**SELF STUDY REPORT**

ON

**“SELF SURVEILLANCE ROBOT USING**

**ULTRASONIC SENSORS”**

Submitted in fulfilment of the requirements for the completion of

**SELF STUDY FOR MICROCONTROLLER COURSE (15ES4GCMCS)**

IN

**ELECTRONICS AND COMMUNICATION ENGINEERING**

SUBMITTED BY:

**Students name Likith S USN 1BM16EC052**

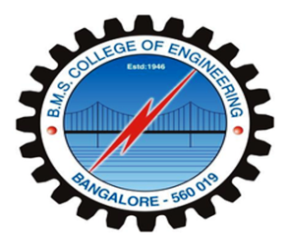
**Karthik Raj S S 1BM16EC043**

**Gagan L Naik 1BM16EC034**

**Joswin Preetham Dsouza 1BM16EC040**

Name of the course instructor: SUMA M N

**Jan-May 2018**

****

Department of Electronics and Communication Engineering

**B.M.S COLLEGE OF ENGINEERING**

(Autonomous College Affiliated to Visvesvaraya Technological University, Belgaum)

Bull Temple Road, Basavanagudi, Bangalore-560019

**CONTENTS**

**Chapter-1**

**1.1** Aim of the project…………………………………………………….

**1.2** Area of application……………………………………………………

**1.3** Introduction……………………………………………………………

**Chapter-2**

**2.1** Literature Survey………………………………………………………

**Chapter-3**

**3.1** Problem Statement……………………………………………………. **3.2** Proposed Solution...……………………………………………………

**Chapter-4**

**4.1** Project Execution Steps……………………………………………….. **4.2** Estimated Bill of Materials……………………………………………..

**Chapter-5**

**5.1** Future Enhancements…………………………………………………

**5.2** Conclusion……………………………………………………………

**Chapter 1**

**1.1 AIM**

This innovative project is an effective self surveillance robot using ultrasonic sensors, here arduino is used to implement the project.

The aim of the project is to create a system that moves around the path specified by the designer and detect objects in its path and give the information back to the user. It uses a line follower robot along with ultrasonic sensor which detects the objects in the path. All this is done without intervention of humans which makes it a unique mini project.

**1.2 AREA OF APPLICATION**

* Industrial automated equipment carriers.
* Automated cars.
* For security purpose.
* For military application.
* Tour guides in museums and other similar applications.
* Space exploration .(example: Curiosity rover (mars))

**1.3 INTRODUCTION**

The line follower is a self operating robot that detects and follows a line that is drawn on the floor. The path consists of a black line on a white surface (*or it may be* *reverse of that*). The control system used must sense a line and maneuver the robot to stay on course, while constantly correcting the wrong moves using feedback mechanism, thus forming a simple yet effective closed loop System. The robot is designed to follow very tight curves.

Radar is an object detection system that uses electromagnetic waves to identify range, altitude, direction, or speed of both moving and fixed objects such as aircraft, ships, vehicles, weather formations, and terrain. When we use ultrasonic waves instead of electromagnetic waves, we call it ultrasonic radar

**Chapter 2**

**2.1 LITERATURE SURVEY**

In recent years a great deal of time and effort has been spent of developing systems to enable an autonomous robot to follow a marked path using a vision system .Not surprisingly, the majority of this research has been towards modifying, or designing from scratch, a full sized road vehicle so that it can drive on ordinary roads without human supervision. Due to the large amount of space available in an ordinary road vehicle, high performance computers can be used to perform complex image processing and typically to maintain a mathematical model of the vehicle and the environment.

Autonomous robots are independent of any controller and can act on their own. The robot is programmed to respond in a particular way to an outside stimulus. The bump and go robot is good example. This robot uses bumper sensors to detect obstacle. When the robot is turned on, it moves in a straight direction and when it hits an obstacle, the crash triggers the bumper sensors. The robot gives programming instructions that asks the robot to back up, turns to right direction moves forward. This is its response to every bump. In this way the robot can change direction every time, it encounters an obstacle.

RADAR is an object detection system which uses radio waves to determine the range, altitude, direction, or speed of objects. Radar systems come in a variety of sizes and have different performance specifications. Some radar systems are used for air-traffic control at airports and others are used for long range surveillance and early-warning systems. A radar system is the heart of a missile guidance system. Small portable radar systems that can be maintained and operated by one person are available as well as systems that occupy several large rooms .

