Measurements, Sensors and Data Logging Course

Week 4

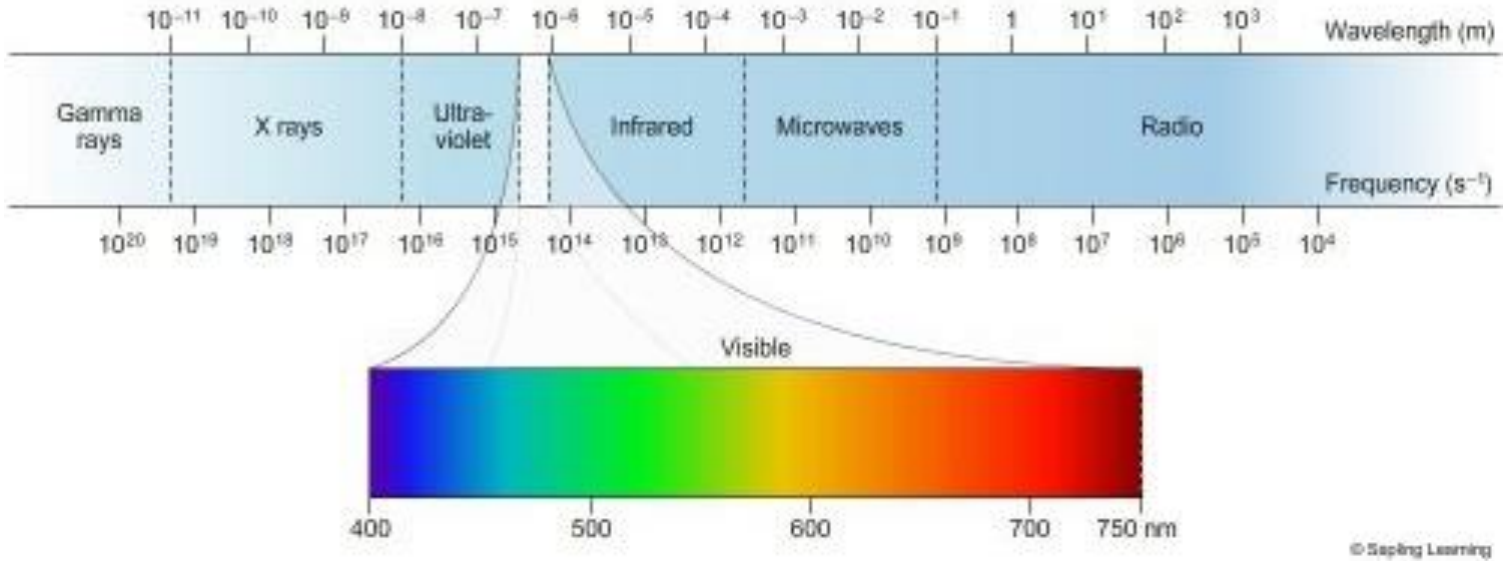
Upcoming Weeks

- Office Hours
  - Monday Nov 15 @ 7:00 PM
- Weekly Session
  - Thursday Nov 18 @ 7:00 PM

Sensors & Applications

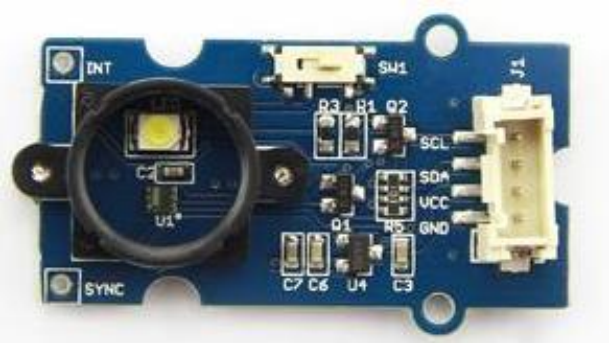
Sensors & Applications: Color Sensors

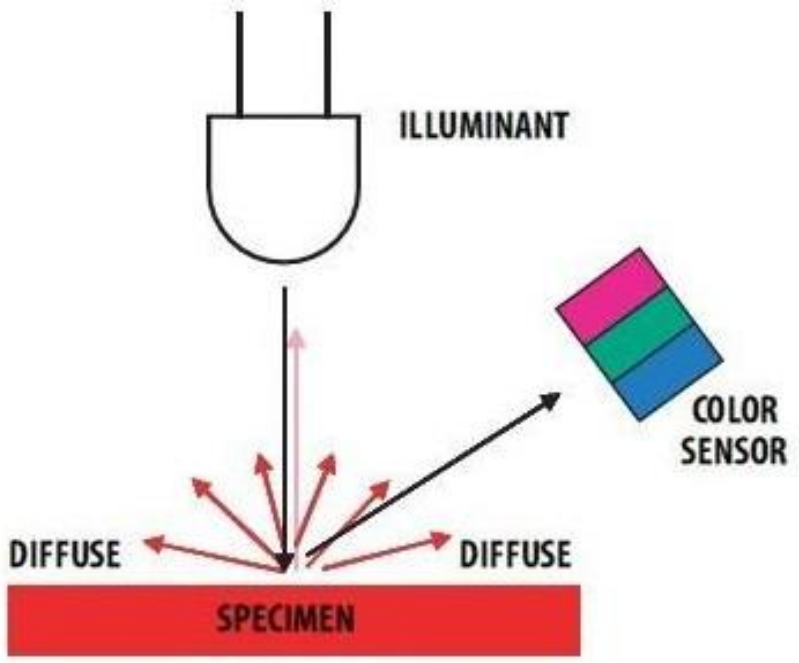
- Think of how Red, Green and Blue combine to make colors
  - Sensors have individual photodiodes that are sensitive to a frequency band of light (color) and measure the intensity of that frequency.
  - Filters are used to make the photodiodes sensitive to limited frequency bands
  - Data from the individual diodes is combined to create a color measurement.



Sensors & Applications: Color Sensors

- Object Color Detection
  - Typical Application: light that shines out, reflected wavelengths are measured
  - Able to measure intensity of ambient light

