Zagazig University
Faculty of Engineering
Computer and Systems Engineering Dept.

Collision Avoidance Robot


Omar Ra'fat


Mustafa Kamel

Title:

CAR: Collision Avoidance Robot

Keywords:

Robotics, Mechatronics, Embedded Systems, Control, Simulation, Arduino

Project Period: Winter Semester 2016

Project Group: Group Name

Participants:

e Omar Ra'fat Abd-El-Latif sk1, 18
e Mustafa Kamel El-Sayed sk1, 24

Supervisor(s): Dr.Ing. Mohammed Ahmed

Numbers of Pages: **10**

Date of Completion: Friday, December 16, 2016

Abstract:

The robot consists of a car that avoids collisions by taking another path free of barriers. It uses an ultrasonic sensor to sense the obstacles then take a decision according to the input signal and the prespecified responses.

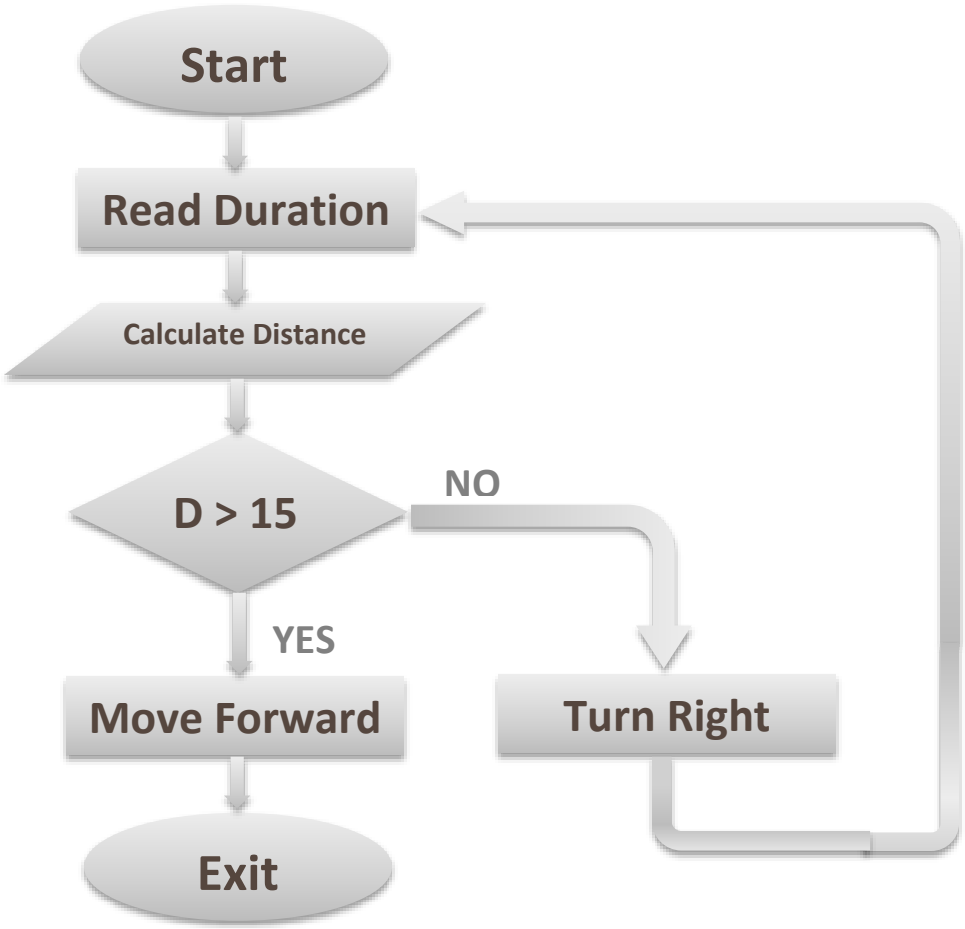
Contents

Description

The robot consists of a car that avoids collisions by taking another path free of barriers. It uses an ultrasonic sensor to sense the obstacles then take a decision according to the input signal and the prespecified responses.

At first the robot will send a signal to detect if there is an object in its path, if there is an object the signal will reflect and the ultrasonic sensor will detect the reflected signal then the distance will be calculated from the equation of the voice propagation speed in air and the calculated distance will be compared against 15cm, if the distance is larger than 15 the robot will go straight, but if the distance is less than or equal 15 the robot will keep turning right and measuring the distance until it found a distance larger than 50cm then it will move forward. The robot will repeat the past algorithm until it is terminated by the administrator.

