A Real time Obstacle Detecting and Avoiding Car

Akash Das[1,], Avishek Ray[2], Animesh Chandra[1]*

[1]Department of Electronics and Communication Engineering, Institute of Engineering and Management, Kolkata-700091
[2]Post Doctoral Fellow, Dept of Energy Sc and Engineering, IIT Bombay

Email: akashdas250901@gmail.com

Abstract

An important function in self-driven cars is Obstacles Avoidance. Mapping the environment is indeed necessary for such vehicle to make a path planning and control. In this project we have concentrated on making an obstacle avoider, sensing the surrounding by using SONAR principle. This system provides a low cost replacement of LIDAR and SLAM technology. It is quite accurate to avoid obstacle in real time.

Keywords: *detection, obstacle, avoiding, SONAR, real time, low cost, solution*

1. Introduction

Autonomous vehicle system needs path planning, localization and obstacle detection and avoidance. Avoidance of both static and dynamic obstacles in front of the vehicle is an essential requirement for this type of vehicle. It is one of the critical technologies we are going to deal with. This will be of utmost importance in the future days, in the days of automation. Here we use a small prototype of the big project using the AT-mega micro- controller Arduino and ultrasonic sensor.

2. Materials required:

- Arduino Uno,
- Ultrasonic sensor HC-SR04,
- Motor Driver L293D,
- Servo Motor,
- Geared Motors*2,
- Robot chassis,
- Power Supply

3. Component description –

Arduino Uno is an ATmega 328p Microcontroller based prototyping board. It is an open source electronic prototyping platform which can be used with various sensors and actuators.Arduino Uno has 14 digital I/O pins out of which 6 pins are used in this project.

Fig 1

HC – SR04

It is a sensor for finding Ultrasonic Range Finder. It is a non-contact based distance measurement device having ability to measure distance of 2cm to 4m.

Fig 2

L298n -

It is a motor driver which can provide bi-directional drive current for two motors.

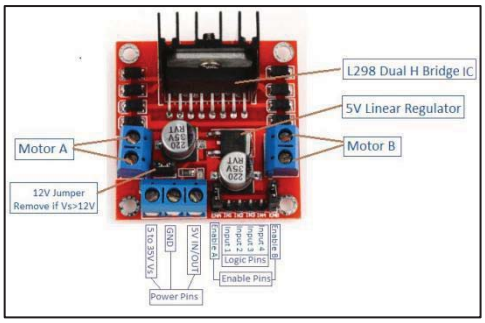

Fig 3

Servo Motor

The Tower Pro SG90 is a simple Servo Motor which can rotate 90 degrees in each direction (approximately 180 degrees in total).

Fig 4

Related work –

Here we will discuss the procedure to make a simple obstacle avoider car to act as a simple demo for self-driving machinery. It's a fully autonomous robot which can avoid obstacles. Simple when it meets an obstacle it will automatically stop and look in both sides to search for open space so that it can complete its path.

Here the ultrasonic sensor will send a beam of ultrasound and will receive it after it gets reflected from an obstacle. It will note the time evolved and using the speed of sound it will measure the distance of the obstacle from the vehicle.

Distance= (Time x Speed of Sound in Air (343 m/s))/2

Then if the distance measured is a less than a specified range, the motors will stop and will turn in a direction where it is free and will again start moving.

Assemble the parts. Solder two wires to the two geared dc motors. And fix the motor on the chassis. Mount the arduino Uno and l298n Servo motor on the chassis as per the fig 6.
Connect the pins of ultrasonic sensor to the arduino pins and mount it over the chassis .Connect the jumper wires as per the circuit diagram:

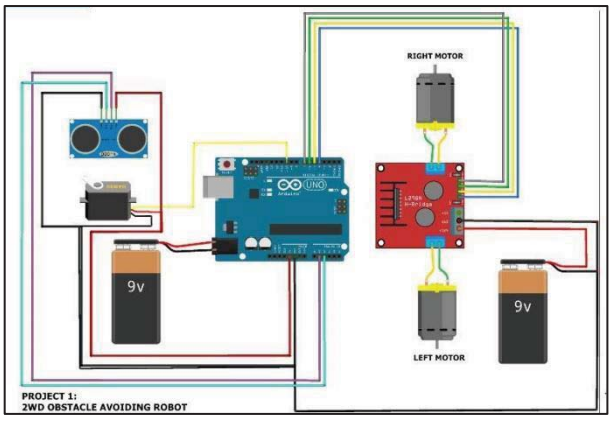

Fig 5

4. **Connections:**

L298n motor driver:

+12V → Lipo battery (+)

GND → Lipo battery (-) important: connect the GND to lipo battery (-) and to arduino board any GND pin
+5V → arduino Vin

In1 → arduino digital pin 7

In2 → arduino digital pin 6

In3 → arduino digital pin 5

In4 → arduino digital pin 4

OUT1 → Motor 1

OUT2 → Motor 1

OUT3 → Motor 2

OUT4 → Motor 2 Breadboard:

Connect two jumper wires to arduino board 5V and GND pins, then connect both wires to breadboard. now you can use this as +5V supply.

