ULTRASONIC SENSOR

AIM:our goal in this project is to measure the distance to an object by using sound waves through ultrasonic sensors.

Software Required: Arduino IDE

Components Required:

1. System -1
2. Arduino Uno Board -1
3. Arduino dumping cable -1
4. Ultrasonic sensor -1
5. Resistor 330 ohms-1
6. Breadboard-1
7. Connecting Wires -Required

Theory:-An Ultrasonic sensor is a device that can measure the distance to an object by using sound waves. It measures distance by sending out a sound wave at a specific frequency and listening for that sound wave to bounce back. By recording the elapsed time between the sound wave being generated and the sound wave bouncing back, it is possible to calculate the distance between the sonar sensor and the object.

Since it is known that sound travels through air at about 344 m/s (1129 ft/s), you can take the time for the sound wave to return and multiply it by 344 meters (or 1129 feet) to find the total round-trip distance of the sound wave. Round-trip means that the sound wave traveled 2 times the distance to the object before it was detected by the sensor; it includes the 'trip' from the sonar sensor to the object AND the 'trip' from the object to the Ultrasonic sensor (after the sound wave bounced off the object). To find the distance to the object, simply divide the round-trip distance in half.

It is important to understand that some objects might not be detected by ultrasonic sensors. This is because some objects are shaped or positioned in such a way that the sound wave bounces off the object, but are deflected away from the Ultrasonic sensor. It is also possible for the object to be too small to reflect enough of the sound wave back to the sensor to be detected. Other objects can absorb the sound wave all together (cloth, carpeting, etc), which means that there is no way for the sensor to detect them accurately. These are important factors to consider when designing and programming a robot using an ultrasonic sensor.

Specification:

- Working Voltage : 5V(DC)
- Working Current : max 15 ma
- Working frequency : 40HZ
- Output Signal : 0-5V (Output high when obstacle in range)
- Sentry Angle : max 15 degree
- Sentry Distance : 2cm - 500cm
- High-accuracy : 0.3cm
- Input trigger signal : 10us

- TTL impulse Echo : output TTL PWL signal
- Size : 45*20*15mm

Usage:-

Supply module with 5V, the output will be 5V while obstacle in range, or 0V if not.

The out pin of this module is used as a switching output when anti-theft module, and without the feet when ranging modules.

Note: the module should be inserted in the circuit before being powered, which avoids producing high levels of disoperation ; if not, then power again.

Module Working Principle:

(1) Adopt IO trigger through supplying at least 10us sequence of high level signal,

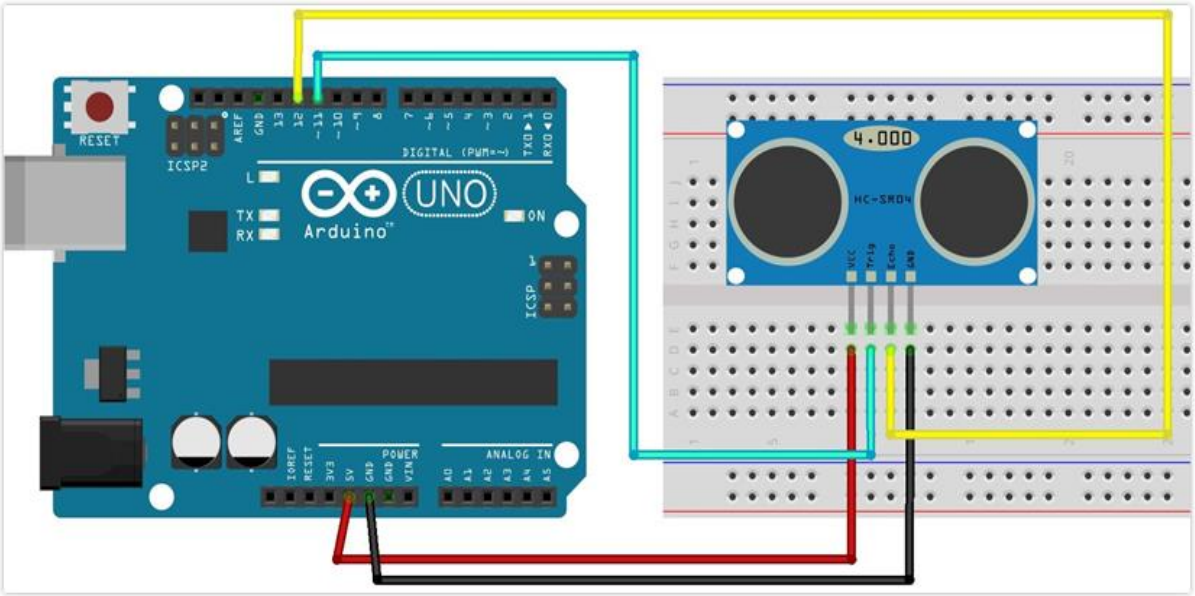
(2) The module automatically sends eight 40 kHz square wave and automatically detect whether receive the returning pulse signal,

