digitalWrite(A2,0);**

**digitalWrite(A2,1);**

**delayMicroseconds(10);**

**digitalWrite(A2,0);**

**raw\_dist1=pulseIn(A3,1);**

**dist1 = raw\_dist1\*(0.0172);**

**if(dist1<0)**

**{**

**readings();**

**}**

**return dist1; //returning distance to main program....♥♥**

**}**

**void rightlight()**

**{**

**digitalWrite(12,1); //right light**

**}**

**void drightlight()**

**{**

**digitalWrite(12,0); //right light**

**}**

**void leftlight()**

**{**

**digitalWrite(13,1); //left light**

**}**

**void dleftlight()**

**{**

**digitalWrite(13,0); //left light**

**}**

**void frontlight()**

**{**

**digitalWrite(6,1); //front light**

**}**

**void dfrontlight()**

**{**

**digitalWrite(6,0); //front light**

**}**

**void backlight()**

**{**

**digitalWrite(7,1); //back light**

**}**

**void dbacklight()**

**{**

**digitalWrite(7,0); //back light**

**}**

**void standby()**

**{**

**digitalWrite(8,0);**

**digitalWrite(10,0);**

**digitalWrite(9,0);**

**digitalWrite(11,0);**

**lcd.setCursor(0, 1); // setting at column 0 in line 2..**

**lcd.print(" STANDBY ");**

**Serial.println("standby");**

**demergency();**

**}**

**void forward()**

**{**

**n=0;**

**float y;**

**y=readings();**

**if(y>=20)**

**{**

**digitalWrite(8,0);**

**digitalWrite(10,0);**

**digitalWrite(9,1);**

**digitalWrite(11,1);**

**Serial.println("forward");**

**lcd.setCursor(0, 1); // setting at column 0 in line 2..**

**lcd.print(" FORWARD ");**

**frontlight(); //front light**

**}**

**else**

**standby();**

**}**

**void backward()**

**{**

**digitalWrite(8,1);**

**digitalWrite(10,1);**

**digitalWrite(9,0);**

**digitalWrite(11,0);**

**Serial.println("backward");**

**lcd.setCursor(0, 1); // setting at column 0 in line 2..**

**lcd.print(" BACKWARD ");**

**backlight(); //back light**

**}**

**void right()**

**{**

**digitalWrite(8,0);**

**digitalWrite(10,0);**

**digitalWrite(9,0);**

**digitalWrite(11,1);**

**Serial.println("right");**

**lcd.setCursor(0, 1); // setting at column 0 in line 2..**

**lcd.print(" RIGHT ");**

**frontlight(); //front light**

**rightlight(); //right light**

**}**

**void left()**

**{**

**digitalWrite(8,0);**

**digitalWrite(10,0);**

**digitalWrite(9,1);**

**digitalWrite(11,0);**

**Serial.println("left");**

**lcd.setCursor(0, 1); // setting at column 0 in line 2..**

**lcd.print(" LEFT ");**

**frontlight(); //front light**

**leftlight(); //left light**

**}**

**void emergency()**

**{**

**frontlight(); //front light**

**backlight(); //back light**

**rightlight(); //right light**

**leftlight(); //left light**

**}**

**void demergency()**

**{**

**dfrontlight(); //front light**

**dbacklight(); //back light**

**drightlight(); //right light**

**dleftlight(); //left light**

**}**

**void bluetooth()**

**{**

**if(Incoming\_value=='F')**

**{**

**forward();**

**}**

**else if(Incoming\_value=='S')**

**{**

**standby();**

**}**

**else if(Incoming\_value=='R')**

**{**

**right();**

**}**

**else if(Incoming\_value=='L')**

**{**

**left();**

**}**

**else if(Incoming\_value=='G')**

**{**

**backward();**

**}**

**else if(Incoming\_value=='X')**

**{**

**emergency();**

**}**

**else if(Incoming\_value=='Y')**

**{**

**demergency();**

**}**

**else if(Incoming\_value=='E')**

**{**

**forward();**

**right();**

**}**

**else if(Incoming\_value=='Q')**

**{**

**forward();**

**left();**

**}**

**else if(Incoming\_value=='Z')**

**{**

**backward();**

**left();**

**}**

**else if(Incoming\_value=='C')**

**{**

**backward();**

**right();**

**}**

**else if(Incoming\_value=='M' || Incoming\_value=='m')**

**{**

**if(Incoming\_value=='M')**

**{**

**frontlight();**

**}**

**else**

**{**

