ROBOTIC CAR PROJECT

- AIM: Robotic Car using Arduino UNO Board

1. Autonomous Robot (Line Follower)

- **Components Required:**
  1. Arduino UNO Board
  2. Arduino USB Cable
  3. Mini-Breadboard
  4. 2 IR Sensors
  5. L293D Motor Driver Module
  6. Robotics Kit Part: Robo Chasis, 2 DC Motor, Clamps, Screw Sets, Screw Driver, Nut Driver, 2 Wheels.
  7. Jumper Wires M-M & M-F.

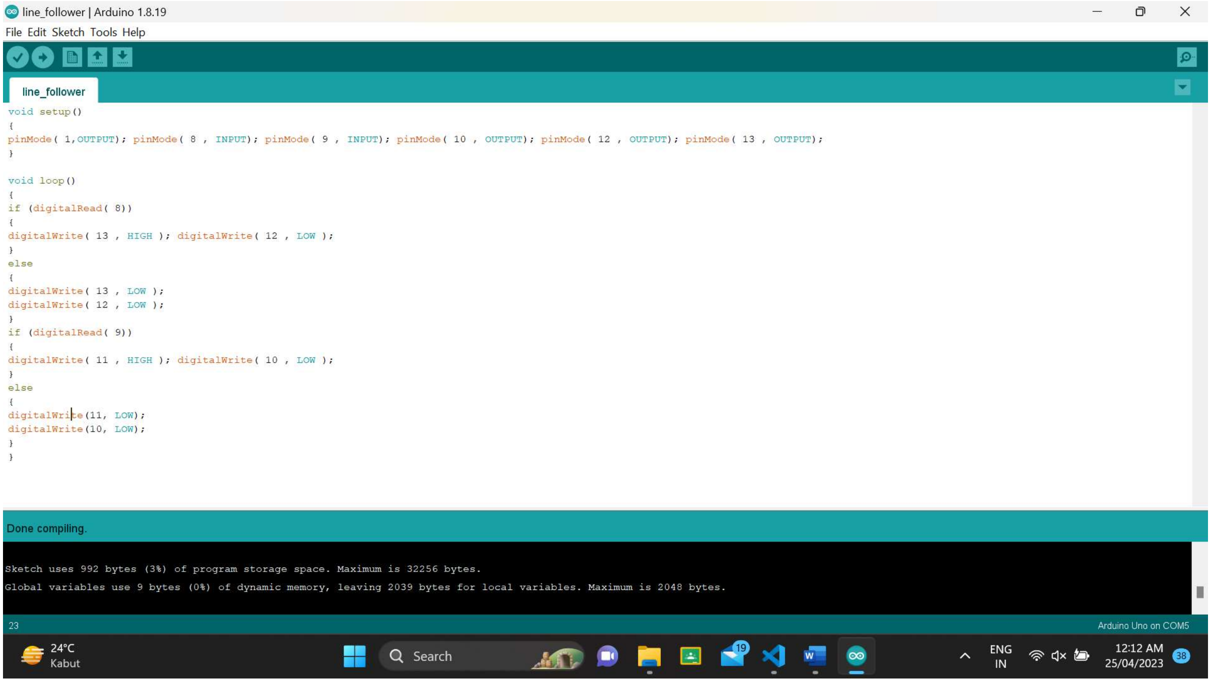
- **Steps to Follow:**

1. Please assemble the Robot as instructed by Instructor.
2. Connect Vout of Left Sensor to Pin 8 on Arduino.
3. Connect Vcc of Left Sensor to +ve (5V) Common Terminal on Breadboard.
4. Connect GND of Left Sensor to –ve Terminal of Bread Board (GND) on Breadboard.
5. Connect Vout of Right Sensor to Pin 9 on Arduino.
6. Connect Vcc of Right Sensor to +ve (5V) Common Terminal on Breadboard.
7. Connect GND of Right Sensor to –ve Terminal of Bread Board (GND) on Breadboard.
8. Connect 5v from L293D Motor driver Module to +ve (5V) Common Terminal on Breadboard.
9. Connect GND of Motor Driver to –ve Terminal of Bread Board (GND) Terminal on Breadboard.
10. Connect 12v from L293D Motor driver Module to +ve (5V) Common Terminal on Breadboard.
11. Connect M1-A/Pin 1 from Motor Driver Module to Pin 13 on Arduino.
12. Connect M1-B/Pin 2 from Motor Driver Module to Pin 12 on Arduino.
13. Connect M2-A/Pin 3 from Motor Driver Module to Pin 11 on Arduino.
14. Connect M2-B/Pin 4 from Motor Driver Module to Pin 10 on Arduino.
15. Connect Left Motor Red & Black Wire on R & B M1 Terminal on Motor Driver Module.
