Hochschule Hamm-Lippstadt

Line follower Vehicle

Prototyping and System Engineering

Khaled Abdallattif

6/25/2023

Prof. Stefan Henkler

Electronics Engineering Department

Introduction and Motivation:

This report documents the design, implementation, and evaluation of an autonomous line-following car, aiming to develop a robust and efficient system capable of navigating predefined paths, detect the obstacle with avoidance. The project involved integrating advanced sensors, control algorithms, and motor control mechanisms to enable the car to autonomously follow a line on the ground with precision and stability.

Motivation:

The motivation behind this project stems from the increasing demand for autonomous robotic systems in various industries. Autonomous vehicles have gained significant attention due to their potential to revolutionize transportation, logistics, and manufacturing. By developing a capable line-following car, we aim to contribute to the advancement of autonomous navigation technologies and explore their practical applications.

The primary motivation for this project is to showcase the feasibility and effectiveness of autonomous line-following cars in real-world scenarios. Such vehicles offer numerous benefits, including increased efficiency, reduced human intervention, and improved safety. They can be utilized in applications such as automated material handling, warehouse operations, and even autonomous transportation systems.

Additionally, this project aims to address the technical challenges associated with developing an autonomous line-following car. These challenges include sensor integration, accurate path detection, real-time control, and robustness in the face of varying environmental conditions. By tackling these obstacles and documenting our findings, we aim to contribute to the existing knowledge base and provide insights for future researchers and developers in the field of autonomous systems.

System Specification for Line following vehicle with object detection.

Overview

It is a Line follower vehicle which can follow the line on the ground using IR-Sensor with motor controller. It can detect the line and adjust its speed and direction accordingly. The system can also avoid obstacles and stop at any given destination by using Ultrasonic sensor besides ignore the obstacle to follow the line again.

Requirements

- The vehicle must be able to follow the lines on the ground using IR sensor, and a motor controller with its driver to allow the speed control.

- The vehicle must be able to avoid obstacles and stop by ultra sonic sensor at any given destination and ignore it by using the code.

- The vehicle must be able to operate on battery power for at least 30 minutes.

#Components

- The vehicle consists of the following components:

  - A chassis with two wheels and a differential drive mechanism.

  - IR sensors mounted on the front of the chassis that can catch the line and follow it successfully.

  - A motor controller with its driver that can control the speed and direction of the wheels based on the input from the IR-Sensors and.

  - A microcontroller or Arduino that can process IR-Sensors Ultrasonic to send commands to the motor controller.

  - A battery pack that can provide power to the microcontroller/ Arduino, the motor controller, and the camera.

- 2 wheels and On/Off Button.

Functions

- The vehicle performs the following functions:

  - Line following: The vehicle follows the line on the ground by adjusting its speed and direction based on the color and position of the line. The vehicle follows different rules for different colors of lines

- Obstacle avoidance: The vehicle avoids any obstacles in its path by changing its direction or stopping if necessary by using ultrasonic Sensor.

  - Start: Start following the line on the ground.

  - Stop: Stop following the line and stay in place.

  - Object detection: Ignore the object when detect it by turning right or left.

  - Turn left: Turn left at the next intersection or obstacle.

  - Turn right: Turn right at the next intersection or obstacle.

