M2-EmbSys

Module 2- Overview of embedded software, design, development processes and standards.

REPORT:

PROJECT TITLE: DISTANCE MEASUREMENT METER.

Submitted by: Sujata Naikar

PS:99007669

Table of content.

1.Abstract and Introduction.

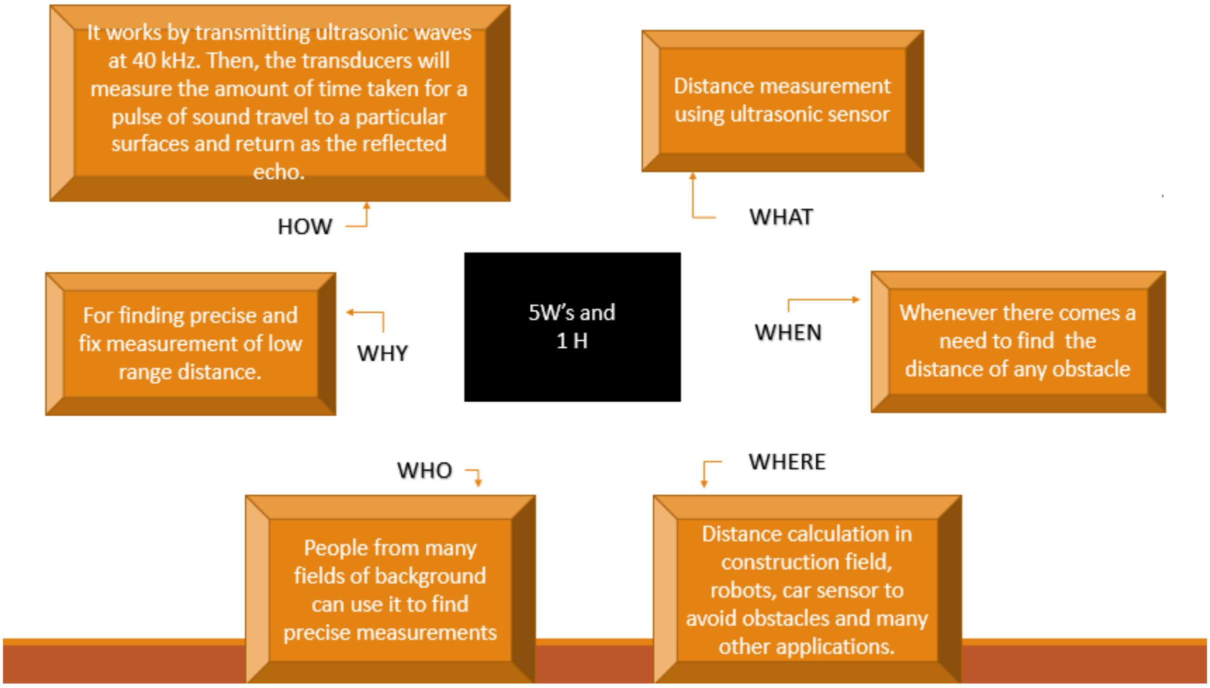
1.1 ABSTRACT.

Nowadays, we have some difficulties in obtaining the distance that we want to measure. Even though, measuring tape is an easy option, but this kind of tool will have a limitation of manual error.. Before this, engineers have produced a range finder module but in the end, they find out the module have many disadvantages like limitation for distance, different result for different coloured obstacles, and need a calibration for every time before starts using it. Manual distance measuring is always done at the expense of human error. Precise and fix measurement of low range distance, is the main objective for this project. This device can measure distance in the range of 2cm-400cm with an accuracy of 3mm.This project is used to measure the distance by using ultrasonic sensors. It works by transmitting ultrasonic waves at 40 kHz. Then, the transducers will measure the amount of time taken for a pulse of sound travel to a particular surfaces and return as the reflected echo. After that, the circuit that have been programmed with AT mega microcontroller will calculate the distance based on the speed of sound at 25°C which an ambient temperature and also the time taken. The distance then will be display on a LCD module.

1.2 INTRODUCTION.

Ultrasonic sensors are great tools to measure distance and detect objects without any actual contact with the physical world. It is used in several applications, like in measuring liquid level, checking proximity and even more popularly in automobiles to assist in self-parking or anti-collision systems. This is an efficient way to measure small distances precisely. In this project, we have used the HC-SR04 Ultrasonic Sensor with microcontroller or can also be interfaced with an Arduino to determine the distance of an obstacle from the sensor. The basic principle of ultrasonic distance measurement is based on ECHO. When sound waves are transmitted in the environment then waves return back to the origin as ECHO after striking on the obstacle.

5W's & 1H and S.W.O.T analysis.



5W's and 1 H

HOW

It works by transmitting ultrasonic waves at 40 kHz. Then, the transducers will measure the amount of time taken for a pulse of sound travel to a particular surfaces and return as the reflected echo.

WHAT

Distance measurement using ultrasonic sensor

WHY

For finding precise and fix measurement of low range distance.

WHEN

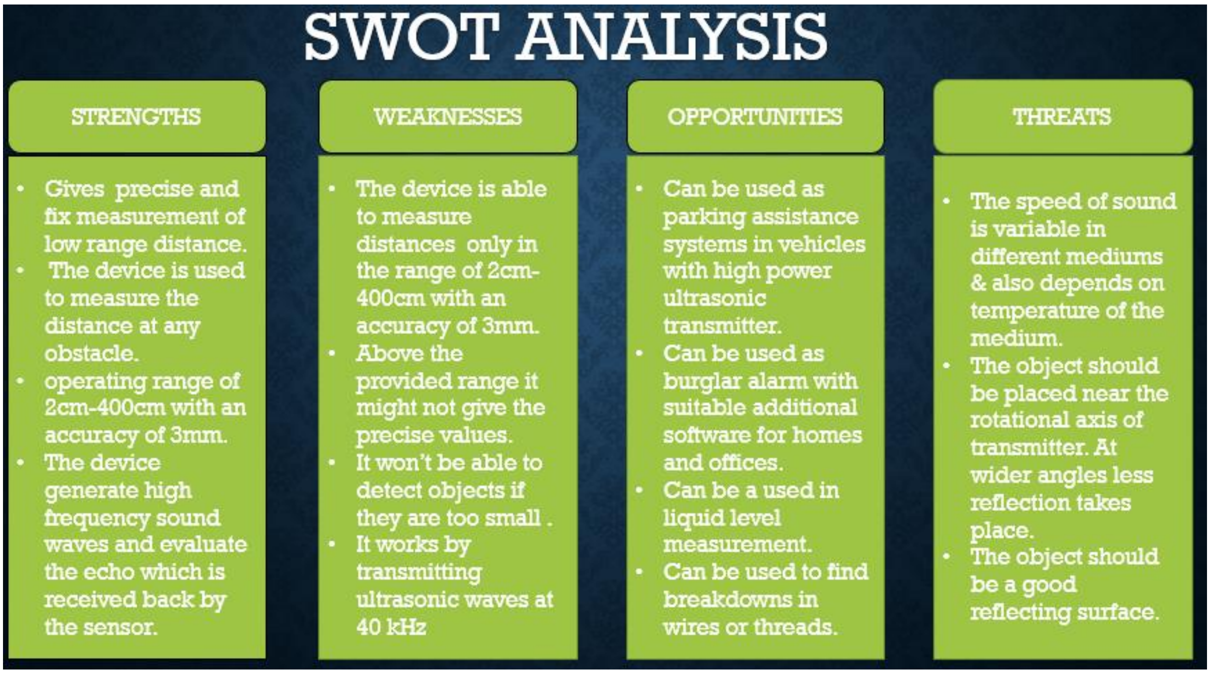
Whenever there comes a need to find the distance of any obstacle

