Bluetooth Controlled Arduino Car

Using L293D bluetooth motor shield & remote code uploader *by CRUX.*

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 L293D bluetooth motor driver placed in car is used as receiver, motor driver and code uploader simultaneously. 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.

A. Parts List:

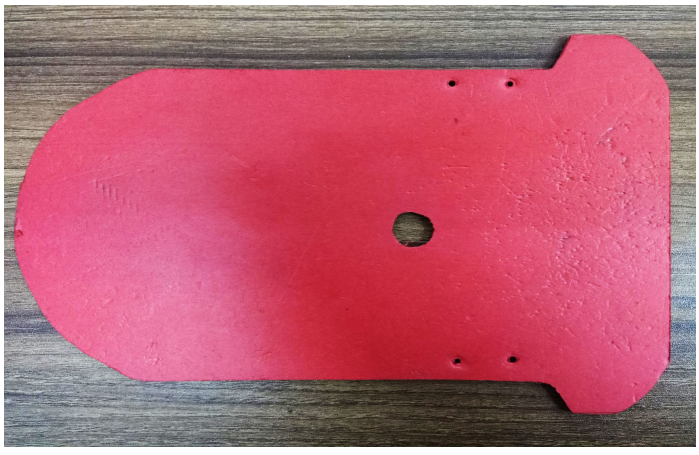
- Arduino uno r3.
- L293D bluetooth motor shield.
- 12v gear motor , 300-600 rpm.
- 9v/12v battery.
- Caster Wheel.
- PVC board.
- Connector wire.
- Rubber wheel.

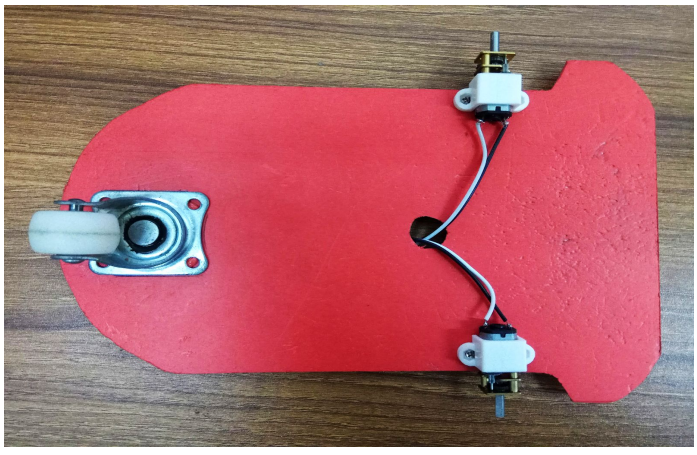
B. Softwares:

- Bluino Loader - Arduino IDE.
- Bluetooth RC Controller.

Step 1. Building chassis:

Cut down the pvc board for the chassis as you like and attach the caster wheel and mount those motors under the chassis as the picture shown below. Anti-cutter is very useful for cutting pvc boards.

