Datta Meghe College of Engineering

Sector 3, Airoli, Navi Mumbai-400 708
Department of Electronics Engineering
Academic year 2017-2018

CERTIFICATE

This is to certify that the students performed and completed satisfactorily In subject Mini project 2, semester (VI) of Bachelor of engineering course of

University of Mumbai

Project Report On

ARDUINO RC CAR

Name of Students	Division	Roll No.	Examination Seat No.
PRANIT KUMBHAR	A	30	
RUTURAJ KADAM	A	22	
PRIYA KHANOLKAR	A	27	
SAKSHI GUJAR	A	16	

Practical Incharge   Head of Department   Principal

Internal Examiner   External Examiner

Date:

ACKNOWLEDGEMENT

We take this opportunity to express our profound gratitude and deep regards to our guide **prof. Nandwalkar Jayant** for this exemplary guidance, monitoring, and constant encouragement throughout the course of this project work.
We also take this opportunity to express a deep sense of gratitude to **Dr. Pete** HOD of ELEX Dept. for this cordial support, valuable information and guidance, which helped us in completing this task through various stages.
We are obliged to staff members of **Datta Meghe College of Engineering,** for the valuable information provided by them in their respective fields. We are grateful for their cooperation during the period of our project work.
Lastly, we thank almighty,our parents,our family and friends for their constant encouragement.

ABSTRACT

In recent years, there have been rise in the number of applications based on Radio Frequency Identification (RFID) systems and have been successfully applied to different areas as diverse as transportation, health-care, agriculture, and hospitality industry to name a few. RFID technology facilitates automatic wireless identification using electronic passive and active tags with suitable readers. In this paper, an attempt is made to solve recurrent lecture attendance monitoring problem in developing countries using RFID technology. The application of RFID to student attendance monitoring as developed and deployed in this study is capable of eliminating time wasted during manual collection of attendance and an opportunity for the educational administrators to capture face-to-face classroom statistics for allocation of appropriate attendance scores and for further managerial decisions.

Keywords: RFID, Lecture, Attendance, Passive tag, Reader

CONTENTS

ARDUINO RC CAR

PRIYA KHANOLKAR,SAKSHI GUJAR,PRANIT KUMBHAR,RUTURAJ KADAM

Datta Meghe College Of Engineering
Sector – 3, Airoli
Navi Mumbai 400708

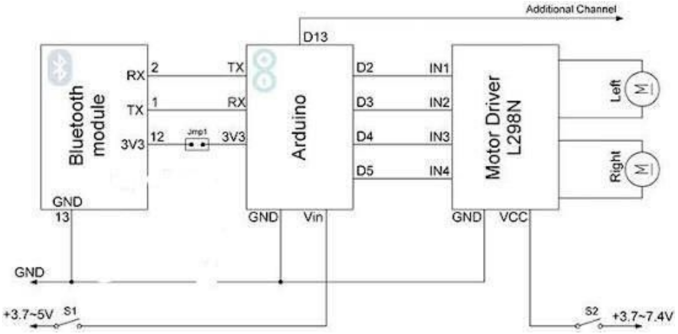
***Abstract* — the project presents a circuit which uses LDR for logging the changes in light intensity of LED. The circuit consists of Wheatstone network, IC LM 358, capacitor and LED. The overall aim of the circuit is to save energy by supplying electricity only when it's required and switching it off when not.**

Keywords-Arduino uno, L298 motor driver, HC05 Bluetooth module (key words)

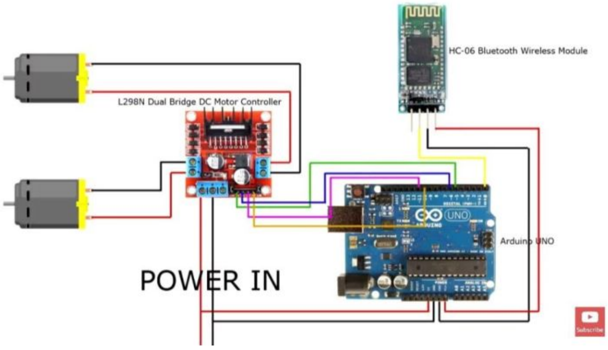
I. INTRODUCTION

The Arduino is platform open-source hardware, based on a simple board with inputs and outputs, analog and digital, in a development environment that is based on the programming language processing. It is a device that connects the physical world with the virtual world, or the analogue world with the digital. Bluetooth controlled car is controlled by using Android mobile phone instead of any other method like buttons, gesture etc. Here only needs to touch button in android phone to control the car in forward, backward, left and right directions. So here android phone is used as transmitting device and Bluetooth module placed in car is used as receiver. Android phone will transmit command using its in-built Bluetooth to car so that it can move in the required direction like moving forward, reverse, turning left, turning right and stop.

