RoboVR

SRB MUSICIANS

About the game:

For Musicians at RoboVR, participants have to build a minimum of one robot per team. The robot can be remote-controlled or autonomous.

Components and its Specifications:

Sr. No.	**Components**	**Specifications**
1.	Johnson motor x4	100 rpm*
2.	Android Mobile	With a Bluetooth Control application
3.	LIPO Battery	11.1 v , 2000mAh *
4.	Chassis	Aluminum Sheet (5mm thick)
5.	Wheels x4	Off-road type
6.	Clamp	Aluminum Sheet (2mm thick)
7.	Bluetooth HC05	v2.0+EDR
8.	Arduino UNO	ATmega328P
9.	Jumper wires	M-M, M-F
10.	L298N motor driver	2A

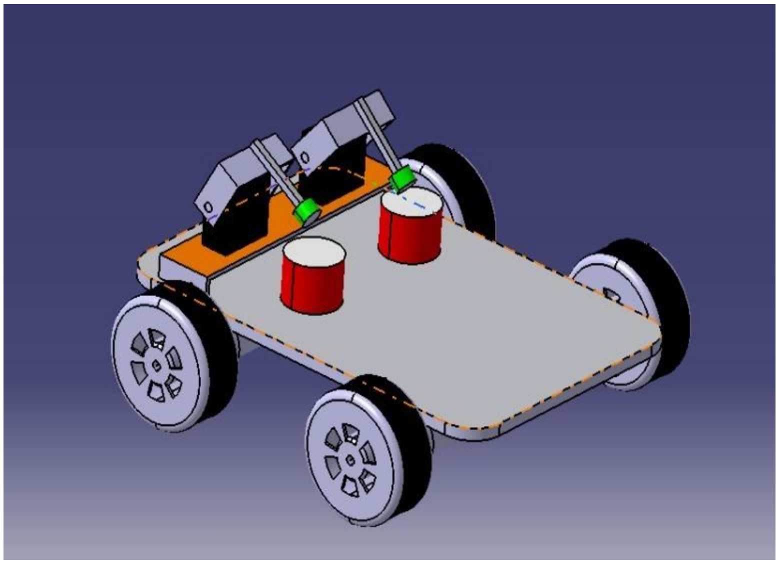
Robot Details:

Robot Control: Wireless
Robot Drive: 4 Wheel Drive

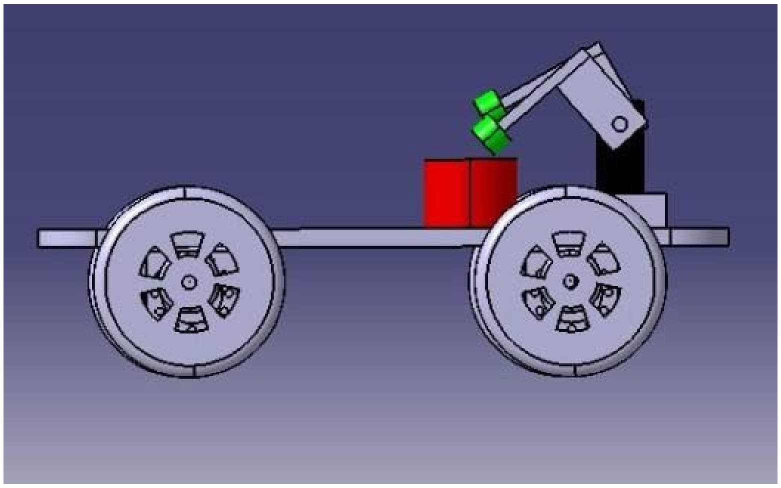
Mechanical Design:

