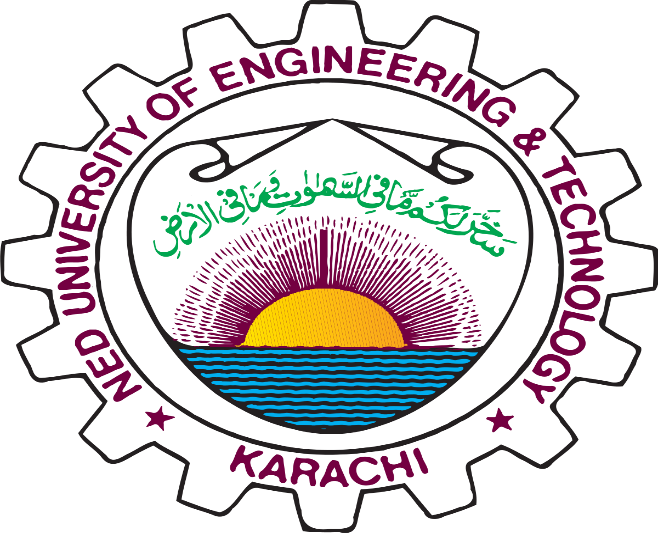
**Arduino Based RADAR System**

****

**Group members name:**

Muhammad Asadullah, Syed Rafay Hashmi, Zain Mansoor

**Group members Roll.no:**

EL-103, EL-098, EL-093

**Course title:**

Electrical Machines

S.E Electronic Engineering Department

Section C

Batch 2017-18

****

**Abstract**

RADAR is an object detection system which uses radio waves to determine the range, altitude, direction, or speed of objects. The radar dish or antenna transmits pulses of radio waves or microwaves which bounce off any object in their path. Arduino is a single-board microcontroller to make using electronics in multidisciplinary projects more accessible. This project aims at making a RADAR that is efficient, cheaper and reflects all the possible techniques that a radar consists of.

**I.INTRODUCTION**

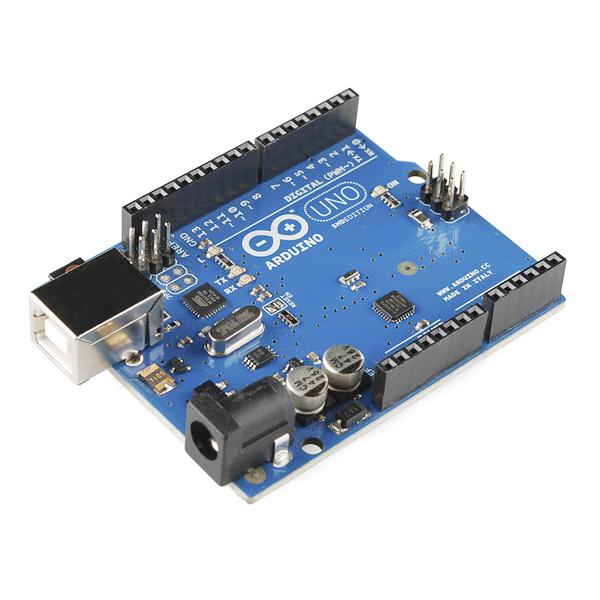
RADAR is an object detection system which uses radio waves to determine the range, altitude, direction, or speed of objects. Radar systems come in a variety of sizes and have different performance specifications. Some radar systems are used for air-traffic control at airports and others are used for 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 and are capable of extracting useful information from very high noise levels.

**II. LITERATURE SURVEY**

“The Idea” Army, Navy and the Air Force make use of this technology. The use of such technology has been seen recently in the self parking car systems launched by AUDI, FORD etc. And even the upcoming driverless cars by Google like Prius and Lexus. This setup can be used in any systems the customer may want to use like in a car, a bicycle or anything else. The use of Arduino in this provides even more flexibility of usage of the above-said module according to the requirements. The idea of making an RADAR came as a part of a study carried out on the working and mechanism of “Automobiles of Future”. Hence this time I was able to get a hold of one of the Arduino boards, Arduino UNO R3. So, knowing about the power and vast processing capabilities of the Arduino, I thought of making it big and a day to day application specific module that can be used and configured easily at any place and by anyone. Moreover, in this fast moving world there is an immense need for the tools that can be used for the betterment of the mankind rather than devastating their lives. Hence, from the idea of the self driving cars came the idea of self parking cars. The main problem of the people in the world is safety while driving. So, this gave up a solution to that by making use of this project to continuously scan the area for traffic, population etc. and as well as protection of the vehicles at the same time to prevent accidents or minor scratches to the vehicles.