WHO

People from many fields of background can use it to find precise measurements

WHERE

Distance calculation in construction field, robots, car sensor to avoid obstacles and many other applications.



SWOT ANALYSIS

STRENGTHS	WEAKNESSES	OPPORTUNITIES	THREATS
• Gives precise and fix measurement of low range distance. • The device is used to measure the distance at any obstacle. • operating range of 2cm-400cm with an accuracy of 3mm. • The device generate high frequency sound waves and evaluate the echo which is received back by the sensor.	• The device is able to measure distances only in the range of 2cm-400cm with an accuracy of 3mm. • Above the provided range it might not give the precise values. • It won't be able to detect objects if they are too small . • It works by transmitting ultrasonic waves at 40 kHz	• Can be used as parking assistance systems in vehicles with high power ultrasonic transmitter. • Can be used as burglar alarm with suitable additional software for homes and offices. • Can be a used in liquid level measurement. • Can be used to find breakdowns in wires or threads.	• The speed of sound is variable in different mediums & also depends on temperature of the medium. • The object should be placed near the rotational axis of transmitter. At wider angles less reflection takes place. • The object should be a good reflecting surface.

2.Requirements.

There are mainly two types of requirements;

1. High level requirements.

2. Low level requirements.

- Though both the types of requirements are more or less depended on each other while functioning, according to the need and usage they are termed as high and low level requirements.

2.1 High level requirements.

Id	High level requirements.
HLR1	The system shall be used to find precise and fix measurement of low range distance.
HLR2	The system shall be able to measure a distance of any obstacle.
HLR3	The system shall be able to operate at the range of 2cm-400cm with an accuracy of 3mm.

2.2 Low level requirements.

Id	Low level requirements.
LLR1.1(for HLR1)	Ultrasonic sensors generate high frequency sound waves and evaluate the echo which is received back by the sensor.
LLR1.2(for HLR1)	calculates the distance depending upon the time taken by the echo signal to travel back after reflecting from the desired target.

LLR2.1(For HLR2)	The size of the object should be of the order of wavelength of signal for it to be detected. So object should not be very small.
LLR2.2(For HLR2)	The object should be placed near the rotational axis of transmitter. At wider angles less reflection takes place.
LLR2.3(For HLR2)	The object should be a good reflecting surface.
LLR3.1(For HLR3)	Sensors calculate the time interval between sending the signal and receiving the echo to determine the distance to an object in the particular range of distance.
LLR3.2(For HLR3)	Make sure about provided range of operation because above the range of the specification provided it might won't provide accurate values.

3.Components

The main components are,

- Arduino Uno.
- Ultrasonic sensor Module
- 16x2 LCD.
- Scale
- Bread board
- 5 volt battery
- Connecting wires

- Arduino Uno: Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online.

- ❖ Ultrasonic sensor Module:(The ultrasonic Sensor is considered to be a digital sensor because the microcontroller reads a digital signal from it.)

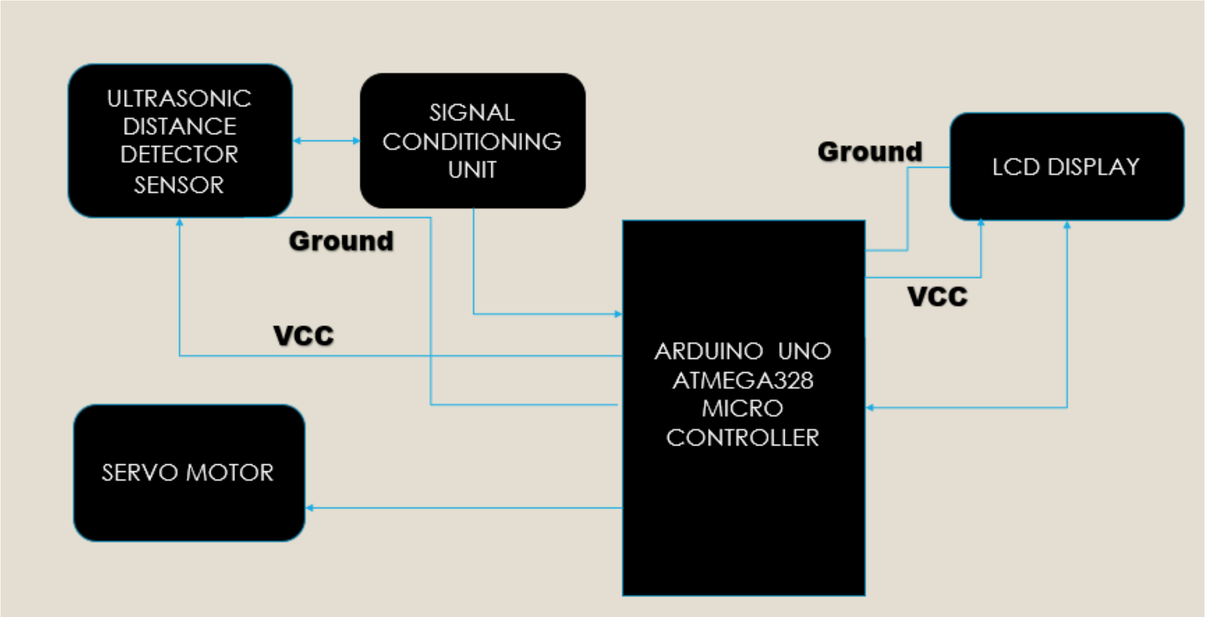
There are many types of Arduino distance sensors, but in this project we have used the HC-SR04 to measure distance in range of 2cm-400cm with an accuracy of 3mm. The sensor module consists of an ultrasonic transmitter, receiver and control circuit. The working principle of ultrasonic sensor is as follows:

- High level signal is sent for 10us using Trigger.
- The module sends eight 40 KHz signals automatically, and then detects whether pulse is received or not.
- If the signal is received, then it is through high level. The time of high duration is the time gap between sending and receiving the signal.