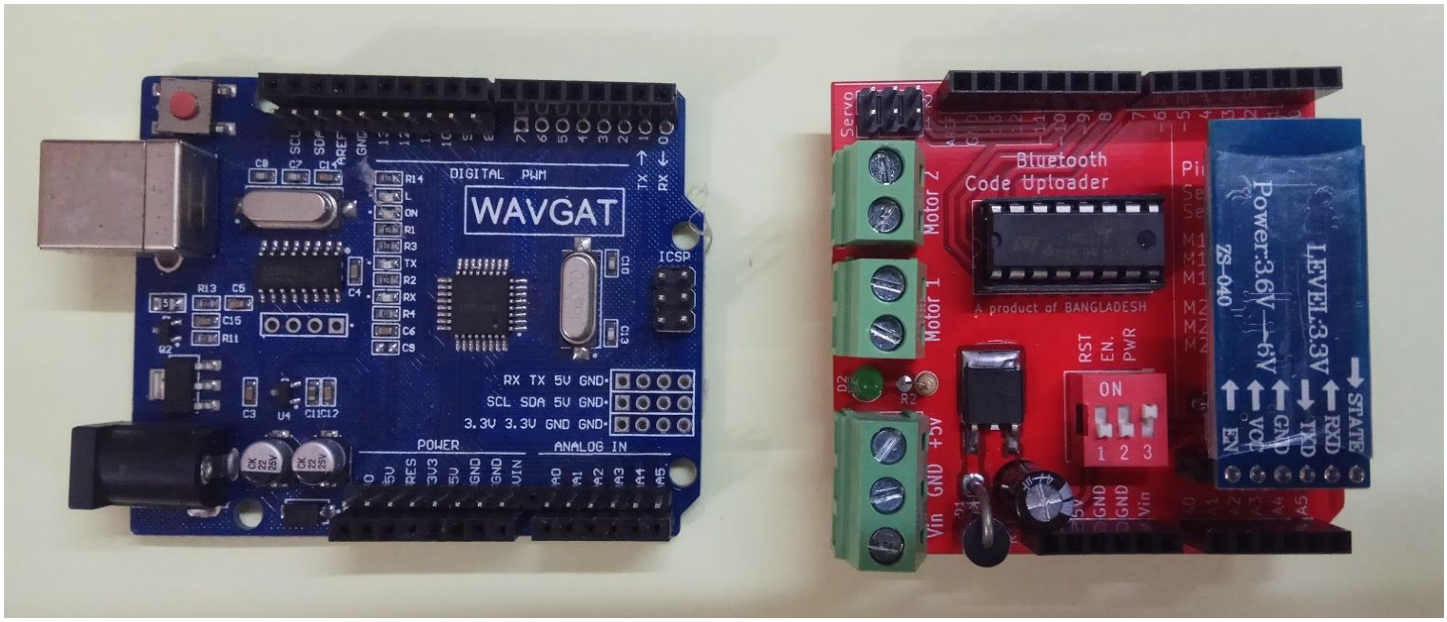


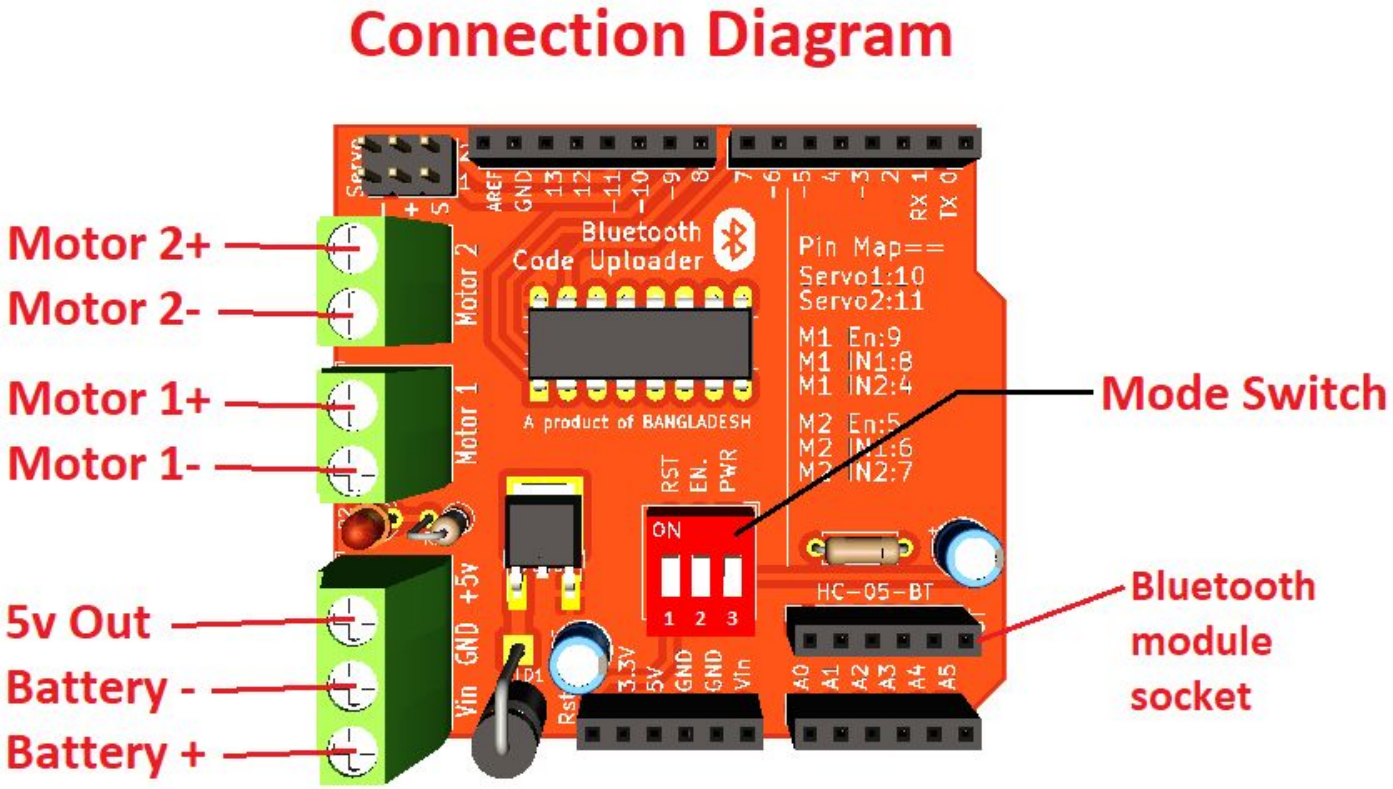


Step 2. Setup bluetooth motor shield:

It wouldn't be an Arduino project without an Arduino! Actually, any microcontroller that is compatible with the Arduino IDE and supports PWM and UART will do the trick, but to keep things simple I recommend the Arduino UNO R3.

Connect motor and battery with the shield as shown in the picture below.