**III. COMPONENTS REQUIRED**

**A. Arduino UNO SMD**

The Arduino Uno SMD is a version of the [Arduino Uno](https://www.arduino.cc/en/Main/ArduinoBoardUno), but uses an surface mount version of the Atmega328P instead of the through-hole version. This version was made in response to a shortage in supply of the through-hole Atmega328P. The board is based on the ATmega328 . It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter."Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](https://www.arduino.cc/en/Main/Boards).

**B.Ultrasonic Sensor**

Ultrasonic sensors (also known as transceivers when they both send and receive, but more generally called transducers) work on a principle similar to radar or sonar which evaluate attributes of a target by interpreting the echoes from radio or sound waves respectively. Ultrasonic sensors generate high frequency sound waves and evaluate the echo which is received back by the sensor. Sensors calculate the time interval between sending the signal and receiving the echo to determine the distance to an object. This technology can be used for measuring wind speed and direction (anemometer), tank or channel level, and speed through air or water. For measuring speed or direction a device uses multiple detectors and calculates the speed from the relative distances to particulates in the air or water. To measure tank or channel level, the sensor measures the distance to the surface of the fluid. Further applications include: humidifiers, sonar, medical ultra sonography, burglar alarms and non-destructive testing. Systems typically use a transducer which generates sound waves in the ultrasonic range, above 18,000 hertz, by turning electrical energy into sound, then upon receiving the echo turn the sound waves into electrical energy which can be measured and displayed.

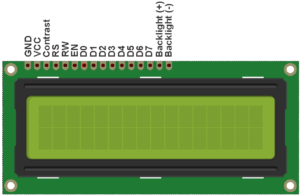
**C. Servo Motor** A servomotor is a rotary actuator that allows for precise control of angular position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors. Servomotors are not a different class of motor, on the basis of fundamental operating principle, but uses servomechanism to achieve closed loop control with a generic open loop motor. Servomotors are used in applications such as robotics, CNC machinery or automated manufacturing.

**SERVO MOTOR MG996R:**

Stall torque: 9.4kg/cm (4.8v); 11kg/cm (6.0v)  
Operating speed: 0.19sec/60degree (4.8v); 0.15sec/60degree (6.0v)  
Operating voltage: 4.8~ 6.6v  
Gear Type: Metal gear  
Servo wire length: 32cm  
It is stronger and lighter than copper.

360 degree rotation

**D.LCD Pinout**

All driver based LCDs have 14 or 16 line interface. Backlight connections are connected to +5V supply through 330 Ohm current limiting resistor. In most cases we don’t read the display so it is better to connect R/W LCD pin to GND. For contrast setting, connect LCD contrast Pin 3 to ground through a 1K Ohm resistor. It gives optimum value of contrast. We don’t need to adjust the contrast every time using variable resistor so it is better to use fixed resistor, saves cost and space on PCB.The LCDs have a parallel interface, meaning that the microcontroller has to manipulate several interface pins at once to control the display. The interface consists of the following pins:

1.A register select (RS) pin that controls where in the LCD’s memory you’re writing data to. You can select either the data register, which holds what goes on the screen, or an instruction register, which is where the LCD’s controller looks for instructions on what to do next.

2.A Read/Write (R/W) pin that selects reading mode or writing mode

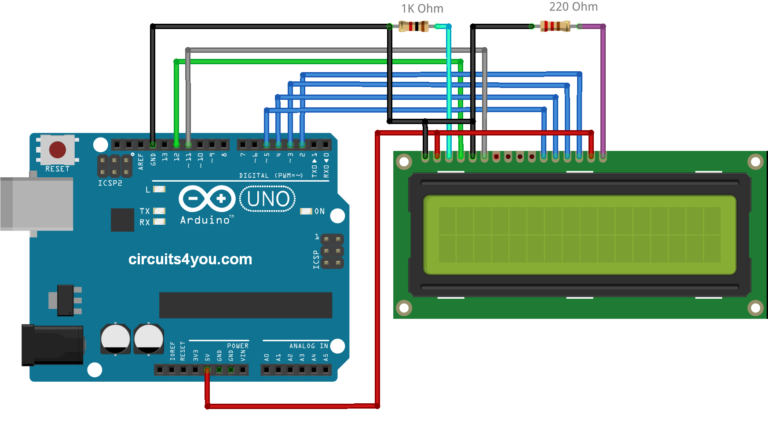
3.An Enable pin that enables writing to the registers

4.8 data pins (D0 -D7). The states of these pins (high or low) are the bits that you’re writing to a register when you write, or the values you’re reading when you read.