- ❖ LCD(Actuator):This display offers a way to interface your microcontroller to the outside world. It's one of the main output devices in your kit.

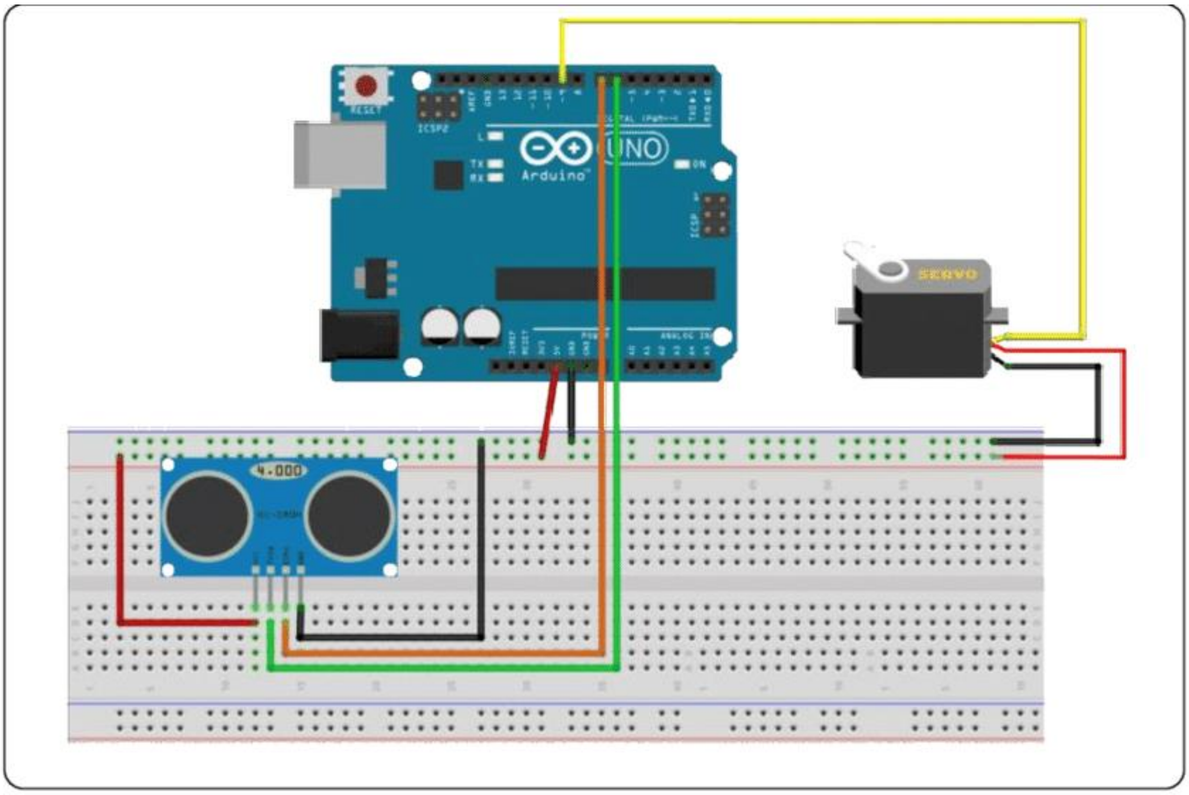
4. Architecture.

4.1 PRIMARY BLOCK DIAGRAM..


ULTRASONIC DISTANCE DETECTOR SENSOR
SIGNAL CONDITIONING UNIT
Ground
LCD DISPLAY
Ground
VCC
VCC
ARDUINO UNO ATMEGA328 MICRO CONTROLLER
SERVO MOTOR

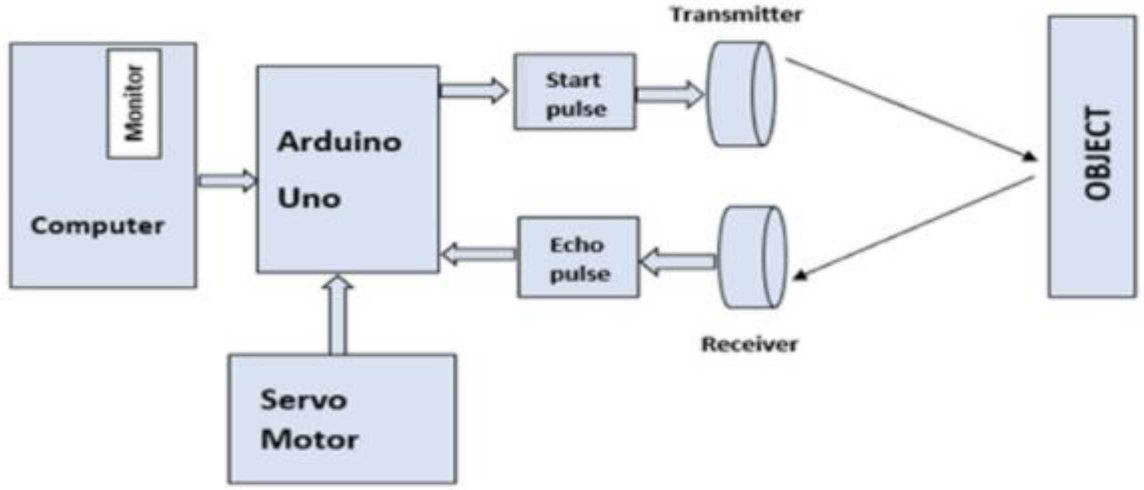
* Fig 4.1.1The above figure gives the overall idea about how the system is arranged in the form of block diagram.

