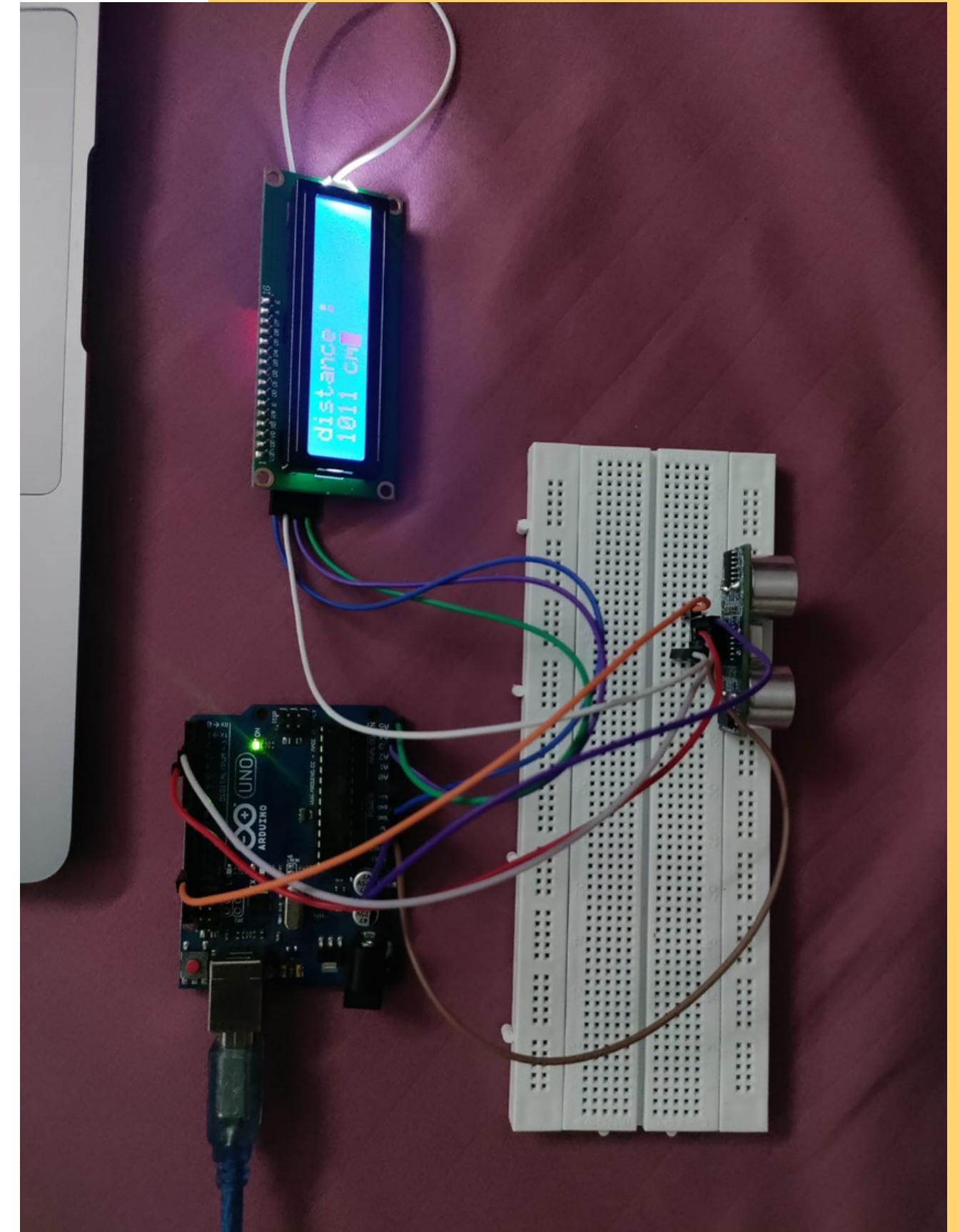


INPUT OUTPUT DEVICES

WHAT I MADE

I created a circuit using an ultrasonic sensor to measure distances, and displayed the output on an LCD display. This helped me in collecting data about distances and depths of objects in my surroundings.

check out the video by clicking on the hyperlink below the image!



[Click here to watch the video!](#)

HOW I MADE IT

- First, i connected the sensor to the arduino, then i connected the LCD display 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
-

