Measurements, Sensors and Data Logging Course

Week 3

Upcoming Weeks

- Office Hours
  - Monday Feb 1 @ 7:00 PM
- Weekly Session
  - Thursday Feb 4 @ 7:00 PM

Lesson 6: Temp and Humidity Sensor

Using a Library to read a sensor

Temperature Sensor Introduction

Lesson 6: Temp and Humidity

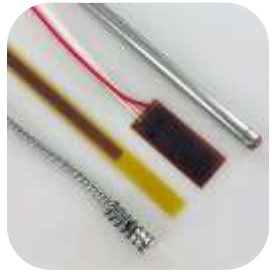
- Temperature Sensor: Measure temperature of object, substance, medium
- Where are they used: Everywhere! fluids, gases, thermostats, electronics, appliances
- How are they used: Depends.
  - For fluid, air and object temp measurement: sensor needs to be in the medium (fluid/air) or against the object being measured
  - For IR sensors: sensor remotely mounted, a signal reflects against surface of the object being measured

Temperature Sensor Introduction

Lesson 6: Temp and Humidity

- Sensor Types: 4 most common
  - Thermocouples: 2 dissimilar metals joined together, produce a voltage (Seebeck effect)
    - Very small voltage produced, need to use an amplifier to measure.
    - Many different “types”, ex: K-Type, J-Type

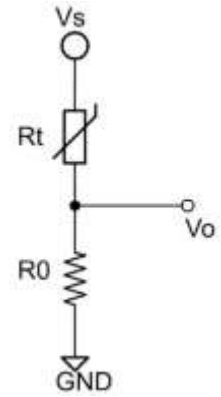
  - RTD (Resistance Temperature Detector): Resistance changes with temperature
    - Current drivers required to measure



- More Information: https://www.digikey.com/en/blog/types-of-temperature-sensors#:~:text=There%20are%20four%20types%20of,based%20integrated%20circuits%20(IC

Temperature Sensor Introduction

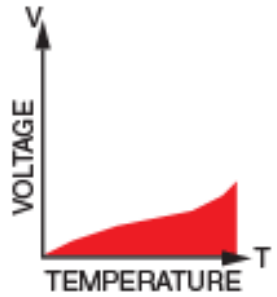
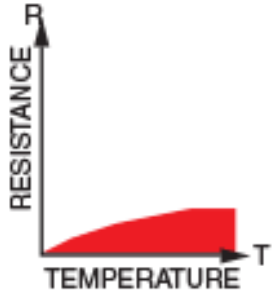
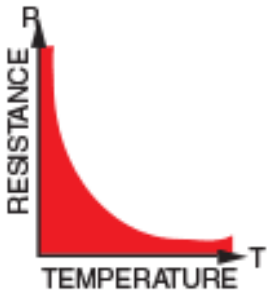
Lesson 6: Temp and Humidity



- Sensor Types: 4 most common
  - Thermistor: Resistance changes with temperature
    - Use voltage divider to measure
  - Semiconductor based ICs: measure the physical properties of a transistor

Temperature Sensor Introduction

Lesson 6: Temp and Humidity

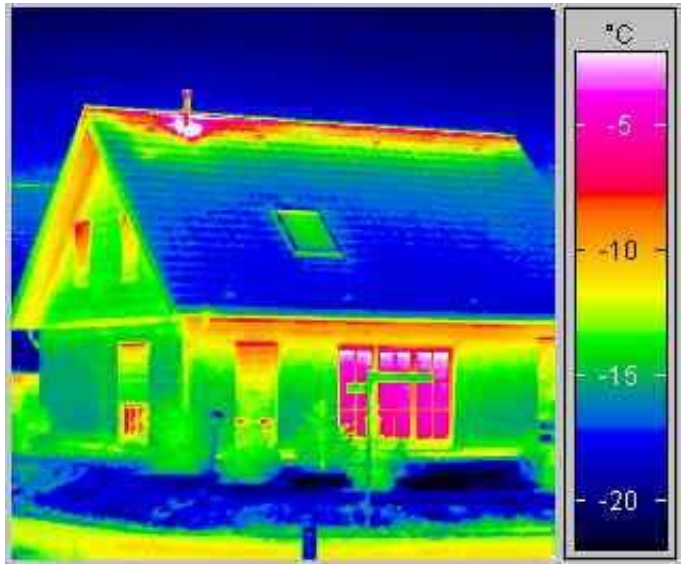
	Thermocouple	RTD	Thermistor	I. C. Sensor
				
Advantages	▪ Self-powered ▪ Simple ▪ Rugged ▪ Inexpensive ▪ Wide variety ▪ Wide temperature range	▪ Most stable ▪ Most accurate ▪ More linear than thermocouple	▪ High output ▪ Fast ▪ Two-wire ohms measurement	▪ Most linear ▪ Highest output ▪ Inexpensive
Disadvantages	▪ Non-linear ▪ Low voltage ▪ Reference required ▪ Least stable ▪ Least sensitive	▪ Expensive ▪ Current source required ▪ Small Δ R ▪ Low absolute resistance ▪ Self-heating	▪ Non-linear ▪ Limited temperature range ▪ Fragile ▪ Current source required ▪ Self-heating	▪ T<200°C ▪ Power supply required ▪ Slow ▪ Self-heating ▪ Limited configurations

Source: https://in.omega.com/prodinfo/integrated-circuit-sensors.html

Temperature Sensor Introduction

Lesson 6: Temp and Humidity

- IR Sensor: IC based temperature sensor
  - How it Works: Detect infrared radiation and convert it to a temperature
  - Thermal cameras use the same principle



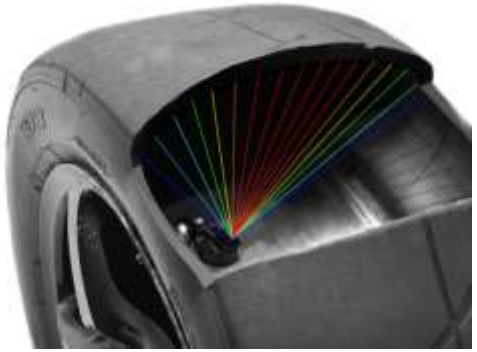
- More information: https://www.flir.com/discover/rd-science/temperature-guns-versus-thermal-imaging-technology/#:~:text=Both%20spot%20pyrometers%20and%20thermal,it%20into%20a%20temperature%20reading.&text=A%20spot%20pyrometer%20reads%20the,of%20the%20entire%20thermal%20image

Temperature Sensor Introduction

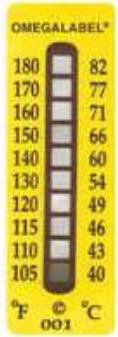
Lesson 6: Temp and Humidity

- Temperature sensors on a race car *(x47 – 131)*
  - **Fluid:** coolant *x2*, engine oil *x2*, gearbox oil *x2*, hydraulic fluid *x0-6*
  - **Gases:** ambient air, intake air *x2-4*, tire air *x4*, exhaust *x2-10*, cockpit + HVAC *(x1-5)*
  - **Object:** tire surface *x12-64*, brake rotor *x4*, brake caliper *x4*, track surface, clutch, force sensor correction *x0-12*
  - **Internal Electronics:** Device, Microcontroller and PCB *x10-12*
  - **Surface Temp Stickers + Paint:** Brakes, Electronics + More *x8-20*

Thermocouple Amplifier



Internal Tire Temp (IR)



Caliper Temp (sticker)

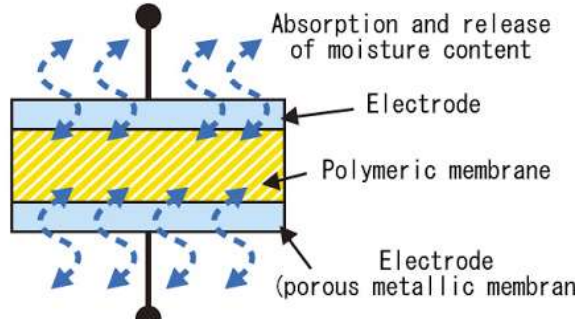
Rotor Temp (paint)

Track Surface Temp (IR)

Humidity Sensor Introduction

Lesson 6: Temp and Humidity

- Humidity Sensor: measures relative humidity (water vapor in the air)
  - To determine dew point and absolute humidity: combine relative humidity and temperature measurements
- Types
  - **Capacitive** (most common): A thin strip of non-conductive polymer film between two electrodes. The electrical capacity of the film changes with the atmosphere's relative humidity. Measure a change in the capacity of the film.
  - **Resistive:** Similar construction to the capacitive sensor. The resistance of the material between the electrodes changes with humidity.
  - **Thermal:** Two thermal sensors conduct electricity based upon the humidity of the surrounding air. One sensor is encased in dry nitrogen while the other measures ambient air. The difference between the two measures the humidity.



- Applications: HVAC, weather, environment

- http://blog.servoflo.com/humidity-sensors-capacitive-vs-resistive
- https://www.electronicsforu.com/tech-zone/electronics-components/humidity-sensor-basic-usage-parameter#:~:text=A%20humidity%20sensor%20(or%20hygrometer,both%20moisture%20and%20air%20temperature.&text=Humidity%20sensors%20work%20by%20detecting,Capacitive

Lesson 6 Hardware

Lesson 6: Temp and Humidity

- What hardware will we need for this Lesson?
  - Grove Temperature and Humidity Module on pin D3
  - Seeeduino Lotus (Arduino Uno compatible board)
    - The Arduino has the serial port hardware built into the device

