SUMMER INTERNSHIP PROGRAMME IN NITW

PROJECT REPORT ON BLUETOOTH CONTROLLED CAR

BY

Team SQUADRON

Team Members: B.Vaishnavi
B.Samyuktha
S.Lohitha

College Name: CMR Institute of Technology

Duration of Project: 15/06/19 to 27/06/19

PROBLEM STATEMENT

Conventionally, robots controlled by wireless communication employ radio frequency (RF) circuits, which have the drawbacks of limited working range, limited frequency range and the limited control. Use of a mobile phone for robotic control can overcome these limitations. It provides the advantage of robust control, working range as large as the coverage area of the service provider.

INTRODUCTION TO THE PROBLEM

Smartphones are becoming the latest sensation in our country. It currently has by far the biggest margins of any product in the tech sector. One of its important functions is wireless communication which is usually done by a wireless technology called Bluetooth. In this project our aim is to use that Bluetooth for controlling a four wheeler car. We are much successful in achieving our goal. Our car runs well and it can be controlled by three modes. One, from any Bluetooth activated computers. Second, from an android phone by touching buttons and third by the android's default accelerometer. This is actually a prototype which can be applied in our daily day to day jobs by adding some extension to it like robotic arms. Through this project we are able to learn many things which will be very helpful in our coming future.

WORKING PRINCIPLE

The working principle is kept as simple as possible. The working principle of the circuit has been elaborated with the help of a block diagram, of the system interconnection as shown in Figure 6. As seen from the Figure 6. A DC power supply is required to run the system. The DC power supple feeds the Microcontroller and the Bluetooth module. The Bluetooth module receives the signal sent from an android smart-phone, where the application software coded in C language is installed. The microcontroller, thereby, sends instructions, which when executed, helps in functioning of the motor driver. The movement and functioning of the motor can be controlled by using the android based application software.

REQUIREMENTS

SL No.	COMPONENT NAME	QTY.
1.	Arduino UNO	1
2.	HC-05 Bluetooth Module	1
3.	Motor Driver Shield (IC L293D)	1
4.	Wheels	4
5.	TT Gear Motor	4
6.	9V Battery	1
7.	Acrylic Sheet	2
8.	Male and Female jumper wires	As required
9.	DC Power Switch	1
10.	Connecting Wires	As Required

SOFTWARE REQUIRED:

Arduino UNO (version 1.8.8)

Bluetooth RC Controller App

PROJECT DESCRIPTION

Our system aims to achieve the target to design a system that can provide following functionalities with a simple and easy-to-use interface:

a) Develop an android application that will act as an remote of a robot.

b) Develop a robot which will be helpful for travelling.

c) Here the focus is on the latest technology of android and robot also called as ‘Mobot’ .

d) An android smart-phone and the technology of android is vast and can be used to interact with embedded system.

e) Mobile, robot and Bluetooth are the on-going technologies which can be used for the benefit of mankind.

f) Hardware of this project consists of Arduino UNO, Bluetooth module and a motor driver IC.

g) The Bluetooth module is connected with the Arduino UNO board for the connection with the user.

h) Through the Bluetooth module for monitoring and controlling the particular motor reaches the board and process accordingly and the output of the Arduino goes to the motor driver IC and it controls the particular motor.

The system consists of following parts:

a) Arduino UNO (ATMEGA 16u2)

b) Bluetooth module (HC-05)

c) Smart phone

d) Motor driver (l293D)

e) Arduino software (version 1.8.8)

The basic building blocks of the project have been described below:

