Project report

On

“AUTO DISTANCE MEASURING DEVICE”

Project by

SUTHAR NIRMAL HARESHKUMAR

ACKNOWLEDGEMENT

A part from the efforts of me, the success of any project depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project.

I wish to express my deep sense of graduate to my internal guide, **Darshan K. Patel**, Sankalchand Patel College of Engineering for them able guidance and useful suggestions, which helped me in completing the project work, in time.

We would like to express our sincere thanks to our Head of Department **Prof. Kirit. B. Modi.**

Finally, yet importantly, I would like to express heartfelt thanks to my beloved parents for their blessings, my friends for their help and wishes for the completion of this project up to now.

With sincere regards,

Nirmal Suthar

ABSTRACT

"AUTO DISTANCE MEASURIMG DEVICE" is a height measuring device project, will be used to measuring height of any object like building, person, pen etc. .This project is prototype .This is mainly used by the civil constructor and civil engineering of city and operated by it's owner. This project provides good and better services to measuring the height. It also provides the distance of two objects. Using this device people can easily understand the measurement. They can carried out easily and handle those device and given perfect measurement to the user.

People can be used at any time and from anywhere. It is reliable and very efficient to use.

INDEX

Table No.	List of Tables	Pg. No.
1.	INTRODUCTION	6
	1.1 PURPOSE OF THE PROJECT	
	1.2 SCOPE OF THE PROJECT	
	1.3 PROPOSED SYSTEM V/S EXISTING SYSTEM	
	TOOLS/PLATFORMS/LANGUAGE USED:-	
.	TOOLS	7
	I. ARDUION UNO	7-8
	II. ULTRASONIC SENSOR	8-9
	III. SERVO MOTER	10-11
	IV. ACCELEROMETER	11-14
	V. JUMP WIRE	14
	VI. LASER LIGHT	14-15

	VII. BREAD BOARD	15
3.	PLATFORMS/LANGUAGE USED:	16
4.	FINAL EXECUTION(LOGIC)	16-17
5.	SCREANSHOTS	18-21
6.	BLOCK DIAGRAM	22
7.	REFERANCE	23
8.	CONCLUSION	23

1. INTRODUCTION:-

Auto Distance Measuring Device (ADMD) is device that will be used to measuring height and calculating distance between two object.
It is User-friendly human machine interface (HMI) for quickly measure the height.

1.1 PURPOSE OF THE PROJECT :-

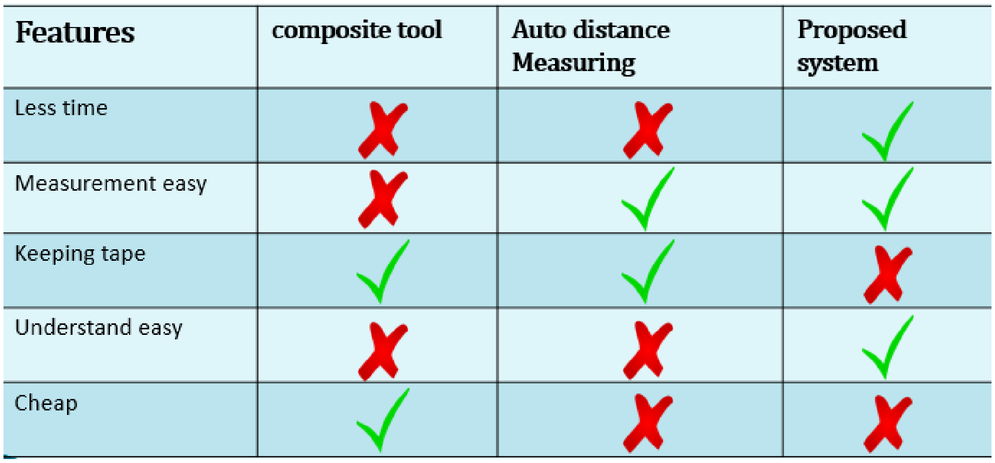
This device can calculating accurate measurement of distance and height. The result is display on screen.

1.2 SCOPE OF THE PROJECT:-

This project is prototype project.

As we are making the height measuring device with some simple mathematical functions, like tangents this device may be further get upgrade so that it can measure the bigger height and may be used by civil engineers.