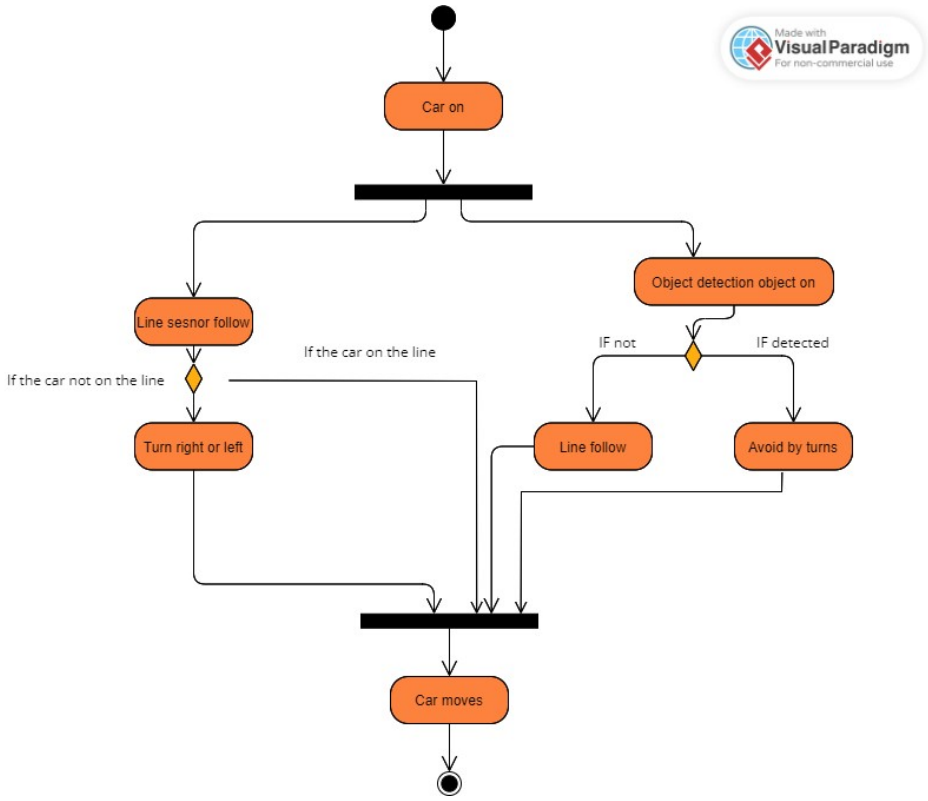
  - Line again: looking for the line and detect it once again.

  -Follow the Line again: after object detected and ignore it return back to the line.

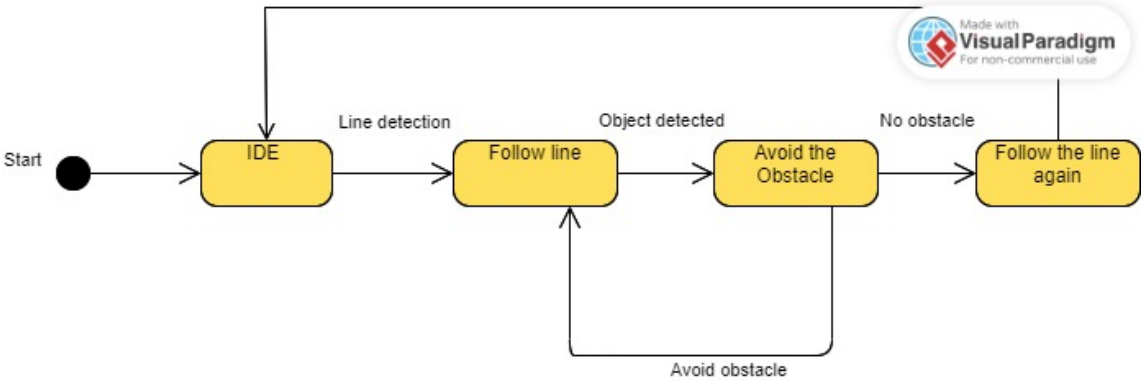
  - Battery level: Request the battery level of the vehicle.

-Sys ML.