Arduino UNO

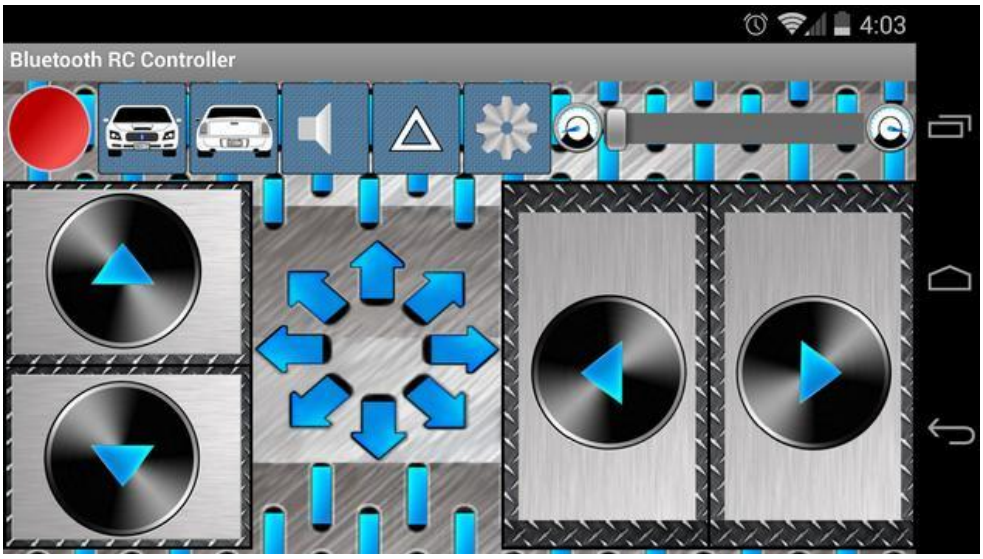
Microcontroller will act as the brain of the robot. The robot movement will be decided by the microcontroller. In this system we will be using microcontroller named Arduino UNO which contains ATMEGA 16u2 microcontroller chip. The microcontroller is programmed with the help of the Embedded C programming. Arduino has it own programming burnt in its Read Only Memory (ROM). C-program is very easy to implement for programming the Arduino UNO.

Bluetooth Module (HC-05)

The Bluetooth module will act as an interface between Smartphone and microcontroller. We will be using HC-05 Bluetooth module for the system, which can be used as either receiver or transmitter. Generally our transmitter will be smart-phone and receiver will be Bluetooth module . Bluetooth module will give the commands given by smart-phone to the microcontroller.

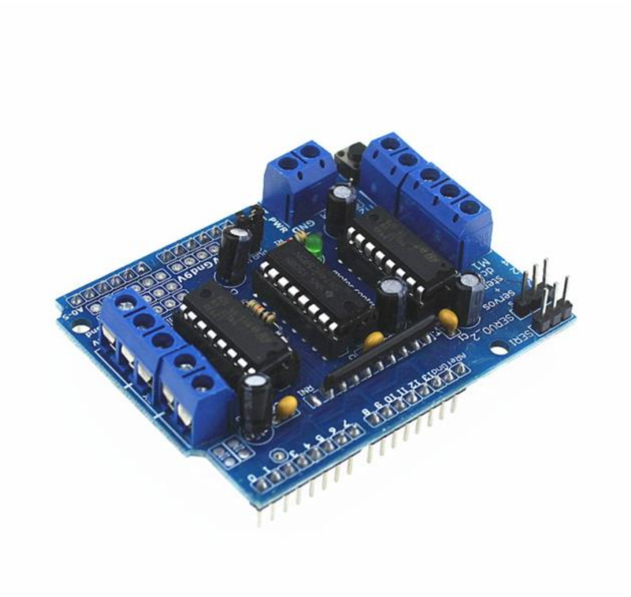
Smart Phone

The smart phone is the transmitter of this circuit. It sends the data to microcontroller through Bluetooth module. It also helps to send the instruction of forward, backward, left, right to the microcontroller. Actually, the smart phone is used as a remote of this system. Here we the Bluetooth RC Controller application as the operating remote of this system. The advantage of this project is that the application software designed for android phones is kept simple but attractive with all necessary built-in functions. The novelty lies in the simplicity of the design and functioning.