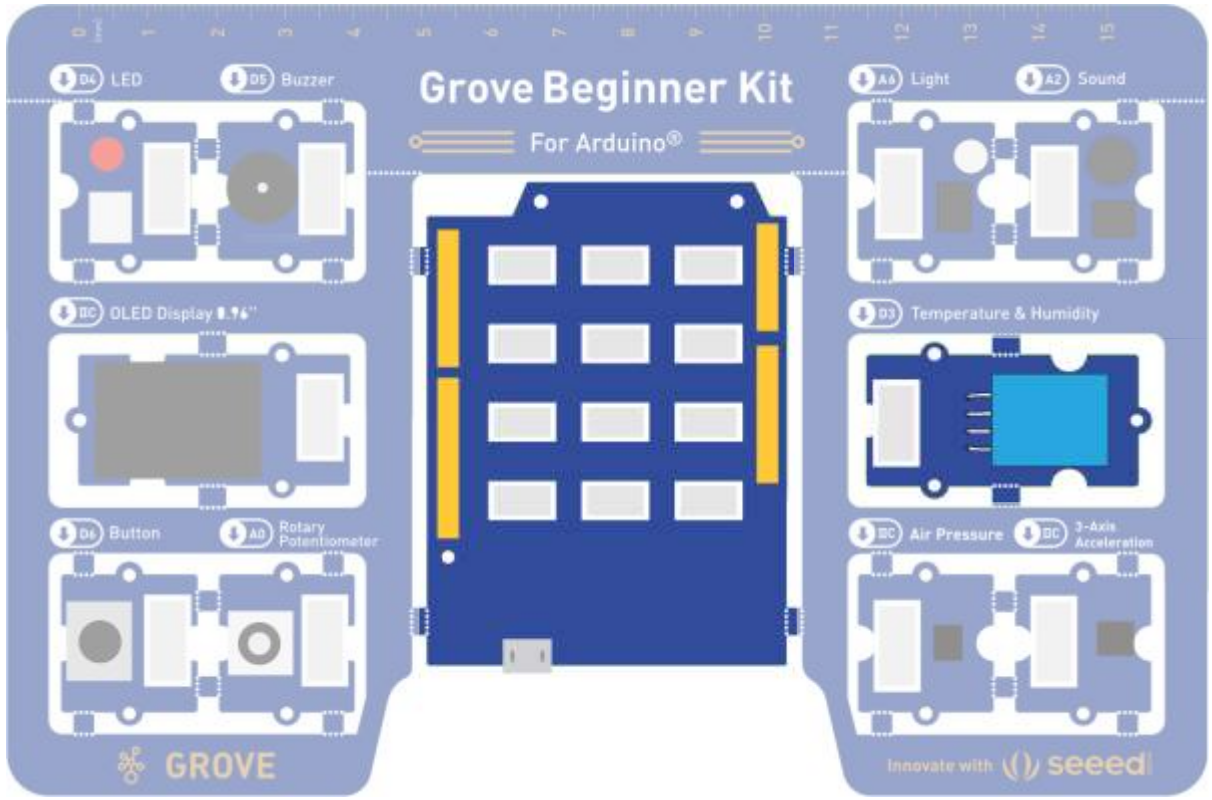
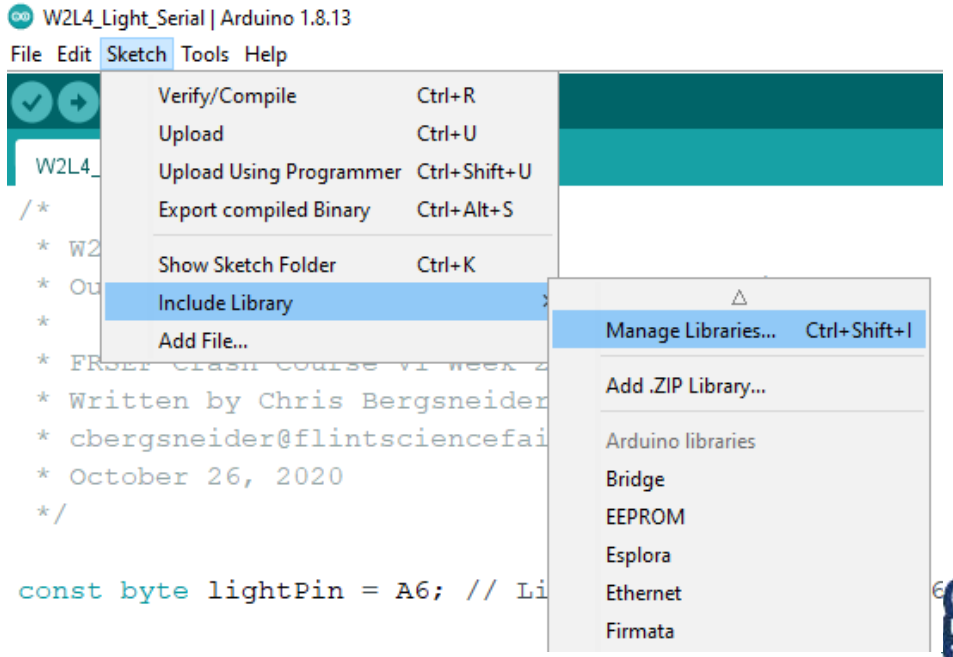


Image modified from https://files.seeedstudio.com/wiki/Grove-Beginner-Kit-For-Arduino/res/Grove-Beginner-Kit-For-ArduinoPDF.pdf

Libraries

Lesson 6: Temperature and Humidity

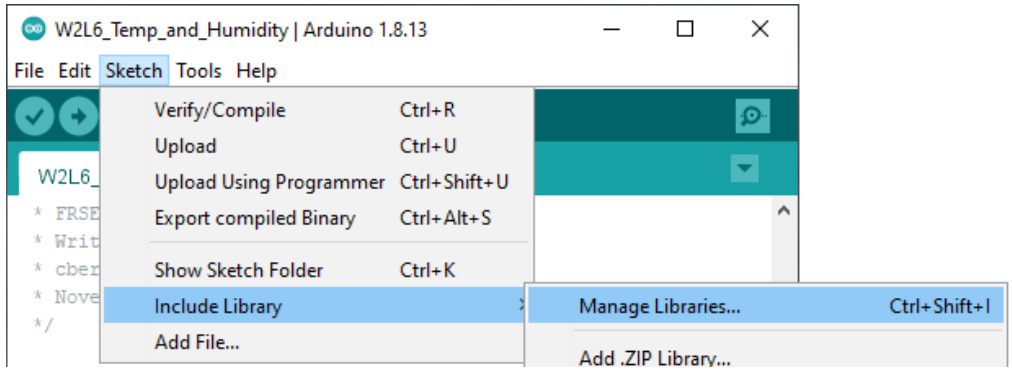
- Library in Arduino
  - What it is: Collecition of pre-defined code.
  - What is does: Provides extra functionalities for sketches, easily.
  - Why: Simplify our sketches, easily integrate components and functions (sensor, display, math, etc)
  - How: Arduino Library Manager
    - *Sketch/Include Library / Manage Libraries*

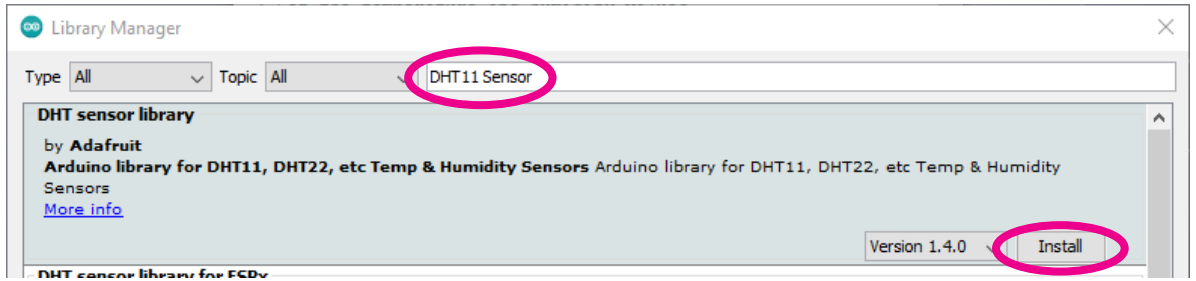


Installing a Library

Lesson 6: Temperature and Humidity

1. Open the Manage Libraries Dialog
   a. **Sketch → Include Library → Manage Libraries…**
2. Install DHT11 Library
   a. Search “**DHT11 Sensor**” in the search Box
   b. Click **Install** on **DHT sensor library** by Adafruit
   c. Click **Install all** to also install the Adafruit Unified Sensor library dependency
   d. Close out of the Library Manager





Code Analysis: Including a Library

Lesson 6: Temp and Humidity

```
#include <DHT.h>
```

  - Make the classes, methods and functions in the installed DTH library available to our sketch
- The `#include` preprocessor directive is used to include a library into your sketch.
- Libraries offer access to a large pool of functions and capabilities written by others and offered to you through the open source community
- Syntax:

```
#include <lib.h>
```

  - Installed library include. These libraries are located in the installed library folder, but the must be installed first

```
#include "lib.h"
```

  - Local library include. These libraries are searched for in the same folder as your sketch
- More Infromation:
  - https://www.arduino.cc/reference/en/language/structure/further-syntax/include/

Open and Upload Sketch

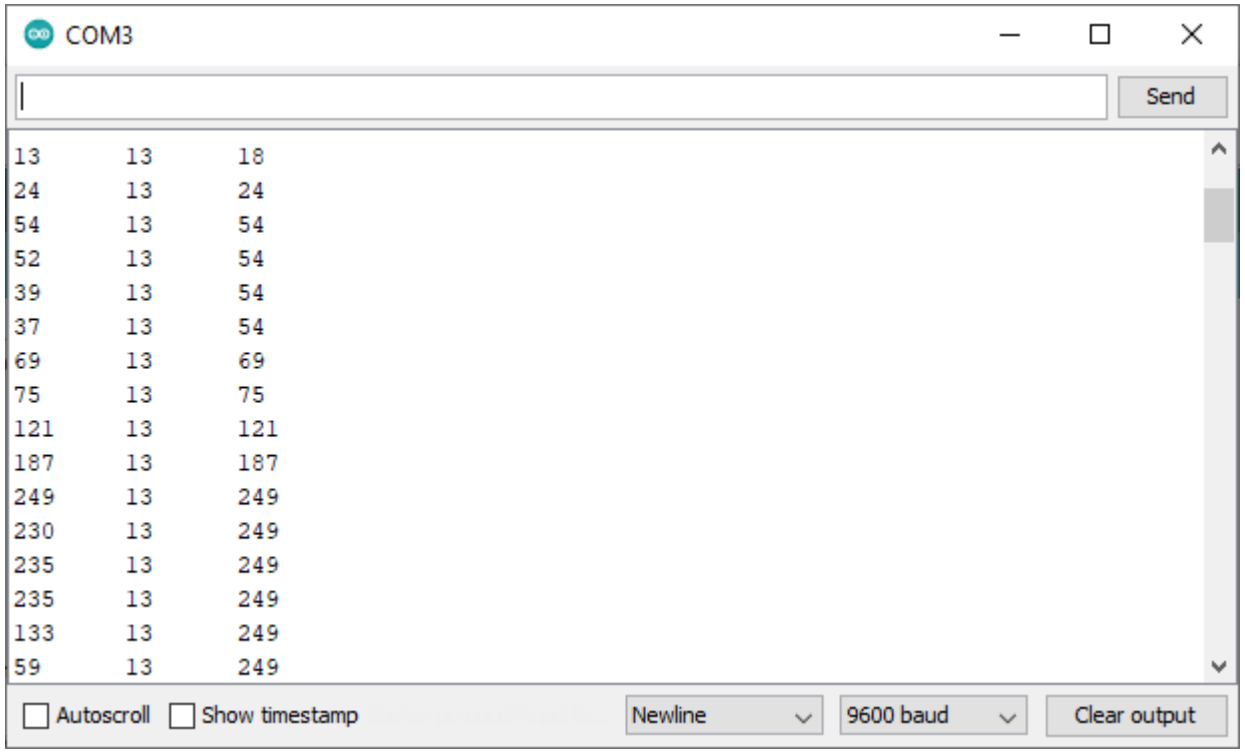
Lesson 6: Temp and Humidity

1. Open Temp and Humidity Sketch
   a. **File → Sketchbook → FRSEF_Crash_Course → Week_3 → W3L6_Temp_and_Humidity.ino**
2. Upload the sketch to your Arduino by clicking the Upload Button.
   a. The sketch should compile, and then upload to your Arduino.
3. Open the serial monitor.
   a. **Tools → Serial Plotter** (Ctrl+Shift+L)
4. Observe the output in the Serial Plotter