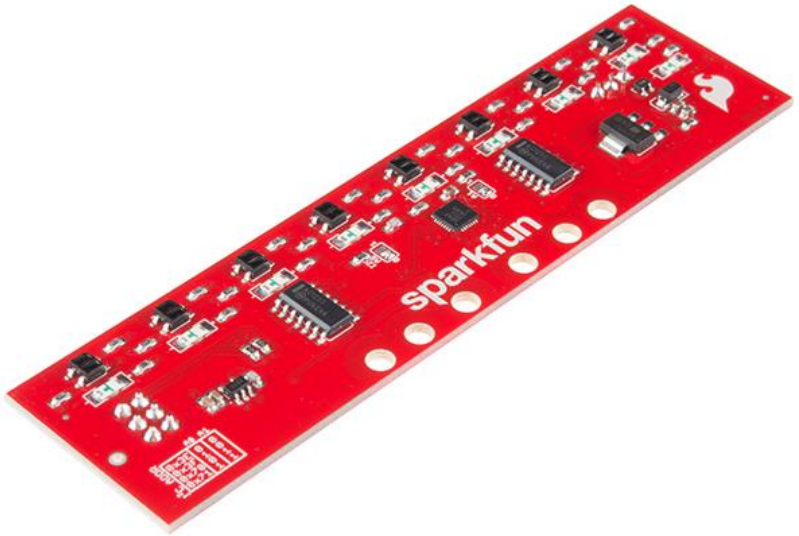




- https://wiki.seeedstudio.com/Grove-I2C_Color_Sensor/

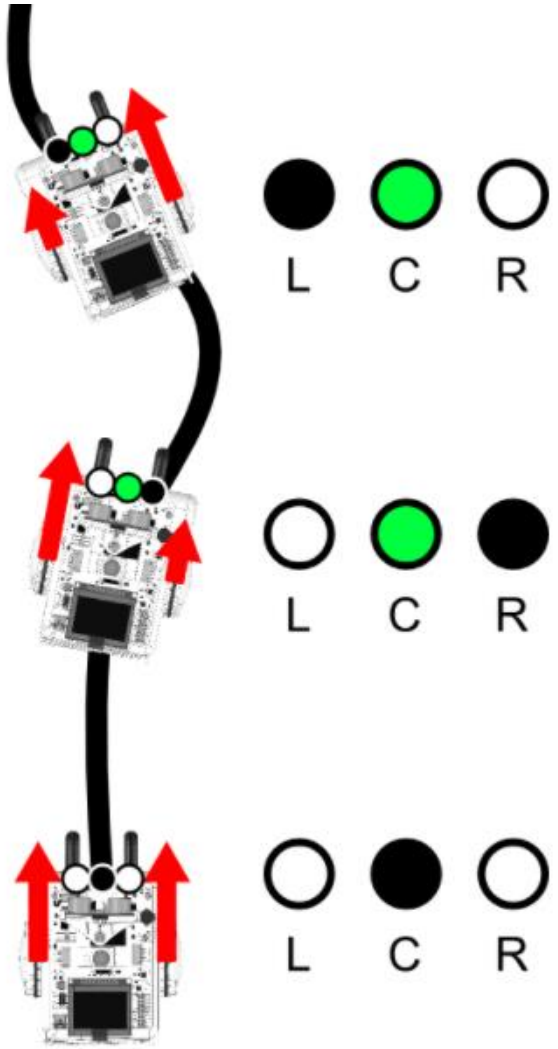
Sensors & Applications – Line Following Sensors

- Typically utilize IR (InfraRed) sensors
  - IR sensor consists of an LED and phototransistor
    - LED emits an IR light (humans an unable to see this)
    - Phototransistor is measuring IR light that is reflected back
      - White surface: reflects light back to the phototransistor
      - Black surface: absorbs light



Sensors & Applications – Line Following Sensors

- In-Use



Operators

Lesson

Code Analysis – C++ Comparison Operators

Operators

```
for(unsigned int i = 0; i < 1<<filterConstant; i++)
```

- Is **i** less than $2^{\text{filterConstant}}$?

==	Equal To	→ TRUE if the left side is **equal to** the right side
!=	Not Equal To	→ TRUE if the left side is **not equal to** the right side
<	Less Than	→ TRUE if the left side is **less than** the right side
<=	Less Than or Equal To	→ TRUE if the left side is **less than or equal to** the right side
>	Greater Than	→ TRUE if the left side is **greater than** the right side
>=	Greater Than or Equal To	→ TRUE if the left side is **greater than or equal to** the right side

- More Information:
  - https://beginnersbook.com/2017/08/cpp-operators/
  - https://www.arduino.cc/reference/en/

Code Analysis – C++ Arithmetic Operators

Operators

+	Addition	`A = 1 + 2` →	`A = 3`
-	Subtraction	`B = 3 - 1` →	`B = 2`
*	Multiplication	`C = 2 * 4` →	`C = 8`
/	Division	`D = 6 / 3` →	`D = 2`
%	Modulo (remainder)	`E = 7 % 4` →	`E = 3`

- More Information:
  - https://beginnersbook.com/2017/08/cpp-operators/
  - https://www.arduino.cc/reference/en/

Code Analysis – C++ Auto-increment and Auto-decrement Operators

Operators

```
for(unsigned int i = 0; i < 1<<filterConstant; i++)
```

- Increment **i** by 1 at the end of the for loop.

++ Auto-increment `i++` → `i = i + 1`

- Increments the value of a variable by 1

-- Auto-decrement `j--` → `j = j - 1`

- Decrements the value of a variable by 1

- More Information:
  - https://beginnersbook.com/2017/08/cpp-operators/
  - https://www.arduino.cc/reference/en/

Code Analysis – `for` Loop

Operators

```
for(unsigned int i = 0; i < 1<<filterConstant; i++)
{/* Do Something */}
```

- Repeat code inside the curly braces $2^{\texttt{filterConstant}}$ times

- `for()` Loops are used to repeat code that appears between its curly braces
- Syntax:

Iterator Variable Initialization

Update Iterator

```
for(initialization; condition; increment)
{
  // Do Something
}
```

End Condition

- More Information:
  - https://www.arduino.cc/reference/en/language/structure/control-structure/for/
  - https://beginnersbook.com/2017/08/cpp-for-loop/

Code Analysis – C++ Assignment Operators

Operators

```
micValueLong += micValue;
```

- Add **micValue** and **micValueLong** then store the result in **micValueLong**

=	Equals	Assigns value of right side to the left side	
+=	Plus Equals	`A += 2` →	`A = A + 2`
-=	Minus Equals	`B -= 3` →	`B = B - 3`
*=	Multiply Equals	`C *= 4` →	`C = C * 4`
/=	Divide Equals	`D /= 5` →	`D = D / 5`
%=	Modulo Equals	`E %= 6` →	`E = E % 6`

- More Information:
  - https://beginnersbook.com/2017/08/cpp-operators/
  - https://www.arduino.cc/reference/en/

Code Analysis – Averaging Filter

Operators

- What is a filter?
  - A filter is used to remove an unwanted component of a signal.
  - For sensor measurements a **low pass filter** is often used to reduce noise or some high frequency component.
  - There are many different types of filters, and numerous ways to implement filters.
- What is averaging?
  - Averaging is taking the mean value of a signal over the sampling period.
- More Information:
  - https://en.wikipedia.org/wiki/Filter_(signal_processing)
  - https://en.wikipedia.org/wiki/Average
  - https://www.mathsisfun.com/mean.html

```
// Average filter
int average;
int sumSamples = 0;
for(int i = 0; i < numSamples; i++)
{
 sumSamples += analogRead(A2);
}
average = sumSamples / numSamples;
```