1.3 PROPOSED SYSTEM V/S EXISTING SYSTEM:-



Features	composite tool	Auto distance Measuring	Proposed system
Less time	✗	✗	✓
Measurement easy	✗	✓	✓
Keeping tape	✓	✓	✗
Understand easy	✗	✗	✓
Cheap	✓	✗	✗

2 TOOLS/PLATFORMS/LANGUAGE USED:-

TOOLS:-

I. Arduino UNO
II. Ultrasonic sensor
III. Servo motor
IV. Accelerometer
V. Jump wire
VI. Laser light
VII. Breadboard

I. ARDUINO UNO :-

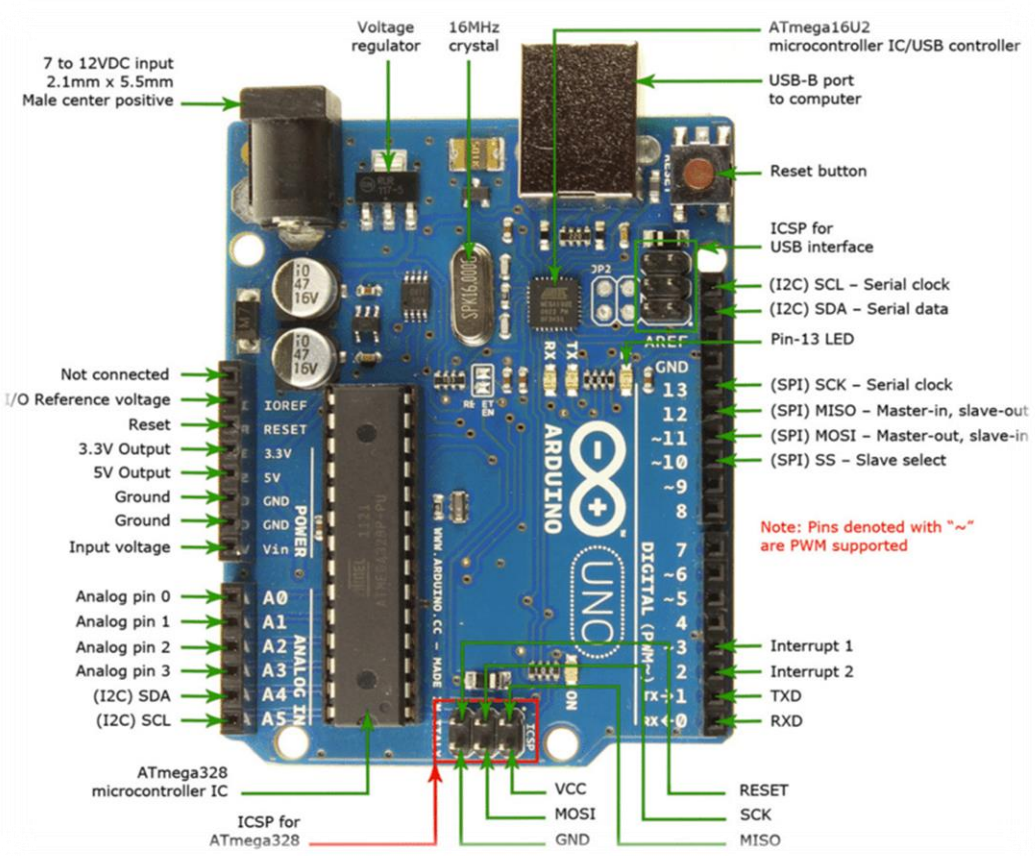
- **What is Arduino?**

  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.

- Arduino is the open source hardware where one can suppose to control the hardwares by programming.

Why Arduino?

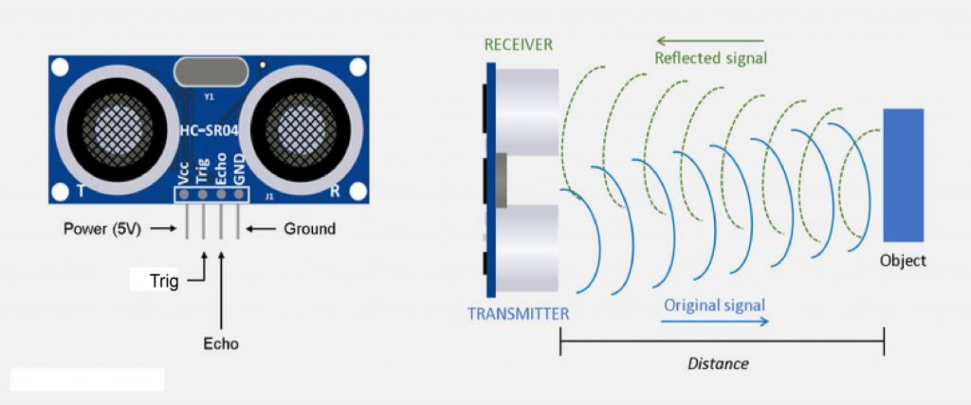
Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things.



