Sensor Systems Lab Assignment #3

Obstacle Avoiding Mobile Robot

Group G8

Aytekin Dönmez 190412029

Umut Korkmaz 190412071

Berkehan Akyüz 190412052

Working Principle and Aim of the Design.

Our Project has aimed to product a obstacle avoiding robot car. We approach to the Project 2 main subject. First was the scaning the left front , right front and forward ways. The other subject was the give motion to the system. So according to this, We chose our components which is explained in components part. Then We start to product out design. Firstly we draw our base part then printed it on the 3D printer. Then we assemble the system starting with the motors. After that we start the write our codes. We were need some extra Arduino library which is afmotor because of the l293d motor shield. We decide de use power grinds ,2 and 13 pins. So we soldered some male headers to there.

Components

DC Gear Motor & Wheel

A gear motor is an all-in-one combination of a motor and gearbox. The addition of a gear head to a motor reduces the speed while increasing the torque output

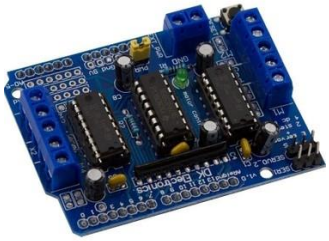
HC- SR04 Ultrasonic Sensor

The ultrasonic sensor works on the principle of SONAR and RADAR system which is used to determine the distance to an object. An ultrasonic sensor generates the high-frequency sound (ultrasound) waves.

Arduino

Arduino is an open-source electronics platform that is based on beginner-level hardware and software. The hardware component of an Arduino board is a programmable circuit board that is also known as a microcontroller

L293D Motor Shield



L293D motor driver shield can control up to four bi-directional DC motors with 8-bit speed selection, two stepper motors, and two servo motors.

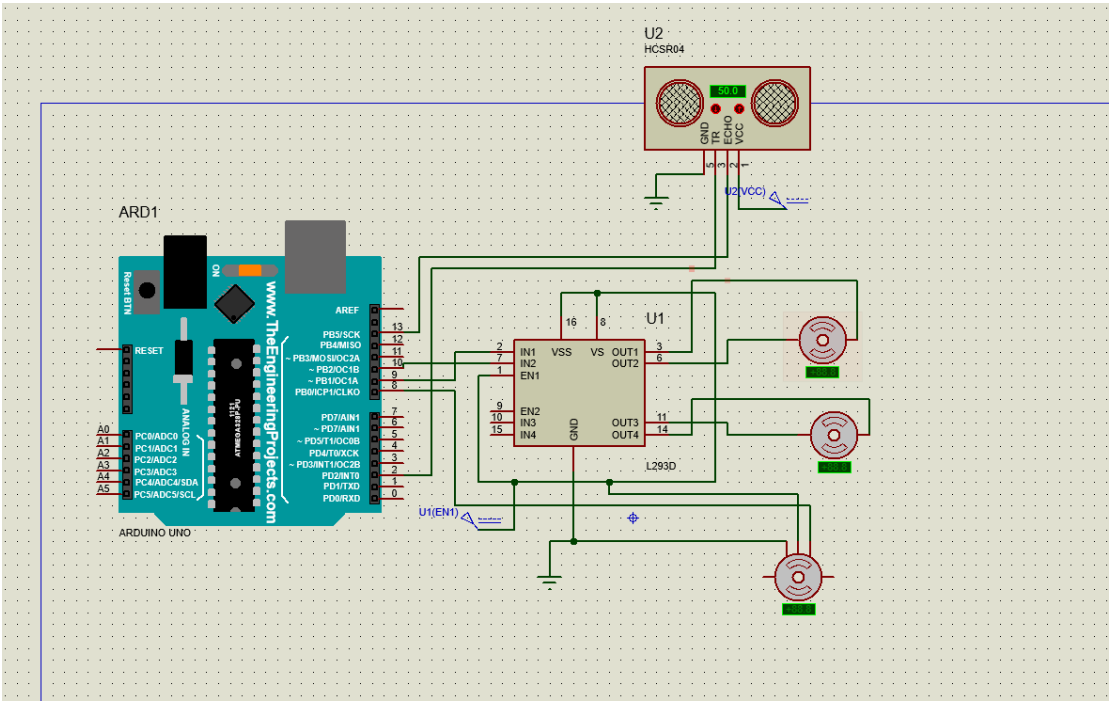
SG90 RC Servo Motor

Servo motor works on PWM (Pulse width modulation) principle, means its angle of rotation is controlled by the duration of applied pulse to its Control PIN. Basically servo motor is made up of DC motor which is controlled by a variable resistor (potentiometer) and some gears.