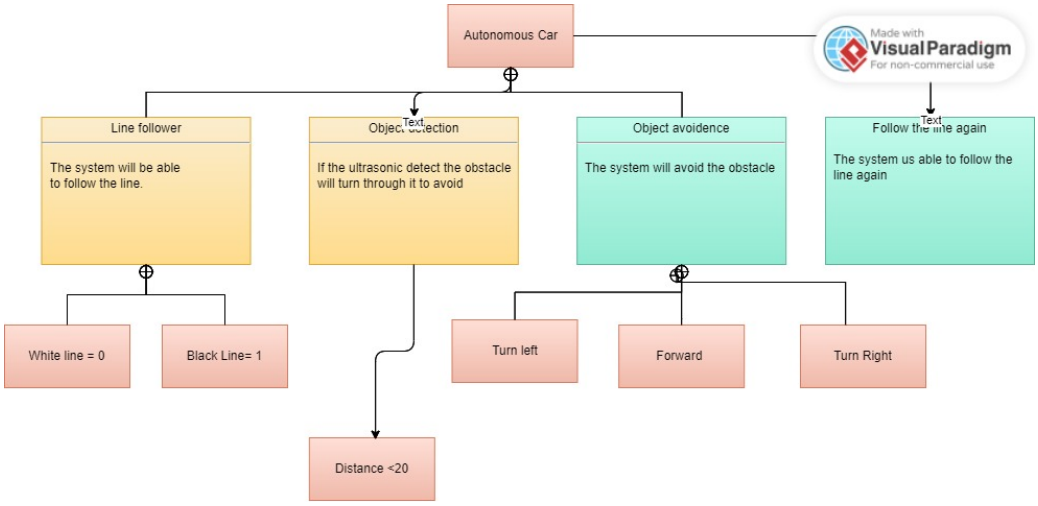
Activity Diagram.