In this project distance of the object is measured through ultrasonic distance sensor and the sensor output is connected to signal conditioning unit and after that it is processed through Arduino microcontroller. The measured results are displayed in liquid crystal display. The results are transferred to personal computer. The sensor is attached to servo motor to find the polar distance around the sensor upto 1800 rotations. This application is also used to find the obstacles detection and the exact distance can also be obtained. The measured distance is displayed on the LCD display.



* Fig 4.1.2The above figure shows how each of the components are connected to each other.(Hardware Representation)

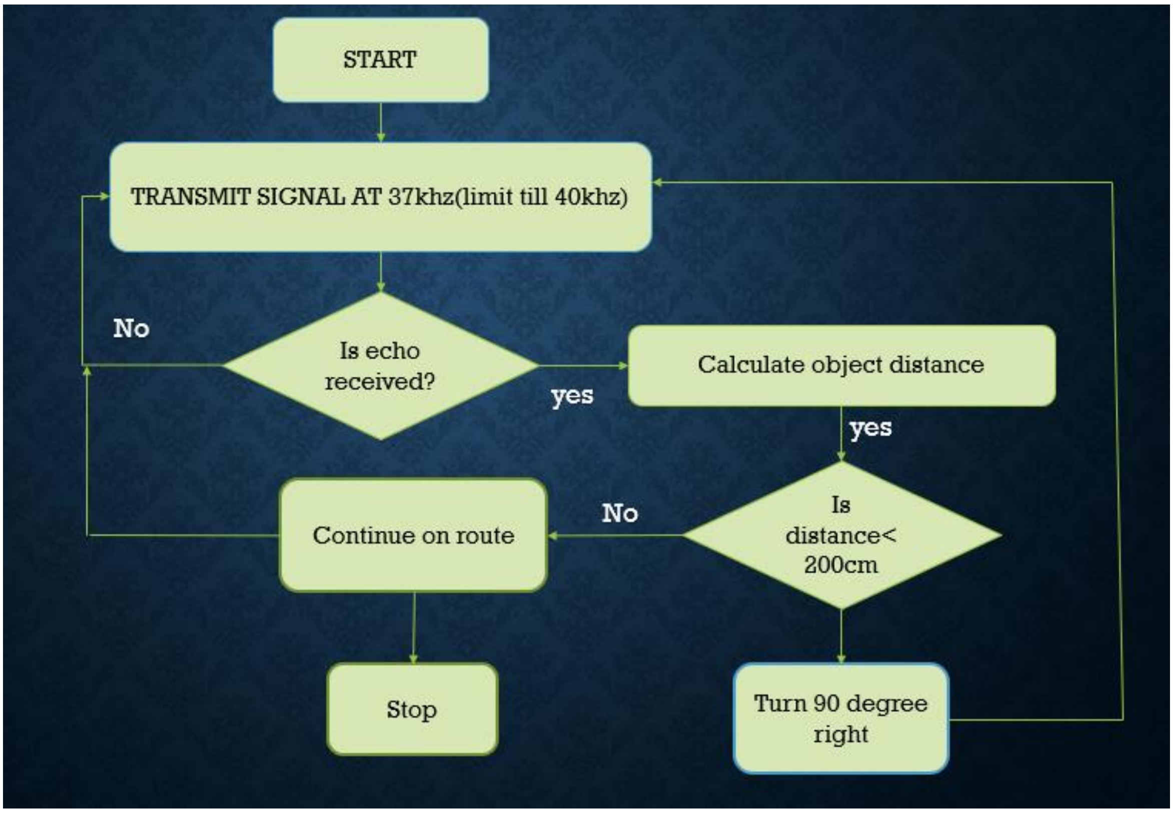
4.2 Structural diagrams.



* Fig 4.1.3The above figure also explains about how the blocks are arranged with respect to each other.

Block diagram of the working principle of an ultrasonic radar As shown in the above Fig, Arduino UNO receives power supply from the computer. Then microcontroller sends signals to the other two components: the ultrasonic sensor and servo motor. The feedback from these components is sent back to the microcontroller. The ultrasonic sensor has a transmitter to send out the ultrasound and a receiver to catch the reflected sound waves. The total travelling time of this sound wave is the distance.

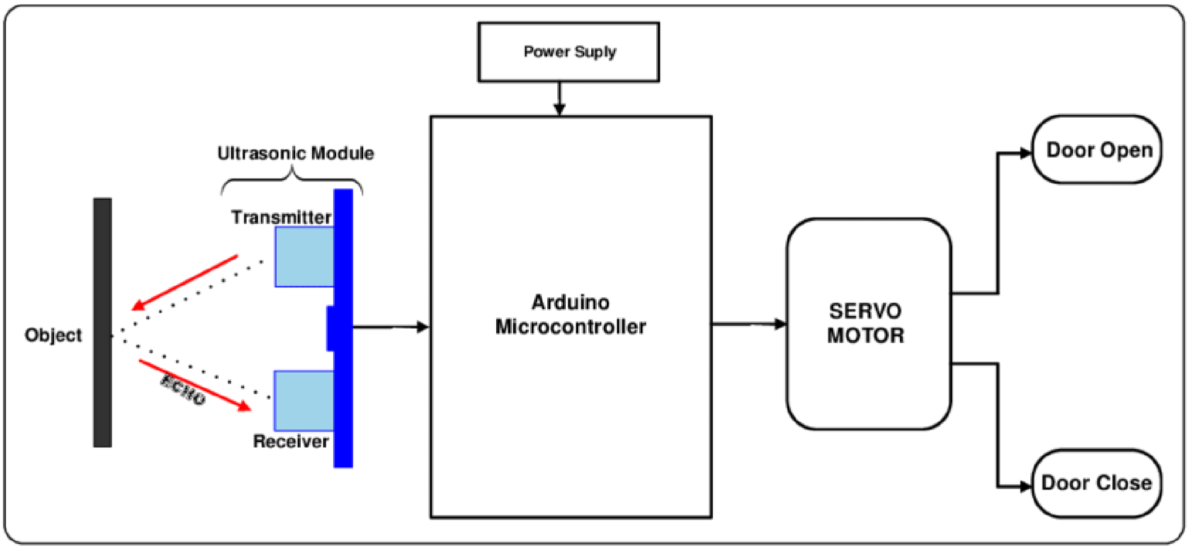
4.3 Flowcharts.



