

Bluetooth Controlled Robot

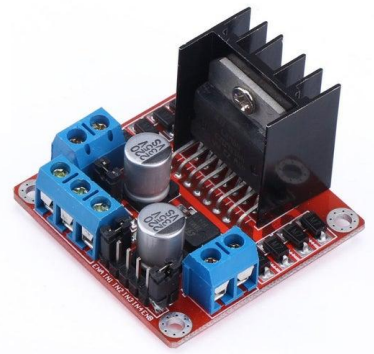
Maanit Kalsotra

<https://www.kalsotra.com>

Features

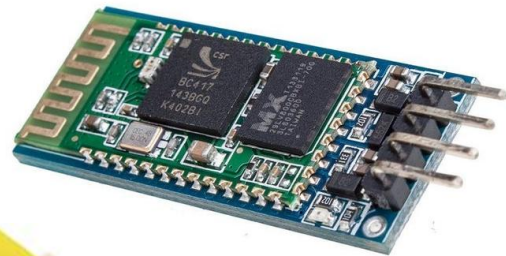
- ▶ The Robot is controlled by android app through Bluetooth
- ▶ The range of the robot is 10-20 metre
- ▶ The robot is completely built using recycled or reused parts
- ▶ The robot is working on a 7.2 volts source but can also work in the range of 6-12 volts
- ▶ The robot has clock Rate of 16 MHz and computing speed of 16 MIPS
- ▶ The robot also features a SRAM of 2KB and EEPROM of 1KB





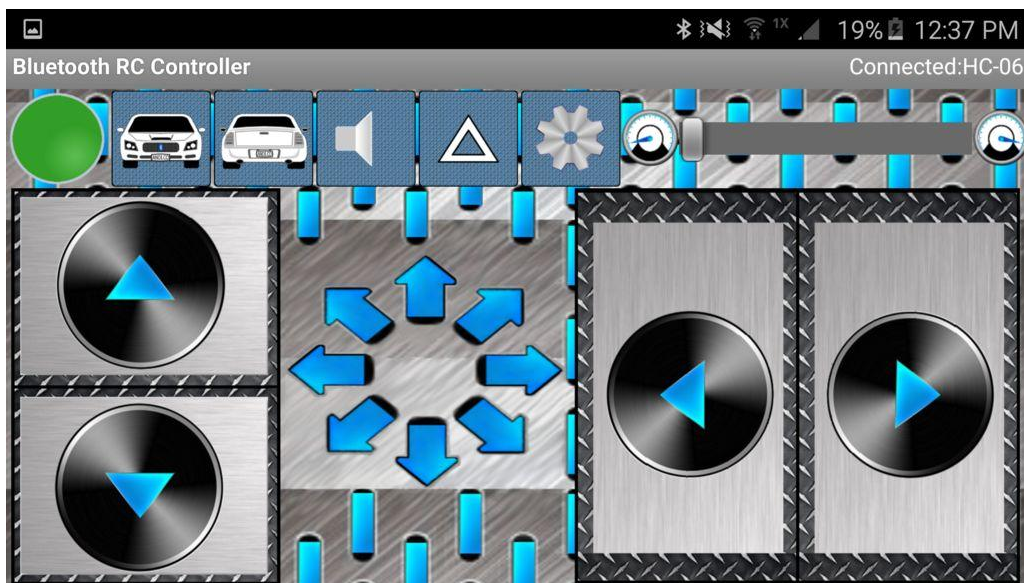
Hardware Used

- ▶ Atmega 328P microcontroller
- ▶ L298N motor driver H-bridge
- ▶ HC-05 Bluetooth Module
- ▶ Old wood piece used as double-decker chassis
- ▶ 18650 lithium ion battery from an old laptop battery pack
- ▶ Geared motors and wheels
- ▶ Pen refill used as an antenna