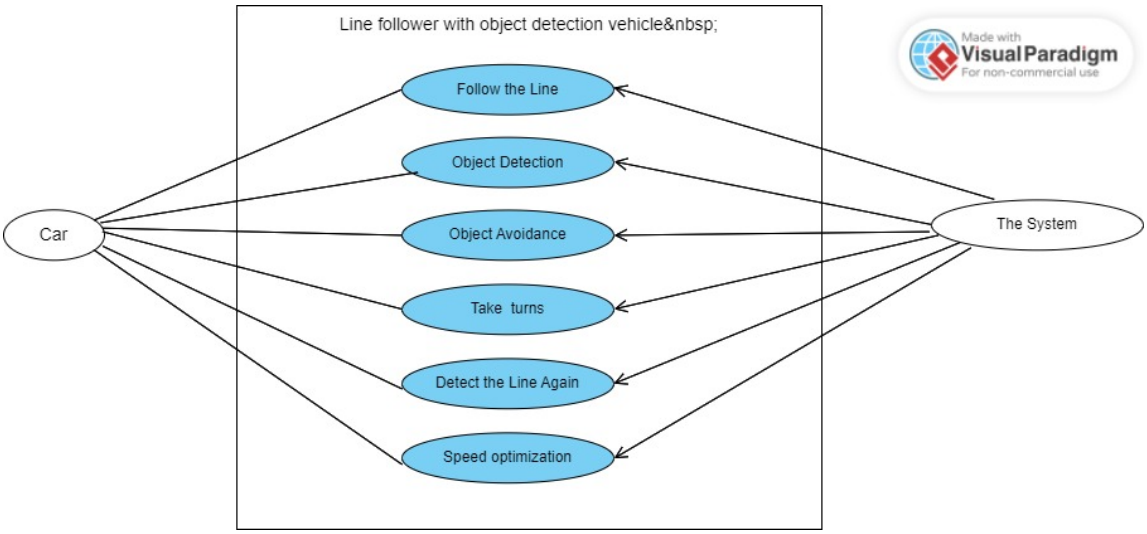
State Machine Diagram



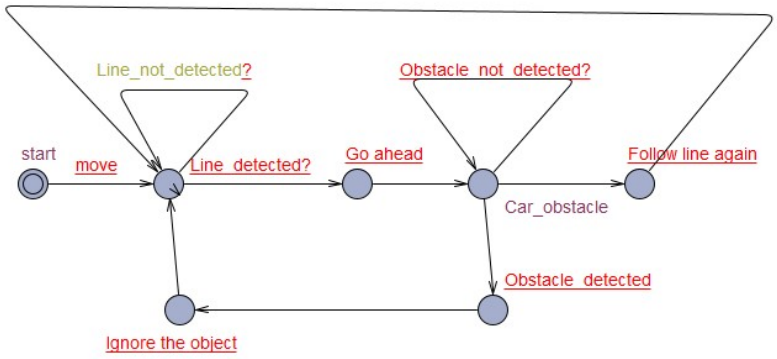
Requirement diagram



User Case Diagram



Uppaal Diagram



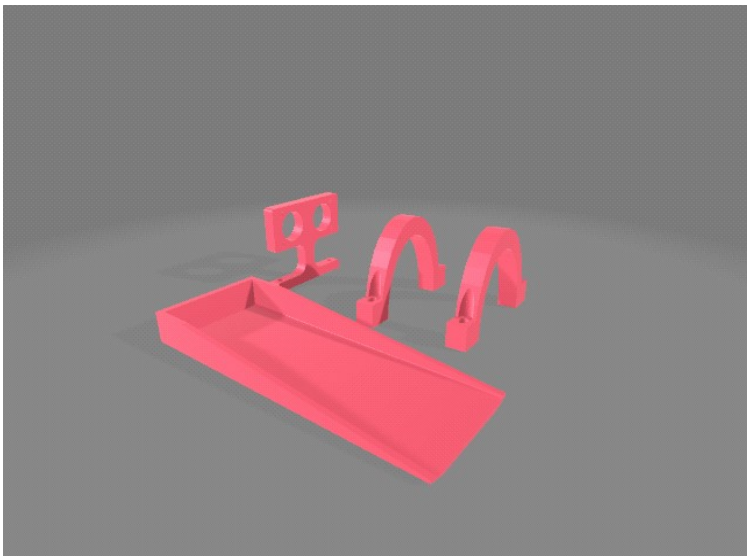
Design Using TinkerCad:

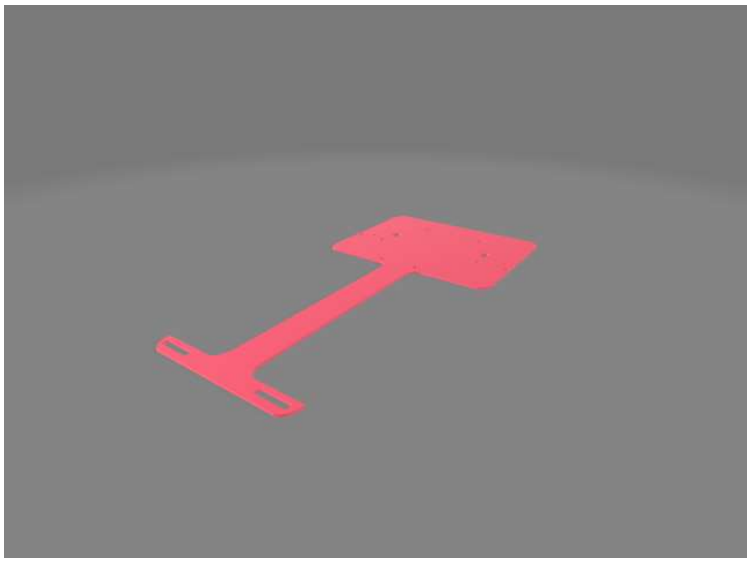
First, we will used laser cutting wood piece as the car chassis. These dimensions are ideal for making a Motor Car. Also I have used 3d printing to print out Car Parts such as motor, battery holder and clamps..

The car parts dimensions:

Arduino: L 80 w 58 Motors: L 52 w 38 Rind around the motor: 37 with width 2 Cm H-Bridge: L 44 W 42 IR-Sensor: L 42 w 11 depth 2 mm Ultrasonic-Sensor: L 46 w 22 depth 2 Wheel: L 63 w 27 * 38 support wheel the whole width with support 47 Battery: L 140 w 47 Depth 26. the length with the cables 255 distance between motors and arduino 20 mm and 20 mm The entire width is : 164, Entire Length is : 320 mm Scolder: L 16, W 22

