Abstract

The motto behind making this Obstacle Avoiding Robot is to provide a solution in factories where lots of human power is required. This Robot will not only work continuously but will also detect obstacles in his way. It has advanced features like turn left/right direction to avoid an object in its path. This robot will stop the work immediately and begin with the same force when it has a clear way. Here the system has an LCD screen that shows the current RPM (Rotation Per Minute) of a DC motor. There is a shaft rotor motor that shows the arm movement of the robot. The system has Ultrasonic sensors which will detect the presence of an object. Ultimately, the system is a perfect stimulation of a real time robot which is ready to deploy in the factories.

1. Introduction

In this current world, every employer needs a quick and accurate output from all the employees. Every employer needs reliable and less costing manpower to make more profit. But the current market requires modern solutions where humans can be replaced by robots. This situation will reduce the risk of the employee injury during work, it will also reduce the possibilities of errors in the work, and it will be a one time investment for the employer. This solution will provide them reliable and accurate working outputs.

Currently, many multinational companies are using robots at their manufacturing units. But those giant robots are designed for specific work only, they all are giant sized, hard to relocate and require more and frequent maintenance. This solution would be small. It would be even more easy to relocate, it would be cheaper and less-easy maintenance. The demo model is designed in MATLAB Simulink. It is very easy to design on an Arduino circuit.

This project is a study project, which we have developed under the Department of Computer and Electrical Engineering at the University of Windsor. This project is an alternative option of the final exam for the subject, “Computational Methods and Modeling for Engineering Applications”. This project is developed by a team in which all students have expertise in different fields. The learning outcomes from the project are: critical thinking, problem-solving, MATLAB and Simulink usage, understanding of teamwork and time management.

2. Components used in Obstacle Avoiding Robot

2.1. Arduino Uno

Arduino Uno is an open-source microcontroller board. The Arduino Uno board is enabled with a group of digital and analog input/output pins. Microcontrollers are widely used in embedded systems. There are many kinds of Arduino boards such as Arduino Uno, Arduino Due, Arduino Leonardo and Arduino Mega where Arduino Uno and Arduino Mega are the most common versions. The figure of Arduino Uno is shown below:

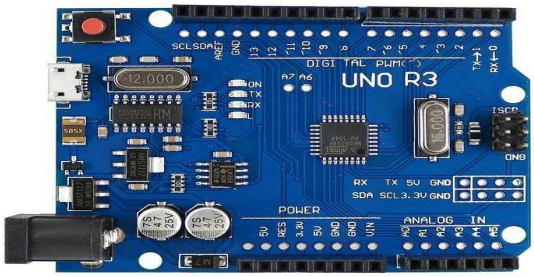

Figure - 1 Arduino Uno

2.2. LCD Display

The LCD display is controlled by the LiquidCrystal library. It usually consists of a 16 pin interface. The LCDs work on a parallel interface, meaning that the microcontroller has to operate several interface pins at one time to control the display.

Figure - 2 LCD Display

2.3. Servo Motor

Servo motor consists of a servo arm which can diverge to 180 degrees. The servo can turn to a specified position with the help of Arduino. It mostly contains everything inbuilt like a motor, a motor driver and a feedback circuit. Servo motor on the Arduino board is controlled by the servo library.

Figure - 3 Servo Motor

2.4. Ultrasonic Sensor

SONAR is used by an ultrasonic/distance sensor to detect the distance of the object. The working of the sensor is not affected by black material or sunlight. It consists of an ultrasonic transmitter and an ultrasonic receiver module.

Figure - 4 Ultrasonic Sensor

2.5. 3-6V DC Motor

There are 3 types of motor of which DC(Direct Current) motor is the most common. It

generally consists of two leads i.e., one positive and one negative. The DC motor will rotate to change direction from right to left or left to right whenever the object is detected.

