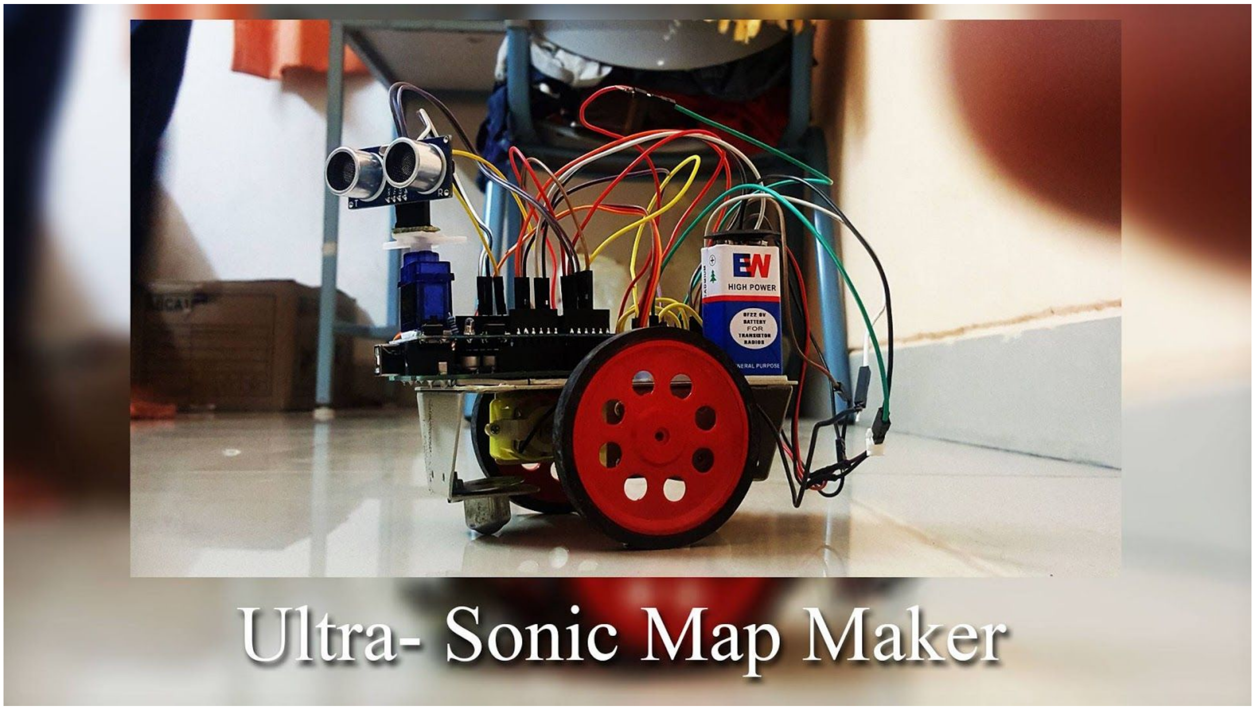
Ultrasonic Map Maker BOT using Arduino Uno



Akash Katragadda U101115FCS108 S4
Taniya Mangwani U101115FCS163 S7
Kshiteej Gilda U101115FCS110 S4
Harshit Sharma U101115FCS097 S4

Abstract

Problem Statement:
Can the principle of SONAR be used to map the surrounding environment?

