VIETNAM AVIATION ACADEMY

Department of Telecommunication - Electronics Engineering Technology

LOCATED IN HO CHI MINH CITY

PROJECT REPORT:

"Radar detector module using Arduino"

Written by

Nguyen Van Anh Tuan

Roll.No.1753020018

Under the guidance of

Msc.Cao Xuan Kim Anh

May 25, 2020

VIETNAM AVIATION ACADEMY

Department of Telecommunication - Electronics Engineering Technology

LOCATED IN HO CHI MINH CITY

PROJECT REPORT:

"Radar detector module using Arduino"

Written by

Nguyen Van Anh Tuan

Roll.No.1753020018

Under the guidance of

Msc.Cao Xuan Kim Anh

May 25, 2020

PREAMBLE

Radar is an object detection system. It uses Microwaves to determine the range, altitude, or speed of objects. The radar can transmit radio waves or microwaves which bounce off any object in their path. So, we can easily determine any object in the radar range. Arduino is a single-board microcontroller to make electronics more discipline. The radar system has different performance specifications and also it comes in a verity of size.

Auth.Nguyen Van Anh Tuan

WORDS OF THANKS

Special thanks to Mrs.Kim Anh and my friends for helping me completing this project.

I have tried my best to do this project. However, due to my lack of experience and knowledge, there are still some unexpected mistakes in the project. Please let me know your opinions and criticizes. Once again, thank you so much.

Auth.Nguyen Van Anh Tuan

Contents

ABSTRACT

In this project, we will find out about radar detector and the component i used in this project. First, i will introduce and show data of them. Next, i will show and analysis paradigm, circuit diagram and my product when i finished the project. And indispensable is code i used for my project.

Chapter 1

Introduction

1.1 PRELIMINARY INTRODUCTION

1.1.1 The reason why to choose project

With the passion for aviation as well as passion for technology and equipment realted to it, i decided to choose an aviation-related project in this project. Fortunately, my project this time is on topic of embedded programming. So, i choose project named "PSR radar detector module using Arduino". In this project, i will rely on the PSR radar to make a small scale PSR radar detector model. So, to get started in this project, we need to know what PSR radar is and how it works.

Recognize the continuous development of aviation technology. I want to add my own knowledge about how a radar system works and a bit of creative idea for this device that came along during i make this project. And that's why i choose this project for myself.

Block diagram

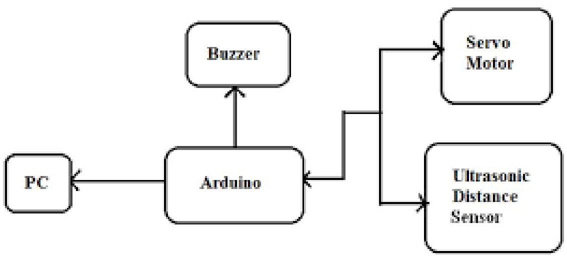


Figure 1.1: Block diagram for radar model

You may ask how the processing application works here. It's very simple, the Ultrasonic sensor collects the object information with the help of Arduino and passes it to processing application, there is a simple Graphics application implemented which mimic a radar screen.

1.1.2 Target Research

The short term goal of this topic is with the desire to learn and supplement the knowledge that in the course of research. With the long term goal, i want to perform the topic in the best way i can. As well as improve the errors of myselft. And also, i want to additional the knowledge i haven't learned at my school.

1.1.3 Object and position research

- **Object Research:** The object that i study is the sensor system installed on the air traffic control station or installed on robots that detect objects and avoid them.
- **Position Research:** My reserach is based on the application of radar to detect missing vehicles or to apply air traffic control as call as "Primary Surveillance Radar".

1.1.4 Method of research

- **Observation Method:** By observing directly at air traffic control and also via movies or aviation videos on internet.
- **Method of analysis:** Looking for some similar projects that have been made available online, from the detailed data of those projects, i draw some methods and experience for my project. Avoid mistakes in my project.

1.1.5 Structure Project

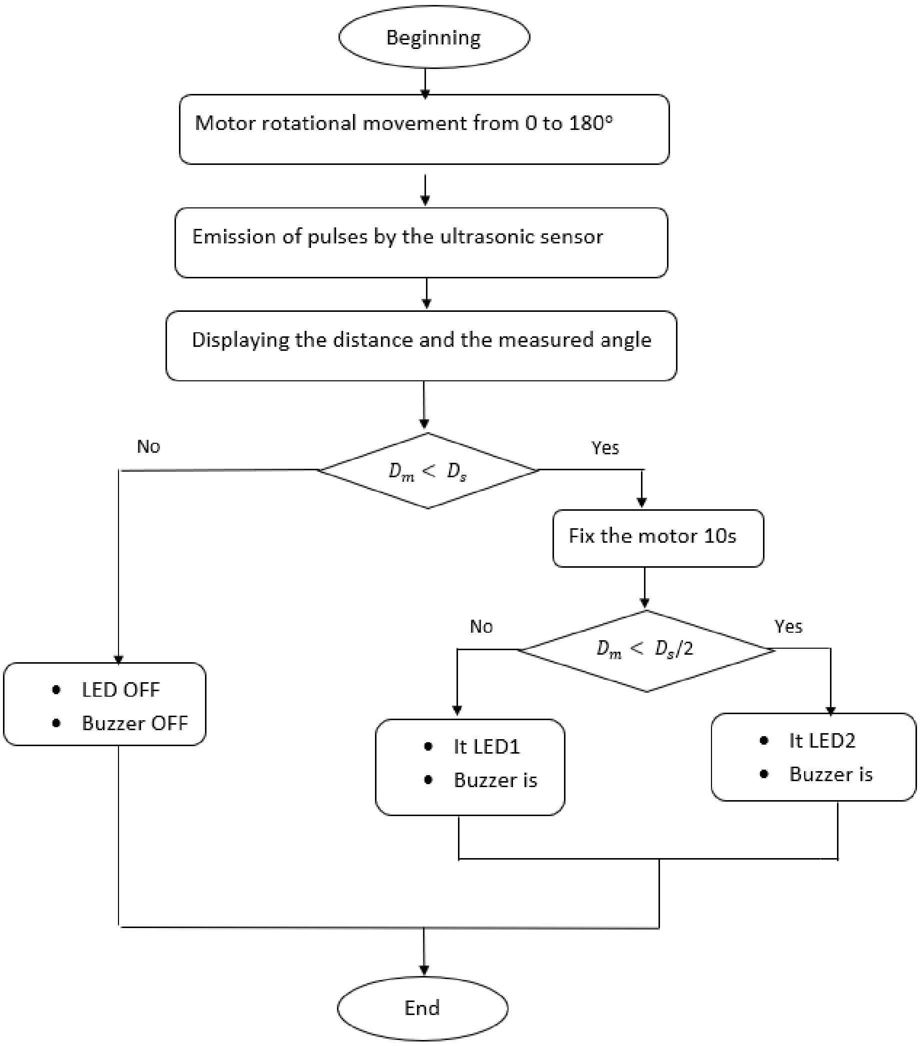


Figure 1.2: Flowchart for project

Arduino board sends a signal of 5V to TRIG pin of Ultrasonic Sensor HC-SR04 which triggers the sensor. Then it procides rotational action at the servo motor mechanically fitted along with Ultrasonic sensor so that it can detec the moving objects and locate within 180 degrees.

The Arduino sends a HIGH pulse width of (10 S) on the TRIGGER pin of the sensor to regenerate a series of ultrasonic waves which propagate through the air, until it touches an obstacle and returns in the opposite direction towards the sensor pin ECHO. The sensor detecs the width of the pulse to calculate the distance.

The signal on ECHO pin of the sensor remains at the HIGH position during transmission, thereby measuring the duration of the round trip of ultrasound and thus determine the distance.