16. Connect Right Motor Red & Black Wire on R & B M2 Terminal on Motor Driver Module.
17. Connect a Jumper Wire from Arduino GND to Bread Board –ve(GND) Terminal Strip.
18. Connect a Jumper Wire from Arduino +5V to Bread Board +ve (5V) Terminal Strip.
19. Create Sketch/Program for Line Follower Robot Project.
20. Connect the USB cable with Arduino & Computer.
21. Click on Upload Option to download the program in Arduino.
22. Disconnect USB cable from Arduino & Connect 9V DC Battery to Arduino by using battery connector and then check your robot.

Anshit Kumar Srivastava
ECE

- **Program**

```
void setup()
{
pinMode( 1,OUTPUT);
pinMode( 8 , INPUT);
pinMode( 9 , INPUT);
pinMode( 10 , OUTPUT);
pinMode( 12 , OUTPUT);
pinMode( 13 , OUTPUT);
}
void loop()
{
if (digitalRead( 8))
{
digitalWrite( 13 , HIGH );
digitalWrite( 12 , LOW );
}
else
{
digitalWrite( 13 , LOW );
digitalWrite( 12 , LOW );
}
if (digitalRead( 9))
{
digitalWrite( 11 , HIGH );
digitalWrite( 10 , LOW );
}
else
{
digitalWrite(11, LOW);
digitalWrite(10, LOW);
}
}
```



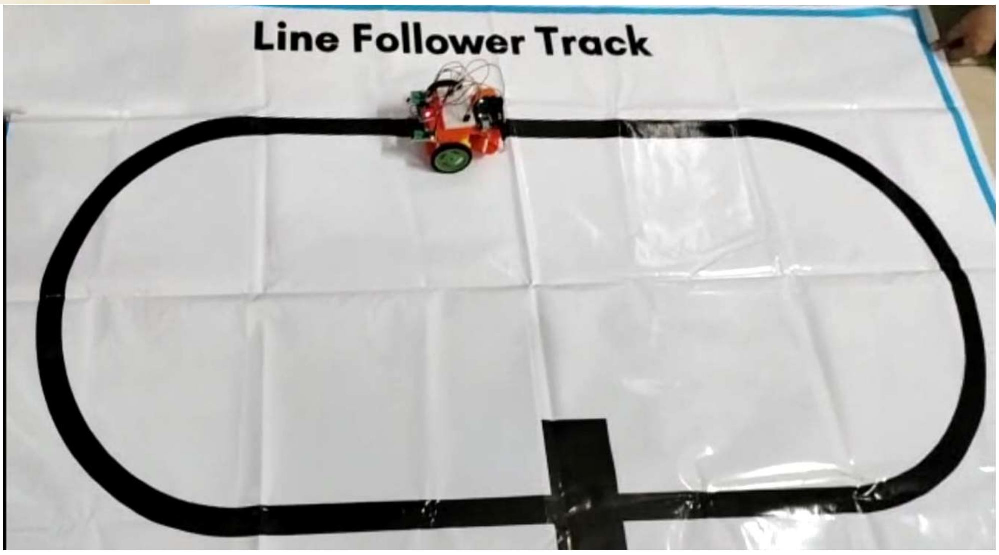
```
void setup()
{
pinMode( 1,OUTPUT); pinMode( 8 , INPUT); pinMode( 9 , INPUT); pinMode( 10 , OUTPUT); pinMode( 12 , OUTPUT); pinMode( 13 , OUTPUT);
}

void loop()
{
if (digitalRead( 8))
{
digitalWrite( 13 , HIGH ); digitalWrite( 12 , LOW );
}
else
{
digitalWrite( 13 , LOW );
digitalWrite( 12 , LOW );
}
if (digitalRead( 9))
{
digitalWrite( 11 , HIGH ); digitalWrite( 10 , LOW );
}
else
{
digitalWrite(11, LOW);
digitalWrite(10, LOW);
}
}
```

