# CH5 : Obstacle avoiding Robot

## Introduction

Obstacle Avoidance Robot which can automatically sense and overcome obstacles on its path. It contains of an Arduino board to process data, and Ultrasonic sensors to detect the obstacles on its path.

First of all we learn how to control Servo motor using Arduino
then how to measure the distance of the object in front of the robot using Ultrasonic sensor
how to use functions in programming to make the code organized
and final how to integrate the whole component together the Ultrasonic sensor – Servo motor and robot car to make the Obstacle Avoidance Robot

Let's talk about the required component to make the obstacle avoiding robot

First of all the Arduino board which will send a trigger signal to the ultrasonic sensor to measure the distance to an object in front of the robot and based on that distance Arduino drive the servo motor so robot car can look to right and left to check where to go and finally drive the motors so the Robot car can move forward or rotate
the average price of Arduino is 3$

the breadboard which is used for temporary prototype with electronics and test circuit designs
the average price of breadboard is 1.25$

Ultrasonic sensor which will be used to measure the distance in front of the robot
the average price of the Ultrasonic sensor is 3.8**$** for piece

tilt kit with servo motor to give the robot ability to look right and left
the average price of the tilt kit with servo motor is 7$

the 4 wheels robot car and it's average price is 15$

wires used to interconnect the components of a breadboard

Motor driver L293 which takes a low-current control signal from the Arduino and then turn it into a higher-current signal that can drive a motor
the average price of the Motor driver L293 is 1.23$

the overall cost of the obstacle avoiding robot will not exceed 40$

# Control Servo Motor using Arduino

Servo motors give us the ability to turn to a specified position.
they have a servo arm that can turn 180 degrees. Using the Arduino, we can tell a servo to go to a specified position.
We gonna fix the Ultrasonic sensor on the top of Servo motor so the Robot car can look to the right and to the left then the robot can avoid obstacle

Using just one input pin, they receive the position from the Arduino and they go there. Internally, they have a motor driver and a feedback circuit that makes sure that the servo arm reaches the desired position. But what kind of signal do they receive on the input pin?
It is a square wave Each cycle in the signal lasts for 20 milliseconds and for most of the time, the value is LOW. At the beginning of each cycle, the signal is HIGH for a time between 1 and 2 milliseconds. At 1 millisecond it represents 0 degrees and at 2 milliseconds it represents 180 degrees. In between, it represents the value from 0–180.

Servo motors have three wires: power, ground, and signal. The power wire is typically red, and should be connected to the 5V pin on the Arduino board. The ground wire is typically black or brown and should be connected to a ground pin on the Arduino board. The signal pin is typically yellow, orange or white and should be connected to a digital pin on the Arduino board. Note that servos draw considerable power, so if you need to drive more than one or two, you'll probably need to power them from a separate supply ( not the +5V pin on your Arduino) we gonna clarify how to use more than one servo motor later.
Let's write the code to make the servo motor look left for 2 sec and look right for 2 sec and check the final result

first of all the code contain two main function setup and loop
include the servo library to give us the ability to control servo motors with easy and simple instructions with the instruction #include <Servo.h>
Create a servo object using the instruction Servo Servo1;
in setup function it's required to attach the servo to the used pin number in the Arduino and we attached the Servo to the digital terminal D8 in the Arduino so we write
Servo1.attach(8)
in the loop function make servo go to 0 degrees with the instruction
Servo1.write(0)
then tell Arduino to do nothing for 2 second → delay(2000)
make servo go to 180 degrees with the instruction → `Servo1.write(180)`
then tell Arduino to do nothing for 2 second → delay(2000)
we gonna talk about libraries and objects in details later but right now all you have to know about Servo library that it contain codes that give us the ability to control servo motors with easy and simple instructions and Servo1 is an object from the Servo library that give us the ability to deal with the library
that it let's upload the code and check the final result

# Read the Distance using Ultrasonic sensor and Arduino

the function of ultrasonic sensor uses sonar is determine distance to an object.
It offers excellent non-contact range detection with high accuracy From 2cm to 400 cm.
We gonna use the Ultrasonic sensor to determine whether there is an object in front of the robot or not

first of all let's talk about how the ultrasonic sensor and Arduino works to measure the distance ?
1 – Arduino send a trigger signal to the ultrasonic sensor to measure the distance to an object in front of it
2 – Ultrasonic sensor high-frequency sound pulse and in the same time it output a 5V signal on the Echo terminal (which is connected with Arduino)
3 – Ultrasonic sensor waits till the echo of the sound to reflect back and then it output a 0V signal on the Echo terminal (which is connected with Arduino)
4 – Arduino determine the time of difference between sending and receiving the sound pulse to determine the distance to an object (as The speed of sound 340 m/sec in air)

the Ultrasonic sensor contains 4 terminals power,Ground, Trig and Echo
Connect Vcc terminal in the Ultrasonic sensor with +5V in the Arduino
Connect Gnd terminal in the Ultrasonic sensor with Gnd in the Arduino
Connect trig terminal in the Ultrasonic sensor with D2 in the Arduino
connect Echo terminal in the Ultrasonic sensor with D4 in the Arduino
let's write the code to read the distance to an object in front of the Ultrasonic sensor