Serial Monitor

Lesson 6: Temp and Humidity Sensor

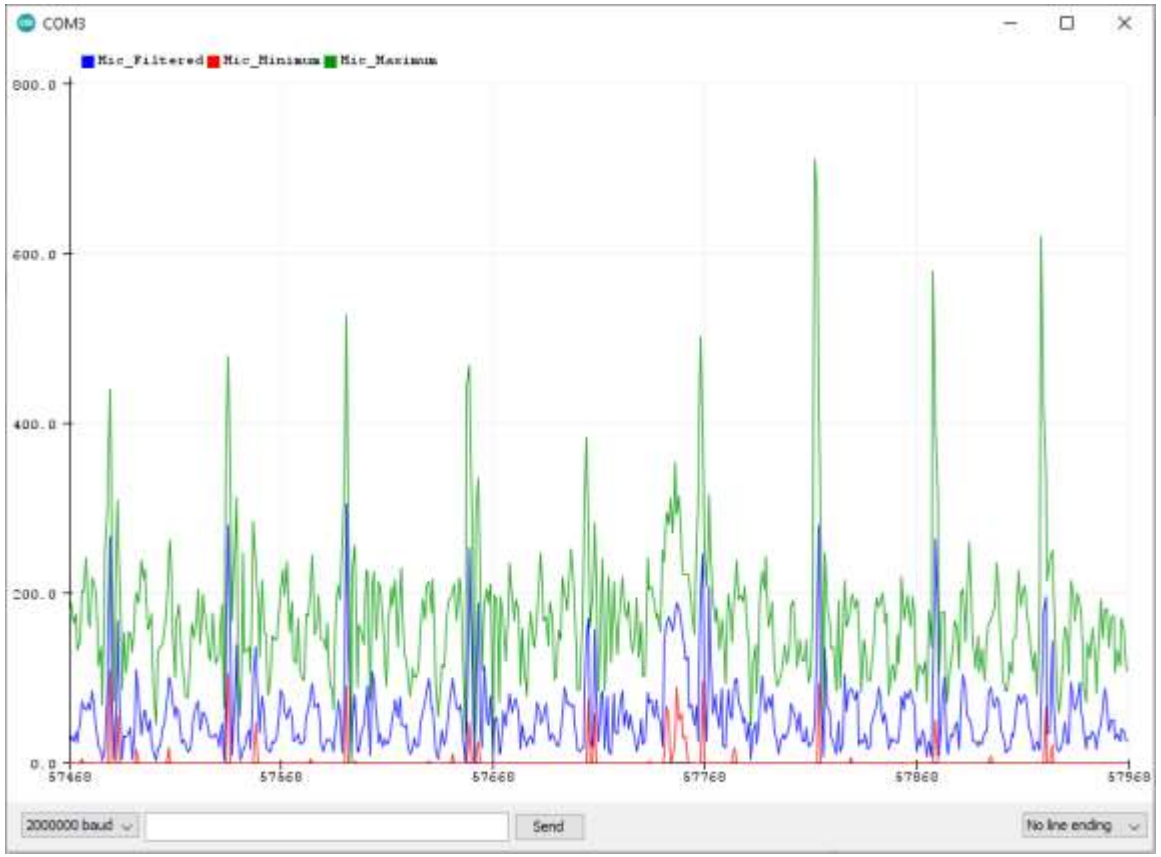
- What is the Serial Monitor?
  - The Serial Monitor is a feature of the Arduino IDE that gives you a serial terminal to see what is being sent to the COM port and allows you to send stuff out of the COM port.
  - We use this for receiving data from the Arduino.
  - We can also use this to help us debug our sketches.



Serial Plotter

Lesson 6: Temp and Humidity Sensor

- What is the Serial Plotter?
  - The Serial Plotter is a feature of the Arduino IDE that gives you a graphical representation of what is being sent to the COM port.
  - We use this for receiving data from the Arduino.
  - The serial plotter will display up to 500 consecutive sample periods.



- More Information:
  - https://arduinogetstarted.com/tutorials/arduino-serial-plotter

Serial Plotter

Lesson 6: Temp and Humdity Sensor

- How do we use the serial plotter?
  - Optionally we start off with a header using the syntax:

```
Serial.println("header_1 header_2");
```

    - We can add more headers by separating them with a space
  - To display the values, we use the **`Serial.print()`** and **`Serial.println()`** functions to send values to the Serial Plotter similar to how we sent values to the Serial Monitor.
  - Each value in a sample period should be separated by tab **`'/t'`** character. Each new sample period should be separated by a newline character or using the **`Serial.println()`** function.

Example Serial Plotter Code

```
void setup()
{
  Serial.begin(9600);
  Serial.println("header1 header2");
}

void loop()
{
  // get values to display
  int val1 = analogRead(A0);
  int val2 = analogRead(A2);
  Serial.print(val1);
  Serial.print('/t');
  Serial.println(val2);
}
```

Code Analysis: Creating a Function

Lesson 6: Temp and Humidity

```
float CtoF(float tempC)
{
  return tempC * 1.8 + 32;
}
```

- Function that takes a float representing a Celsius temperature and returns a float representing the Fahrenheit conversion
- What is a function?
  - A function is a block of code that
- Syntax:

```
return_type function_name(parameter_type parameter_name, …)
{
  // your code here
  return return_value
}
```

- More information:
  - https://www.arduino.cc/en/Reference/FunctionDeclaration

Code Analysis: `#define` macro

Lesson 6: Temp and Humidity

```
#define DHTTYPE DHT11
```

  - before compile, find all instances of “DHTTYPE” and replace with “DHT11”
- The `#define` preprocessor directive is used to name constants at the beginning of the file before compiling
- Caution: this preprocessor acts like a “find and replace all” command without regard to context. It is recommended to use all caps and be verbose when defining your constant names to minimize the possibility that the constant name was used elsewhere when using this method.
- Syntax:

```
#define CONSTANTNAME value
```

  - `CONSTANTNAME` – name of the constant or macro to define
  - `value` – value to assign to the constant or macro
- More Information:
  - https://www.arduino.cc/reference/en/language/structure/further-syntax/define/

Code Analysis: Using the DHT library, DHT Class

Lesson 6: Temp and Humidity

```
DHT dht11(dht11Pin, DHTTYPE);
```

- Create a DHT class instance called dht11 using pin dht11Pin and sensor type DHTTYPE
- What is a class?
  - A class is a collection of related variables and functions.
  - Each instance of a class is called an object
  - I like to think of classes as a super variable that has its own functions
- Syntax:

```
class_name object_name(initializing_value(s), …)
```

- More information:
  - https://www.w3schools.com/cpp/cpp_classes.asp

Activities

Lesson 6: Temp and Humidity

- Output the temperature in F and K
- Change the interval we read the sensor from 1000 msec to 100 msec

Lesson 7: Timing

Code Analysis: A new timing method

Lesson 7: Timing

```
Const unsigned int interval = 1000;
static unsigned long nextTime = 0;
unsigned long time = millis();
if (time >= nextTime)
{
  // your code here
  nextTime = time + interval;
}
```

- The above code shows a better way to keep more accurate timing of code execution
- It uses the `millis()` function to keep track of the actual elapsed time and calculates when it needs to run the code again.
- It is different from the `delay()` function because `delay()` waits for a period of time whereas using the `millis()` method can compensate for the time it takes to execute code.
- See timing demos for an example of the running differences.

Activities

Lesson 7: Timing

- Upload Sketch 7a and copy the output
- Upload Sketch 7b and copy the output
- Compare the outputs, which would you want to use for an application that requires precise timing?

Application - Agriculture

Agriculture Applications

Sensor Calibration + Water Sensor

- What can we measure for agriculture applications?
  - Small scale - individual plants, gardens
  - Large scale - farms

Agriculture Applications

Sensor Calibration + Water Sensor

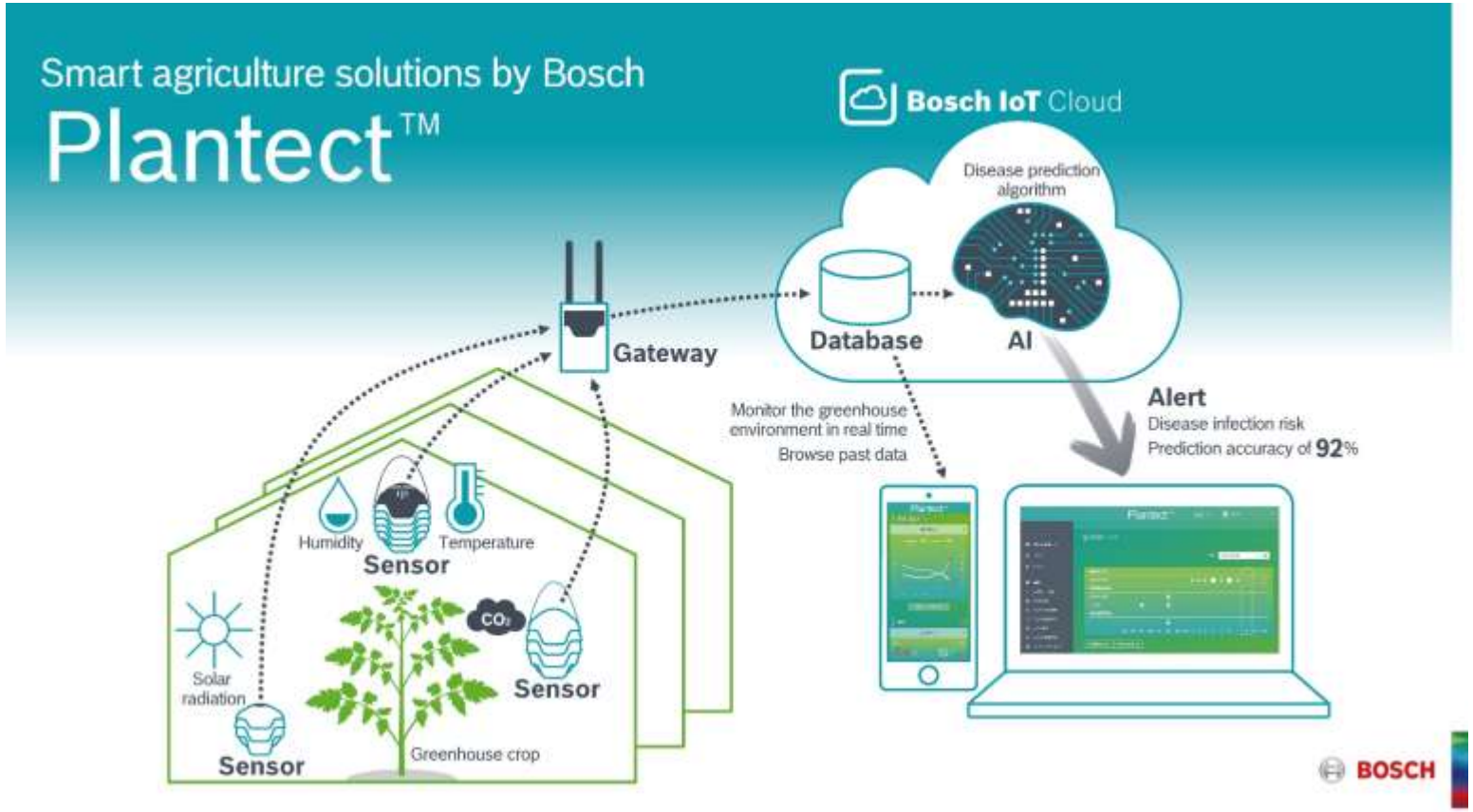
- What can we measure for agriculture applications?
  - Individual plants
    - Environment
      - Temperature
      - Air Quality
      - Air Content: CO, CO2, Oxygen
      - Humidity
    - Soil: moisture, temperature
    - Fertilizer
    - Plant Height
  - Large Scale: entire fields
    - Optical (soil reflectance, color, height)
    - Cameras (identify weeds, where plants are growing)

