

EWH EDUCATION TEAM PRESENTS

ARDUINO CRASH COURSE

Learn the basics of Arduino!

COVERED TODAY

A BRIEF OUTLINE

- Who are we?
- What is an Arduino?
- Writing programs for an Arduino
- Arduino Sensors
- Lets Build a Heart Rate Monitor
- What's Next?

What is EWH?

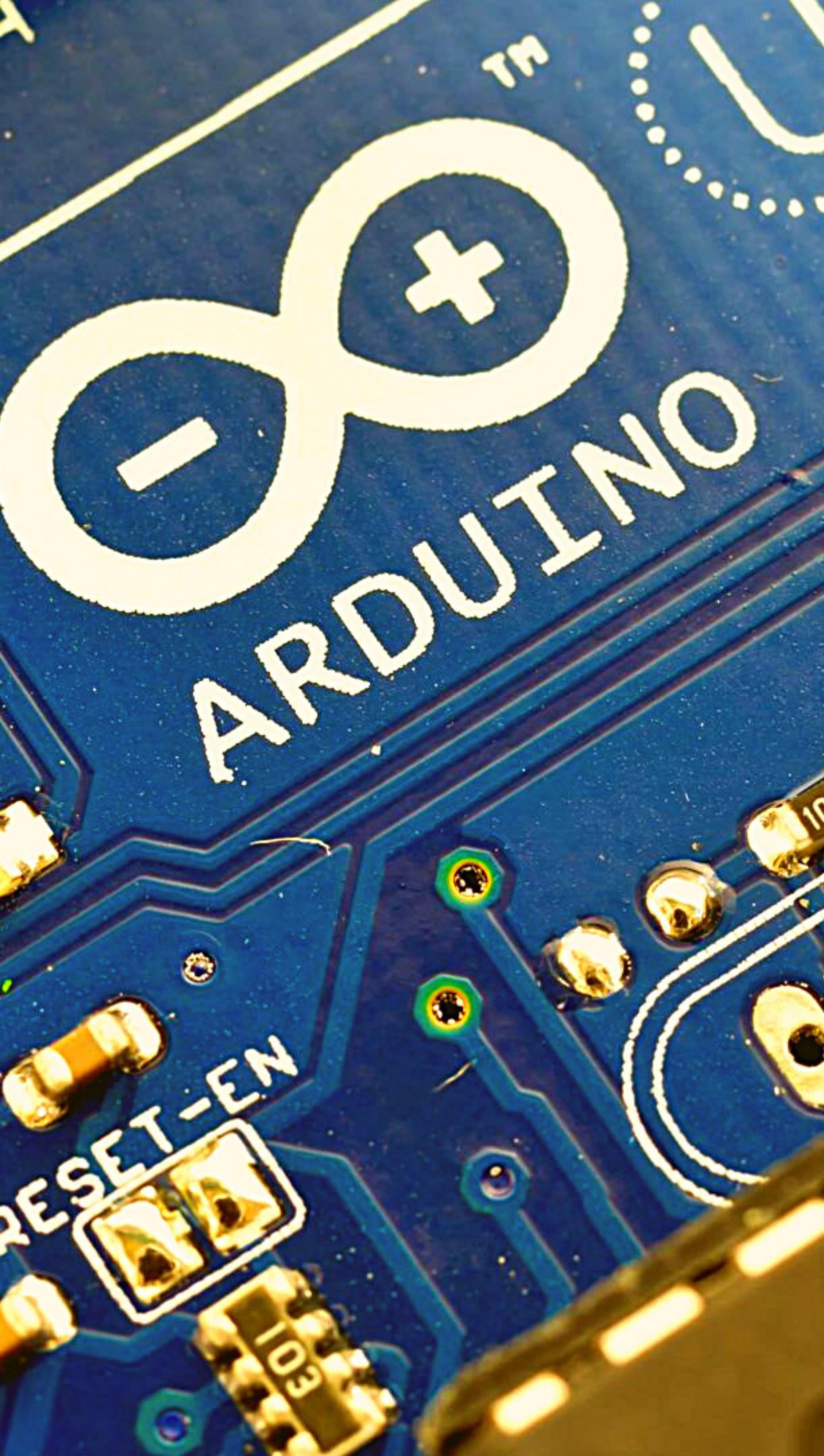


WHAT IS AN ARDUINO?