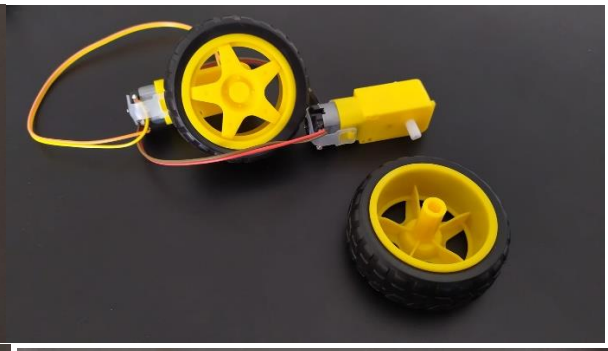
Chester Wheel

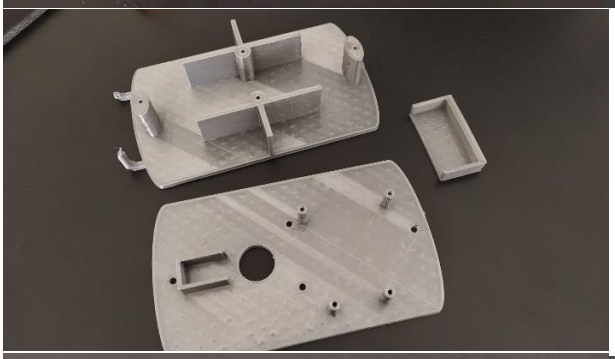
A caster is an undriven wheel that is designed to be attached to the bottom of a larger object to enable that object to be moved.