Figure - 5 DC Motor

2.6. L293D Motor Driver Chip

The motor driver chip is an integrated circuit chip used to control motors in automated robots. It is used to control small dc motors and toy motors. It behaves as an interface between motors and Arduino. The L293 series of motor driver chips is the most commonly used chip.

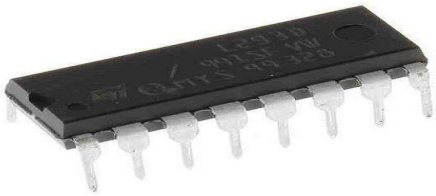

Figure - 6 L293D Motor Driver Chip

2.7. 220-ohm Resistor

The 220-ohm resistor contains 4 color strips on it. The one strip at the end which shows tolerance. A resistor is an electrical component having two terminals. Resistors are used to decrease the current flow, simultaneously it also reduces the voltage within the circuit.

Figure - 7 220-ohm Resistor

2.8. LED

LED stands for Light Emitting Diode. It is a small light that works efficiently with little power. It only lights up when you orient the legs properly in the Arduino board. It consists of polarity, the long leg connects to the digital pin of the Arduino board and the short leg connects to the GND(ground).

Figure - 8 LED

2.9. Jumper Wires and Male-to-Female Jumper Wires

A jumper wire is a simple electrical wire having a connector at each end. Jumper wires are used to connect the slots in breadboards by inserting their end connectors. Male to female jumper wires consist of two different ends, male ends have pins and can plug in while female ends have nothing but things they can plug into.

Figure - 9 Wires (a) Jumper Wires, (b) Male to female jumper wires

3. Explanation of Simulink Blocks

In this section, every add ons used in the project and various simulink blocks used in the project are explained. The Matlab version used for the project is 2018a.

3.1. Add on used in this project

1. Simulink support package for arduino hardware
2. Simulink support for Liquid Crystal Display
3. SImulink support for Arduino sensors

3.2. Different Simulink blocks used in the project

1. User Defined Function

User Defined function in Simulink is used for users to create their own function. This function can take variable inputs and can return a variable number of outputs. In this project, 6 user defined functions have been used. Every function's logic has been discussed in the Logic section.

2. Uniform Random Number

The Uniform Random Number block generates uniformly distributed random numbers over a specifiable interval with a specifiable starting seed. In this project, the arm of the servo motor should be rotated randomly. So, a random number block has been used in this project. The minimum number generated for this project is 0 and Maximum is 1.

3. Data Store Memory

Data Store Memory is used for storing the value of variables in Simulink. Data store memory can save most types of data like integer, real values, double, single and Boolean. In this project, 2 data store memories have been used named 'A' and 'B'.

First one is ‘A’. A has been used to save the position of the servo motor. The value of A would be 2, when the Robot is going forward. If the value of A is less than 0.5 then the servo motor’s arm is at left or 0 degree and if the value of A is greater than 0.5 and less than 1 then the servo motor’s arm is at right direction or 180 degree.

Second is ‘B’. B has been used for saving the value of A when the random number is generated. The value of B will be 0 or 1. First the value of B would be 0 and if the obstacle is detected by a sensor then it will become 1. Random numbers are generated every 0.5 seconds, but we need random numbers only when an obstacle is detected. So, when an obstacle is detected, a random number is saved in A and the value of B would become 1 and when the obstacle is not detected the value of B will become 0 again. So, basically the data store B changes the value of A according to object detection.

4. **Digital Output**

   Digital Output writes the logical value 0 or 1 to the digital pin of Arduino. Digital outputs have been used for LED pin 2 and the DC motor is connected to pin 10 and 11.

5. **Ultrasonic Sensor**

   Ultrasonic sensor measures the distance between sensor and object in meters and returns a double-precision value. If the object is placed beyond the sensor’s detection then output returns 0. This block is used for detecting objects in the project. Ultrasonic sensor has 2 pins, trigger and echo pin. Trigger pin is connected to pin 9 and echo pin is connected to pin 8 in Arduino.