The LCD display displays the calculated distance and the angle of rotation. The buzzer is an additional component, it rings when there is a detection (Tone 1 and Tone 2) along with the LEDs. These both LEDs along with buzzer determine the field where the object is located (near or distant).

We will have to convert this time into centermet to calculate the distance traveled. We will use the following equation to calculate it.

$$\mathbf{S = V^*t}$$

The ultrasonic wave is basically a sound wave that travels at a speed of 340m/s (0.034cm/s). The ultrasonic is measuring the time it takes to hit the object and then come back but we need only time that it takes to hit the object. So we will divide it by 2.

$$\mathbf{S = (t * 0.034)/2}$$

1.2 BASIC THEORY

1.2.1 Theory about the radar

At first, before we learn about how this project works, let's take a quick look at the definition of radar. Now we can see objects in the world around us because light(usually from the Sun) reflects off them into your eyes. If you want to walk at night, you can shine a torch in front to see where are you going.

Radar works in much the same way. The word "Radar" stands for **RA**dio **D**etection **A**nd **R**anging, and that gives a pretty big clue as to what it does and how it works. Imagine an airplane flying at night through thick fog. The pilot can't see where are they going, so they use the radar to help them.

An airplane's radar is a bit like a torch that uses radio waves instead of light. The plane transmits and intermittent radar beam (so it sends a signal only part of the time) and for the rest of the time.

1.2.2 Some research related to the project

Some research ideas related to my project:

- The function contained in some robots, helps robots scan the terrain and detect objects to avoid.
- Radar in the Air traffic control tower named "Primary Surveillance Radar"

1.2.3 Theory concepts related to research issues

- **PSR(Primary Surveillance Radar)**: A Surveillance radar system which uses reflected radio signals.
- **US(Ultrasonic Sensor)**: As the name indicates, ultrasonic sensors measure distance by using ultrasonic waves.
- **PWM(Pulse-Width Modulation)**: is a method of reducing the average power delivered by an electrical signal, by effectively chopping it up into discrete parts.
- **RAM(Random-Access Memory)**: is a form of computer memory that can be read and changed in any order, typically used to store working data and machine code.
- **CPU(Central Processing Unit)**: also called a central processor or main processor, is the electronic circuitry within a computer that executes instruction that make up a computer program.

1.2.4 Components Required

1. For power: Micro USB-B
2. For radar model:
   (a) Arduino UNO
   (b) Servo motor
   (c) Ultrasonic Sensor HRF-04
   (d) Buzzer

(e) LCD 16x02
(f) LED (green, red)
(g) Test board

1.2.5 Component Description

1.2.5.1 Arduino Uno

1. **Introduction about Arduino UNO**

   Arduino UNO is a microcontroller board developed by Arduino.cc which is open-source electronics platform mainly based on AVR microcontroller ATMega328.

   First Arduino projcet was started in Interaction Design Institute Ivrea in 2003 by David Cuartielles and Massimo Banzi with the intention of providing a cheap and flexible way to students and professional for controlling a number of devices in the real world.

   The current version of Arduino UNO comes with USB interface, 6 analog input pins, 14 I/O(input/output) digital ports that are used to connect with external electronic circuit. Out of 14 I/O ports, 6 pins can be used for PWM output.

   It allows the designers to control and sense the external electronic devices in the real world.

   Apart from USB, battery or AC to DC adopter can also be used to power the board.

   There are many versions of Uno board available. However, Arduino Uno V3 and Arduino Uno are the most official versions that come with ATMega328 8-bit AVR Atmel microcontroller where RAM memory is 32KB.

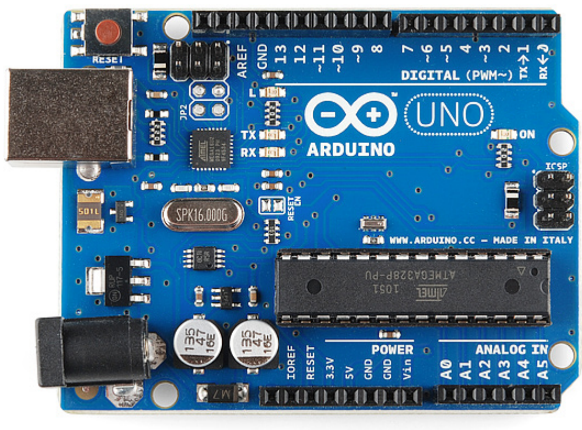


Figure 1.3: Arduino Uno board

2. **Features**

   - Comes with USB interface.
   - USB port is added on the board to develop serial communication with the computer
   - ATMega328 microcontroller is placed on the board that comes with a number of features like timers, counters, interrupts, PWM, CPU, I/O pins and based

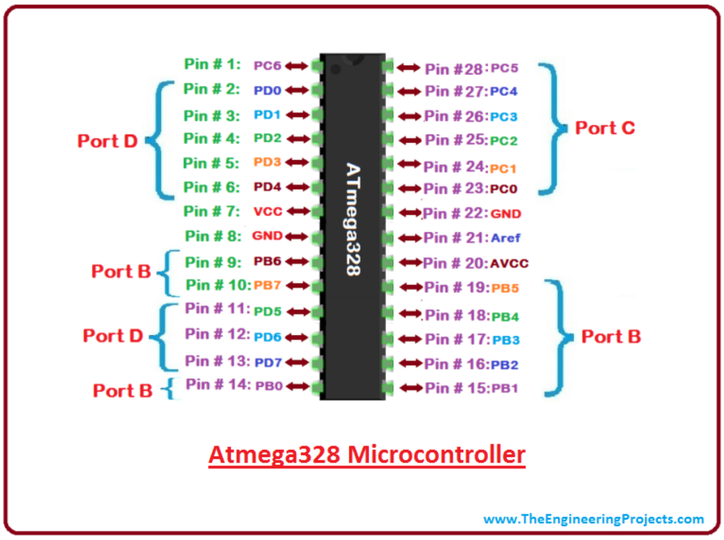


Figure 1.4: ATMega328 Microcontroller

on a 16MHz clock that helps in producing more frequency and number of instruction per cycle.

- It's an open-source platform where anyone can modify and optimize the board based on the number of instructions and task they want to achieve.
- Come with a built-in regulation feature which keeps the voltage under control when the device is connected to the external device.
- There are 14 pins I/O digital and 6 analog pins in the board that allows the external connection with any circuit with the board. These pins provide the flexibility and ease of use to the external devices that can be connected through these pins.
- 6 analog pins are marked as A0 to A5 and come with a resolution of 10bits. These pins measure from 0V to 5V, however, they can be configured to the high range using analogReference() function and AREF pin.
- 13KB of flash memory is used to store the number of instructions in the form of code.
- Only 5V is required to turn the board on, which can be achieved directly using USB port or external adopter, however, it can support external power source up to 12V which can be regulated and limit to 5V or 3.3V based on the requirement of the project.

3. **Pinout**

Arduino Uno is based on AVR microcontroller call ATMega328. This controller comes with 2KB RAM, 32KB of flash memory, 1KB of EEPROM. Arduino board comes with 14 digital pins and 6 analog pins.

4. **Pin Description**

LED: comes with build-in LED which is connected through pin 13. Providing HIGH value to the pin will turn it ON and LOW will turn it OFF.