ITS A MICROCONTROLLER

- Think of it as a bridge between electronic circuits and a computer program

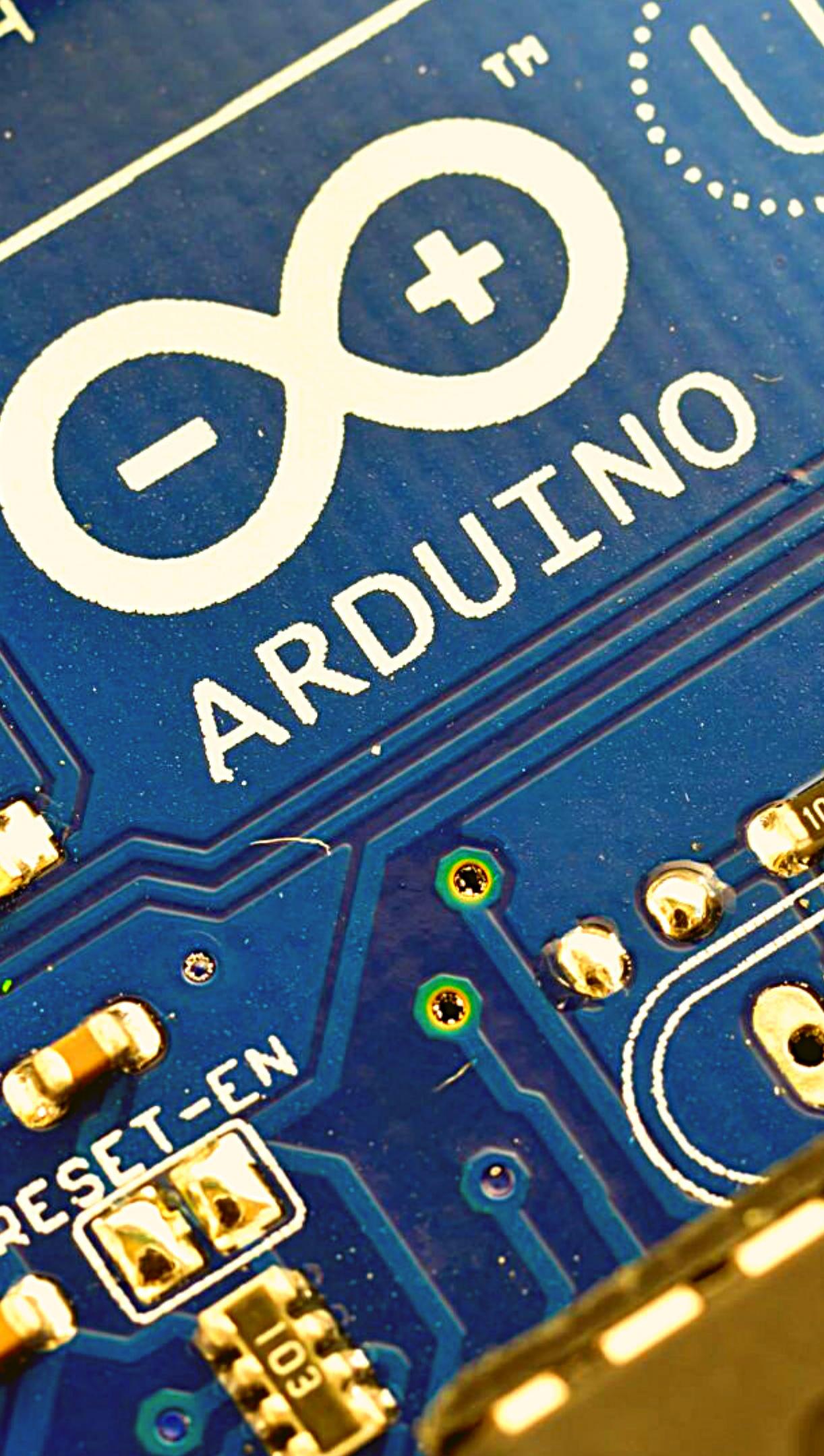
You can make objects that interact with their environment



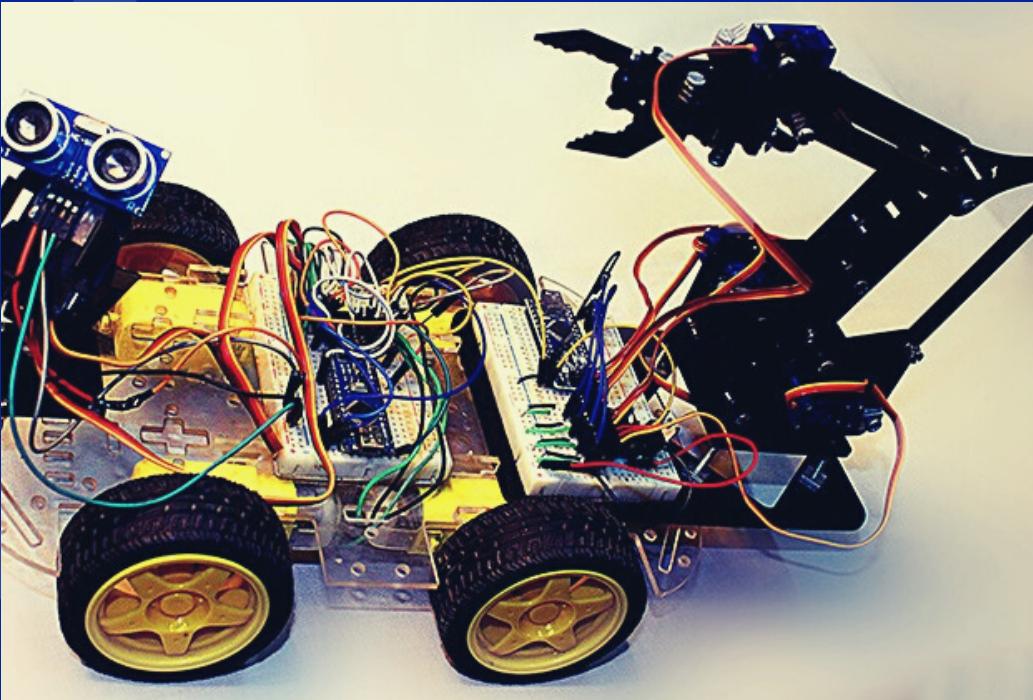
WHY DO WE USE THEM?

THEY ARE VERSATILE

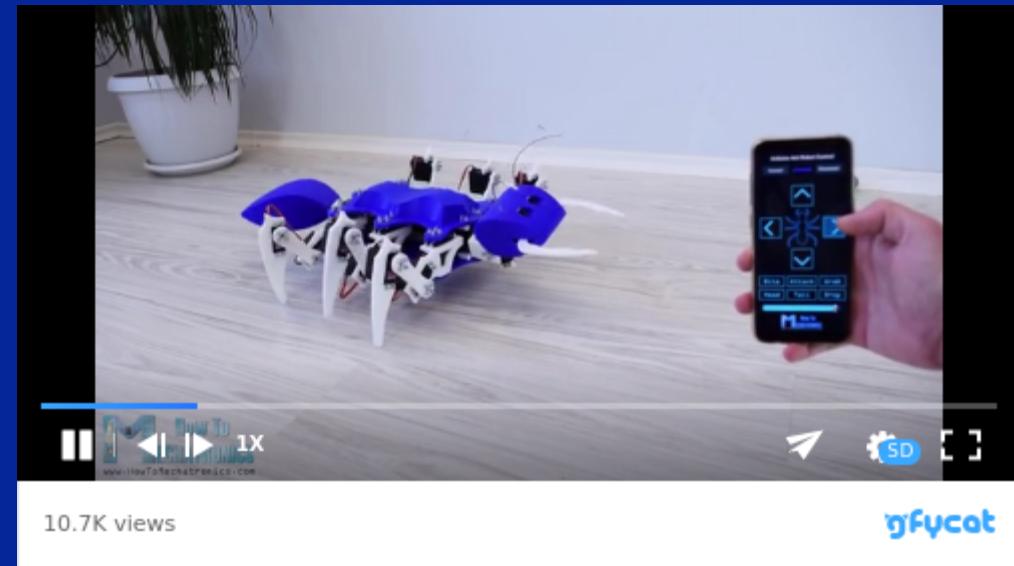
- You can configure them to do almost anything
- Manufacturers make sensors and attachments specifically for the Arduino, so they are easily compatible
- There is a large community around the Arduino, with many people sharing their designs and codes online.