II. CIRCUIT BLOCK DAIGRAM



III. CIRCUIT DIAGRAM


HC-06 Bluetooth Wireless Module
L298N Dual Bridge DC Motor Controller
Arduino UNO
POWER IN

IV. CIRCUIT EXPLANATION

All calculations are performed in the Android application, and immediately calculate the values and the PWM for the left and right motor. Application has flexible settings, such as the range of the PWM, the sensitivity of tilt, the minimum threshold for the PWM and other.
Example commands transmitted by Bluetooth:
L-255\rR-120\r
L - the command to the left engine, R - for the right
A dash means the motor rotation to move back
255 - PWM value (for Arduino is the maximum speed of rotation) On this command RC car will move forward and slightly rotated to the right, as right engine rotates slowly left. L255\rR-255\r On this command the left engine will rotate back and forward right, forcing a car to rotate around its axis counterclockwise.
In the MCU control program provides a timer that shuts off the engine if the last command was received more than n-seconds ago. The data are stored in the EEPROM memory of the controller and can be changed from Android device. The range of this setting is from 0.1 seconds to 99.9 seconds. This setting can be disabled. To work with EEPROM provides commands: Fr - reading values 2‹2‹and Fw - record values.
From the block diagram
As we can see, the Arduino connects to Bluetooth module and a motor driver with two or four connected motors.

A. ARDUINO UNO

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards or Breadboards (*shields*) and other circuits. The boards feature serial communications interfaces, including 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 and C++. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

The Arduino UNO is a widely used open-source microcontroller board based on the ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits.[1] The board features 14 Digital pins and 6 Analog pins. It is programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable.[4] It can be powered by a USB cable or by an external 9 volt battery.

Special Pin Functions:

Each of the 14 digital pins and 6 Analog pins on the Uno can be used as an input or output, using pinMode,digitalWrite, and digitalRead functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller.The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference function.

Most Arduino boards contain a light-emitting diode (LED) and a load resistor connected between pin 13 and ground, which is a convenient feature for many tests and program functions.[61] A typical program for a beginning Arduino programmer blinks a LED repeatedly. This program uses the functions *pinMode()*, *digitalWrite()*, and *delay()*, which are provided by the internal libraries included in the IDE environment.[62][63][64] This program is usually loaded into a new Arduino board by the manufacturer.

Technical Specification.

- Microcontroller: ATmega328P
- Operating Voltage: 5v
- Input Voltage: 7-20v
- Digital I/O Pins: 14 (of which 6 provide PWM

  output)
- Analog Input Pins: 6
- DC Current per I/O Pin: 20 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB of which 0.5 KB used

  by bootloader
- SRAM: 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz
- Length: 68.6 mm
- Width: 53.4 mm

DataSheet Of Arduino Uno

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital

input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a

USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to

support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC

adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI

USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version

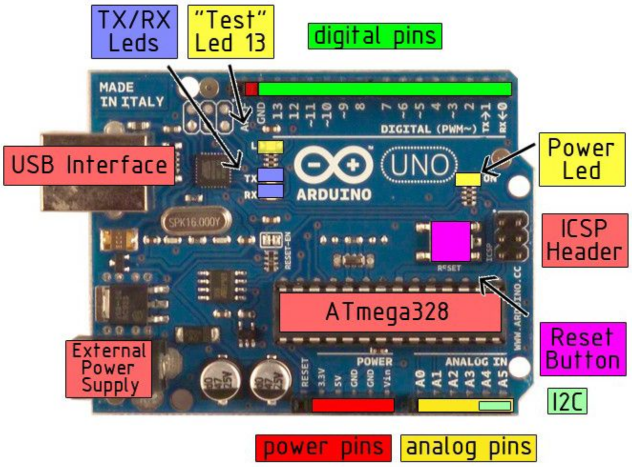
1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB

Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions,

see the index of Arduino boards.

Technical Specifications

The Board



Power

he Arduino Uno can be powered via the USB connection or with an external power supply. The power

source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a

battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V

pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage

regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

· VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to

5 volts from the USB connection or other regulated power source). You can supply voltage through

this pin, or, if supplying voltage via the power jack, access it through this pin.