https://cropwatch.unl.edu/ssm/sensing
https://www.mouser.com/applications/smart-agriculture-sensors/
https://www.wespeakiot.com/robust-sensors-and-the-power-of-the-cloud-the-perfect-recipe-for-smart-farming/

Agriculture Applications + Livestock

Sensor Calibration + Water Sensor

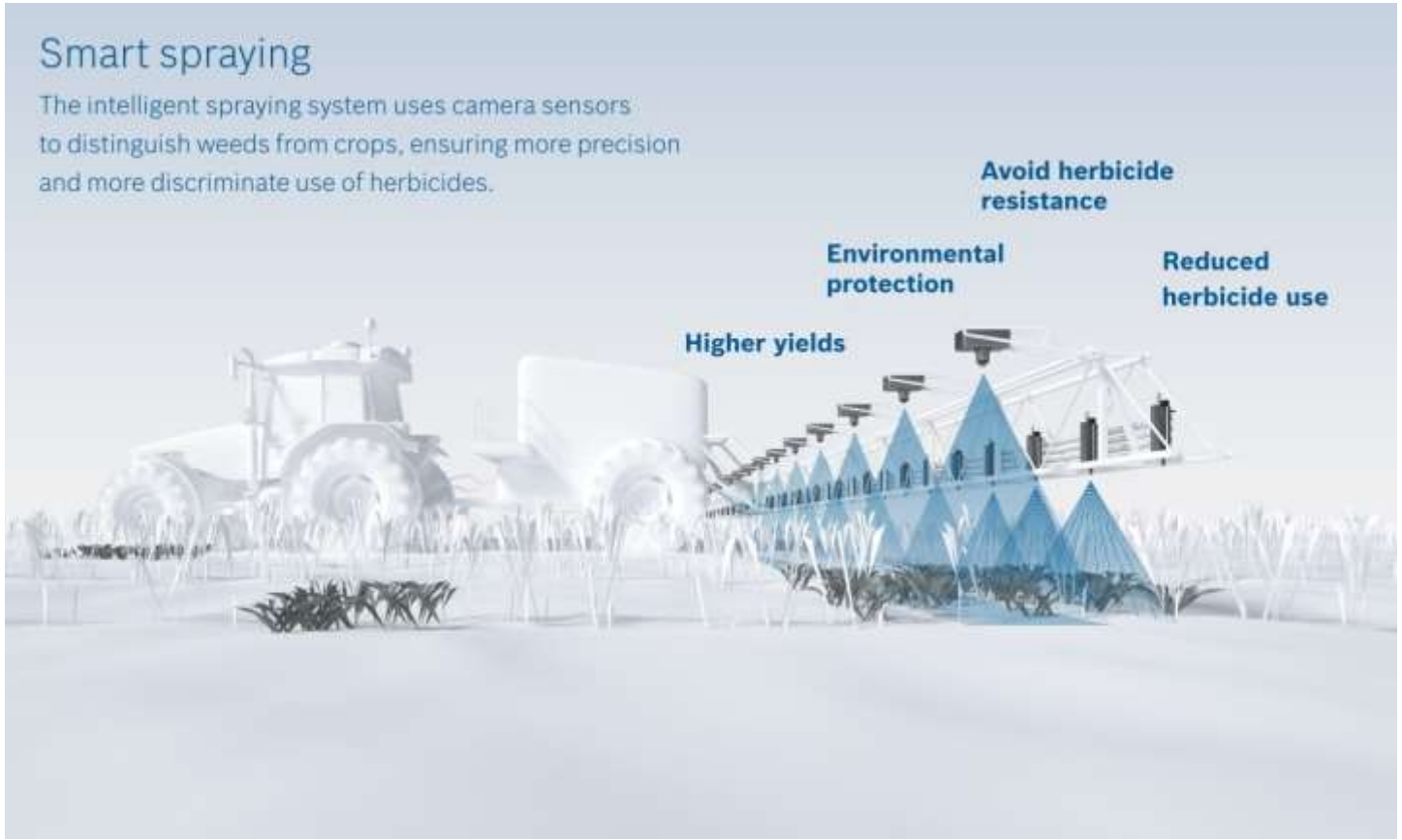
https://www.bosch-presse.de/pressportal/de/en/smart-agriculture-101824.html



Agriculture Applications + Livestock

Sensor Calibration + Water Sensor

https://www.bosch-presse.de/pressportal/de/en/smart-agriculture-101824.html



Livestock Applications

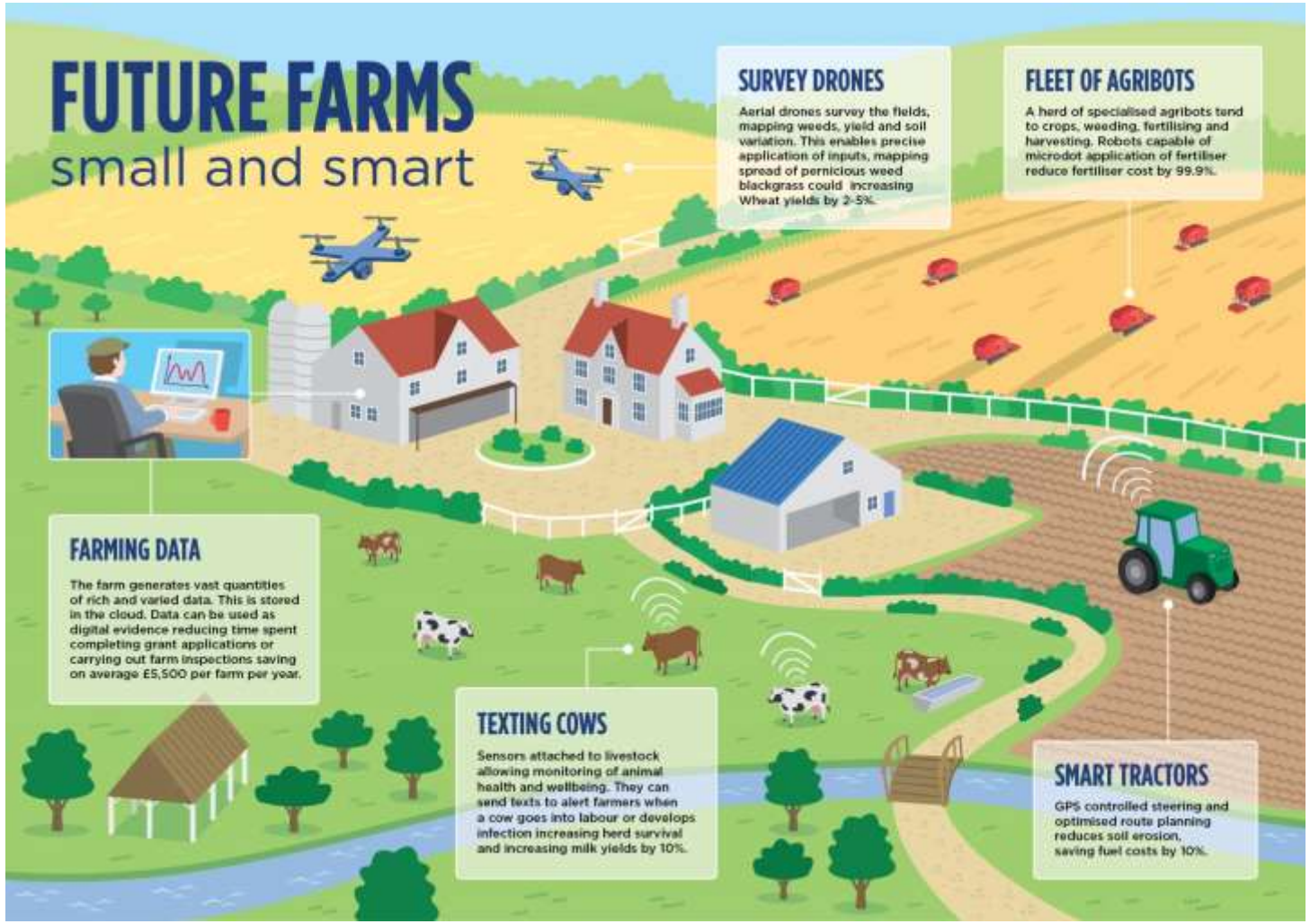
Sensor Calibration + Water Sensor

- Livestock Applications
  - Location
    - GPS
    - Near-Field (near feed bunk, water)
  - Health
    - Temp
    - Pulse-Ox
    - Accel
  - Calving (birth)
  - Weight

https://www.moovement.com.au/farm-management-platform/

Agriculture Applications

Sensor Calibration + Water Sensor



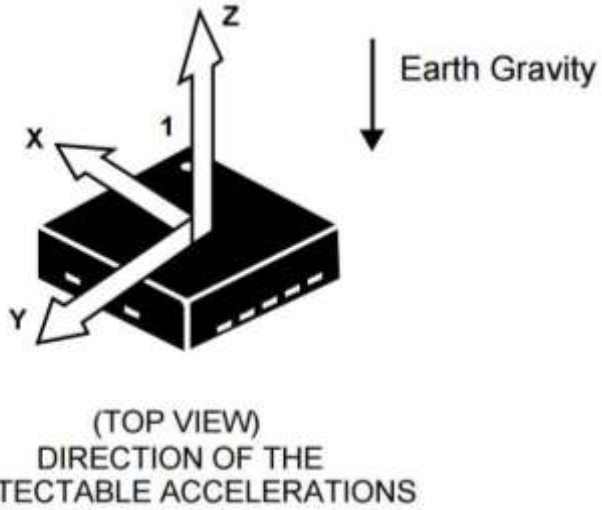
Accelerometer

Lesson 10

Motion Sensors

Accelerometer

- Accelerometers are used to measure acceleration, in linear directions
- Measurements in m/s^2
  - 1g = 9.8 m/s^2
  - 0g = object not moving or in free fall
- Types: analog, digital (IIC, SPI), PWM
- Uses
  - Position tracking
  - Force measurement
  - Vibration measurement