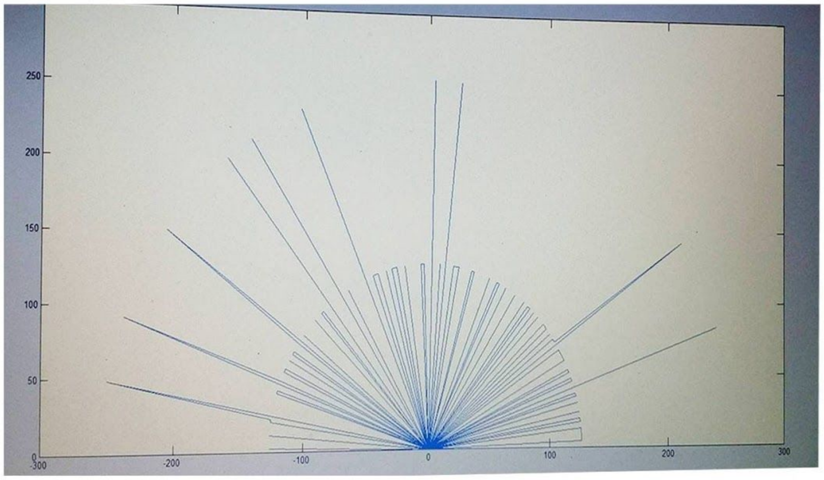
Introduction:
This project portrays the use of SONAR in mapping the surroundings with the help of an Ultrasonic sensor and Arduino Uno Board. Bats and Dolphins use SONAR to aid their vision and to move around. The same principle is used by cruise ships to navigate and get readings from deep inside the ocean. The same method of depth perception has been used over here to plot the 2-D graph of the surrounding environment. Active SONAR has been used to get the idea of the distance between the BOT and the surrounding environment.

Procedure Used:

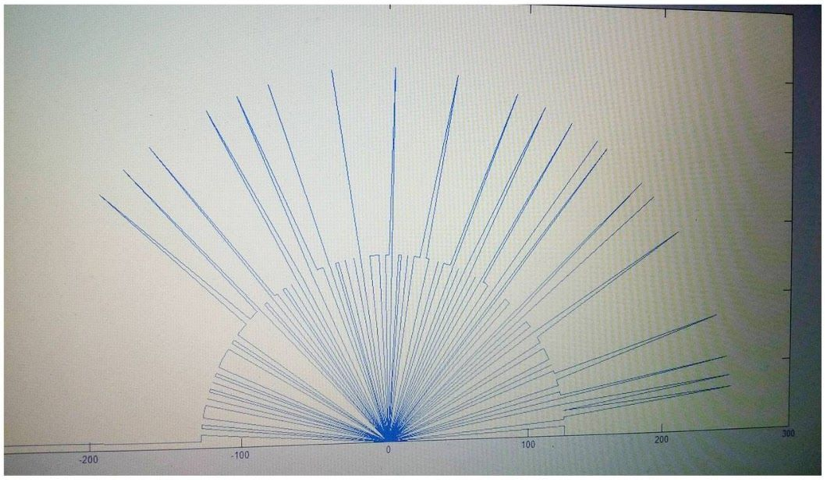
1. The BOT will place itself in a position relatively at the center of the surrounding environment. This will be accomplished by comparing the data from the sensor in x and y directions, and sending the bot in the direction where the ping time of the sensor is more.
2. The servo will now move from 0 degrees to 180 degrees. At every degree position the sensor will send sound pulses 10 times and averages this ping time to calculate the distance which will be sent to MATLAB through the serial port.
3. The MATLAB program will plot this incoming data through the port on a 2-D graph by calculating the x and y coordinates from the angle and distance.

Result Obtained:

The graph obtained is more accurate provided the surrounding environment is more closer to the sensor. SONAR has a drawback that the sound pulses will echo when they are aimed at corners and the data at uneven surfaces is ambiguous for plotting the graph. The echo feedback can be fixed by using better quality sensors. Better calibration between the sensor and BOT can be used to move the BOT at various positions. Data from this positions can be averaged to get a more accurate reading. For the applications of depth perception SONAR does a fine job and the BOT is able to navigate itself based on sensor inputs accurately.



