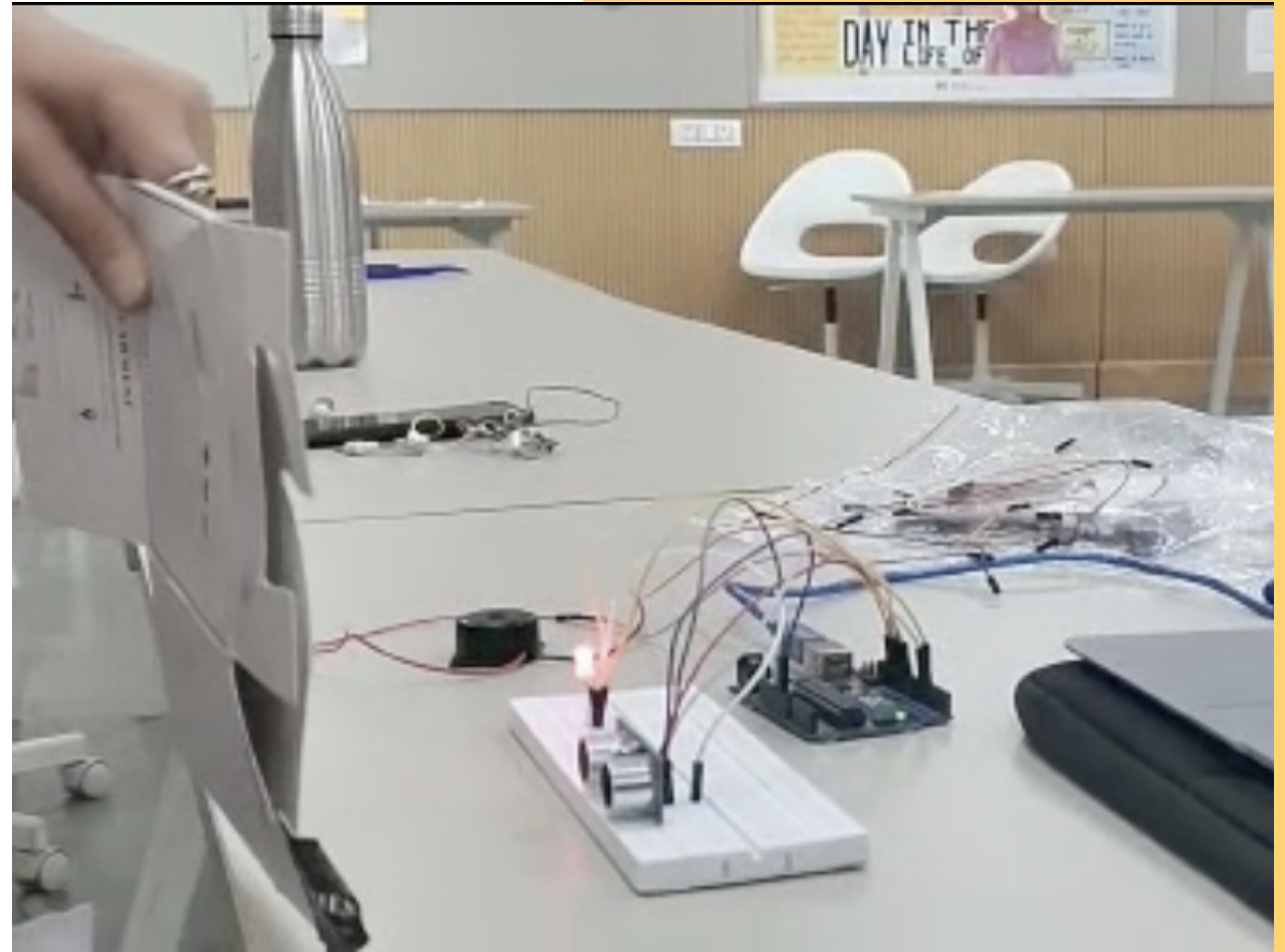


USING ARDUINO

BEEP BEEP

For this assignment, I combined the proximity sensor, RGB LED, Buzzer & Arduino.