· 5V. The regulated power supply used to power the microcontroller and other components on the

board. This can come either from VIN via an on-board regulator, or be supplied by USB or another

regulated 5V supply.

· 3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

· GND. Ground pins.

Memory

The Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the bootloader); It has also 2 KB of SRAM and 1

KB of EEPROM (which can be read and written with the EEPROM library).

Input And Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and

digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and

has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have

specialized functions:

· Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. TThese pins are

connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .

· External Interrupts: 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. See the attachInterrupt() function for details.

· PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.

· SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which,

although provided by the underlying hardware, is not currently included in the Arduino language.

· LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is

on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By

default they measure from ground to 5 volts, though is it possible to change the upper end of their range

using the AREF pin and the analogReference() function. Additionally, some pins have specialized

functionality:

· I2C: 4 (SDA) and 5 (SCL). Support I2C (TWI) communication using the Wire library.

There are a couple of other pins on the board:

· AREF. Reference voltage for the analog inputs. Used with analogReference().

· Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to

shields which block the one on the board.

See also the mapping between Arduino pins and Atmega328 ports.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other

microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on

digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB

and appears as a virtual com port to software on the computer. The '8U2 firmware uses the standard USB

COM drivers, and no external driver is needed. However, on Windows, an *.inf file is required..

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the

Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-toserial

chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Uno's digital pins.

The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire

library to simplify use of the I2C bus; see the documentation for details. To use the SPI communication,

please see the ATmega328 datasheet.

Programming

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno w/

ATmega328" from the Tools > Board menu (according to the microcontroller on your board). For details,

see the reference and tutorials.

The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code

to it without the use of an external hardware programmer. It communicates using the original STK500

protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial

Programming) header; see these instructions for details.

The ATmega8U2 firmware source code is available . The ATmega8U2 is loaded with a DFU bootloader,

which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and

then resetting the 8U2. You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac

OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer

(overwriting the DFU bootloader).

Automatic Software Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a

way that allows it to be reset by software running on a connected computer. One of the hardware flow control

lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nanofarad

capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The

Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the

Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR

can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or

Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

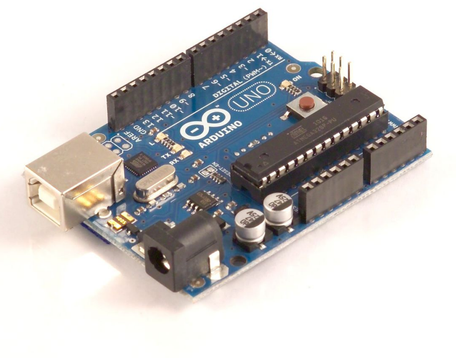
The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

USB Overcurrent Protection

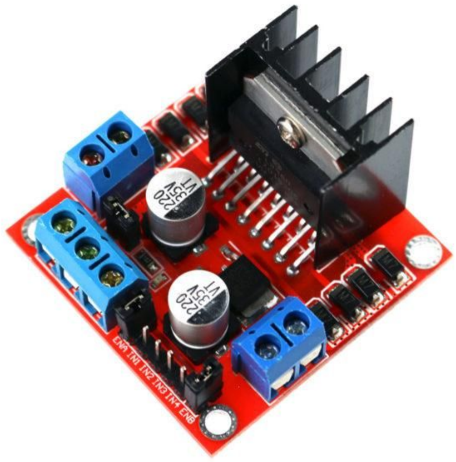
The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.



B. L298D MOTOR DRIVER



The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage. 300RPM 12V DC geared motors for robotics applications. Very easy to use and available in standard size. Nut and threads on shaft to easily connect and internal threaded shaft for easily connecting it to wheel.

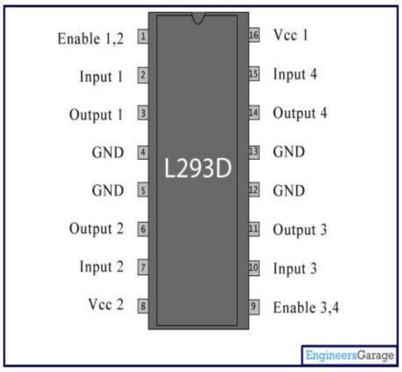
Key Features:

operating supply voltage up to 46v
low saturation voltage
total dc current up to 4a
logical \"0\" input voltage up to 1.5 v (high noise immunity)
overtemperature protection

DataSheet:

L293D is a dual H-bridge motor driver integrated circuit (IC). Motor drivers act as current amplifiers since they take a low-current control signal and provide a higher-current signal. This higher current signal is used to drive the motors. L293D contains two inbuilt H-bridge driver circuits. In its common mode of operation, two DC motors can be driven simultaneously, both in forward and reverse direction. The motor operations of two *motors can be controlled by input logic at pins 2 & 7 and* 10 & 15. Input logic 00 or 11 will stop the corresponding motor. Logic 01 and 10 will rotate it in clockwise and anticlockwise directions.