Lesson 4: Comms

See the output of the light sensor in the Serial Monitor

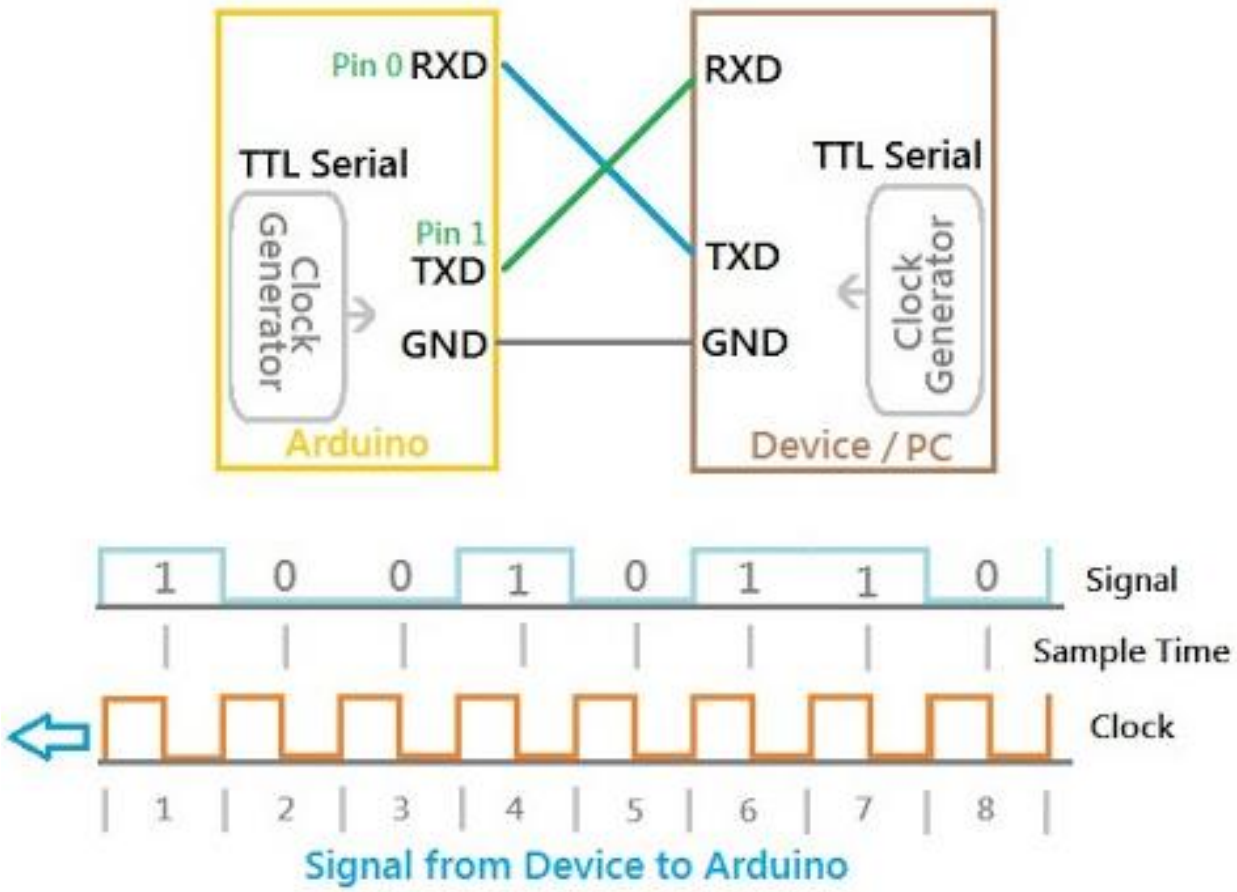
Serial Introduction

Lesson 4: Light Sensor

- What is Serial Communication?
  - A digital signal where data is sent one bit at a time over a single channel.
  - Serial communications include RS232, RS485, UART, USART, USB, Ethernet, CAN, I^2C, SPI, SATA, etc.
  - Serial (without descriptors) typically refers to RS-232 and related communication signaling standards (UART or USART for a microcontroller).
- Where are serial communications used?
  - Internet, computers, cell
- More information:
  - https://en.wikipedia.org/wiki/Serial_communication
  - https://www.codrey.com/embedded-systems/uart-serial-communication-rs232/

Serial Introduction

Lesson 4: Light Sensor

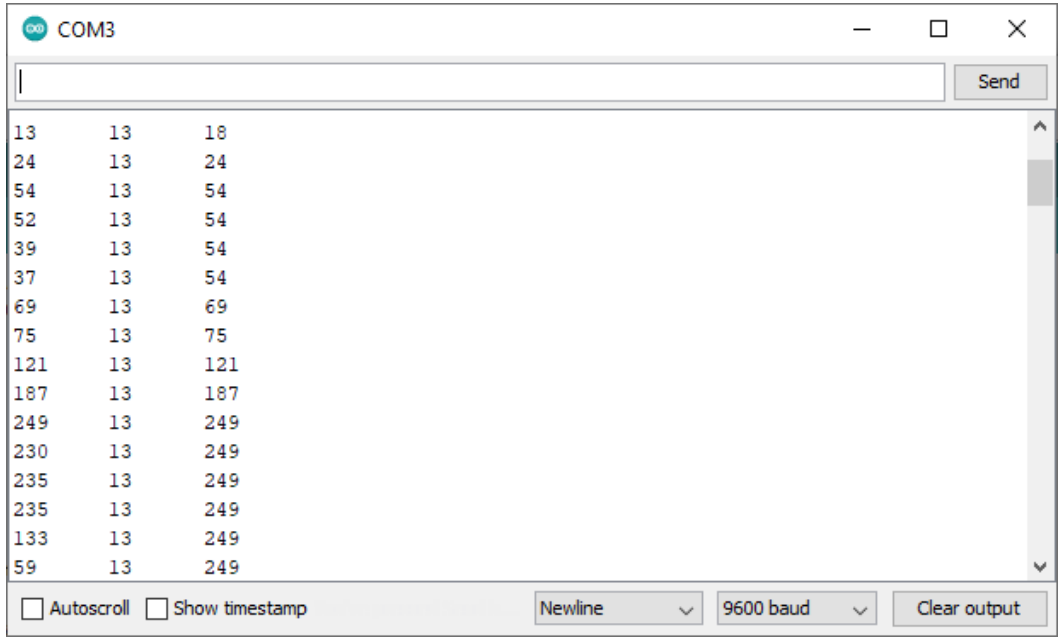


- More information:
  - http://elextutorial.com/learn-arduino/arduino-serial-communication-write-port-example-test-begin/

Open and Upload Sketch

Lesson 4: Light Sensor

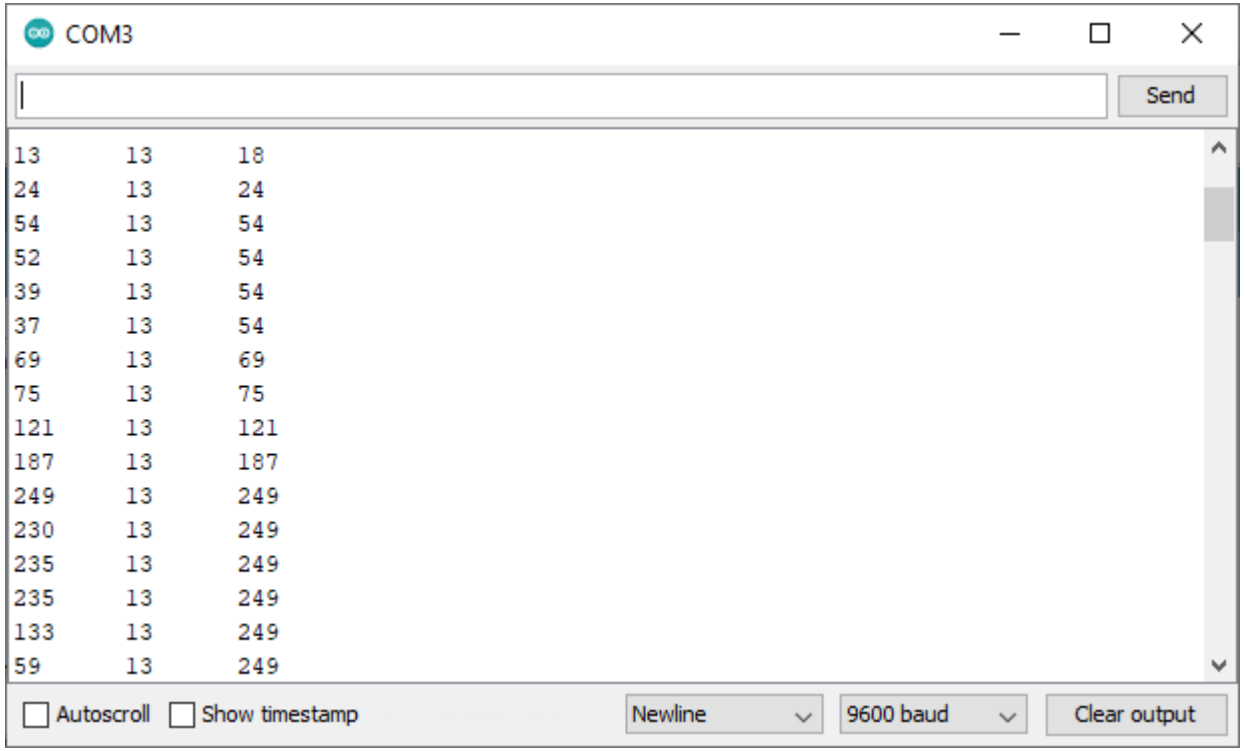
1. Open Light_Serial Sketch
   a. **File → Sketchbook → CrashCourse_Jan → L4_Light_Serial.ino**
2. Verify the sketch by clicking the Verify Button.
   a. The sketch should compile with no errors.
3. Upload the sketch to your Arduino by clicking the Upload Button.
   a. The sketch should re-compile, and then upload to your Arduino.
4. Open the serial monitor.
   a. **Tools → Serial Monitor** (Ctrl+Shift+M)
5. Observe the output in the Serial Monitor



Serial Monitor

Lesson 4: Light 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.