- It consist 14 digital pins and 6 analog pins.
- Most Arduino boards consist of an atmel 8-bit AVR microcontroller (ATmega8, ATmega168, ATmega328, ATmega1280, ATmega2560) with varying amounts of flash memory, pins, and features.

II. ULTRASONIC SENSOR :-

An ultrasonic sensor is an electronic device that measures the distance of a 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 (i.e. the sound that humans can hear).



- Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target).

- **Here we used ultrasonic sensorHC-SR04 module.**

- **How to implement ultrasonic sensor in Arduion IDE:-**

```
{
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    distance= duration*0.034/2;

    return distance;
}
```

III. SERVO MOTER:-

Types of Servo Motors are classified into different types based on their application, such as the AC servo motor, and DC servo motor.

- There are three main considerations to evaluate servos motors. First based on their current type – AC or DC, and secondly on the type of Commutation used, whether the motor uses brushes and the third type of consideration is the motors rotating field, the rotor, whether the rotation is synchronous or asynchronous.

- Servos are controlled by sending an electrical pulse of variable width, or **pulse width modulation** (PWM), through the control wire. There is a minimum pulse, a maximum pulse, and a repetition rate.

- A servo motor can usually only turn 90° in either direction for a total of 180° movement. The motor's neutral position is defined as the position where the servo has the same amount of potential rotation in the both the clockwise or counter-clockwise direction.

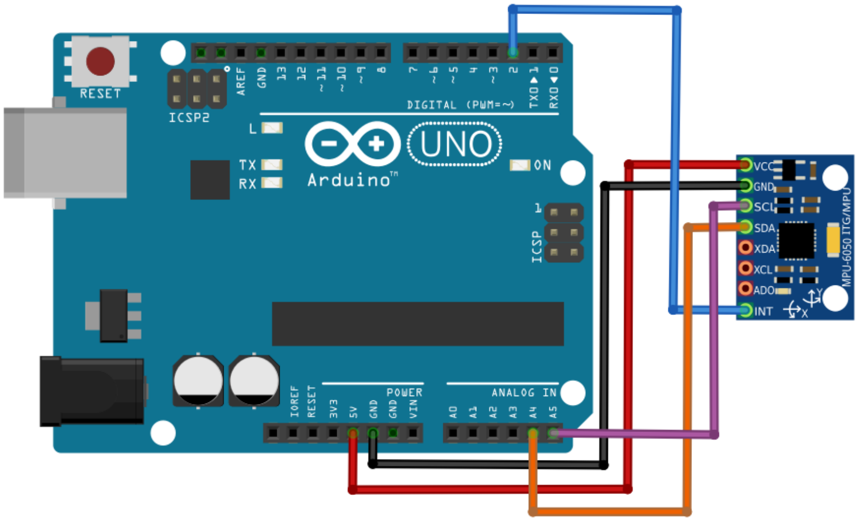
IV. ACELEROMETER:-

An accelerometer is an electromechanical device used to measure acceleration forces. Such forces may be static, like the continuous force of gravity or, as is the case with many mobile devices, dynamic to sense movement or vibrations.

- Acceleration is the measurement of the change in velocity, or speed divided by time.

- **What is the working principle of accelerometer?**

The basic underlying **working principle** of an **accelerometer** is such as a dumped mass on a spring. When acceleration is experienced by this device, the mass gets displaced till the spring can easily move the mass, with the same rate equal to the acceleration it sensed.



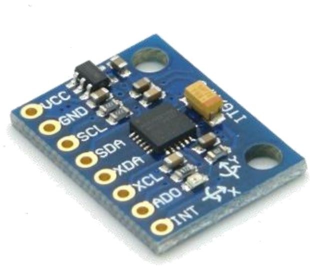
- **Pitch is rotation on the Y-axis, which means the object is rotated up or down:**

    - If the accelerometer is **tilted up**, the **pitch** will be a **positive** angle (between 1° to 180°).
    - If the accelerometer is **tilted down**, the **pitch** will be a **negative** angle (between -1° to -180°).

- **Roll is rotation on the X-axis, which means the object is rotated to the right or left:**

    - If the accelerometer is **tilted right**, the **roll** will be a **positive** angle (between 1° to 180°).
    - If the accelerometer is **tilted left**, the **roll** will be a **negative** angle (between -1° to -180°).