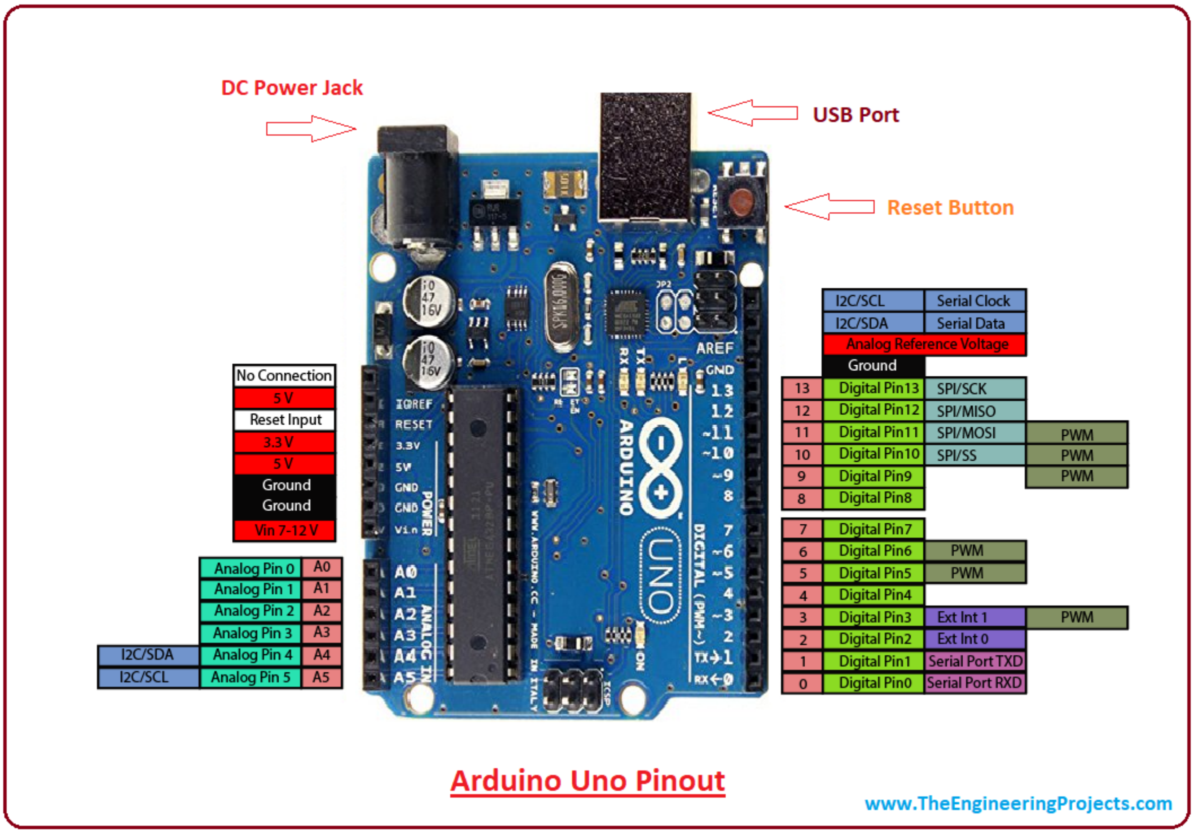


Figure 1.5: Arduino Uno Pinout

Vin: input voltage provided to the Arduino Uno board. It's different than 5V supplied through the USB port. This pin is used to supply voltage. If a voltage is provided through power jack, it can be accessed through this pin.

5V: comes with the ability to provide voltage regulation. 5V pin is used to provide output regulated voltage. The board is powered up using the 3 ways: USB, Vin pin of the board or DC power jack.

GND: ground pins. More than one ground pins are provided on the board which can be used as per requirement.

Reset: this is incorporated on the board which resets the program running on the board. Instead of physical reset on the board, IDE comes with a feature of resetting the board through programming.

IOREF: this pin very useful for providing voltage reference to the board. A shield is used to read the voltage across this pin which then select the proper power source.

PWM: is provided by 3,5,6,9,10; 11 pins. These pins are configured to provided 8-bit output PWM.

SPI: It is known as Serial Peripheral Interface. 4 pins 10(SS), 11(MOSI), 12(MISO), 13(SCK) provide SPI coummunication with the help of SPI library.

AREF: called Analog Reference. This pin is used for providing a reference voltage to the analog inputs.

TWI: called Two-tire Interface. TWI communication is accessed through Wire

Library. A4 and A5 pins are used for this purpose.

Serial Communication: is carried out through 2 pins call pin 0(Rx) and pin 1(Tx). Rx pin is used to receive data while Tx pin is used to transmit data.

External Interrupts: Pin 2 and 3 are used for providing external interrupts. An interrupt is called by providing LOW or changing value.

5. **Communication and Programming**

   Arduino Uno comes with an ability of interfacing with other Arduino boards, microcontrollers and computer. The ATMega328 placed on board provides serial communication using pins like Rx and Tx.

   Arduino Uno is programmed using Arduino Software which is a cross-platform application call IDE(Intergrated Development Environment) written in Java. The AVR microcontroller ATMega328 laid out on the base comes with built-in bootloader that sets you free from using a separate burner to upload the program on the board.

6. **Application**

   Arduino Uno comes with a wide range of Applications. A larger number of people are using Arduino boards for developing sensors and instruments that are used in scientific research. Following are some main applications of the board:

   »Embedded System
   »Security and Defense System
   »Digital Electronics and Robotics
   »Parking Lot Counter
   »Weighing Machines
   »Traffic Light Count Down Timer
   »Medical Instrument
   »Emergency Light for Railways
   »Home Automation
   »Industrial Automation

   There are a lot of other microcontrollers available in the market that are more powerful and cheap as compared to Arduino board.

   Actually, Arduino comes with a big community that is developing and sharing the knowledge with a wide range of audience. Quick support is available pertaining to technical aspects of any electronic project.

1.2.5.2 Servo Motor

1. **Introduction**

   Servo Motors(or servos) are self-contained electric devices that rotate or push parts of machine with great precision. Servos are found in many places, from toys to home electronics to cars and airplanes. If you have a radio-controlled model car, airplanes, or helicopter, you are using at least a few servos. By rotating a shaft connected to the engine throttle, a servo regulates the speed of a fuel-powered car or aircraft.

Servo also appear behind the scene in devices we use every day. Electronic device such as DVD players use servos to extend or retract the disc trays.

Figure 1.6: Assortment of Servos

2. **How does a servo work?**

The simple of a servo among the features that make them so reliable. The heart of a servo is a small direct current (DC) motor, similar to what you might find in an cheap toy. These motors run on electricity from a battery and spin at high RPM(Rotations per minute) but put out very low **torque**(a twisting force used to do work you apply torque when you open a bottle).

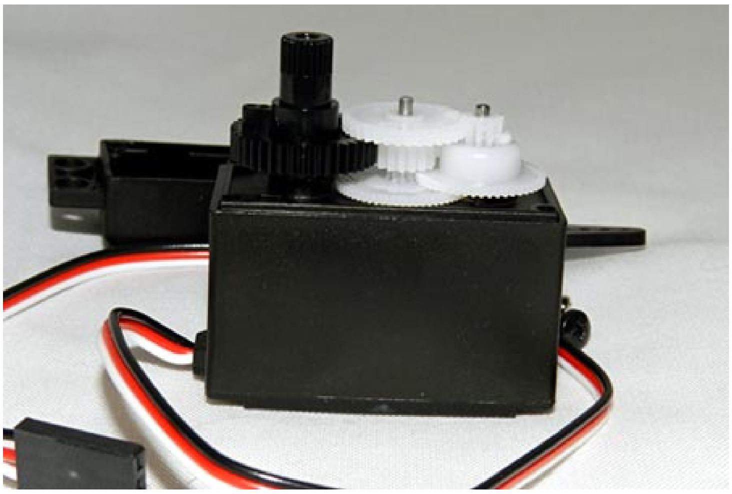


Figure 1.7: The Gears in a Typical Standard-size servo

In high-power servo, the plastic gears are replaced by metal ones for strength. The motor is usually more powerful than in a low-power servo and the overall output torque can be as much as 20 times higher than a cheaper plastic one. Better quality is more expensive, and high-output servos can cost two or three times as much as standard ones.