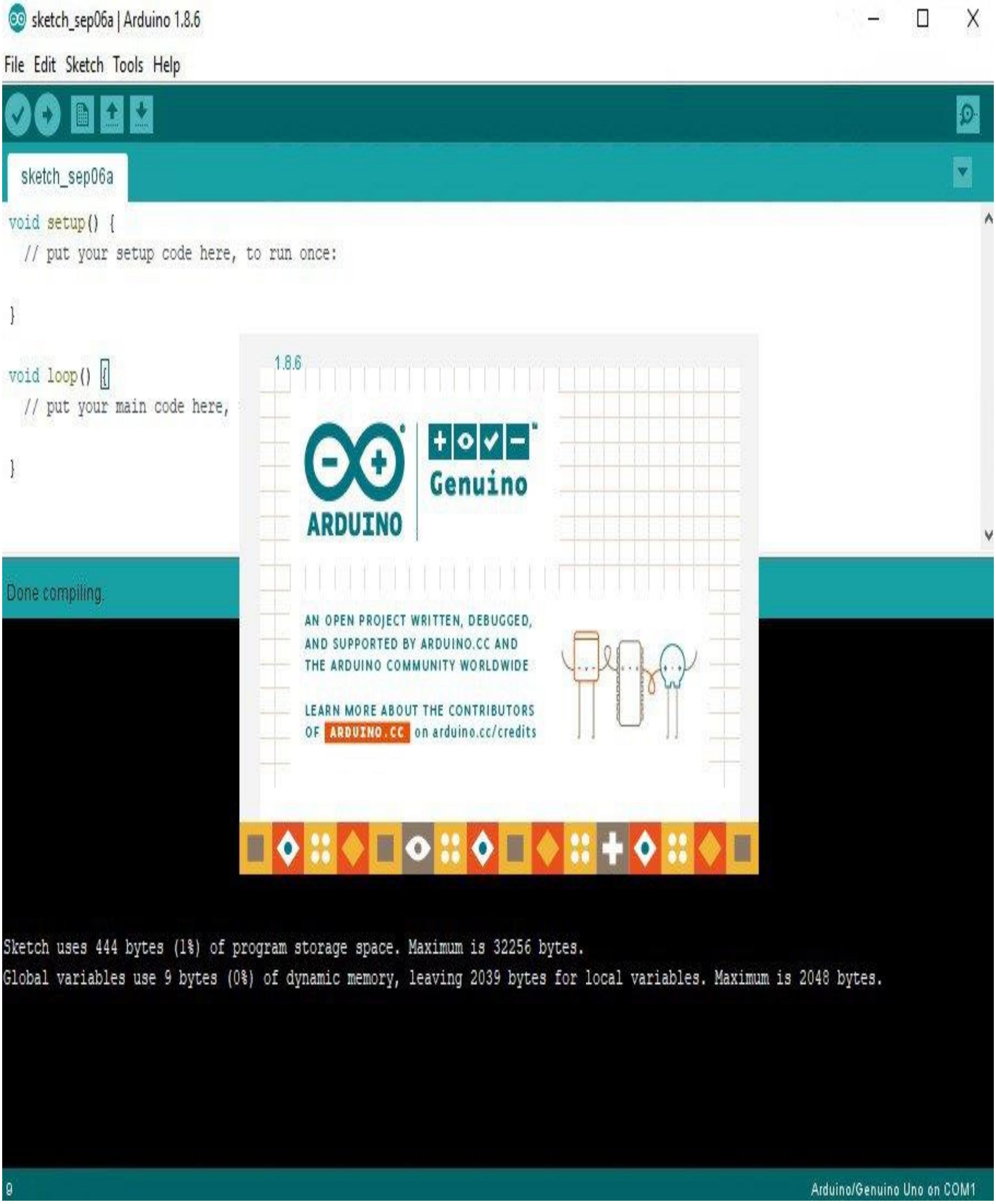
Motor Driver (L293D)

Motor driver IC is used to control the dc motors. It is also interfaced with the microcontroller and with circuit connections



Arduino Software (Version 1.8.8)

Arduino software is used to put the instruction of whole functions of this system to the microcontroller. Here we use programming language ‘C’ for coding. The program for executing this project has been written in C language. The program is burnt in the microcontroller using burner software. The program is stored in the EEPROM of the microcontroller, which is present in the Arduino board. By this software we put the data and instruction for forward, backward, left, right operation of this system. In android application when we press a button, a corresponding signal is sent through the Bluetooth to Bluetooth module (HC-05) which is connected with the Arduino board. When signal data arrives the Arduino the pin which corresponds to the particular input is set to high. Now that pin gives the output to the motor driver section. Motor driver switches accordingly the data bit, if the data bit is low then the corresponding pin of the motor driver doesn’ t work else high bit then the corresponding pin of the motor driver is on. We have used Arduino IDE version 1.8.8 for writing program for Arduino. There are two steps of the programming. First set up section where we define all the variables. Second loop part where the program runs continuously.