HC-SR04 Ultrasonic Sonar sensor:

VCC → breadboard +5V

Trig → arduino analog pin 1

Echo → arduino analog pin 2

GND → breadboard GND TowerPro micro servo 9g: orange wire → arduino digital pin 10 red wire →

breadboard +5V brown wire → breadboard GND

Now comes the programming part :- We have to install the arduino IDE for that . After we have to write the following code in that code editor and then upload the code to our micro controller that is the arduino uno.

Code snippet:-

```
//Coded by Akash Das.
#include <Servo.h>          //Servo motor library. This is standard library
#include <NewPing.h>
//Ultrasonic sensor function library.
//You must install this library
//our L298N control pins
```

```
const int LeftMotorForward = 7;
const int LeftMotorBackward = 6;
const int RightMotorForward = 4;
const int RightMotorBackward = 5;

//sensor pins
#define trig_pin A1 //analog input 1
#define echo_pin A2 //analog input 2
#define maximum_distance 200
boolean goesForward = false;
int distance = 100;

NewPing sonar(trig_pin, echo_pin, maximum_distance);   //sensor function
Servo servo_motor; //our servo name
void setup() {
  pinMode(RightMotorForward, OUTPUT); pinMode(LeftMotorForward, OUTPUT);
  pinMode(LeftMotorBackward, OUTPUT);
  pinMode(RightMotorBackward, OUTPUT);
  servo_motor.attach(10); //our servo pin
  servo_motor.write(115);
  delay(2000);
  distance = readPing();
  delay(100);
  distance = readPing();
  delay(100);
  distance = readPing();
  delay(100);
  distance = readPing();
  delay(100);

}

void loop() {

  int distanceRight = 0;
  int distanceLeft = 0;
  delay(50);
```

```
  if (distance <= 20) {
    moveStop(); delay(300);
    moveBackward(); delay(400);
    moveStop();
    delay(300);
    distanceRight = lookRight();
    delay(300);
    distanceLeft = lookLeft();
    delay(300);
    if (distance >= distanceLeft)
    {
      turnRight();
      moveStop();
```

```
    } else {
      turnLeft();
      moveStop();
    }
  } else
  {
    moveForward();
  }
  distance = readPing();
}
int lookRight()
{
  servo_motor.write(50);
  delay(500);
  int distance = readPing();
  delay(100);
  servo_motor.write(115);
  return distance;
}

int lookLeft() {
  servo_motor.write(170);
  delay(500);
  int distance = readPing();
  delay(100);
  servo_motor.write(115);
  return distance;
  delay(100);
}
int readPing() {
  delay(70); int cm = sonar.ping_cm();
  if (cm == 0) {

cm = 250;
  } return cm;
}

void moveStop() {
  digitalWrite(RightMotorForward, LOW);
  digitalWrite(LeftMotorForward, LOW);
  digitalWrite(RightMotorBackward, LOW); digitalWrite(LeftMotorBackward,
      LOW);
}

void moveForward() {
  if (!goesForward) {
    goesForward = true;
    digitalWrite(LeftMotorForward, HIGH);
    digitalWrite(RightMotorForward, HIGH);
    digitalWrite(LeftMotorBackward, LOW);
    digitalWrite(RightMotorBackward, LOW);
```

```
  }
}



void moveBackward() {
  goesForward = false;
  digitalWrite(LeftMotorBackward, HIGH);
  digitalWrite(RightMotorBackward, HIGH);
  digitalWrite(LeftMotorForward, LOW);
  digitalWrite(RightMotorForward, LOW);
}
void turnRight() {
  digitalWrite(LeftMotorForward, HIGH);
  digitalWrite(LeftMotorBackward, HIGH);
  digitalWrite(LeftMotorBackward, LOW);
  digitalWrite(RightMotorForward, LOW);
  delay(500);
  digitalWrite(LeftMotorForward, HIGH);
  digitalWrite(RightMotorForward, HIGH);
  digitalWrite(LeftMotorBackward, LOW);
  digitalWrite(RightMotorBackward, LOW);
}
void turnLeft() {
  digitalWrite(RightMotorBackward, HIGH);
  digitalWrite(RightMotorForward, HIGH);
  digitalWrite(LeftMotorForward, LOW);
  digitalWrite(RightMotorBackward, LOW);
  delay(500);
  digitalWrite(LeftMotorForward, HIGH);
  digitalWrite(RightMotorForward, HIGH);
  digitalWrite(LeftMotorBackward, LOW);
  digitalWrite(RightMotorBackward,LOW);
}
```