Radar was secretly developed by several nations before and during the World War II. The term RADAR itself, not the actual development, was coined in 1940 by United States Navy as an acronym for Radio Detection and ranging. The modern uses of radar are highly diverse, including air traffic control, radar, astronomy, air-defense systems, antimissile systems, antimissile systems; marine radars to locate landmarks and other ships; aircraft anti-collision systems; ocean surveillance systems, outer space surveillance and rendezvous systems; meteorological precipitation monitoring; altimetry and flight control precipitation monitoring; altimetry and flight control systems; guided missile target locating systems; and ground-penetrating radar for geological observations. High tech radar systems are associated with digital signal processing.

**Chapter 3**

**3.1 PROBLEM STATEMENT**

Many of the military bases and highly secure systems require continuous manning. Hence high security is often employed at these places and is manned 24/7.This requires a lot of man power and the cost involved in maintaining security is extremely high. Also manual manning is not as efficient. Human error can lead to extreme fatal issues. Even though the security personal are provided with sophisticated weapons and devices to fight against intruders, it often turns out to be a slow response when compared to automated ones. Even the accuracy is not as good as that of automated devices. Loss of lives is a major issue, instead using these devices can save precious lives.

The number of employees needed to employed is high, the cost involved in maintaining these employees for manning steeply increases. This is a serious issue as the large part of funds needs to be directed only for security reasons.

**3.2 PROPOSED SOLUTION**

We can use an automated robot with appropriate sensors for manning these systems. A specific path surrounding the required system is chosen. A simple line follower is designed to follow the specified path. An ultrasonic sensor is interfaced with the micro controller. The ultrasound sensor is fixed on a servo motor so that the sensor rotates along the axis of the motor. Thus the robot maneuvering along the specific path can detect objects along the path so that appropriate action can be initiated.

Principle of sonar is used to calculate the distance from the object. The ultra sound sensor keeps transmitting waves from its transmitter. When some of the waves hits an object, the waves are reflected back and is received by the receiver. Now the time taken to transmit the and receive the wave is calculated. The distance between the object and the robot is calculated by using the simple formula:

Distance= (speed of the wave \* time)/2

Also the angle at which the object is located wrt the robot can be calculated. With such ample information we can decide the course of action that needs to be initiated when such objects are located by the sonar. Also the output of these operations can be displayed on a pc.

* The circuit is designed in such a way that two green leds and two red leds glow during the time interval of 10 seconds.
* There is a change of state during 9 sec, hence the yellow led’s glows for a second before switching to red or green.

**CHAPTER 4**

**4.1 PROJECT EXECUTION STEPS**

Ultra sound radar system

The Arduino integrated development environment (IDE) is a [cross-platform](http://en.wikipedia.org/wiki/Cross-platform) application written in [Java,](http://en.wikipedia.org/wiki/Java_(programming_language)) and is derived from the IDE for the [Processing programming language](http://en.wikipedia.org/wiki/Processing_(programming_language)) and the [Wiring](http://en.wikipedia.org/wiki/Wiring_(development_platform)) projects. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as [syntax highlighting,](http://en.wikipedia.org/wiki/Syntax_highlighting)[brace matching,](http://en.wikipedia.org/wiki/Brace_matching) and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click. A program or code written for Arduino is called a "sketch".

Arduino programs are written in [C](http://en.wikipedia.org/wiki/C_(programming_language)) or [C++.](http://en.wikipedia.org/wiki/C%2B%2B) The Arduino IDE comes with a [software library](http://en.wikipedia.org/wiki/Software_library) called "Wiring" from the original Wiring project, which makes many common input/output operations much easier. Users only need define two functions to make a run able [cyclic executive](http://en.wikipedia.org/wiki/Cyclic_executive) program:

* Setup (): a function run once at the start of a program that can initialize settings
* Loop (): a function called repeatedly until the board powers off. Open the Arduino IDE software and select the board in use. To select the board:
* Go to Tools.
* Select Board.
* Under board, select the board being used, in this case Arduino Uno.
* Go to Tools and to Port and select the port at which the Arduino board is connected.
* We write the code in the space provided and click on compile. Once the code is compiled, click on upload to upload the sketch to the Arduino board.

