Calibration (pH Sensor)

IMPORTANT: *The longevity of the pH sensor and the quality of the data it produces will be drastically reduced by leaving the probe exposed to the air or in a solution for an extended period of time (more than a few minutes). When it is not being used, keep the cap screwed on with enough of the electrode solution (1M KCl) so that the glass bulb is fully submerged.*

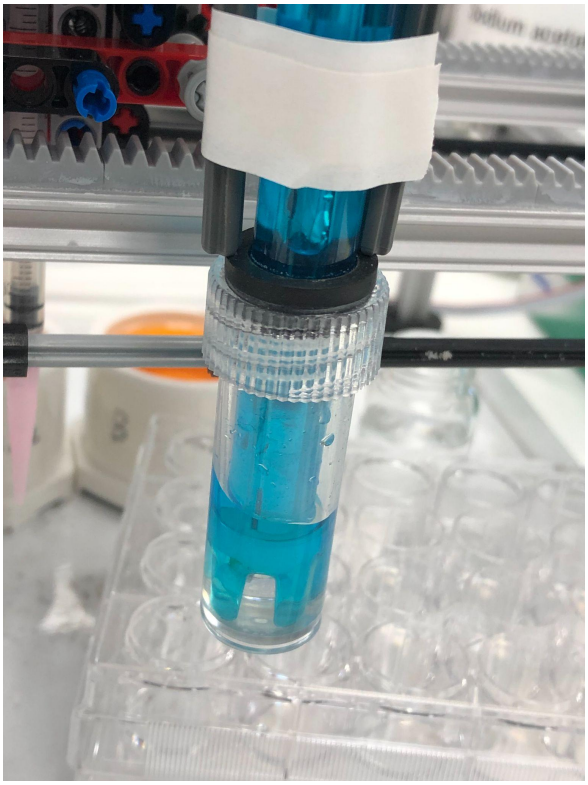


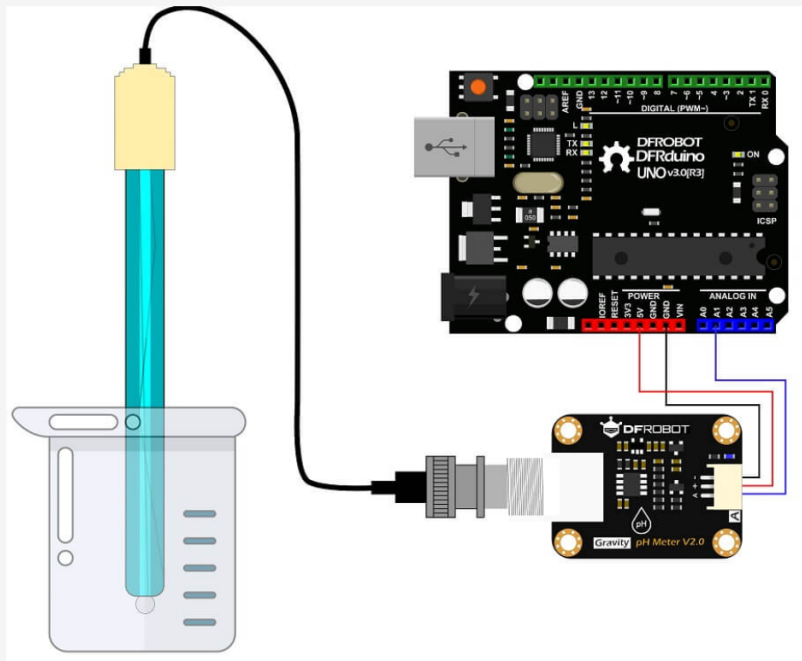
Fig. 12: *When not in use, make sure that the cap containing electrode solution is screwed on and the bulb is fully submerged inside the solution.*

The pH sensor is an electrochemical cell that returns a voltage value that is linearly proportional to the pH of the solution being measured. Calibration involves fitting a slope and intercept to this linear relationship so that a voltage value is associated with the correct pH. To do this, we will use buffer solutions. In this experiment, our pH values are expected to be between 4 and 7, so we do not need to calibrate outside this range (only need 2 point calibration).

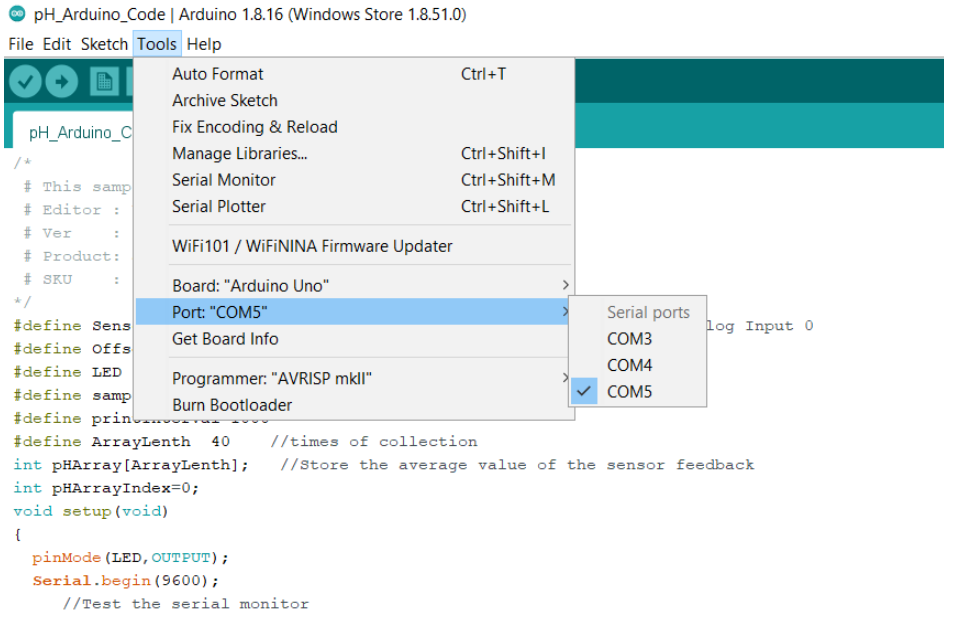
1. Fill up some glass sample tubes with the color coded buffer solutions. **DO NOT** stick the pH sensor in the buffer solution reservoirs, as this can contaminate them.