In this application, when the sensor senses any object within 10 cm, the RGB LED turns red and the buzzer starts to buzz. If not in the 10 cm range then the RGB LED stays green. Check out the video by clicking the hyperlink below the image on the side.



[Watch the video here!](#)

HOW I MADE IT

- First, i connected the sensor to the arduino, then i connected the RGB LED & buzzer to the arduino.
 - I made these connections by understanding how these components work individually.
 - Then I typed the code in arduino IDE
 - While typing the code I had to make sure the connections in the circuit are precisely used in the code without error.
 - Combining the codes of the individual elements was the major part in this application, how to make sure all the elements work together properly.
 - You'll find the code used and the circuit that i made for this assignment in the following slides
-

ARDUINO CODE

```
sketch_apr20a.ino
1 #define trigpin 9
2 #define echopin 10
3 #define R 6
4
5 #define G 5
6 const int buzzer = 13;
7 int tone_duration = 1000;
8
9
10
11
12 void setup()
13 {
14   //serial monitor and pin setup.
15   Serial.begin(9600);
16   pinMode(trigpin,OUTPUT); //set trigpin as output
17   pinMode(echopin,INPUT); //set echopin as input
18   pinMode(R,OUTPUT); // set R,G and B as outputs
19   pinMode(G,OUTPUT);
20
21
22   // put your setup code here, to run once:
23 }
24
25
26
27 void loop()
28 {
29
30
31   //the trigpin sends out a signal, which bounces off an obstacle and comes back, the
32   //echopin recieves this signal and gives out +5v setting the arduino pin on which it is connected to high.
33   //distance= time*speed, but this distnce is divided by 2 because signal sent out returns
34   //so distance= (the time it takes for the signal to leave and return)/2.
```

Output

Sketch uses 3840 bytes (11%) of program storage space. Maximum is 32256 bytes.
Global variables use 209 bytes (10%) of dynamic memory, leaving 1839 bytes for local variables. Maximum is 2048 bytes.

Ln 20, Col 2 Arduino Uno [not connected]

```
sketch_apr20a.ino
33 //distance= time*speed, but this distnce is divided by 2 because signal sent out returns
34 //so distance= (the time it takes for the signal to leave and return)/2.
35 //i.e if the time is 6s the distance = (6s/2) = 3m or cm.
36
37 int duration, distance; //declare distance and duration as integers
38 digitalWrite(trigpin,HIGH); // trigin send out signal
39 _delay_ms(1000); //coninously for 1000ms
40 digitalWrite(trigpin, LOW); // then goes low
41
42 duration=pulseIn(echopin,HIGH); // duration is the pulseIn to the echopin
43 distance=(duration/2)/29.1; // the 29.1 is used to convert the distnce to cm, the value varies for other units.
44
45 if(distance > 0 && distance <= 10){ //distance is greater than 0 and less than 20cm
46   digitalWrite(G,LOW); //green led is off digitalWrite(B,LOW); //blue led is off
47
48   _delay_ms(100); //delay
49   digitalWrite(R,HIGH); //red led is on
50   _delay_ms(100);
51   digitalWrite(buzzer, HIGH);
52   tone(buzzer, 100, tone_duration); // Send 100Hz sound signal...
53 }
54 else if(distance > 10){ //distance is greater than 20 and less than 100cm
55   digitalWrite(R,LOW); //red led is off
56   digitalWrite(G,HIGH); //green led is on
57   _delay_ms(100);
58   digitalWrite(buzzer, LOW);
59
60
61 }
62
63 Serial.print("cm");
64 Serial.println(distance); //print values on serial monitor
65 _delay_ms(100);
66 }
```