sketch_sep06a | Arduino 1.8.6
File Edit Sketch Tools Help
sketch_sep06a

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here,

}
```

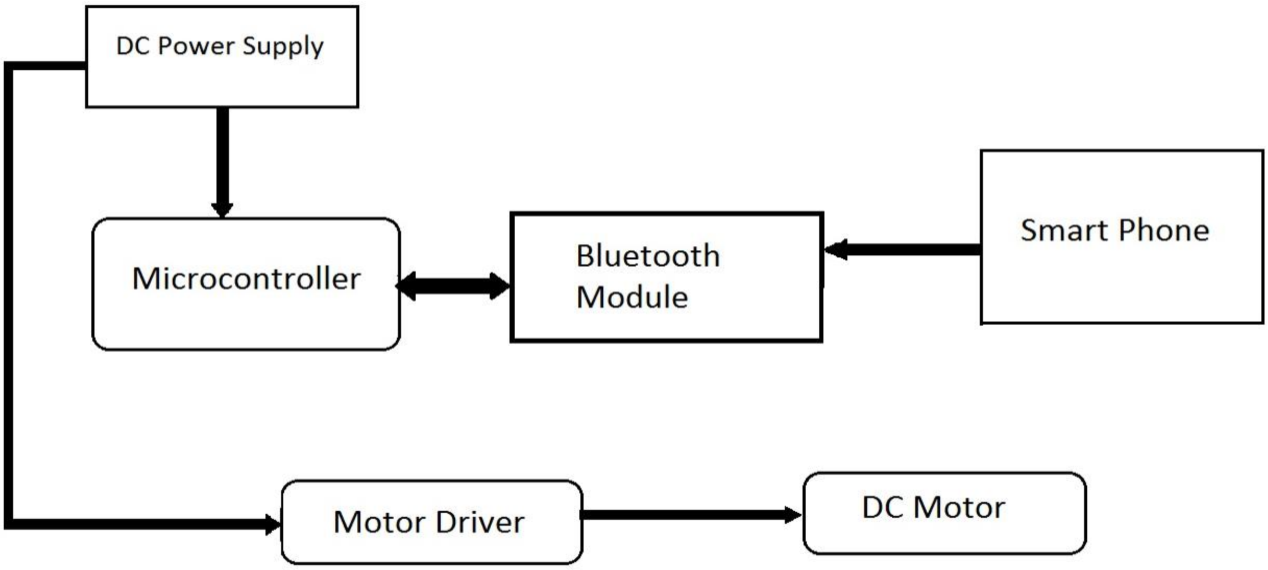
1.8.6
ARDUINO
Genuino
AN OPEN PROJECT WRITTEN, DEBUGGED, AND SUPPORTED BY ARDUINO.CC AND THE ARDUINO COMMUNITY WORLDWIDE
LEARN MORE ABOUT THE CONTRIBUTORS OF ARDUINO.CC on arduino.cc/credits

Done compiling.

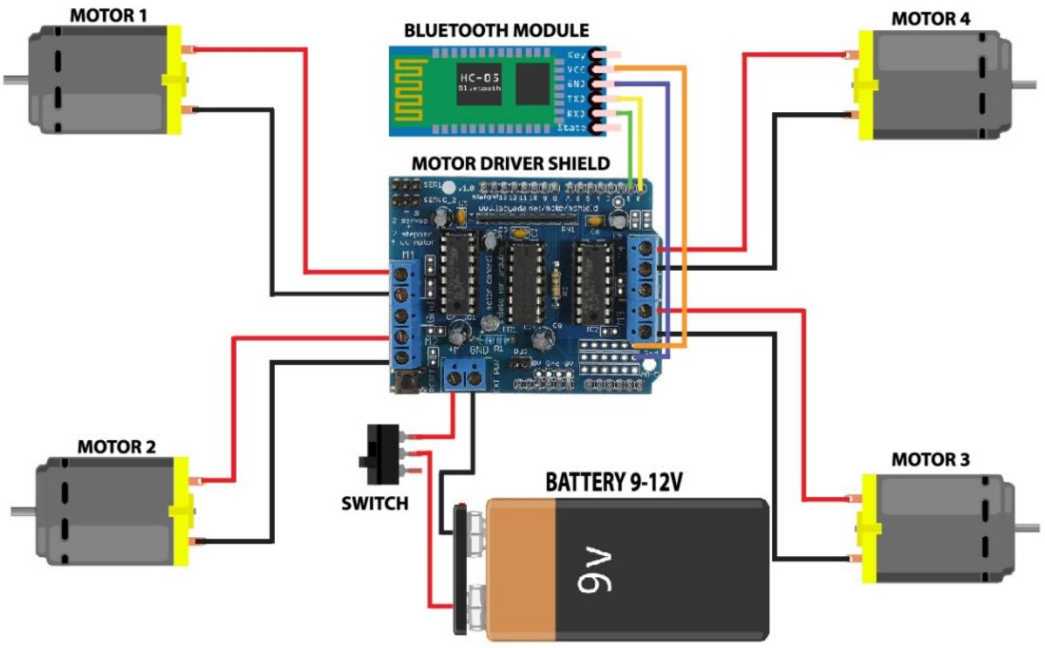
```
Sketch uses 444 bytes (1%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.
```

9
Arduino/Genuino Uno on COM1

Block Diagram



Circuit diagram



CODE