Code Analysis – `Serial` Functions

Lesson 4: Light Sensor

- **`Serial.begin(9600);`**
  - Start the Serial port at a 9600 baud
  - Put this function in the setup() function
  - Must call this function before using any other serial function
- **`Serial.print("string");`**
  - print a string or value to the serial port
- **`Serial.println("string");`**
  - same as print but add a new line character at the end of the string or value
- Special characters:
  - `'\t'` is a Tab character
  - `'\n'` is a New Line (some operating systems [⊞] use `"\r\n"`)
- More Information:
  - https://www.arduino.cc/reference/en/language/functions/communication/serial/
  - https://en.wikipedia.org/wiki/Control_character

Lesson 6: Temp and Humidity Sensor

Using a Library to read a sensor

Temperature Sensor Introduction

Lesson 6: Temp and Humidity

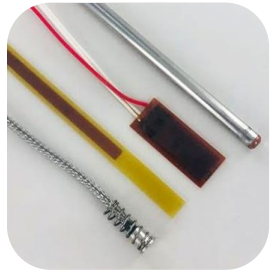
- Temperature Sensor: Measure temperature of object, substance, medium
- Where are they used: Everywhere! fluids, gases, thermostats, electronics, appliances
- How are they used: Depends.
  - For fluid, air and object temp measurement: sensor needs to be in the medium (fluid/air) or against the object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

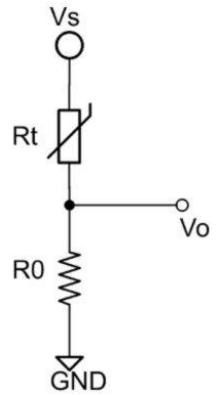
  - RTD (Resistance Temperature Detector): Resistance changes with temperature
    - Current drivers required to measure



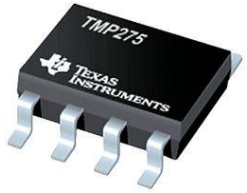
- More Information: https://www.digikey.com/en/blog/types-of-temperature-sensors#:~:text=There%20are%20four%20types%20of,based%20integrated%20circuits%20(IC

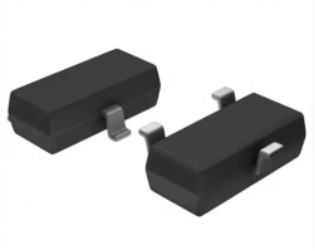
Temperature Sensor Introduction

Lesson 6: Temp and Humidity



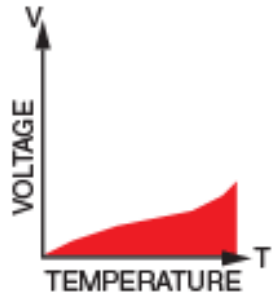
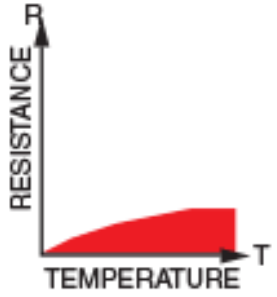
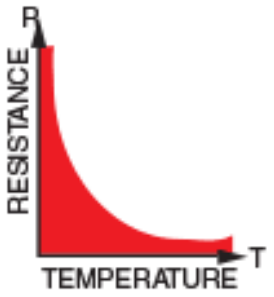
- Sensor Types: 4 most common
  - Thermistor: Resistance changes with temperature
    - Use voltage divider to measure
  - Semiconductor based ICs: measure the physical properties of a transistor





Temperature Sensor Introduction

Lesson 6: Temp and Humidity

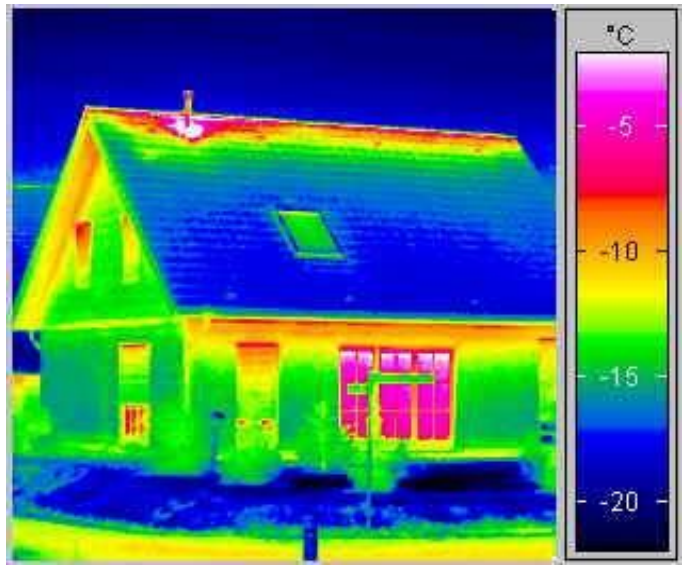
	Thermocouple	RTD	Thermistor	I. C. Sensor
				
Advantages	▪ Self-powered ▪ Simple ▪ Rugged ▪ Inexpensive ▪ Wide variety ▪ Wide temperature range	▪ Most stable ▪ Most accurate ▪ More linear than thermocouple	▪ High output ▪ Fast ▪ Two-wire ohms measurement	▪ Most linear ▪ Highest output ▪ Inexpensive
Disadvantages	▪ 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



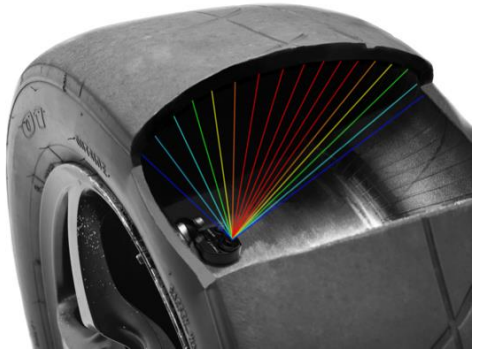
- More information: https://www.flir.com/discover/rd-science/temperature-guns-versus-thermal-imaging-technology/#:~:text=Both%20spot%20pyrometers%20and%20thermal,it%20into%20a%20temperature%20reading.&text=A%20spot%20pyrometer%20reads%20the,of%20the%20entire%20thermal%20image

Temperature Sensor Introduction

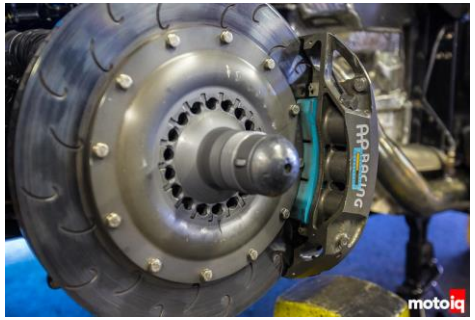
Lesson 6: Temp and Humidity

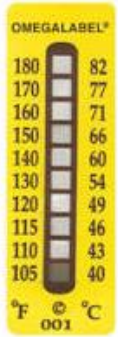
- Temperature sensors on a race car *(x47 – 131)*
  - **Fluid:** coolant *x2*, engine oil *x2*, gearbox oil *x2*, hydraulic fluid *x0-6*
  - **Gases:** ambient air, intake air *x2-4*, tire air *x4*, exhaust *x2-10*, cockpit + HVAC *(x1-5)*
  - **Object:** tire surface *x12-64*, brake rotor *x4*, brake 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)

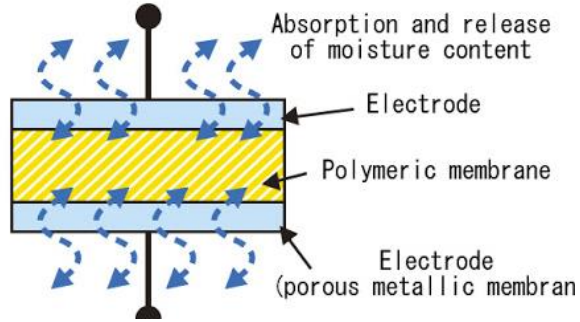
Rotor Temp (paint)

