**Distance Sensing and Measurement**

**with**

**HC-SR04 Ultrasonic Sensor**

1. **Abstract:**

This project uses Ultrasonic Sensor HC-SR04. There is light indication, over certain distance coverage. Red light, Green light and Yellow light will glow and tell us about distance range, according to Arduino Programming. Distance is displayed in Inches and Centimeters. Interaction is modified and enhanced with user interface built of Visual Studio latest version.

1. **Introduction to Sensor :**

The HC-SR04 ultrasonic sensor uses sonar signals to determine distance to an object. It offers excellent range accuracy and stable readings in an easy-to-use package. Its operation is not affected by sunlight or black material. However, acoustically soft materials like cloth can be difficult to detect. It uses sonar to detect objects at a distance of 2 cm to 4 meters.  This sensor is widely used in robotics to build robots that move and should divert or avoid obstacles.

Moreover, it is easy to control it through the Arduino, because it only has 4 pins described below.

* **VCC** – 5V (ranging from 4.5 V to 5.5 V)
* **Trig** – Sensor input (trigger)
* **Echo** – Output Sensor (Echo)
* **GND** – Ground



1. **Project Scope:**

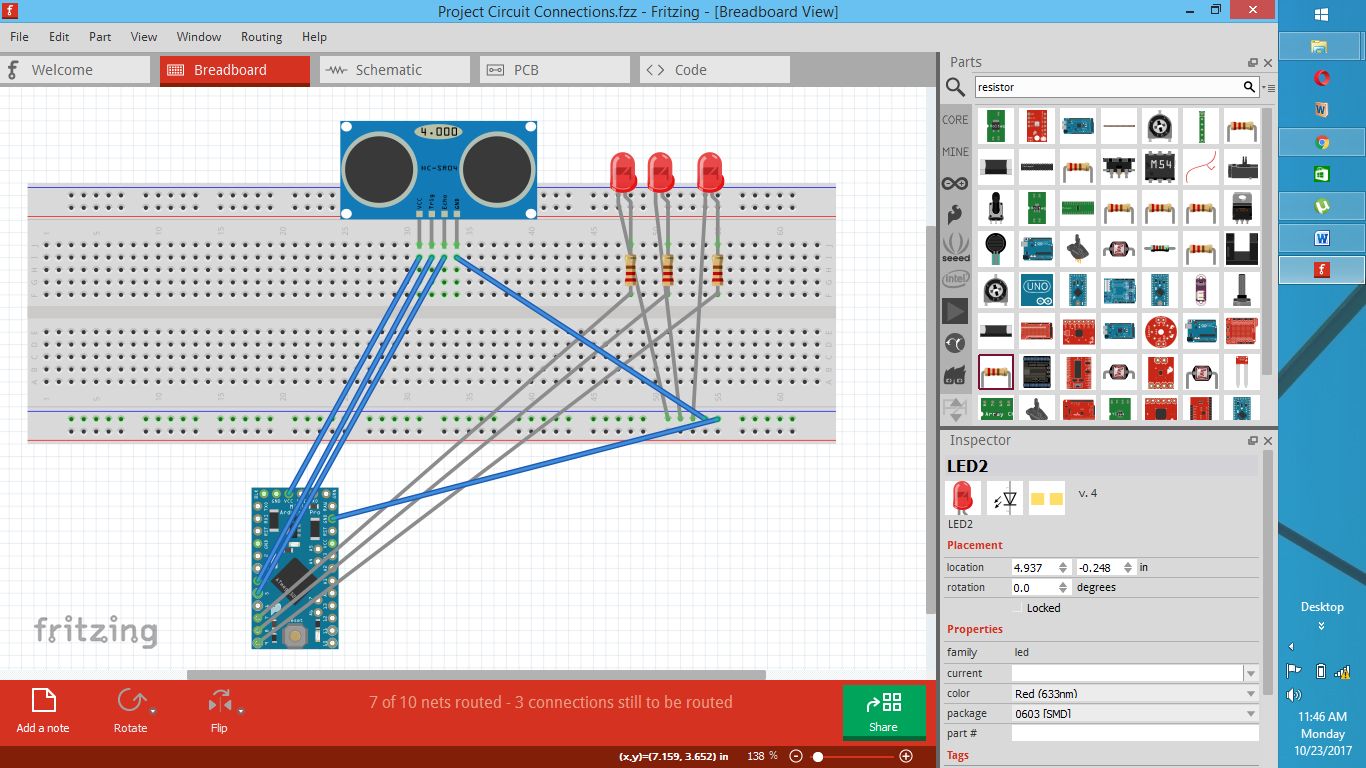
This device will enable you to recognize the presence of obstacle and distance within certain limit. This feature can be used in real time gaming and security applications. That feature can help in Security applications, Medical field machines, Gaming etc.