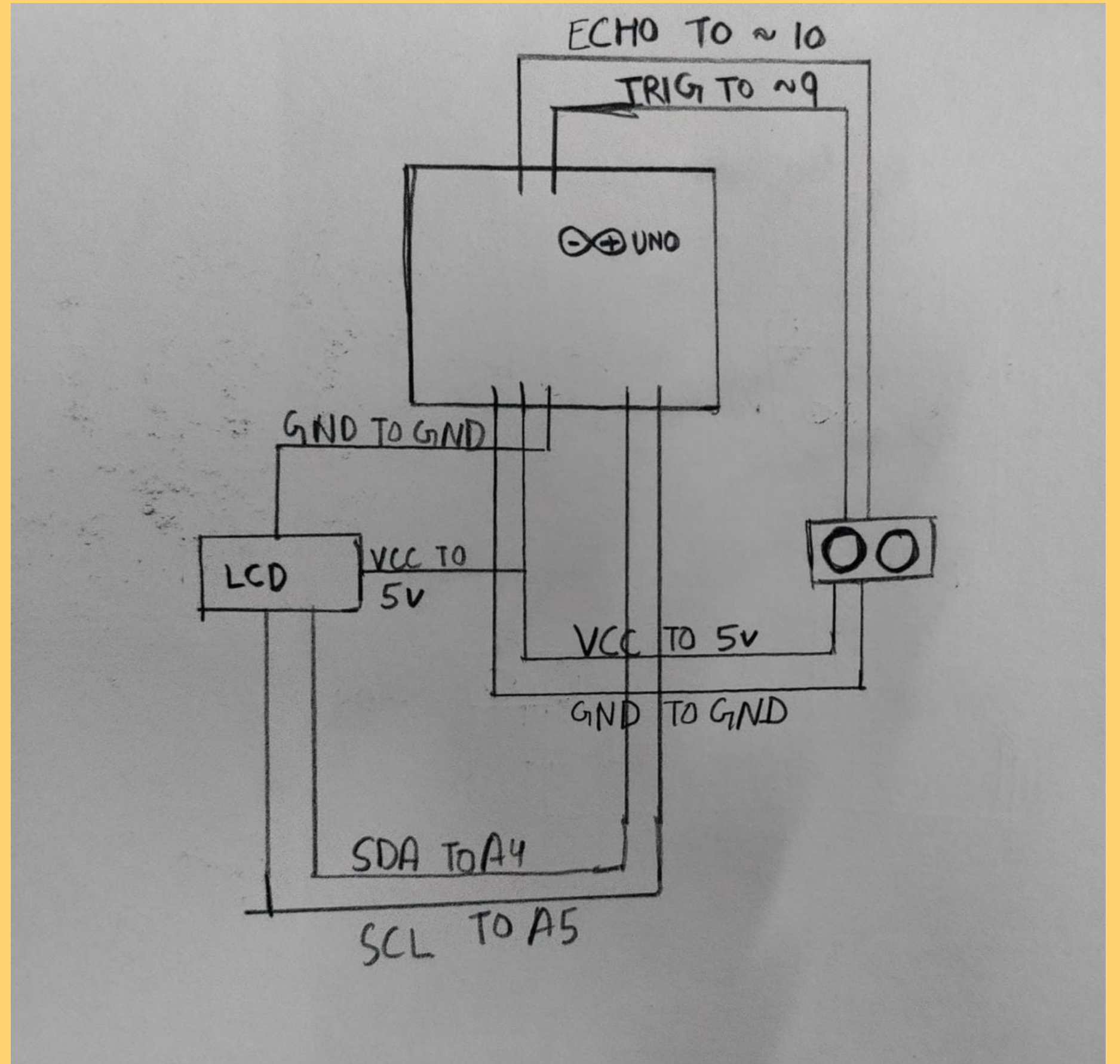
THE CODE

```
1  #include <Wire.h>
2  #include <LiquidCrystal_I2C.h>
3
4  LiquidCrystal_I2C lcd(0x27,16,2); // set the LCD address to 0x27 for a 16 chars and 2 line display
5
6  #define echoPin 3 // Echo Pin
7  #define trigPin 2 // Trigger Pin
8  #define LEDPin 13 // Onboard LED
9
10 long duration, distance; // Duration used to calculate distance
11
12 void setup() {
13
14     Serial.begin (9600);
15     pinMode(trigPin, OUTPUT);
16     pinMode(echoPin, INPUT);
17     pinMode(LEDPin, OUTPUT); // Use LED indicator (if required)
18
19     lcd.init(); // initialize the lcd
20     // Print a message to the LCD.
21     lcd.backlight();
22     // Blinking block cursor
23     lcd.blink_on();
24 }
25
26 void loop() {
27
```

THE CODE

```
28 // Delay 50ms before next reading.
29 delay(150);
30
31 /* The following trigPin/echoPin cycle is used to determine the
32 distance of the nearest object by bouncing soundwaves off of it. */
33 digitalWrite(trigPin, LOW);
34 delayMicroseconds(2);
35
36 digitalWrite(trigPin, HIGH);
37 delayMicroseconds(10);
38
39 digitalWrite(trigPin, LOW);
40 duration = pulseIn(echoPin, HIGH);
41
42 //Calculate the distance (in cm) based on the speed of sound.
43 distance = duration/58.2;
44
45 // display on lcd
46
47 String disp = String(distance);
48
49 lcd.clear();
50 lcd.print("distance :"); // first line
51 lcd.setCursor(0, 1); // second line
52 lcd.print(disp);
53 lcd.print(" cm");
54
```