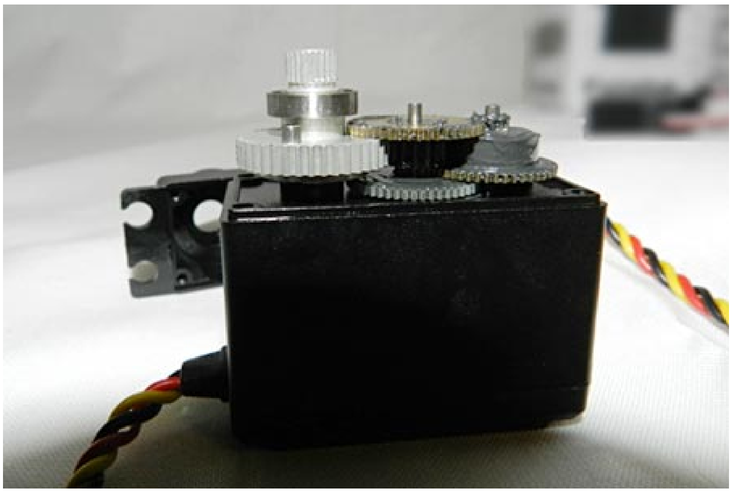


Figure 1.8: The gears in a Typical High-power servo

With a small DC motor, you apply power from a battery, and the motor spins. Unlike a simple DC motor, however, a servo's spinning motor shaft is slowed way down with gears. A positional sensor on the final gear is connected to a small circuit board. The sensor tells this circuit board how far the servo output shaft has rotated. The electronic input signal from the computer or the radio in a remote-controlled vehicle also feeds into that circuit board. The electronics on the circuit board decode the signals to determine how far the user wants the servo to rotate. Then compares the desired position to the actual position and decides which direction to rotate the shaft so it gets to the desired position.

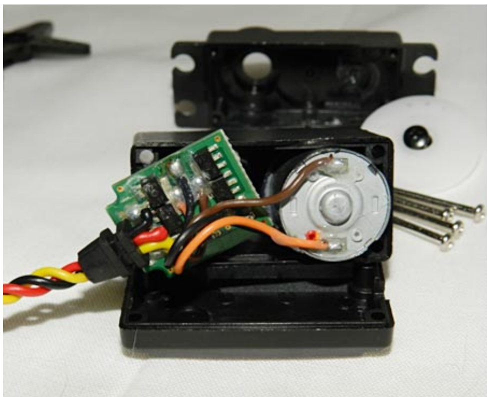


Figure 1.9: The circuit board and DC motor in a high-power servo

3. **Types of servo motors**

Servos come in many sizes and in 3 basic types:

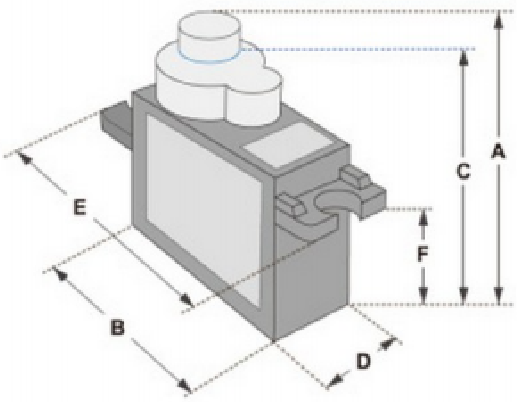
- **Positional rotation servo:** This is the most common type of servo motor. The output shaft rotates in about half of a circle, or 180 degrees. It has physical stops placed in the gear machanism to prevent turning beyond these limits to protect the rotational sensor. These common servos are found in radio-controlled cars, aircraft, toys, robots and many applications.

- **Continuous rotation servo:** This is quite similar to the common positional rotation servo motor, except it can turn in either direction indefinitely. The control signal, rather than setting the static position of the servo, is interpreted as the direction and speed of rotation. The range of possible commands causes the servo to rotate clockwise or counterclockwise as desired, at varying speed, depending on the command signal. You might use a servo of this type on a radar dish if you mounted one on a robot. Or you could use one as a drive motor on a mobile robot.
- **Linear servo:** This is also like the positional rotation servo motor described above, but with additional gears(usually a **rack and pinion** mechanism) to change the output from circular to back-and-forth. These servos are not easy to find, but you can sometimes find them at hobby stores where they are used as actuators in larger model airplanes.

4. **Selecting a servo motor**

Figure 1.10: Micro Servo 90

Micro servo motor 90(SG90) is tiny and lightweight with high output power. This servo can rotate approximately 180 degrees (90 in each direction), it works just like the standard kinds but smaller. We can use any servo code, hardware or library to control this servo. It good for beginners who want to make stuff move without building a motor controller with feedback and gear box, especially since it will fit in small places.



Dimensions & Specifications
A (mm) : 32
B (mm) : 23
C (mm) : 28.5
D (mm) : 12
E (mm) : 32
F (mm) : 19.5
Speed (sec) : 0.1
Torque (kg-cm) : 2.5
Weight (g) : 14.7
Voltage : 4.8 - 6

Figure 1.11: Section of SG90

Block Diagram of SG90

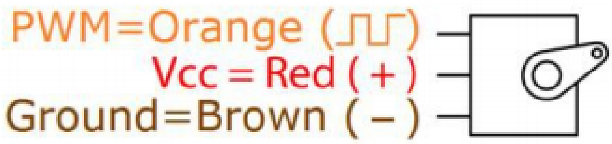


Figure 1.12: SG90 datasheet

5. **Controlling a servo motor**

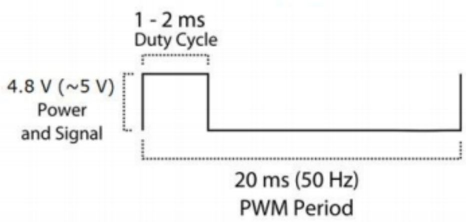


Figure 1.13: PWN Period

Servo take commands from a series of pulse sent from the computer or radio. A pulse is a transition from low voltage to high voltage which stays high for a short time, and then returns to low. In battery devices such as servos, "low" is considered be ground or 0V and "high" is the battery voltage. Servos tend to work in a range of 4.5 to 6 volts, so they are extremely hobbyist computer-friendly.

1.2.5.3 HC-SR04 (Ultrasonic Sensor)

1. **Introduction**

   An Ultrasonic Sensor is an electronic device that measures the distance of target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (the sound that human can hear).

   Transmitter and Receiver are two main parts of the sensor where former converts an electrical signal to ultrasonic waves while later converts that ultrasonic signal back to electrical signals.

   Following table shows the main features of this ultrasonic sensor

Parameter	Value
Main Parts	Transmitter and Receiver
Operating Voltage	5V
Operating Frequency	4MHz
Detection Range	2cm to 400cm
Measuring Angle	30 degrees
Resolution	3mm
Operating Current	< 15mA
Sensor Dimensions	45mm x 20mm x 15mm

Table 1.1: Feature of HC-SR04

   In order to calculate the distance between the sensor and the object, the sensor measures the time it takes between the emission the sound by the transmitter to its contact with the receiver. The formula for this calculation:

$$\mathbf{D = 1/2 * T * C}$$

   Where:

   - D: the distance
   - T: the time
   - C: the speed of sound   343 meter/second

2. HC-SR04 Pinout and Description

   HC-SR04 contain 4 pins in total.

   This figure below here is labelled these HC-SR04 Pinout for better visualization:

3. **How does it work?**

   The **HC-SR04 Ultrasonic Sensor(US)** is an ultrasonic transducer that comes with 4 pin interface named as Vcc, Trigger, Echo, and Ground. It is very useful for accurate distance measurement of the target object and mainly works on the sound waves.

   As we connect the module to 5V and initialize the input pin, it starts transmitting the sound waves which then travel through the air and hit the required object.