6. **Data Store Memory read**

   Data store memory read block read the value from specific data store memory. This block can be used to read different variables. In the project, different functions have used data store memory read to read the value of A and B at different instant.

7. **Data store memory write**

   Data store memory write block writes value in the data store memory. After manipulation of different variables, this block will change the value in data store memory. In the project, different functions read values from data store memory read and change the value according to conditions and write them to data store memory write.

8. **Standard Servo read**

   Standard servo read block reads the position of the shaft of the servo motor in degrees. The value will be between 0 and 180. Display block will read the value of the servo motor from standard servo read and display in Simulink.

9. **Standard Servo write**

   Standard servo write block sets the position of the shaft of the servo motor in degrees. The range for standard servo write is between 0 and 180. In the project, standard servo writes the position according to the value of A. Standard servo write is connected to pin 7 of the Arduino.

10. **LCD display**

    LCD display is a 16 X 2 display and takes 2 inputs. It can display a maximum 16 character strings of two rows. The block accepts two 1-D arrays of uint8 precision. First

input displays the top row and the second input displays the bottom row. Connections for LCD display are as follows. Rs, En, D0, D1, D2, D3 are connected to pins 13, 12, 6, 5, 4, 3 respectively.

11. Display Block

Display blocks take input values and display their numeric values in Simulink. It can be used to check if different values are correct or not.

4. Overall Logic of the System and Flowchart

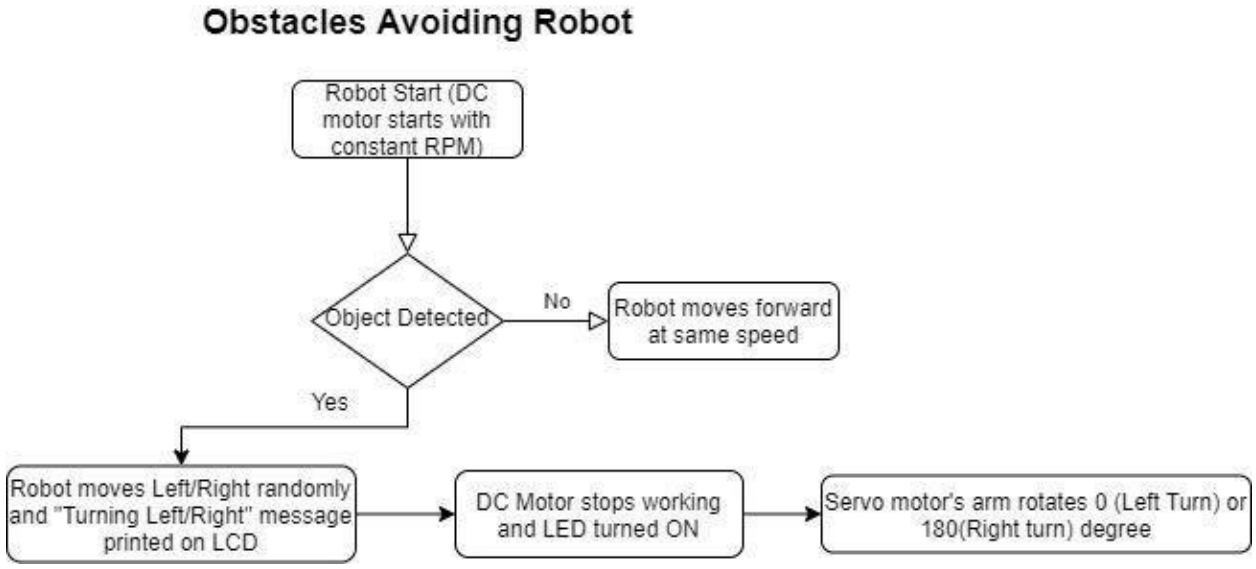

Figure 10 - Flow chart

Obstacle avoiding robot is a robot which moves forward until it does not detect any object and after detecting an object it moves randomly left or right until the object is not detected.