Flow chart



Software

Declaration area:

```
// defining pins numbers

const int vcc = 8;            // Input power of the ultrasonic sensor

const int gnd = 11;           // Ground of the ultrasonic sensor

const int trigPin=9; // Trigger [the input pin of the ultrasonic sensor]

const int echoPin=10; // Echo [the output pin of the ultrasonic sensor]

// defining variables

int currentDist; // Current distance away from the obstacle [estimated by the ultrasonic sensor]

int rightf=4; // Right wheel forward actuator

int rightr=5; // Right wheel reverse actuator

int leftf=6;// Left wheel actuator [forward only]
```

Setup Function:

```
void setup(){

  pinMode(trigPin,OUTPUT); // Sets the trigPin as an Output

  pinMode(echoPin,INPUT); // Sets the echoPin as an Input

  pinMode(rightf,OUTPUT); // Sets the right as an Output

  pinMode(rightr,OUTPUT); // Sets the right as an Output

  pinMode(leftf,OUTPUT); // Sets the left as an Output

  pinMode(vcc,OUTPUT); // Sets the right as an Output

  pinMode(gnd,OUTPUT); // Sets the right as an Output
}
```

Distance Function:

```
int distance(){

/*The purpose of this method is to calculate the actual distance

* away from the nearst object by sending a signal and measuring

* the time between sending it and receiving it then the distance

* can be calculated from the voice propagtion speed in air equation

* the method provides an easy way to measure the distance instead

* of writing the code everytime

*/

  // defining variables

 long duration; // holds the duration between sending the signal and receiving it

 int distance; // holds the calculated distance in cm unit

  // Clearing the trigPin signal by setting it to low for 2 microseconds

 digitalWrite(trigPin,LOW);

 delayMicroseconds(2);

 /* Setting the trigPin on HIGH state for 10 microseconds to

 send a signal to detect if there is an obstacle in the path*/

 digitalWrite(trigPin,HIGH);

 delayMicroseconds(10);

 digitalWrite(trigPin,LOW);
```

```
 // Reading the echoPin, returns the sound wave travel time in microseconds
 duration=pulseIn(echoPin, HIGH);
 /* Calculating the distance as the duration holds the signal
  *length in microseconds and the voice propagation speed in air
  *is 340 m/s, this equation gives the distance in cm unit
  */
 distance=duration*.034/2;
 return distance;
}
```

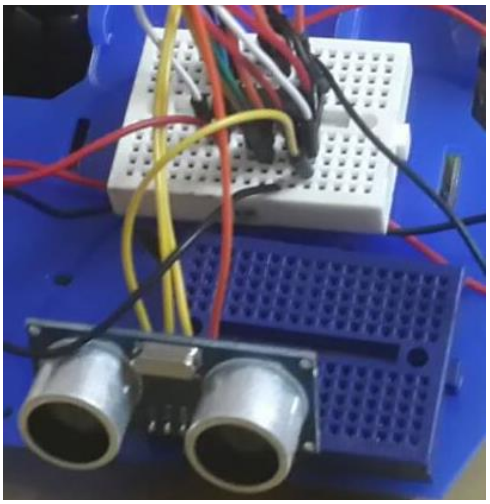
Loop Function:

```
void loop(){
//Vcc and gnd for the ultrasonic sensor
 digitalWrite(vcc,HIGH);
 digitalWrite(gnd,LOW);
 /*calling the distance method to
 * calculate the current distance away from the nearest obstacle
 * then test if the current distance is more than 15 cm keep
 * moving forward otherwise turn right until it finds a free
 * path with at a path at least 50cm away then move forward
 */
```

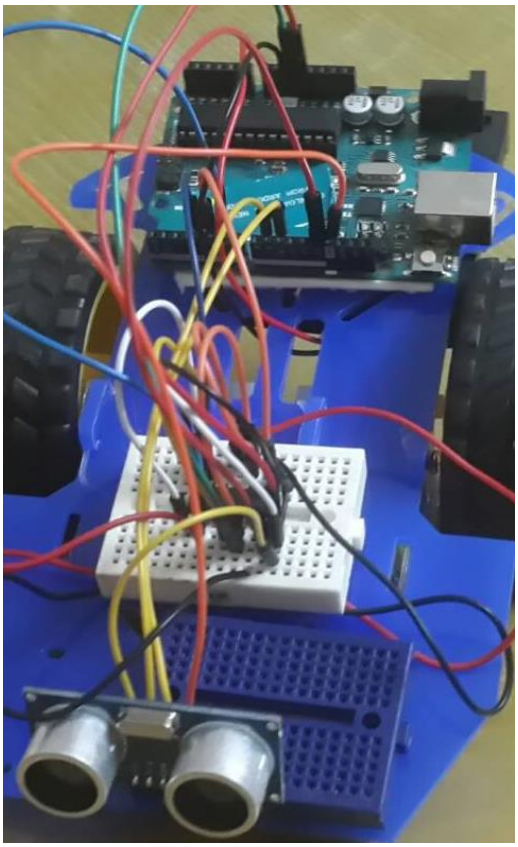
```
currentDist=distance();
 if (currentDist > 15){
 // moving forward can be done by running the two wheels forward
   digitalWrite(rightf,HIGH);
   digitalWrite(rightr,LOW);
   digitalWrite(leftf,HIGH);
 }
 else if(currentDist <= 15){
   do{
/* this loop will continue in turning the robot right until
 the distance to the nearest object becomes less than or equal 50cm*/
 /* turning right can be done by runnig the right wheel
   reverse to back and the left wheel forward*/
   digitalWrite(rightf,LOW);
   digitalWrite(rightr,HIGH);
   digitalWrite(leftf,HIGH);
   currentDist=distance();
   }while(currentDist <=50);
 }
}
```

Hardware

Actuator Circuit



Overall Circuit



Summary

The final product will be like