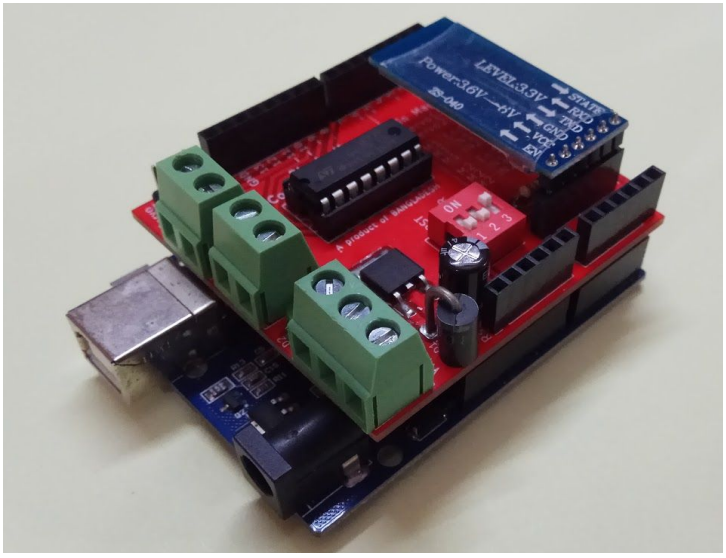


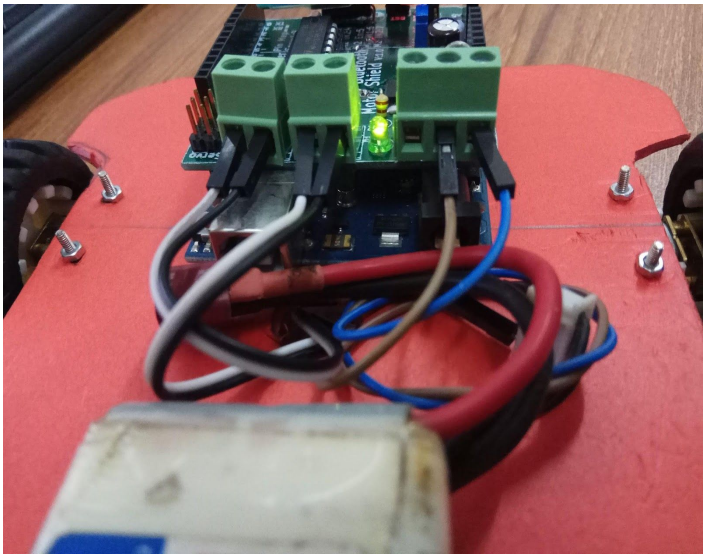


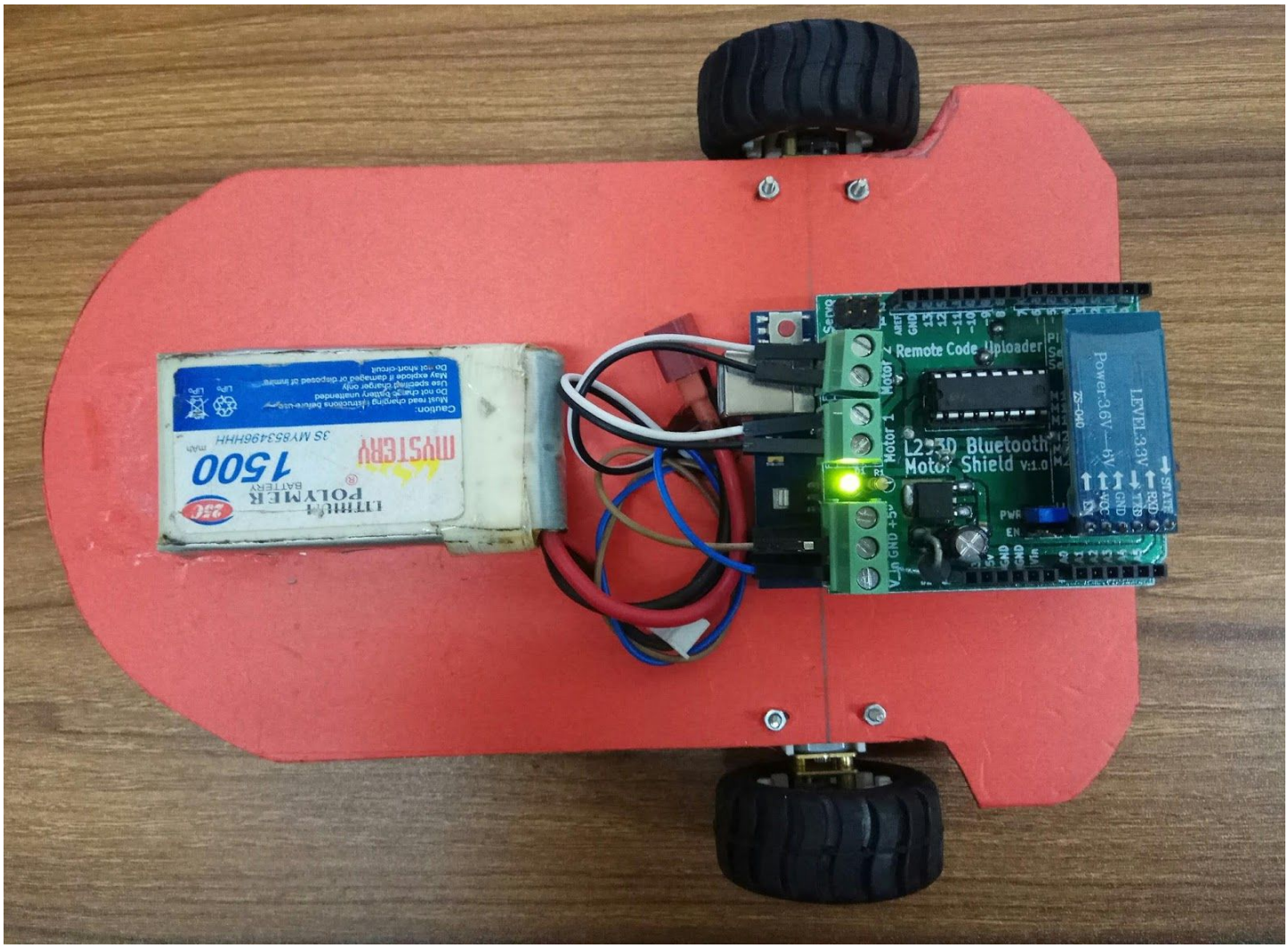
Mode Switch				
Switch			Operation	Note
1	2	3		
Off	Off	Off	USB Code Uploading Mode.	All 3 switches must be turned off during code uploading from arduino IDE using usb cable from computer.
Off	Off	On	Bluetooth Communication Mode.	In this mode the bluetooth module can be used as a normal communication device to communicate with other bluetooth devices.
On	Off	On	Wireless Code Uploading Mode.	For wireless code uploading from bluino loader, the baud rate of the bluetooth module must be 115200.
Off	On	On	HC-05 AT Command Mode.	To enable AT command mode and use AT commands or change the internal setting of the HC-05, this mode is used.