(3) If there is a signal returning, through outputting high level and the time of high level continuing is the time of that from the ultrasonic transmitting to receiving.

Test distance = (high level time * sound velocity (340M/S) / 2

Ultrasonic Sensing Applications:

Migatron has been using advanced technology to solve difficult sensing and control problems for nearly 40 years across a broad range of industries. With Ultrasonic Sensing's unique advantages over conventional sensors and the rapidly increasing range of applications, ultrasonic sensors are becoming widely accepted as an industry standard across the board.



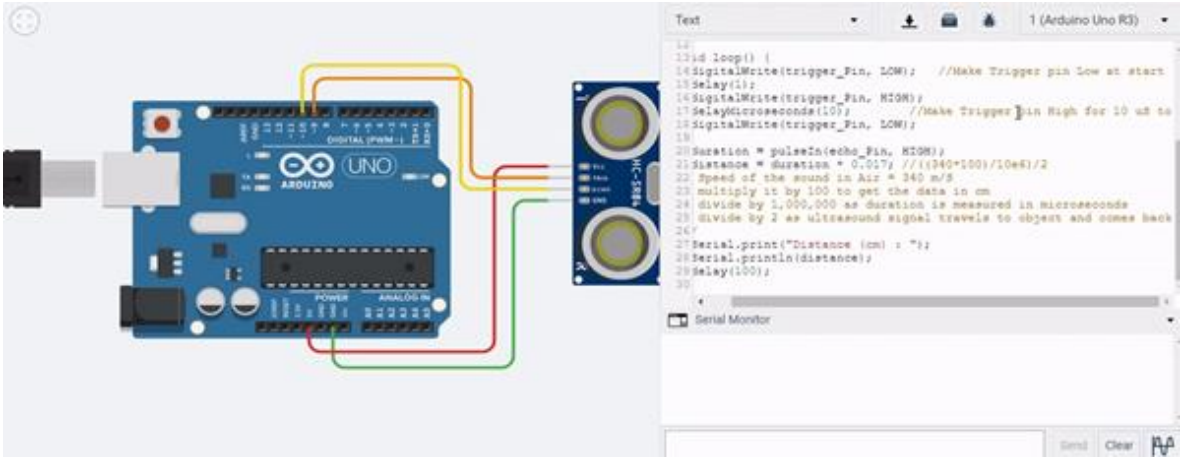
The connections are as follows:

- Vcc to 5V Pin of the Arduino.
- Gnd to Gnd Pin of the Arduino.
- Trig to Digital Pin 9

Procedure:-

1. Open Arduino IDE.
2. Write the code in the text editor.
3. Save the sketch with .ino extension.

4. Connect the hardware circuit and Connect your Arduino Board to the USB port of your computer.
5. Select the serial device of the Arduino board from the Tools | Serial Port menu.
6. Compile the file by clicking on the verify button.
7. If successful, the message "Done Compiling." will appear in the status bar.
8. If there are any errors it will show them in the Transcript window, rectify those errors and compile it again.
9. Push the reset button on the board then click the Upload button in the IDE. Wait a few seconds. If successful, the message "Done uploading." will appear in the status bar.
10. Click the serial monitor button in the toolbar and select the same baud rate used in the call to begin ().

