

FINAL PROJECT

MVP PROGRESS:

3D MODEL



INPUT DEVICE HealthCare Clinic

Kassia's Tuesday Section
10.10.2023

TODAY'S LAB

We are setting up a healthcare clinic with limited resources. We brought our Arduino board, some resistors, basic sensors, copper tape, and a 3D printer. Let's make some basic devices to help with our wellness check-ups!



1

Voltage Divider
Sensors

2

IR Heart Rate
Monitor

3

Capacitive
Sensors

4

Breathing Monitor

5

Object-Oriented
Programming

VOLTAGE DIVIDERS

EXAMPLES

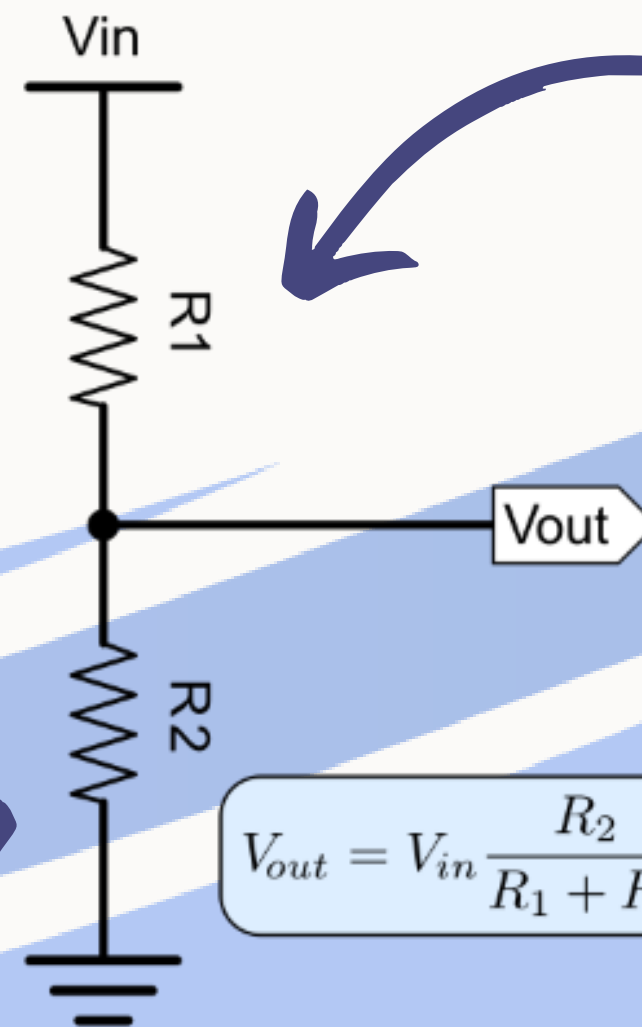
- Light
- IR
- Force/Flex
- Temperature



WHAT IS IT?

- It is a circuit setup that generates an output voltage that is a fraction of the input voltage
- A simple example is 2 resistors in series, with the output reading in between them

Replace with 'resistive' sensor. It's **resistance** changes based off of the thing that it is sensing



Replace with 'transistor-type' sensor. It's changes output **current** based off of the thing that it is sensing

$$V_{out} = V_{in} \frac{R_2}{R_1 + R_2}$$

BUILD A HEART RATE MONITOR



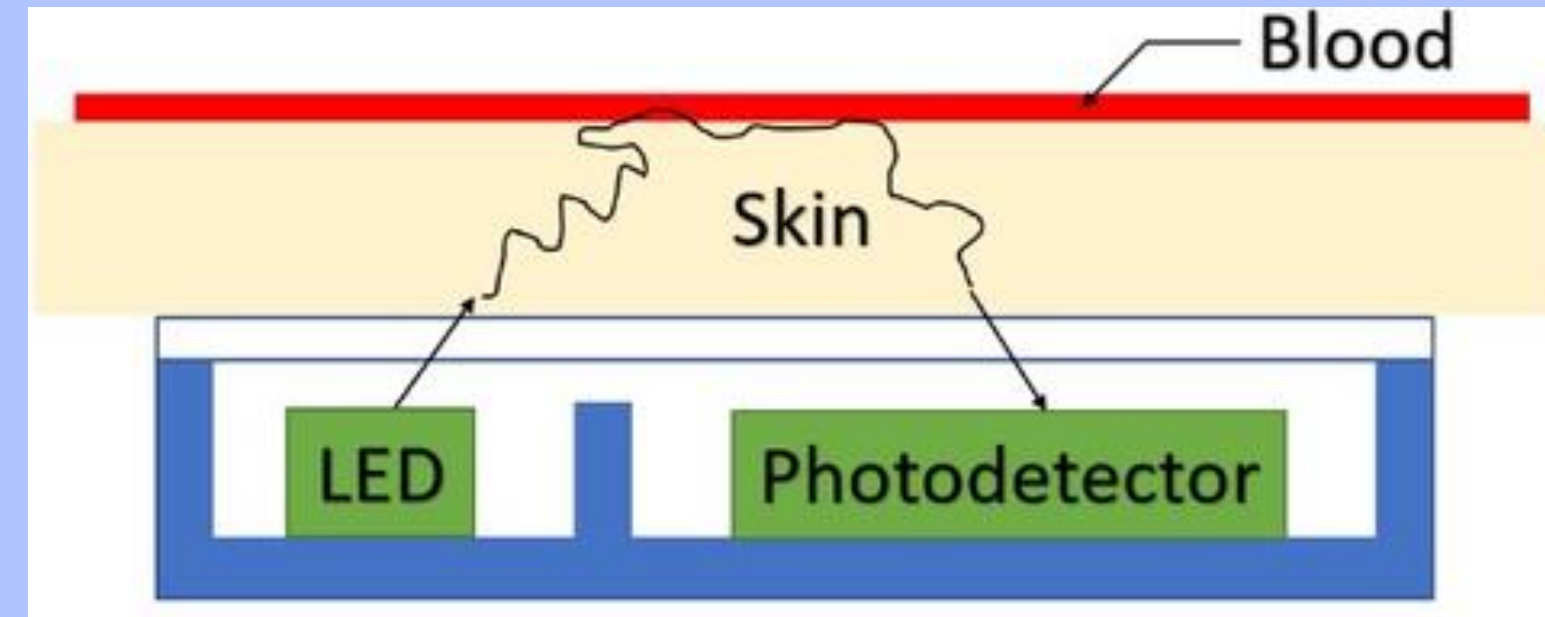
PULSE OXIMETRY

01

PULSE (FOCUS OF TODAY)

When the heart beats, more blood is pumped into the finger, making the finger absorb more light.

With more light absorbed, less light is being reflected to the detector.