5.There’s also a display constrast pin (Vo), power supply pins (+5V and Gnd) and LED Backlight (Bklt+ and BKlt-) pins that you can use to power the LCD, control the display contrast, and turn on and off the LED backlight, respectively.

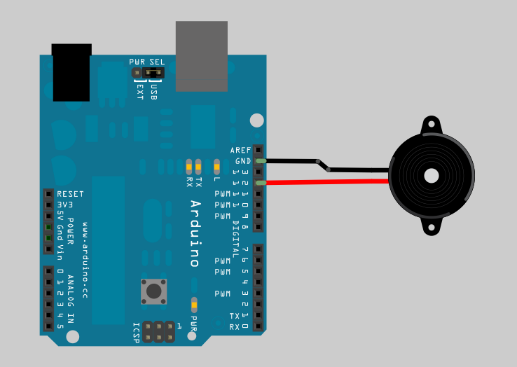
**Connecting LCD to Arduino UNO**

Before wiring the LCD screen to your Arduino UNO or Genuino board we suggest to solder a pin header strip to the 14 (or 16) pin count connector of the LCD screen, as you can see in the image above.  
To wire your LCD screen to your board, connect the following pins:

* LCD RS pin to digital pin 12
* LCD Enable pin to digital pin 11
* LCD D4 pin to digital pin 5
* LCD D5 pin to digital pin 4
* LCD D6 pin to digital pin 3
* [](https://circuits4you.com/wp-content/uploads/2016/05/LCD-Arduino-Circuit.png)LCD D7 pin to digital pin 2

Additionally, wire a 1k resistor in between GND and  LCD screens VO pin (pin3). A 220 ohm resistor is used to power the backlight of the display, usually on pin 15 and 16 of the LCD connector

**E. LED bulb & Buzzer :**

We use total 6 leds , in which 3 are red and other 3 are green. Hence we choose this color combination because the color red is a sign of danger so we use it for a danger sign. For example if any object comes to it and the distance between the sensor and object is less than 10cm it gives red signal and while for other ranges it gives green signal. While we use buzzer in it. The use of buzzer is more specific than other components. When the object come closer to it, it gives red light that is danger and buzzer is also starts. And for the connections we use jumber wires and to give it voltage we use battery.

**IV. USING ARDUINO SOFTWARE**

The Arduino integrated development environment (IDE) is a cross-platform application written in Java, and is derived from the IDE for the Processing programming language and the Wiring 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, 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 or C++. The Arduino IDE comes with a 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 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.

• 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.

**V.Coding**

#include <Servo.h>

#include <LiquidCrystal.h>

Servo myservo;

LiquidCrystal lcd(10, 11, 14, 15, 16, 17); // Creates an LCD object. Parameters: (rs, enable, d4, d5, d6, d7)

int pos = 0;

const int trigPin = 8;

const int echoPin = 9;

const int moteur = 2;

const int buzzer = 6;

const int ledPin1 = 4;

const int ledPin2 = 7;

float distanceCm, DistanceSec,duration;

void setup() {

myservo.attach(moteur); // attache le Servo moteur a la pin numéro 11

lcd.begin(16,2); // Initialiser l'interface de Lcd avec leurs Dimensions

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

pinMode(buzzer, OUTPUT);

pinMode(ledPin1, OUTPUT);

pinMode(ledPin2, OUTPUT);

DistanceSec=20;}

void loop() {

for (pos = 0; pos <= 180; pos += 10) {

myservo.write(60);

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distanceCm= duration\*0.034/2;

if (distanceCm <= DistanceSec)

{myservo.detach();

if(distanceCm <= DistanceSec/2)

{tone(buzzer, 10); // Send 1KHz sound signal...

digitalWrite(ledPin1, LOW);

digitalWrite(ledPin2, HIGH);

delay(700);

noTone(buzzer); // Stop sound...