-Preliminary Design



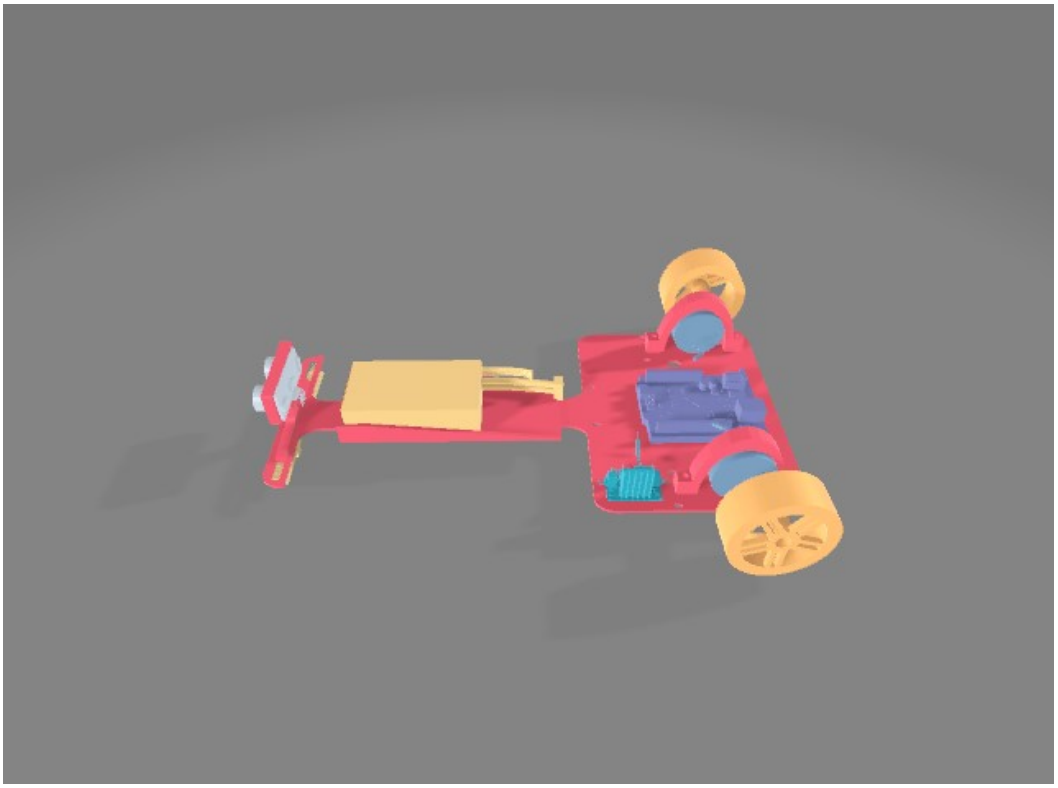


Final Design:

1-Connect the IR Sensors as I showed in the photo. Then connect the necessary wires with the Arduino UNO. Just Follow the Wiring Diagram in the schematics.

- Now Rotate the Potentiometers one by one slowly clockwise till the LED of the IR Sensor Turns on.
- Next, slowly move the car to the Left and Right in such a way that the car Left and the Right sensor will be on the black line. and then the car wheels rotate in a different direction.
- More specifically, If you put the Left Sensor on the black line the car will try to move in the right direction. Similarly, If the right sensor is in dark then The car moves in the Left direction.
- This is because to place the car all-time on the Blackline. One main corn of this configuration is that you can't go too fast in this configuration. As a beginner project, this is great.