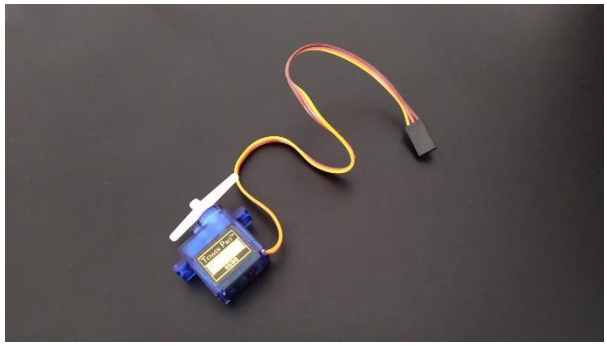
Simulation of the Circuit

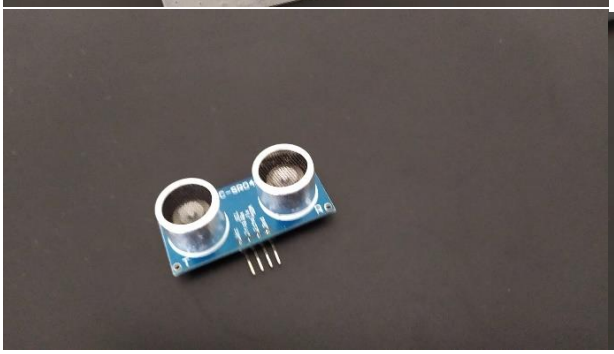


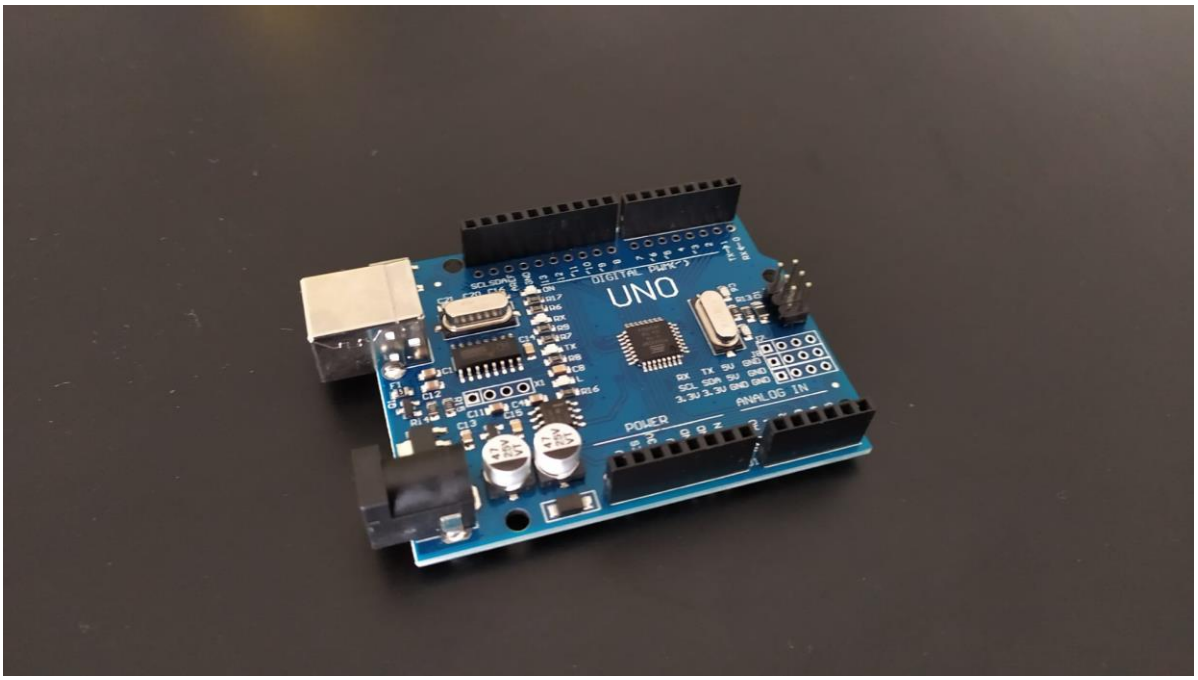
Photos of Components











Arduino Code to Drive Robot

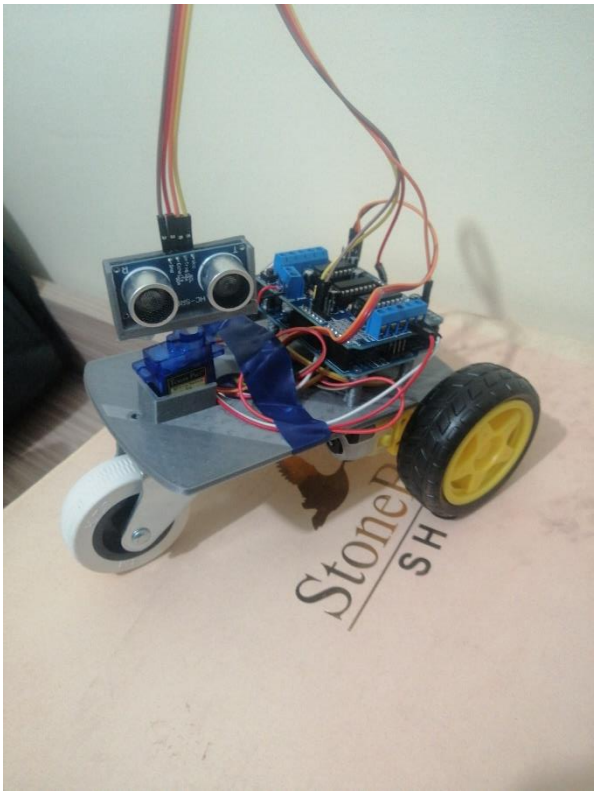
Robot_Code §

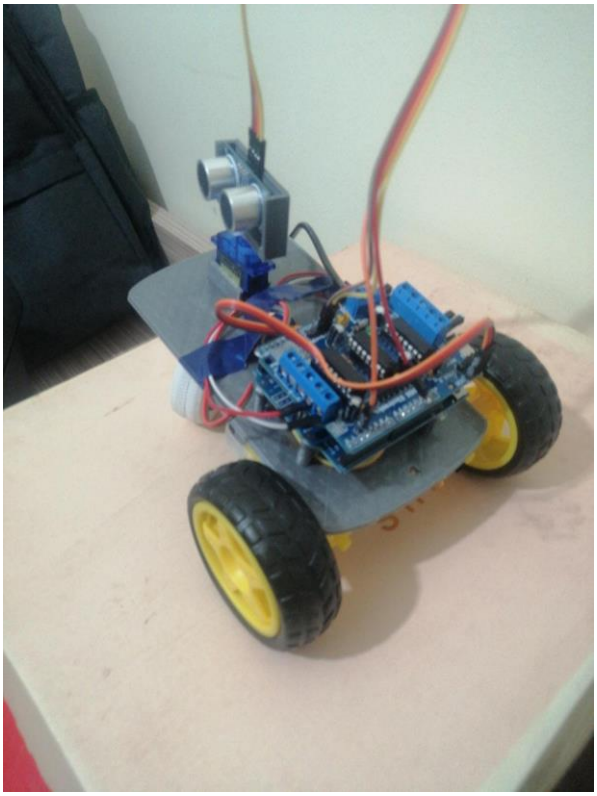
```
#include <AFMotor.h>
#include <Servo.h>

AF_DCMotor rightBack(1);
AF_DCMotor leftBack(4);
Servo servoLook;

byte trig = 2;
byte echo = 13;
byte maxDist = 150;
byte stopDist = 40;
float timeOut = 2*(maxDist+10)/100/340*1000000;

byte motorSpeed = 70;
int motorOffset = 10;
int turnSpeed = 50;


void setup()
{
  rightBack.setSpeed(motorSpeed);
  leftBack.setSpeed(motorSpeed+motorOffset);
  rightBack.run(RELEASE);
  leftBack.run(RELEASE);
  servoLook.attach(10);
  pinMode(trig,OUTPUT);
  pinMode(echo,INPUT);
}

void loop()
{
  servoLook.write(90);
  delay(750);
  int distance = getDistance();
  if(distance >= stopDist)
  {
    moveForward();
  }
  while(distance >= stopDist)
  {
    distance = getDistance();
  {
    distance = getDistance();
    delay(250);

  int turnDir = checkDirection();
  Serial.print(turnDir);
  switch (turnDir)
  {
    case 0:
      turnLeft (400);
      break;
    case 1:
      turnLeft (700);
      break;
    case 2:
      turnRight (400);
      break;
  }