Step 3. Building the car:

Attach the shield on arduino and mount on the car chassis. Connect motor wires and battery with the shield. Attach rubber wheels with the motor. Now the car is ready for uploading code.







Step 4. Code uploading:

At first download or copy the code. Keep in mind, baud rate in the code always will be 115200.

Code:

```
// Bluetooth Arduino Car using L293D Bluetooth Motor Shield
// Programmed by CRUX


const int motor1_IN1 = 8;
const int motor1_IN2 = 4;
const int motor1_Enable = 9;
const int motor2_IN1 = 6;
const int motor2_IN2 = 7;
const int motor2_Enable = 5;
byte serialA;

void setup() {
  Serial.begin(115200);
```

```
  pinMode(motor1_IN1, OUTPUT);
  pinMode(motor1_IN2, OUTPUT);
  pinMode(motor1_Enable, OUTPUT);
  pinMode(motor2_IN1, OUTPUT);
  pinMode(motor2_IN2, OUTPUT);
  pinMode(motor2_Enable, OUTPUT);
}

void loop() {

if (Serial.available() > 0) {serialA = Serial.read();Serial.println(serialA);}

  switch (serialA) {
   // forward
  case 'F':
      digitalWrite(motor1_IN1, HIGH);
      digitalWrite(motor1_IN2, LOW);
      digitalWrite(motor2_IN1, LOW);
      digitalWrite(motor2_IN2, HIGH);
      digitalWrite(motor1_Enable, HIGH);
      digitalWrite(motor2_Enable, HIGH);
      break;

       // left
   case 'L':
      digitalWrite(motor1_IN1, HIGH);
      digitalWrite(motor1_IN2, LOW);
      digitalWrite(motor2_IN1, HIGH);
      digitalWrite(motor2_IN2, LOW);
      digitalWrite(motor1_Enable, HIGH);
      digitalWrite(motor2_Enable, LOW);
      break;

   // right
   case 'R':
      digitalWrite(motor1_IN1, LOW);
      digitalWrite(motor1_IN2, HIGH);
      digitalWrite(motor2_IN1, LOW);
      digitalWrite(motor2_IN2, HIGH);
      digitalWrite(motor1_Enable, LOW);
      digitalWrite(motor2_Enable, HIGH);
      break;
```

```
// forward left
case 'G':
   digitalWrite(motor1_IN1, HIGH);
   digitalWrite(motor1_IN2, LOW);
   digitalWrite(motor2_IN1, LOW);
   digitalWrite(motor2_IN2, LOW);
   digitalWrite(motor1_Enable, HIGH);
   digitalWrite(motor2_Enable, HIGH);
   break;


// forward right
case 'I':
   digitalWrite(motor1_IN1, LOW);
   digitalWrite(motor1_IN2, LOW);
   digitalWrite(motor2_IN1, LOW);
   digitalWrite(motor2_IN2, HIGH);
   digitalWrite(motor1_Enable, HIGH);
   digitalWrite(motor2_Enable, HIGH);
   break;


// backward left
case 'H':
   digitalWrite(motor1_IN1, HIGH);
   digitalWrite(motor1_IN2, LOW);
   digitalWrite(motor2_IN1, HIGH);
   digitalWrite(motor2_IN2, LOW);
   digitalWrite(motor1_Enable, HIGH);
   digitalWrite(motor2_Enable, HIGH);
   break;

// backward right
case 'J':
   digitalWrite(motor1_IN1, LOW);
   digitalWrite(motor1_IN2, HIGH);
   digitalWrite(motor2_IN1, LOW);
   digitalWrite(motor2_IN2, HIGH);
   digitalWrite(motor1_Enable, HIGH);
   digitalWrite(motor2_Enable, HIGH);
   break;
```