```
#include <AFMotor.h>

//Arduino Bluetooth Controlled Car//

//// Before uploading the code you have to install the necessary library//

//AFMotor Library https://learn.adafruit.com/adafruit-motor-shield/library-install //



//initial motors pin

AF_DCMotor motor1(1);

AF_DCMotor motor2(2);

AF_DCMotor motor3(3);

AF_DCMotor motor4(4);


char command;


void setup()

{

  Serial.begin(9600);  //Set the baud rate to your Bluetooth module.

}


void loop(){

  if(Serial.available() > 0){

    command = Serial.read();

    Stop(); //initialize with motors stoped

    //Change pin mode only if new command is different from previous.
    //Serial.println(command);
    switch(command){
    case 'F':
      forward();
      break;
    case 'B':
       back();
      break;
    case 'L':
      left();
      break;
    case 'R':
      right();
      break;
    }
  }
}

void forward()
{
  motor1.setSpeed(255); //Define maximum velocity
  motor1.run(FORWARD); //rotate the motor clockwise
  motor2.setSpeed(255); //Define maximum velocity
  motor2.run(FORWARD); //rotate the motor clockwise
  motor3.setSpeed(255) ;//Define maximum velocity
  motor3.run(FORWARD); //rotate the motor clockwise
  motor4.setSpeed (255) ;//Define maximum velocity
  motor4.run(FORWARD); //rotate the motor clockwise
}

void back()
{
  motor1.setSpeed(255); //Define maximum velocity
  motor1.run(BACKWARD); //rotate the motor anti-clockwise
  motor2.setSpeed(255); //Define maximum velocity
  motor2.run(BACKWARD); //rotate the motor anti-clockwise
  motor3.setSpeed(255); //Define maximum velocity
  motor3.run(BACKWARD); //rotate the motor anti-clockwise
  motor4.setSpeed(255); //Define maximum velocity
  motor4.run(BACKWARD); //rotate the motor anti-clockwise
}

void left()
{
  motor1.setSpeed(255); //Define maximum velocity
  motor1.run(BACKWARD); //rotate the motor anti-clockwise
  motor2.setSpeed(255); //Define maximum velocity
  motor2.run(BACKWARD); //rotate the motor anti-clockwise
  motor3.setSpeed(255); //Define maximum velocity
  motor3.run(FORWARD); //rotate the motor clockwise
  motor4.setSpeed(255); //Define maximum velocity
  motor4.run(FORWARD); //rotate the motor clockwise
}
```