Done compiling.

```
Sketch uses 992 bytes (3%) 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.
```

➢ <u>Picture of Line Follower-</u>



Anshit Kumar Srivastava
ECE

2. Obstacle Avoider Robot

Components Required:

1. Arduino UNO Board
2. Arduino USB Cable
3. Mini-Breadboard
4. 2 IR Sensors
5. L293D Motor Driver Module
6. Robotics Kit Part: Robo Chasis, 2 DC Motor, Clamps, Screw Sets, Screw Driver, Nut Driver, 2 Wheels.
7. Jumper Wires M-M & M-F.

Steps to Follow:

1. Please assemble the Robot as instructed by Instructor.
2. Please follow the same Steps from Step 1 to Step 18 of Line Follower Robot.
3. Create Sketch/Program for Obstacle Avoider Project.
4. Connect the USB cable with Arduino & Computer.
5. Click on Upload Option to download the program in Arduino.
6. Disconnect USB cable from Arduino & Connect 9V DC Battery to Arduino by using battery connector and then check your robot.

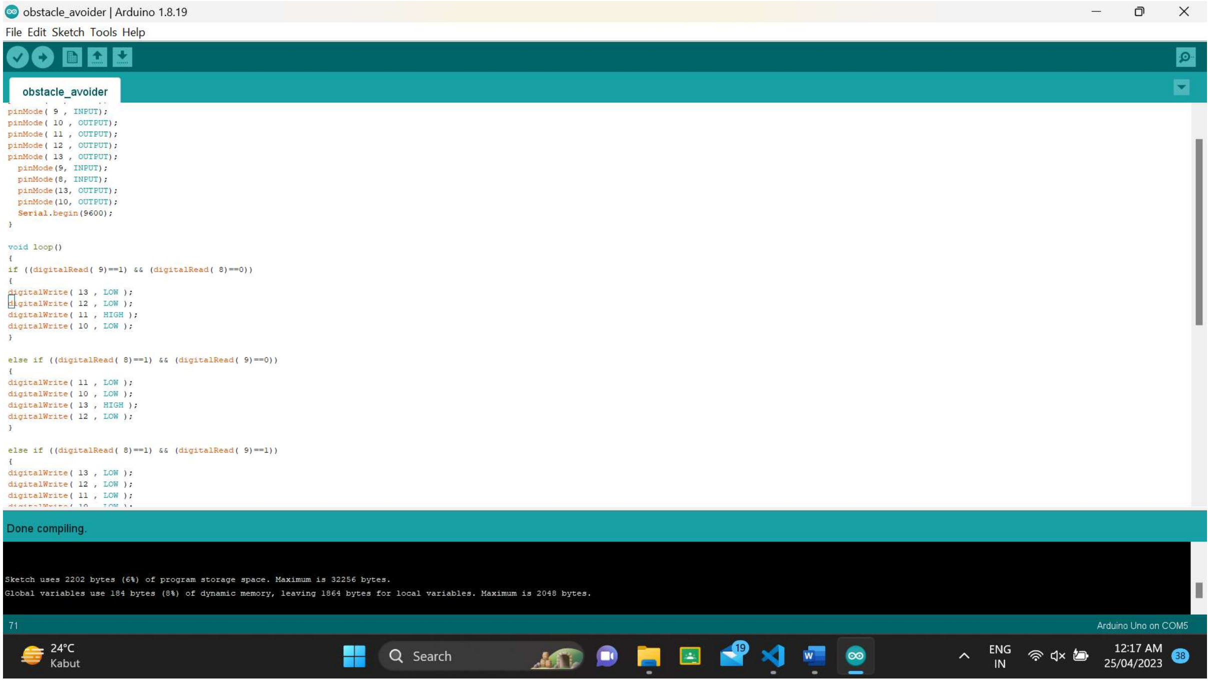
Program

```
void setup()
{
pinMode( 11 , OUTPUT);
pinMode( 8 , INPUT);
pinMode( 9 , INPUT);
pinMode( 10 , OUTPUT);
pinMode( 12 , OUTPUT);
pinMode( 13 , OUTPUT);
}

void loop()
{
if ((digitalRead( 9)==1) && (digitalRead( 8)==0))
{
digitalWrite( 13 , HIGH );
digitalWrite( 12 , LOW );
digitalWrite( 11 , LOW );
digitalWrite( 10 , LOW );
}

else if ((digitalRead( 8)==1) && (digitalRead( 9)==0))
{
digitalWrite( 11 , HIGH );
digitalWrite( 10 , LOW );
digitalWrite( 13 , LOW );
digitalWrite( 12 , LOW );
}

else if ((digitalRead( 8)==1) && (digitalRead( 9)==1))
{
digitalWrite( 13 , LOW );
digitalWrite( 12 , LOW );
digitalWrite( 11 , LOW );
digitalWrite( 10 , LOW );
}
else if ((digitalRead( 8)==0) && (digitalRead( 9)==0))
{
digitalWrite( 13 , HIGH );
digitalWrite( 12 , LOW );
digitalWrite( 11 , HIGH );
digitalWrite( 10 , LOW );
}
}
```



```
pinMode( 9 , INPUT);
pinMode( 10 , OUTPUT);
pinMode( 11 , OUTPUT);
pinMode( 12 , OUTPUT);
pinMode( 13 , OUTPUT);
  pinMode(9, INPUT);
  pinMode(8, INPUT);
  pinMode(13, OUTPUT);
  pinMode(10, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
if ((digitalRead( 9)==1) && (digitalRead( 8)==0))
{
digitalWrite( 13 , LOW );
digitalWrite( 12 , LOW );
digitalWrite( 11 , HIGH );
digitalWrite( 10 , LOW );
}

else if ((digitalRead( 8)==1) && (digitalRead( 9)==0))
{
digitalWrite( 11 , LOW );
digitalWrite( 10 , LOW );
digitalWrite( 13 , HIGH );
digitalWrite( 12 , LOW );
}

else if ((digitalRead( 8)==1) && (digitalRead( 9)==1))
{
digitalWrite( 13 , LOW );
digitalWrite( 12 , LOW );
digitalWrite( 11 , LOW );
```