02

OXYGEN SATURATION

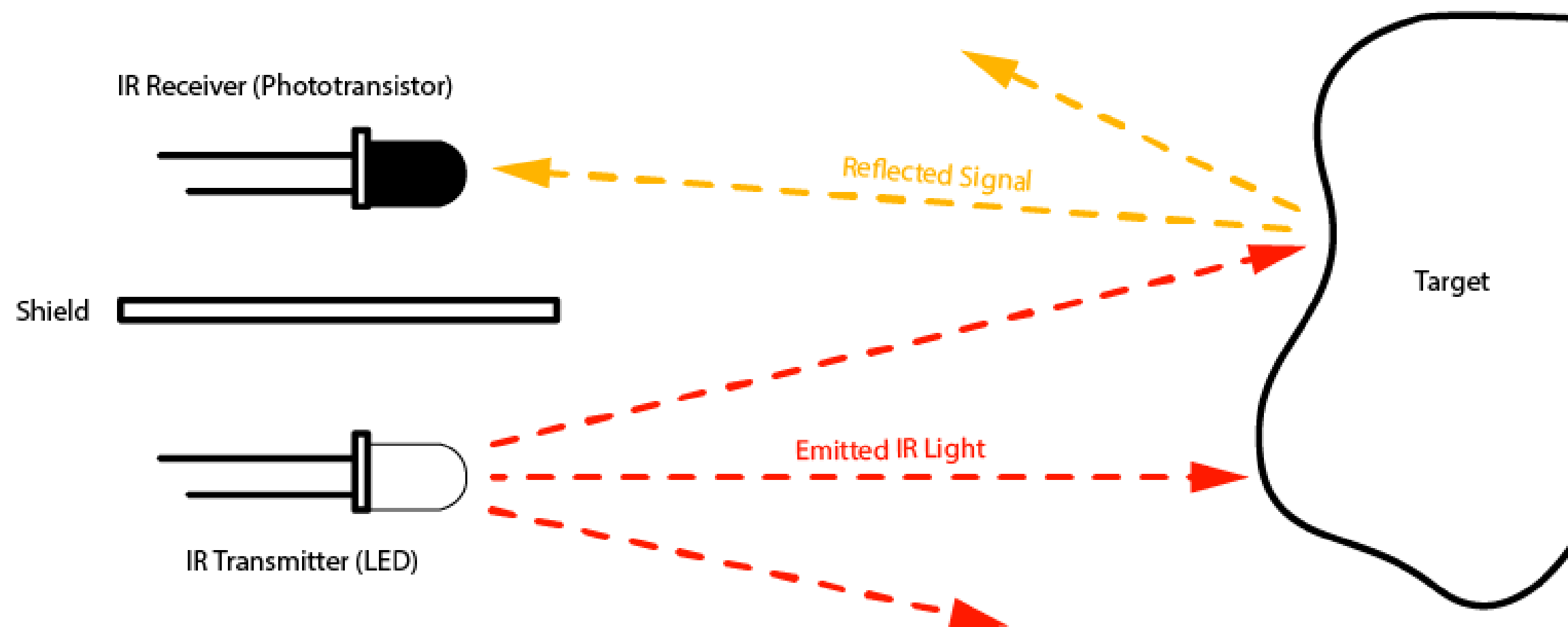
Deoxygenated and oxygenated hemoglobin absorb light at different wavelengths

Shine a red led on the finger, the light travels through your finger to the sensor.

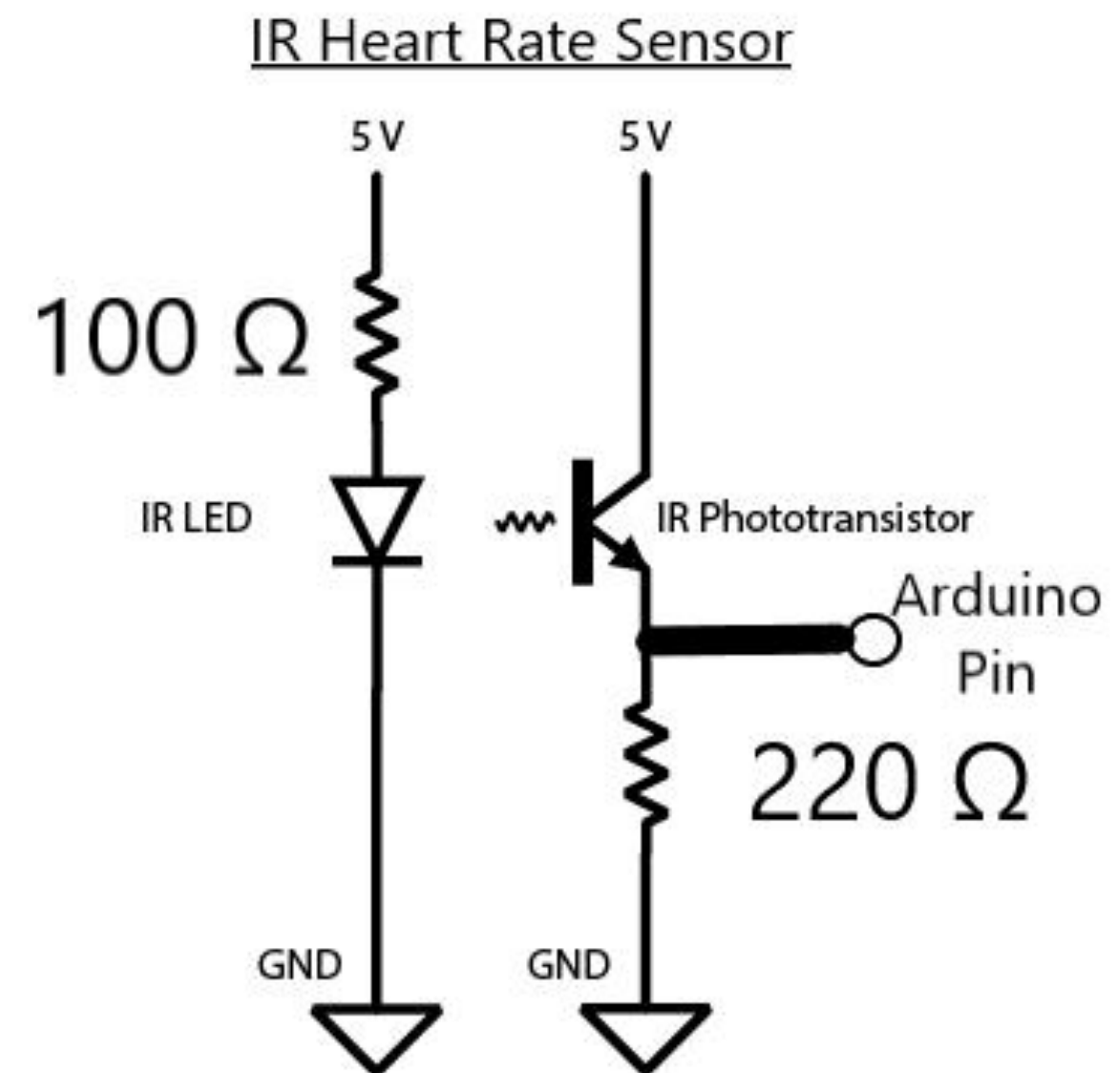
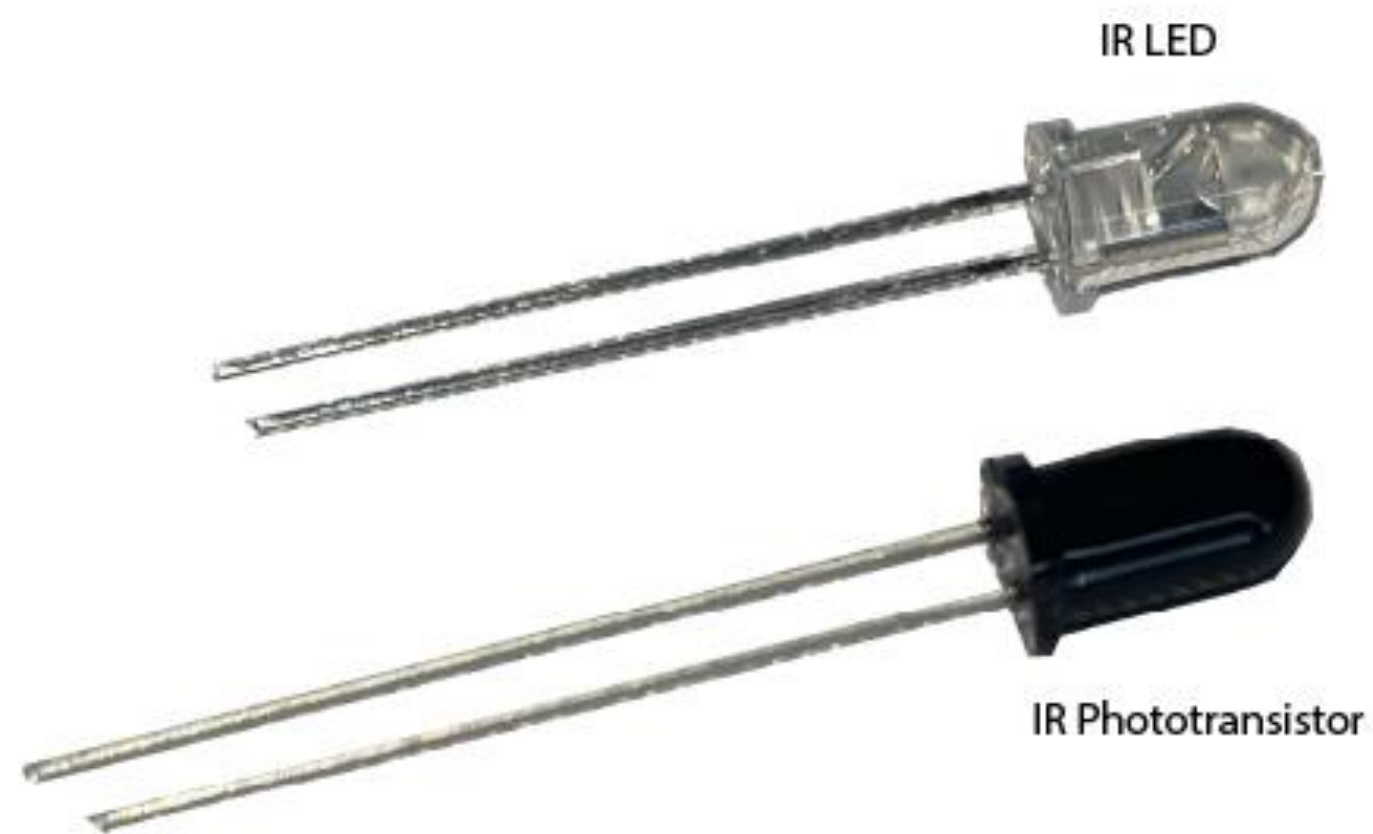
Sensor algorithm converts sensor value to amount of oxygen in blood.