Robot with Drum

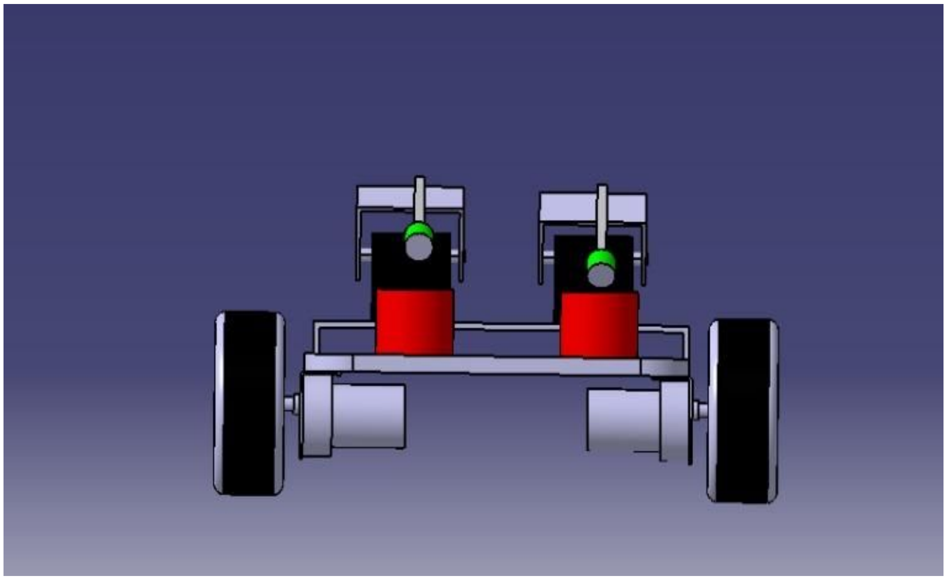
Isometric View



Side View

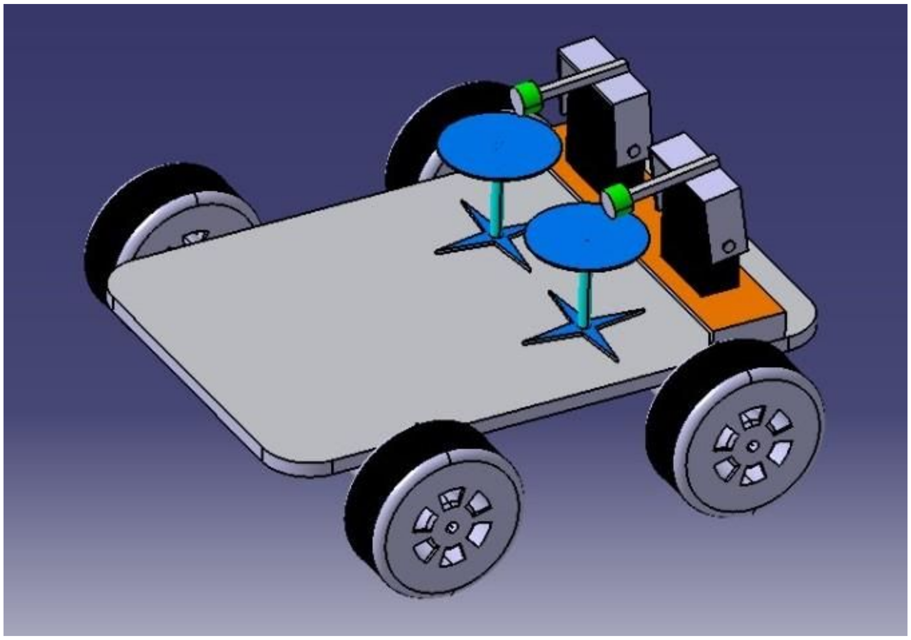


Front View

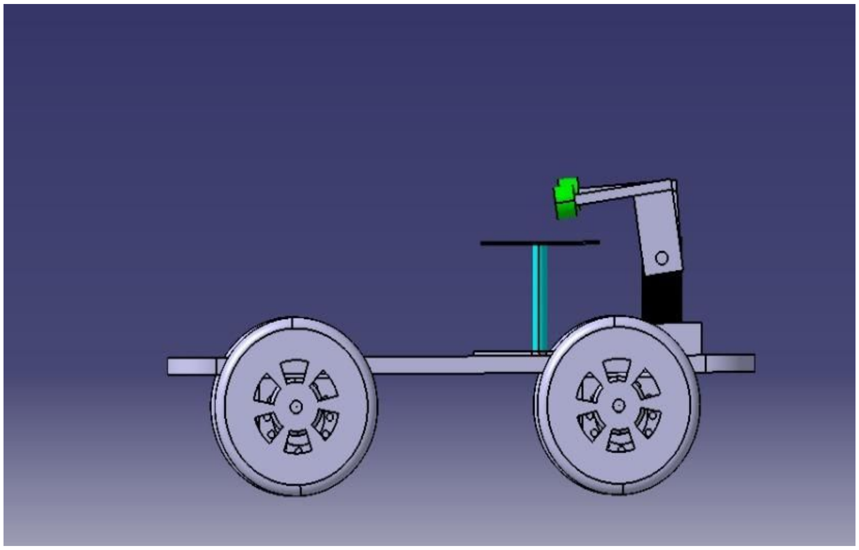


A robot with Crash/Ride Cymbal

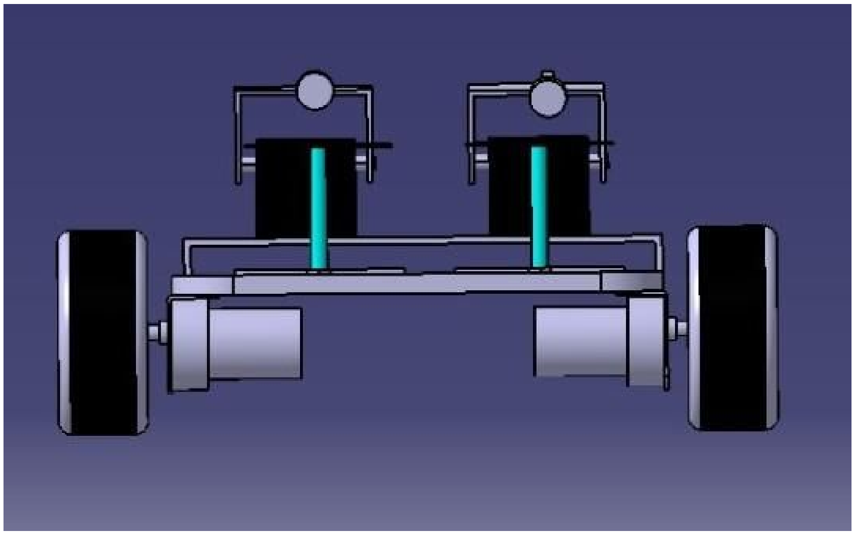
Isometric View



Side View



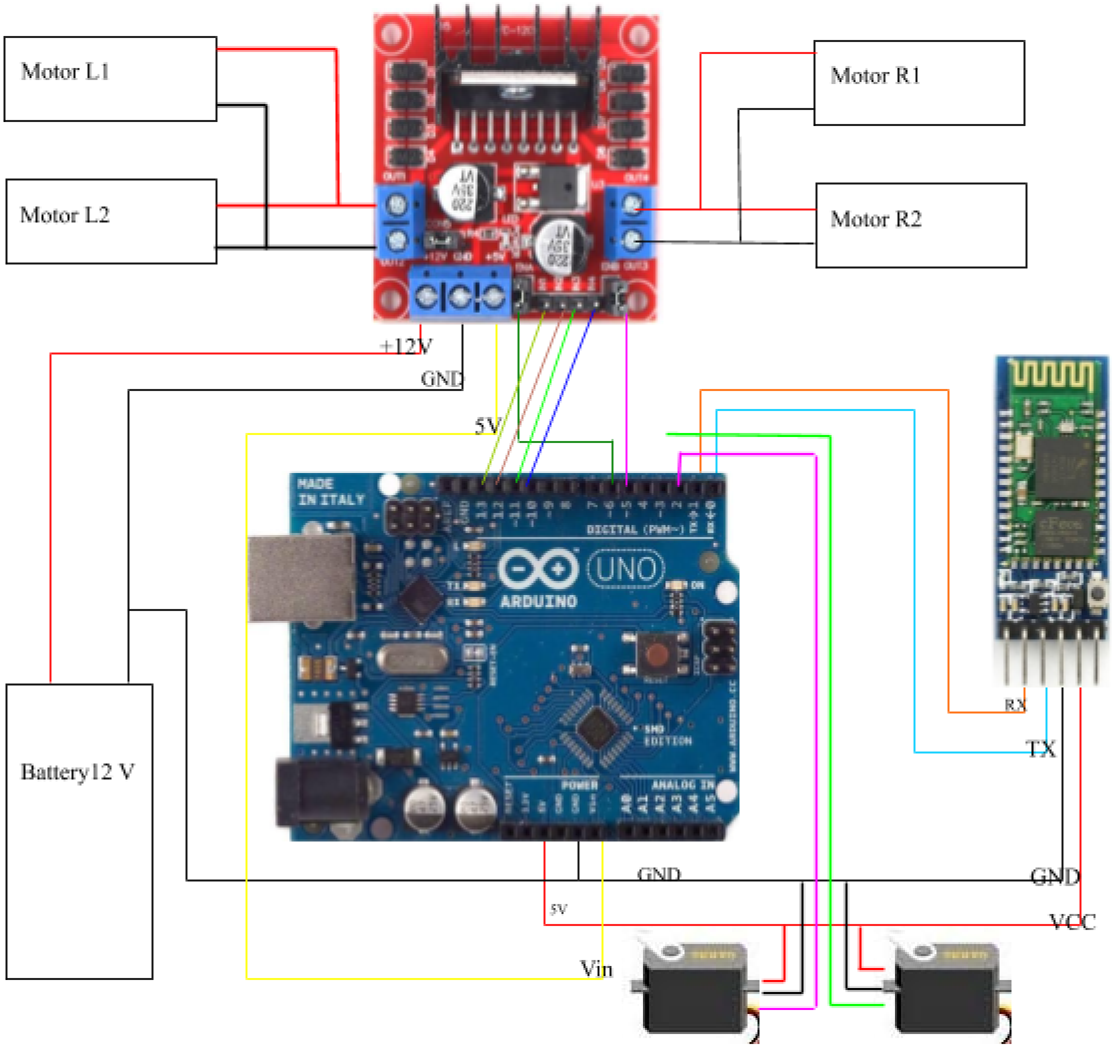
Front View



Working:

The robot is controlled with remotely operated from a mobile phone. For this Arduino UNO is used as microcontroller and Bluetooth module HC05 is mounted on Arduino for wireless operating. Motor driver L298N is used to amplify current. The two musician bots will perform the tasks by using the servo motors as a hand to hit the instrument. The circuit diagram and Arduino code calibrating the servo motor is given below.

Connections:



Codes:

```
for(i=0;i<180;i++)
{
servo.write(i);
}
```

This is the **basic code for rotate any servo motor** attached to any Arduino board.

But calibrating the rotating degrees and deciding which motors should run during the movement of each leg is the most tricky part of coding. It can be done by another Sketch called (Servo_Test). By testing the degree of rotation of each motor through serial communication through the Arduino board, we can calibrate every motor.

Servo Testing Arduino Code:

```
String readString;
#include <Servo.h>
Servo myservo;
void setup()
{
  Serial.begin(9600);
  myservo.attach(27);  //the pin for the servo control
  Serial.println("servo-test"); // so I can keep track of what is loaded
}

void loop()
{
  while (Serial.available())
  {
    delay(11);
    if (Serial.available() >0)
    {
      char c = Serial.read();  //gets one byte from serial buffer
      readString += c; //makes the string readString
    }
  }

  if (readString.length() >0)
  {
    Serial.println(readString);  //so you can see the captured string
    int n;
    char carray[6]; //converting string to number
    readString.toCharArray(carray, sizeof(carray));
    n = atoi(carray);
    myservo.writeMicroseconds(n); // for microseconds
    myservo.write(n); //for degees 0-180
    readString="";
  }
}
```