Pin Diagram:



Pin Description:

Pin No	Function	Name
1	Enable pin for Motor 1; active high	Enable 1,2
2	Input 1 for Motor 1	Input 1
3	Output 1 for Motor 1	Output 1
4	Ground (0V)	Ground
5	Ground (0V)	Ground
6	Output 2 for Motor 1	Output 2
7	Input 2 for Motor 1	Input 2
8	Supply voltage for Motors; 9-12V (up to 36V)	Vcc $_2$
9	Enable pin for Motor 2; active high	Enable 3,4
10	Input 1 for Motor 1	Input 3
11	Output 1 for Motor 1	Output 3
12	Ground (0V)	Ground
13	Ground (0V)	Ground
14	Output 2 for Motor 1	Output 4
15	Input2 for Motor 1	Input 4
16	Supply voltage; 5V (up to 36V)	Vcc $_1$

C. Bluetooth module HC-05

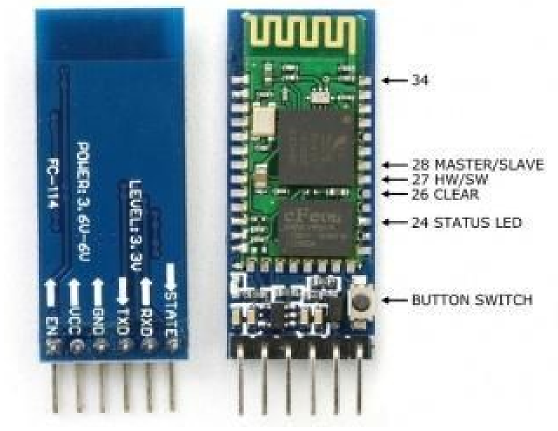
The Bluetooth module HC-05 is a MASTER/SLAVE module.By default the factory setting is SLAVE.The Role of the module (Master or Slave) can be configured only by AT COMMANDS.The slave modules cannot initiate a connection to another Bluetooth device, but can accept connections.Master module can initiate a connection to other devices.The user can use it simply for a serial port replacement to establish connection between MCU and GPS, PC to your embedded project, etc.Just go through the datasheet for more details

Hardware Features

Typical - 80 dBm sensitivity.Up to +4dBm RF transmit power.3.3 to 5 V I/O.PIO(Programmable Input/Output) control.UART interface with programmable baud rate.With integrated antenna.With edge connector.

Software Features

Slave default Baud rate: 9600, Data bits:8, Stop bit:1,Parity:No parity.Auto-connect to the last device on power as default.Permit pairing device to connect as default.Auto-pairing PIN CODE:”1234” as default.



Pin Description

The HC-05 Bluetooth Module has 6 pins. They are as follows:

ENABLE : When enable is pulled LOW, the module is disabled which means the module will not turn on and it fails to communicate.When enable is left open or connected to 3.3V, the module is enabled i.e the module remains on and communication also takes place.

Vcc : Supply Voltage 3.3V to 5V

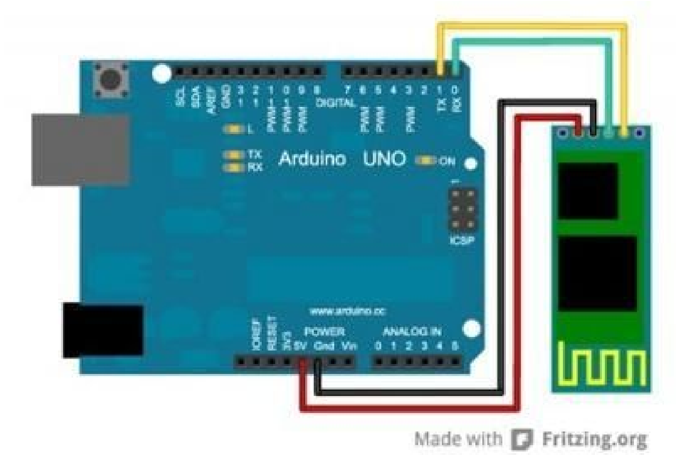
GND : Ground pin

TXD & RXD : These two pins acts as an UART interface for communication

STATE : It acts as a status indicator.When the module is not connected to / paired with any other bluetooth device,signal goes Low.At this low state,the led flashes continuously which denotes that the module is not paired with other device.When this module is connected to/paired with any other bluetooth device,the signal goes High.At this high state,the led blinks with a constant delay say for example 2s delay which indicates that the module is paired.