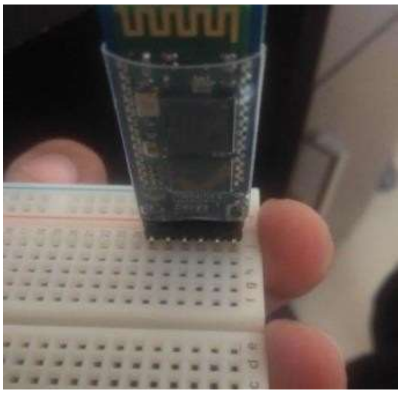
```
Sketch uses 2202 bytes (6%) of program storage space. Maximum is 32256 bytes.
Global variables use 184 bytes (8%) of dynamic memory, leaving 1864 bytes for local variables. Maximum is 2048 bytes.
```

3. Voice Controlled Robot

- **Components Required:**

- Arduino UNO Board
- Arduino USB Cable
- Breadboard
- HC-05 Bluetooth Module
- L293D Motor Driver Module
- Robotics Kit Part: Robo Chasis, 2 DC Motor, Clamps, Screw Sets, Screw Driver, Nut Driver, 2 Wheels.
- Jumper Wires M-M & M-F.

- **Steps to Follow:**



1. Please assemble the Robot as instructed by Instructor.
2. Please follow the same Steps from Step 8 to Step 18 of Line Follower Robot.
3. Mount the HC-05 Bluetooth Module between J25 to J30 wherein STATE Pin on J25 & EN Pin on J30.
4. Connect a Jumper Wire between breadboard –ve Terminal of Bread Board (GND) to G28.
5. Connect a Jumper between G27 to Rx (Pin 0) on Arduino.
6. Connect a Jumper between G26 to Tx (Pin 1) on Arduino.
7. Connect a Jumper Wire between G29 to +ve (5V) Common Terminal on Breadboard.
8. Open the Sketch/Program for Voice-Controlled-Robot Program.
9. Connect the USB cable with Arduino & Computer.
10. Click on Upload Option to download the program in Arduino. Note: While downloading the program please disconnect the Rx & Tx Jumper wire from Arduino Board. Once you finish with the download then re connect the Rx & Tx Jumper Wire.
11. Disconnect USB cable from Arduino & Connect 9V DC Battery to Arduino by using battery connector.
12. Run the AMR_Voice App in your Android Smartphone.
13. App will ask you to enable the Bluetooth. Allow It.
14. Search for your Bluetooth Device HC-05-(Group No). Once Connected Red Led on Bluetooth module will blink once per second instead of fast blinking.
15. Then control the devices from your Voice Commands on AMR_Voice App.
16. Voice Commands to be used: forward, backward, left, right, stop.