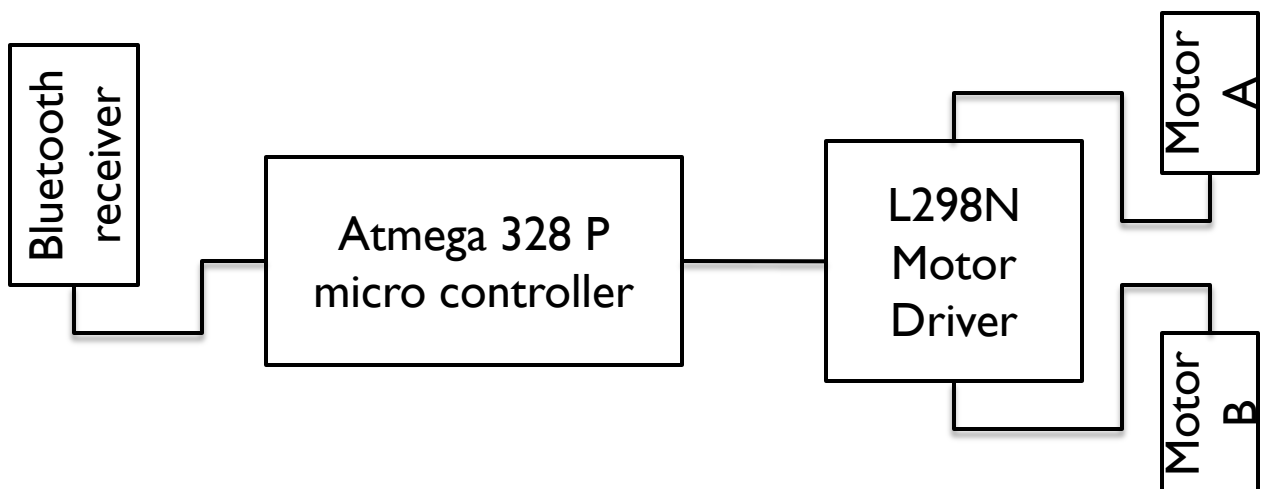
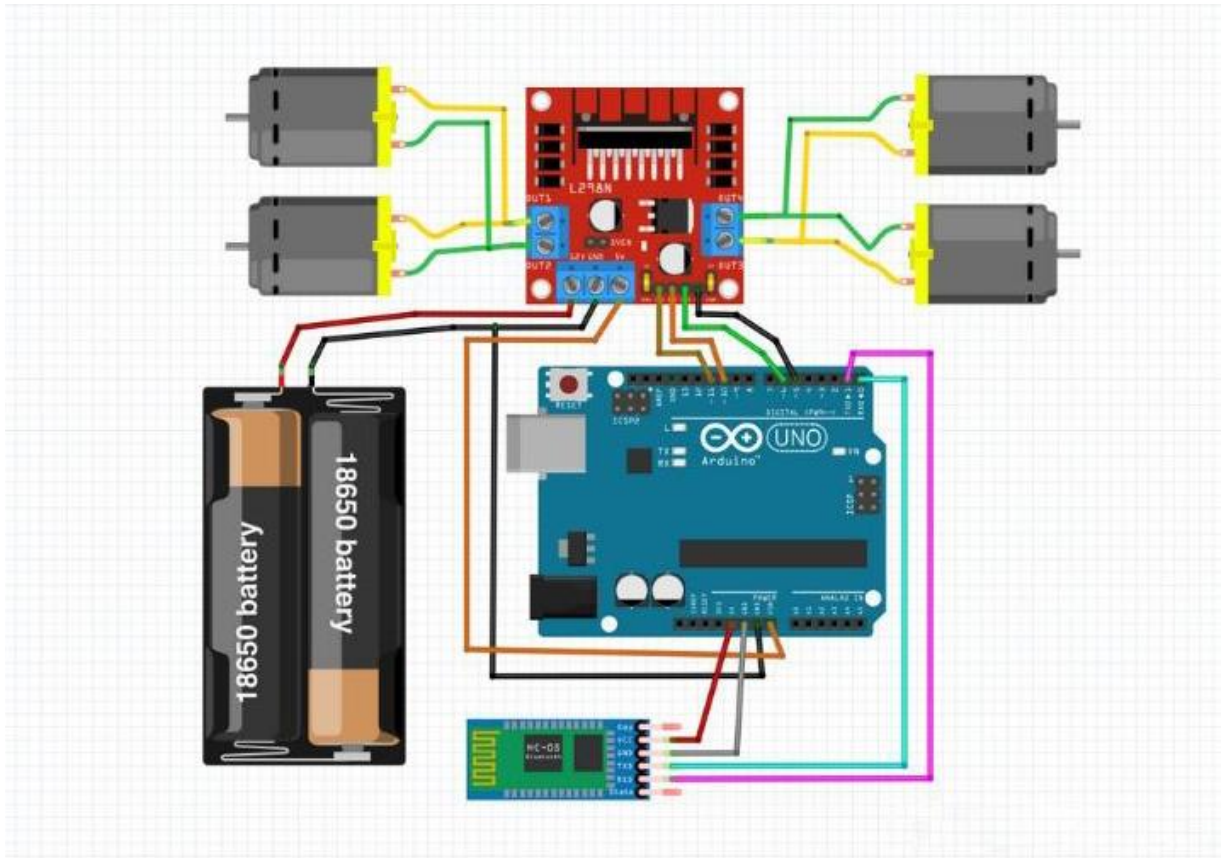
Software Used

- ▶ Android application
android application based on Java model
to control the robot using single character
serial communication.