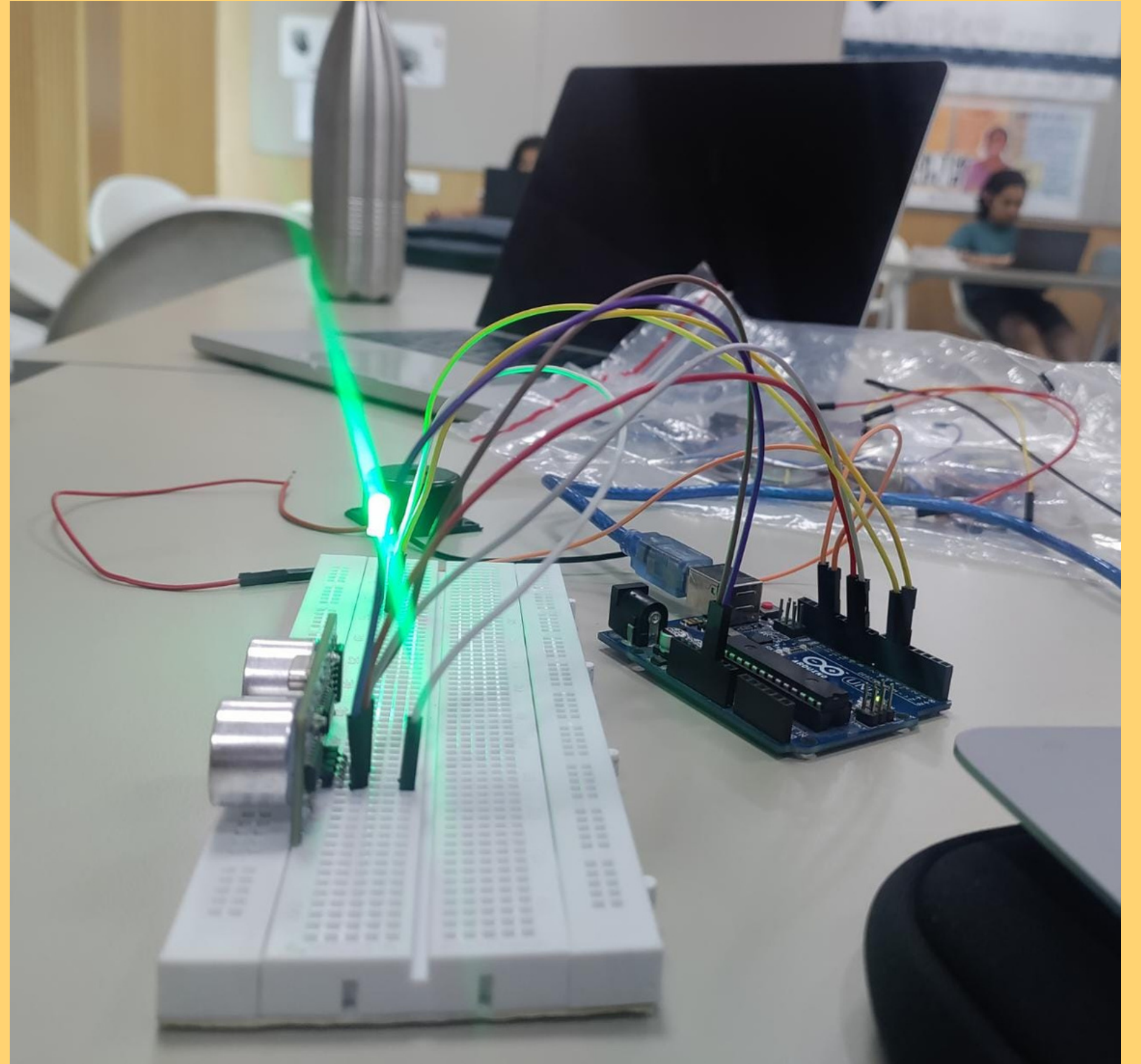
Output

Sketch uses 3840 bytes (11%) of program storage space. Maximum is 32256 bytes.
Global variables use 209 bytes (10%) of dynamic memory, leaving 1839 bytes for local variables. Maximum is 2048 bytes.