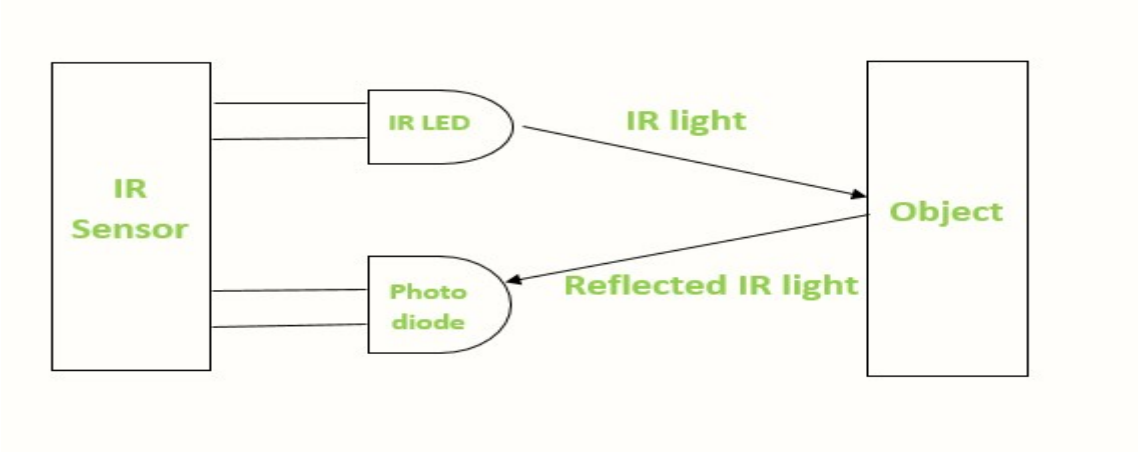
So, the calibration process is bit easier in this case. Now you can turn on the car and place on the where you want to follow and the car will follow the line.

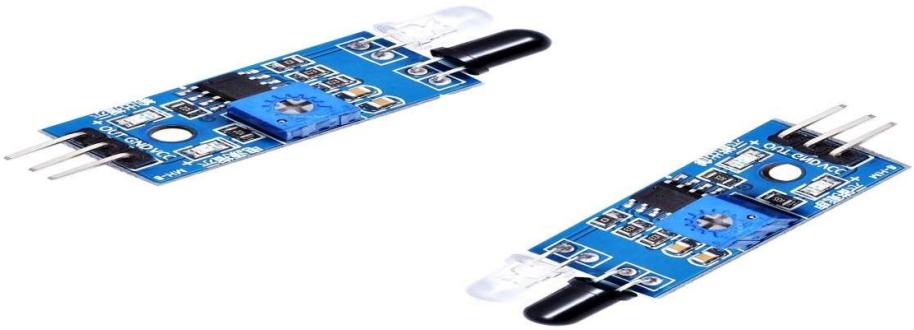


Hardware Component

- IR-Sensor

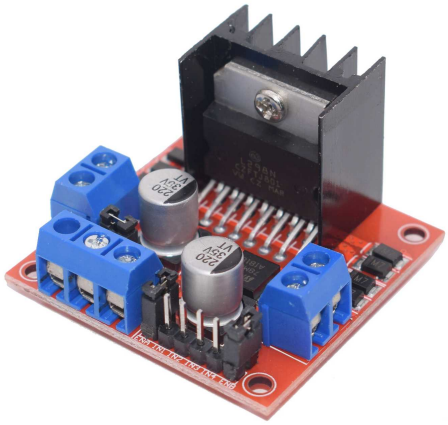
Active infrared sensors work with radar technology and they both emit and receive infrared radiation. This radiation hits the objects nearby and bounces back to the receiver of the device.





Motor Driver Controller Board

-It is an electrical or electronic device that regulate motor speed, torque, and position outputs. The drive modifies the power input to the motor to achieve the desired output. -The controller circuits are commonly integrated with the drive circuits as one stand-alone unit, thus the terms motor drive and motor controller are frequently used interchangeably

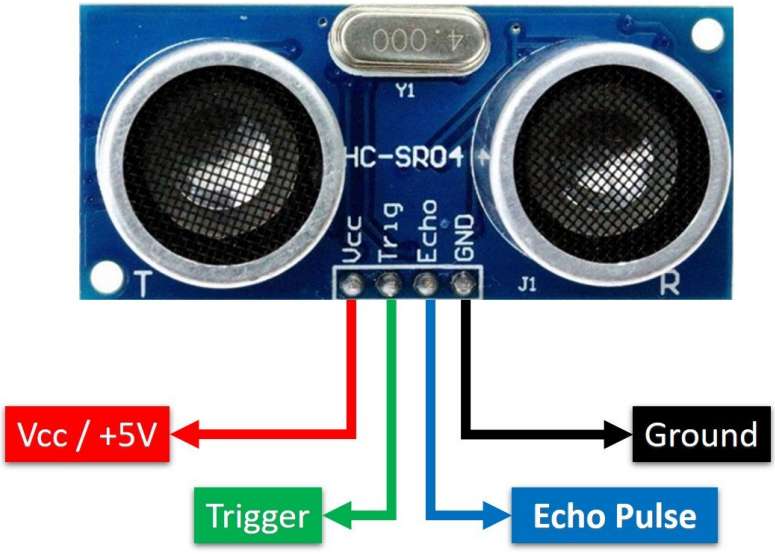


DC- Motor.