Hardware connections:

A servo motor is connected to a digital pin of Arduino Uno to which an ultrasound sensor is attached to its rotor. This motor helps the sensor to rotate about 180 degrees along its axis. The following code is uploaded on the board

Code for Sonar:

// Includes the Servo library

#include <Servo.h>.

// Defines Tirg and Echo pins of the Ultrasonic Sensor

constinttrigPin = 10;

constintechoPin = 11;

// Variables for the duration and the distance

long duration;

int distance;

Servo myServo; // Creates a servo object for controlling the servo motor

void setup() {

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

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

Serial.begin(9600);

myServo.attach(12); // Defines on which pin is the servo motor attached

}

void loop() {

// rotates the servo motor from 15 to 165 degrees

for(inti=15;i<=165;i++){

myServo.write(i);

delay(30);

distance = calculateDistance();// Calls a function for calculating the distance measured by the Ultrasonic sensor for each degree

Serial.print(i); // Sends the current degree into the Serial Port

Serial.print(","); // Sends addition character right next to the previous value needed later in the Processing IDE for indexing

Serial.print(distance); // Sends the distance value into the Serial Port

Serial.print("."); // Sends addition character right next to the previous value needed later in the Processing IDE for indexing

}

// Repeats the previous lines from 165 to 15 degrees

for(inti=165;i>15;i--){

myServo.write(i);

delay(30);

distance = calculateDistance();

Serial.print(i);

Serial.print(",");

Serial.print(distance);

Serial.print(".");

}

}

// Function for calculating the distance measured by the Ultrasonic sensor

intcalculateDistance(){

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

// Sets the trigPin on HIGH state for 10 micro seconds

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH); // Reads the echoPin, returns the sound wave travel time in microseconds

distance= duration\*0.034/2;

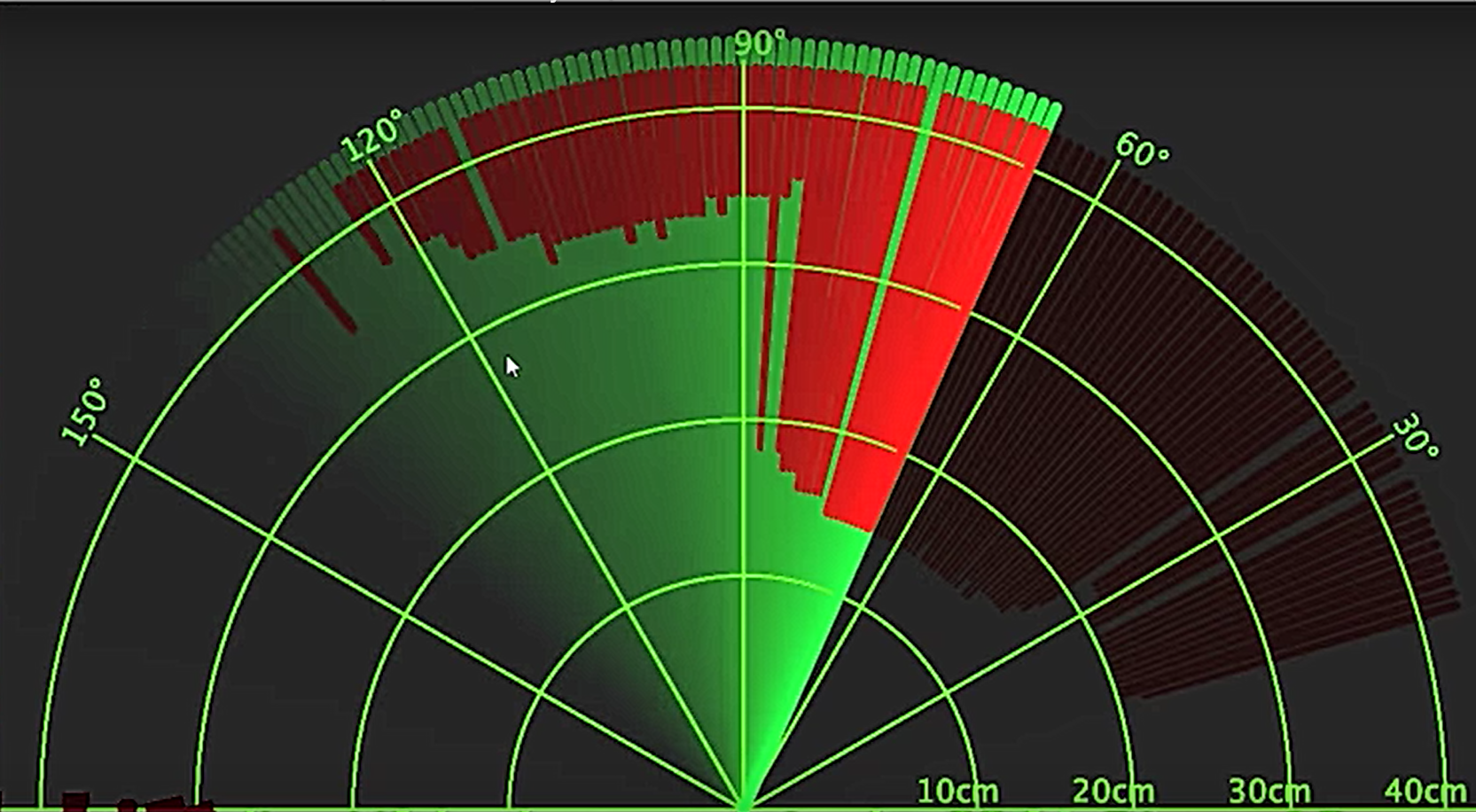
return distance;