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

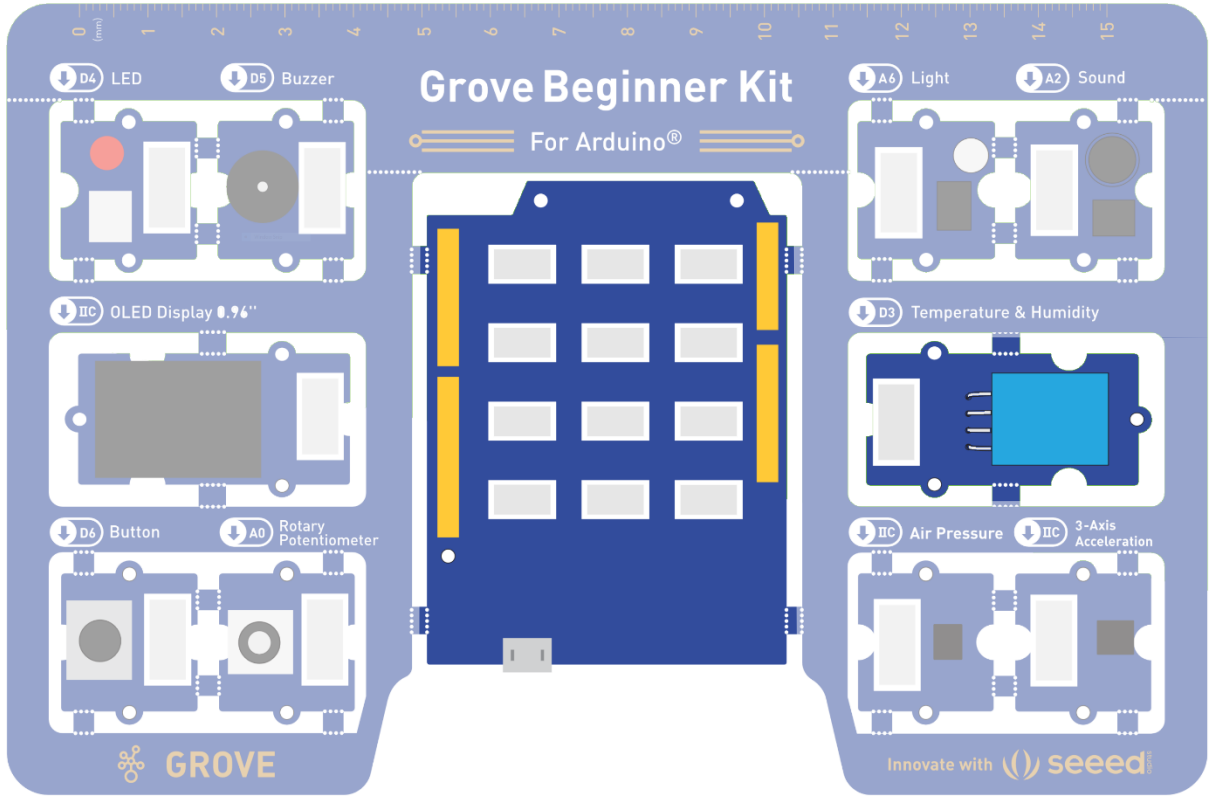
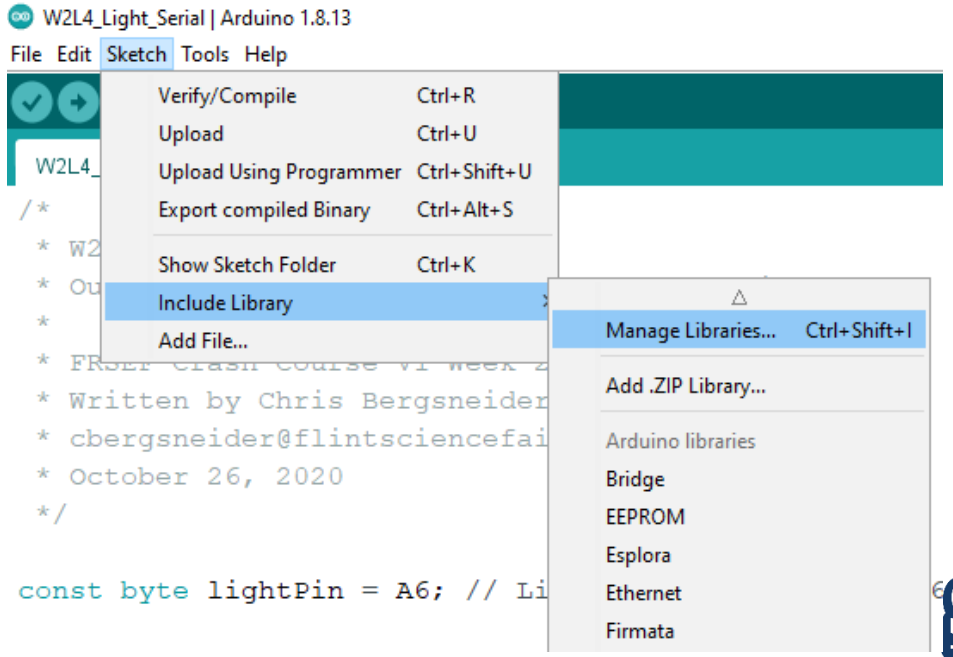


Image modified from https://files.seeedstudio.com/wiki/Grove-Beginner-Kit-For-Arduino/res/Grove-Beginner-Kit-For-ArduinoPDF.pdf

Libraries

Lesson 6: Temperature and Humidity

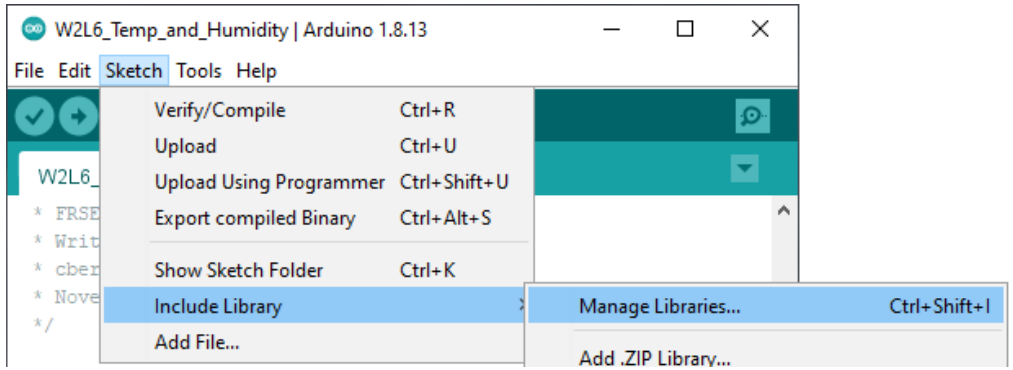
- Library in Arduino
  - What it is: Collecition of pre-defined code.
  - What is does: Provides extra functionalities for sketches, easily.
  - Why: Simplify our sketches, easily integrate components and functions (sensor, display, math, etc)
  - How: Arduino Library Manager
    - *Sketch/Include Library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