1. **Components Required:**

* HC- SR04
* Three resistors of 220 ohms
* Three resistors of 330 ohms
* Jumper wires;
* Three LEDs of different colors;
* Breadboard

1. **Connections:**

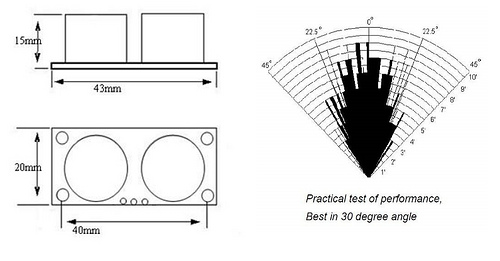
* Digital ports 10, 11 and 12 of the resistors connected Arduino, which in turn are connected the legs of the LEDs positive;
* Cathode pin of the LED in the GND (ground) of the Arduino;
* VCC pin of the ultrasonic sensor HC-SRO4 in Arduino 5V;
* TRIG pin sensor HC-SRO4 digital port on the Arduino 6;
* ECHO pin sensor HC-SRO4 the digital port 7 of the Arduino;
* GND pin HC-sensor SRO4 in Arduino GND.



1. **Working:**

A short ultrasonic pulse is transmitted at the time 0, reflected by an object. The sensor receives this signal and converts it to an electric signal. The next pulse can be transmitted when the echo is faded away. This time period is called cycle period. The recommended cycle period should not be less than 50ms. If a 10μs width trigger pulse is sent to the signal pin, the Ultrasonic module will output eight 40kHz ultrasonic signal and detect the echo back. The measured distance is proportional to the echo pulse width and can be calculated by the formula above. If no obstacle is detected, the output pin will give a 38ms high level signal.

* If distance is greater than 20 cm, then LED Green will be turned ON.
* If distance is less than & equal to 20 cm and greater than & equal to 10, then LED Yellow will be turned ON.
* If distance is less than 10 cm, then LED Red will be turned ON.



1. **Source Code:**

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* MUBASHARA REHMAN

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* PROJECT CODE

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ID# 6217000042

#include "Ultrasonic.h"

#define SERIAL\_PLOTTER 2

Ultrasonic ultrasonic(6,7);

//-------------------------------------------------------------------------------

//-------------------------------------------------------------------------------

static int outputType = SERIAL\_PLOTTER;

//Declaration of constant for reference of Digital ports

//Pin 6 and 7 are useD, Trigger - Digital Port 6, Echo - Digital Port 7

const int trigPin = 6; //Trig pin

const int echoPin = 7; //Echo pin

const int ledgreen = 12;

const int ledyellow = 11;

const int ledred = 10;

long microsec = 0;

float distanceCM = 0;

//-------------------------------------------------------------------------------

//-------------------------------------------------------------------------------

void setup ()

{

Serial.begin(9600); //starts the serial communication

pinMode(ledgreen,OUTPUT); //sets Green led as output

pinMode(ledyellow,OUTPUT); //sets Yellow led as output

pinMode(ledred,OUTPUT); //sets Red led as output

}

//-------------------------------------------------------------------------------

//-------------------------------------------------------------------------------

void loop ()

{

long duration, inches, cm;

microsec = ultrasonic.timing();

distanceCM = ultrasonic.convert(microsec, Ultrasonic::CM);

leddistance();

pinMode(trigPin, OUTPUT);

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

pinMode(echoPin, INPUT);

duration = pulseIn(echoPin, HIGH);

// convert the time into a distance

inches = microsecondsToInches(duration);

cm = microsecondsToCentimeters(duration);