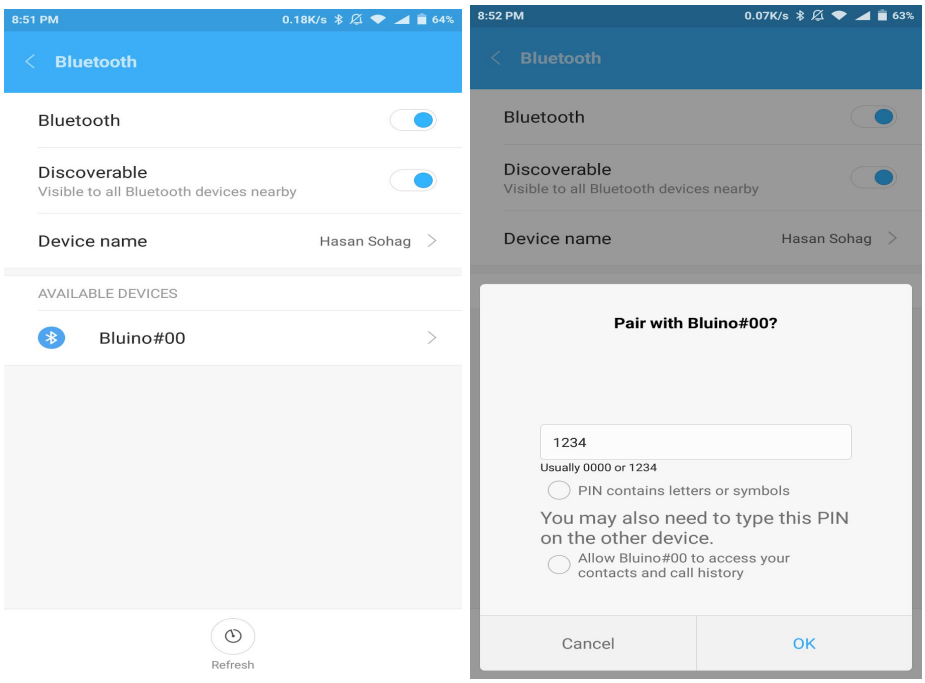
```
  // backward
  case 'B':
     digitalWrite(motor1_IN1, LOW);
     digitalWrite(motor1_IN2, HIGH);
     digitalWrite(motor2_IN1, HIGH);
     digitalWrite(motor2_IN2, LOW);
     digitalWrite(motor1_Enable, HIGH);
     digitalWrite(motor2_Enable, HIGH);
     break;

  // Stop
   case 'S':
     digitalWrite(motor1_IN1, LOW);
     digitalWrite(motor1_IN2, LOW);
     digitalWrite(motor2_IN1, LOW);
     digitalWrite(motor2_IN2, LOW);
     digitalWrite(motor1_Enable, LOW);
     digitalWrite(motor2_Enable, LOW);
    }
  }
```