After uploading the code to arduino uno we just have to power the arduino from an external source ,remove the usb cable and can easily see how the vehicle works automatically and avoid both static and dynamic obstacles.

Finished Project :-

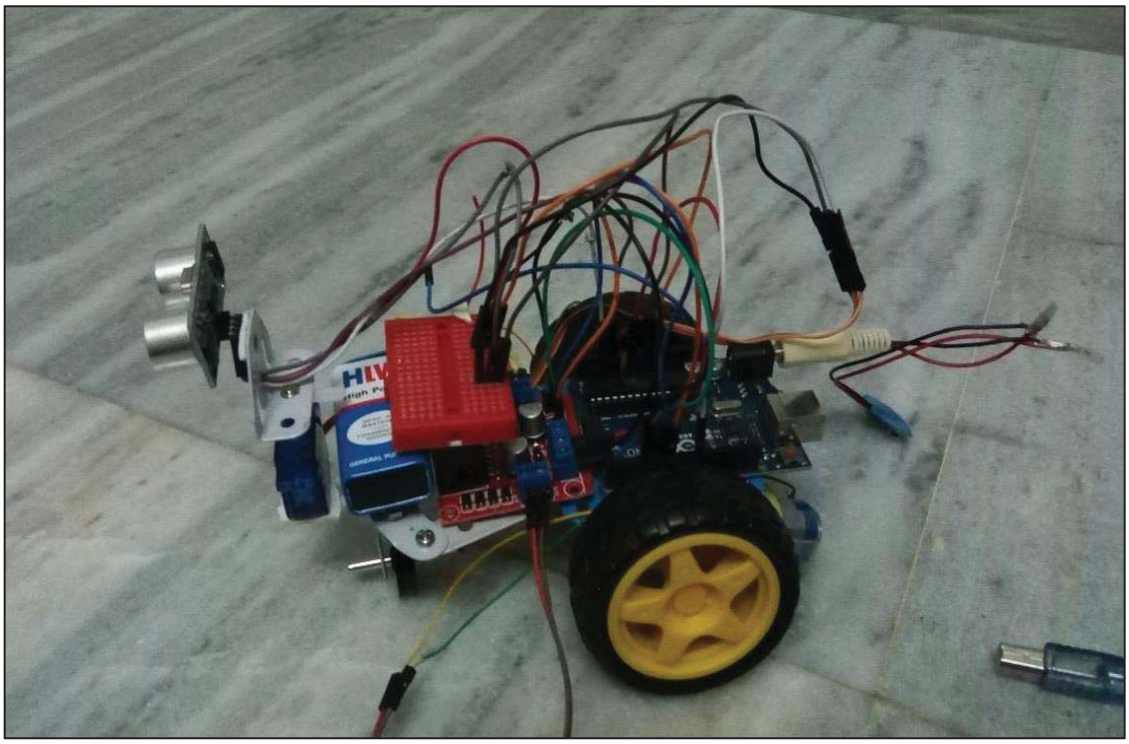

Fig 6

5. Conclusion

This project we have done is a low cost measure to avoid any sort of obstacles. With a little increase in the accuracy and range of the ultrasonic sensor we can perform our job efficiently and with no mistake. In practical case in self-driven cars or other forms of automatic vehicles , rovers it use a complex machinery and algorithm to perform the same , using LIDAR and SLAM technologies which are costly for regular purpose . So we think that working on SONAR principle can reduce the cost and can bring the facilities of this automatic vehicles in real life.

References

[1] Circuit Digest.

[2] Instructables, Lee, U.; Yoon, S.; Shim, H.; Vasseur, P.; Demonceaux, C. Local path planning in a complex environment for self-driving car

[3] Li, X.; Sun, Z.; Cao, D.; He, Z.; Zhu, Q. Real-Time Trajectory Planning for Autonomous Urban Driving

[4] Wang, P.; Gao, S.; Li, L.; Cheng, S.; Zhao, L. Automatic steering control strategy for unmanned vehicles based on Robust Back stepping sliding mode control theory