Serial.print(inches);

Serial.print("in, ");

Serial.print(cm);

Serial.print("cm");

Serial.println();

delay(2000);

}

//-------------------------------------------------------------------------------

//-------------------------------------------------------------------------------

void leddistance ()

{

digitalWrite(ledgreen,LOW); //initially setting all led's to off

digitalWrite(ledyellow,LOW);

digitalWrite(ledred,LOW);

if (distanceCM > 20)

{

digitalWrite(ledgreen,HIGH); //turn the LED on (HIGH is the voltage level)

}

if (distanceCM <=20 and distanceCM >= 10)

{

digitalWrite(ledyellow,HIGH); //turn the LED on (HIGH is the voltage level)

}

if (distanceCM < 10) {

digitalWrite(ledred,HIGH); //turn the LED on (HIGH is the voltage level)

}

}

//-------------------------------------------------------------------------------

//-------------------------------------------------------------------------------

long microsecondsToInches(long microseconds)

{

// According to Parallax's datasheet for the PING))), there are

// 73.746 microseconds per inch (i.e. sound travels at 1130 feet per

// second). This gives the distance travelled by the ping, outbound

// and return, so we divide by 2 to get the distance of the obstacle.

return microseconds / 74 / 2;

}

//-------------------------------------------------------------------------------

//-------------------------------------------------------------------------------

long microsecondsToCentimeters(long microseconds)

{

// The speed of sound is 340 m/s or 29 microseconds per centimeter.

// The ping travels out and back, so to find the distance of the

// object we take half of the distance travelled.

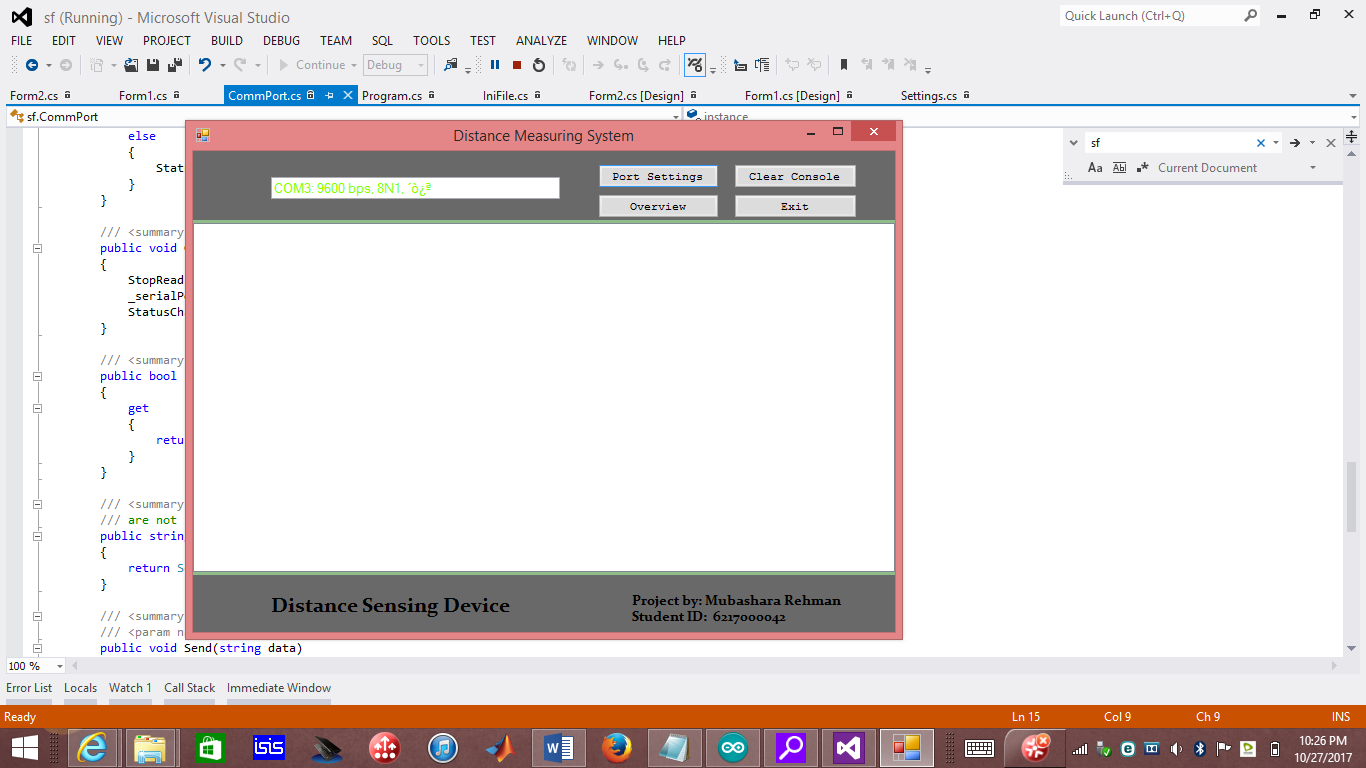
return microseconds / 29 / 2;