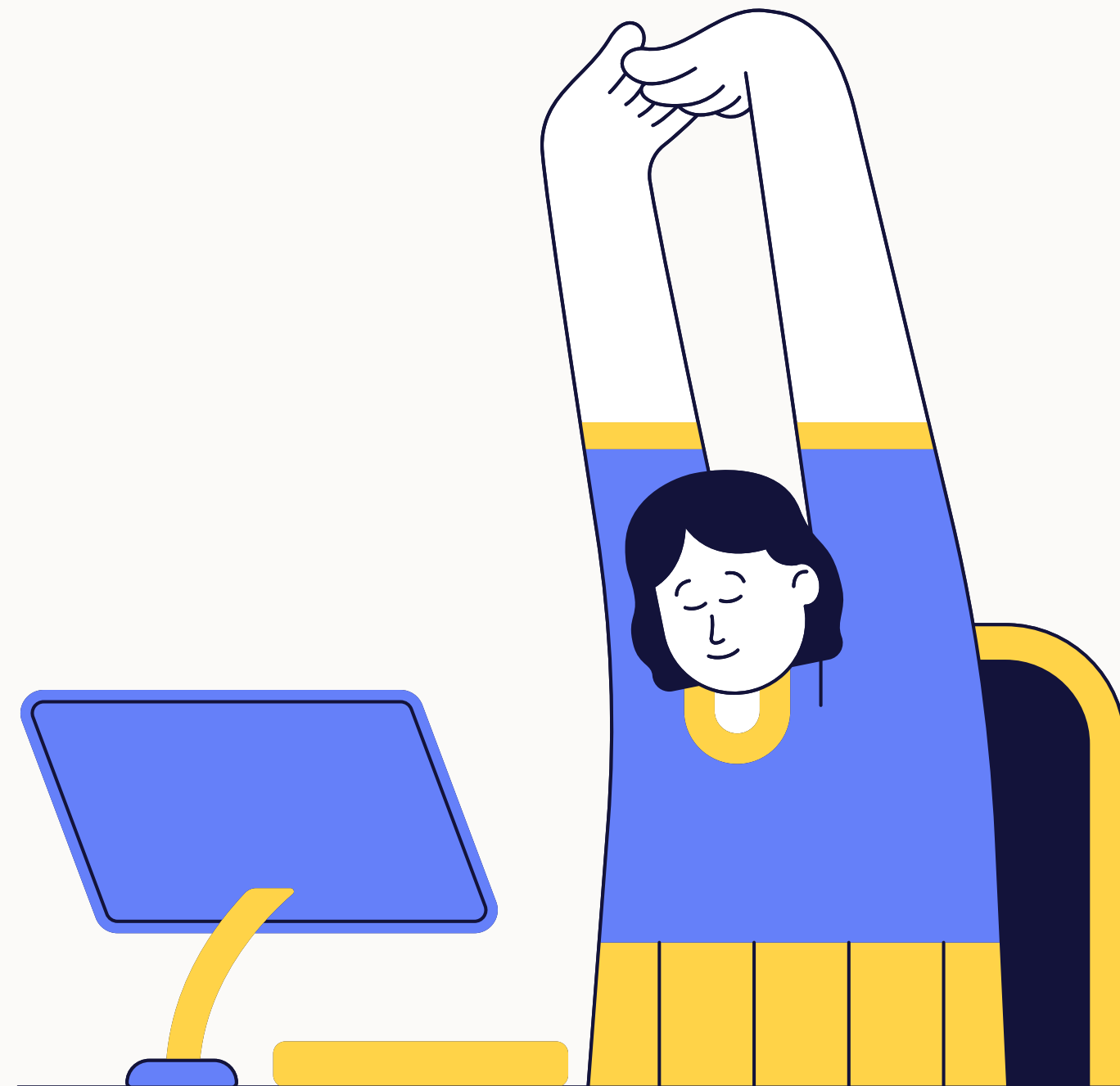
IR SENSORS



IR SENSORS



LET'S CODE

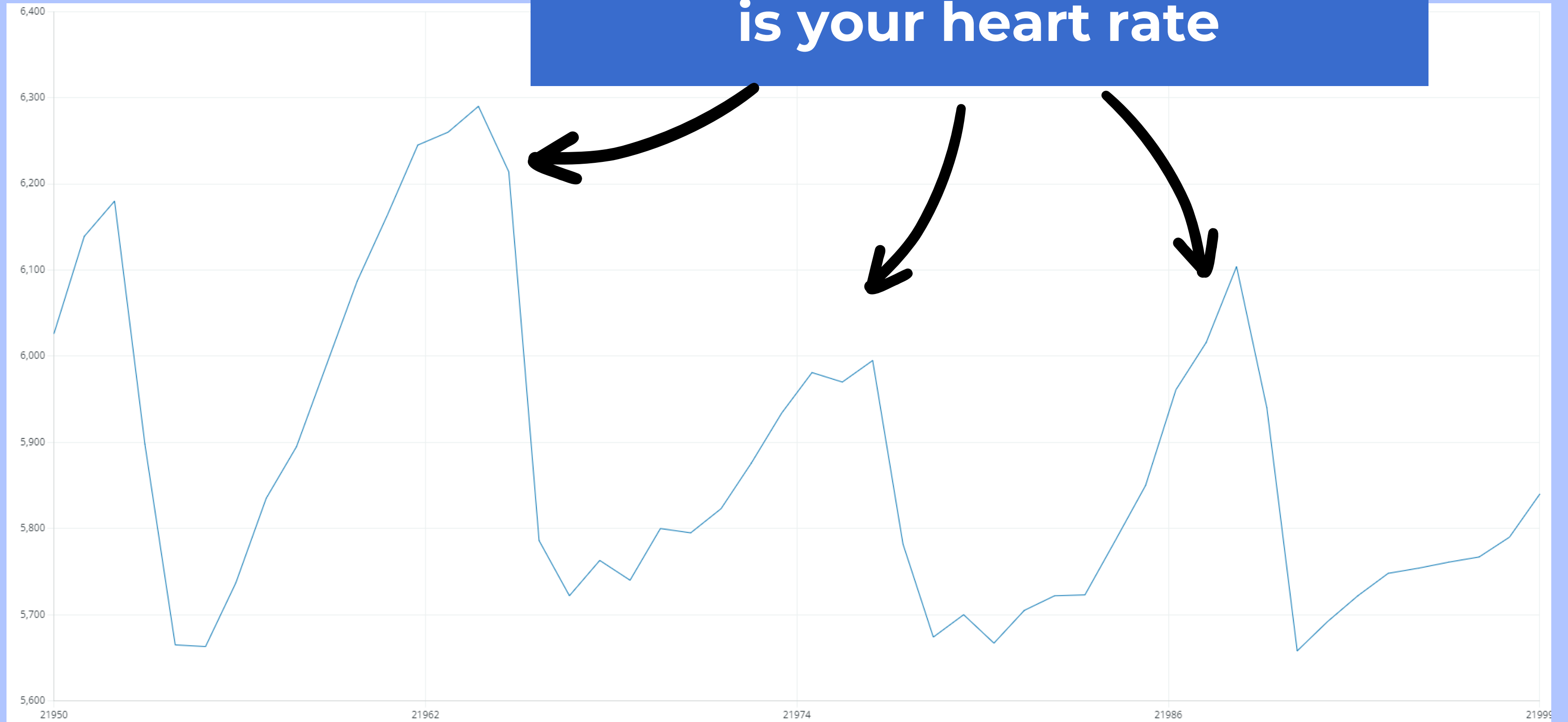


CODE

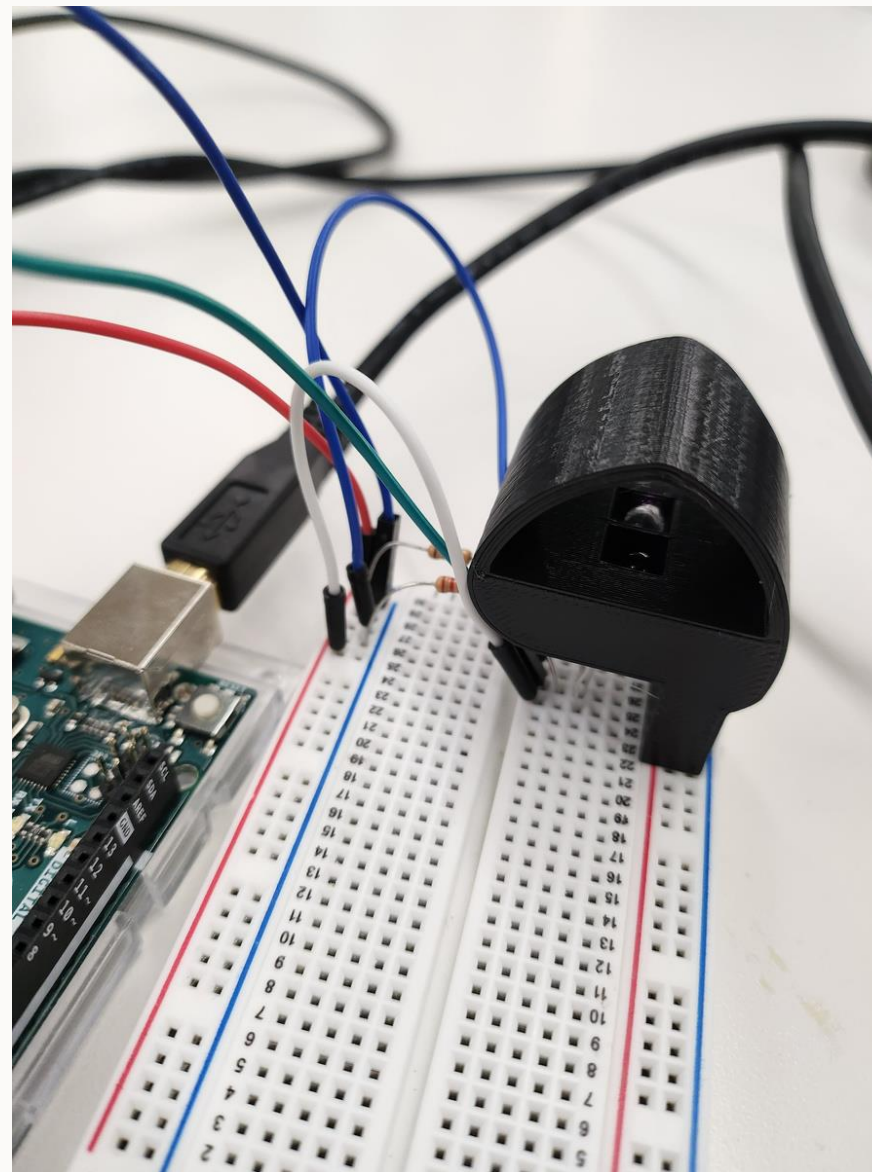
```
1  int sum;
2  int N_samples = 500;
3  int cur_val;
4
5  void setup() {
6      Serial.begin(9600);
7  }
8
9  void loop() {
10     sum = 0;
11
12     //Add N_samples of data together to smooth the signal
13     for (int i=0; i<N_samples; i++){
14         cur_val = analogRead(A0);
15         sum += cur_val;
16     }
17     Serial.println(sum);
18 }
```