```
void right()
{
  motor1.setSpeed(255); //Define maximum velocity
  motor1.run(FORWARD); //rotate the motor clockwise
  motor2.setSpeed(255); //Define maximum velocity
  motor2.run(FORWARD); //rotate the motor clockwise
  motor3.setSpeed(255); //Define maximum velocity
  motor3.run(BACKWARD); //rotate the motor anti-clockwise
  motor4.setSpeed(255); //Define maximum velocity
  motor4.run(BACKWARD); //rotate the motor anti-clockwise
}

void Stop()
{
  motor1.setSpeed(0); //Define minimum velocity
  motor1.run(RELEASE); //stop the motor when release the button
  motor2.setSpeed(0); //Define minimum velocity
  motor2.run(RELEASE); //rotate the motor clockwise
  motor3.setSpeed(0); //Define minimum velocity
  motor3.run(RELEASE); //stop the motor when release the button
  motor4.setSpeed(0); //Define minimum velocity
  motor4.run(RELEASE); //stop the motor when release the button
}
```

Step by Step Procedure:

Step1: solder each motor with a black and a red wire and attach them with the chassis.

join left side motor wires together as: red wire --> red wire and black wire --> black wire

similarly join motors on right side together as: red wire --> red wire and black wire --> black wire

step2: Join Wheels to All the Motors

step3: Connect Motors to Motor Drive

Join the red and black terminal of motors on each side, to the motor drive outputs.

Step4: Connect Motor Drive to Arduino

Step5: Join Bluetooth Module to Arduino

Connect bluetooth module (BT) HC-05 to arduino as shown in circuit diagram.

join BT module to arduino as: VCC --> 5V and GND --> GND

step6: Connect Motor Drive to Battery

Try to keep the voltage <= 9 volts. I used a battery of 9 volts. If you use to high voltage (like >= 12 volts, there is a chance that your components will get heated and might burn).

You can also add a switch to start or stop the car as you wish.

Step7: connect the switch to the battery and motor driver shield as shown in the circuit diagram.

Step8: Upload the Code & Download the App

Now compile and upload the given code to the arduino.

Disconnect Rx and Tx of HC-05 from arduino while uploading the code.

After uploading, disconnect the arduino from pc.