Plot 2
(MatLab)



Plot 1
(MatLab)

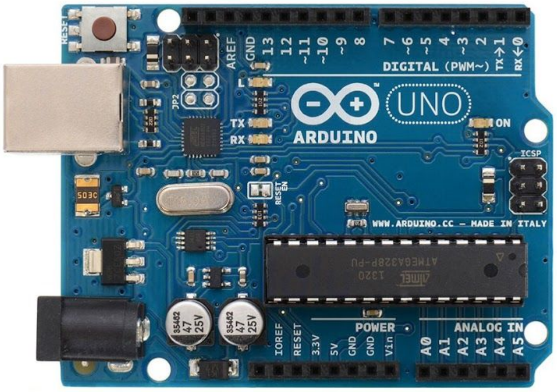
Objective

The objective of the project is to map the surrounding environment on a 2-d graph in Matlab through the use of SONAR. We are going to use Ultrasound waves emitted through an Ultrasonic sensor to measure the distance between the BOT and the surroundings. This sensor data taken over 180 degrees will be sent over to the MATLAB program through the serial port and plotted using the same.

Components Used

Microcontroller:

A microcontroller is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. The one we used is the standard ATMEGA328PU which comes with the Arduino UNO.



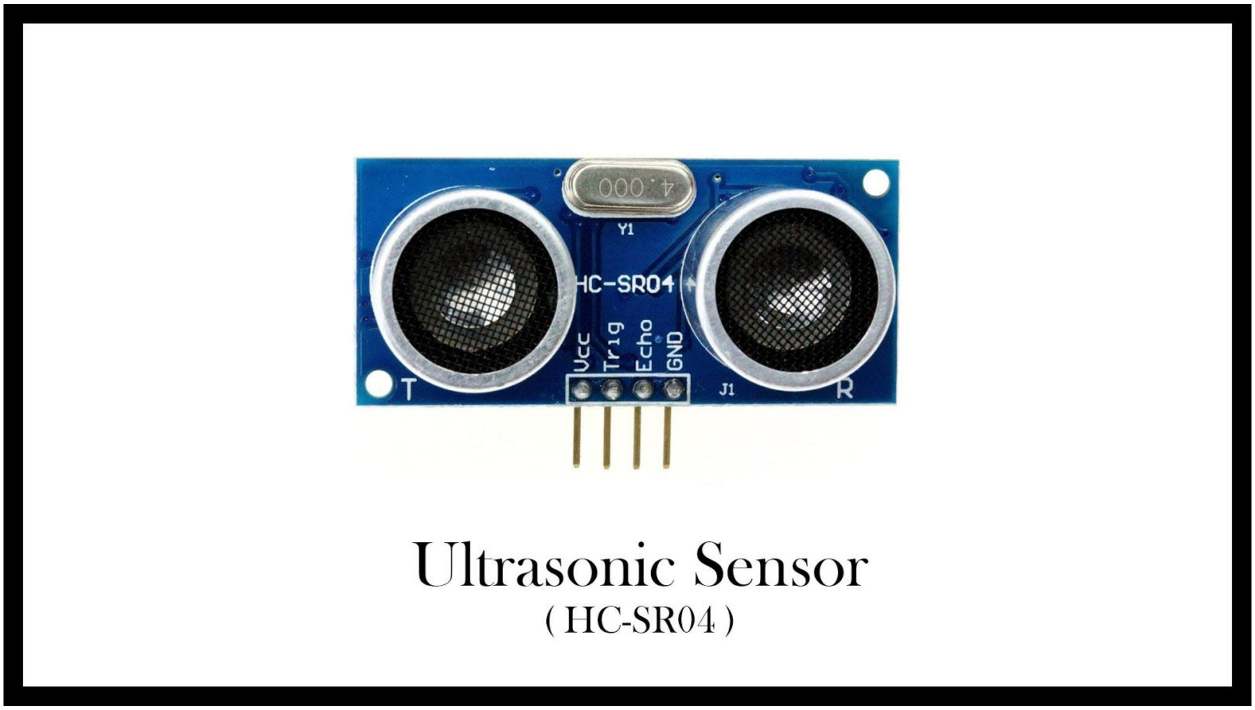
Arduino Board
(UNO)

Arduino is a single-board microcontroller designed to make the process of using electronics in multidisciplinary projects more accessible. The hardware consists of a simple open source hardware board designed around an 8-bit Atmel AVR microcontroller. An Arduino board consists of an Atmel 8-bit AVR microcontroller with complementary components to facilitate programming and incorporation into other circuits. An important aspect of the Arduino is the standard way that connectors are exposed, allowing the CPU board to be connected to a variety of interchangeable add-on modules known as shields.