No.	Pin name	Pin Description
1	VCC	The power supply pin of the sensor that mainly operates at 5V DC
2	TRIG Pin	It plays a vital role to initialize measurement for sending ultrasonic waves. It should be kept high for 10us for triggering the measurement
3	ECHO Pin	This pin remains high for short period based on the time taken by the ultrasonic waves to bounce back to the receiving end
4	Ground	This pin is connected to ground

Table 1.2: Pinout of HC-SR04

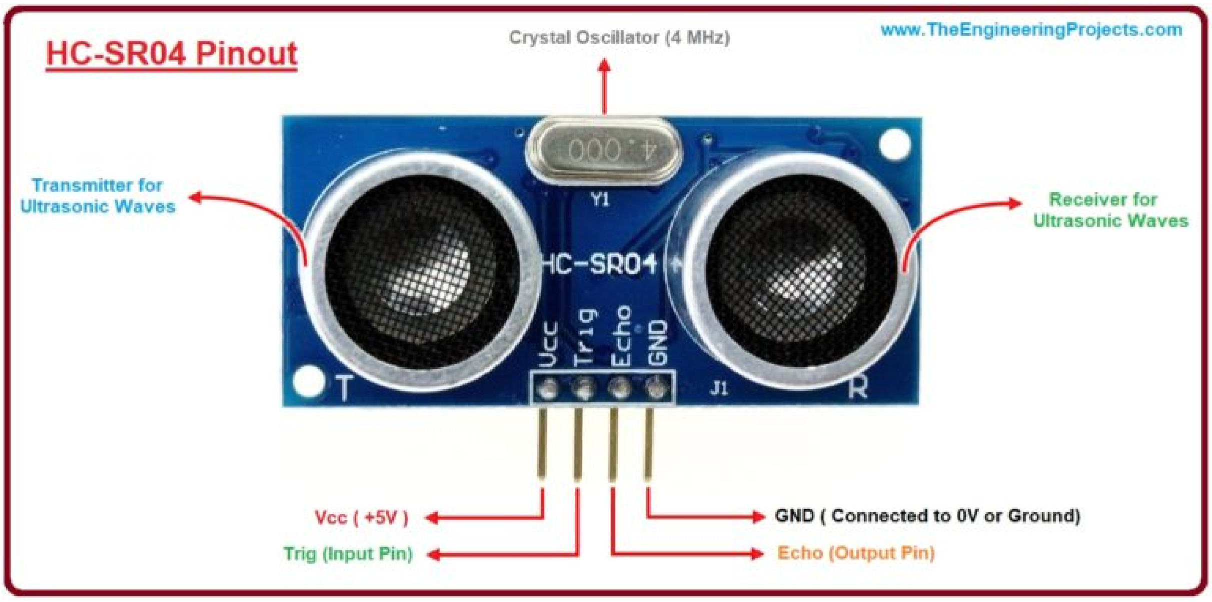


Figure 1.14: HC-SR04 Pinout

These waves hit and bounce back from the object and then collected by the receiver of the module.
Distance is directly proportional to the time these waves require to come back at the receiving end. The more the time taken, more the distance will be.

The waves will be generating if the Trig pin is kept high for 10μS. These waves will travel at the speed of sound, creating 8 cycle sonic burst that will be collected in the Echo pin.

The Echo pin remains turned on for the time these waves take to travel and bounce back to the receiving end. This sensor is mainly incorporated with Arduino to measure the required distance.

Following the formula is used to calculated the distance of the object.

$$\mathbf{S = (V*t)/2}$$

Where S is the required distance, V is the speed of sound and t is the time sound waves takes to come back after hitting the object. We need to divide the value by 2 because time will be double as the waves travel and bounce back from the initial point. Diving it by 2 will give the actual distance of the target object.

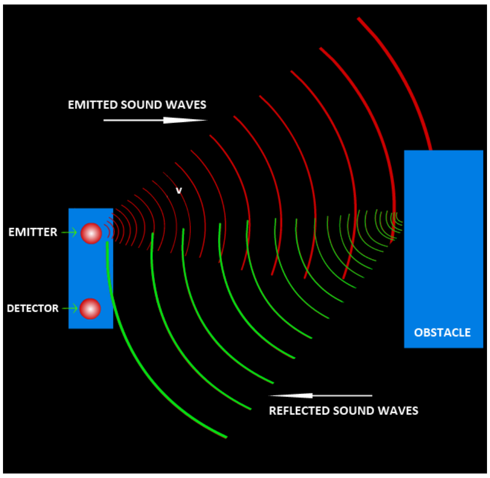


Figure 1.15: Ultrasonic sensor diagram

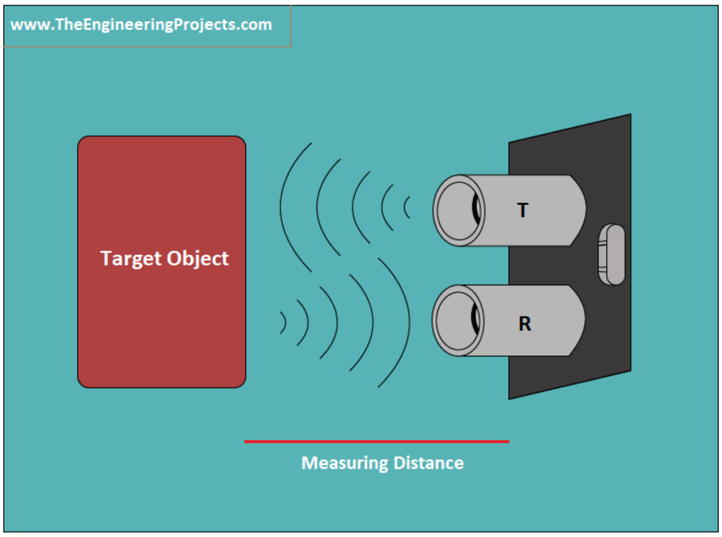


Figure 1.16: Principle of wave transmission

4. **Application**

HC-SR04 comes with a widt range of applications mainly targeting distance and direction measurement. Following are the major applications it can be used for:

»Speed and direction measurement

»Wireless charging

»Humidifiers

»Medical ultrasonography

»Burglar alarms

»Embedded system

»Depth measurement

»Non-destructive testing

1.2.5.4 LCD 16x02

1. **Introduction**

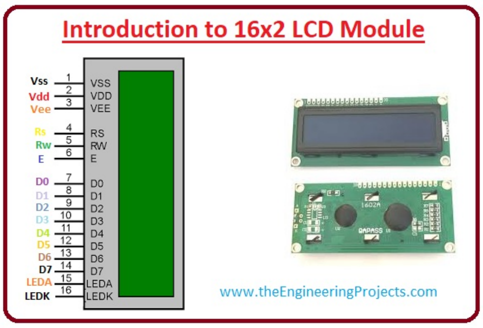


Figure 1.17: LCD 16x2 Module

The liquid crystal display are normally used in different embedded projects due to its low cost, easy access and flexibility to get programmed.

Almost each and every electronic device we daily see it like in your mobile, calculator and some other devices.

There is a type of liquid display that has sixteen column and two rows so it is known as 16 x 02 LCD modules.

LCD also available in different arrangements like (8x1), (10x2), (16x1), but the (16x2) liquid crystal is normally used in embedded projects.

In this liquid crystal display, there are thirty-two characters and each of them consists of 5x8 pixels.

So we can say that character consists of forty pixels or dots and total pixels in this liquid crystal display can be fined as (32x40) or 1280 pixels.