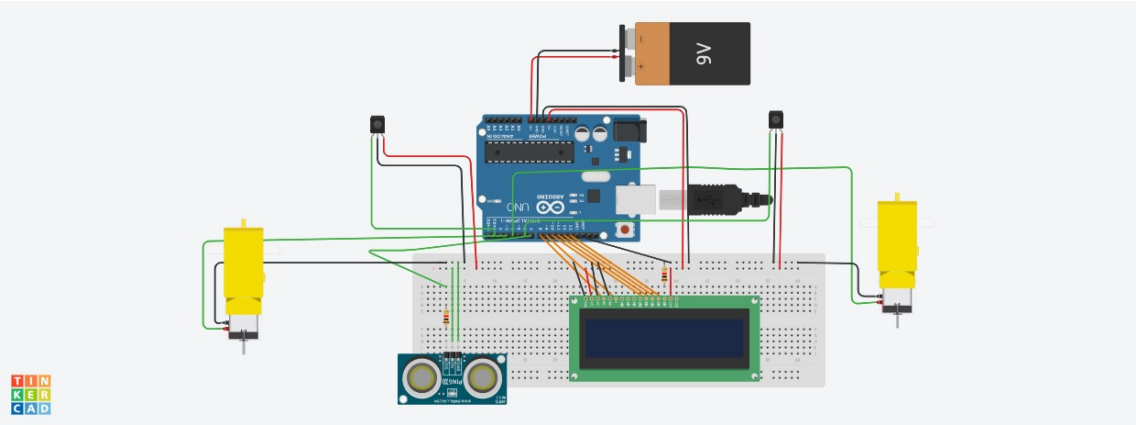
DC 12 V 100 RPM Gear Motor High Torque Electric Micro Speed, You also can adjust the speed, but the motor speed can only be reduced, can not be raised. Low noise and high torque, such as 10rpm can drive 15 kilograms.

Ultrasonic Sensor

An ultrasonic sensor emits a sound pulse in the ultrasonic range. This sound pulse propagates at the speed of sound through air (about 344 meters per second) until the sound pulse encounters an object. The sound pulse bounces off the object and is returned in reverse to the sensor where this "echo" is received.