}

void accelerate()
{
    for (int i=0; i<motorSpeed; i++)
    rightBack.setSpeed(i);
    loop speed
    leftBack.setSpeed(i+motorOffset);
    delay(10);
  }
}

void decelerate()
{
  for (int i=motorSpeed; i!=0; i--)
  {
    rightBack.setSpeed(i);
    leftBack.setSpeed(i+motorOffset);
    delay(10);
  }
}

void moveForward()
```

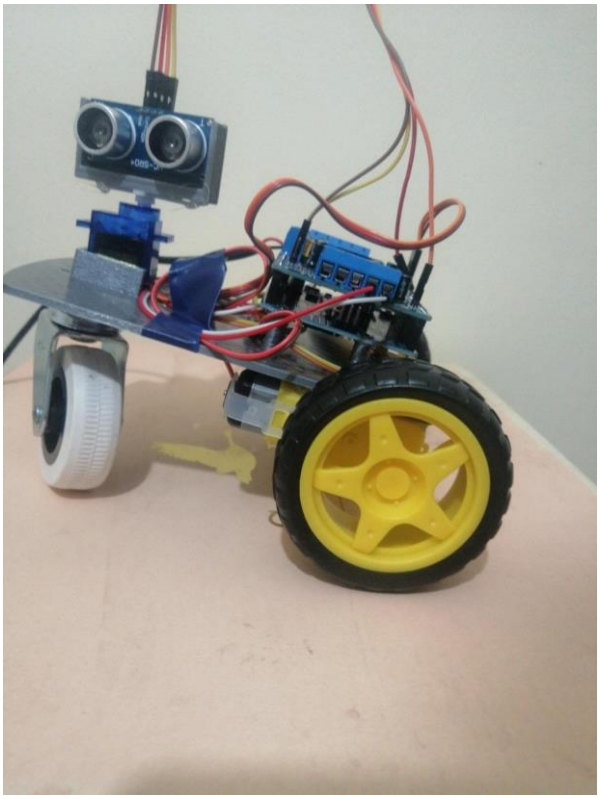
```
  rightBack.run(BACKWARD);
  leftBack.run(FORWARD);
  delay(duration);
  rightBack.setSpeed(motorSpeed);
  leftBack.setSpeed(motorSpeed+motorOffset);
  rightBack.run(RELEASE);
  leftBack.run(RELEASE);
}

int getDistance()
{
  unsigned long pulseTime;
  int distance;
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);
  pulseTime = pulseIn(echo, HIGH, timeOut);
  distance = (float)pulseTime * 340 / 2 / 10000;
  return distance;
}

int checkDirection()
{
  int distances [2] = {0,0};
  int turnDir = 1;
  servoLook.write(180);
  delay(500);
  distances [0] = getDistance();
  servoLook.write(0);
  delay(1000);
  distances [1] = getDistance();
  if (distances[0]>=200 && distances[1]>=200)
    turnDir = 0;
  else if (distances[0]<=stopDist && distances[1]<=stopDist)
    turnDir = 1;
  else if (distances[0]>=distances[1])
    turnDir = 0;
  else if (distances[0]<distances[1])
    turnDir = 2;
  return turnDir;
}
```