2. **Block Diagram**

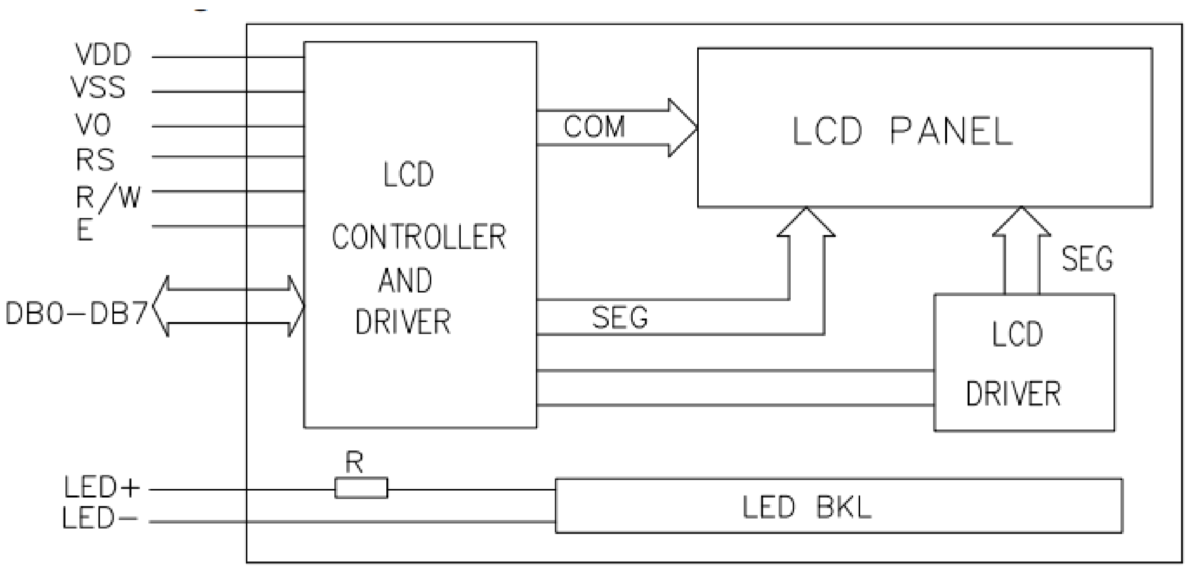


Figure 1.18: Block Diagram of LCD

3. **Pinout Description**

These are the main pinouts of 16x2 LCD that are described here with detailed:

Pin No:	Pin name	Parameters
1	Ground	This pin is used to connect the ground
2	+5 Volt	At this pinout plus five volts are applied to on the LCD
3	VE	This pin used to select the contract of the display
4	Register Select	This pinout is used to MCU controller connected led to a shift from command to data mode
5	Read and Write	It used for reading and wiring of data
6	Enable	It linked with the MCU to toggle among zero and one
7	Data PIN 0	The pinouts from zero to seven are data pinouts and these are linked with the MCU for transmission of data. This liquid crystal module can also operate on the four-bit mode by working on o, 1, 2, and 3 pinouts and others are free
8	Data PIN 1	
9	Data PIN 2	
10	Data PIN 3	
11	Data PIN 4	
12	Data PIN 5	
13	Data PIN 6	
14	Data PIN 7	
15	LED Positive	This pinout for turn backlight of led into positive
16	Led Negative	Backlight liquid crystal display pinout negative terminal

Table 1.3: Pinout of 16x2 LCD

4. **Command code for 16x2 LCD Module**

Sr.NO	Hex Code	Parameters
1	1	This command will remove data displaying on the screen of lcd
2	2	It used to move back home
3	4	It used to change location of a cursor to left side
4	6	It changes the position of cursor to right side
5	5	It used for shift display on right
6	7	It used for Shift display one left
7	8	It used to off the display and cursor will also off
8	0A	It used for both display off, a cursor on
9	0C	It used for display on, cursor also off
10	0E	By using this command we can on display, the cursor will be blinking
11	0F	By this command Display will be on, the cursor also blinking
12	10	It changes the location of a cursor to left
13	14	It set cursor location to right
14	18	It changes the location of the complete display to the left side
15	1C	It changes the location of the complete display to right side
16	80	It used to move the cursor to the first line
17	C0	It send the cursor to starting of the second line
18	38	2 lines and 5x7 matrix

Table 1.4: Command code

5. **Features**

   »These are some features of 16x2 LCD module that are described with the detailed
   »Its functioning voltages are from 4.7-5.3 volts
   »It uses one mA current for operation
   »In this liquid crystal display, we can work both alphabets and numbers
   »On this module, there are rows each has sixteen characters
   »Every character of this board has 5 x 8 or 40 pixels
   »It works on both four and eight bits mode
   »It display screen backlight is two colour green and blue

1.2.5.5 Buzzer

1. **General Introduction**

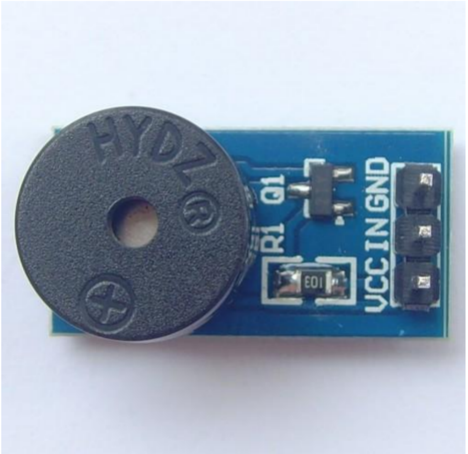


Figure 1.19: Buzzer

A buzzer or beeper is an audio signalling device, which maybe mechanical, electromechanical, or piezoelectric(piezo for short). Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke.

Buzzer is an intergrated structure of electronic transducers, DC power supply, widely used in computers, printers, copiers, alarms, electronic toys,... Active buzzer 5V rated power can be directly connected to a continuous sound, this section dedicated sensor expansion module and the board in combination, can complete a simple circuit design, to "plug and play".

2. **Types of Buzzer**

   - **Electromechanical:** Early devices were based on an electromechanical system identical to an electric bell without the metal gong. Similarly, a relay maybe connected to interrupt its own actuating current, causing the contacts to buzz. Often these units were anchored to a wall or ceiling to use it as a sounding board. The word "buzzer" comes from the rasping noise that electromechanical buzzers made.

Figure 1.20: Electromechanical Buzzer

   - **Mechanical:** A joy buzzer is an example of a purely mechanical buzzer and they require drivers. Other examples of them are doorbells.

Figure 1.21: Mechanical Buzzer

- **Piezoelectric:** driven by an oscillating electronic circuit or other audio signal source, driven with a "piezoelectric audio amplifier". Sounds commonly used to indicate that a button has been pressed are a click, a ring or a beep.

Figure 1.22: Piezoelectric Buzzer

3. **Pin Configuration**

Pin Number	Pin Name	Description
1	Positive	Identified by (+) symbol or longer terminal lead. Can be powered by 6V DC
2	Negative	Identified by short terminal lead. Typically connected to the ground of the circuit

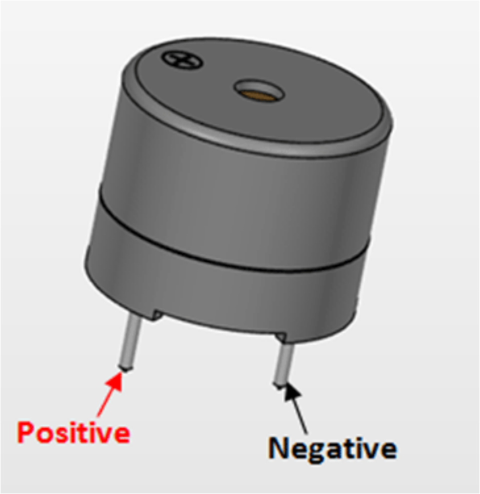


Figure 1.23: Buzzer Pinout

4. **Features and Specifications**

   »Rated Voltage: 6V DC
   »Operating Voltage: 4-8V DC
   »Rated Current: <=30mA
   »Sound Output at 10cm: >=85dB
   »Sound Type: Continous Beep
   »Resonant Frequency: 2300 Hz
   »Operating Temperature: $-25°C$ to $80°C$
   »Storage Temperature: $-30°C$ to $85°C$
   »Weight: 2g
   »Small and neat sealed package
   »Breadboard and Perf board friendly

5. **Applications**

   - Alarming Circuits, where the user has to be alarmed about something
   - Communication equipments
   - Automobile electronics
   - Portable equipments, due to its compact size