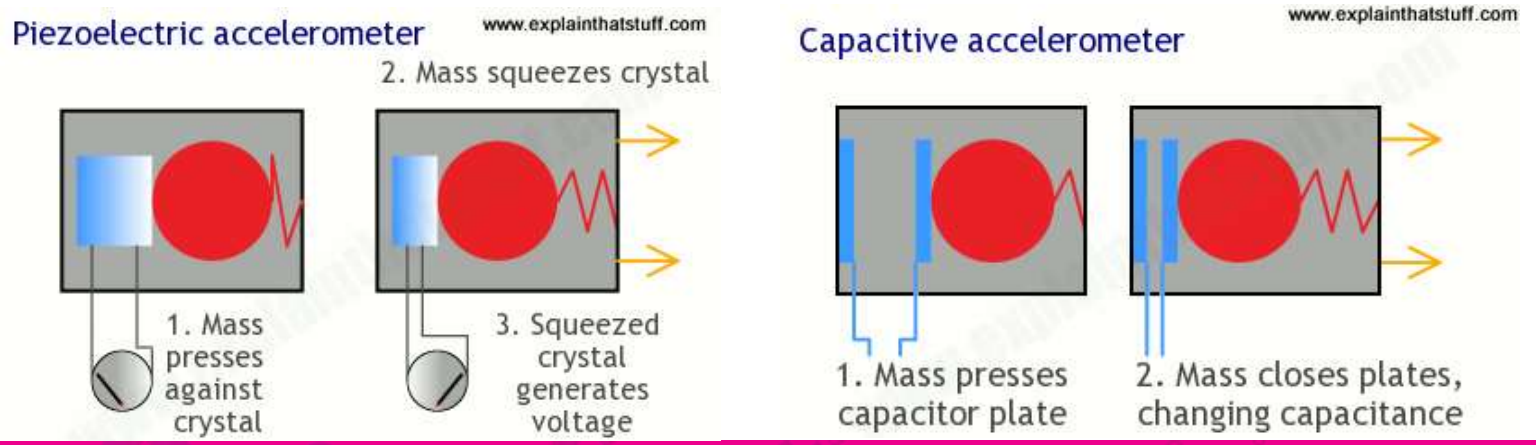


- https://www.adafruit.com/category/521
- https://learn.sparkfun.com/tutorials/accelerometer-basics/all
- https://www.sparkfun.com/pages/accel_gyro_guide?_ga=2.260802829.124947883.1606791953-761679650.1605223049
- https://www.seeedstudio.com/blog/2019/12/24/what-is-accelerometer-gyroscope-and-how-to-pick-one/

Motion Sensors

Accelerometer

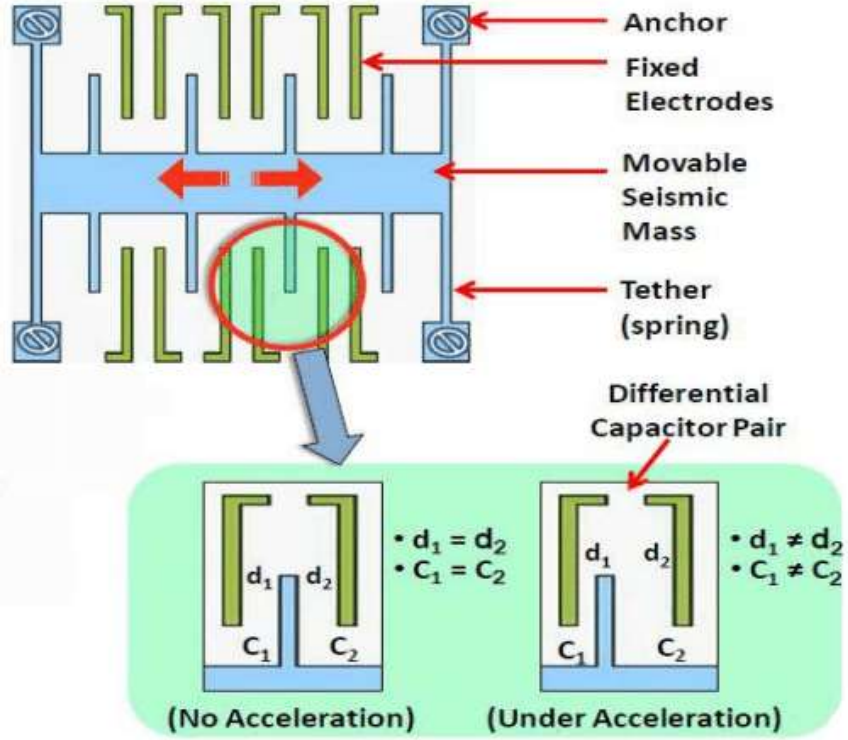
- Accelerometers are used to measure acceleration, in linear directions
- How they work
  - Capacitive: capacitive plates internally, some fixed and others on springs, motion between plates causes change in capacitance
  - Piezo Electric: Small mass on springs around piezo-electric materials. Electrical charges are created.



Motion Sensors

Accelerometer

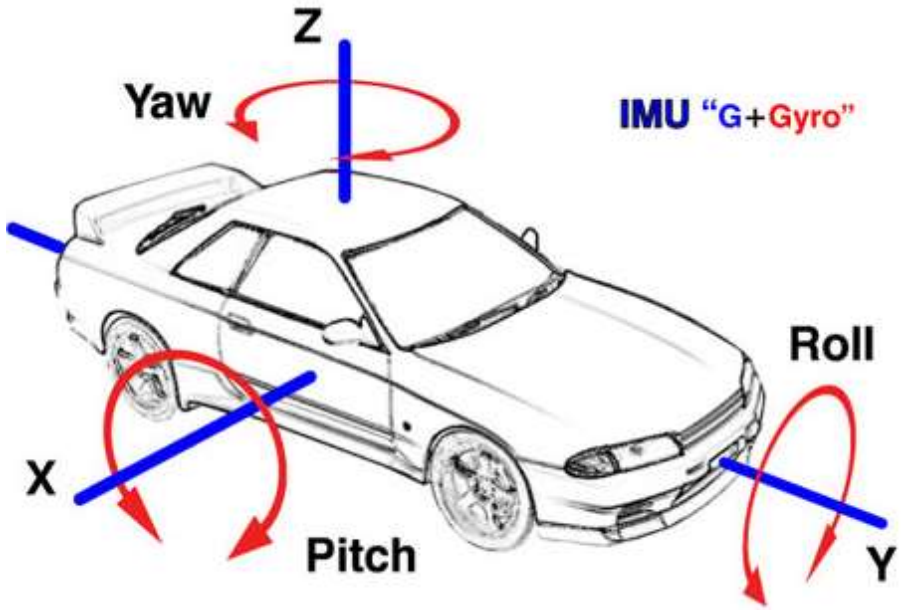
- How they work
  - MEMS: microscopic, silicon based moving mass
    - Uses either piezo or capacitive changes
    - Cool Graphic



Motion Sensors

Accelerometer

- Other commonly used motion sensors:
  - Gyroscope: measure rotational motion
  - Magnetometer: measure magnetic force, typically magnetic north
- IMU: Accelerometer + Gyroscope



Hardware
Accelerometer

- What hardware will we need for this Lesson?
  - Grove 3-axis Accelerometer Module on IIC
  - Seeeduino Lotus (Arduino Uno compatible board)
  - SD Card Shield + SD Card
- Please assemble parts the same way we did last week

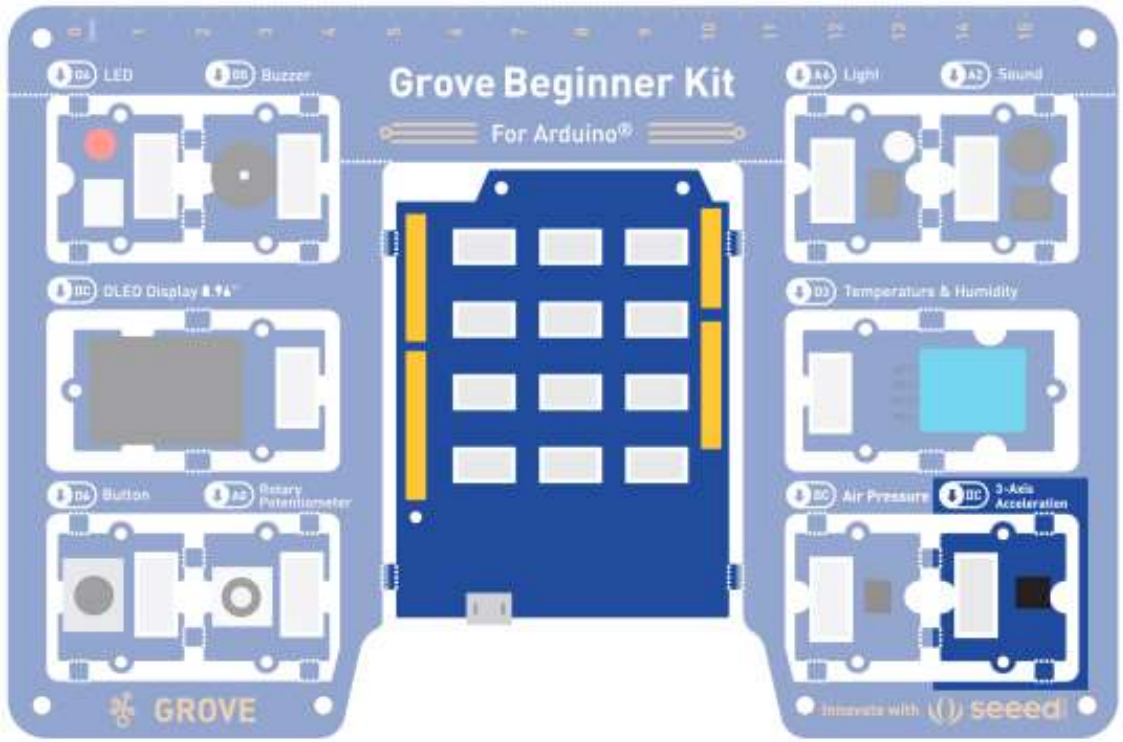


Image copied from https://www.amazon.com/HiLetgo-Logging-Recorder-Logger-Arduino/dp/B00PI6TQWO/

Image copied from https://www.microcenter.com/product/485234/micro-center-64gb-microsdxc-class-10--uhs-1-flash-memory-card

Library
Accelerometer

- Library to use: *sparkfun LIS3DH*
  - Search for ***Sparkfun LIS3DH*** in the library manager and install it.
  - There are multiple variants available for the LIS3DH sensor.
  - `#include <SparkFunLIS3DH.h>`
  - `LIS3DH myIMU(I2C_MODE, 0x19);`
- More Information:
  - https://github.com/sparkfun/SparkFun_LIS3DH_Arduino_Library

Open and Upload Sketch

Lesson 10: Accelerometer

1. Open Simple Datalogger Sketch
   - **File → Sketchbook → FRSEF_Crash_Course → Week_5 → L10_Accelerometer.ino**
2. Upload the sketch to your Arduino by clicking the Upload Button.
   - The sketch should compile, and then upload to your Arduino, assuming you have the correct
3. Move the board around so that different faces point down for a couple seconds then gently shake or drop your board (from a couple inches)
4. Unplug the USB connector
5. Remove SD card and plug into your computer
6. Look at the data in your spreadsheet program
   - note the name of the file is now a number repressing the day, hour, minutes and seconds that the log started at. This prevents us from overwriting or appending data to older files!