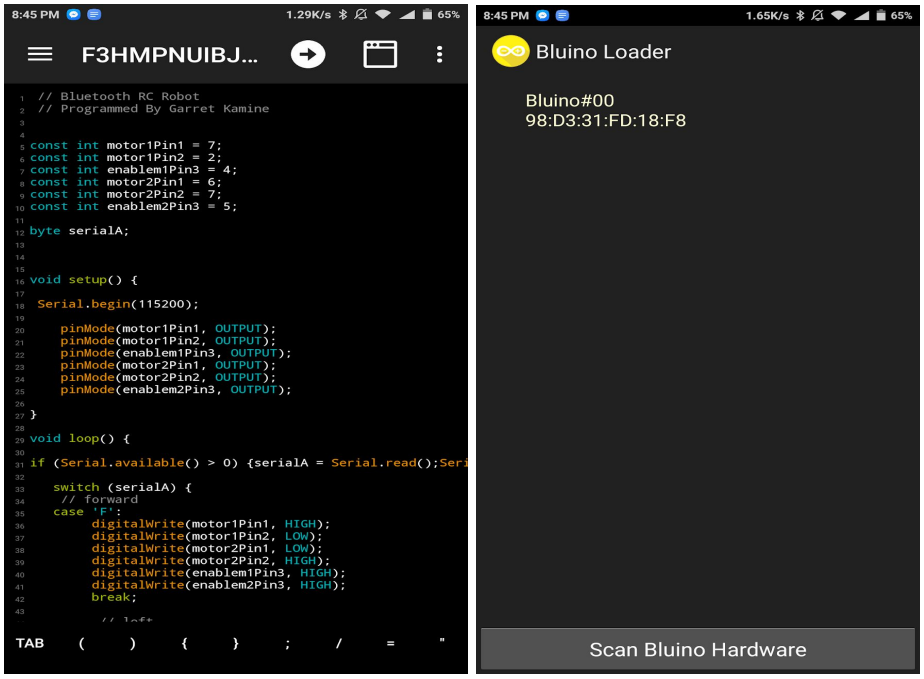
....

Now paste the code in the arduino IDE or download Bluino Loader - Arduino IDE in your android device and open the code in Bluino Loader. If you want to upload the code from your laptop just use the Arduino IDE instead of Bluino loader.

With the battery connected, the Bluetooth module should be blinking a red LED. Follow these steps to pair and connect your device and upload the code:

1. Turn Bluetooth "ON" in the device settings
2. Scan for Bluetooth devices. Be sure your device is set to "Visible".
3. Select "Bluino#00", password- 1234.
4. Now after loading the code in Bluino Loader tap on the arrow button up top.
5. Tap on "Bluino#00".
6. If everything is ok, the code will be uploaded.