Chapter 2

CONTENT AND RESULT

2.1 CONTENT

2.1.1 Block Diagram and Principles of Operation

2.1.1.1 Block Diagram

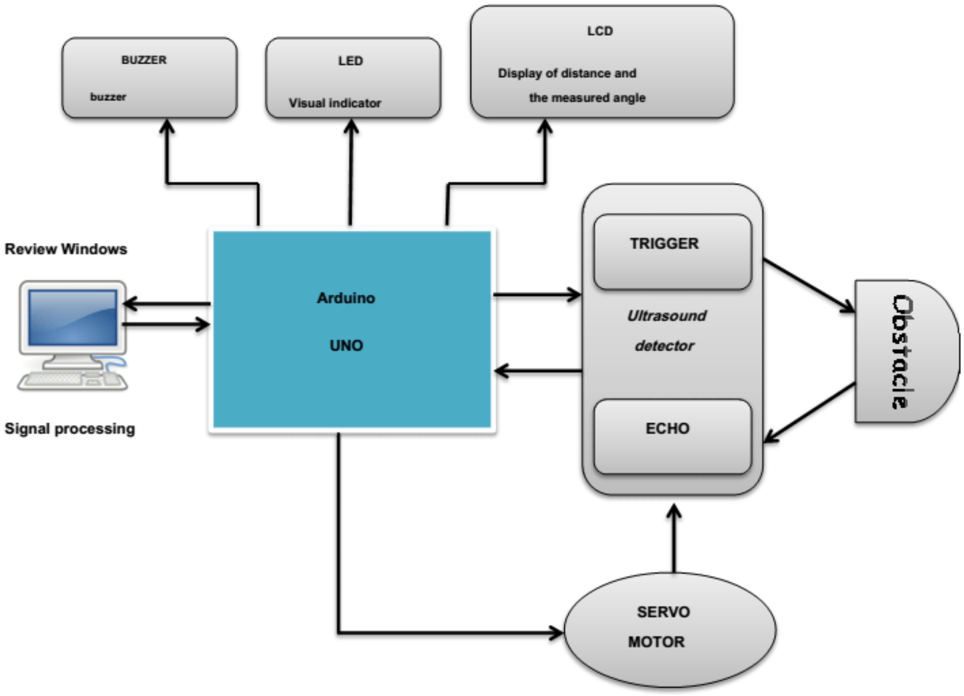


Figure 2.1: Arduino Radar Using Ultrasonic Sensor

This diagram tell that Arduino is the interfaced with ultrasonic sensor. When the object is detected, the signal is given to the Arduino board which will below budget LED and LCD. That why mounted on ultrasonic sensors that will be rotating.

2.1.1.2 Working of the project

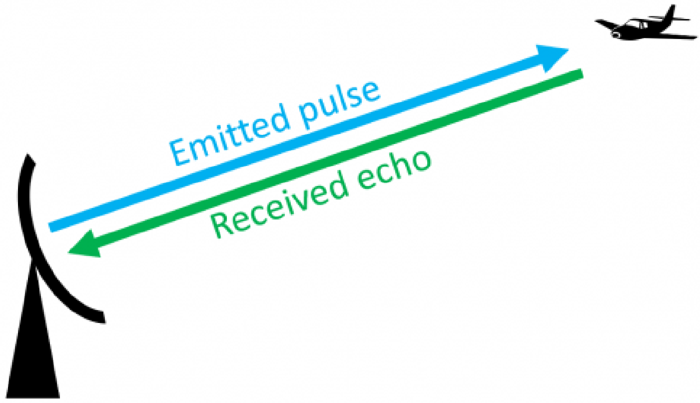


Figure 2.2: PSR principle of operation

Before we learn about how to this project working. We should find out how to conventional radar work.

In Aircraft Traffic Controlled, we have PSR (Primary Surveillance Radar), the antenna rotates (usually at 5-12rpm) emits a pulse of radiio wave. Upon reaching an aircraft (or other object) the wave is reflected and some of the energy is returned to the atenna.

The output data uses polar coordinate system - it provides range and bearing of the targets found in respect of the atenna position. Note that the range is the slant distance from the atenna and not the horizontal distance.

The range is determined by the time difference of the emitted and received pulse (the speed of propagation is the speed of the light) and the bearing is obtained from the atenna azimuth. The rotation speed of the atenna is usually between 5 and 12 rpm.

The antenna radiation pattern is a narrow beam when seen from above and, with some approximation, can be considered as a trapezium if seen from the side.

For the conventional radar, here is a summary of how radar works:

1. Magnetron generates high-frequency radio waves.
2. Duplexer switches magentron through the antenna.
3. Antenna acts as transmitter, sending narrow beam of radio waves through the air.
4. Radio waves hit enemy airplane and reflect back.
5. Atenna picks up reflected waves during a break between transmissions. Note that the same antenna acts as both transmitter and receiver, alternately sending out radio waves and receiving them.
6. Duplexer switches antenna through to receiver unit.
7. Computer in receiver unit processes reflected waves and draws them on TV screen.
8. Enemy plane shows up on TV radar display with any other nearby targets.