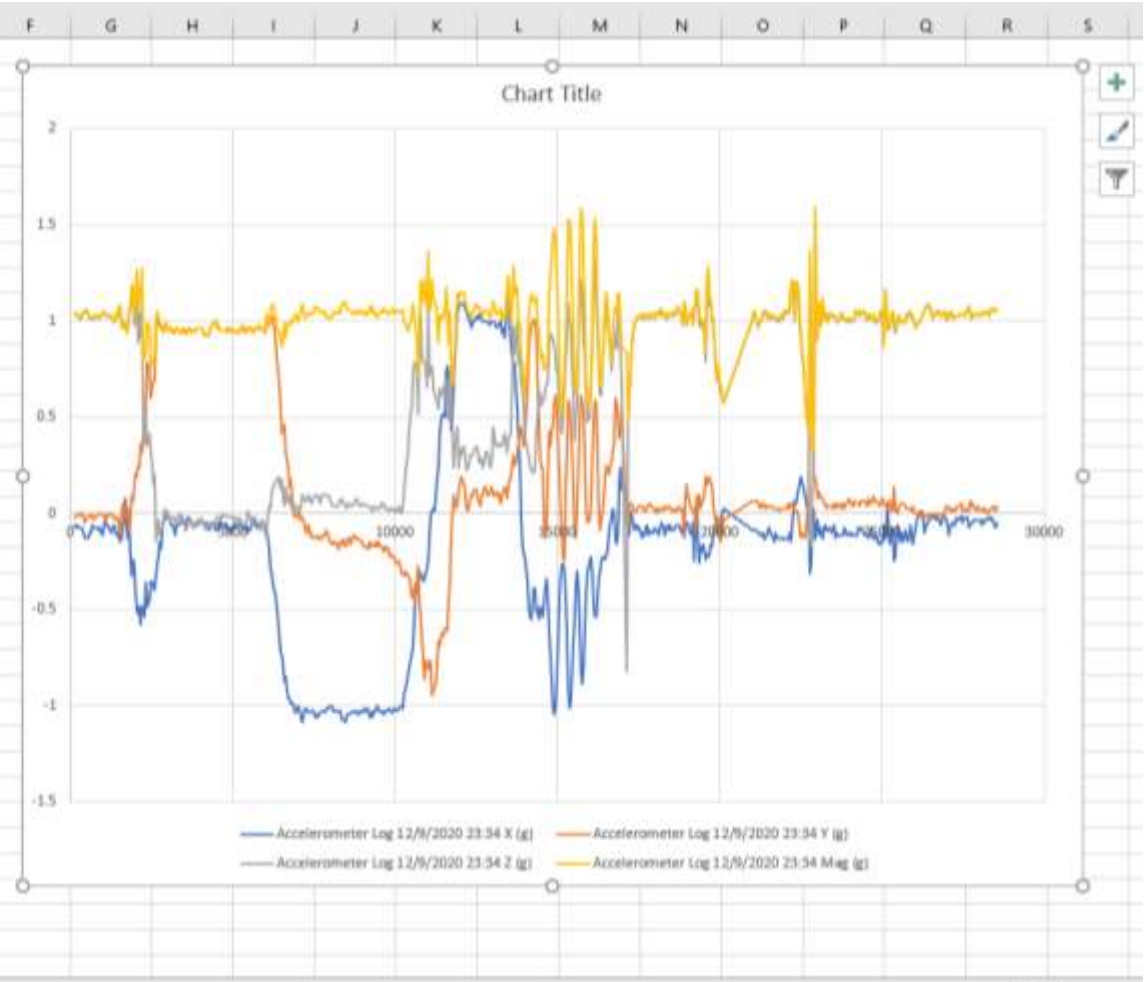
Activity
Accelerometer

- With SD Card shield attached, start recording and conduct multiple drop tests.
  - Conduct test at different logging rates.
  - Be careful as the SD card may lose connection
- When complete, unplug the USB cable and remove the SD card
- Insert the SD card into your PC and plot the data

Example Datalog

Lesson 10: Accelerometer

	A	B	C	D	E
1	Accelerometer Log				
2	########				
3	Time (ms)	X (g)	Y (g)	Z (g)	Mag (g)
4	144	-0.0625	-0.025	1.0375	1.039681
5	184	-0.075	-0.0125	1.0375	1.040282
6	224	-0.0625	-0.0125	1.025	1.02698
7	264	-0.075	0	1.025	1.02774
8	304	-0.075	0	1.0125	1.015274
9	344	-0.075	-0.0125	1.0125	1.015351
10	384	-0.075	0	1.0375	1.040207
11	424	-0.1	0	1.0375	1.042308
12	464	-0.125	-0.05	1.05	1.058596
13	504	-0.125	-0.025	1.05	1.05771
14	544	-0.125	-0.0125	1.025	1.032669
15	584	-0.1	-0.0125	1.0125	1.017503
16	624	-0.1	0	1.0125	1.017426
17	664	-0.0625	-0.025	1.0125	1.014735
18	704	-0.075	-0.0375	1.025	1.028424
19	744	-0.075	-0.0125	1.0125	1.015351
20	784	-0.075	-0.0125	1.025	1.027816
21	824	-0.075	-0.0125	1.025	1.027816
22	864	-0.0875	0	1.025	1.028728
23	904	-0.075	0	1.0125	1.015274
24	944	-0.1	0	1.025	1.029866
25	984	-0.1125	0	1.0375	1.043582
26	1024	-0.075	-0.0125	1.05	1.052749
27	1064	-0.0375	-0.0375	1.0375	1.038854
28	1104	-0.0625	-0.025	1.025	1.027208
29	1144	-0.0625	0	1.025	1.026904
30	1184	-0.075	-0.0125	1.025	1.027816
31	1224	-0.0875	0	1.025	1.028728
32	1264	-0.1	0	1.025	1.029866
33	1304	-0.075	-0.0125	1	1.002887
34	1344	-0.0625	-0.0125	1	1.002029
35	1384	-0.0875	-0.0375	1	1.004521
36	1424	-0.1125	-0.025	1.0375	1.043881
37	1464	-0.1	-0.025	1.025	1.03017



09233437

Software Settings

Accelerometer

```
myIMU.settings.adcEnabled = 0;
myIMU.settings.tempEnabled = 0; // set to 1 to enable temp sensor readings
```

- Turn the output of the accelerometer on or off (1 = on)

```
myIMU.settings.accelSampleRate = sampleRate; // Hz.Can be:0,1,10,25,50,100,200,400,1600, 5000 Hz
```

- Sensor sample rates: how fast will the sensor update its output (0,1,10,25,50,100,200,400,1600,5000 Hz)

```
myIMU.settings.accelRange = 16;   // 16 to read the sudden stop at the end of a drop, max G readable.
```

- Range of the accelerometer (options: 2, 4, 8, 16)
  - 2: -2g to +2g (-19.6 m/s^2 to +19.6 m/s^2)
  - 16: -16g to +16g (-156.8 m/s^2 to +156.8 m/s^2)

```
myIMU.settings.xAccelEnabled = 1;
myIMU.settings.yAccelEnabled = 1;
myIMU.settings.zAccelEnabled = 1;
```

- Turn the output of the accelerometer on or off (1 = on)

https://github.com/sparkfun/SparkFun_LIS3DH_Arduino_Library

Reading the Accelerometer

Accelerometer

`myIMU.readFloatAccelX()`

– returns a floating point number of the acceleration in g's for the X axis

`myIMU.readFloatAccelY()`

– returns a floating point number of the acceleration in g's for the Y axis

`myIMU.readFloatAccelZ()`

– returns a floating point number of the acceleration in g's for the Z axis

- More Information:
  - https://github.com/sparkfun/SparkFun_LIS3DH_Arduino_Library

Code Analysis: Dynamically creating a filename

Accelerometer

- `createFilename()`
  - Function that uses the Time from the RTC to create a unique filename each time it is called
  - Stores the filename in a global String variable called `filename`
  - Filename is formatted as DDHHMMSS.csv using the time that the file was created.
    - You can change this behavior if you need a different format say YYYYMMDD if you are creating a file every few days or so.
  - Log the data several times to see this function in action! You should find several files on your SD card.