In implementation of this system, the most important variable is A. Range of A is 0 to 1 and 2. If the object is detected in the sensor then A will take the value between 0 and 1 randomly, we want the probability of turning left and right equally, so we have taken the values between 0 and 0.5 as turning left and the values between 0.5 and 1 as turning right. Otherwise the value of A will be 2.

If the value of A is 2 then the robot is going forward and the object has not been detected in an ultrasonic sensor. If the value of A is less than 0.5 then the robot is turning left because the object has been detected and if A's value is greater than 0.5 but less than 1 then the robot is turning right because the object has been detected.

The whole system works according to the value of A. If the value of A is 2 then the robot is moving forward because the object has not been detected so, DC motor will turn on, LED will be off, servo motor's arm will be 90 degrees and LCD display will show the RPM of the DC Motor.

If the value of A is between 0 and 0.5 then the robot is moving left because the object has been detected, so DC motor will turn off, LED will start glowing and servo motor's arm will be rotated to 0 degree to show the robot has turned left and LCD display show "Turning

Left" and "Alert".

If the value of A is between 0.5 and 1, then the robot will start moving to right because the object has been detected, so DC motor will turn off, LED will turn on and servo motor's arm will be rotated to 180 degree to show the robot has turned right and LCD display will show "Turning Right" and "Alert".

5. MATLAB functions' explanation

Obstacle avoiding robot has total 6 user defined matlab functions and their explanation has given below:

5.1. DC motor on/off

This function takes care of turning the DC motor on and off. It takes A data store's value and changes the value of the arduino digital pin 11 according to condition. If the value of A is 2, which means the robot is going forward and we have to keep the DC motor on. So, it changes the Arduino digital pin's value to 1 and if the value of A is other than 2, then we have to turn the DC motor off by changing the digital pin's value to 0.

5.2. LED on/off

This function takes A's value as input and turns the LED light on or off.

If the value of A is 2, then the value of the Arduino digital pin 2 will be 0, which means the LED is off and the robot is going forward.

If the value of A is other than 2, then the LED will be turned on and show that the robot is turning left or right randomly.

5.3. Object Detect and Change A

This function is the core of the project. It takes 4 inputs and they are A, B, uniform random number generator and the ultrasonic sensor's reading. It changes the value of A and B according to conditions and outputs them into A and B data store write.

If the object is detected in less than 10 cm, then this function takes a random number between 0 and 1 and stores it into data store memory A by checking the value of B.

If the value of B is 0, which means we have to change the value of A by taking the random number generated by the Uniform random number generator block and change the value of B to 1.

If the value of B is 1 and the object is detected in less than 10 cm then we will take the previously generated random number and change the value of B to 1.

If the object is not detected then we will have to change the value of A into 2, which shows that object is not detected and we will also change the value of B to 0, which shows that next if the object is detected then change the value of A.

5.4. Display Speed and Alert

This function displays the first line of the LCD display. It takes 1 input of the value of A and according to condition print the instructions.

If the value of A is 2, which means the robot is going forward and when it is going forward we will print "DC motor RPM" in first line of LCD and LCD takes only uint8 data type as input we convert the sentence to unit8 by writing uint8('DC motor RPM').

If the value of A is other than 2, which means the robot is turning left or right randomly and we will print “ALERT!!” in the first line of the LCD by typing unit8(‘ALERT!!’) to convert it into an unsigned integer.

5.5. Display RPM and left or right

This function displays the second line of the LCD display by taking the value of A as input. To print in the LCD we have to convert the sentences into uint8 format.

If the value of A is 2, which means the robot is going forward then we will print the RPM of the motor by typing uint8(‘300’).

If the value of A is less than 0.5, which means the robot is turning left, then we will print “Turning Left” by typing uint8(‘Turning Left’).