Hardware connections of Bluetooth module HC-O5 with Arduino uno:

As we know that Vcc and Gnd of the module goes to Vcc and Gnd of Arduino.The TXD pin goes to RXD pin of Arduino and RXD pin goes to TXD pin of Arduino i.e(digital pin 0 and 1).The user can use the on board Led.But here,Led is connected to digital pin 12 externally for betterment of the process.



COMMAND and DATA TRANSFER MODES:

The module has two modes of operation, Command Mode where we can send AT commands to it and Data Mode where it transmits and receives data to another bluetooth module.

The default mode is DATA Mode, and this is the default configuration, that may work fine for many applications:

Baud Rate: 9600 bps, Data : 8 bits, Stop Bits: 1 bit, Parity : None, Handshake: None

Passkey: 1234

Device Name: HC-05

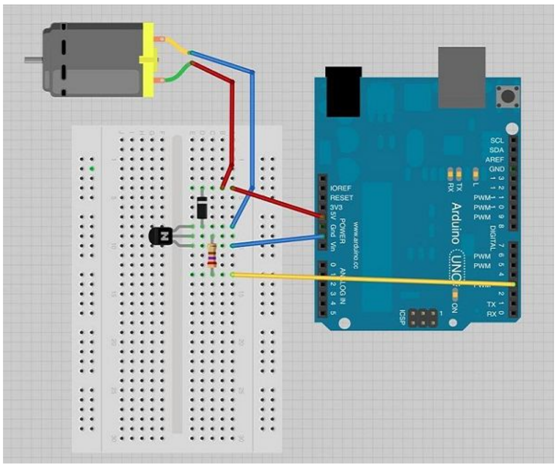
D. DC Motor

A **DC motor** is any of a class of rotary electrical machines that converts direct current electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the direction of current flow in part of the motor.

DC motors were the first type widely used, since they could be powered from existing direct-current lighting power distribution systems. A DC motor's speed can be controlled over a wide range, using either a variable supply voltage or by changing the strength of current in its field windings. Small DC motors are used in tools, toys, and appliances. The universal motor can operate on direct current but is a lightweight motor used for portable power tools and appliances. Larger DC motors are used in propulsion of electric vehicles, elevator and hoists, or in drives for steel rolling mills. The advent of power electronics has made replacement of DC motors with AC motors possible in many applications.

Features

- 300RPM 12V DC motors with Gearbox
- 3000RPM base motor
- 6mm shaft diameter with internal hole
- 125gm weight
- Same size motor available in various rpm
- 0.35kgcm torque
- No-load current = 60 mA(Max), Load current = 300 mA(Max)



Take the following precautions while making the connections.

- First, make sure that the transistor is connected in the right way. The flat side of the transistor should face the Arduino board as shown in the arrangement.
- Second, the striped end of the diode should be towards the +5V power line according to the arrangement shown in the image.

Spin Control Arduino Code

```
int motorPin=3;
void setup()
 {
}
void loop()
 {
 digitalwrite(motorPin, HIGH);
}
```