The software consists of a standard programming language compiler and a boot loader that executes on the microcontroller.

Ultrasonic Sensor (Active Sensor) :

The sensor HC-SR04 is basically what guides the bot and takes all the readings. It emits ultrasonic waves which hits an obstacle and reflects back to the sensor after which it calculates the distance between the bot and the obstacle. There are four pins on the sensor.



VCC, TRIG, ECHO and GND. Each of these pins are connected to the arduino board via jumper wires. The VCC pin is connected to the 5V volt power supply, the TRIG and ECHO pins are connected to output and input pins respectively on the arduino board and the GND to one of the three ground connection pins on the board.

-> ***Advantages of Using the HC-SR04 Sensor***

- Low cost.
- Can effectively provide distance to objects.
- Produces tractable amount of data for interpretation.

-> ***Disadvantages of Using the HC-SR04 Sensor***

- Poorer discriminatory ability than vision
- Susceptible to noise/distortion
- Can produce erroneous data (reflections)

Servo Motor :

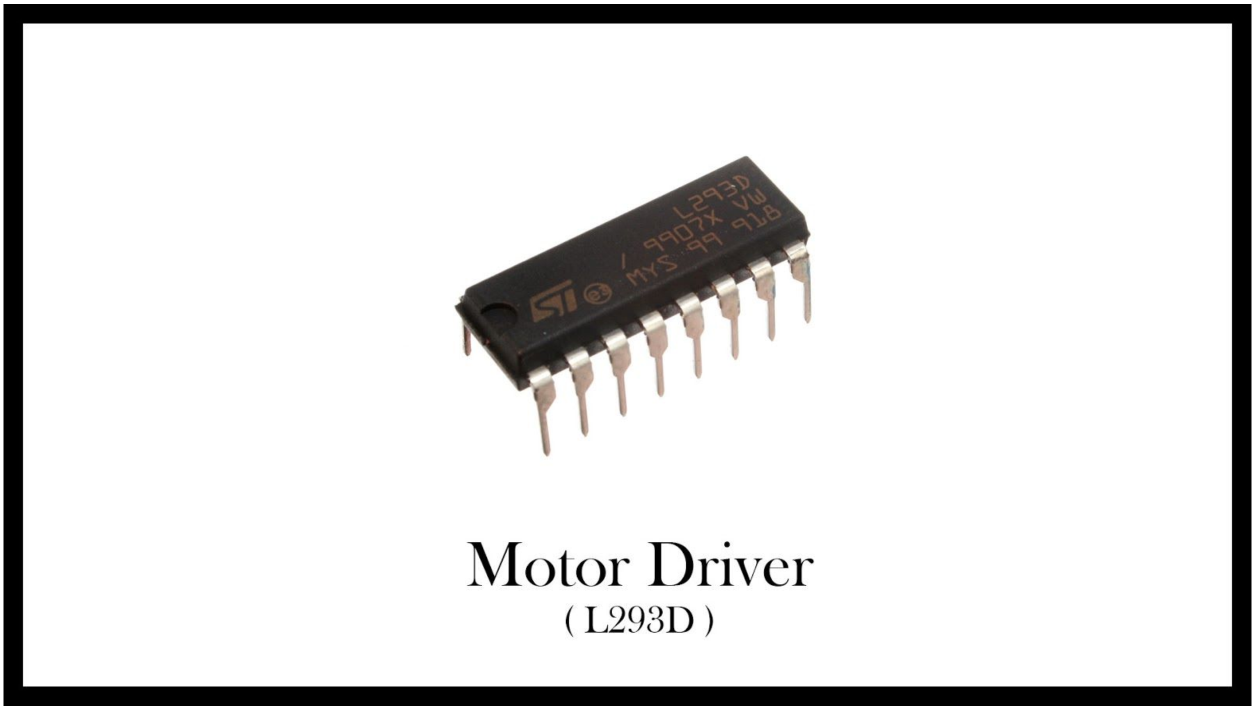
The ultrasonic sensor is placed on the servo so that it can take a 180-degree reading of its surroundings. The servo has three wires coming out of it, of the three one is for power, one for grounding and another for input.