}

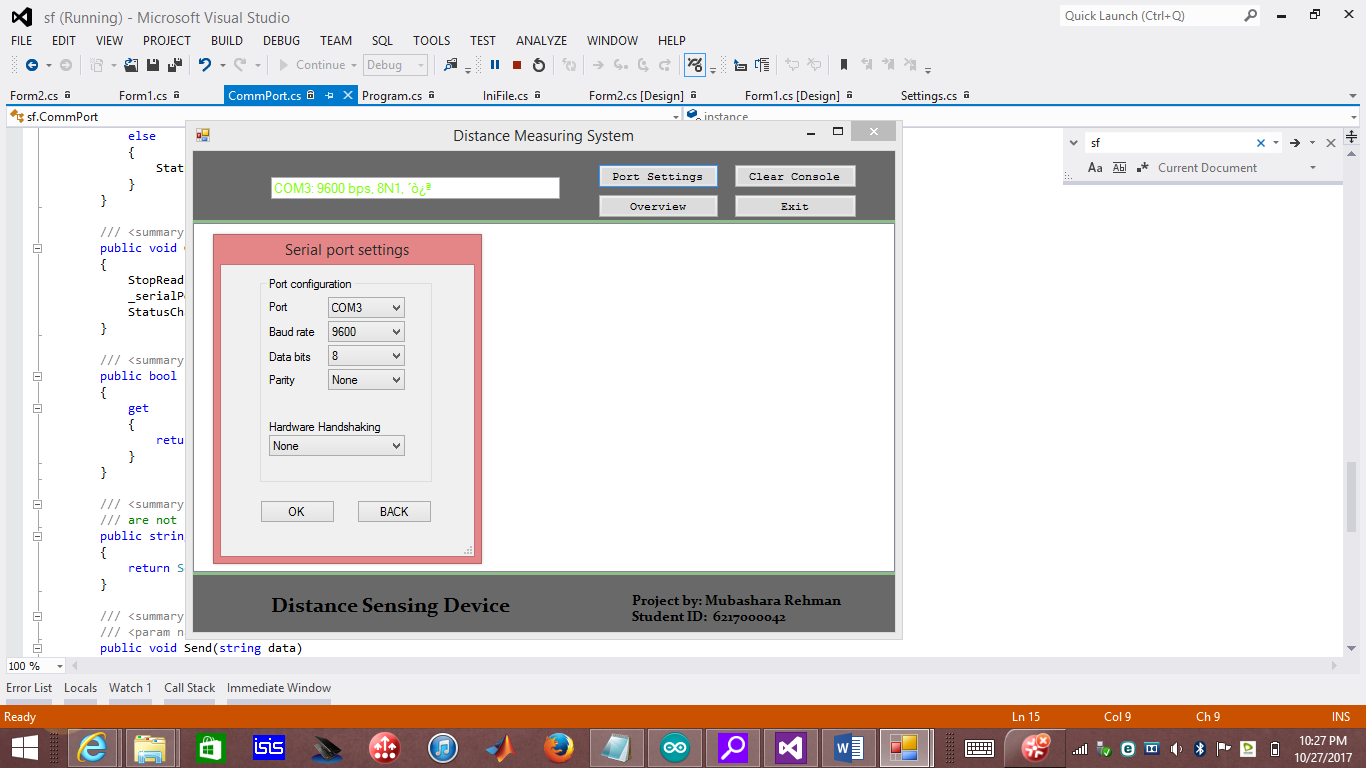
//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* THE END