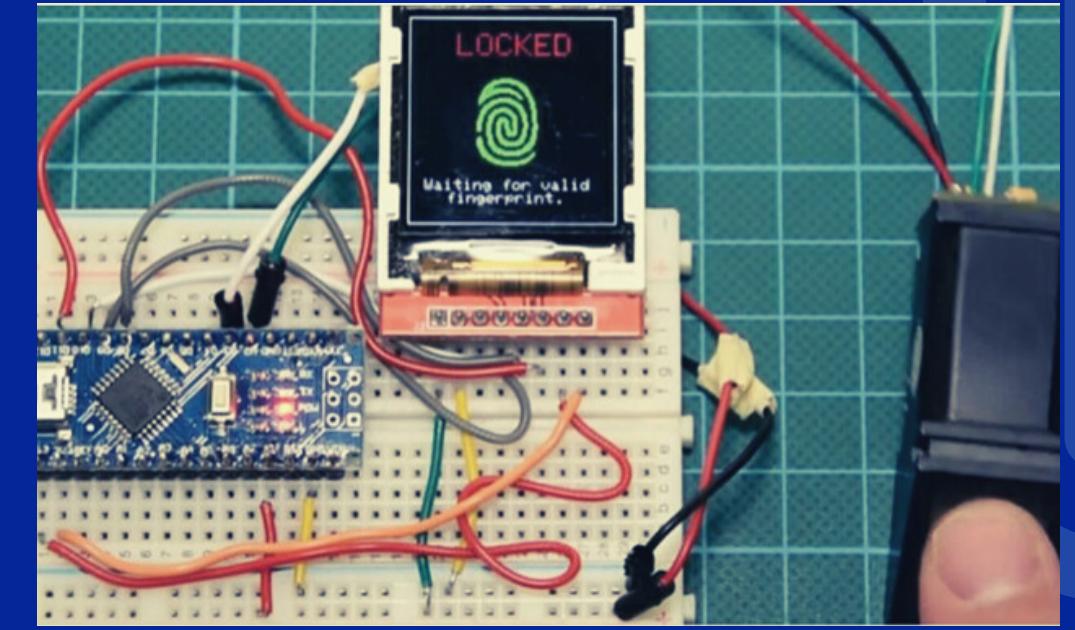
What can I make using an Arduino?



ROBOTS



WHATEVER THIS IS?



SENSORS

WRITING PROGRAMS FOR ARDUINO

CODING BASICS

There are many different types of coding languages like JAVA, PYTHON, SQL and more, but Arduino uses C/C++

All these languages uses different mechanisms to write code, but something they do have in common is:

- Data Types
 - Loops
 - Conditionals



DATA TYPES

- These are the different types of data that are built into the programming language
- Can include Numbers, characters or strings
- These data types are variable so can be changed later on in the program

The data is stored in memory and can be called upon later and updated as operations are done on it

Integers (Or int) take up wo bytes of data in memory, if more precision is needed, it can be stored as a floats (4 bytes) or doubles (8 bytes)

Booleans (True or False) can also be stored when comparing two numbers or characters

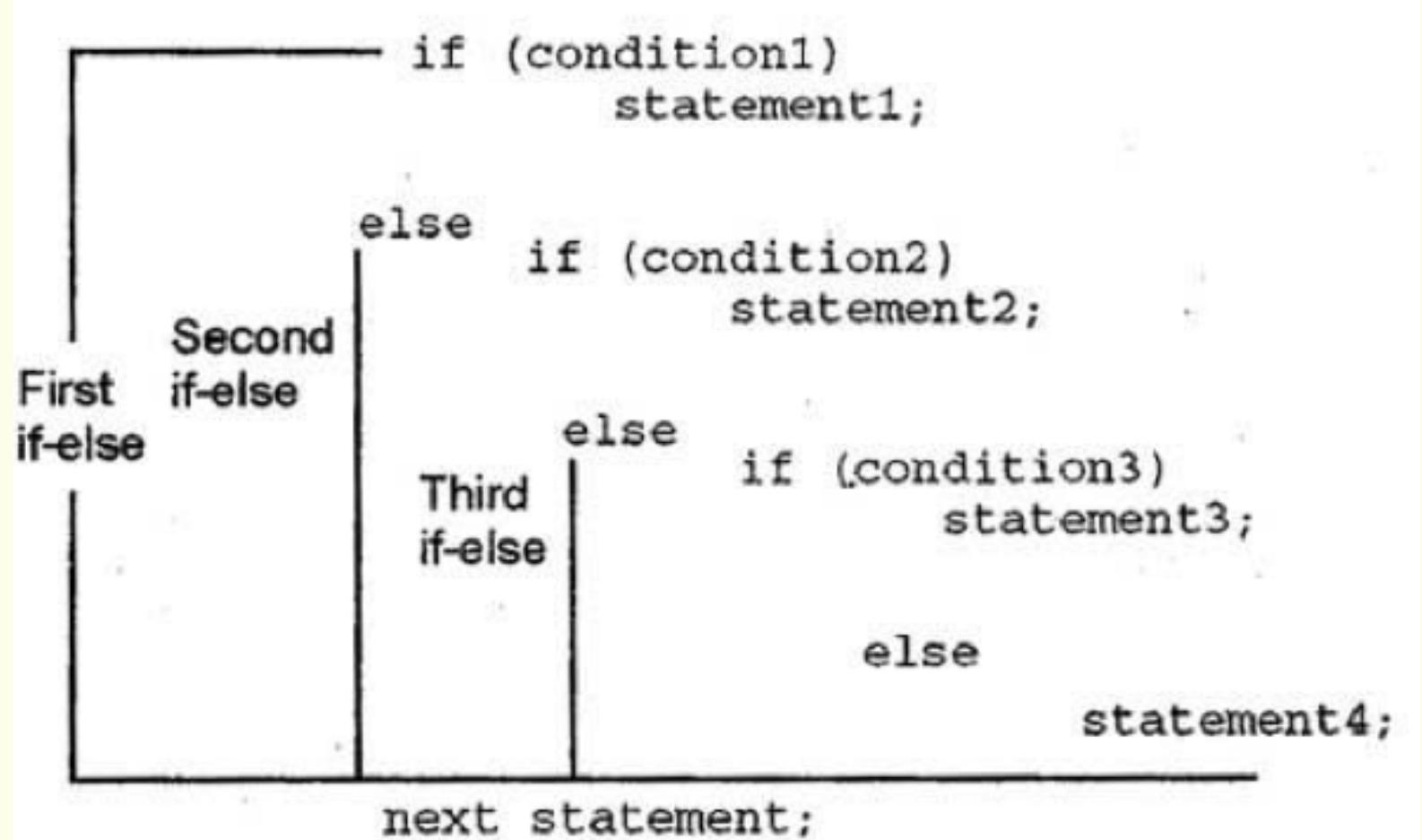
You have to choose a name for each data type you generate such as:

- int number = 8;
- char letter = 'a';



CONDITIONALS

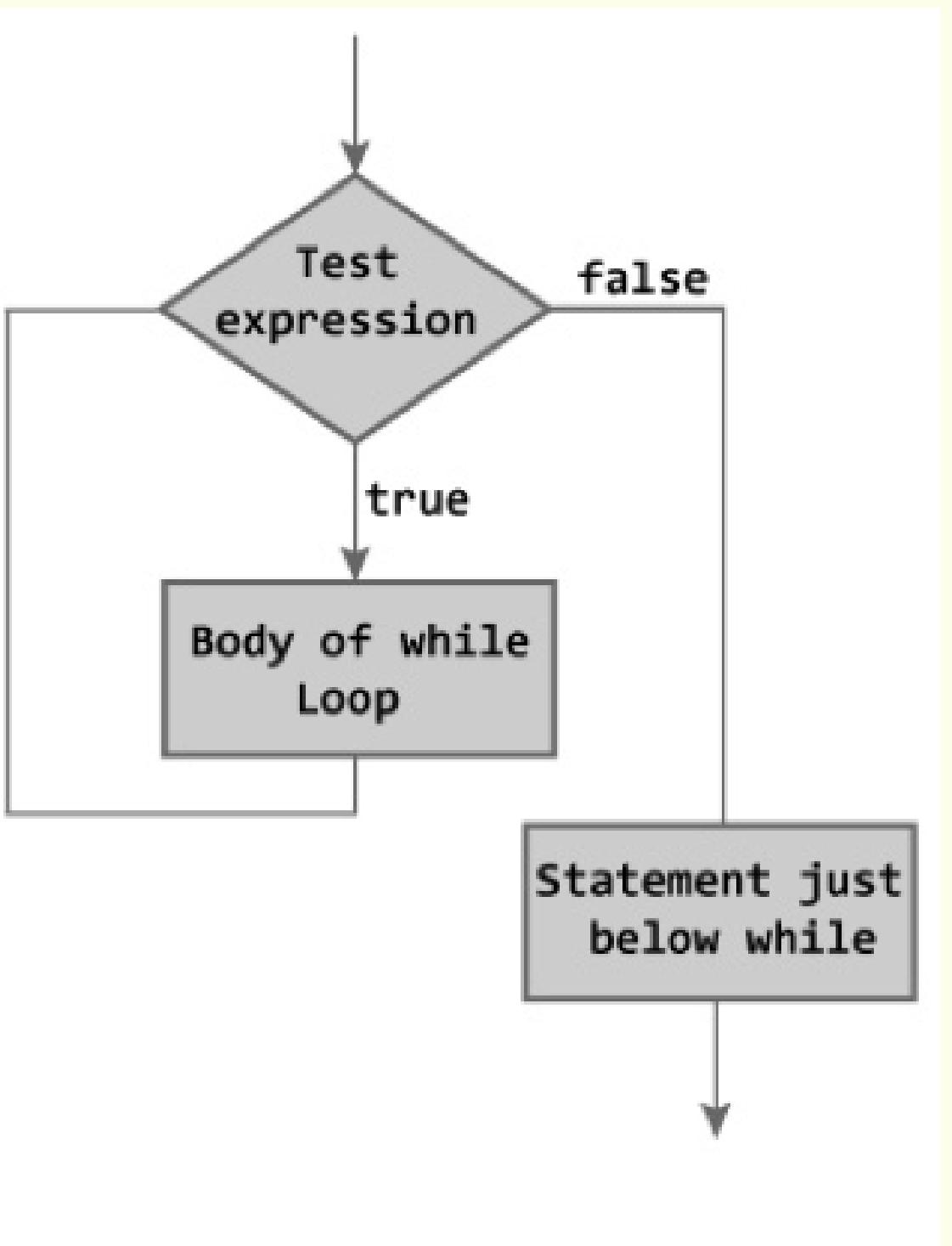
- Also called IF Statements
- Only run code if specific conditions are met.



```
char ch;  
  
cout<<"Enter a Letter:";  
  
cin>>ch;  
  
if( (ch>='A') && (ch<='Z'))  
  
    cout <<"The alphabet is in  
upper case";  
  
else  
  
    if ( (ch>=' a') && (ch<=' z' ) )  
  
        cout<<"The alphabet is in  
lower case";  
  
else  
  
    cout<<"It is not an  
alphabet";  
  
return 0;
```