If the value of A is greater than 0.5 and less than 1 then we will print “Turning right” by typing uint8(‘Turning Right’).

5.6. Servo Motor Position

This function changes the position of the shaft of the servo motor. Servo motor can rotate between 0 and 180 degrees by writing into pin 7 of standard servo write.

If the value of A is 2, which means the robot is going forward, then the servo motor’s arm will be at 90 degree position, so we write 90 in output.

If the value of A is less than 0.5, which means the robot is turning left, we will write the value 0 in output and the servo motor’s arm will be at 0 degree.

If the value of A is greater than 0.5 and less than 1, which means the robot is turning left and in this condition we will change the value of the servo motor to 180 degree by writing value 180 in output.

6. Appendix

Here below is the code for the Obstacle avoiding robot with each function. Matlab 2018a version is used for making the system.

1. **DC Motor On/off function**

```
function y = fcn(A)
if A==2
   y =1;
else
   y =0;
end
```

2. **LED on/off function**

```
function y = fcn(A)
if A == 2
```

```
    y = 0;
else
    y= 1;
end
```

3. **Object detect and change A function**

```
function [data2, y]= fcn( B, A, random, distance)
if  distance < 0.10
    if B == 0
        y = random;
        data2 = 1;
    else
        y = A;
        data2 = 1;
    end
else
    y = 2;
    data2 = 0;
end
```

4. **Display speed and alert function**

```
function y = fcn(u)
if u == 2
    y =uint8('DC Motor RPM');
else
    y = uint8('ALERT!!     ');
end
```

5. **Display RPM, left and right function**

```
function y = fcn(u)
if u == 2
    y =uint8('300         ');
else
```

```
    if u< 0.5
    y = uint8('Turning Left ');
    else
    y= uint8('Turning Right');
    end
end
```

6. **Servo motor Position function**

```
function y = fcn(A)
if A == 2
    y = 90;
else
    if A < 0.5
        y = 0;
    else
        y = 180;
    end
End
```

7. Screenshot of Simulink model

Here in figure 11, shows the screenshot of the simulink model used in the project.