* fig 4.3.1 Flowchart of the Ultrasonic Distance meter.

The ultrasonic sensor emits a high-frequency sound pulse and calculates the distance depending upon the time taken by the echo signal to travel back after reflecting from the desired target. The speed of sound is 341 meters per second in air. After the distance is calculated, it will be displayed on the LCD display

4.4 Sub-systems.



* Fig 4.4.1 Schematic diagram of Ultrasonic Distance Meter.

4.5 Sensors.

1.Ultrasonic sensor :(The ultrasonic Sensor is considered to be a digital sensor because the microcontroller reads a digital signal from it.)

The sensor module used here is HC-SR04, which is a non-contact Ultrasonic measurement device. This small module is capable of measuring the distance within the range of 2cm – 400cm. It is a really accurate sensor, it can measure upto 3mm.

The sensor consists of Ultrasonic transmitter and Ultrasonic receiver. The working principle is really simple. Firstly, the IO trigger sends high level signal for 10µs. Then the sensor module sends eight 40 kHz cycle of ultrasonic sound and detect if the pulse signal is received back or not. And if the signal is received, through high level, the time duration by which the IO trigger stays high is the time from sending to receiving the signal.

Distance = (Time of High Level x Velocity of Sound in Air (340M/s))/2

It can can be used to detect the door's position. For example, in refrigerators, a reed sensor is mounted to the frame of the appliance and a permanent magnet actuator is mounted to the door.

4.6 Actuators.

1. LCD(Liquid Crystal Display) :

LCD is a type of flat panel display which uses liquid crystals in its primary form of operation.Here we have used (hd444780) lcd display for displayong the distance that is measured.

2. Servo motor :

A servomotor is a rotary actuator / linear actuator that allows for precise control of angular or linear 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 used in applications such as robotics, CNC machinery or automated manufacturing. A servomotor is a closed-loop servomechanism that uses position feedback to control its motion and final position. The motor is attached with the sensor to find the distance range around the sensor for 1800 rotations. The motor is controlled and interfaced with Arduino microcontroller to rotate in clockwise and anticlockwise.

Working:

* The Ultra Sonic sensor works as a burst signal is transmitted for short duration (is emitted) by the emitter. After that there will be a silent period. This period is actually called “response time” and is the time waiting for reflected waves. The acoustic emitted signal may find an obstacle or not. If an obstacle is found, the acoustic signal will be bounced back from the obstacle. This backbounced signal is called “echo”. The echo is received by the receiving transducer and is converted into electrical signal. Usually this signal is amplified, filtered and can be converted into digital form. Using the elapsed time between transmission and reception, the distance between the Ultra Sonic system and obstacle/object can be calculated. The servo motor has programmed to sweep the shaft in to and fro motion. The servo moves the arm in a range of 180 degrees clockwise and anti-clockwise direction. With this sweeping movement of the servo, the ultrasonic sensor placed on the servo arm scans the entire area comes in front of the sensor. Here then the measured instantaneous distance values are printed on the

serial monitor. In an ideal case, the ultrasonic sensor will return a value for each degree of the servo. That is a complete clockwise and anticlockwise movement of the servo shaft returns a total of 360 values.

5. Implementation

5.1. LCD Display:

```
#include <LiquidCrystal.h>

//constants for the number of rows and columns in the LCD
//In my example, I use a 20x4 LCD, thus it has 4 rows and 20 columns
//If you use another LCD, type in the appropriate rows and columns below
const int numRows= 4;
const int numCols= 20;

//initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup()
{
lcd.begin(numCols, numRows);
lcd.print("Hello there");
}

void loop()
{
//set the cursor to column 0, line 1
//displays the number of seconds which has elapsed since code is uploaded
lcd.setCursor(0,1);
delay(1000);
lcd.print(millis()/1000);
lcd.print(" seconds elapsed");
}
```

5.2 Ultrasonic sensor

```
#include <LiquidCrystal.h>

//constants for the number of rows and columns in the LCD
//In my example, I use a 20x4 LCD, thus it has 4 rows and 20 columns
//If you use another LCD, type in the appropriate rows and columns below
const int numRows= 4;
const int numCols= 20;
#define echoPin 11 // attach pin D2 Arduino to pin Echo of HC-SR04
```

```
#define trigPin 13 //attach pin D3 Arduino to pin Trig of HC-SR04

// defines variables
long duration; // variable for the duration of sound wave travel
int distance; // variable for the distance measurement



//initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup()
{
lcd.begin(numCols, numRows);
lcd.print("Distance measurement meter");
pinMode(trigPin, OUTPUT); // Sets the trigPin as an OUTPUT
  pinMode(echoPin, INPUT); // Sets the echoPin as an INPUT
  Serial.begin(9600); // // Serial Communication is starting with 9600 of
baudrate speed
  Serial.println("Ultrasonic Sensor HC-SR04 Test"); // print some text in
Serial Monitor
  Serial.println("with Arduino UNO R3");
}

void loop()
{
//set the cursor to column 0, line 1
//displays the number of seconds which has elapsed since code is uploaded
lcd.setCursor(0,1);
delay(1000);
lcd.print(millis()/1000);
lcd.print(" seconds elapsed");
// Clears the trigPin condition
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin HIGH (ACTIVE) for 10 microseconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  distance = duration * 0.034 / 2; // Speed of sound wave divided by 2 (go and
back)
  // Displays the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
}
```

5.3 Servo motor

```
#include <Servo.h>

Servo myservo;  // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0;    // variable to store the servo position

void setup() {
  myservo.attach(2);  // attaches the servo on pin 2 to the servo object
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in
variable 'pos'
    delay(15);                       // waits 15 ms for the servo to reach the
position
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservo.write(pos);              // tell servo to go to position in
variable 'pos'
    delay(15);                       // waits 15 ms for the servo to reach the
position
  }
}
```