However there are a couple of downsides of using the SG90 tower pro servo motor they are:

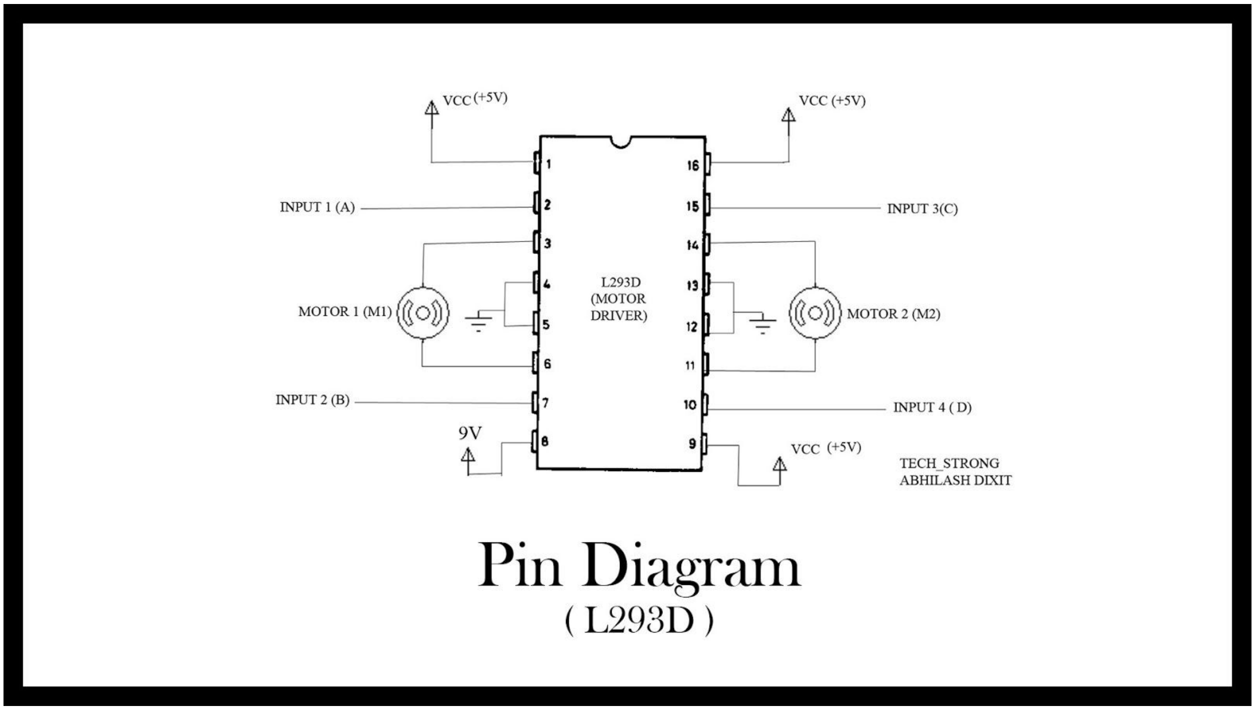
- The motor cannot take readings of surroundings of more than 180.
- The motor only moves horizontally and not vertically.

Motor Driver :

L293D is a typical Motor driver or Motor Driver IC which allows DC motor to drive on either direction. L293D is a 16-pin IC which can control a set of two DC motors simultaneously in any direction. It means that you can control two DC motor with a single L293D IC.



It works on the concept of H-bridge. H-bridge is a circuit which allows the voltage to be flown in either direction. In a single L293D chip there are two h-Bridge circuit inside the IC which can rotate two dc motor independently. Due its size it is very much used in robotic application for controlling DC motors.