LOOPS

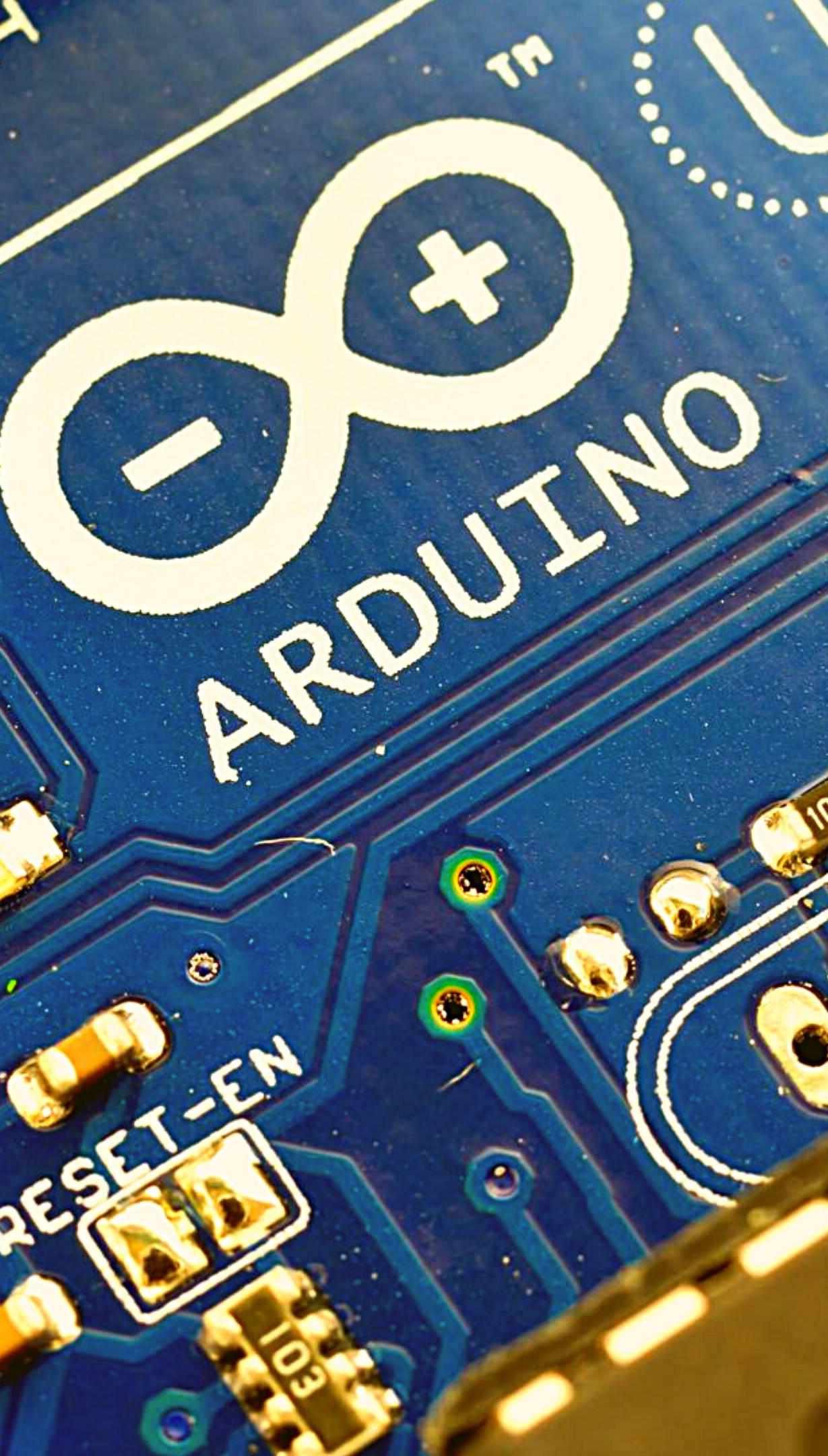
- Continue to execute code in the loop over and over again as long as the specific conditions are met



