

# Robotics Pro Course

BY VAIBHAV SANU

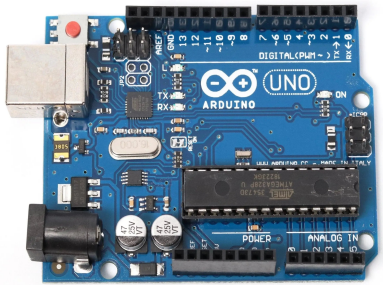
# What I've learned

- 🚗 C++ programming on the Arduino Uno
- 🚗 Used many sensors (photoresistor, IR sensors, proximity sensors)
- 🚗 Current: Flow of electrons
- 🚗 Voltage: Potential difference between 2 points
- 🚗 Resistance: Blocks current
- 🚗 Anode: Positive charge
- 🚗 Cathode: Negative charge

# Challenges I've faced

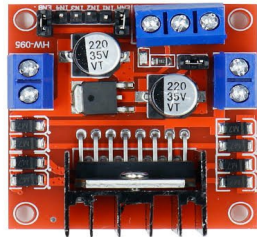
- 🚗 Programming multiple sensors simultaneously
- 🚗 Sensor sensitivity adjustment
- 🚗 Improper wiring (motors work differently)
- 🚗 HIGH & LOW signal confusion (how should robot operate)

# Parts of the robot



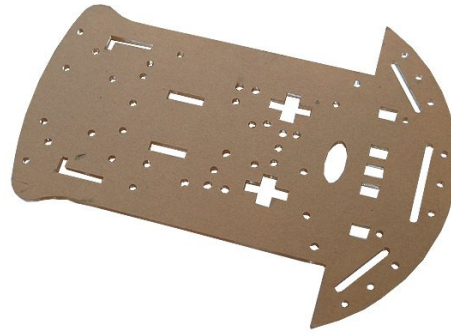
Arduino Uno R3

Runs C++ program from computer



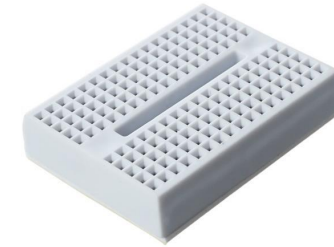
L298 Motor Driver

Translates C++ code from Arduino to DC motor movement.



Robot Chassis

Serves as body of robot



Mini Breadboard

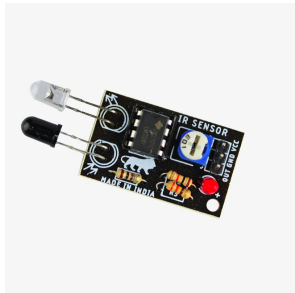
Provides connections from Arduino to sensors



2 DC Motors with Wheels

Initiates physical movement for the robot (forward, backward, left, right)

# Parts of the robot (continued)



IR Sensor

Detects infrared light, enabling the robot to follow black line



Proximity Sensor

Used for object detection; allows robot to follow/avoid object



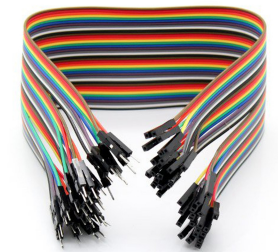
Photoresistor Sensor

Responds to light intensity; allows robot to follow/avoid light



4 AA Batteries

Power source for the Arduino controlled robot



Jumper Wires

Connects all parts of robot together to function properly

# C++ Program (Black line tracker)

Blackline

```
1 void setup() {
2   // put your setup code here, to run once:
3   pinMode(2, INPUT);
4   pinMode(3, INPUT);
5   pinMode(4, INPUT);
6   pinMode(8, OUTPUT);
7   pinMode(9, OUTPUT);
8   pinMode(10, OUTPUT);
9   pinMode(11, OUTPUT);
10
11 }
12
13 void loop() {
14   // put your main code here, to run repeatedly:
15   int rs = digitalRead (2); Reads voltage from right IR sensor
16   int ls = digitalRead (3); Reads voltage from left IR sensor
17   int cs = digitalRead (4); Reads voltage from central proximity sensor
18
```

Mapped to sensors from Arduino pin #

Mapped to DC motors from Arduino pin #  
(controlled by motor driver)

