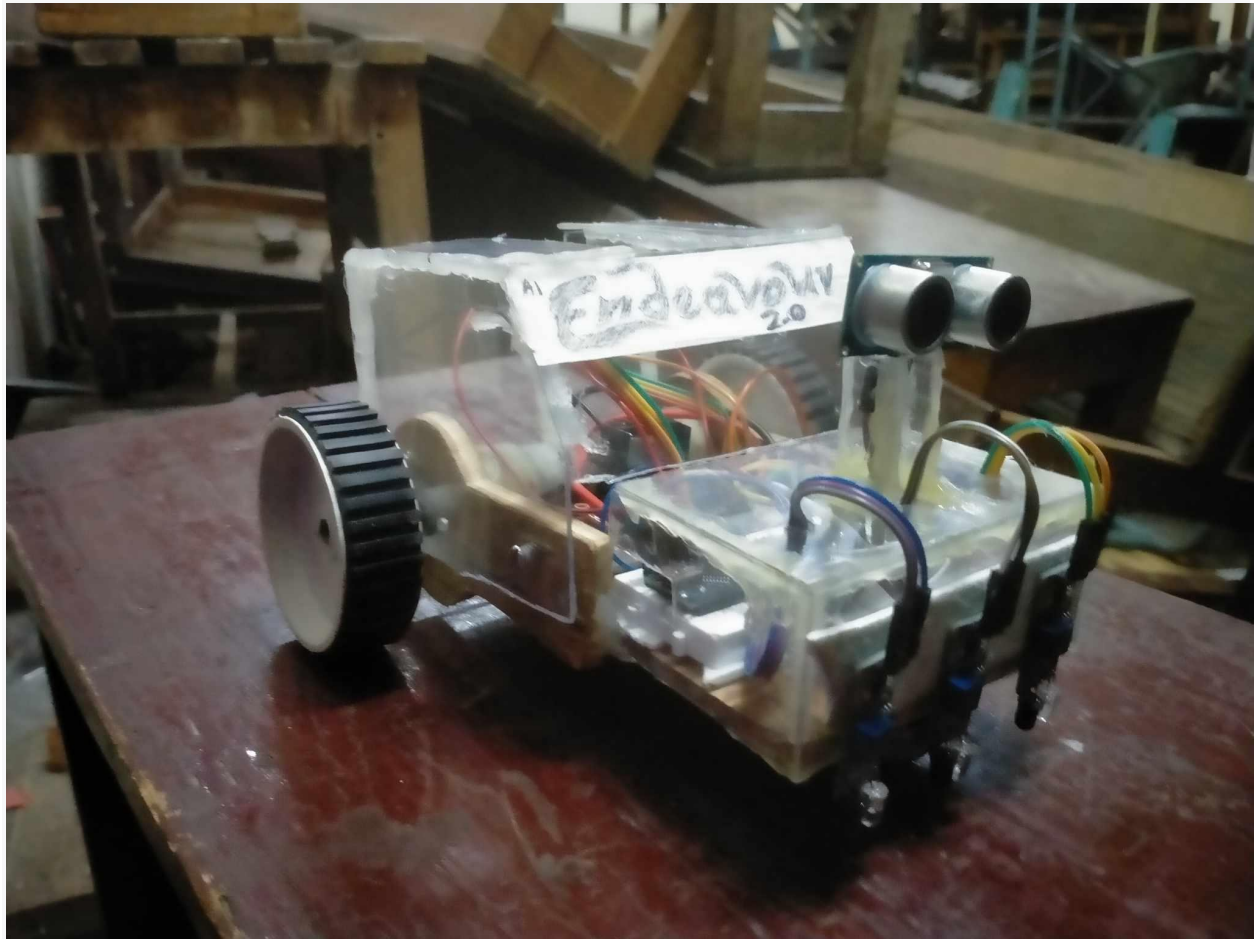# A PROJECT MADE BY A1

A UNIQUE DESIGN OF LINE-TRACKING ROBOT WITH VERSATILE FUNCTIONALITY

In this publication, we present a novel and ingenious design of a line-tracking robot that operates by faithfully tracing a black line on a white surface. This robot boasts two distinct operational modes: line-tracking mode and obstacle detection mode. With a sleek and car-like appearance, it gracefully maneuvers its way around obstacles encountered on its path. Its remarkable capabilities allow it to effortlessly follow a predefined route marked by a black line, while simultaneously detecting obstacles on both sides. To bring this design to life, we leverage the power of an Arduino Nano, calling for expertise in Arduino programming, seamless integration of electrical circuits with the designated code, and fundamental mechanical engineering principles to optimize obstacle detection in dual directions. The robot is thoughtfully powered by a rechargeable lithium-ion battery, ensuring sustained performance. Equipped with three infrared sensors intricately connected to a motor driver IC, alongside two sonars, this versatile robot embarks on a multitude of functions and tasks, displaying its prowess and adaptability.