Arduino Sensors

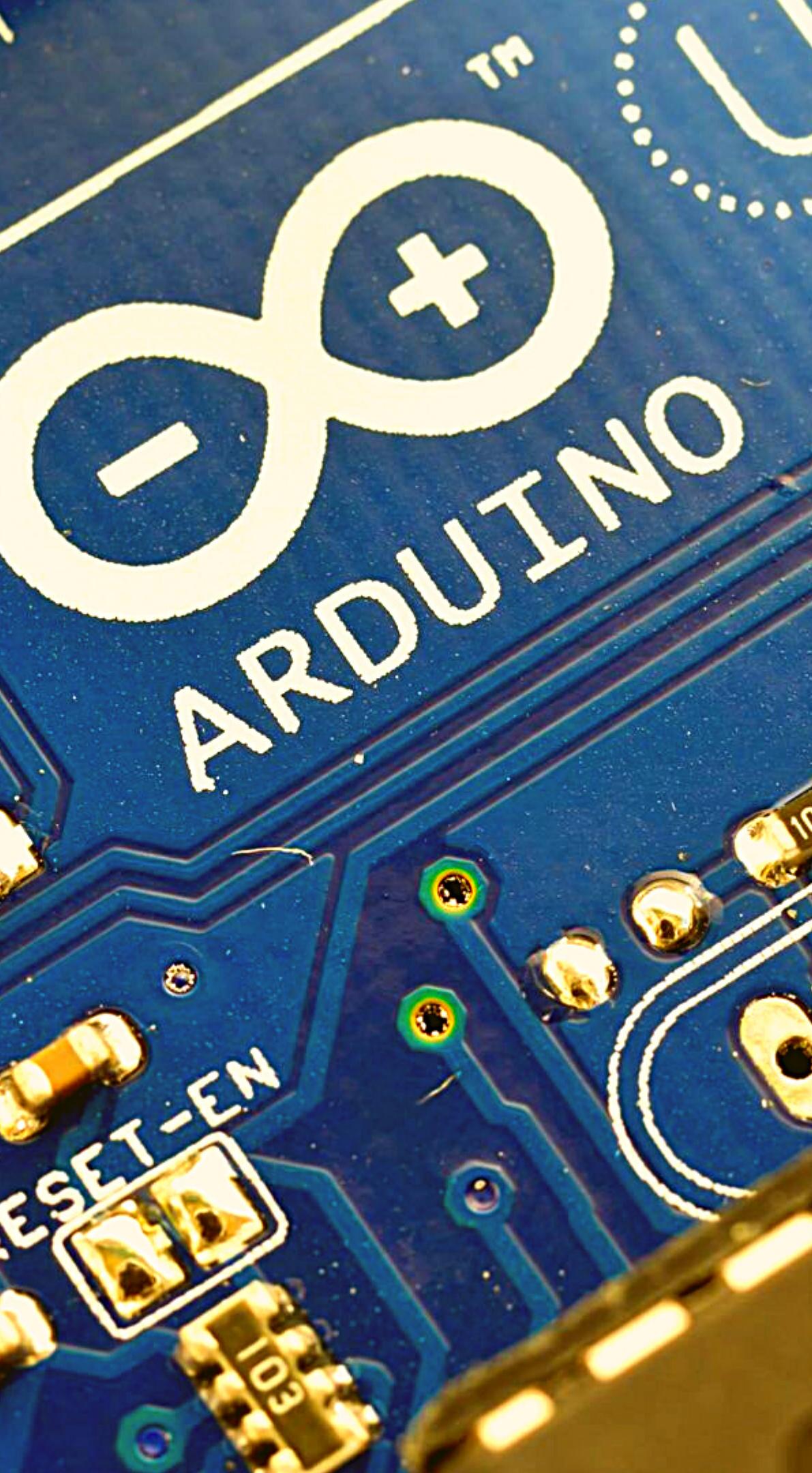
SENSORS

- Sensors collect data to be processed in the Arduino
- These sensors are plugged directly into the Arduino circuit board shown below
- Data collected can be processed and displayed on an LCD or on your computer through Arduino IDE



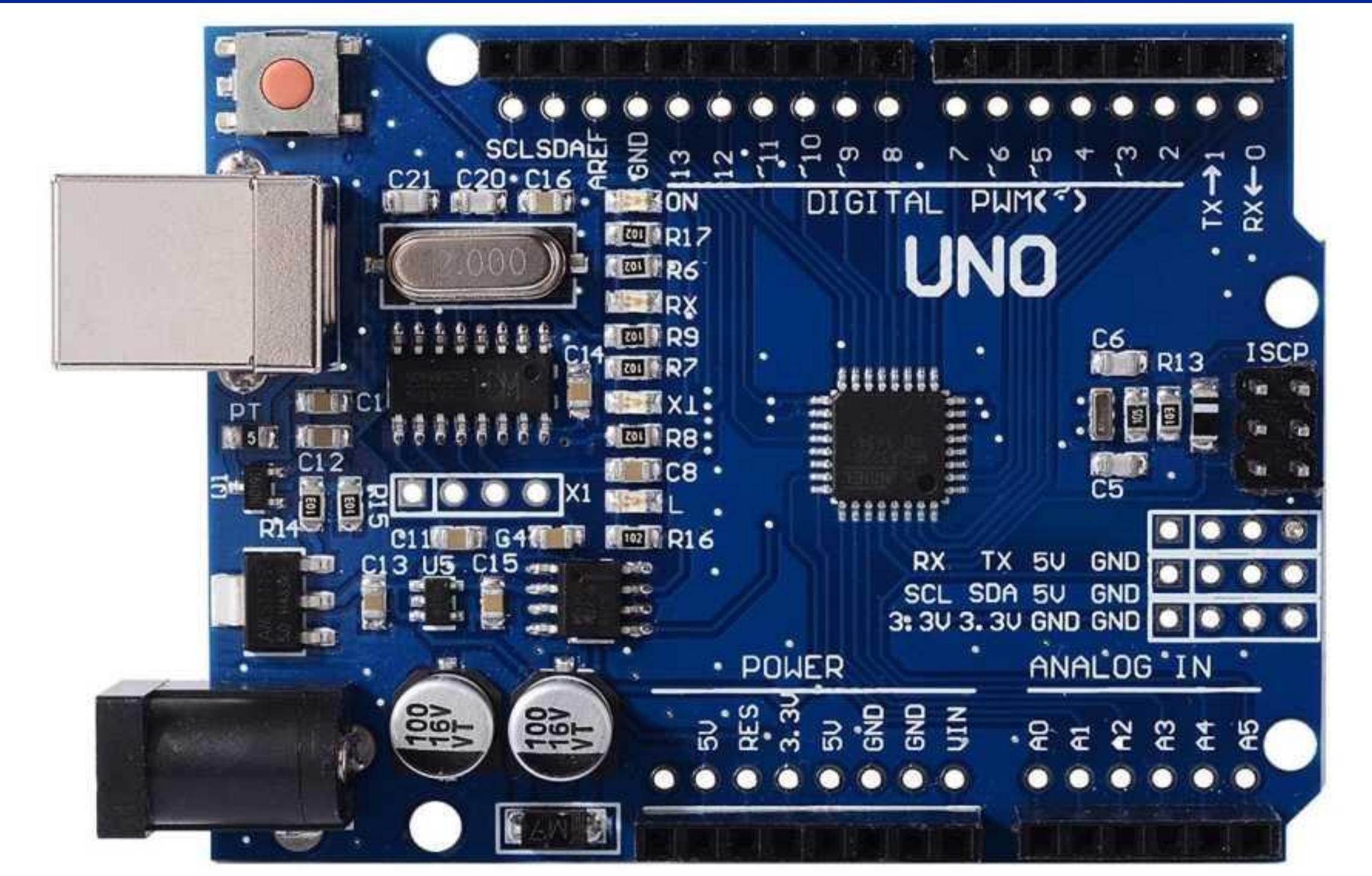
THE HEART RATE SENSOR

- Two main types of heart rate sensors:
 - Electrical
 - Optical
- Electrical sensors are commonly implemented as chest straps
- They're more accurate, but bulky and can't be used underwater
- Optical sensors are commonly implemented in smart watches, phones or Arduino sensors!
- They're less accurate, but they can be used anywhere and are small and convenient



Let's Make a Heart Rate Monitor!

What's on an Arduino?



How to set up the HR sensor

The Arduino IDE

Getting readings from the HR sensor

Why does it . . . suck?