lcd.setCursor(0,0);

lcd.print("Distance: ");

lcd.print(distanceCm);

lcd.print(" cm ");

delay(10);

lcd.setCursor(0,1);

lcd.print("Object is near");

delay(2000);}

else

{myservo.detach();

digitalWrite(buzzer, HIGH);

digitalWrite(ledPin2, LOW);

digitalWrite(ledPin1, HIGH);

delay(100);

digitalWrite(buzzer, LOW);

lcd.setCursor(0,0);

lcd.print("Distance: ");

lcd.print(distanceCm);

cd.print(" cm ");

delay(10);

lcd.setCursor(0,1);

lcd.print("Object is near ");

delay(2000);}}

else{ myservo.attach(2);

digitalWrite(buzzer, LOW); //here

digitalWrite(ledPin1, LOW);

digitalWrite(ledPin2, LOW);

myservo.detach();}

myservo.attach(2);

lcd.setCursor(0,0);

lcd.print("Distance: ");

lcd.print(distanceCm);

lcd.print(" cm ");

delay(10);

lcd.setCursor(13,1);

lcd.print(" ");

lcd.setCursor(0,1);

lcd.print("Object is far");

delay(80); }

for (pos = 180; pos >= 0; pos -= 10) {

myservo.write(110);

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distanceCm= duration\*0.034/2;

if (distanceCm <= DistanceSec){

myservo.detach();

if(distanceCm <= DistanceSec/2)

{tone(buzzer, 10); // Send 1KHz sound signal...

digitalWrite(ledPin1, LOW);

digitalWrite(ledPin2, HIGH);

delay(700);

noTone(buzzer); // Stop sound...

lcd.setCursor(0,0);

lcd.print("Distance: ");

lcd.print(distanceCm);

lcd.print(" cm ");

delay(10);

lcd.setCursor(0,1);

lcd.print(" Object is near");

myservo.detach();

delay(2000);}

else

{digitalWrite(buzzer, HIGH);

digitalWrite(ledPin2, LOW);

digitalWrite(ledPin1, HIGH);

delay(100);

digitalWrite(buzzer, LOW);

lcd.setCursor(0,0);

lcd.print("Distance: ");

lcd.print(distanceCm);

lcd.print(" cm ");

delay(10);

lcd.setCursor(0,1);

lcd.print("Object is near");

delay(2000);}}

else{myservo.attach(2);

digitalWrite(buzzer, LOW);

digitalWrite(ledPin1, LOW);

digitalWrite(ledPin2, LOW);}

lcd.setCursor(0,0);

lcd.print("Distance: ");

lcd.print(distanceCm);

lcd.print(" cm ");

delay(10);

lcd.setCursor(13,1);

lcd.print(" ");

lcd.setCursor(0,1);

lcd.print("Object is far");

delay(80);}}

**VIII. APPLICATIONS**

The idea of making an Ultrasonic RADAR appeared to us while viewing the technology used in defense, be it Army, Navy or Air Force and now even used in the automobiles employing features like automatic/driverless parking systems, accident prevention during driving etc. The applications of such have been seen recently in the self parking car systems launched by AUDI, FORD etc. And even the upcoming driverless cars by Google like Prius and Lexus

1. **Air Force**

In aviation, aircraft are equipped with radar devices that warn of aircraft or other obstacles in or approaching their path, display weather information, and give accurate altitude readings. The first commercial device fitted to aircraft was a 1938 Bell Lab unit on some United Air Lines aircraft. Such aircraft can land in fog at airports equipped with radar-assisted ground-controlled approach systems in which the plane's flight is observed on radar screens while operators radio landing directions to the pilot.

1. **Naval Applications**

Marine radars are used to measure the bearing and distance of ships to prevent collision with other ships, to navigate, and to fix their position at sea when within range of shore or other fixed references such as islands, buoys, and lightships. In port or in harbor, vessel traffic service radar systems are used to monitor and regulate ship movements in busy waters.

1. **Applications in Army**

Two video cameras automatically detect and track individuals walking anywhere near the system, within the range of a soccer field. Low-level radar beams are aimed at them and then reflected back to a computer, which analyzes the signals in a series of algorithms. It does this by comparing the radar return signal (which emits less than a cell phone) to an extensive library of “normal responses.” Those responses are modeled after people of all different shapes and sizes (SET got around to adding females in 2009). It then compares the signal to another set of “anomalous responses” – any anomaly, and horns go off. Literally, when the computer detects a threat, it shows a red symbol and sounds a horn. No threat and the symbol turns green, greeting the operators with a pleasant piano riff.