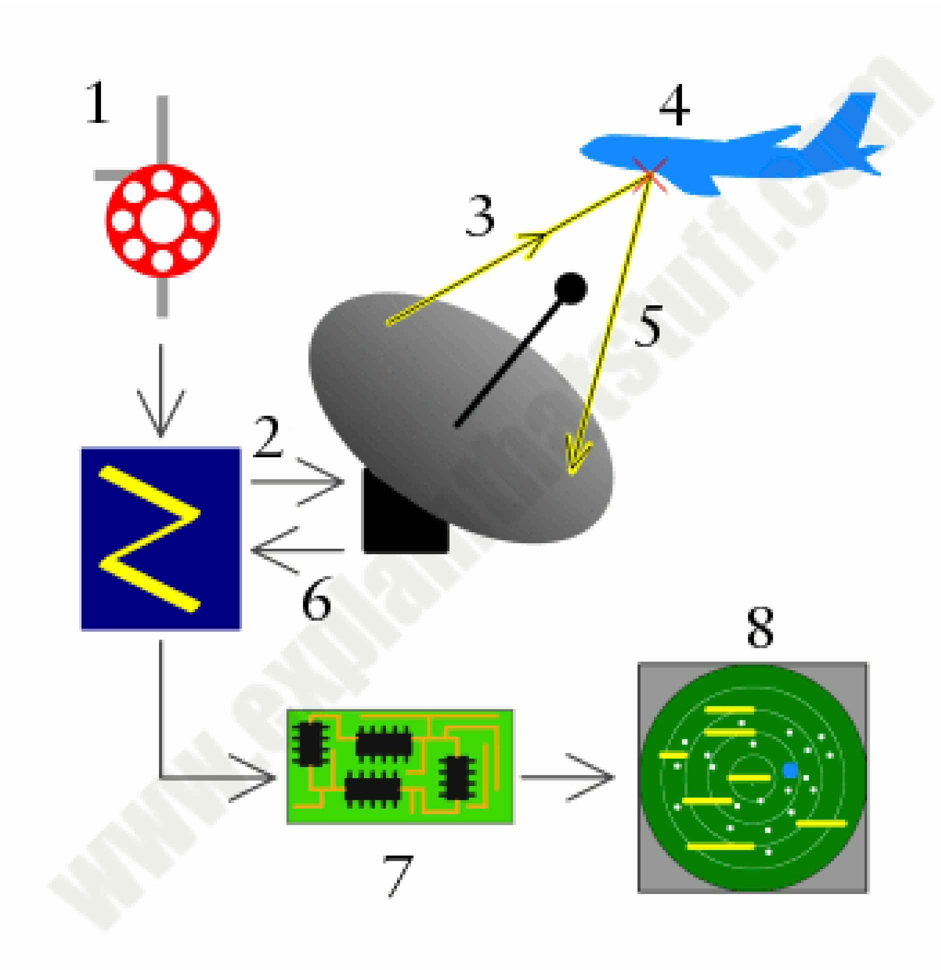


Figure 2.3: How a radar works

2.2 CIRCUIT DIAGRAM OF PROJECT

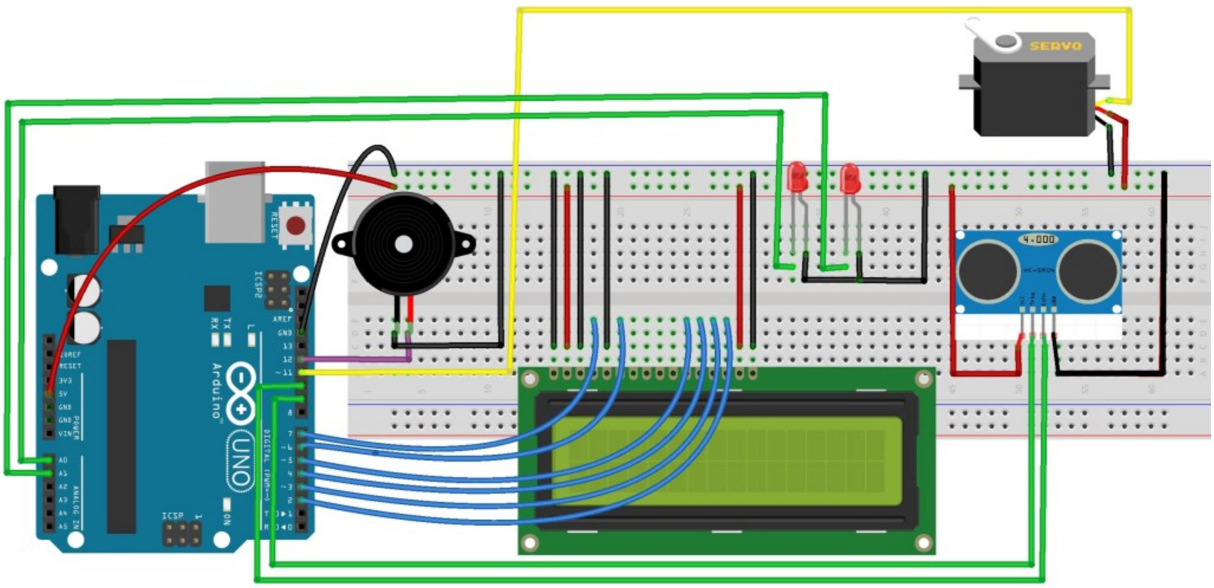


Figure 2.4: Circuit Diagram

Chapter 3

APPENDIX

```
#include <Servo.h>
#include <LiquidCrystal.h>
#include <Wire.h>
```

Listing 3.1: Insert Library

```
Servo myServo;
LiquidCrystal lcd(7, 6, 5, 4, 3, 2); // Creates an LCD object. Parameters: (rs, enable
    , d4, d5, d6, d7)
```

Listing 3.2: Create Variable and LCD object

```
#define TRIGPIN 9
#define ECHOPIN 10
#define MOTEUR 11
#define BUZZER 12
#define LEDPIN1 13
#define LEDPIN2 14

float distanceCm, DistanceSec, duration;
```

Listing 3.3: Define the Variables with the Value of PIN

```
void configPin();
void leftToRight();
void rightToLeft();
void configSensor();
void detectThings();
void displayLcd(uint8_t distance, int pos);
```

Listing 3.4: Call the Funtions

```
void configPin(){
    pinMode(TRIGPIN, OUTPUT);
    pinMode(ECHOPIN, INPUT);
    pinMode(BUZZER, OUTPUT);
    pinMode(LEDPIN1, OUTPUT);
    pinMode(LEDPIN2, OUTPUT);
}
```

Listing 3.5: Define "configPin()" Function

```
void setup(){
    myServo.attach(MOTEUR); //Attach the servo motor to pin 11
    lcd.begin(16,2); //Initialize the lcd interface with their Size
    configPin();
    DistanceSec = 400;
}
```

Listing 3.6: Setup Funtion

```
void leftToRight(){
    for (int pos=0; pos <= 180; pos++){ //Go from 0 to 180 degrees.
        myServo.write(pos); //servo programming to get to the position(pos).
        configSensor();
        detectThings();
    }
}
void rightToLeft(){
    for (int pos = 180; pos >= 0; pos--) { //goes from 180 to 0 degree
        myServo.write(pos); //
        configSensor();
        detectThings();
    }
}
```

Listing 3.7: Define leftToRight and rightToLeft Function

```
void configSensor(){
    digitalWrite(TRIGPIN, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIGPIN, HIGH); //send a pulse of 10 microseconds.
    delayMicroseconds(10);
    digitalWrite(TRIGPIN, LOW);
}
void detectThings(){
    duration = pulseIn(ECHOPIN, HIGH);
    distanceCm= duration*0.034/2;
    if (distanceCm <= DistanceSec){
        if(distanceCm <= DistanceSec/2){
            tone(BUZZER, 10); // Send 1KHz sound signal...
            digitalWrite(LEDPIN1, LOW);
            digitalWrite(LEDPIN2, HIGH);
            delay(700);
            noTone(BUZZER); // Stop sound...
            displayLcd(distanceCm);
        }
        else{
            digitalWrite(BUZZER, HIGH);
            digitalWrite(LEDPIN2, LOW);
            digitalWrite(LEDPIN1, HIGH);
            delay(100);
            digitalWrite(BUZZER, LOW);
            displayLcd(distanceCm);
        }
    }
    else{
        digitalWrite(BUZZER, LOW);
        digitalWrite(LEDPIN1, LOW);
        digitalWrite(LEDPIN2, LOW);
    }
    displayLcd(distanceCm);
    delay(100); // wait 100ms for the servo to find its position
}
```

Listing 3.8: Define detectThings and configSensor Function

```
void displayLcd(uint8_t distance, int pos){
    lcd.setCursor(0,0); // Position the cursor at 0.0
    lcd.print("Distance: "); // Print "Distance" sur LCD
    lcd.print(distance); // Print the distance to LCD
    lcd.print(" cm "); // Printe the unit to LCD
    delay(10);
    lcd.setCursor(0,1);
    lcd.print("Angle : ");
    lcd.print(pos);
    lcd.print(" deg ");
}
```

Listing 3.9: Define displayLCD Funtion

Chapter 4

CONCLUSIONS AND RECOMMENDATIONS

4.1 CONCLUDES

4.1.1 Advantages

- The project works well.
- The sensor works well, from which it scan objects and detect them, then display them on the LCD screen.
- The circuit diagram and wiring of this project aren't too complicated to be able to do.

4.1.2 Defects

- Too much wire to connect all the component.
- The operating range of the sensor does not go too far, less than 20 cm.
- The ingredients of this project are quite expensive.

4.2 RECOMMENDATIONS

In this project, i think i should be broader and more advanced than the present one, looking for ways to overcome the cumbersome wiring in this project. Besides, myself also need to be more careful and focused when implementing the project.

Chapter 5

THE REFERENCES

- Do Huu Toan, June 26th 2016, "*Che tao radar voi Arduino*".
- Mr.Alam, June 14th 2018, "*Arduino RADAR Model using Ultrasonic Sensor for Detection and Ranging*".
- Chris Woodford, November 14th 2019, "*Radar*".
- Skybrary, December 30th 2019, "*Primary Surveillance Radar (PSR)*".
- Shambaditya Mukherjee, December 12nd 2018, "*Arduino Radar Sensor: Working Principle, Advantages, DIsadvantages*".
- RAVI, June 14th 2018, "*Arduino Radar Project*".
- Adnan Aqeel, June 21st 2018, "*Introduction to Arduino Uno*".
- Sciencebuddies.org, "*Introduction to Servo Motors*".