4. Gesture Controlled Robot

➢ Components Required:

- Arduino UNO Board
- Arduino USB Cable
- Breadboard
- HC-05 Bluetooth Module
- L293D Motor Driver Module
- Robotics Kit Part: Robo Chasis, 2 DC Motor, Clamps, Screw Sets, Screw Driver, Nut Driver, 2 Wheels.
- Jumper Wires M-M & M-F.

➢ Program

```
String voice;
int
M11 = 13, //Connect Motor 1 Blue wire To Pin #13
M12 = 12, //Connect Motor 1 Violet wire To Pin #12
M21 = 11, //Connect Motor 2 Blue wire To Pin #11
M22 = 10; //Connect Motor 2 Violet wire To Pin #10

//-------------------------Call A Function------------------------------//
void forward() {
    digitalWrite(M11, HIGH);
    digitalWrite(M12, LOW);
    digitalWrite(M21, HIGH);
    digitalWrite(M22, LOW);
    }
void backward(){
    digitalWrite(M11, LOW);
    digitalWrite(M12, HIGH);
    digitalWrite(M21, LOW);
    digitalWrite(M22, HIGH);
}
void left(){
    digitalWrite(M11, LOW);
    digitalWrite(M12, LOW);
    digitalWrite(M21, HIGH);
    digitalWrite(M22, LOW);
    }
void right(){
    digitalWrite(M11, HIGH);
    digitalWrite(M12, LOW);
    digitalWrite(M21, LOW);
    digitalWrite(M22, LOW);
    }
void stay(){
    digitalWrite(M11, LOW);
    digitalWrite(M12, LOW);
    digitalWrite(M21, LOW);
    digitalWrite(M22, LOW);
    }

//-----------------------------------------------------------------------------//
 void setup() {
  Serial.begin(9600);
  pinMode(M11, OUTPUT);
  pinMode(M12, OUTPUT);
  pinMode(M21, OUTPUT);
  pinMode(M22, OUTPUT);

}
//-----------------------------------------------------------------------------//
 void loop() {
  while (Serial.available())
{
  delay(10); //Delay added to make thing stable
  char c = Serial.read(); //Conduct a serial read
  if (c == '#') {break;} //Exit the loop when the # is detected after the word
  voice += c; //Shorthand for voice = voice + c
  }
  if (voice.length() > 0)
 {
    Serial.println(voice);
      if(voice == "*forward") {forward();}  //Move Robot Forward (Call Function)
  else if(voice == "*backward"){backward();} //Move Robot Backward (Call Function)
  else if(voice == "*left"){left();} //Move Robot Left (Call Function)
  else if(voice == "*right"){right();} //Move Robot right (Call Function)
  else if(voice == "*stop"){stay();} //Stop Robot (Call Function)
voice="";
}
} //Reset the variable after initiating
```

➢ Steps to Follow:

- Connection & Program will be same as Voice Controlled Robot except.
- Run the AMR_Gesture App in your Android Smartphone for controlling your robot by using Gesture. Follow video tutorial for creating gesture & using this app.

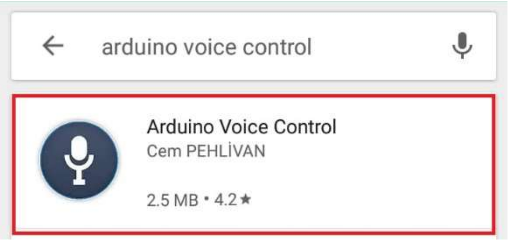
➢ Apps to be used in Android Mobile Phones for Ardubotics Workshop

We are going to use two apps in Andoid phone. Both apps are completely free. Please download them from Google Playstore in your mobile phone.