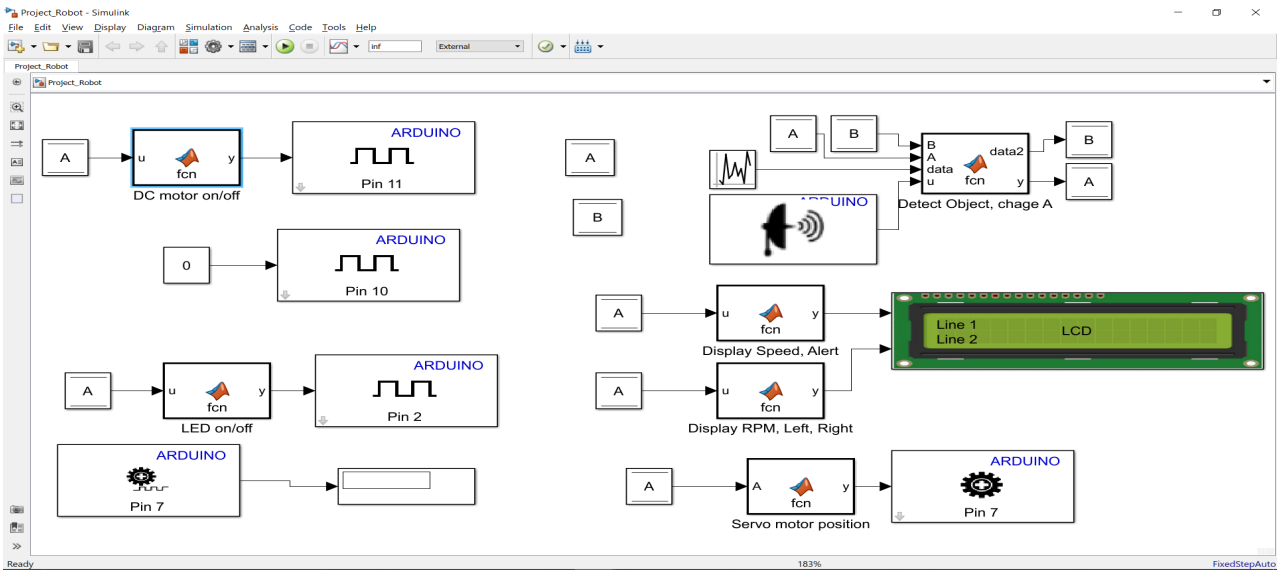

Figure 11 - Simulink SLX diagram

8. Circuit Diagram

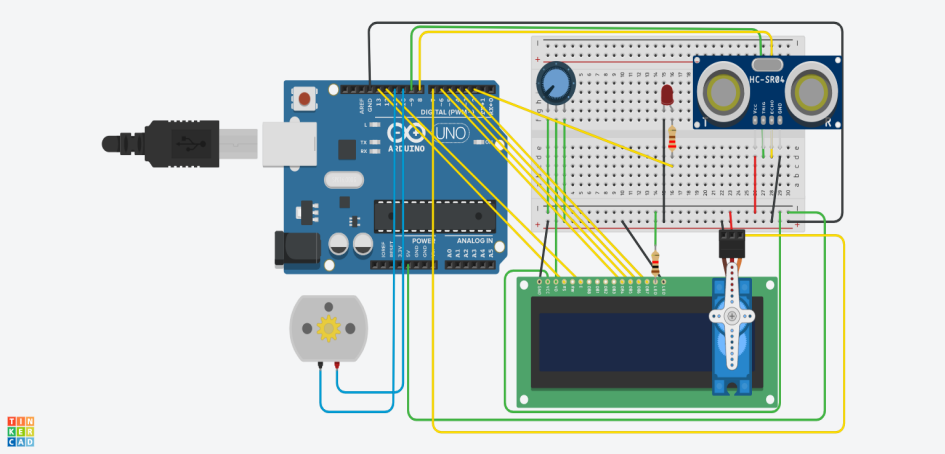

Figure 12 - Circuit Diagram

9. Benefits and Drawbacks of the project

Advantages of the Obstacle Avoiding Robot

- Whenever the robot detects the obstacle, it automatically reroutes its direction to left or right without the guidance of humans and hence, can be used as a movable surveillance system.
- It can also be operated as a spy, terrorist attack, bomb disposal.
- It can be managed remotely and is independent of manpower.
- The hardware components used in making this robot are very cheap.
- It can be used in a dark environment due to the ultrasonic sensor.

Disadvantages of the Obstacle Avoiding Robot

- The system requires precise programming for correct functioning.
- The system cannot be controlled by humans .
- Obstacles avoiding robots can be used for short-distance only.
- The system cannot be used in a vacuum.
- Soft materials can affect the sensing accuracy.
- Obstacles avoiding robots have a limited detection range.
- The system can consume more time to work efficiently.
- The robot is not able to distinguish the size of the object.

10. Outputs, Results and Testing Procedures

Case #	Given Input	Expected Output	Real Output	Test Pass/Fail
1	System started	The robot should start moving forward at a fixed RPM.	Robot working fine	Pass

2	Object detected	The robot should turn left or right randomly.	Robot turns left or right randomly	Pass
3	Object removed	Robot should go forward	Robot goes forward	Pass
4	Object detected	The LED should turn on, the servo arm should rotate 180(right) or 0(left) degrees, and the DC motor should stop.	LED turns on, servo arm rotates 180(right) or 0(left) degree.DC motor stops.	Pass
5	Object removed	LED should turn off, servo arm should rotate at 90 degrees, DC motor should start	LED turns off, servo arm rotates to 90 degrees, DC motor starts	Pass

Table - 1 Test cases