Now connect Rx of Hc-05 to Tx of arduino and Tx of Hc-05 to Rx of arduino.

(do not connect these before uploading the code otherwise it may burn your arduino while uploading the code)

Step9: Finally, download the Arduino Bluetooth control app.

Step10: Pair with Bluetooth Module

Start the Car. Check that the LED of Bluetooth module is blinking fast without pairing.

Pair the HC-05 Bluetooth module with your smartphone.
Enter password 1234. (if it not works try 0000)

After pairing open the app and choose HC-05 to pair with. Check the LED of Bluetooth module, its blinking rate would have been very slow now.

Step11: Test Drive

Go to App --> Buttons

Press 1: Car moves forward. (all wheels start moving forward)

Press 1: Car moves in reverse. (all wheels start moving backward)

Press 3: Car turns to left side. (Only right wheels move)

Press 4: Car turns to right side. (Only left wheels move)

CODE DUMPING PROCESS:

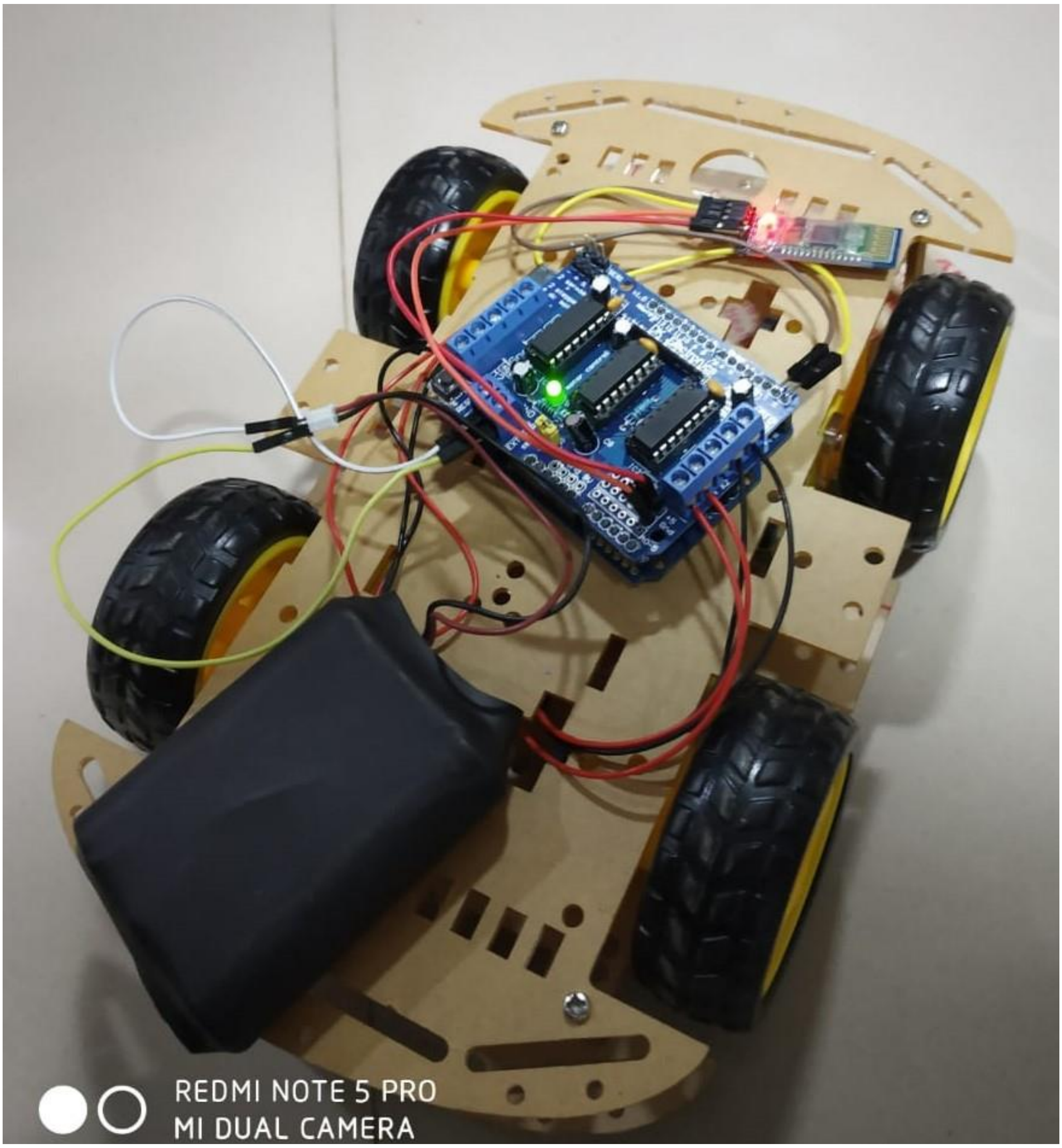
1. Download the Arduino environment
   Download the Arduino software version 1.8.8

2. +Connect the board
   connect the board to a USB port on your computer.
   The power LED should go on.
3. Upload a program
   Select the serial device of the Arduino board from the Tools | Serial Port menu| COM3.

   Push the reset button on the board then click the *Upload* button in the IDE. Wait a few seconds. If successful, the message "Done uploading." will appear in the status bar.

   A few seconds after the upload finishes, you should see the amber (yellow) LED on the board start to blink.

REAL TIME PICTURE



Limitations:

- As the range of the Bluetooth Communication is limited (a maximum of 10 meters for class 2 devices for example) the control range of Bluetooth Controlled Robot is also limited.
- Make sure that sufficient power is provided to all the modules especially the Bluetooth Module. If the power is not sufficient, even though the Bluetooth Module powers on, it cannot transmit data or cannot be paired with other Bluetooth devices.

Applications:

- Low range Mobile Surveillance Devices
- Military Applications (no human intervention)
- Assistive devices (like wheelchairs)
- Home automation