1. Arduino Voice Control (Arduino Voice Control)
2. BT Gesture Control for Arduino (AMR Gestures)

How to Install:

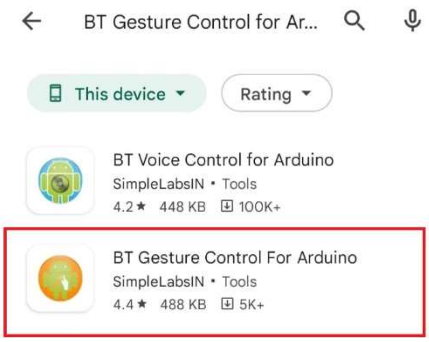
1. Arduino Voice Control: Simply search for Arduino Voice Control in Google Playstore in your Android Smartphone. First option that says Arduino Voice Control, simply touch on it to open it. Once it open, Click on Install. Wait for it to download & install it in your android smartphone. This app will be used for Voice Controlled Robot.



arduino voice control

Arduino Voice Control
Cem PEHLİVAN
2.5 MB • 4.2★

2. BT Gesture Control for Arduino: Simply search for BT Gesture Control for Arduino in Google Playstore in your Android Smartphone. Option that says BT Gesture Control for Arduino, simply touch on it to open it. Once it open, Click on Install. Wait for it to download & install it in your android smartphone. This AMR:Gestures app will be used for Gesture Controlled Robot.



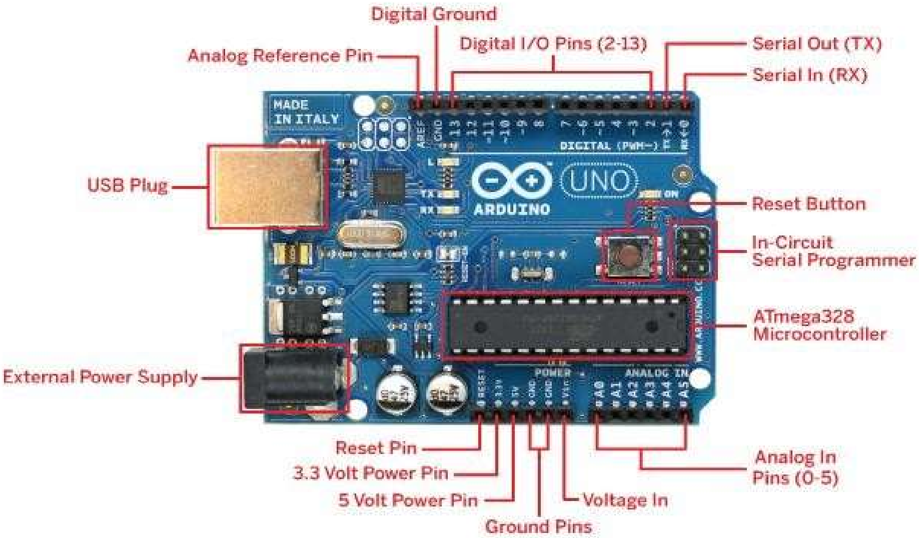
BT Gesture Control for Ar...

BT Voice Control for Arduino
SimpleLabsIN • Tools
4.2★ 448 KB 100K+

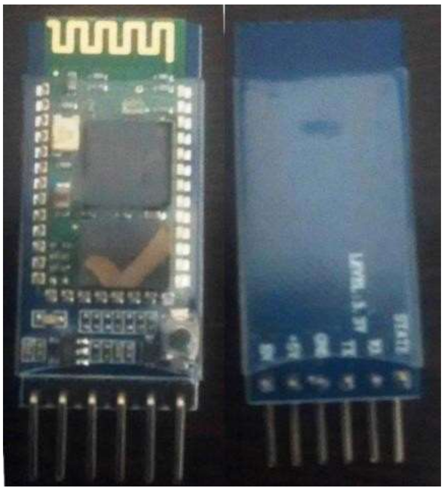
BT Gesture Control For Arduino
SimpleLabsIN • Tools
4.4★ 488 KB 5K+