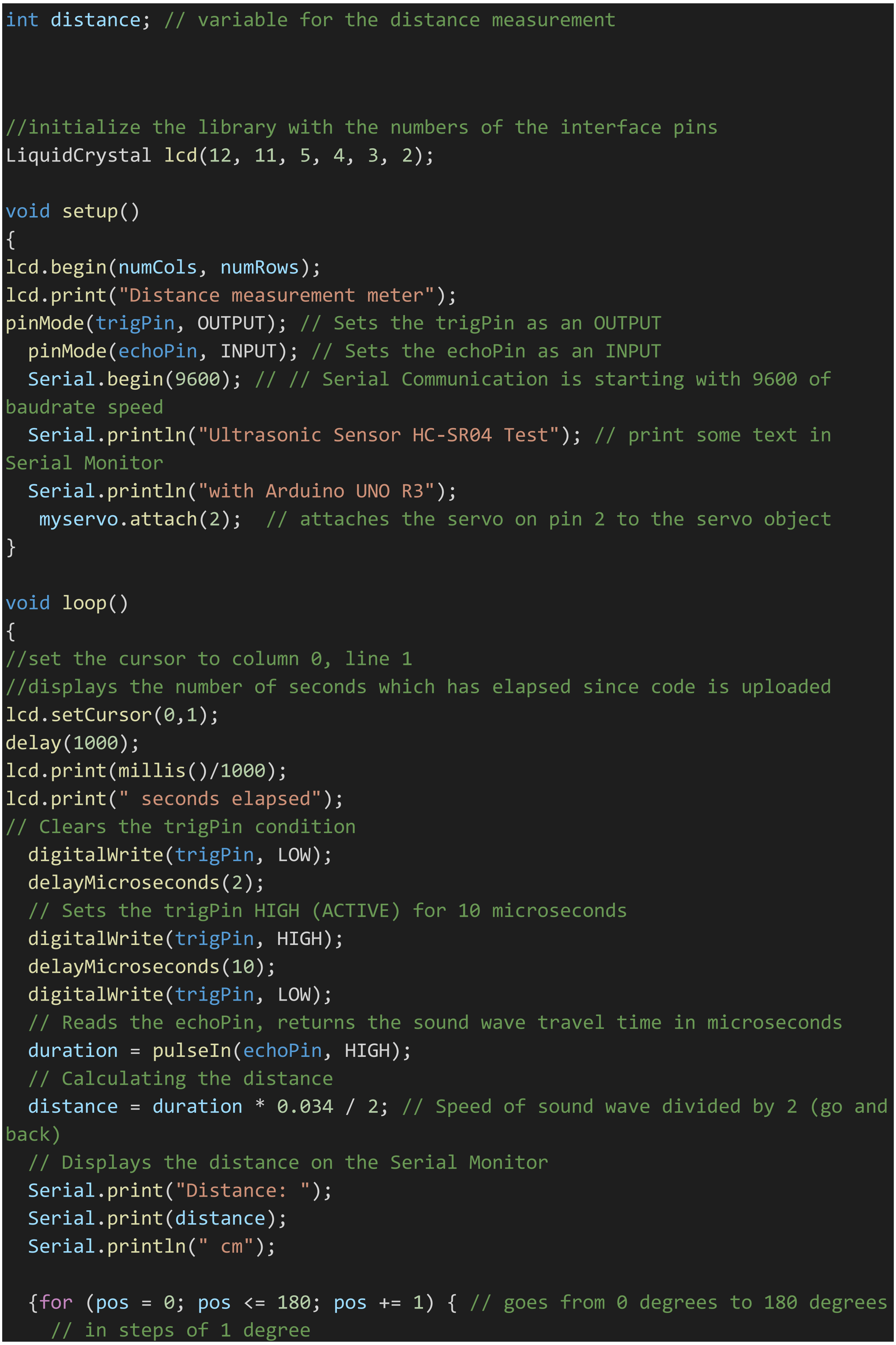
5.4 Complete code.

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

Servo myservo;  // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0;    // variable to store the servo position
//constants for the number of rows and columns in the LCD
//In my example, I use a 20x4 LCD, thus it has 4 rows and 20 columns
//If you use another LCD, type in the appropriate rows and columns below
const int numRows= 4;
const int numCols= 20;
#define echoPin 11 // attach pin D2 Arduino to pin Echo of HC-SR04
#define trigPin 13 //attach pin D3 Arduino to pin Trig of HC-SR04

// defines variables
long duration; // variable for the duration of sound wave travel
```



```
int distance; // variable for the distance measurement



//initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup()
{
lcd.begin(numCols, numRows);
lcd.print("Distance measurement meter");
pinMode(trigPin, OUTPUT); // Sets the trigPin as an OUTPUT
  pinMode(echoPin, INPUT); // Sets the echoPin as an INPUT
  Serial.begin(9600); // // Serial Communication is starting with 9600 of baudrate speed
  Serial.println("Ultrasonic Sensor HC-SR04 Test"); // print some text in Serial Monitor
  Serial.println("with Arduino UNO R3");
   myservo.attach(2);  // attaches the servo on pin 2 to the servo object
}

void loop()
{
//set the cursor to column 0, line 1
//displays the number of seconds which has elapsed since code is uploaded
lcd.setCursor(0,1);
delay(1000);
lcd.print(millis()/1000);
lcd.print(" seconds elapsed");
// Clears the trigPin condition
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin HIGH (ACTIVE) for 10 microseconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  distance = duration * 0.034 / 2; // Speed of sound wave divided by 2 (go and back)
  // Displays the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");