OUTPUT

The frequency of these peaks
is your heart rate



3D PRINT A CASE

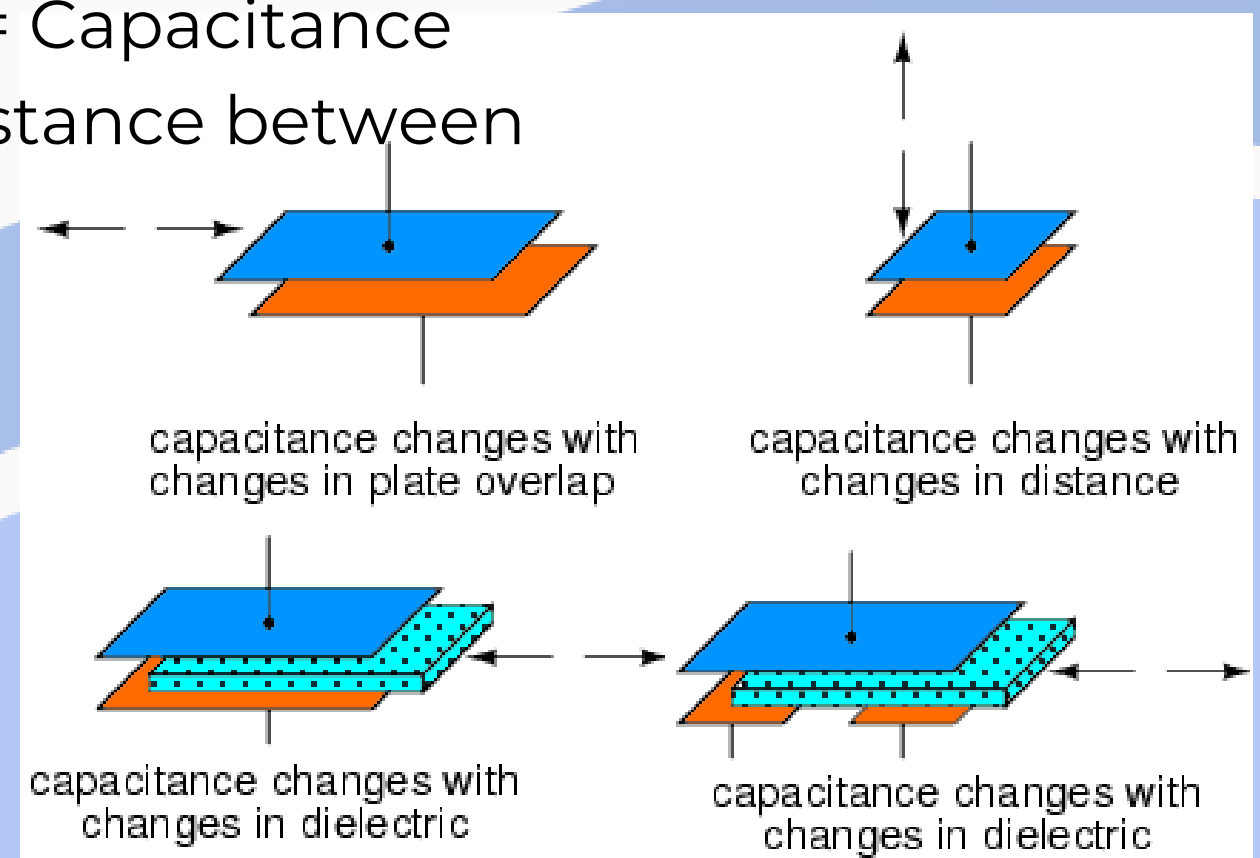
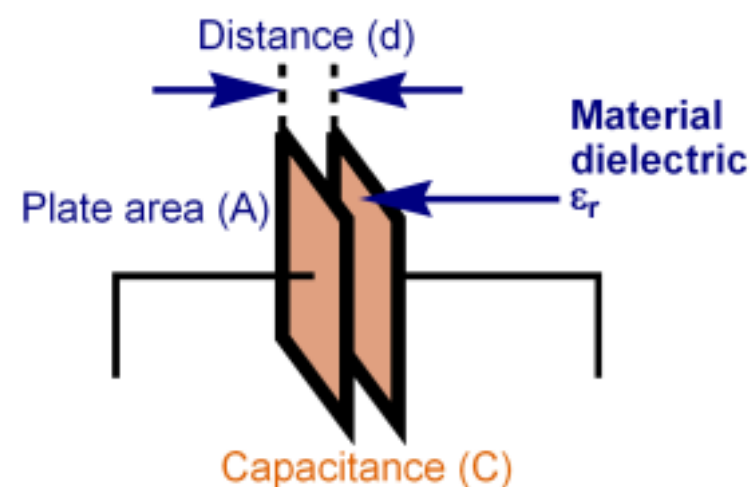


CAPACITIVE SENSORS



WHAT IS IT?

- Simply, two conductors with a dielectric material between them
- Conductor: a substance or material that allows electricity to flow through it
- Dielectric: a substance or material that is a poor conductor
- When there is a voltage across the plates, an electric field forms between them, and we store electrical energy = Capacitance
- Capacitance changes with size of plates and distance between plates

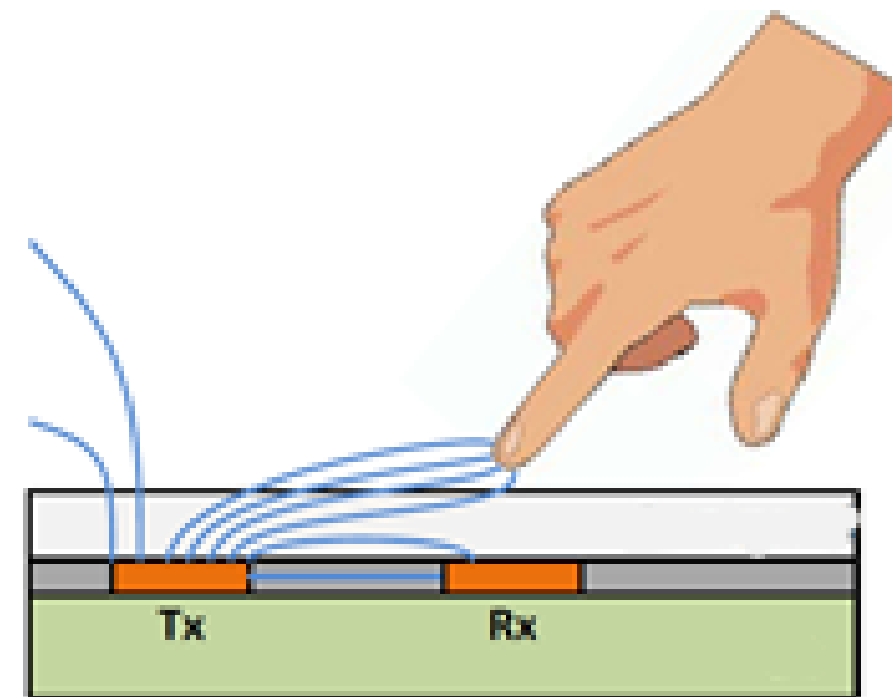
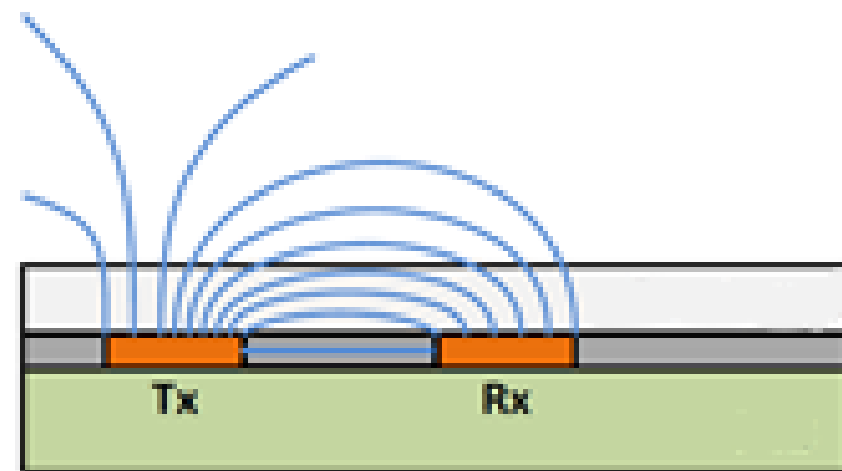