Simulation & Wiring



Code

```
//motors
#define input1 13
#define input2 12
#define input3 7
#define input4 6
#define enA 11
#define enB 5
  int LEFT_SENSOR;
  int RIGHT_SENSOR;
#define echo 8    //Echo pin
#define trigger 9 //Trigger pin
//ir sensors
#define LirSensor 3
#define RirSensor 4

int maxSpeed = 95;   //100
int turnSpeed = 75;

int duration;
int distance;

void setup() {
  Serial.begin(9600);
  pinMode(input1, OUTPUT);
```

```
 pinMode(input2, OUTPUT);
 pinMode(input3, OUTPUT);
 pinMode(input4, OUTPUT);

 pinMode(enA, OUTPUT);
 pinMode(enB, OUTPUT);

 pinMode(LirSensor, INPUT);
 pinMode(RirSensor, INPUT);

pinMode(echo, INPUT);
pinMode(trigger, OUTPUT);
}
void loop() {
// Clears the trigPin
 digitalWrite(trigger, LOW);
 delayMicroseconds(2);
 // Sets the trigPin on HIGH state for 10 micro seconds
 digitalWrite(trigger, HIGH);
 delayMicroseconds(10);
 digitalWrite(trigger, LOW);
 // Reads the echoPin, returns the sound wave travel time in microseconds
 duration = pulseIn(echo, HIGH);
 // Calculating the distance
 distance = duration * 0.034 / 2;
 //for Line follow
 int LEFT_SENSOR = digitalRead(LirSensor);
 int RIGHT_SENSOR = digitalRead(RirSensor);

 Serial.print("leftsenor: ");
 Serial.print(LEFT_SENSOR);
 Serial.print("   ");
 Serial.print("RIghtsenor: ");
 Serial.print(RIGHT_SENSOR);
```

```
 if (RIGHT_SENSOR == 0 && LEFT_SENSOR == 0) { // 1 =black, 0= white
   forward(); //FORWARD
 }
 else if (RIGHT_SENSOR == 1 && LEFT_SENSOR == 0) {
   right(); //Move Right
 }
 else if (RIGHT_SENSOR == 0 && LEFT_SENSOR == 1) {
   left(); //Move Left
 }
 else if (RIGHT_SENSOR == 1 && LEFT_SENSOR == 1) {
 }
avoidobject();
}
void forward()
{
 digitalWrite(input1, HIGH);
 digitalWrite(input2, LOW);
 digitalWrite(input3, HIGH);
 digitalWrite(input4, LOW);

 analogWrite(enA,maxSpeed);
 analogWrite(enB, 130);
}
void right()
{
 digitalWrite(input1, LOW);
 digitalWrite(input2, HIGH);
 digitalWrite(input3, HIGH);
 digitalWrite(input4, LOW);

 analogWrite(enA, turnSpeed);
 analogWrite(enB, 105);
}
void left()
```

```
{
  digitalWrite(input1, HIGH);
  digitalWrite(input2, LOW);
  digitalWrite(input3, LOW);
  digitalWrite(input4, HIGH);

  analogWrite(enA, turnSpeed);
  analogWrite(enB, 105);
}
void Stop()
{
  digitalWrite(input1, LOW);
  digitalWrite(input2, LOW);
  digitalWrite(input3, LOW);
  digitalWrite(input4, LOW);
}
void avoidobject(){
if (distance<20) {
   left();
delay(700);
  forward();
  delay(1700);
  right();
  delay(1400);
   forward();
while(LEFT_SENSOR == 0){
LEFT_SENSOR = digitalRead(LirSensor);
}
Stop();
delay(750);
forward();
delay(500);
Stop();
delay(750);
```

```
left();
delay(900);
}
}
```

Git Repository & Tasks Responsibilities.

- https://github.com/Khaledgowied/Line-Follower-hochschule

Xcel –Sheet.

Short Summary	khaled abdallattif	
	To Do Deadline	Done
SRS creation & Overall content	11/04/2023	11/04/2023
UML Diagram	11/04/2023	11/04/2023
SysML Diagram	11/04/2023	11/04/2023
SysML Diagram	11/04/2023	11/04/2023
Task Tracking	11/04/2023	11/04/2023
Simulation Circuit Design	25/04/2023	23/04/2023
First phase of design 3D CAD	02/05/2023	01/05/2023
Second phase of 3D CAD & Printing	10/05/2023	09/05/2023
Drilling and mounting the component among design	14/05/2023	17/05/2023
Data sheet and connection and wiring	17/05/2023	17/05/2023
Upaal design	24/05/2023	24/05/2023
First Phase coding and testing linie follow	24/05/2023	24/05/2023
Second phase coding and testing object detection	31/05/2023	31/05/2023
Third phase coding and testing	04/06/2023	04/06/2023
Forth Phase for softwae and coding	14/06/2023	14/06/2023
Repeating UML & SRS & SYSML & System Specification	18/06/2023	18/06/2023

References

1- https://www.tinkercad.com/dashboard?type=tinkercad&collection=designs
2- https://www.lucidchart.com/pages/examples/
3- https://forum.arduino.cc/t/line-follower-code/374963
4- https://circuitdigest.com/microcontroller-projects/line-follower-robot-using-arduino
5- https://www.alldatasheet.com/
6- https://online.visual-paradigm.com/
7- https://joy-it.net/de/products/SEN-Color