- **When the accelerometer is perfectly level, the pitch and roll will both be equal to 0 (zero).**
- **Here we used Accelerometer mpu6050 module.**



(Accelerometer mpu6050)

How to implement Accelerometer mpu6050 in Arduion IDE:-

```
int angle()
          {
          Wire.beginTransmission(MPU_addr);
          Wire.write(0x3B);
          Wire.endTransmission(false);
          Wire.requestFrom(MPU_addr,14,true);
          AcX=Wire.read()<<8|Wire.read();
          AcY=Wire.read()<<8|Wire.read();
          AcZ=Wire.read()<<8|Wire.read();
          int xAng = map(AcX,minVal,maxVal,-90,90);
          int yAng = map(AcY,minVal,maxVal,-90,90);
          int zAng = map(AcZ,minVal,maxVal,-90,90);
          x= RAD_TO_DEG * (atan2(-yAng, -zAng)+PI);

          y= RAD_TO_DEG * (atan2(-xAng, -zAng)+PI);

          z= RAD_TO_DEG * (atan2(-yAng, -xAng)+PI);

          return y;

          }
```

V. JUMP WIRE:-

A **jump wire** (also known as jumper wire, or jumper) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned").

It is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

VI. LASER LIGHT:-

Here we used laser light for the point on the object.

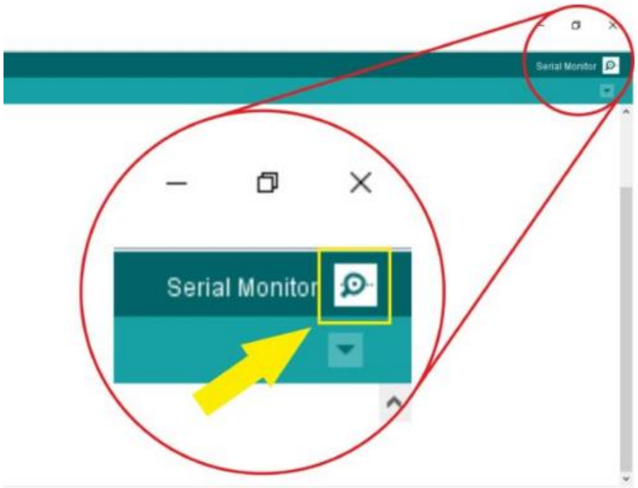
VII. BREADBORD:-

A **breadboard** is a rectangular plastic board with a bunch of tiny holes in it.

3. PLATFORMS/LANGUAGE USED:-

The **Arduino** Integrated Development Environment - or **Arduino** Software (**IDE**) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the **Arduino** and Genuino hardware to upload programs and communicate with them.

- The **Arduino** language is **C**++, but it is very different from most **C++** varieties. The **Arduino** language has a lot of abstraction built in, especially in the hardware interfaces, which makes it very simple to use.
- **IDE SOFTWARE " https://www.arduino.cc/en/software "**
- **In Aduino IDE the reading shows in serial monitor.**