There are 4 input pins for l293d, pin 2,7 on the left and pin 15 ,10 on the right as shown on the pin diagram. Left input pins will regulate the rotation of motor connected across left side and right input for motor on the right hand side. The motors are rotated on the basis of the inputs provided across the input pins as LOGIC 0 or LOGIC 1.

- Pin 2 = Logic 1 and Pin 7 = Logic 0 | Clockwise Direction
- Pin 2 = Logic 0 and Pin 7 = Logic 1 | Anticlockwise Direction
- Pin 2 = Logic 0 and Pin 7 = Logic 0 | No rotation, Hi-Impedance state
- Pin 2 = Logic 1 and Pin 7 = Logic 1 | No rotation

Programming Platforms:

We have used the default arduino IDE to code Arduino and to display data which is taken by our ultrasonic map maker we have used MATLAB software.

Arduino :

The Arduino code controls the motion of the servo motor and dual shaft DC motors of bot to control bot’s motion , and when the readings from the ultrasonic sensor are captured and how frequently. It also pushes the sensor data to the serial port.

```
#include <Servo.h>
#include <NewPing.h>
#define rightWheeltwo 3
#define leftWheeltwo 6
#define rightWheelseven 4
#define leftWheelseven 7
#define TRIGGER_PIN  12
#define ECHO_PIN     11
#define MAX_DISTANCE 200

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
Servo myservo;
int pos = 0;
int it = 10;

void setup()
{
        myservo.attach(9);
        Serial.begin(9600);
        pinMode(leftWheeltwo, OUTPUT);
        pinMode(rightWheeltwo, OUTPUT);
        pinMode(leftWheelseven, OUTPUT);
        pinMode(rightWheelseven, OUTPUT);
        delay(3000);
}

void loop()
{

        int i = 0;
        int t = 0;
        int a = 0;
        int z = 0;
        int m = 0;
        for (i = 0; i < 4; i++)
```

```
  {
          myservo.write(0);
          delay(1000);
          unsigned int back = sonar.ping();
          myservo.write(180);
          delay(1000);
          unsigned int front = sonar.ping();
          if (front > back)
  {
  digitalWrite(leftWheeltwo, HIGH);
  digitalWrite(leftWheelseven, LOW);
  digitalWrite(rightWheeltwo, HIGH);
  digitalWrite(rightWheelseven, LOW);
  delay(1000);
  }
          else
  {
        digitalWrite(leftWheeltwo, LOW);
     digitalWrite(leftWheelseven, HIGH);
          digitalWrite(rightWheeltwo, LOW);
     digitalWrite(rightWheelseven, HIGH);
  delay(1000);
  }
digitalWrite(leftWheeltwo, LOW);
digitalWrite(leftWheelseven, LOW);
digitalWrite(rightWheeltwo, LOW);
digitalWrite(rightWheelseven, LOW);

          myservo.write(90);
          delay(1000);
          unsigned int right = sonar.ping();
          myservo.write(180);
          delay(1000);
          {
          digitalWrite(rightWheeltwo, HIGH);
  digitalWrite(rightWheelseven, LOW);
  delay(1000);
           }
           unsigned int left = sonar.ping();
           if (right > left)
  {
  digitalWrite(leftWheeltwo, HIGH);
  digitalWrite(leftWheelseven, LOW);
           delay(1000);

  digitalWrite(leftWheeltwo, HIGH);
  digitalWrite(leftWheelseven, LOW);
  digitalWrite(rightWheeltwo, HIGH);
  digitalWrite(rightWheelseven, LOW);
  delay(500);

  digitalWrite(rightWheeltwo, HIGH);
           digitalWrite(rightWheelseven, LOW);
  delay(600);
  }
           else
  {
         digitalWrite(leftWheeltwo, HIGH);
     digitalWrite(leftWheelseven, LOW);
  digitalWrite(rightWheeltwo, HIGH);
  digitalWrite(rightWheelseven, LOW);
  delay(500);

  digitalWrite(leftWheeltwo, HIGH);
  digitalWrite(leftWheelseven, LOW);
  delay(600);
           }
digitalWrite(leftWheeltwo, LOW);
digitalWrite(leftWheelseven, LOW);
digitalWrite(rightWheeltwo, LOW);
digitalWrite(rightWheelseven, LOW);
  }
```

```
        for (i = 0; i < 180; i ++)
        {
        unsigned int uS = sonar.ping();
        myservo.write(i);
        delay(20);
        for (t = 0; t < it; t++)
        {
                uS = sonar.ping();
                a = uS/US_ROUNDTRIP_CM + a;
                delay(30);
        }
                a = a / (it-1);
        t = 0;

    Serial.println(a);
    a = 0;
  }
}
```