E. ARDUINO RC CAR CODE

```
int motorLpin1=2;
int motorLpin2=3;
int motorRpin1=4;
int motorRpin2=5;
int motorLpwm=10;
int motorRpwm=11;

void setup() {
  Serial.begin(9600);
  Serial.flush();
  pinMode(motorLpin1,OUTPUT);
  pinMode(motorLpin2,OUTPUT);
  pinMode(motorRpin1,OUTPUT);
  pinMode(motorRpin2,OUTPUT);
}

void loop() {
  String input="";
  while(Serial.available()){
    input+=(char)Serial.read();
    delay(5);
  }

  if(input=="n"){
    stp();
  }
  else if(input=="F"){
    fwd();
  }
  else if(input=="R"){
    rev();
  }
  else if(input.indexOf("TL")>-1){
    lft();
  }
  else if(input.indexOf("TR")>-1){
    rght();
  }
}

void fwd(){
  digitalWrite(motorLpin1,1);
  digitalWrite(motorLpin2,0);
  digitalWrite(motorRpin1,1);
  digitalWrite(motorRpin2,0);
}

void rev(){
  digitalWrite(motorLpin1,0);
  digitalWrite(motorLpin2,1);
  digitalWrite(motorRpin1,0);
  digitalWrite(motorRpin2,1);
}

void lft(){
  digitalWrite(motorLpin1,0);
  digitalWrite(motorLpin2,1);
  digitalWrite(motorRpin1,1);
  digitalWrite(motorRpin2,0);
}

void rght(){
  digitalWrite(motorLpin1,1);
  digitalWrite(motorLpin2,0);
  digitalWrite(motorRpin1,0);
  digitalWrite(motorRpin2,1);
}

void stp(){
  digitalWrite(motorLpin1,1);
  digitalWrite(motorLpin2,1);
  digitalWrite(motorRpin1,1);
  digitalWrite(motorRpin2,1);
}
```

F. WORKING

This car have two dc motors at its front and rear side. Front side motor is used for giving direction to car means turning left or right side (like real car steering feature). And rear side motor is used for driving the car in forward and backward direction. A Bluetooth module is used to receive command from android phone and Arduino UNO is used for controlling the whole system.

Bluetooth controlled car moves according to button touched in the android Bluetooth mobile app. To run this project first we need to download Bluetooth app form Google play store. We can use any Bluetooth app that supporting or can send data. Here are some apps' name that might work correctly.

- Bluetooth Spp pro

- Bluetooth controller

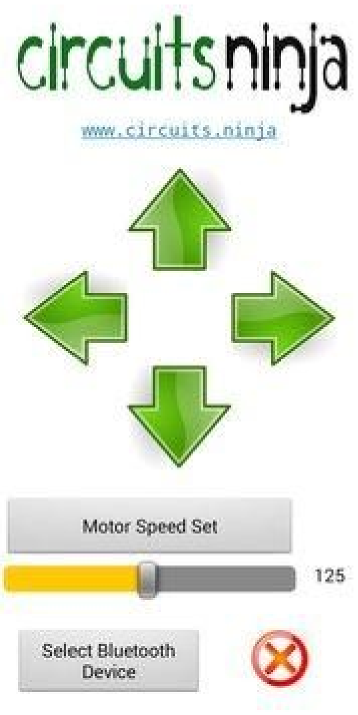
After installing app you need to open it and then search Bluetooth device and select desired Bluetooth device. And then configure keys. Here in this project we have used Bluetooth controller app.

1. Download and install Bluetooth Controller.
2. Turned ON mobile Bluetooth.
3. Now open Bluetooth controller app
4. Press scan
5. Select desired Bluetooth device
6. When we touch forward button in Bluetooth controller app then car start moving in forward direction and moving continues forward until next command comes.
7. When we touch backward button in Bluetooth controller app then car start moving in reverse direction and moving continues reverse until next command comes.
8. When we touch left button in Bluetooth controller app then car start moving in left direction and moving continues left until next command comes. In this condition front side motor turns front side wheels in left direction and rear motor runs in forward direction.
9. When we touch right button in Bluetooth controller app then car start moving in right direction and moving continues right until next command comes. In this condition front side motor turns front side wheels in right direction and rear motor runs in forward direction.

G. SOFTWARE

CIRCUIT NINJA

Although designed specifically for Circuits Ninja bluetooth vehicle projects, this app will work with any Arduino Uno compatible robot using a L298N H-Bridge driver, HC-06 bluetooth module, and running our control code.



H. CONCLUSION

The Arduino is an open source device that has been the brain fornumerous projects. The Arduino has everything that is required by the user which includes its inbuilt converter, i/o pins etc. With the combination of Arduino, and the Bluetooth Shield we can control over many other things, like home Lightings, air conditioner and many more through our cell phones. By doing this Project we found out a lot about the Arduino, and how it has made us easier to convert digital signals into physical movements. One more advantage of Arduino is that once a program is burned we don‘t need to worry about the program getting erased as long as it is not RESET. Arduino has also over all other microcontroller because of its efficiency and user friendly property.

I. REFERENCE

https://www.youtube.com/results?search_query=arduino+rc+car

https://www.youtube.com/watch?v=4zGlPofT53A

https://www.arduino.cc/