CAPACITIVE SENSORS



HOW IT WORKS?

- By changing the capacitance through distance, area, or dielectric material
- Send pulses of voltage to charge and discharge one of the conductors and see what the other conductor senses. If there is a change in capacitance, it will 'sense' this change.



- TX: Transmitter
- RX: Receiver

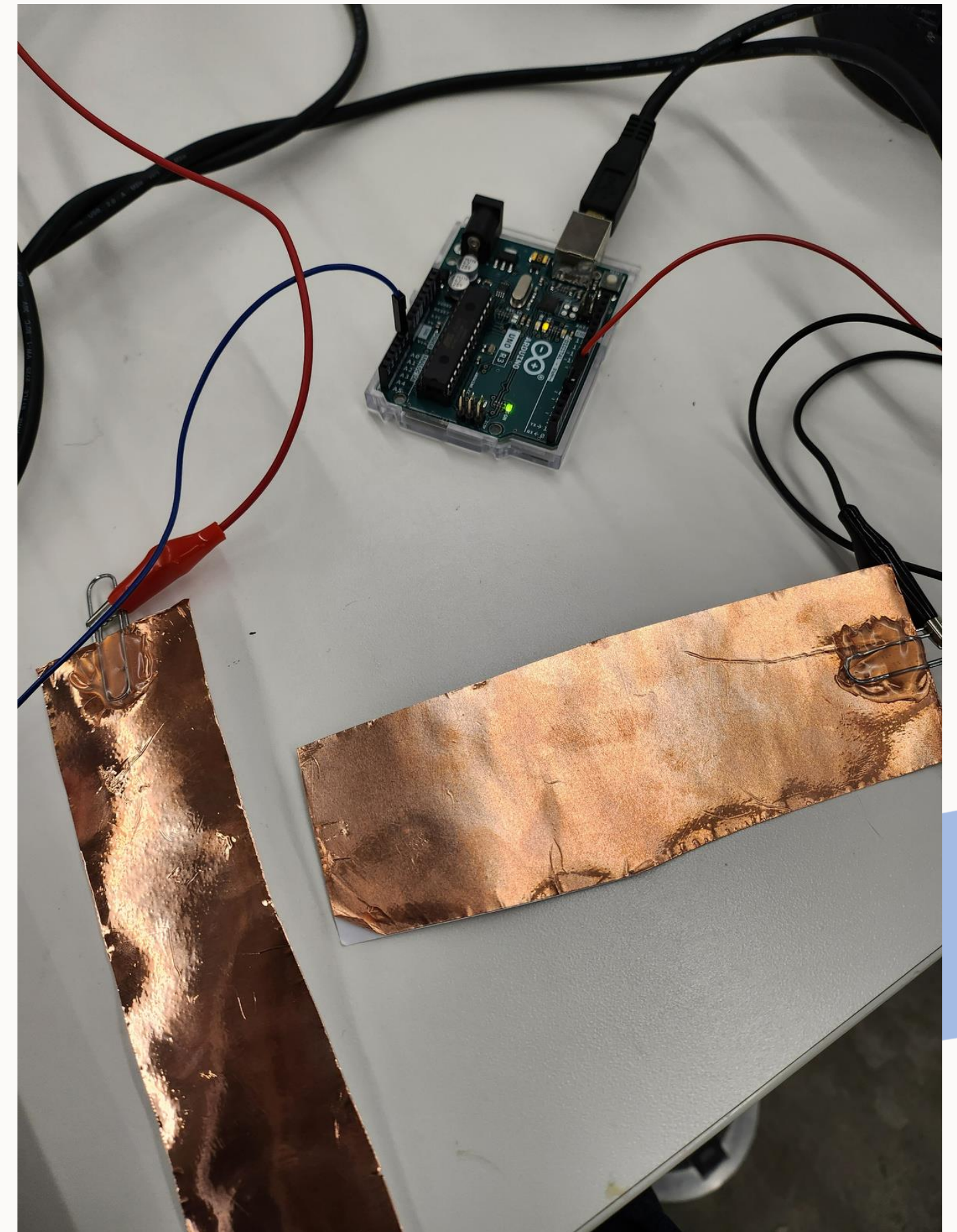
BUILD A BREATHING MONITOR



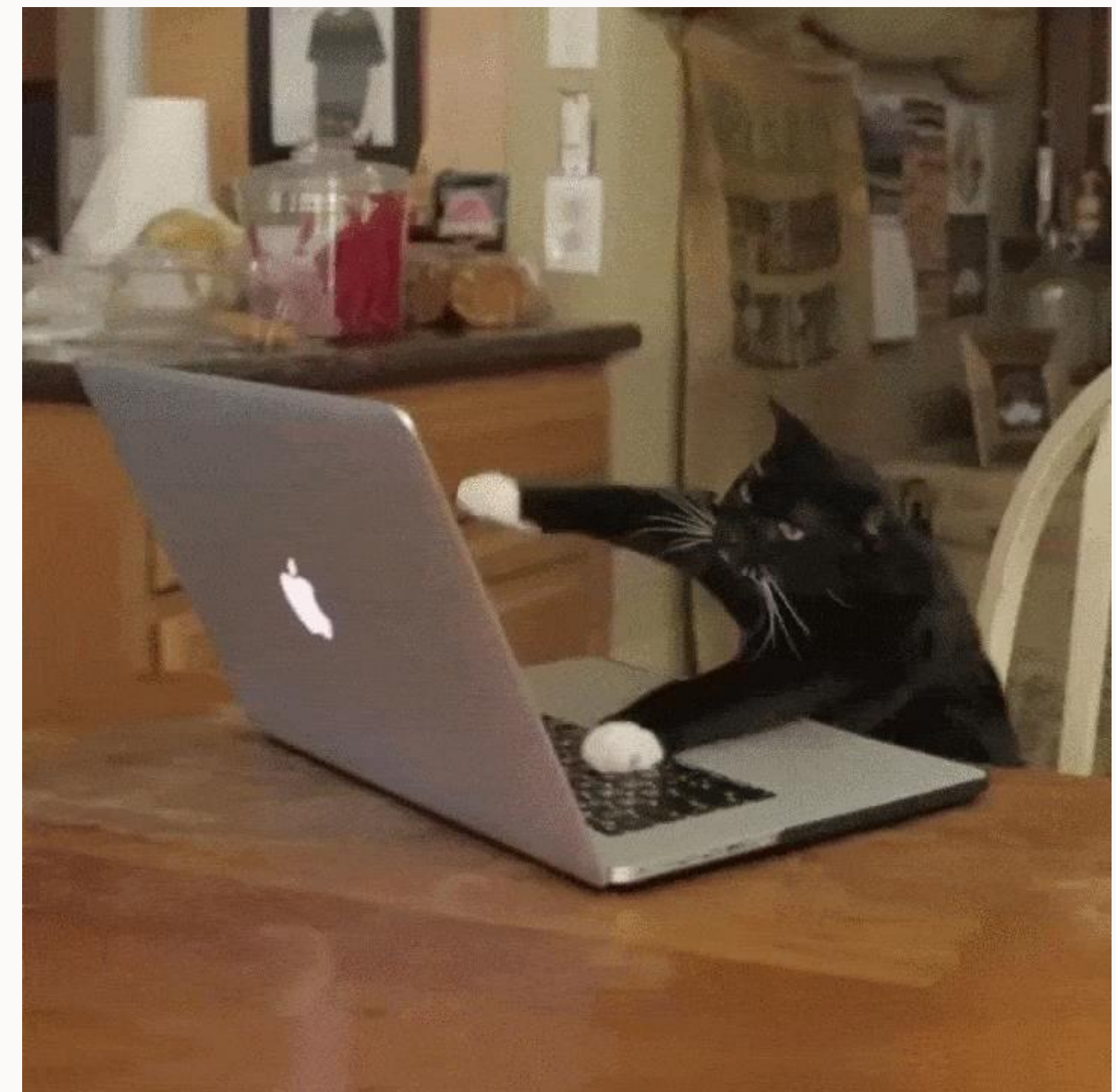
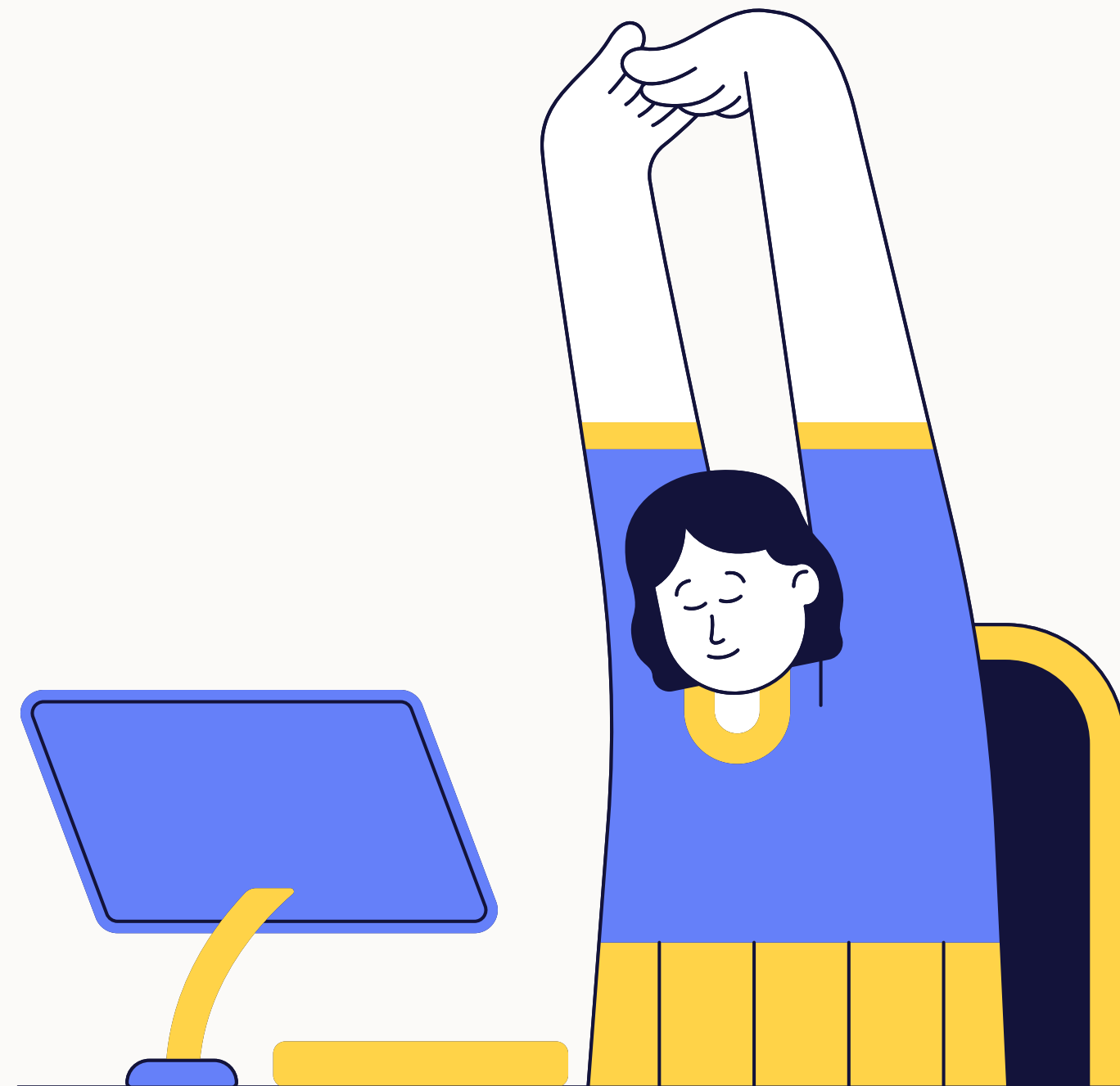
CAPACITIVE SENSORS

One copper tape plugged into A0, this is the receiver

One copper tape plugged into pin 4, this is the transmitter



LET'S CODE



CODE

```
1  int tx_pin = 4;
2  int rx_pin = A0;
3  int read_high;
4  int read_low;
5  int read_result;
6  int sum;
7  int N_samples = 100;
8
9  void setup() {
10     pinMode(tx_pin, OUTPUT);
11     Serial.begin(9600);
12 }
13
14 void loop() {
15     sum = 0;
16
17     for (int i=0; i<N_samples; i++){
18         digitalWrite(tx_pin, HIGH);
19         read_high = analogRead(rx_pin);
20         digitalWrite(tx_pin, LOW);
21         read_low = analogRead(rx_pin);
22         read_result = read_high - read_low;
23         sum += read_result;
24     }
25     Serial.println(sum);
26
27 }
```