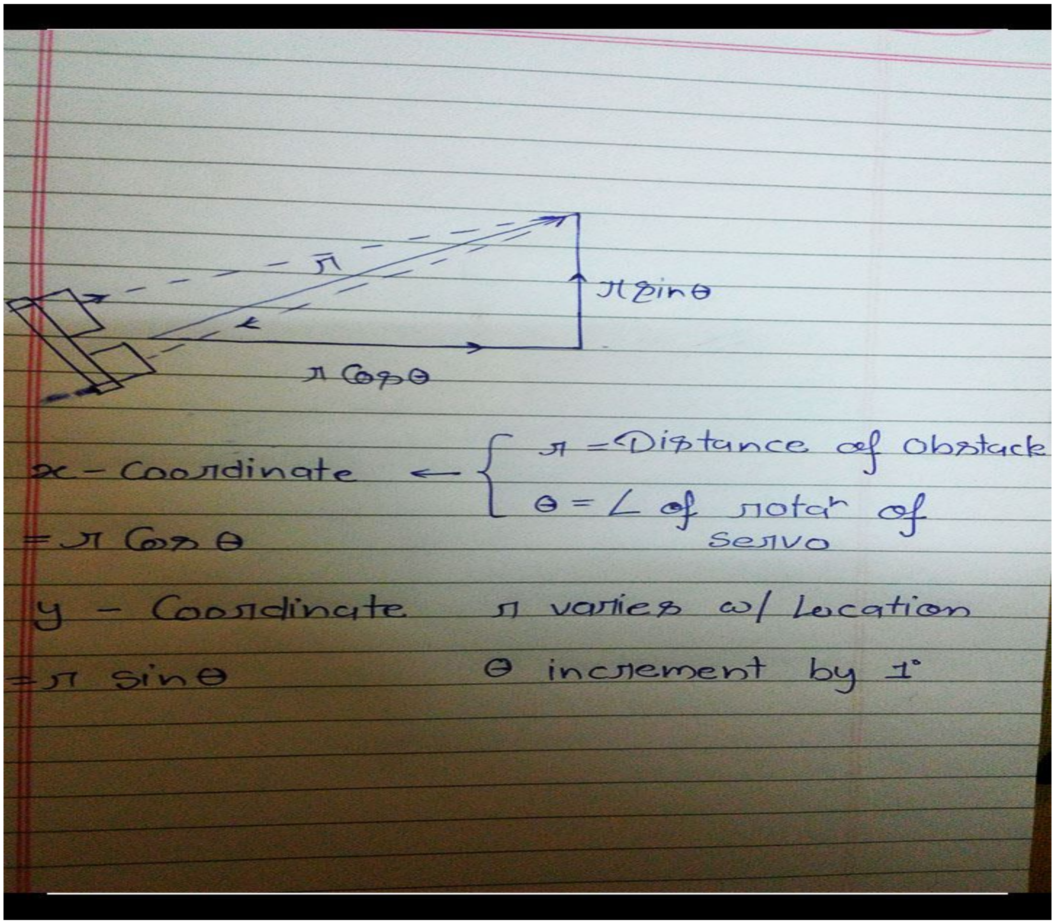
MATLAB :