Voltage is either read  
as HIGH or LOW signal

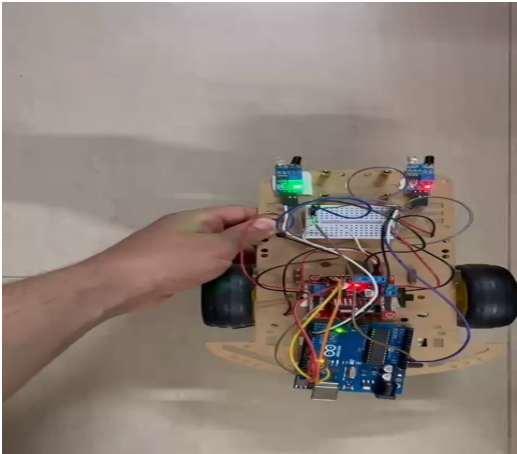
# C++ Program (Black Line Tracker cont'd)

```
19 if (rs == LOW && ls == LOW && cs == HIGH)
20 {digitalWrite(8,LOW);      Robot moves forward
21   digitalWrite(9,HIGH);
22   digitalWrite(10,LOW);
23   digitalWrite(11,HIGH);
24 }
25 if (rs == HIGH && ls == LOW && cs == HIGH)
26 {digitalWrite(8,LOW);      Robot turns left
27   digitalWrite(9,HIGH);
28   digitalWrite(10,LOW);
29   digitalWrite(11,LOW);
30 }
```

```
31 if (rs == LOW && ls == HIGH && cs == HIGH)
32 {digitalWrite(8,LOW);
33   digitalWrite(9,LOW);      Robot turns right
34   digitalWrite(10,LOW);
35   digitalWrite(11,HIGH);
36 }
55 if (rs == HIGH && ls == HIGH && cs == LOW)
56 {digitalWrite(8,LOW);
57   digitalWrite(9,LOW);      Robot stops
58   digitalWrite(10,LOW);
59   digitalWrite(11,LOW);} }
```

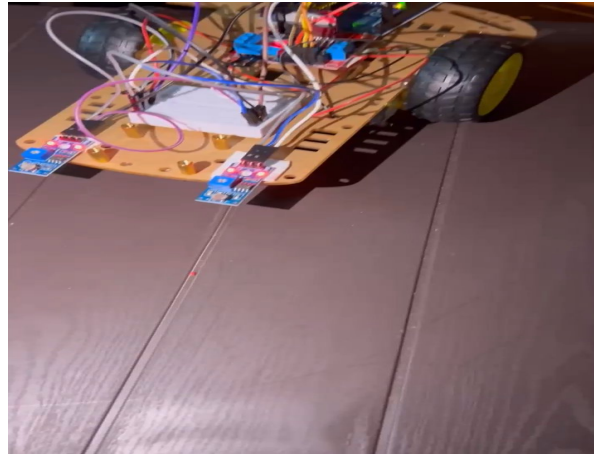
CS stands for “center sensor”. The center sensor is the middle proximity sensor. Whenever CS signal is HIGH, the robot keeps moving. Once CS signal is LOW, robot stops. This means whenever the robot has detected an object ahead, it stops. Watch the 3<sup>rd</sup> video next slide to see the program in action!

# Videos



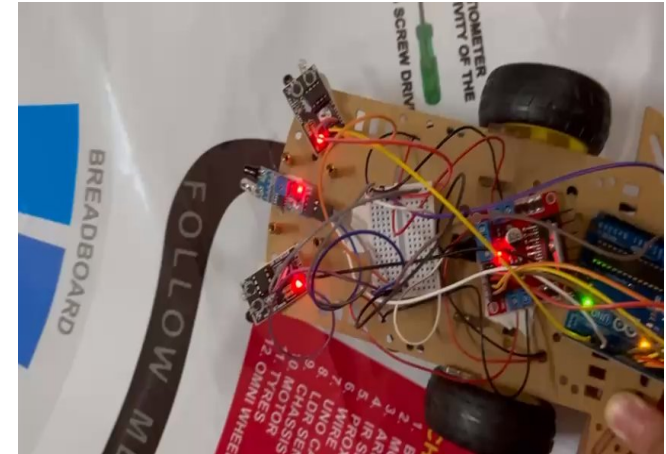
## Object Avoider

Uses 2 proximity sensors. When sensor is signaled, robot moves away from object.



## Light Avoider

Uses 2 photoresistors. When light falls on photoresistor, robot moves away from the light.



## Black Line Tracker

Uses 2 IR sensors, 1 proximity sensor. IR sensors allow robot to follow line. Proximity sensor allows robot to stop when object detected.