first of all the code contain two main function setup and loop
initialize the speed of sending and receiving data between Arduino and laptop so we can display the distance on the screen of the laptop with the instruction Serial.begin
in the loop function
declare two variable as long which is the same as integer but it can contain greater numbers
declare duration and cm as long
Arduino will send the a trigger signal to the Ultrasonic sensor so initialize the digital terminal D2 ( which is connected to trigger terminal in the ultrasonic sensor) as an output using the instruction pinMode(2,OUTPUT)
then send the required signal to activate the ultrasonic sensor to measure the distance
the signal which is +5V for 10 microseconds so
determine the output to be 0V for 2 microseconds then +5V for 10 microseconds the 0V with the instructions
digitalWrite(trigPin, LOW);delayMicroseconds(2);digitalWrite(trigPin, HIGH);
delayMicroseconds(10); digitalWrite(trigPin, LOW);
Arduino will receive the signal from the echo terminal using the digital terminal D4
so initialize D4 as an input using the instruction pinMode(4,INPUT)
the time of the echo of the sound to reflect back will be measured in microseconds using the instruction pulseIn(4,HIGH)
the sound speed in air is 340m/s which mean it takes 29 microsecond for 1 cm
then cm= (duration/29)/2 → we have to divide by 2 because the time is measured while the sound waves reach the object and come back to the sensor and it's required the time only to reach the sensor
let's upload the code and check the final result

# How to use functions to simplify the code

The typical case for creating a function is when one needs to perform the same action multiple times in a program. Functions also help the programmer stay organized.
There are two required functions in an Arduino sketch, setup() and loop(). Other functions must be created outside the brackets of those two functions.

**Let's write a simple function to add two integers**
first of all the code contain two main function setup and loop
initialize the speed of sending and receiving data between Arduino and laptop with the instruction Serial.begin
in the loop function  declare x and y as an integers and initialize value to be 0
declare result as an integer and initialize value to be 0
let's build function to add the 2 numbers

the main structure of the function
Return_data_type  function_name  (parameters){}
let's say the function name will be add
we gonna add two integers so the return data will be integers
and the parameters will be the two integers (int n,int m)
let's write the instructions in the add the functions
int r = n +m ;
and return the value of r

then let's continue in the loop function
result = add (x,y);
we call the function call and the pass the value of x to replace n and  pass the value of y to replace m then the r which contain n +m which is x+y will be copied to the result variable
let's print the result with the instruction serial.println(result)

**Let's write another  function to read the distance of the object in front of the ultrasonic sensor**
the name of the function will be ReadDistance
and there are no required parameter
and the return data type will be long because from the previous workshop we use the type of cm as long

first of all the code contain two main function setup and loop
initialize the speed of sending and receiving data between Arduino and laptop with the instruction Serial.begin

then build the new function long  ReadDistance(){}
declare two variable as long which is the same as integer but it can contain greater numbers
declare duration and cm as long
Arduino will send the a trigger signal to the Ultrasonic sensor so initialize the digital terminal D2 ( which is connected to trigger terminal in the ultrasonic sensor) as an output using the instruction pinMode(2,OUTPUT)
then send the required signal to activate the ultrasonic sensor to measure the distance
the signal which is +5V for 10 microseconds so
determine the output to be 0V for 2 microseconds then +5V for 10 microseconds the 0V with the instructions
digitalWrite(trigPin, LOW);delayMicroseconds(2);digitalWrite(trigPin, HIGH);
 delayMicroseconds(10); digitalWrite(trigPin, LOW);
Arduino will receive the signal from the echo terminal using the digital terminal D4
so initialize D4 as an input using the instruction pinMode(4,INPUT)

the time of the echo of the sound to reflect back will be measured in microseconds using the instruction pulseIn(4,HIGH)
the sound speed in air is 340m/s which mean it takes 29 microsecond for 1 cm
then cm= (duration/29)/2
we have to divide by 2 because the time is measured while the sound waves reach the object and come back to the sensor and it's required the time only to reach the sensor

then return the value of cm

in the loop function initialize variable result as long to store the distance of the object in front of the sensor
then  result =  ReadDistance();
and print the result using Serial.println( result)
that's it let's upload the code and check the result

we gonna use the functions in the obstacle avoiding robot programming to organize the code