4. FINAL EXECUTION(LOGIC):-

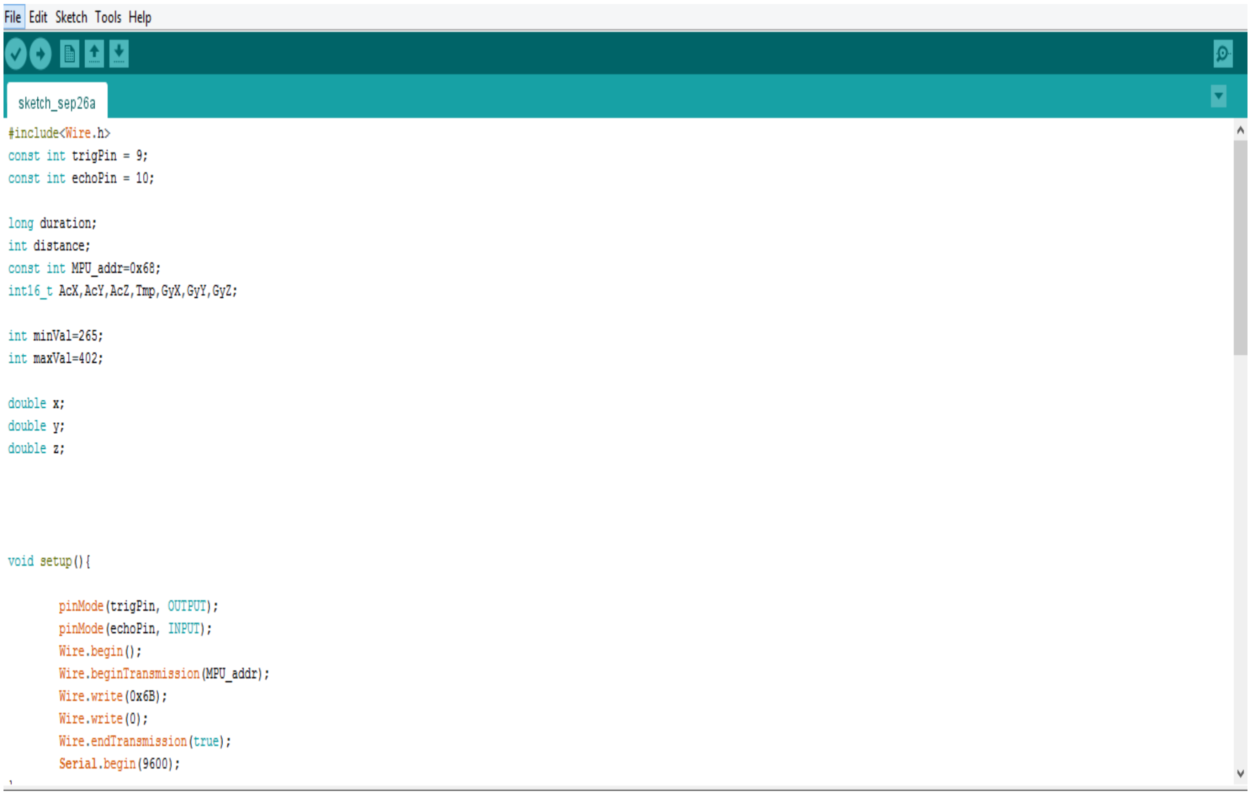
- As we are using the Arduino IDE so the main function is the void loop. The logic behind our project is in the void loop function.

- In void loop function we are connecting the Ultrasonic sensor function (float distance()) and Accelerometer function (int angle()).
- The value which are coming from Ultrasonic sensor are collected stored in the variable.
- Similarly, we are doing that for accelerometer values.
- Once the value are collected then we applying tangential function of maths for calculating the height.
- The tan function takes the radian values as of angle so we are converting our values from degree to radian.
- And finally we got the height.

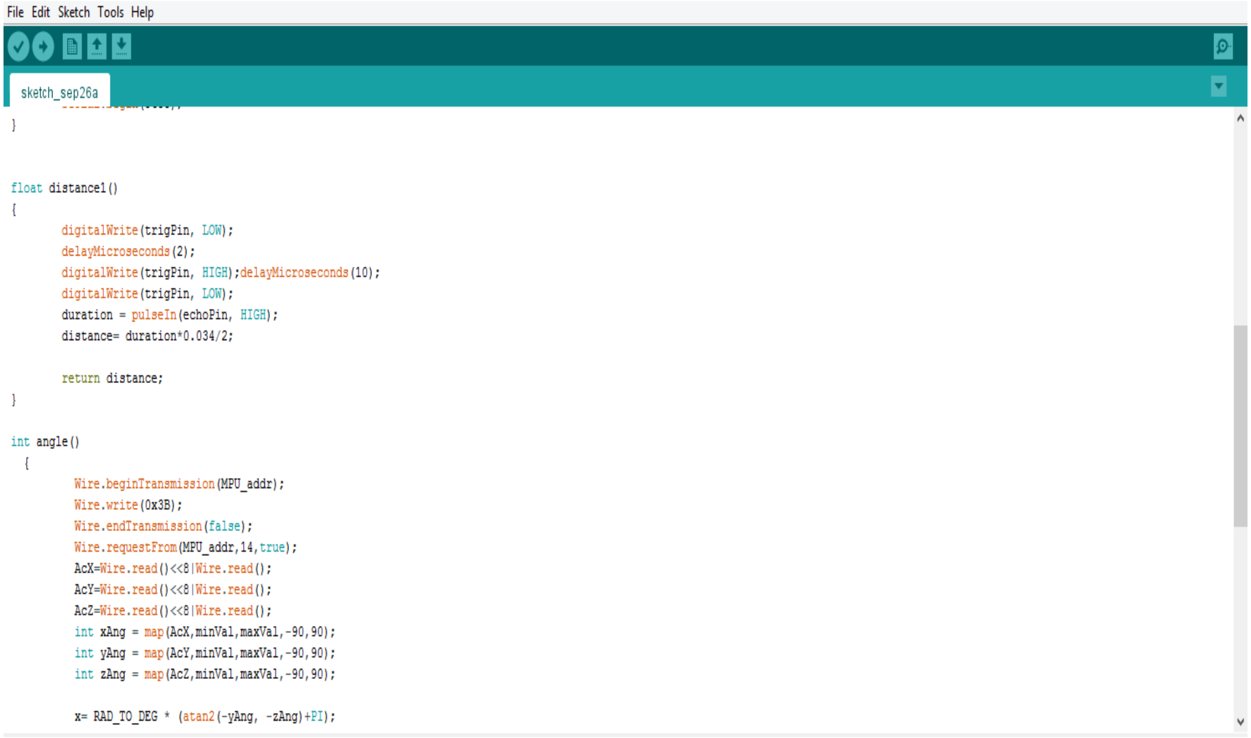
CODE:

```
void loop(){

  float a=distance1();

  int b=angle();
  b=b-98;

   Serial.println("dis");
  Serial.println(a);
   Serial.println("ang");
  Serial.println(b);

  delay(100);
  float h= tan((3.14159/180)*b)*a;
  Serial.println("height");
  Serial.println(h+4);

  Serial.println("-----------------------------------------------");
  delay(1500);
}
```