- ▶ IDE(Integrated desktop environment)
IDE for programming the Atmega328P
chip

Construction

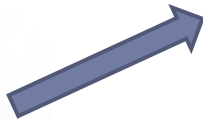
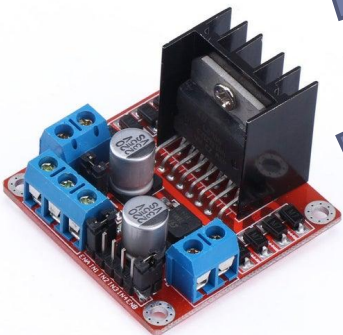
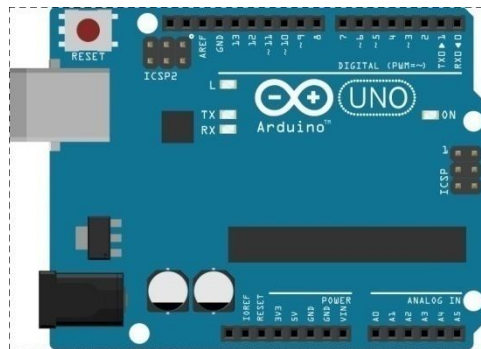
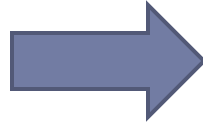


Working

- ▶ The Android application will send a single character as per user selection
- ▶ The character received by the Bluetooth module of robot will send the character to the micro controller
- ▶ The micro controller will process the character to recognise its function and send the required command to the motor driver
- ▶ The motor driver will send the power to the motors of robot
- ▶ And the Robot will work as per the given command



Working (Contd.)



Forward
Backward
Left
Right

Working of each Motor

- Movement in Different combinations by two motors enables the robot to move in all direction

	Motor A	Motor B	Direction
Forward	✓	✓	↑
Backward	✓	✓	↓
Left	✗	✓	↑
Right	✓	✗	↑



Code

```
/*  
Code Name: Arduino Bluetooth Control Car  
Code URI: https://github.com/kryg0n/arduino-bluetooth-robot  
Author: Maanit Kalsotra  
Description: This program is used to control a robot  
using a app  
that communicates with Arduino through a bluetooth  
module.  
App URI: https://bit.ly/2B1MAea  
*/  
  
#define in1 5 //L298n Motor Driver pins.  
#define in2 6  
#define in3 10  
#define in4 11  
#define LED 13  
int command; //Int to store app command state.  
int Speed = 204; // 0 - 255.  
int Speedsec;  
int buttonState = 0;  
int lastButtonState = 0;  
int Turnradius = 0; //Set the radius of a turn, 0 - 255  
Note:the robot will malfunction if this is higher  
than int Speed.
```



Code (Contd.)

```
int brakeTime = 45;
int brkonoff = 1; //1 for the electronic braking system,
    0 for normal.
void setup() {
    pinMode(in1, OUTPUT);
    pinMode(in2, OUTPUT);
    pinMode(in3, OUTPUT);
    pinMode(in4, OUTPUT);
    pinMode(LED, OUTPUT); //Set the LED pin.
    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.
        switch (command) {
            case 'F':
                forward();
                break;
            case 'B':
                back();
                break;
        }
    }
}
```



Code (Contd.)

```
case '2':  
    Speed = 153;  
    break;  
case '3':  
    Speed = 165;  
    break;  
case '4':  
    Speed = 178;  
    break;  
case '5':  
    Speed = 191;  
    break;  
case '6':  
    Speed = 204;  
    break;  
case '7':  
    Speed = 216;  
    break;  
case '8':  
    Speed = 229;  
    break;  
case '9':  
    Speed = 242;  
    break;
```



Code (Contd.)

```
case 'q':
    Speed = 255;
    break;
}
Speedsec = Turnradius;
if (brkonoff == 1) {
    brakeOn();
} else {
    brakeOff();
}
}

void forward() {
    analogWrite(in1, Speed);
    analogWrite(in3, Speed);
}

void back() {
    analogWrite(in2, Speed);
    analogWrite(in4, Speed);
}
```



Code (Contd.)

```
void left() {
    analogWrite(in3, Speed);
    analogWrite(in2, Speed);
}

void right() {
    analogWrite(in4, Speed);
    analogWrite(in1, Speed);
}

void forwardleft() {
    analogWrite(in1, Speedsec);
    analogWrite(in3, Speed);
}

void forwardright() {
    analogWrite(in1, Speed);
    analogWrite(in3, Speedsec);
}

void backright() {
    analogWrite(in2, Speed);
    analogWrite(in4, Speedsec);
}

void backleft() {
    analogWrite(in2, Speedsec);
    analogWrite(in4, Speed);
```



Code (Contd.)

```
}
```

```
void Stop() {  
    analogWrite(in1, 0);  
    analogWrite(in2, 0);  
    analogWrite(in3, 0);  
    analogWrite(in4, 0);  
}
```