}

Processing 3 software:

This software is used to display the output of the ultrasound sensor on a PC. We use Java language to program the output of the given system. The red lines indicate the presence

of an obstacle and the green line represents the free space. The distance and the angle at which the object is present is also shown.



Code for Processing 3

import processing.serial.\*; // imports library for serial communication

import java.awt. event. Key Event; // imports library for reading the data from the serial port

import java.io. IOException;

Serial myPort; // defines Object Serial

// defubes variables

String angle="";

String distance="";

String data="";

String noObject;

float pixsDistance;

intiAngle, iDistance;

int index1=0;

int index2=0;

PFontorcFont;

void setup() {

size (1200, 700); // \*\*\*SCREEN RESOLUTION\*\*\*

smooth();

myPort = new Serial(this,"COM5", 9600); // starts the serial communication

myPort.bufferUntil('.'); // reads the data from the serial port up to the character '.'. So actually it reads this: angle,distance.

}

void draw() {

fill(98,245,31);

// simulating motion blur and slow fade of the moving line

noStroke();

fill(0,4);

rect(0, 0, width, height-height\*0.065);

fill(98,245,31); // green color

// calls the functions for drawing the radar

drawRadar();

drawLine();

drawObject();

drawText();

}

void serialEvent (Serial myPort) { // starts reading data from the Serial Port

// reads the data from the Serial Port up to the character '.' and puts it into the String variable "data".

data = myPort.readStringUntil('.');

data = data.substring(0,data.length()-1);

index1 = data.indexOf(","); // find the character ',' and puts it into the variable "index1"

angle= data.substring(0, index1); // read the data from position "0" to position of the variable index1 or thats the value of the angle the Arduino Board sent into the Serial Port

distance= data.substring(index1+1, data.length()); // read the data from position "index1" to the end of the data prthats the value of the distance

// converts the String variables into Integer

iAngle = int(angle);

iDistance = int(distance);

}

void drawRadar() {

pushMatrix();

translate(width/2,height-height\*0.074); // moves the starting coordinats to new location

noFill();

strokeWeight(2);

stroke(98,245,31);

// draws the arc lines

arc(0,0,(width-width\*0.0625),(width-width\*0.0625),PI,TWO\_PI);

arc(0,0,(width-width\*0.27),(width-width\*0.27),PI,TWO\_PI);

arc(0,0,(width-width\*0.479),(width-width\*0.479),PI,TWO\_PI);

arc(0,0,(width-width\*0.687),(width-width\*0.687),PI,TWO\_PI);

// draws the angle lines

line(-width/2,0,width/2,0);

line(0,0,(-width/2)\*cos(radians(30)),(-width/2)\*sin(radians(30)));

line(0,0,(-width/2)\*cos(radians(60)),(-width/2)\*sin(radians(60)));

line(0,0,(-width/2)\*cos(radians(90)),(-width/2)\*sin(radians(90)));

line(0,0,(-width/2)\*cos(radians(120)),(-width/2)\*sin(radians(120)));

line(0,0,(-width/2)\*cos(radians(150)),(-width/2)\*sin(radians(150)));

line((-width/2)\*cos(radians(30)),0,width/2,0);

popMatrix();