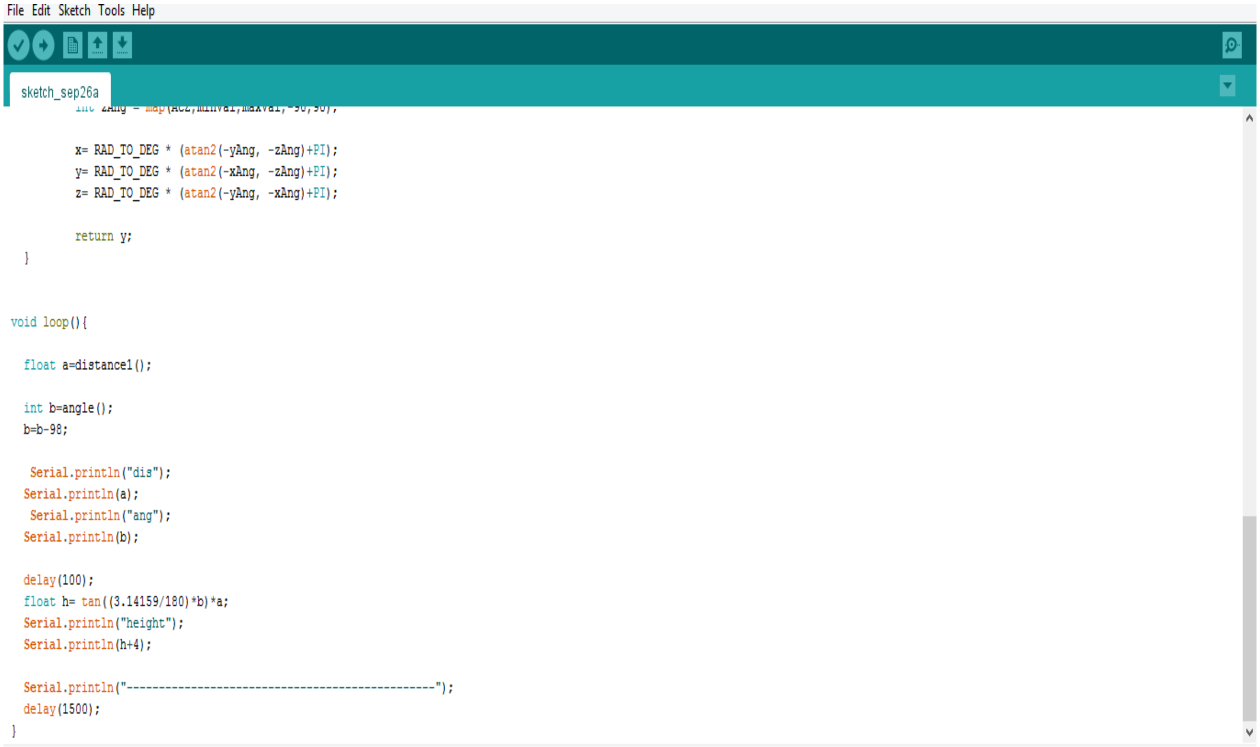
5. SCREENSHOTS:-



```
#include<Wire.h>
const int trigPin = 9;
const int echoPin = 10;

long duration;
int distance;
const int MPU_addr=0x68;
int16_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;

int minVal=265;
int maxVal=402;

double x;
double y;
double z;




void setup(){

        pinMode(trigPin, OUTPUT);
        pinMode(echoPin, INPUT);
        Wire.begin();
        Wire.beginTransmission(MPU_addr);
        Wire.write(0x6B);
        Wire.write(0);
        Wire.endTransmission(true);
        Serial.begin(9600);
}
```