Fig. 13: *Pour the color-coded buffer solutions in sample tubes before calibrating.*

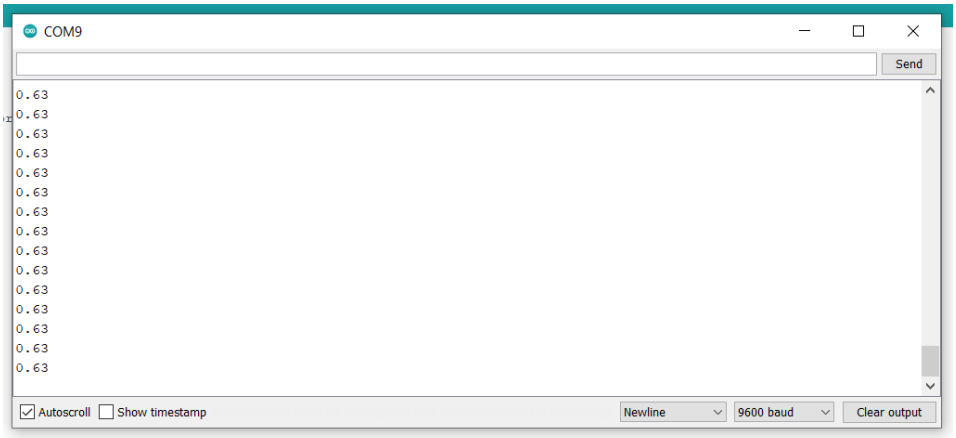
2. Make sure the pH meter is connected to the arduino as shown in the diagram below. Ensure the arduino is also connected by USB to your computer.



3. Open the code provided in the Arduino IDE, ensure it compiles, and upload it to the Arduino by identifying the correct port (you can check what port it is by going to *Tools - Port* , in the IDE).



4. Press the arrow in the top left corner to upload the code to the Arduino. The code will output a pH value (calculated by multiplying the voltage by a slope, and adding an offset). Alter the code so that it prints the voltage instead. You can monitor the voltage by opening up the *serial monitor* (go to *Tools - Serial Monitor*)
It will print the voltage (a value from 0 - 2) about once every second.



5. Raise one of the buffer solutions to the pH sensor and allow the bulb to be completely submerged for 15-30 seconds.

Fig. 17: *Submerge the pH sensor in one of the buffer solutions, as shown.*

Record the **voltage** from the serial monitor once the value stabilizes. Rinse the bulb in DI water for a few seconds and wipe lightly with a kim wipe to remove any moisture. Repeat this again with the other buffer solution and record that voltage. Use these values to determine the proper slope and offset that would return the correct pH values: Ex:

pH 4 Buffer Solution - 0.86 V
pH 7 Buffer Solution - 1.79 V

Slope = (1.79 - 0.86)/(7-4)
*Offset = 4 - slope*0.86*

Replace the slope and Offset values in the code. Then alter the code back so that it prints out **ONLY** the pH.

```
*/
#define SensorPin A1          //pH meter Analog output to Arduino Analog In
#define Offset 1.24           //deviation compensate
#define LED 13
#define samplingInterval 20
#define printInterval 1000
#define ArrayLenth  40    //times of collection
int pHArray[ArrayLenth];   //Store the average value of the sensor feedback
int pHArrayIndex=0;
void setup(void)
{
  pinMode(LED,OUTPUT);
  Serial.begin(9600);
    //Test the serial monitor
}
void loop(void)
{
  static unsigned long samplingTime = millis();
  static unsigned long printTime = millis();
  static float pHValue,voltage;
  if(millis()-samplingTime > samplingInterval)
  {
      pHArray[pHArrayIndex++]=analogRead(SensorPin);
      if(pHArrayIndex==ArrayLenth)pHArrayIndex=0;
      voltage = avergearray(pHArray, ArrayLenth)*5.0/1024;
      pHValue = 3.22*voltage+Offset;
      samplingTime=millis();
  }
  if(millis() - printTime > printInterval)   //Every 800 milliseconds, print
  {

    Serial.println(voltage,2);
        digitalWrite(LED,digitalRead(LED)^1);
        printTime=millis();
  }
```

Figure 18: *Location of the offset, slope, and print values in the code. Alter "voltage" so it says "pHValue" instead*

You can reconfirm that the pH meter is performing well by looking at the serial monitor and immersing the sensor in the solutions again.

Reupload the code to the arduino and then make sure you **close the serial monitor and IDE**. (if you don't you won't be able to monitor the serial port from your jupyter notebook.

6. Replace the COM PORT with the correct one in your jupyter notebook (corresponding to the arduino). Make sure to **close the port after defining it**. (*ser.close()*).

```
#---------------Import Serial to Monitor pH from Arduino port ---------------
import serial
ser = serial.Serial('COM7')
```

```
ser.close() # use this function to keep serial monitor closed until called
```