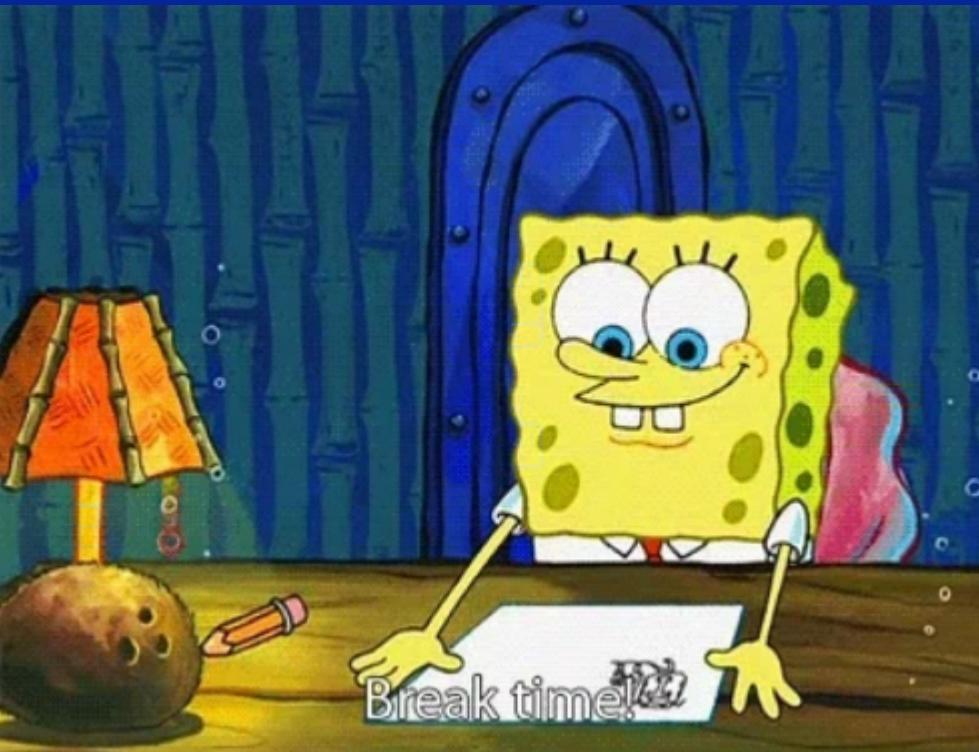


HR SENSOR DOESN'T MEASURE HR

- The sensor isn't actually measuring your heart rate!
- It's just a fancy infrared light detector
- When you place your finger on it, your heartbeat dilates the blood vessels in your finger and blocks less light
- Fluctuations in light creates a pulsing signal
- We need to process the signal to actually get the heart rate!

BREAK!

(5 mins)



GETTING A BETTER OUTPUT

REMOVING NOISE FROM THE SIGNAL

- Noise is basically the error that you have between the signal you want (or predict) and the signal you actually have
- Noise comes from many things: other sources of light, incorrect measurements due to pressure, etc.
- Remove noise by taking an average of the samples every 20ms

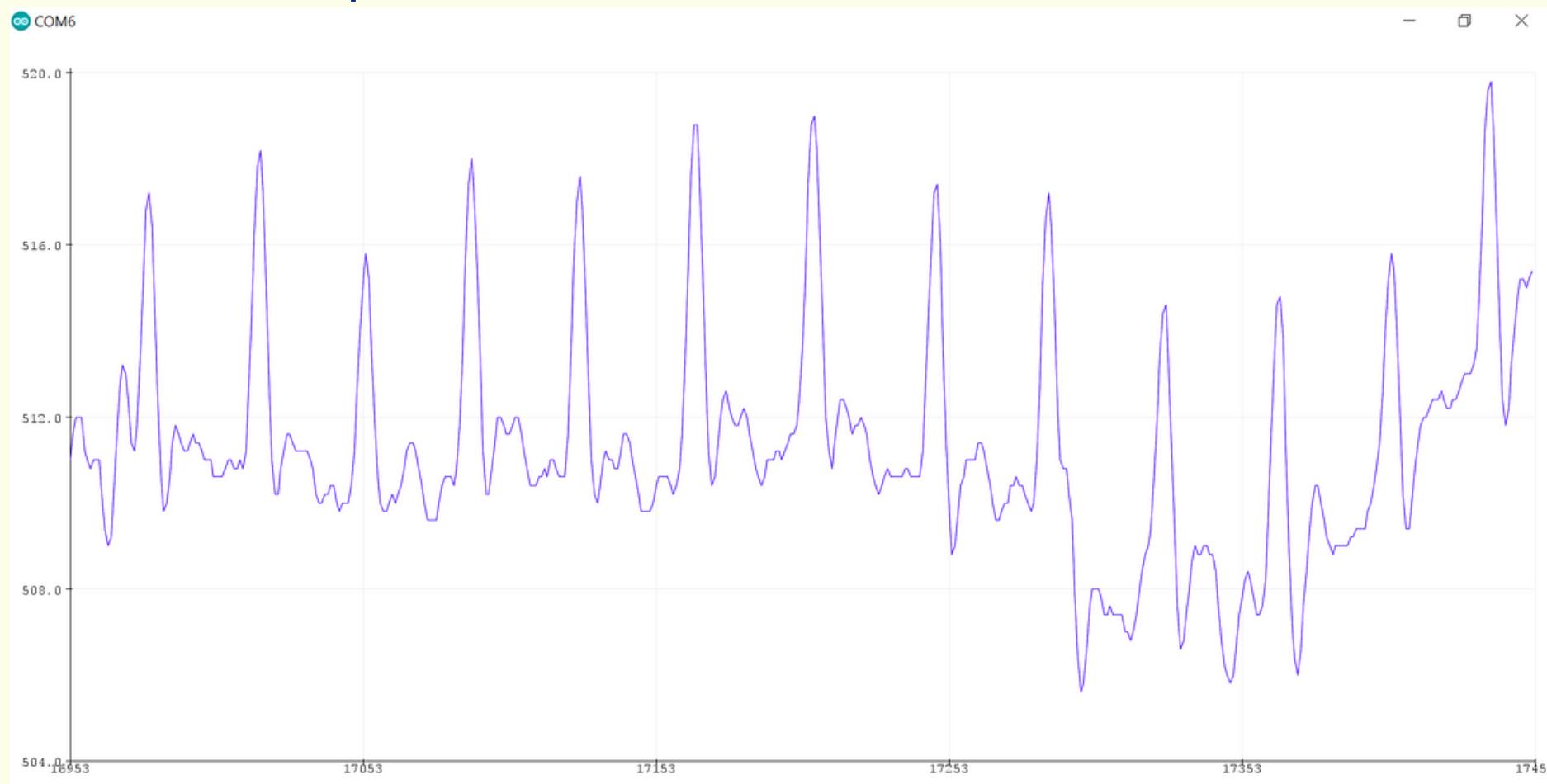
GETTING A BETTER OUTPUT

SMOOTHING OUT THE SIGNAL

- We can make the signal look more smooth by taking an average of the last few readings
- A larger sample size will make the signal look very smooth (potentially missing actual heartbeats)
- A smaller sample size will not make the signal look smooth at all