OBJECT ORIENTED PROGRAMMING



DOING THINGS AT THE SAME TIME

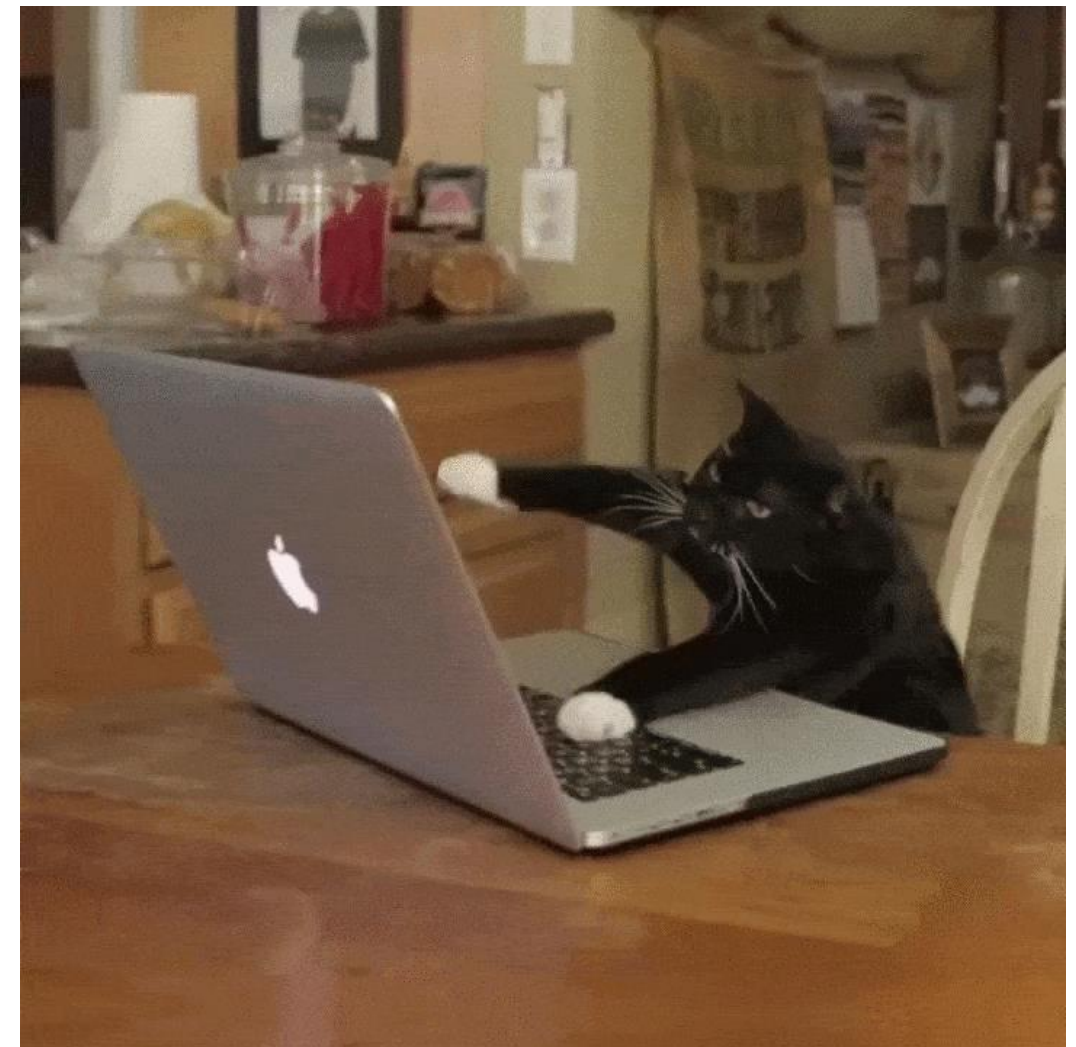
Not too bad, however, since there are for loops involved, and Arduino does tasks from top to bottom, the readings will be off!

Also, this is for 1 patient, what if I had 5 patients to measure at once? The code would be a mess @_@

```
1  int tx_pin = 4;
2  int rx_pin = A0;
3  int read_high;
4  int read_low;
5  int read_result;
6  int sum_breathing;
7  int N_samples = 500;
8
9  int sum_heart;
10 int cur_val;
11
12 void setup() {
13     pinMode(tx_pin, OUTPUT);
14     Serial.begin(9600);
15 }
16
17 void loop() {
18     sum_breathing = 0;
19
20     for (int i=0; i<N_samples; i++){
21         digitalWrite(tx_pin, HIGH);
22         read_high = analogRead(rx_pin);
23         digitalWrite(tx_pin, LOW);
24         read_low = analogRead(rx_pin);
25         read_result = read_high - read_low;
26         sum_breathing += read_result;
27     }
28     Serial.print(sum_breathing);
29     Serial.print(",");
30
31     sum_heart = 0;
32     for (int i=0; i<N_samples; i++){
33         cur_val = analogRead(A1);
34         sum_heart += cur_val;
35     }
36
37     Serial.println(sum_heart);
38
39 }
```

LET'S CODE: A CLASSY SOLUTION

We will create the heart rate monitor Class today. I challenge you to do the breathing monitor Class on your own~



DEFINE THE CLASS

```
1  class HeartRate{
2      //Class Member Variables
3      int N_samples;
4      int sum;
5      int cur_val;
6      int pin;
7  };
8
```

ADD A CONSTRUCTOR

```
1  class HeartRate{
2      //Class Member Variables
3      int N_samples;
4      int sum;
5      int cur_val;
6      int HRpin;
7
8      //Constructor - creates a HeartRate
9      //and initializes the member variables
10     public:
11     HeartRate(int pin){
12         HRpin = pin;
13         pinMode(HRpin,OUTPUT);
14         N_samples = 500;
15     }
16
17 };
```


ADD A FUNCTION

```
1  class HeartRate{
2      //Class Member Variables
3      int N_samples;
4      int sum;
5      int cur_val;
6      int HRpin;
7
8      //Constructor - creates a HeartRate
9      //and initializes the member variables
10     public:
11     HeartRate(int pin){
12         HRpin = pin;
13         pinMode(HRpin,OUTPUT);
14         N_samples = 500;
15     }
16
17     int read(){
18         sum = 0;
19         for (int i=0; i<N_samples; i++){
20             cur_val = analogRead(HRpin);
21             sum += cur_val;
22         }
23         return sum;
24     }
25 };
```


NOW USE THE CLASS

```
1 class HeartRate{
2     //Class Member Variables
3     int N_samples;
4     int sum;
5     int cur_val;
6     int HRpin;
7
8     //Constructor - creates a HeartRate
9     //and initializes the member variables
10    public:
11    HeartRate(int pin){
12        HRpin = pin;
13        pinMode(HRpin,OUTPUT);
14        N_samples = 500;
15    }
16
17    int read(){
18        sum = 0;
19        for (int i=0; i<N_samples; i++){
20            cur_val = analogRead(HRpin);
21            sum += cur_val;
22        }
23        return sum;
24    }
25};
```

```
27 HeartRate patient1(A0);
28 HeartRate patient2(A1);
29
30 int p1;
31 int p2;
32
33 void setup() {
34     Serial.begin(9600);
35 }
36
37 void loop() {
38     p1 = patient1.read();
39     p2 = patient2.read();
40
41     Serial.print(p1);
42     Serial.print(",");
43     Serial.print(p2);
44     Serial.println(",");
45 }
```

A light blue, curved, abstract shape at the top of the slide, resembling a stylized wave or a partial circle.

**WHAT INPUT SENSORS WILL
YOU NEED IN YOUR FINAL
PROJECT?**

SOURCES



https://nathanmelenbrink.github.io/ps70/06_input/index.html

<https://learn.adafruit.com/multi-tasking-the-arduino-part-1/overview>