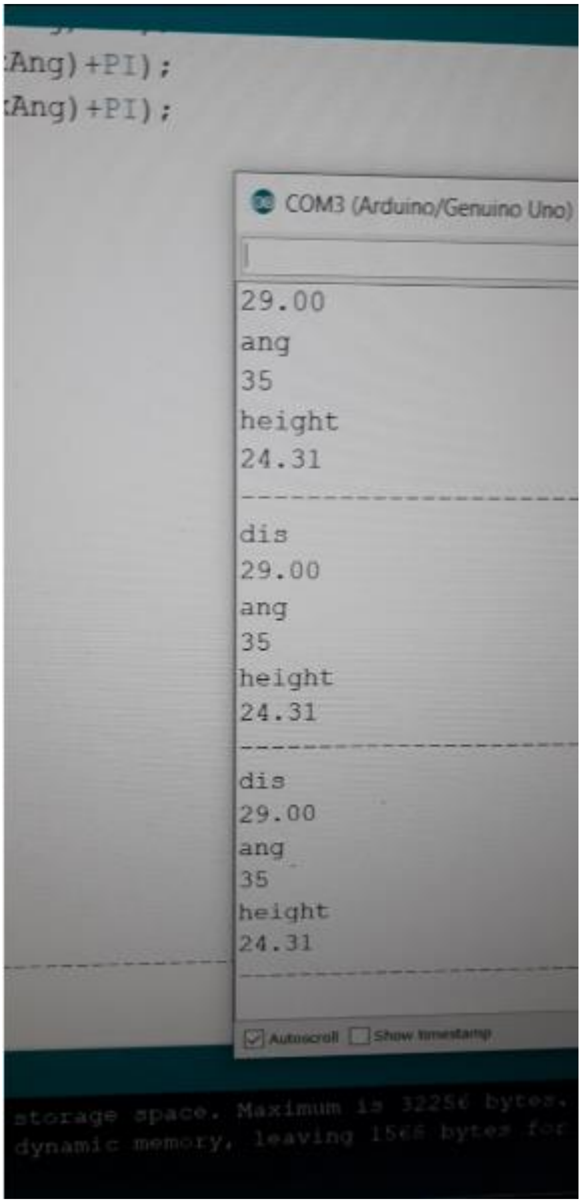


```
}


float distance1()
{
        digitalWrite(trigPin, LOW);
        delayMicroseconds(2);
        digitalWrite(trigPin, HIGH);delayMicroseconds(10);
        digitalWrite(trigPin, LOW);
        duration = pulseIn(echoPin, HIGH);
        distance= duration*0.034/2;

        return distance;
}

int angle()
  {
          Wire.beginTransmission(MPU_addr);
          Wire.write(0x3B);
          Wire.endTransmission(false);
          Wire.requestFrom(MPU_addr,14,true);
          AcX=Wire.read()<<8|Wire.read();
          AcY=Wire.read()<<8|Wire.read();
          AcZ=Wire.read()<<8|Wire.read();
          int xAng = map(AcX,minVal,maxVal,-90,90);
          int yAng = map(AcY,minVal,maxVal,-90,90);
          int zAng = map(AcZ,minVal,maxVal,-90,90);

          x= RAD_TO_DEG * (atan2(-yAng, -zAng)+PI);
```



```
          int zAng = map(AcZ,minVal,maxVal,-90,90);

          x= RAD_TO_DEG * (atan2(-yAng, -zAng)+PI);
          y= RAD_TO_DEG * (atan2(-xAng, -zAng)+PI);
          z= RAD_TO_DEG * (atan2(-yAng, -xAng)+PI);

          return y;
  }


void loop(){

  float a=distance1();

  int b=angle();
  b=b-98;

   Serial.println("dis");
  Serial.println(a);
   Serial.println("ang");
  Serial.println(b);

  delay(100);
  float h= tan((3.14159/180)*b)*a;
  Serial.println("height");
  Serial.println(h+4);

  Serial.println("------------------------------------------------");
  delay(1500);
}
```