# Final Project: Obstacle avoiding Robot

you learn how to use the Ultrasonic sensor to measure the distance of the object in front of the robot  how to control servo motor so the robot can look to right and left
let's integrate that whole component to make the Obstacle avoiding Robot

first of all the motor driver L293 is connected with Arduino the same as it's connection in chapter 2
the signal terminal of the servo motor is connected to the digital terminal D8 and the power and ground of the servo motor is connected to +5V and ground
the Echo terminal in the ultrasonic sensor is connected the digital terminal D4 in the Arduino
the Trig terminal in the ultrasonic sensor is connected the digital terminal D2 in the Arduino
the power and ground of the ultrasonic sensor is connected to +5V and ground in the Arduino
let's write the code and check the final result

first of all the code contain two main function setup and loop
let's make a function to read distance the same as the previous workshop

the name of the function will be ReadDistance
and there are no required parameter
and the return data type will be long because from the previous workshop we use the type of cm as long
long  ReadDistance(){}
declare two variable as long which is the same as integer but it can contain greater numbers
declare duration and cm as long
Arduino will send the a trigger signal to the Ultrasonic sensor so initialize the digital terminal D2 ( which is connected to trigger terminal in the ultrasonic sensor) as an output using the instruction pinMode(2,OUTPUT)
then send the required signal to activate the ultrasonic sensor to measure the distance
the signal which is +5V for 10 microseconds so
determine the output to be 0V for 2 microseconds then +5V for 10 microseconds the 0V with the instructions
digitalWrite(trigPin, LOW);delayMicroseconds(2);digitalWrite(trigPin, HIGH);
 delayMicroseconds(10); digitalWrite(trigPin, LOW);
Arduino will receive the signal from the echo terminal using the digital terminal D4
so initialize D4 as an input using the instruction pinMode(4,INPUT)
the time of the echo of the sound to reflect back will be measured in microseconds using the instruction pulseIn(4,HIGH)
the sound speed in air is 340m/s which mean it takes 29 microsecond for 1 cm
then cm= (duration/29)/2
we have to divide by 2 because the time is measured while the sound waves reach the object and come back to the sensor and it's required the time only to reach the sensor

then return the value of cm

let's make another function for the Robot car to move forward
it's name will be MoveForward and there is no required parameter and it will return nothing
so we write
void → which mean that function will no return any data
and then the function name which is  MoveForward (){
from Ch2 the line follower robot and ch3 smart robot we can make the robot to move forward with
digitalWrite(13,HIGH); digitalWrite(12,LOW); that make one side of robot car move forward
digitalWrite(11,HIGH); digitalWrite(10,LOW); that make the other side of robot car move forward
so the robot will move forward

let's make another function for the Robot car to move backward
it's name will be Move backward and there is no required parameter and it will return nothing
so we write
void → which mean that function will no return any data
and then the function name which is Movebackward (){
from Ch2 the line follower robot and ch3 smart robot we can make the robot to move backward with
digitalWrite(13,LOW); digitalWrite(12,HIGH); that make one side of robot car move backward
digitalWrite(11,LOW); digitalWrite(10,HIGH); that make the other side of robot car move backward
add delay (200);
so the robot will move backward

let's make another function for the Robot car to move Right
it's name will be MoveRight and there is no required parameter and it will return nothing
so we write
void → which mean that function will no return any data
and then the function name which is MoveRight (){
from Ch2 the line follower robot and ch3 smart robot we can make the robot to move Right with
digitalWrite(13,HIGH); digitalWrite(12,LOW); that make one side of robot car move Forward
digitalWrite(11,LOW); digitalWrite(10,HIGH); that make the other side of robot car move backward
so the robot will move right
add delay (600); so the robot rotates for 90 degree

let's make another function for the Robot car to move left
it's name will be Moveleft and there is no required parameter and it will return nothing
so we write
void → which mean that function will no return any data
and then the function name which is Moveleft (){
from Ch2 the line follower robot and ch3 smart robot we can make the robot to move left with
digitalWrite(13,LOW); digitalWrite(12,HIGH); that make one side of robot car move Backward
digitalWrite(11,HIGH); digitalWrite(10,LOW); that make the other side of robot car move Forward
so the robot will move left
add delay (600); so the robot rotates for 90 degree

let's make another function for the Robot car to stop
it's name will be stop and there is no required parameter and it will return nothing
so we write
void → which mean that function will no return any data
and then the function name which is stop (){
from Ch2 the line follower robot and ch3 smart robot we can make the robot to stop with
0V on D13,D12,D11 and D10

include the servo library to give us the ability to control servo motors with easy and simple instructions with the instruction #include <Servo.h>
Create a servo object using the instruction Servo Servo1;
in setup function it's required to attach the servo to the used pin number in the Arduino and we attached the Servo to the digital terminal D8 in the Arduino so we write
Servo1.attach(8)
initialize D13,D12,D11 and D10 and an output with the instruction pinMode (13,12,11 and 10 , OUTPUT);
in the loop function
declare dcenter, dright and dleft as long

read the distance using the function that we build right now and store the result in variable and it's type is long

dcenter = readDistance();

if (  dcenter < 35 ){ Stop(); // stop the robot car
 then check the distance to the right
Servo1.write(0);
wait for 1000ms
read the distance using the function that we build right now and store the result in variable dright and it's type is long
dright = readDistance();

then check the distance to the left
Servo1.write(180);
wait for 1000ms
read the distance using the function that we build right now and store the result in variable dleft  and it's type is long
dleft = readDistance();
Servo1.write(90); //return to center
then
if (leftDistance>rightDistance) //if left is less obstructed { moveLeft(); }
else //if right is less obstructed { moveRight(); }
and finally tell the robot to move forward for 100msecond

let's upload the code and check the final result