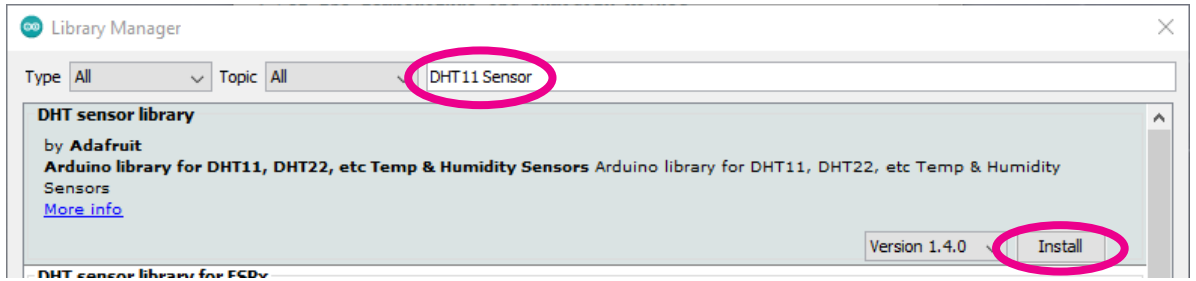


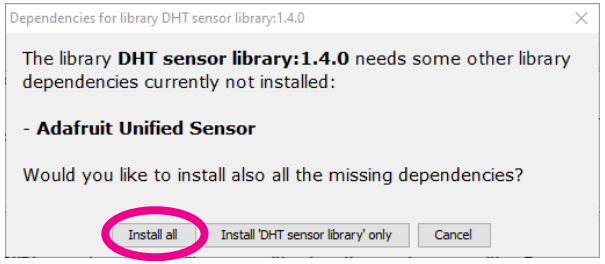
Installing a Library

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

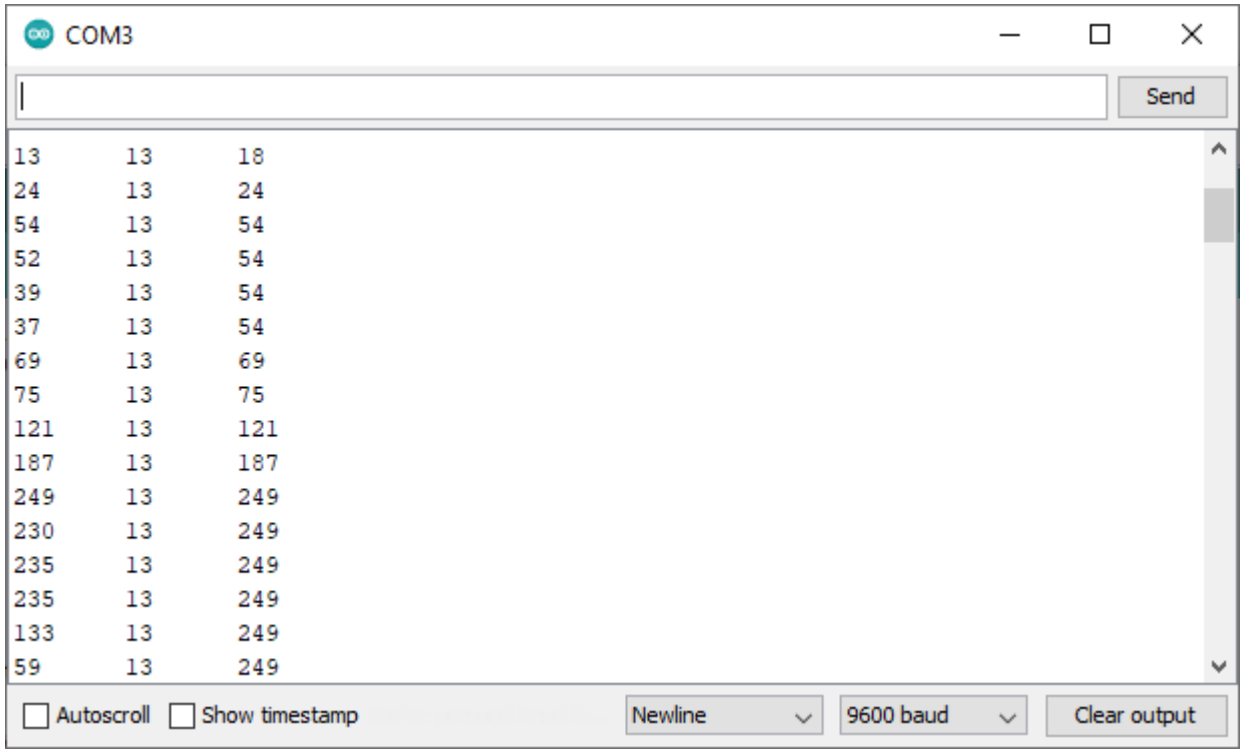
Lesson 6: Temp and Humidity

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

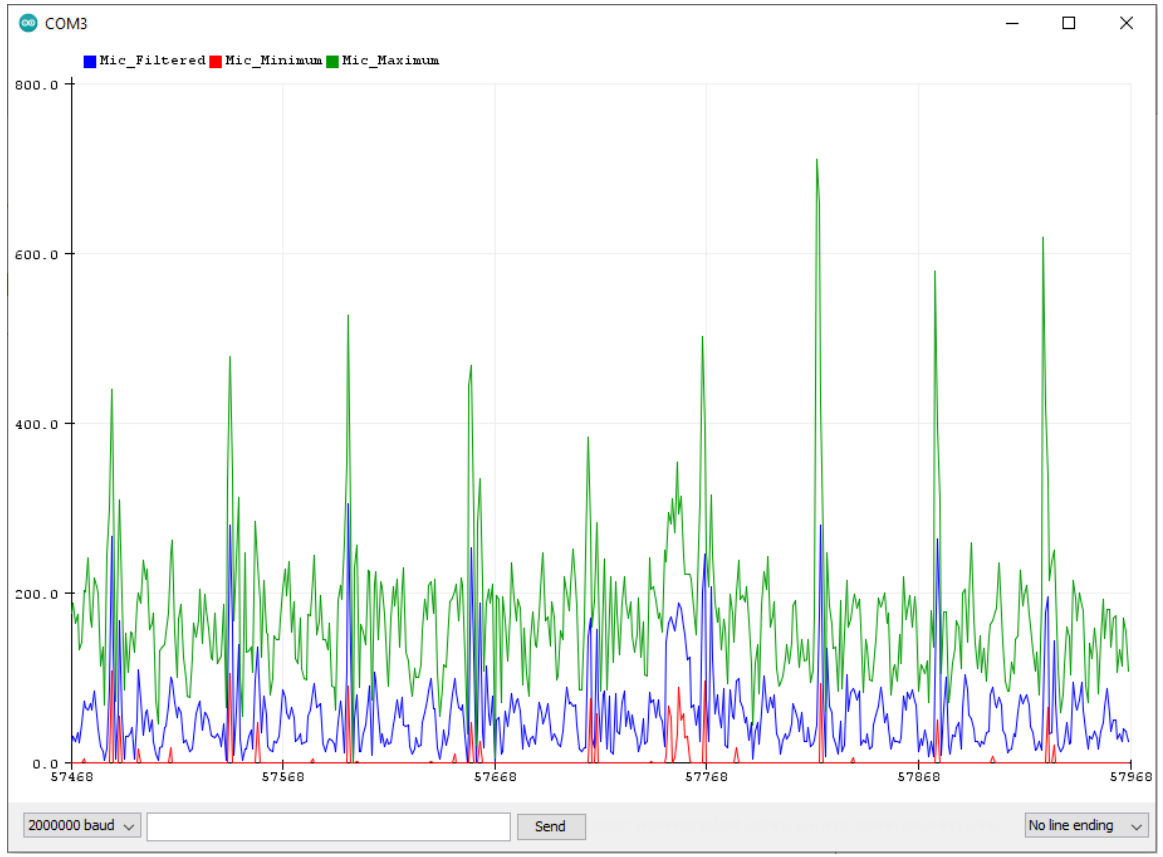
- What is the Serial Monitor?
  - The Serial Monitor is a feature of the Arduino IDE that gives you a serial terminal to see what is being sent to the COM port and allows you to send stuff out of the COM port.
  - We use this for receiving data from the Arduino.
  - We can also use this to help us debug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

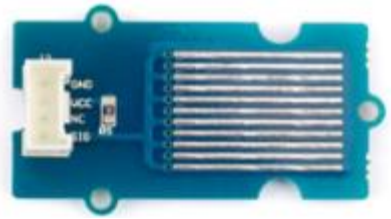
- Output the temperature in F and K
- Change the interval we read the sensor from 1000 msec to 100 msec

Water Sensor

Moisture Sensors

Sensor Calibration + Moisture Sensor

- How it works: uses the conductivity of water to “short” between GND and the signal input
  - Digital Input: water is present or not
  - Analog Input: level of water present
    - Voltage divider with pullup resistor and water between the signal and ground lines.