OUTPUT IN SERIAL MONITOR

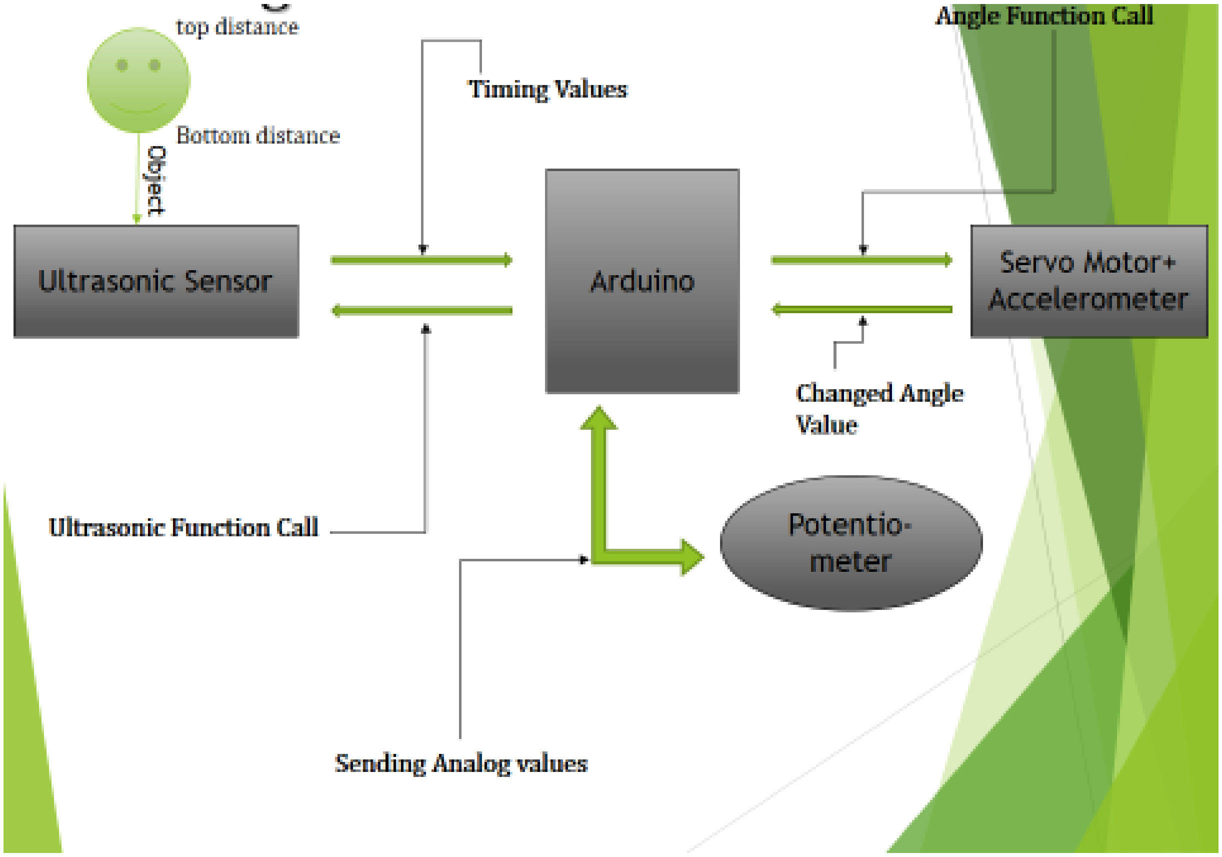


```
Ang)+PI);
Ang)+PI);
```

```
29.00
ang
35
height
24.31
--------------------
dis
29.00
ang
35
height
24.31
--------------------
dis
29.00
ang
35
height
24.31
--------------------
```

```
storage space. Maximum is 32256 bytes.
dynamic memory, leaving 1566 bytes for
```

6. BLOCK DIAGRAM:-



7. REFERENCES:-

- https://circuitdigest.com/microcontroller-projects/arduino-ultrasonic-sensor-based-distance-measurement#:~:text=Connecting%20wires-,Ultrasonic%20Sensor%20Module,transmitter%2C%20receiver%20and%20control%20circuit
- https://www.electronicwings.com/arduino/mpu6050-interfacing-with-arduino-uno

8. CONCLUSION:-

We conclude that, our system will provide services to measure height and as promptly as possible services more efficiently with the help of this device .It is User-friendly human machine interface (HMI) for quickly measure the height.

* * *