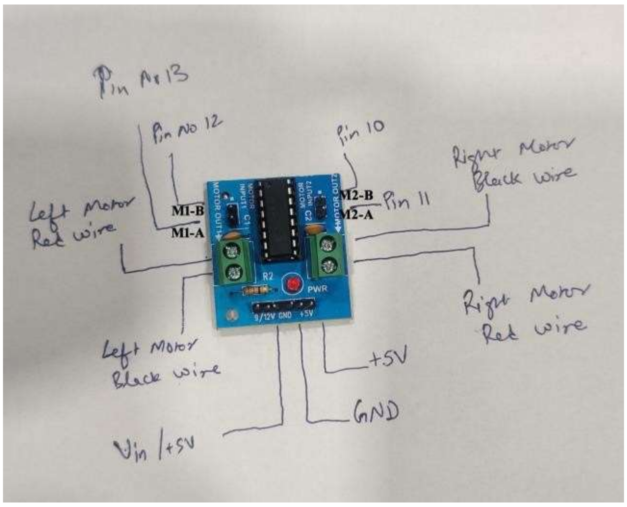
Anshit Kumar Srivastava
ECE

Major Electronics Components Images



Arduino Uno Board



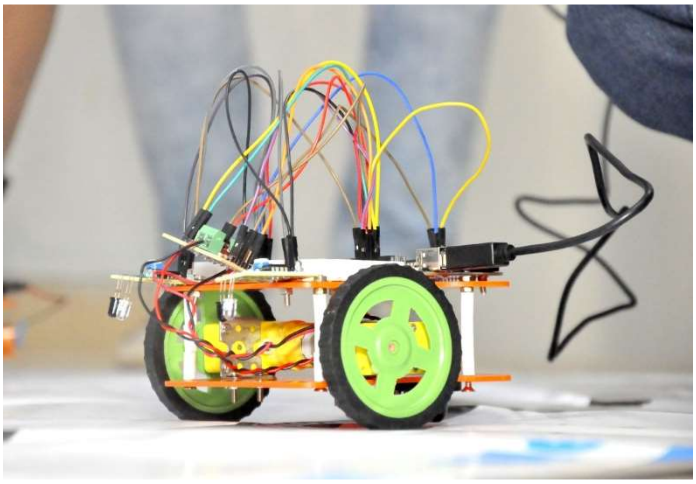


HC-05 Bluetooth Module

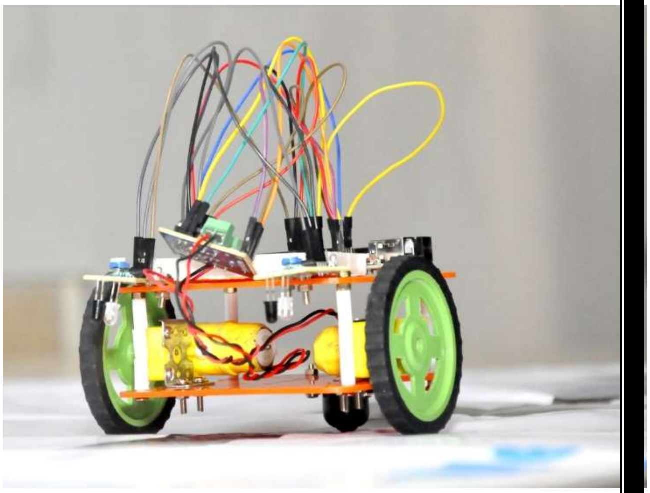
L293D Motor Driver Module

Anshit Kumar Srivastava
ECE

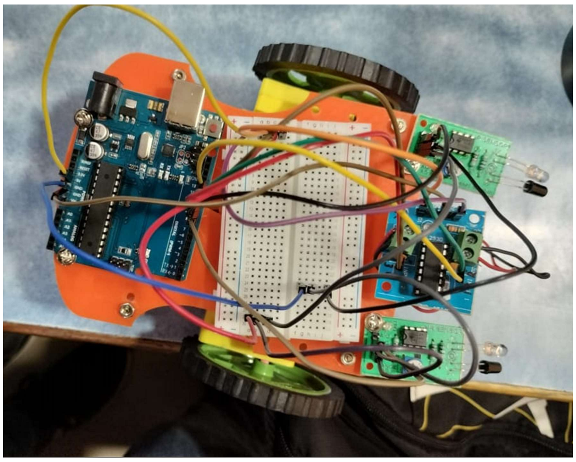
➢ Result



Side View



Front View



Top View

Anshit Kumar Srivastava
ECE