Hardware

Sensor Calibration + Water Sensor

- What hardware will we need for this Lesson?
  - Grove LED, Buzzer and Display
  - Seeeduino Lotus (Arduino Uno compatible board)
  - Grove Water Sensor
    - Connect to D2 Header
    - Use the provided cable

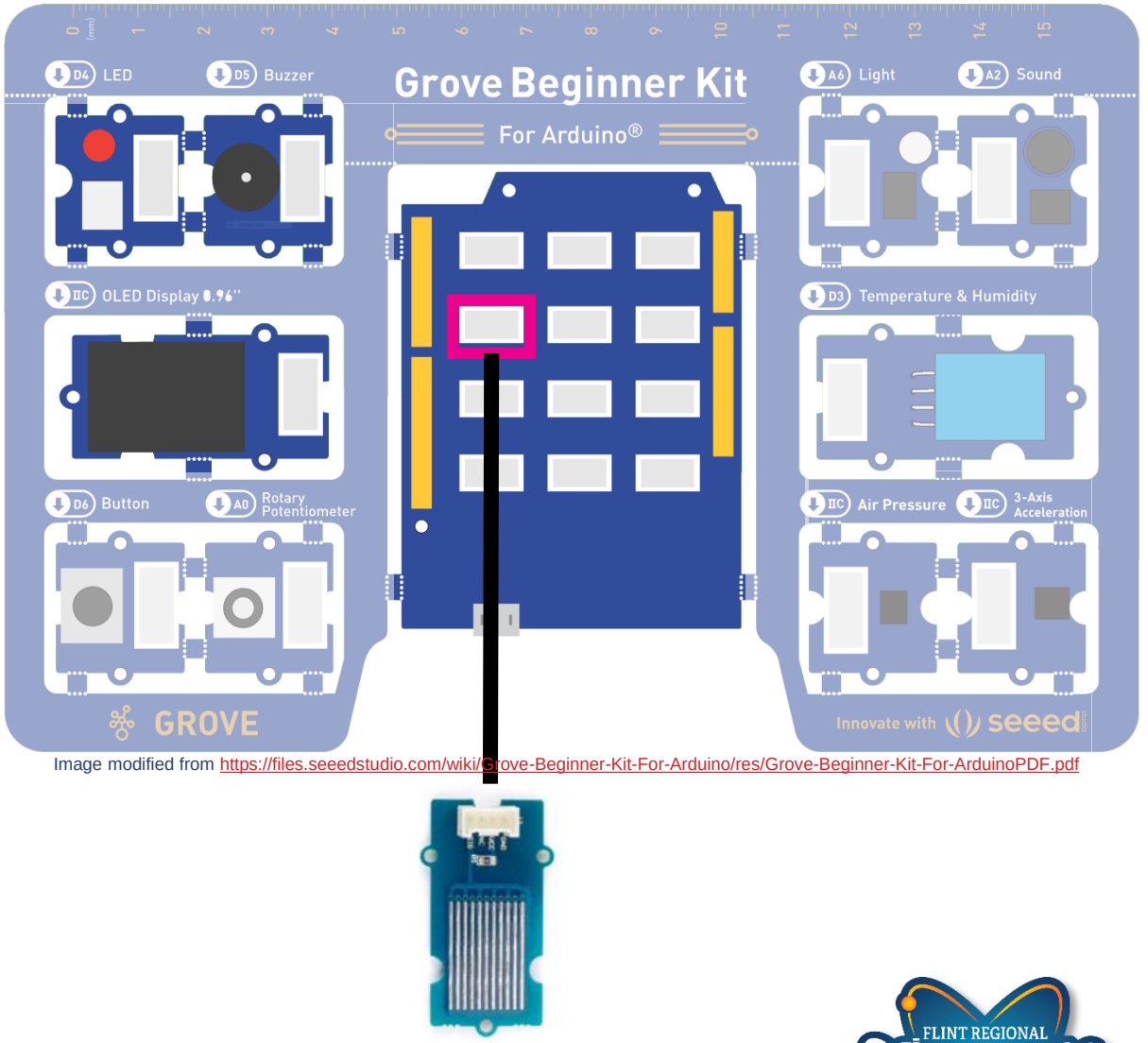


Image modified from https://files.seeedstudio.com/wiki/Grove-Beginner-Kit-For-Arduino/res/Grove-Beginner-Kit-For-ArduinoPDF.pdf

Open and Upload Sketch

Sensor Calibration + Water Sensor

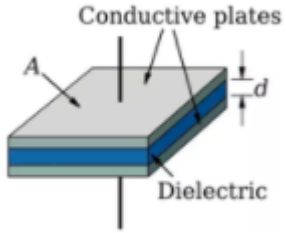
1. Open Simple Datalogger Sketch
   - **File** > **Sketchbook > CrashCourse_Jan > L13_Water_Alert.ino**
2. Upload the sketch to your Arduino by clicking the Upload Button.
   - The sketch should compile, and then upload to your Arduino, assuming you have the correct
3. Touch the water sensor, see what happens.
   - What happens if you place a drop of water on the sensor or dip the sensor into a cup of water?

Soil Moisture Types

Sensor Calibration + Water Sensor

- Common Types
  - Resistive
    - Two probes
    - Current passes through the soil and the resistance value is calculated to measure soil moisture content.
    - Problems: sensor corrodes and resistance continually increases, fertilizer and nutrients can affect resistance
    - Quantitative measurement.
  - Capacitive
    - Single probe, soil contact with electrodes not required (less corrosion)
    - Soil + water form a dialectric, similar to a capacitor
      - Capacity of soil change with change of moisture content
    - Quantitative measurement