INTRODUCTION: A robot represents a remarkable automaton, meticulously crafted to alleviate the burdens of human labor. Its primary purpose is to liberate individuals from monotonous tasks, allowing them to dedicate their time and energy to more intricate and significant endeavors. In an era where autonomous systems are increasingly prevalent, robots emerge as vital allies in reducing human effort. In the past, these mechanical marvels were confined by the limitations of their time, but today's robots have transcended those boundaries, acquiring the ability to perform a vast array of tasks. In line with this remarkable progress, our latest creation surpasses the realm of conventional line tracking. It possesses an extraordinary aptitude for navigating alternative routes in the presence of obstacles while faithfully following a captivating ebony path. To enhance its capabilities, we have incorporated a sophisticated switch that bestows upon it enhanced intelligence, efficiency, and user-friendliness. When our creation momentarily strays from its ebony trajectory, it gracefully pivots, ready to resume its quest upon encountering a pristine black line against a sea of ivory. With unmatched versatility, our line-tracking marvel boasts multiple operational modes, catering to diverse needs. Its uncanny ability to detect obstacles in both directions adds an extra layer of ingenuity, making it an ideal companion for tasks such as efficient item transport within office settings. Moreover, its prowess proves invaluable in safeguarding human lives during critical military operations, ensuring enhanced safety and efficiency. Equipped with a meticulously positioned camera, it assumes the role of a vigilant sentry, capturing crucial surveillance footage in strategic locations. Furthermore, when it embarks on a whimsical journey, faithfully following the ebony path without deviation, it delights and entertains. With its extensive applications and remarkable features, our line-tracking robot serves as an exemplary springboard for engineers, igniting their creativity and inspiring further innovation.

1. About Event

The project was developed as part of an event organized by the Robotics Club at Pashchimanchal Campus. The event aimed to teach participants about the components and tools used in bot making. The specific bot created during the event was a line-following bot, designed to follow a white line on a blackboard.

2. What We Learned

Throughout the event, we gained valuable knowledge and skills related to bot making. We learned about the different components involved in building a bot, such as motors, sensors, and microcontrollers. Additionally, members gained an understanding of the tools and techniques used in the assembly and programming of the bot. Through hands-on experience, our team also learned about problem-solving, logical reasoning, and teamwork.

3. Bot Making Process

The process of creating the maze-solving bot involved several steps:

1. Component Selection:

We learned about the different components required for the bot, such as motors, wheels, sensors, and microcontrollers. We were taught how to choose suitable components based on the project requirements.

2. Assembly:

We were guided through the process of assembling the bot using the selected components and learned how to connect the motors, attach the wheels, and mount the sensors onto the chassis.

3. Circuit:

We learned about the wiring and circuit required for the bot as well as were taught how to connect the motors and sensors to the microcontroller and ensure proper functionality.

4. Programming:

We learned how to program the bot to solve the maze. Then we were introduced to programming languages such as Arduino and were taught how to write code that would enable the bot to follow the white line on the blackboard.