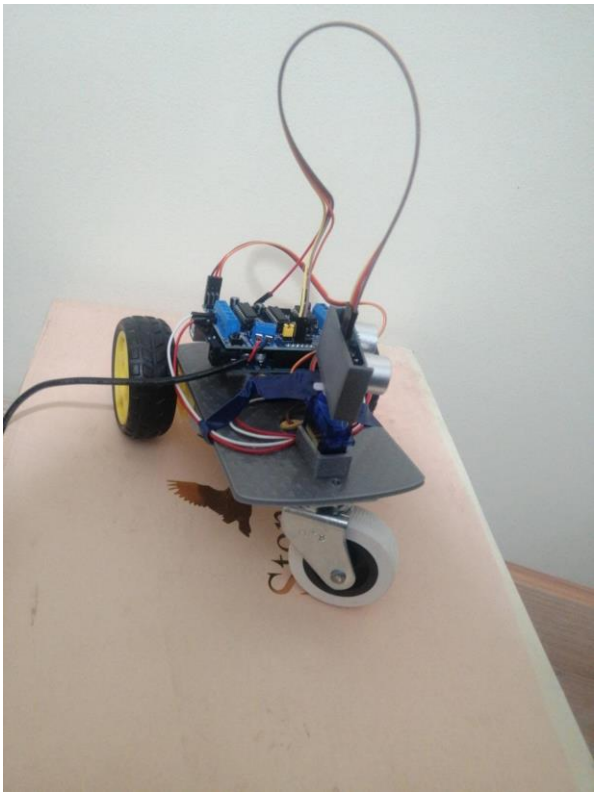
```

void moveForward()
{
  rightBack.run(FORWARD);
  leftBack.run(FORWARD);
}

void stopMove()
{
  rightBack.run(RELEASE);
  leftBack.run(RELEASE);
}

void turnLeft(int duration)
{
  rightBack.setSpeed(motorSpeed+turnSpeed);
  leftBack.setSpeed(motorSpeed+motorOffset+turnSpeed);
  rightBack.run(FORWARD);
  leftBack.run(BACKWARD);
  delay(duration);
  rightBack.setSpeed(motorSpeed);
  leftBack.setSpeed(motorSpeed+motorOffset);
  rightBack.run(RELEASE);
  leftBack.run(RELEASE);

}

void turnRight(int duration)
{
  rightBack.setSpeed(motorSpeed+turnSpeed);
  leftBack.setSpeed(motorSpeed+motorOffset+turnSpeed);
  rightBack.run(BACKWARD);
  leftBack.run(FORWARD);
  delay(duration);
  rightBack.setSpeed(motorSpeed);
  leftBack.setSpeed(motorSpeed+motorOffset);
  rightBack.run(RELEASE);
  leftBack.run(RELEASE);
}

int getDistance()
```

Photos of the Final Design









Conclusion

Consequently, we built a 3-wheeled obstacle-avoiding Arduino robot. In the building part, we used silicon to glue wheels, Arduino, and a servo motor. We soldered the pins 2 and 13 using male headers to the Arduino to connect with a motor driver. The challenging part was wire connections and the speed calibration of DC Motors. To get rid of the cable mess we taped them to the body. Thus, we prevented the contactless between cables. When we finished the wire connections and coding we tried to drive the robot with a 12V dc adaptor. It was working as we wanted. Then, we tried to connect the 12 V battery, unfortunately, the power wasn't enough to drive the motors.