Application - Bio

Sensors & Applications – EEG, ECG EMG

- Measures of biopotential, the electrical output of human activity
  - Electroencephalogram (EEG)
    - Monitors brain activity
    - Measurements at forehead, top of head (potentially) and ears
  - Electrocardiogram (EKG)
    - Measures heart activity
    - Measurements at torso, arms and legs
  - Electromyography (EMG)
    - Electrical activity of muscles
    - Common test is to measure muscle response relative to stimulation of the muscle, measure a specific muscle

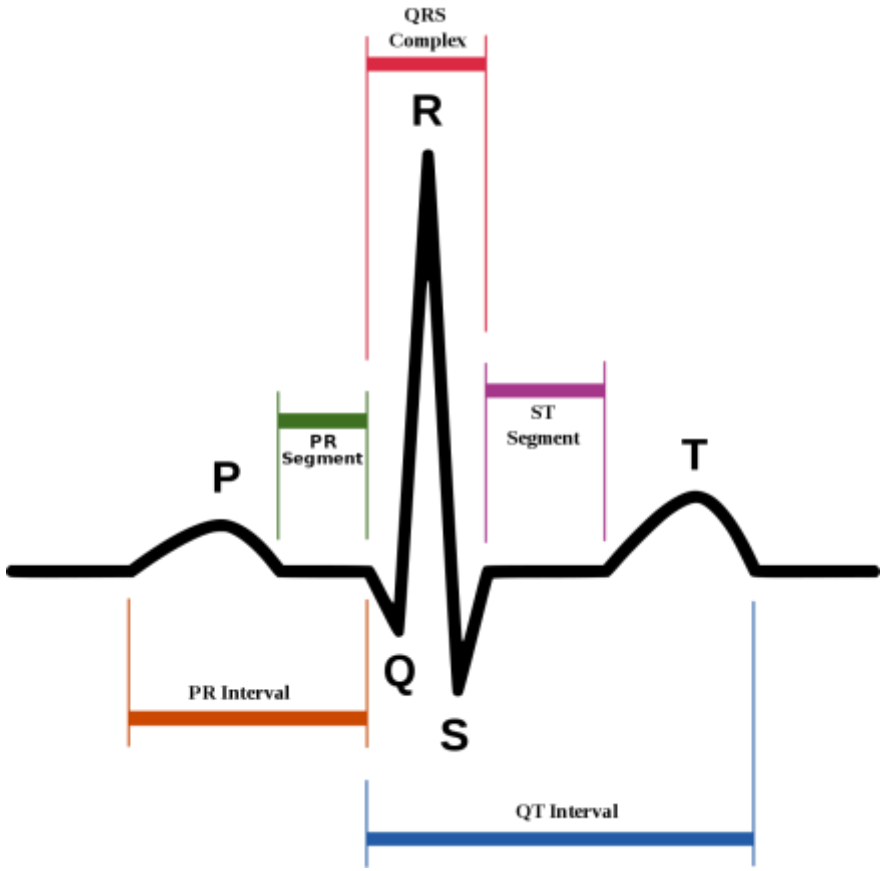
https://www.sensortips.com/featured/what-is-the-difference-between-an-ecg-eeg-emg-and-eog/
https://www.withings.com/de/en/health-insights/about-ecg-ekg-electrocardiogram

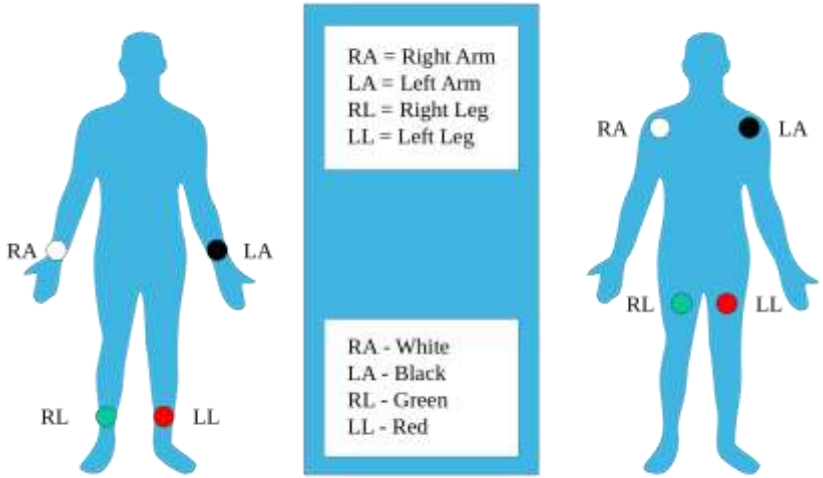
Sensors & Applications – EEG, ECG EMG

- Measuring
  - Amplifier is required (very lower voltages)
  - Electrodes used to “pick up” the voltages

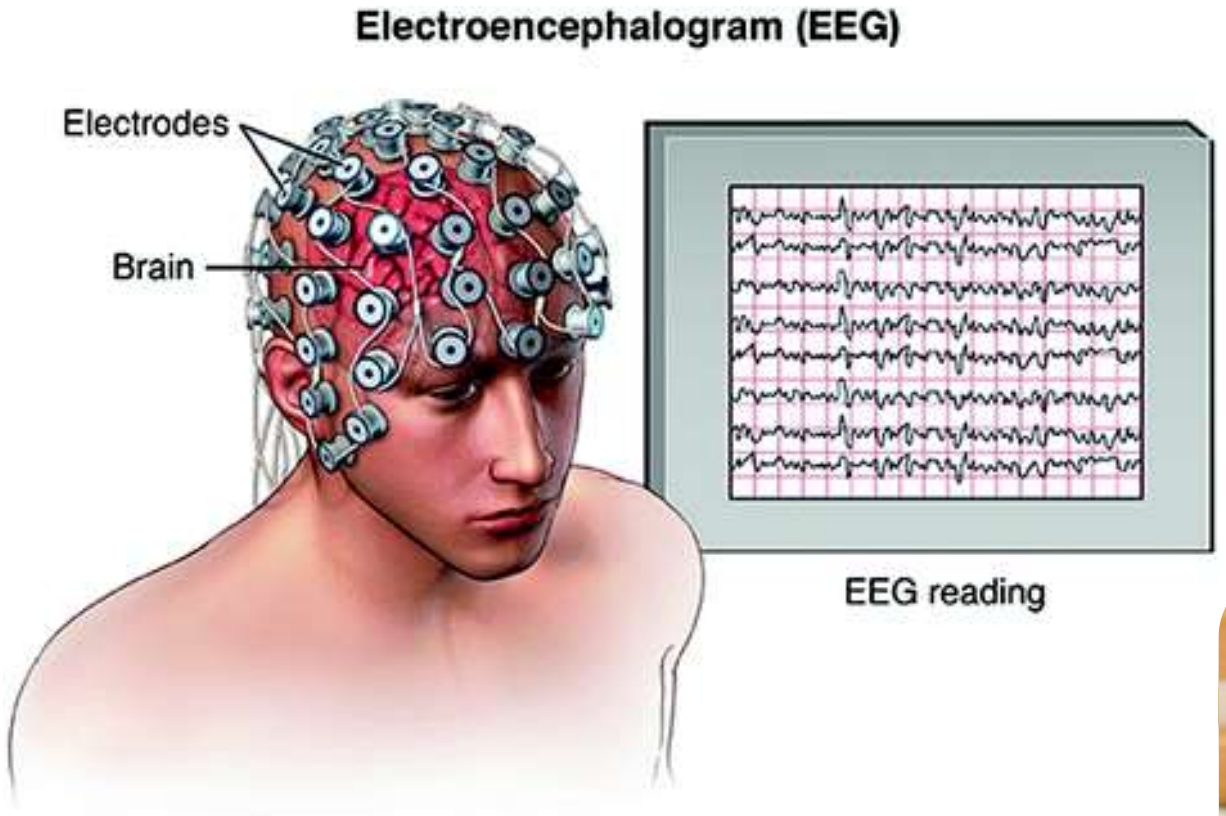
Source	Amplitude (mV)	Bandwidth (Hz)
ECG	1-5	0.05-100
EEG	0.001-0.01	0.5-40
EMG	1-10	20-2000
EOG	0.01-0.1	dc-10