  {for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in
variable 'pos'
    delay(15);                       // waits 15 ms for the servo to reach the
position
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservo.write(pos);              // tell servo to go to position in
variable 'pos'
    delay(15);                       // waits 15 ms for the servo to reach the
position
  }
  }
}
```

6.Test plan and output.

- Testing

The measurement repeatability and stability over time are important for determining the size of errors and corrections' calculation.

6.1 HIGH LEVEL TEST PLAN/Unit test plan

Test ID	Description	Input	Expected Output	Actual Output	Passed/Not
1	Distance	2cm-400cm	The object is detected	The object is detected	Passed
2	Lcd display	Distance measure	Distance measure	Distance measure	Passed
3	Servo motor	Rotates after detecting any object	Rotates after detecting any object	Rotates after detecting any object	Passed

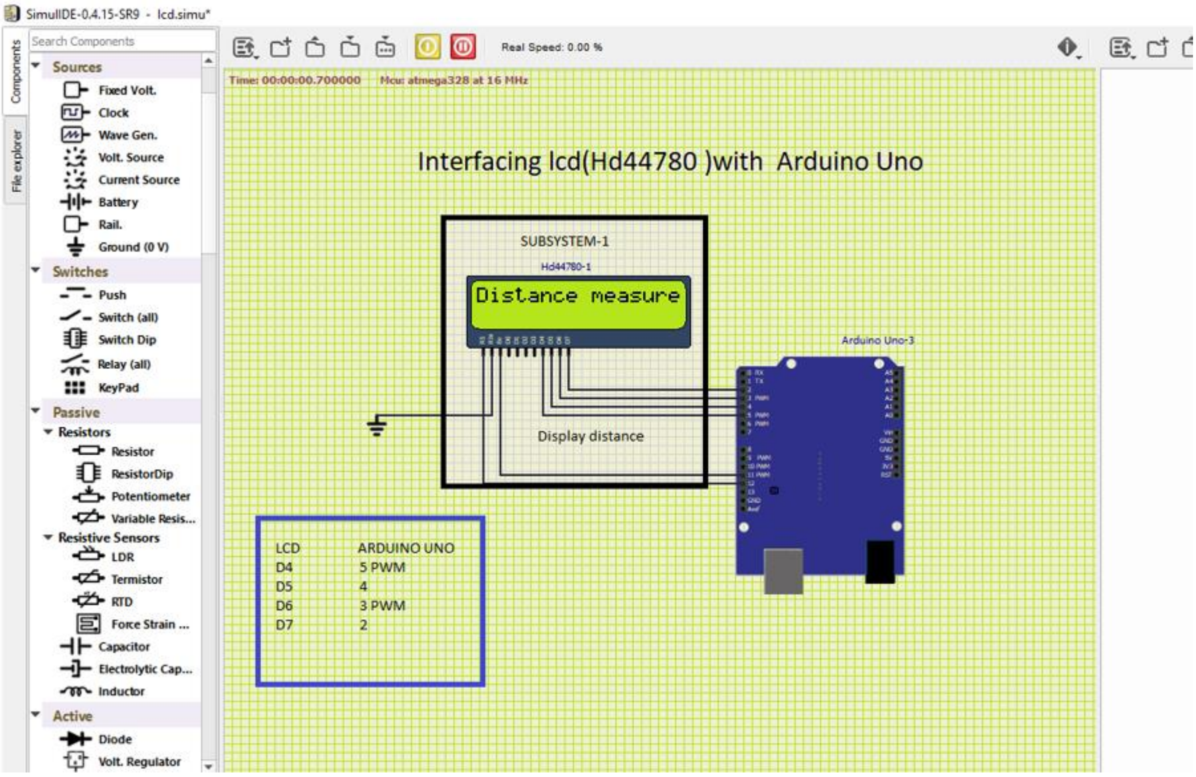
6.2 LOW LEVEL TEST PLAN/Unit test plan

Test ID(for lcd)	Description	Input	Expected Output	Actual output	Passed/not
1	Check for lcd (char)	Distance Measure	Distance Measure	Distance Measure	Passed
2	Check for lcd (String)	Circuit digest	Circuit digest	Circuit digest	Passed

Test ID(for Ultrasonic sensor)	Description	Input	Expected Output	Actual output	Passed/not
1	Check for lcd (char)	Distance cm	Distance cm	Distance cm	Passed