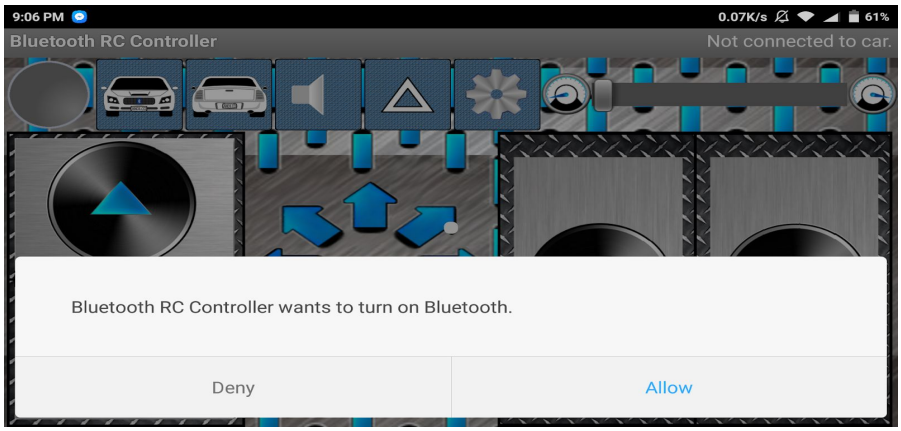


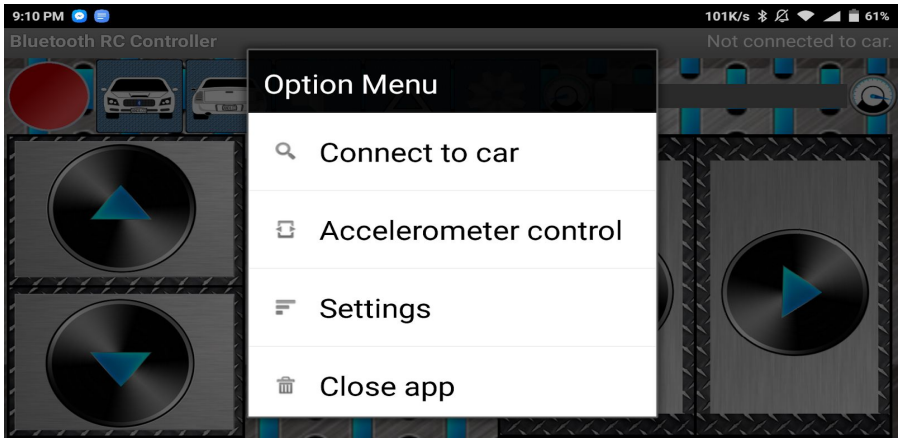


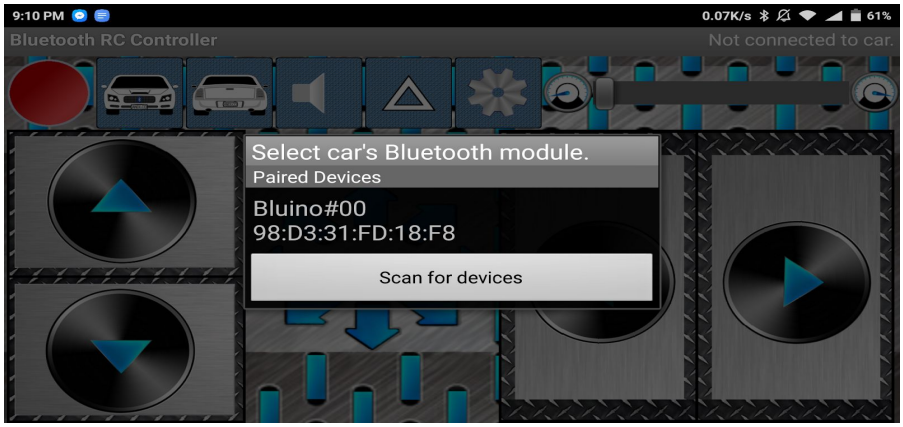
Step 5. Pair your android device and control the car:

Download [Bluetooth RC Controller] app in your android device and install and follow those stapes.

1. Open the "Arduino Bluetooth RC Controller" app.
2. Bluetooth permission request: "Yes" if prompted.
3. Touch the "Settings" button.
4. Touch "Connect to car".
5. Tap on “Bluino#00”.
6. Now you are ready to go.







Step 6. Check Proper Operation

It's easy to mix up a wire or two when building a project like this, so don't be surprised if it only partially works (or not at all) on your first attempt.

If your project responds to commands sent from your Android device, great! Just verify the commands sent correspond with the operation of the motors (for example, the "Forward" button should make both motors spin full speed in the same direction). If a command doesn't match the operation (for example, the "Forward" button makes one motor turn clockwise and the other turn counterclockwise) just exchange the motor wires in between them and look for proper orientation.