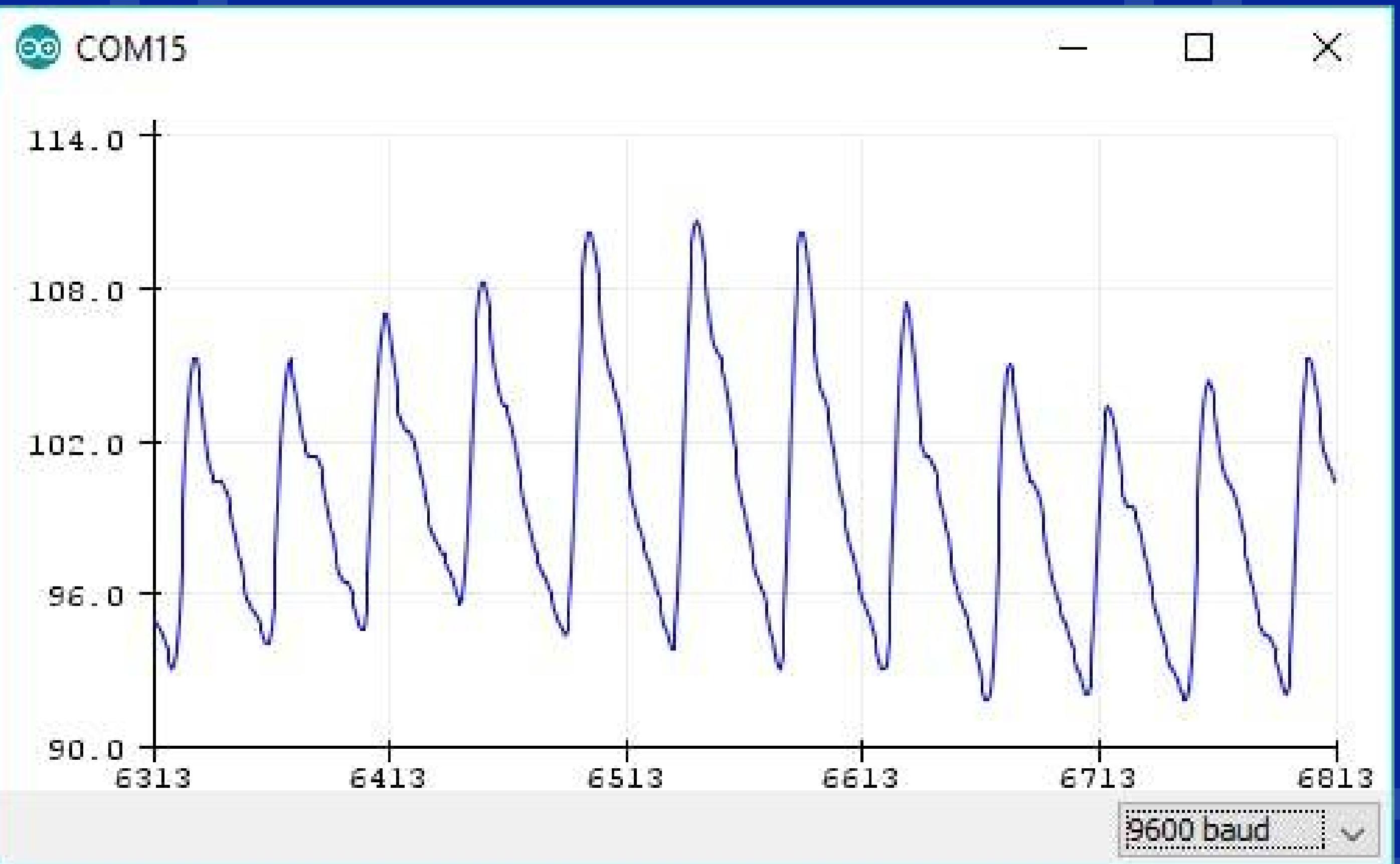
WHAT'S THE DIFFERENCE?

- Removing noise – turning a value/bunch of values into a measurement you can trust
- Smoothing – taking the measurements you trust and making the overall plot more smooth



FINDING THE HEART RATE

- You can find a heartbeat by finding a number of increasing consecutive values (i.e. when the curve rises)
- If you calculate the weighted average of the times between three heartbeats, you get a good approximation of the BPM



WHY IS IT NOT PERFECT?

- There's always a choice between cost and accuracy
- Low cost HR sensors aren't very accurate
- Don't use these as a diagnostics machines!
- Find a balance between them
- Calibration and processing is hard!



imgflip.com

SUMMARY

- Sensors in general are great to use
- With little effort, you can get them working very easily
- Building low cost prototypes using sensors like the HR sensor is great
- Remember, we're prototyping, not making a machine for everyone to use!
- We need to find more accurate sensors if we ever want to use them for actual machines

NEXT STEPS

Learn how to write some programs on your own

- Some good places to start are:
 - The god of all tutorials, YouTube
 - Online Courses (Coursera, etc.)
 - University! (COMP1511, COMP2121)

Learn by example! Create some Arduino projects on your own

- <https://create.arduino.cc> is an amazing place

Check out the EWH Design Team!

They do lots of cool projects involving medical equipment

- Past projects like building a patient monitor

Check out our friends at CREATE!

- They hold heaps of cool projects that anyone can join
 - Droid Racing Competition (DRC)
 - Vivid Sydney



THANK YOU!

bit.ly/2Kv1jWK