**dfrontlight();**

**}**

**}**

**else if(Incoming\_value=='N' || Incoming\_value=='n')**

**{**

**if(Incoming\_value=='N')**

**{**

**backlight();**

**}**

**else**

**{**

**dbacklight();**

**}**

**}**

**}**

**void bluetooth1()**

**{**

**if(Incoming\_value=='F')**

**{**

**standby();**

**}**

**else if(Incoming\_value=='S')**

**{**

**standby();**

**}**

**else if(Incoming\_value=='R')**

**{**

**standby();**

**}**

**else if(Incoming\_value=='L')**

**{**

**standby();**

**}**

**else if(Incoming\_value=='G')**

**{**

**backward();**

**}**

**else if(Incoming\_value=='X')**

**{**

**emergency();**

**}**

**else if(Incoming\_value=='Y')**

**{**

**demergency();**

**}**

**else if(Incoming\_value=='E')**

**{**

**standby();**

**right();**

**}**

**else if(Incoming\_value=='Q')**

**{**

**standby();**

**left();**

**}**

**else if(Incoming\_value=='Z')**

**{**

**backward();**

**left();**

**}**

**else if(Incoming\_value=='C')**

**{**

**backward();**

**right();**

**}**

**else if(Incoming\_value=='M' || Incoming\_value=='m')**

**{**

**if(Incoming\_value=='M')**

**{**

**frontlight();**

**}**

**else**

**{**

**dfrontlight();**

**}**

**}**

**else if(Incoming\_value=='N' || Incoming\_value=='n')**

**{**

**if(Incoming\_value=='N')**

**{**

**backlight();**

**}**

**else**

**{**

**dbacklight();**

**}**

**}**

**}**

**void loop()**

**{**

**float x;**

**x=readings();**

**lcd.print(" DISTANCE : ");**

**lcd.setCursor(11, 0); // setting at column 11 in line 1..**

**lcd.print(x);**

**Serial.print("distance :-");**

**Serial.println(x);**

**if(x>20)**

**{**

**if(Serial.available()) //checking the incoming value from bluetooth**

**{**

**Incoming\_value = Serial.read(); //Read the incoming data and store it into variable Incoming\_value**

**Serial.println(Incoming\_value); //Print Value of Incoming\_value in Serial monitor**

**bluetooth();**

**}**

**else**

**{**

**ldr\_left=analogRead(0);**

**Serial.print("ldr\_left :");**

**Serial.print(ldr\_left);**

**ldr\_right=analogRead(1);**

**Serial.print("ldr\_right:");**

**Serial.println(ldr\_right);**

**demergency();**

**if(ldr\_left >500 && ldr\_right >500)**

**{**

**ldr\_left=0;**

**ldr\_right=0;**

**forward();**

**}**

**else if(ldr\_left >500 && ldr\_right<500)**

**{**

**ldr\_left=0;**

**ldr\_right=0;**

**left();**

**}**

**else if(ldr\_left <500 && ldr\_right>500)**

**{**

**ldr\_left=0;**

**ldr\_right=0;**

**right();**

**}**

**else if(ldr\_left<500 && ldr\_right<500)**

**{**

**ldr\_left=0;**

**ldr\_right=0;**

**standby();**

**}**

**}**

**}**

**else**

**{**

**if(Serial.available()) //checking the incoming value from bluetooth**

**{**

**Incoming\_value = Serial.read(); //Read the incoming data and store it into variable Incoming\_value**

**Serial.println(Incoming\_value); //Print Value of Incoming\_value in Serial monitor**

**bluetooth1();**

**}**

**else**

**{**

**if(n==0)**

**{**

**standby();**

**n=n+1;**

**}**

**ldr\_left=analogRead(0);**

**Serial.println("ldr\_left");**

**ldr\_right=analogRead(1);**

**Serial.println("ldr\_right");**

**demergency();**

**if(ldr\_left >500 && ldr\_right >500)**

**{**

**standby();**

**}**

**else if(ldr\_left >500 && ldr\_right<500)**

**{**

**standby();**

**}**

**else if(ldr\_left <500 && ldr\_right>500)**

**{**

**standby();**

**}**

**ldr\_left=0;**

**ldr\_right=0;**

**}**

**}**

**}**

Result :-

We have successfully studied the working of Arduino UNO & its coding ,LDR (Light dependent resistor) , Motor Driver IC (L298n) ,Bluetooth module (HC -06) , Ultrasonic Sensor (HC-SR04) , LCD (16 X 2) and different coloured LEDs and implemented it in our LFWC Robot.!!