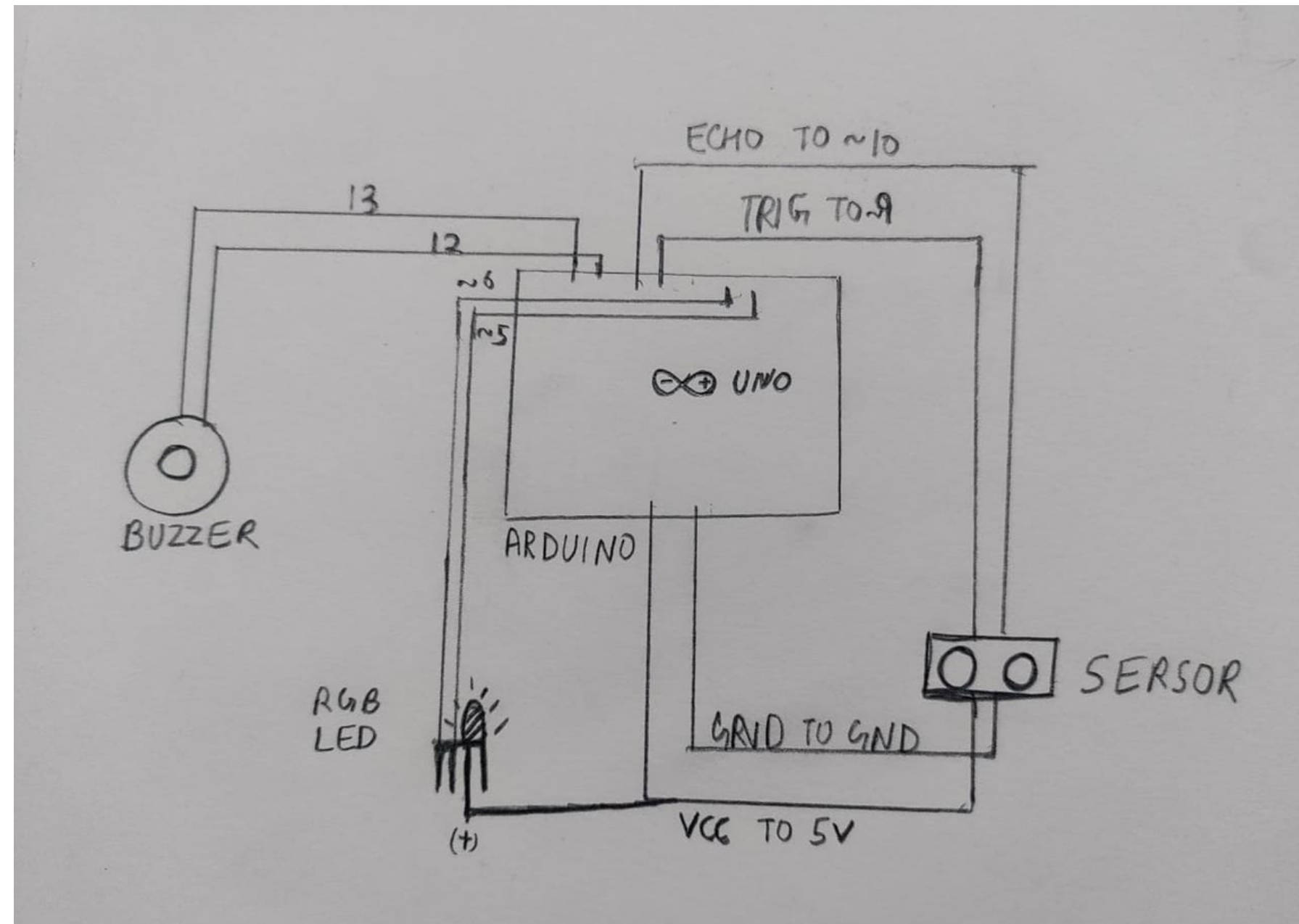
Ln 20, Col 2 Arduino Uno [not connected]

THE CIRCUIT

Go to the next page to see a well drawn circuit diagram!



CIRCUIT DIAGRAM



**THANK
YOU**