The MatLab code deals more with data than the actual control of the board, so all the sensor data is pushed over serial to the PC, where it is read by MatLab.

```
theta = 0:(pi/180):pi;
s = serial('COM10');
s.BaudRate=9600
fopen(s)
i = 0;

inc = 1;

while i<180
   A = fgets(s);
   num(i+1) = str2num(A);
   i = i+1;
end
fclose(s)

j = 1
```

```
while j<181
    tab(j,1) = (j-1)*inc
    tab(j,2) = num(j)
    tab(j,3) = num(j)*cosd((j-1)*inc)
    tab(j,4) = num(j)*sind((j-1)*inc)
    j = j+1
end
%figure
%polar(theta,num)
plot(tab(:,3),tab(:,4))
```

Now, the data that we receive from the Arduino tells us two things. The degree of rotation of the servo and the distance of an obstacle in that direction. Hence, the data that we have at this point is in the Polar coordinate system. For it to make sense to human eyes when visualized, it must be converted to the Cartesian or X-Y coordinate system.



So the MatLab code does just this. It gets data serially from the COM port, saves it into a matrix with the angle of rotation, and then converts it into Cartesian coordinates with the formula given above.

Once it's done, it gives an output by plotting the points on a graph. I placed the board in the box, and I got the following result.

Applications

1)Sonar technology can be used for defence purposes like surveillance of the environment . It can also be used to detect and track enemy vessels. It can also be used underwater for detection of enemy submarines and missiles like torpedos.
2) It can be used to determine the depth of ocean floors underneath the ship which has made navigation safer.
3)Pipeline inspections can now be performed with high frequency scan sonar which can be used to detect cracks to detect leaks of gas and oil.
4)It can be used for surveillance purposes both on ground and underwater with which exploration missions can take place with ease.
5)Detection of irregular parts in machines.
6) It can be used for medical purposes like detection of fetuses
7) While travelling through a bulk of materials, an ultrasonic wave is reflected at an interface between different materials. This nature can be used to detect the presence of flaws in concrete or metals.
8) When an ultrasonic wave is transmitted it is reflected at an interface between different materials and an echo is returned. Measuring the time interval between sending the sound wave and receiving the echo will determine the distance to the target. Ultrasonic distance measurement can be applied to any medium through which sound can travel such as air, liquids and metals.
9) Ultrasonic sound waves are also used for the purpose of welding plastic. High frequency ultrasonic vibrations are used to weld couple of parts of the plastic.

Relevance

1) The ultrasonic sensor can be used for mapping of the environment and detect objects.
2) The sensor can be used for vehicles as parking sensors which can be used for detecting walls and other vehicles.

3) The sensor can also be used as a radar of the environment to detect movements and can also be placed in safe places of the pipelines to detect any leaks in the pipeline
4) Ships can use more accurate and bigger sensors to gain vision deep inside the ocean. Enemy Submarines can also be detected using this sensor.

Feasibility

The real life applications of the project can be measurement of distant objects using the ultrasonic sensor. The accuracy of the sensor can be improved such that it can provide better range of transmission than the sensor used in the project. Its applications can include the usage of this bot to map a particular environment and to find out the locations of the obstacles or the objects that are present on the transmission way.

This can also be used to find out the map of the environment in which the bot is working.

The censors and the motors used in the project can be replaced with better quality and range so that the bot is able to map not only the environment but can provide the maps of far-off objects also which can be used by the Defence Department.

References

1. *https://www.hackster.io/Satyavrat/ultrasonic-map-maker-using-an-arduino-yun-37c72e?ref=channel* – This site is used to understand the basic knowledge about the working, assembly, and the components that are used in the project.
2. *https://www.arduino.cc/en/Main/ArduinoBoardUno* - This site is used to understand the working of arduino and its specifications.
3. *https://www.arduino.cc/en/Main/ArduinoProtoShield* - This site is used to understand the ArduinoProtoShield and its specifications.
4. http://www.rakeshmondal.info/L293D-Motor-Driver - This site is used for understanding the pin diagram , specifications , and the usage of the IC.