2	Check for lcd (String)	Distance mm	Distance cm	Distance cm	Passed

6.3 OUTPUTS:

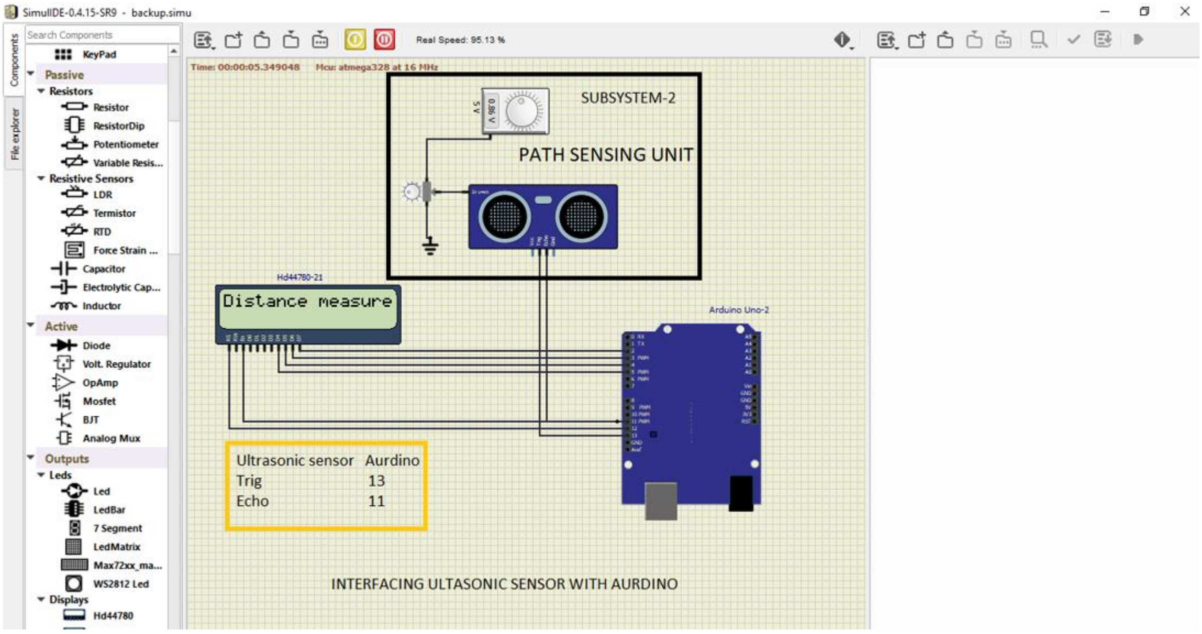
6.3.1 SUBSYSTEM-1:Interfacing LCD WITH AURDINO



Interfacing lcd(Hd44780)with Arduino Uno

LCD	ARDUINO UNO
D4	5 PWM
D5	4
D6	3 PWM
D7	2

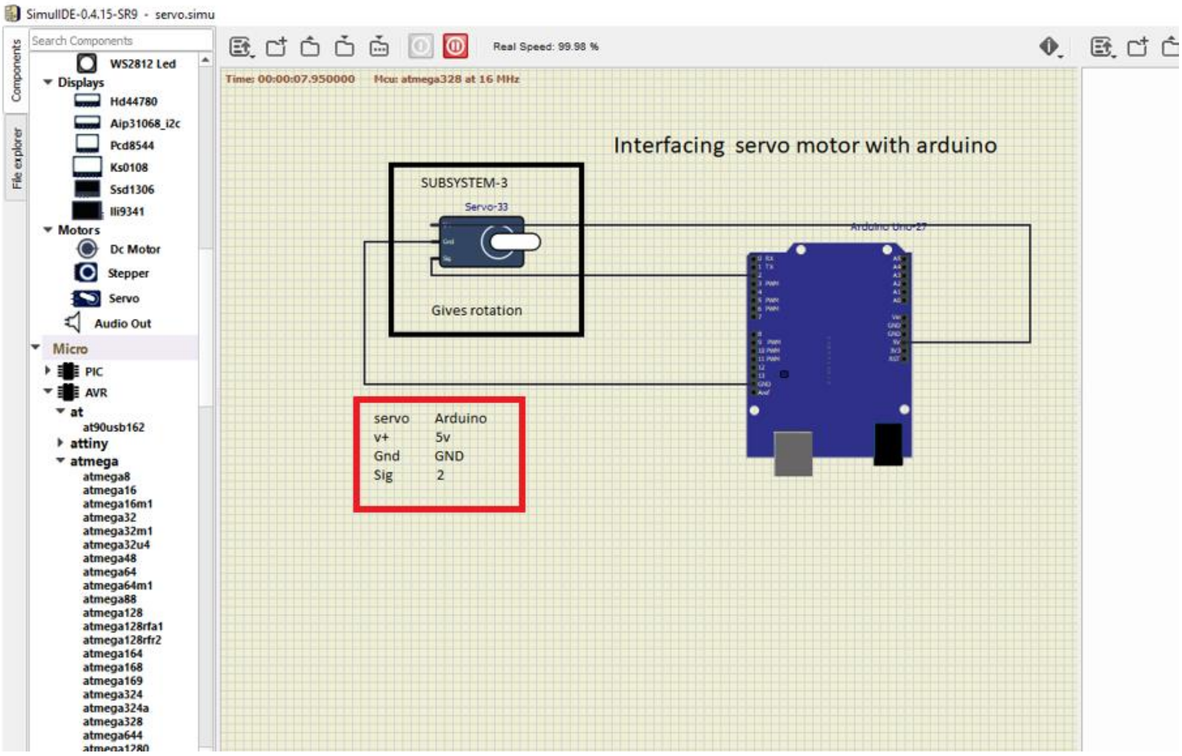
6.3.2 SUBSYSTEM-2:Interfacing ultrasonic sensor with Aurdino.



Ultrasonic sensor	Aurdino
Trig	13
Echo	11

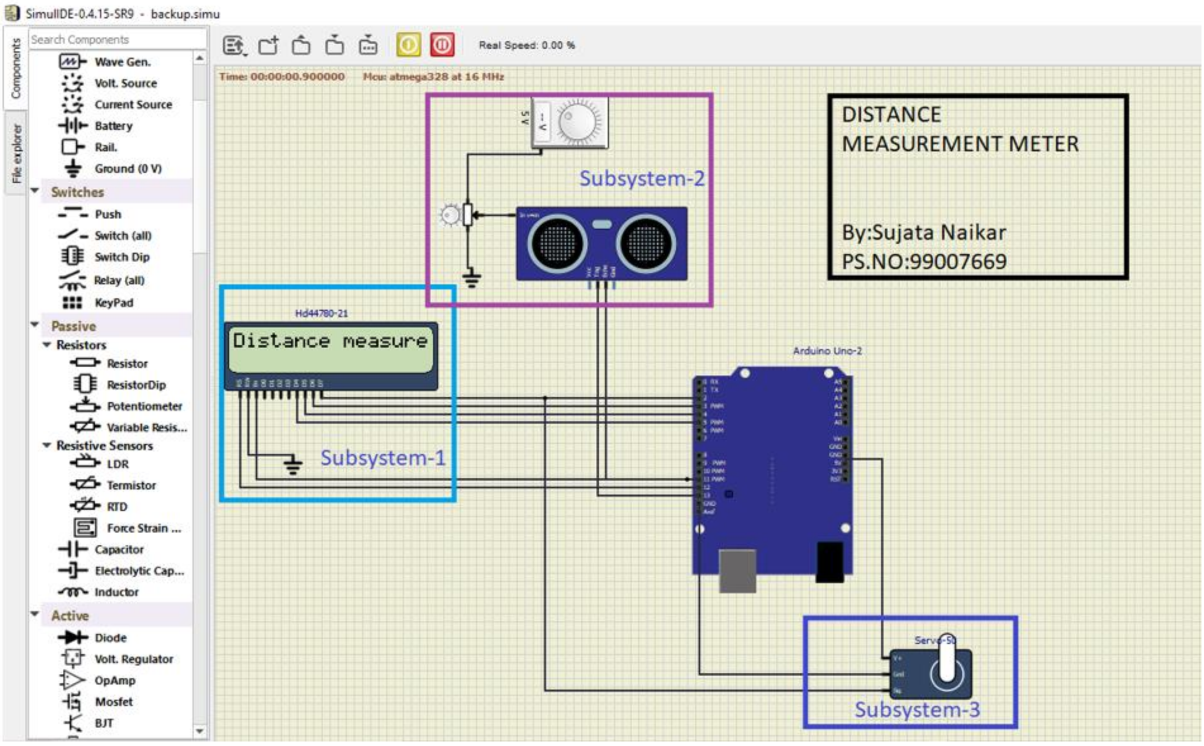
INTERFACING ULTASONIC SENSOR WITH AURDINO

6.3.3 SUBSYSTEM-3:Interfacing Servo motor with Aurdino.



servo	Arduino
v+	5v
Gnd	GND
Sig	2

6.3.4 COMPLETE MODEL OF DISTANCE MEASUREMENT METER



DISTANCE MEASUREMENT METER

By:Sujata Naikar
PS.NO:99007669

7.Applications:

This is a very economic technology and can be used in several other fields as well, few are listed as below:

1. Can be used as parking assistance systems in vehicles with high power ultrasonic transmitter.

2. Can be used as burglar alarm with suitable additional software for homes and offices.

3. Can be a used in liquid level measurement.

4. Can be used to find breakdowns in wires or threads.

8.REFERENCES.

1.https://www.erpublication.org/published_paper/IJETR_APRIL_2014_STE60.pdf

The above link preferred for some objective information.

2. Idea of block diagram is taken from available sources. https://iotdunia.com/distance-measurement/

To know about the connections between components.