}

void drawObject() {

pushMatrix();

translate(width/2,height-height\*0.074); // moves the starting coordinats to new location

strokeWeight(9);

stroke(255,10,10); // red color

pixsDistance = iDistance\*((height-height\*0.1666)\*0.025); // covers the distance from the sensor from cm to pixels

// limiting the range to 40 cms

if(iDistance<40){

// draws the object according to the angle and the distance

line(pixsDistance\*cos(radians(iAngle)),-pixsDistance\*sin(radians(iAngle)),(width-width\*0.505)\*cos(radians(iAngle)),-(width-width\*0.505)\*sin(radians(iAngle)));

}

popMatrix();

}

void drawLine() {

pushMatrix();

strokeWeight(9);

stroke(30,250,60);

translate(width/2,height-height\*0.074); // moves the starting coordinats to new location

line(0,0,(height-height\*0.12)\*cos(radians(iAngle)),-(height-height\*0.12)\*sin(radians(iAngle))); // draws the line according to the angle

popMatrix();

}

void drawText() { // draws the texts on the screen

pushMatrix();

if(iDistance>40) {

noObject = "Out of Range";

}

else {

noObject = "In Range";

}

fill(0,0,0);

noStroke();

rect(0, height-height\*0.0648, width, height);

fill(98,245,31);

textSize(25);

text("10cm",width-width\*0.3854,height-height\*0.0833);

text("20cm",width-width\*0.281,height-height\*0.0833);

text("30cm",width-width\*0.177,height-height\*0.0833);

text("40cm",width-width\*0.0729,height-height\*0.0833);

textSize(40);

text("Radar ", width-width\*0.875, height-height\*0.0277);

text("Angle: " + iAngle +" °", width-width\*0.48, height-height\*0.0277);

text("Distance: ", width-width\*0.26, height-height\*0.0277);

if(iDistance<40) {

text(" " + iDistance +" cm", width-width\*0.225, height-height\*0.0277);

}

textSize(25);

fill(98,245,60);

translate((width-width\*0.4994)+width/2\*cos(radians(30)),(height-height\*0.0907)-width/2\*sin(radians(30)));

rotate(-radians(-60));

text("30°",0,0);

resetMatrix();

translate((width-width\*0.503)+width/2\*cos(radians(60)),(height-height\*0.0888)-width/2\*sin(radians(60)));

rotate(-radians(-30));

text("60°",0,0);

resetMatrix();

translate((width-width\*0.507)+width/2\*cos(radians(90)),(height-height\*0.0833)-width/2\*sin(radians(90)));

rotate(radians(0));

text("90°",0,0);

resetMatrix();

translate(width-width\*0.513+width/2\*cos(radians(120)),(height-height\*0.07129)-width/2\*sin(radians(120)));

rotate(radians(-30));

text("120°",0,0);

resetMatrix();

translate((width-width\*0.5104)+width/2\*cos(radians(150)),(height-height\*0.0574)-width/2\*sin(radians(150)));

rotate(radians(-60));

text("150°",0,0);

popMatrix();

}

Line Follower

Line follower is an automated robot that follows the specified path or trajectory using IR sensing.

There are four conditions in this line following robot that we read by using Arduino. We have used two sensors namely left sensor and right sensor.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Input** | | **Output** | | | | **Movement**  **Of Robot** |
| **Left Sensor** | **Right Sensor** | **Left Motor** | | **Right Motor** | |
| LS | RS | LM1 | LM2 | RM1 | RM2 |  |
| Low | Low | High | Low | High | Low | Forward |
| Low | High | Low | High | High | Low | Turn Left |
| High | Low | High | Low | Low | High | Turn Right |
| High | High | Low | Low | Low | Low | Stop |

Hardware Connections

An IR Sensor array is used with 2 activated IR sensors. A motor driver L293D is used to interface the dc motors. The motor is interfaced to digital pins of Arduino. The motor rotates according to the table specified above. The robot moves in the designed trajectory.