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* THANK YOU

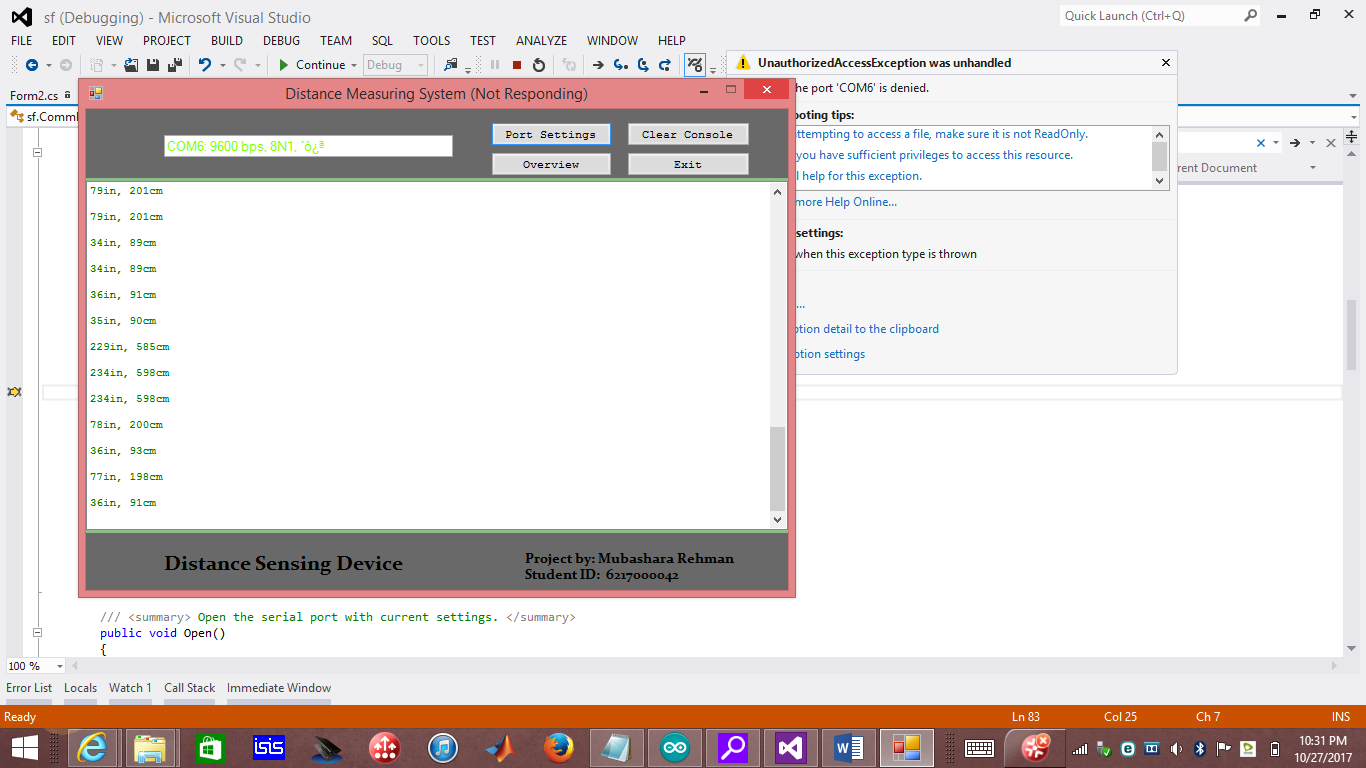
1. **Project Shots:**
2. Main Screen of *Distance Measuring System:*



1. Interface of *Port Settings.*



1. Console Output:



1. **References**:

* <http://www.buildcircuit.com/obstacle-sensor-using-arduino-and-hcsr04/>
* <https://www.youtube.com/watch?v=Fn6mVEKSLZc&feature=youtu.be>
* <http://www.instructables.com/id/Ultrasonic-Sensor-HC-SR04-Arduino-Project/>
* <https://create.arduino.cc/projecthub/onyx/ultrasonic-sensor-alarm-1ec0f3>