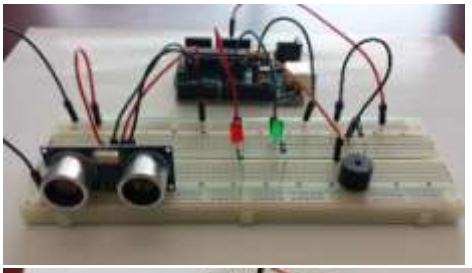
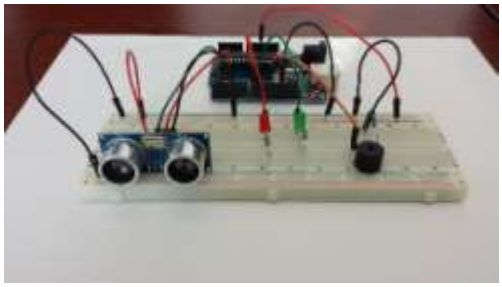
CONTROLLING LEDS AND BUZZER USING ULTRASONIC SENSOR

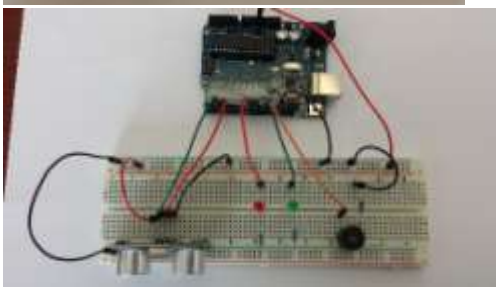
COMPONENTS:

1. ARDUINO UNO
2. BREAD BOARD
3. LED LIGHTS WITH RESISTORS
4. ULTRASONIC SENSOR
5. JUMPER WIRES

SETTING UP:







```
//Ultasonic Sensor


//Pins connected to the ultrasonic sensor
#define trigPin  2
#define echoPin 3
//LED pins
#define ledGreen 9
```

```
#define ledRed 8
//Pin connected to the piezo buzzer
#define alarm 11

int range = 5;//range in inches

void setup() {
  // initialize serial communication:
  Serial.begin(9600);
  //initialize the sensor pins
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  //initialize LED pins
  pinMode(ledGreen, OUTPUT);
  pinMode(ledRed, OUTPUT);
  //set LEDs
  digitalWrite(ledGreen, HIGH);
  digitalWrite(ledRed, LOW);

}
void loop()
{
  // establish variables for duration of the ping,
  // and the distance result in inches and centimeters:
  long duration, inches, cm;

  // The PING))) is triggered by a HIGH pulse of 2 or more microseconds.
  // Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
  digitalWrite(trigPin, LOW);
```

```
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(5);
digitalWrite(trigPin, LOW);

// Take reading on echo pin
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();

if(inches < 5) {
  Serial.println("DANGER");
  digitalWrite(ledGreen, LOW);
  digitalWrite(ledRed, HIGH);
  tone(alarm, 2000);
  delay(100);
} else {
  Serial.println("GOOD");
   digitalWrite(ledGreen, HIGH);
   digitalWrite(ledRed, LOW);
   noTone(alarm);
```

```
    delay(100);
  }

  delay(200);
}

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.
  // See: http://www.parallax.com/dl/docs/prod/acc/28015-PING-v1.3.pdf
  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;
}
```