https://ucanr.edu/sites/CE_San_Joaquin/files/35895.pdf



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

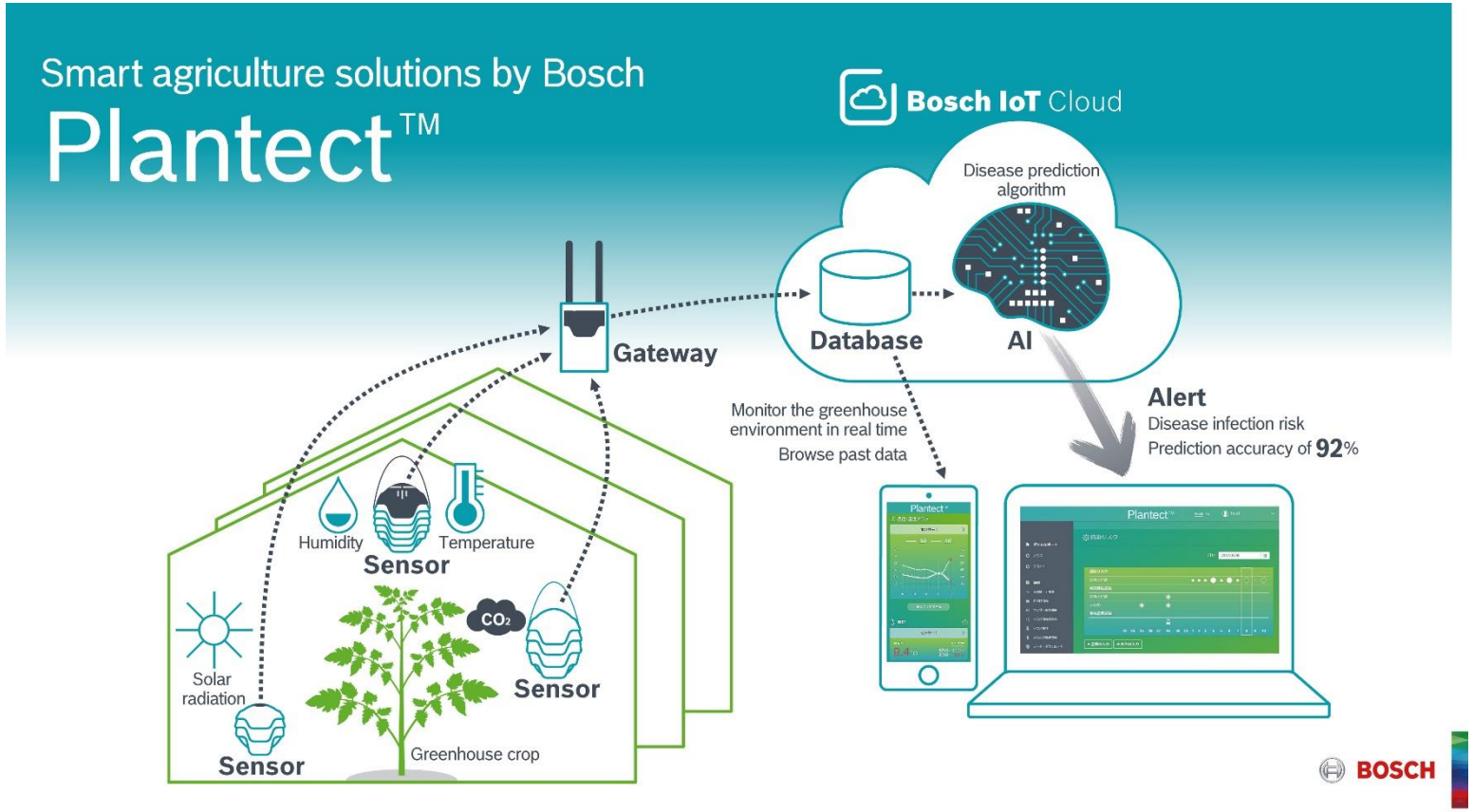
- What can we measure for agriculture applications?
  - Individual plants
    - Environment
      - Temperature
      - Air Quality
      - Air Content: CO, CO2, Oxygen
      - Humidity
    - Soil: moisture, temperature
    - Fertilizer
    - Plant Height
  - Large Scale: entire fields
    - Optical (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

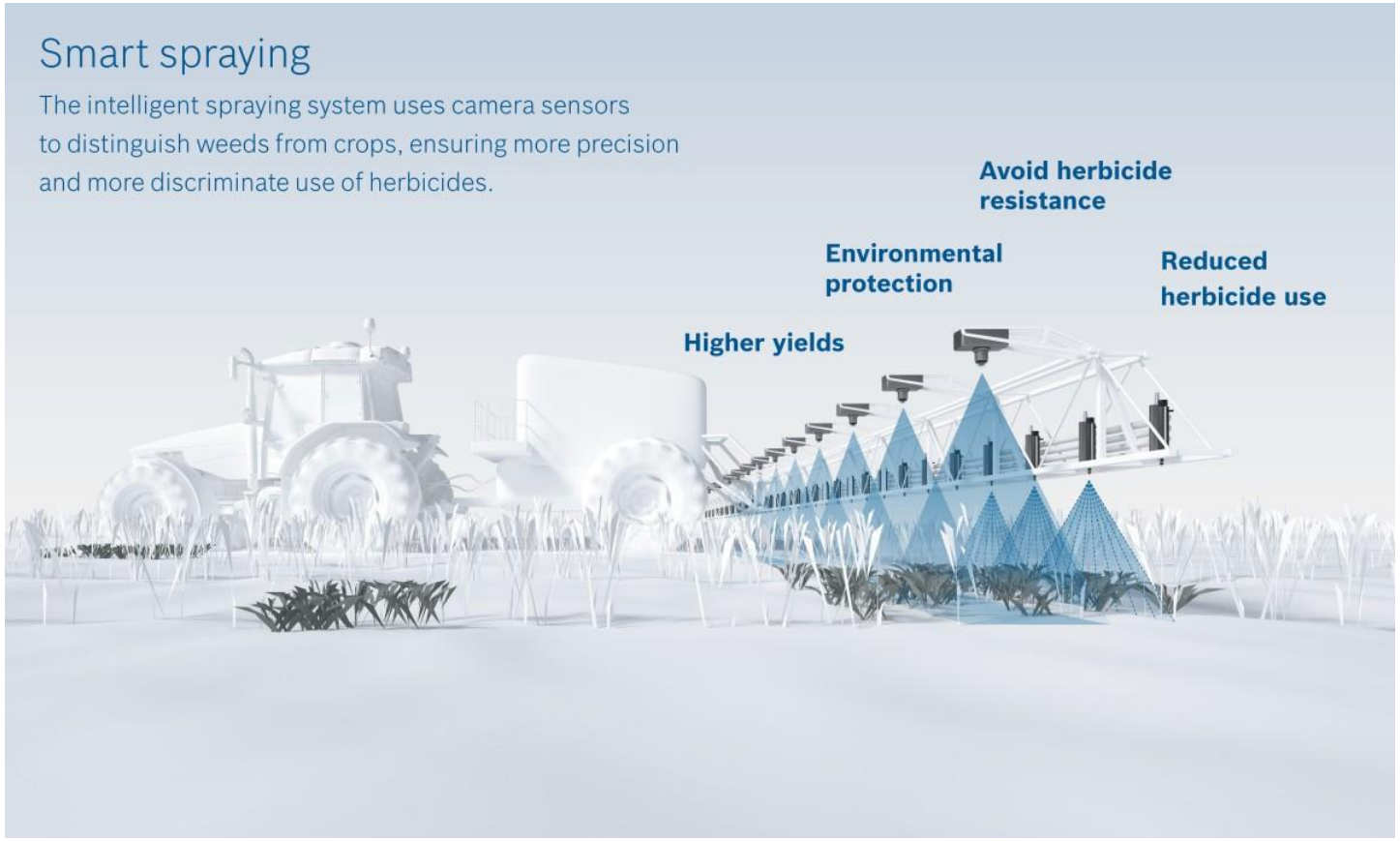
https://www.bosch-presse.de/pressportal/de/en/smart-agriculture-101824.html

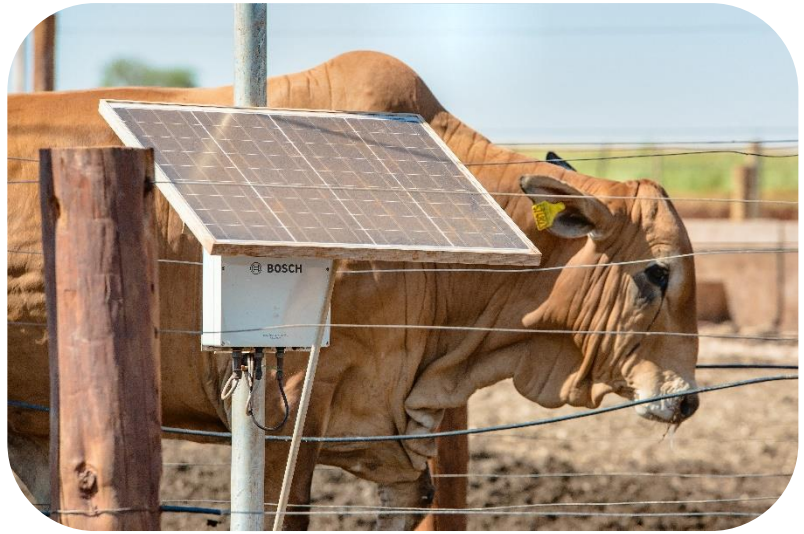


Agriculture Applications + Livestock

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