4. Code

```cpp
#define Left_IR 10

#define Central_IR 11

#define Right_IR 12


#define Left_PWM 6

#define Left_MOTOR_1 4

#define Left_MOTOR_2 5


#define Right_PWM 9

#define Right_MOTOR_3 8

#define Right_MOTOR_4 7


#define Echo 2

#define Trigg 3


float obstacleDistance;

float ultrasonic();


void setup() {
```

```
Serial.begin(9600);

Serial.available();


pinMode(Left_IR, INPUT);

pinMode(Central_IR, INPUT);

pinMode(Right_IR, INPUT);


pinMode(Left_PWM, OUTPUT);

pinMode(Left_MOTOR_1, OUTPUT);

pinMode(Left_MOTOR_2, OUTPUT);


pinMode(Right_PWM, OUTPUT);

pinMode(Right_MOTOR_3, OUTPUT);

pinMode(Right_MOTOR_4, OUTPUT);


pinMode(Echo,INPUT);

pinMode(Trigg,OUTPUT);
}
```

```cpp
void loop() {

  int L,C,R;

    L = digitalRead(Left_IR);

    C = digitalRead(Central_IR);

    R = digitalRead(Right_IR);

// Obstacle detection and handling

obstacleDistance = ultrasonic();

if (obstacleDistance <15.0) {

    Serial.println("Obstacle Detected");

    Serial.println(obstacleDistance);

    analogWrite(Left_PWM,0);

    analogWrite(Right_PWM,0);

    digitalWrite(Left_MOTOR_1, LOW);

    digitalWrite(Left_MOTOR_2, LOW);

    digitalWrite(Right_MOTOR_3, LOW);
```

```cpp
    digitalWrite(Right_MOTOR_4, LOW);


}

else{

  if (L == 1 &&  C == 0 && R == 1){

  Serial.println("forward");

  digitalWrite(Left_MOTOR_1, HIGH);

  digitalWrite(Left_MOTOR_2, LOW);

  digitalWrite(Right_MOTOR_3,


 HIGH);

  digitalWrite(Right_MOTOR_4, LOW);

  analogWrite(Left_PWM,200);

  analogWrite(Right_PWM,200);
}


else if (L == 0 &&  C == 1 && R == 1){

  Serial.println("left");

  digitalWrite(Left_MOTOR_1, HIGH);
```

```
    digitalWrite(Left_MOTOR_2, LOW);

    digitalWrite(Right_MOTOR_3, HIGH);

    digitalWrite(Right_MOTOR_4, LOW);

    analogWrite(Left_PWM,60);

    analogWrite(Right_PWM,200);

    }


else if (L == 0 &&  C == 0 && R == 1){

    Serial.println("Sharp left");

    digitalWrite(Left_MOTOR_1, LOW);

    digitalWrite(Left_MOTOR_2, HIGH);

    digitalWrite(Right_MOTOR_3, HIGH);

    digitalWrite(Right_MOTOR_4, LOW);

    analogWrite(Left_PWM,160);

    analogWrite(Right_PWM,160);

    }


else if (L == 1 &&  C == 1 && R == 0){
```

```cpp
  digitalWrite(Left_MOTOR_1, HIGH);

  digitalWrite(Left_MOTOR_2, LOW);

  digitalWrite(Right_MOTOR_3, HIGH);

  digitalWrite(Right_MOTOR_4, LOW);

   analogWrite(Left_PWM,200);

   analogWrite(Right_PWM,60);

  Serial.println("right");}


  else if (L == 1 &&  C == 0 && R == 0){

  digitalWrite(Left_MOTOR_1, HIGH);

  digitalWrite(Left_MOTOR_2, LOW);

  digitalWrite(Right_MOTOR_3, LOW);

  digitalWrite(Right_MOTOR_4, HIGH);

  analogWrite(Left_PWM,160);

  analogWrite(Right_PWM,160);

  Serial.println("Sharp right");}

 else if(L==0 &&  C==0 && R==0)

  {

  Serial.println("Stop");
```

```
    analogWrite(Left_PWM,0);

    analogWrite(Right_PWM,0);

    digitalWrite(Left_MOTOR_1, LOW);

    digitalWrite(Left_MOTOR_2, LOW);

    digitalWrite(Right_MOTOR_3, LOW);

    digitalWrite(Right_MOTOR_4, LOW);

  }

}

}

 float ultrasonic(){

 float duration;

 float distance;

 digitalWrite(Trigg, LOW);

 delayMicroseconds(2);

 digitalWrite(Trigg, HIGH);

 delayMicroseconds(10);

 digitalWrite(Trigg, LOW);

 duration= pulseIn(Echo, HIGH);

 distance= 0.017*duration;
```

```
 return distance;

}
```

5. Problems and Solutions

During the project, participants encountered various challenges. Here are some of the common problems faced and their corresponding solutions:

1.Obstacle Detection: One challenge was detecting obstacles in the bot's path. This was addressed by incorporating an ultrasonic sensor that could measure the distance to objects and trigger appropriate actions.

2.Line Following: Ensuring accurate line following was another obstacle. By using infrared sensors to detect the white line and implementing conditional statements in the code, we were able to guide the bot along the correct path.

3.Motor Control: Proper control of the bot's motors was crucial. We faced issues related to motor speed and direction control. These were resolved by carefully configuring the motor pins and using appropriate analogWrite() values to adjust the motor speed.

d.Sensor Calibration: As a team, we had to calibrate the infrared sensors to ensure accurate line detection. This involved adjusting the sensor's sensitivity and position to match the line's color and width.