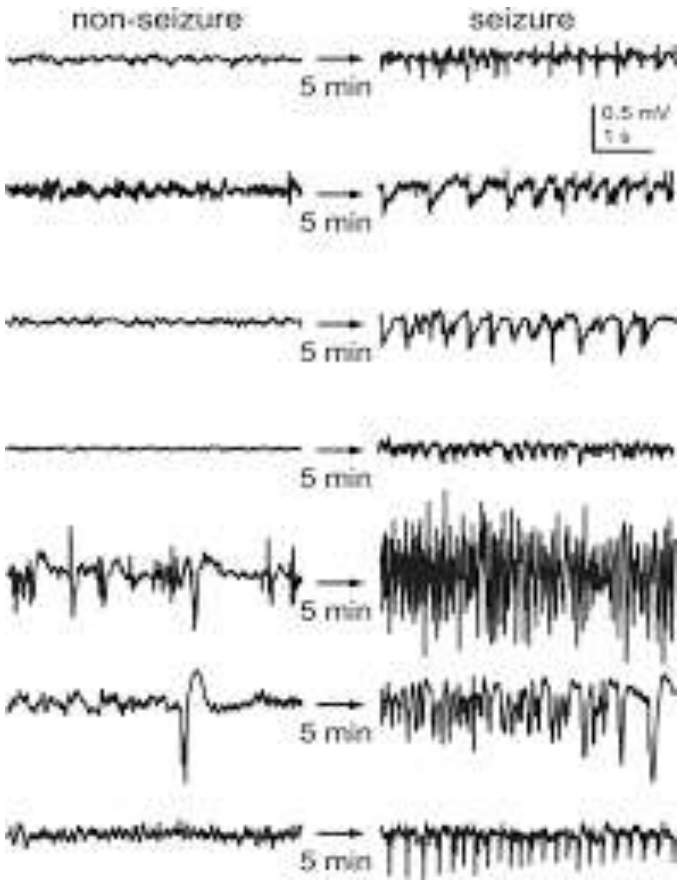
Sensors & Applications – EKG



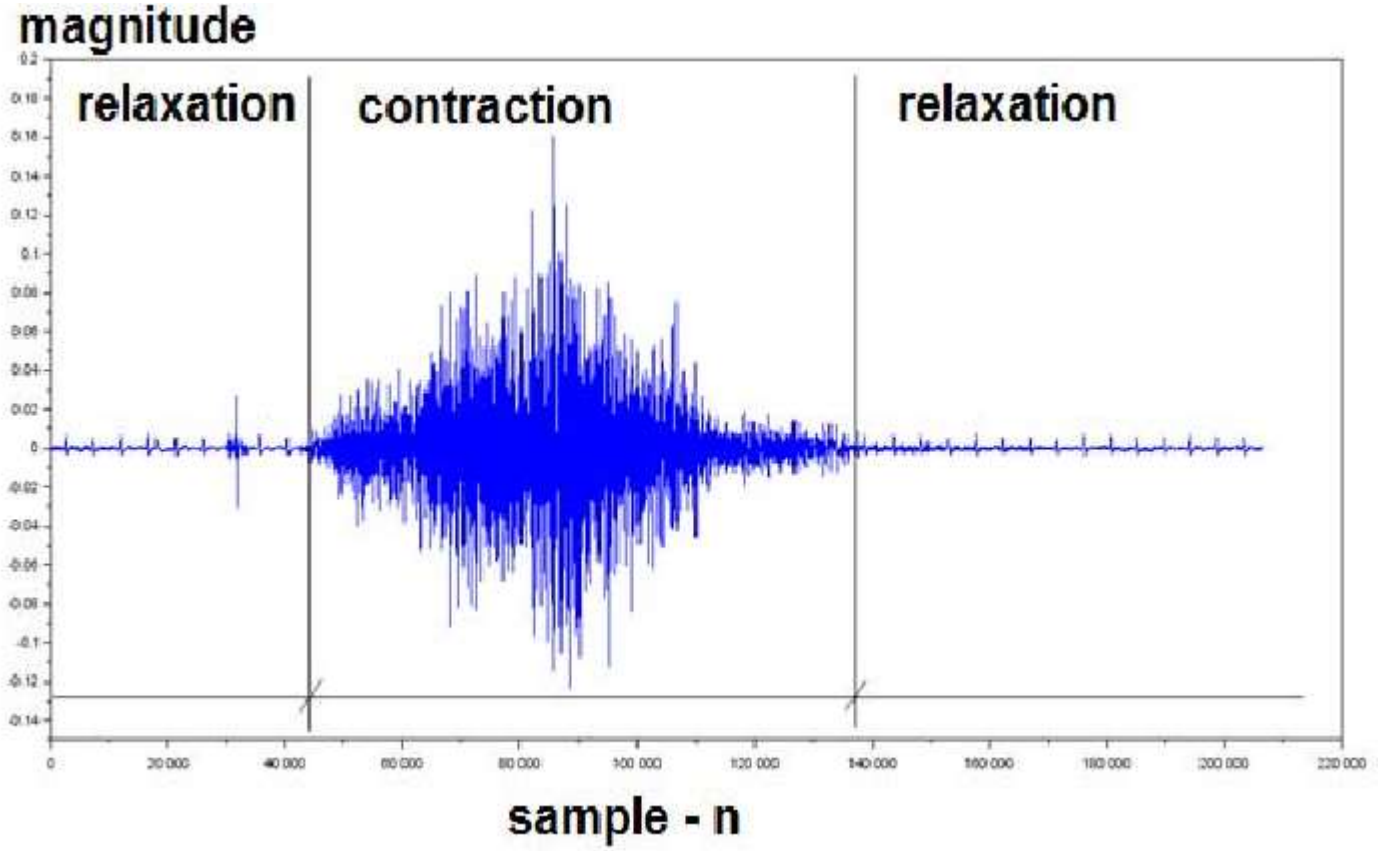


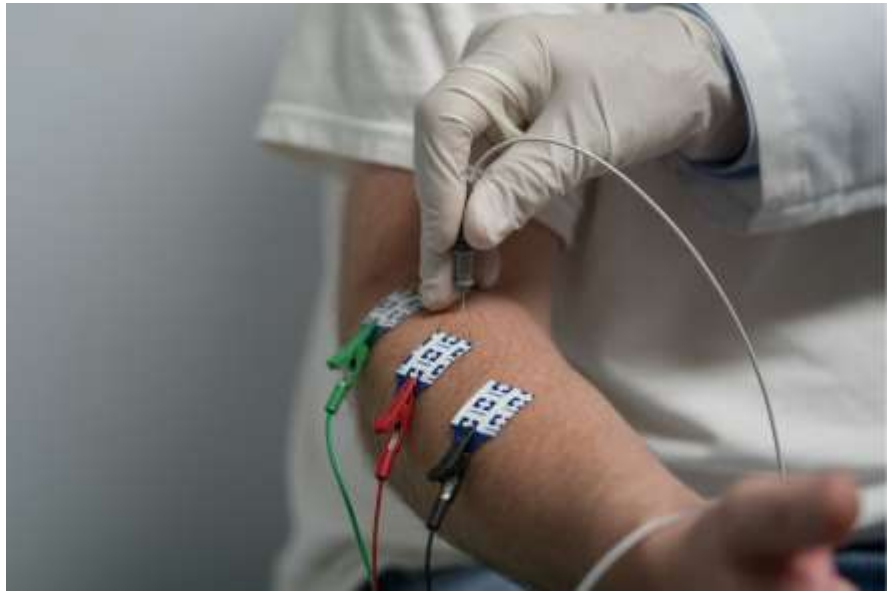
Sensors & Applications – EEG





Sensors & Applications – EMG

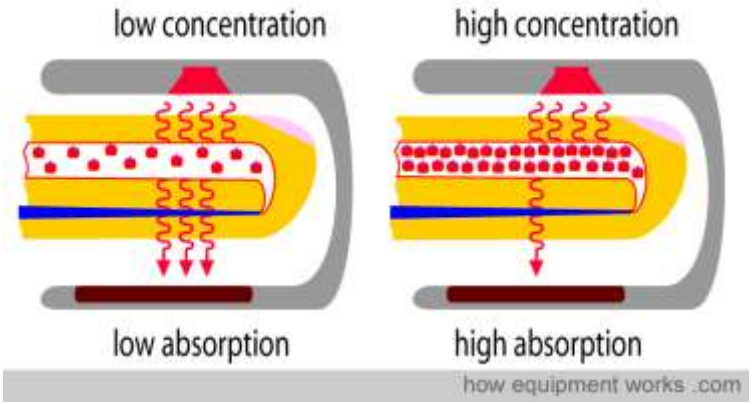


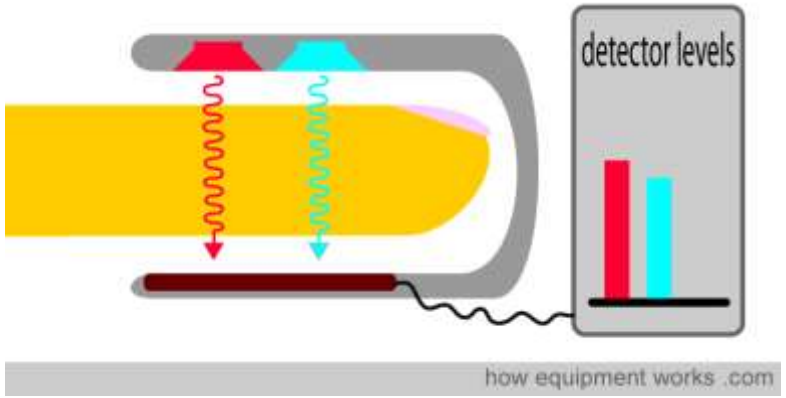


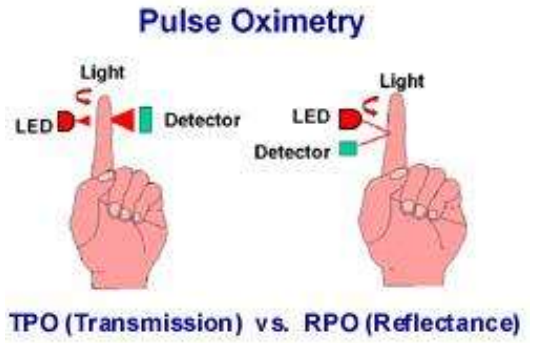
Sensors & Applications – Pulse Ox

- Pulse-Oximetry
  - Measure blood oxygen saturation (Sp02) and calculate heart rate
    - Oxygen molecules attach to hemoglobin
    - Types: Transmission and Reflectance
    - Hemoglobin with and without oxygen absorbs light differently (wavelength of light differs)
      - Oxy Hb absorbs more infrared light than red light
      - Deoxy Hb absorbs more red light than infrared light

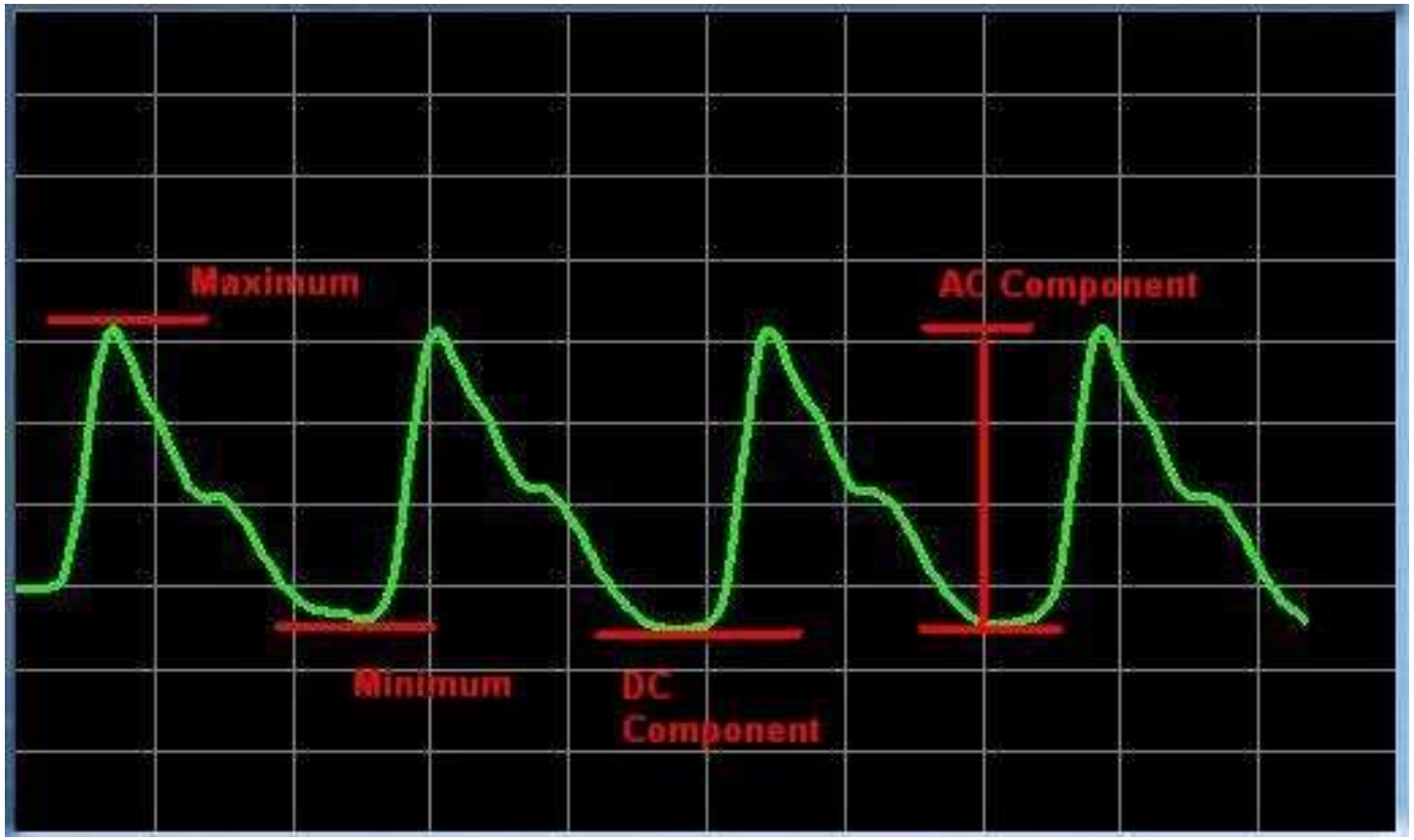
https://www.howequipmentworks.com/pulse_oximeter/







Sensors & Applications – Pulse Ox



Next Week Challenges

- W3_Chal
- W3_Debug