THE CIRCUIT

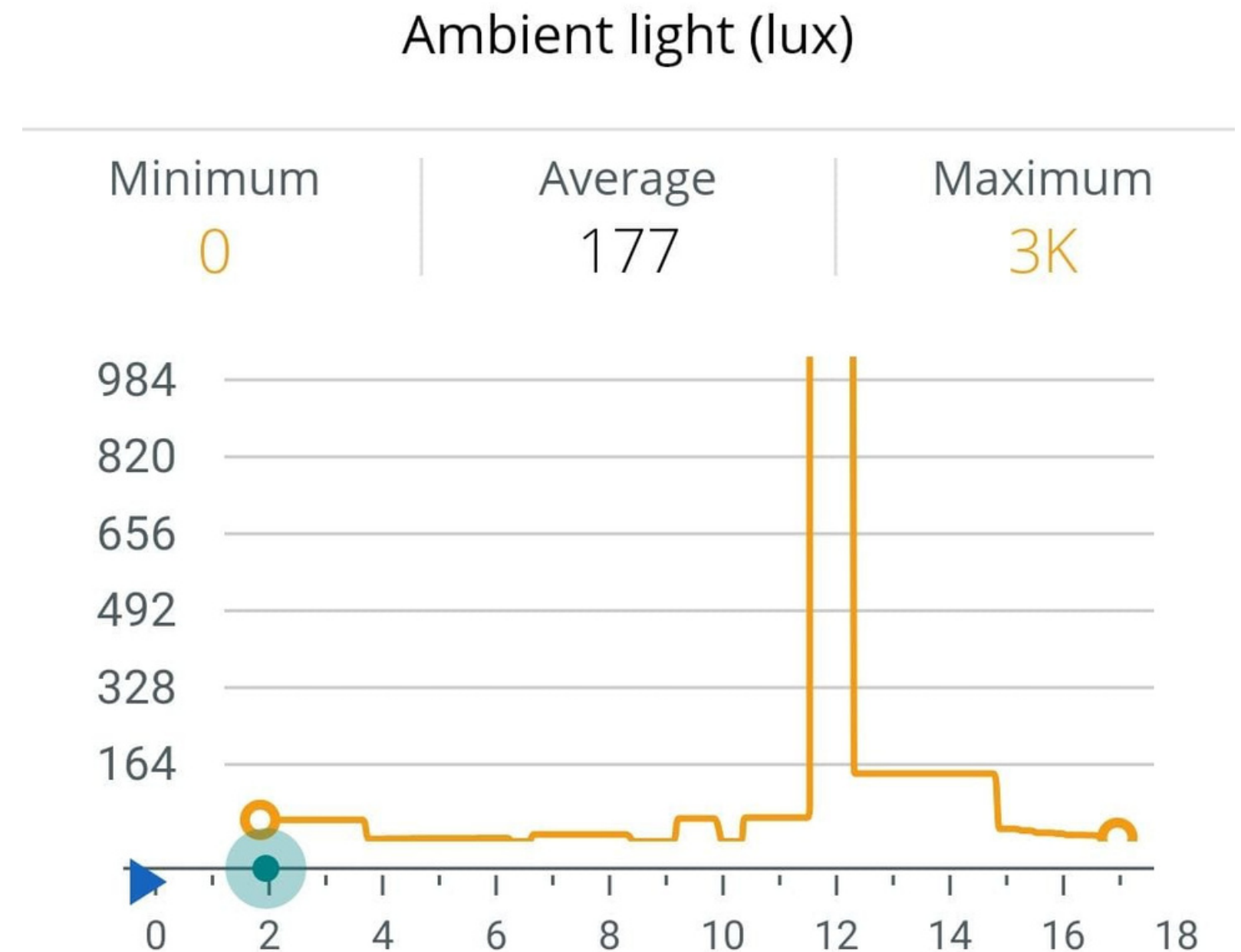


DIFFERENT SENSORS IN MY PHONE

- Accelerometer: This sensor measures orientation and movement to control screen rotation, switch between portrait and landscape modes, and enable motion-based gaming.
 - Gyroscope: This sensor detects movements in rotation, making it useful for controlling motion-based games and virtual reality apps.
 - Proximity Sensor: This sensor detects when the phone is close to your face during a call, which turns off the screen to prevent accidental touch inputs.
 - Ambient Light Sensor: This sensor adjusts the screen brightness based on the lighting conditions in your environment.
 - Magnetometer: This sensor acts like a digital compass and detects magnetic fields, which is useful for navigation and location-based apps.
 - Barometer: This sensor measures changes in air pressure and can be used to track elevation changes, weather patterns, and more.
 - Fingerprint Sensor: Your fingerprints are captured and stored by this sensor so that you can unlock your phone and authenticate transactions.
-

AMBIENT LIGHT EXPERIMENT

In my experiment using the Science Journal app, I turned off the light, took it near the bulb and kept it in the center of the room when the light is turned on while the Ambient light sensor was on and it showed fluctuations. It seemed that the app was able to detect changes in the light as moved to and fro from the light source. It was very interesting to see how this works and it gave me a glimpse into the power of technology.



**THANK
YOU**