Line Follower Code

intrs=11;

int ls=9;

int lm1=4;

int lm2=5;

int rm1=6;

int rm2=7;

int l1;

int r1;

char cond='n';

void setup()

{

Serial.begin(9600);

pinMode(ls,INPUT);

pinMode(rs,INPUT);

pinMode(lm1,OUTPUT);

pinMode(lm2,OUTPUT);

pinMode(rm1,OUTPUT);

pinMode(rm2,OUTPUT);

}

void loop()

{

readS();

Serial.println(cond);

motion(cond);

}

void readS()

{

l1=digitalRead(ls);

r1=digitalRead(rs);

if(l1!=1&&r1!=1)

cond='s';

else if(l1==1&&r1!=1)

cond='r';

else if(l1!=1&&r1==1)

cond='l';

else if(l1==1&&r1==1)

cond='n';

}

void motion(char cond)

{

switch(cond)

{

case 's':

{

forward();

}

break;

case'r':

{

right();

}

break;

case'l':

{

left();

}

break;

case'n':

{

stopie();

}

break;

}

}

void forward()

{

digitalWrite(lm1,1);

digitalWrite(lm2,0);

digitalWrite(rm1,1);

digitalWrite(rm2,0);

}

void left()

{

digitalWrite(lm1,0);

digitalWrite(lm2,1);

digitalWrite(rm1,HIGH);

digitalWrite(rm2,LOW);

}

void right()

{

digitalWrite(lm1,1);

digitalWrite(lm2,0);

digitalWrite(rm1,0);

digitalWrite(rm2,1);

}

void stopie()

{

digitalWrite(lm1,0);

digitalWrite(lm2,LOW);

digitalWrite(rm1,0);

digitalWrite(rm2,LOW);

}

**4.2 ESTIMATED BILL OF MATERIALS**

|  |  |  |  |
| --- | --- | --- | --- |
| Components | No of units | Price per unit | Total |
| Arduino Nano | 1 | 240 | 240 |
| Chassis | 1 | 50 | 50 |
| Wheels | 2 | 20 | 40 |
| Dc Motor | 2 | 50 | 100 |
| IR Sensor Array | 1 | 280 | 280 |
| Motor Driver | 1 | 80 | 80 |
| Arduino Uno | 1 | 430 | 430 |
| Servo Motor | 1 | 100 | 100 |
| Ultra Sound Sensor | 1 | 80 | 80 |
| 9V Battery | 3 | 15 | 45 |
| Total |  |  | Rs 1445 |

**CHAPTER 6**

**6.1 FUTURE ENHANCEMENTS**

The present developed self-independent robot follows a specified path based on IR sensing and scans for any objects in its proximity which in-turn is displayed on monitor using a special type of software called as Processing-3. But there is always a room for development.

* We can develop the present robot as an obstruction senor robot which can avoid obstacle on its path using laser senor or IR sensor.
* As the robot is capable of detecting objects near its proximity ,it has its applications in real time surveillance in and around military installations . An automatic firing system integrated with the sensor to calculate the distance and accurate position of the foreign object using sophisticated codes.
* Now considering , the firing system has been added to the robot ; if the robot is capable of firing at foreign enemy , it is also capable of firing at friendly soldiers and vehicles . In order to overcome such problems we can use the concept of image processing.
* Image processing can be used to identify special notions or symbols designed for the friendly vehicles and prevent unwanted destruction of valuable resources.
* As it is known that each person has different results for Iris scanning; an additional option of Iris/Retinal scan can be used to differentiate people who are authorized to enter restricted areas like an army camp or important data centre from those who aren’t. This can improve the security level of classified information and developed weaponry, which if in wrong hands can lead to mass destruction.
* As we are using it for surveillance and security reasons , power and energy consumption is a major factor to be looked upon as it has to be working all the time .Series of photovoltaic cells can be used to harness solar energy which in-turn can run different peripherals. This

in-turn cuts down cost of batteries and promotes use of renewable sources of energy

**6.2 CONCLUSION**

Self Surveillance Robot using Ultrasonic Sensors has many powerful application including military applications and applications in the field of Aerospace. It can be used to detect harmful weapons and bombs especially in border areas, it also finds it application in security field. Ultrasonic technique is powerful technique which is used in controlling air traffic, to detect planes and other flying objects in the sky. Overall its is a very powerful and useful robot.