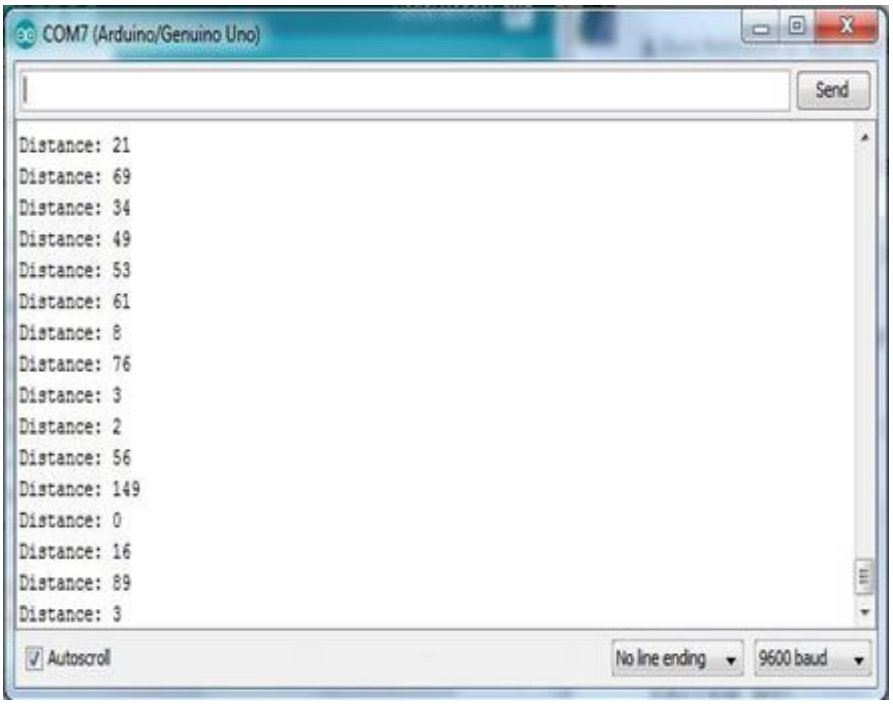


Code:-

```
int trig=10;
int echo=11;
long duration;
long distance;
 void setup()
{
pinMode(trig,OUTPUT);
pinMode(echo,INPUT);
Serial.begin(9600);
 }
void loop()
{
digitalWrite(trig,LOW);
delayMicroseconds(2);
digitalWrite(trig,HIGH);
delayMicroseconds(10);
digitalWrite(trig,LOW);
duration = pulseIn(echo,HIGH);
distance = duration*(0.034/2);
Serial.print("Distance in CM: ");
```

```
Serial.println(distance);
}
```

Result:



Servo motor with ultrasonic sensor

AIM:Our goal in this project is to interface the servo motor with ultrasonic sensor by using Arduino

Software Required: Arduino IDE

Components Required:

1. System -1
2. Arduino Uno Board -1
3. Arduino dumping cable -1
4. Servomotor-1
5. Ultrasonic sensor-1
6. Resistor 330 ohms-1
7. Breadboard-1
8. Connecting Wires -Required

Theory:-when ultrasonic sensor read 5cm distance or less servo will rotate in 120 degree , else servo will rotate 0 degree

Introduction:

Arduino will print the distance on serial print . You can change the code easily if you need to do other things. HardwarePut the trigpin of the ultrasonic sensor in pin 7 on the arduino. Put

the echopin of ultrasonic sensor in pin 6 on the arduino. Put the vcc pin of the ultrasonic sensor to 5v on the arduino. Put servo to pin 8 on arduino. Put gnd of ultrasonic to gnd on arduino. and connect the arduino with Pc.

Code:-

```
#include<Servo.h>//Header file
#define trigPin 7//hardware connection
#define echoPin 6//hardware connection
Servo servo;
void setup()//to run once
{
Serial.begin (9600);//intiate serial monitor
pinMode(trigPin, OUTPUT);//sets input/output
pinMode(echoPin, INPUT);
servo.attach(8);//pin declation
}
void loop()//to run repeatedly
{
long duration, distance;
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = (duration/2) / 29.1;
if (distance < 5)
{
Serial.println("the distance is less than 5");
servo.write(120);
}
else
{
servo.write(0);
}
if (distance > 60 || distance <= 0)
{
Serial.println("The distance is more than 60");
}
else
{
Serial.print(distance);
Serial.println(" cm");
}
delay(500);
}
```

Result: