pH Sensor for Agricultural Applications

Sanya Kaluarachchi
Department of Bioengineering
University of California, San Diego
skaluarachchi@ucsd.edu

Erika Munekata
Department of Bioengineering
University of California, San Diego
emunekata@ucsd.edu

***Abstract*— This paper examines a possible instrumental design based on optimizing efficiency, accuracy, safety, durability, and cost in designing a pH sensor. Using an operational amplifier, microcontroller, resistors, thermistors, momentary push buttons, and switches, along with an LED light for easily read outputs, an effort was made to better understand how existing pH sensors can be improved upon. The device is broken down into four primary functions, including an initial signal acquisition from both a pH electrode and thermistor, signal filtration from the differential amplifier, signal analysis using an Arduino microcontroller, and finally signal output to the LED light. The design has possible future applications for mitigating the dangers associated with contamination into agricultural lands, and also potentially in regards to water treatment and safety.**

I. Introduction

To mitigate both the effects of global warming and the continued spread of industrial and residential developments into agricultural lands, several efforts are being taken to investigate the safety of the everyday consumer's food.

A vital part of growing commercial foods is efficiency. That is, a crop's yield and nutrient density are dependent on the soil's pH they are cultivated in. PH controls many chemical and biochemical processes that take place, and can therefore influence the reactions and forms of different nutrients in soil. This in turn regulates their availability and as a result, the soil and crop productivity. Common pH levels for most crops lie between 5.5 and 7.

Soil pH can be unknowingly altered by weather conditions and industrial waste through altering the chemical environments to where necessary reactions are unable to take place, and the addition of ions such as sodium, calcium, and potassium. Rainwater can lead to soil becoming more acidic due to carbonic acid production in the atmosphere and leaching, which is a process by which water-soluble nutrients and ions drain with the abundance of water. Temperature can also affect reaction rates and water content of the soil, leading to many factors that need to be considered to maintain ideal conditions [1].

The goal of the sensor designed in this report is to measure and display the pH levels of soil in fields through analyzing the data collected by an electrode and thermistor, and the use of three LEDs for simple but effective information relaying. By using the sensor, farmers and others will be able to monitor the pH levels of their soil and therefore more easily manage the conditions of their crops.

A. Equations

As detailed in Equation (2), pH is the negative logarithmic measure of the hydrogen ion concentration present in a given sample. Equation (1) refers to the Nernst Equation, which is used to quantify the relationship between temperature, pH, and electric potential. *R* is the gas constant, *T* is the temperature in Kelvins, *F* refers to Faraday's constant, E_0 refers to the standard electrode potential, and *Q* represents the reaction quotient. In this case, since half of a gaseous hydrogen atom, or H2, is involved in the ionization to H+, the equation can be simplified to allow [H+] to be the reaction quotient, and n to be Hydrogen's valence of 1, and thus the two equations can be combined to find Equation (3).

$$pH = -\log[H^+] \quad (1)$$

$$E_{cell} = E^0 - (RT/nF)*\ln Q \quad (2)$$

$$E = E^0 - (RT/F)*pH \quad (3)$$

Additionally, for calculating the temperature associated with the thermistor measurements, Equation (4) is used, detailing the Steinhart-Hart equation. A, B, and C are constants specific to the thermistor, and the values for this thermistor specifically are detailed more in the Supplemental (1) Arduino code.

$$(1/T) = A + B(\ln R) + C(\ln R)^3 \quad (4)$$

In addition, Equation (5) details the gain, G, that is possible for the AD622 with the use of a singular resistor, R_G. In addition, Equation (6) and Equation (7) both detail the relationship between voltage and temperature for the thermistor of interest.

$$G = 1 + (50.5\ k)/R_G \quad (5)$$

$$R_T = R_0 * e^{\beta(1/T - 1/T_0)} \quad (6)$$

$$V_{out} = V_{in} * (R_T / R_T + R_1) \quad (7)$$

II. Methods

A. Signal Acquisition

PH electrodes use two electrodes, connected on a singular measurement device. One side functions as a reference electrode and directly interacts with the sample, whilst a secondary electrode will act as a standard pH electrode. The standard pH electrode has its own fixed solution of hydrogen ions. At the end of this pH electrode, there's a glass bulb in contact with the sample of interest, that can use a distribution and concentration difference across the glass to measure a membrane potential [3].

This potential will be compared to the reference electrode, which has an open junction that is in direct contact with the sample of interest. Due to the messier particulate matter of soil, it is preferable to use an open junction so that the electrolytic solution of the reference electrode may interact with the sample, though have the electrolytic solution be gel-based so as not to allow large particulate matter into the electrode itself. Although particulate matter is ideally blocked from moving into the electrolytic solution, ionic flow is still possible through the reference electrode solution.

Additionally, as the ambient temperature affects pH and electric potential as detailed by the Nernst equation, there must be some temperature-informing component to the system. Thus, there is also the incorporation of the LM35 temperature sensor, which uses the aforementioned equations to map changes in resistance to changes in ambient temperature and thus acts as a temperature sensing gauge. Its circuitry setup is established in Figure (1) below, with the inclusion of a voltmeter to denote its connection to both the microcontroller's ground and to an analog input pin. The voltmeter is in parallel with the thermistor, which is approximately 10 kΩ at an ambient room temperature of 25℃ but varies in accordance with Equation (6) as explained prior.

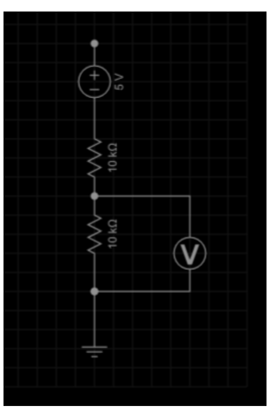

Figure 1. Thermistor circuitry, with 5 volt power supply, comparative resistor, and voltmeter indicating the Arduino pinout

B. Signal Filtration

Although unlikely for the given samples of interest, the expected pH could range from a pH of 2 to a pH of 11, and thus one would likely expect an output range of approximately -200 to 200 mV based on theoretical calculations using the Nernst equation. Additionally, pH electrodes of interest such as that by Mettler Toledo or Hannah Instruments take approximately 1 minute per sample reading, and thus the pH sensor is approximated as an AC voltage source in the Figure (2) schematic. To reduce possible noise and amplify the signal, the AD622 operational amplifier was used, powered by a 6-volt battery supply powered by two individual 3-volt batteries. A theoretical gain of 10 is set with a singular 5 kΩ resistor, and thus an output voltage range from -2 volts to 2 volts is expected to be fed into an analog input pin for the Arduino microcontroller.

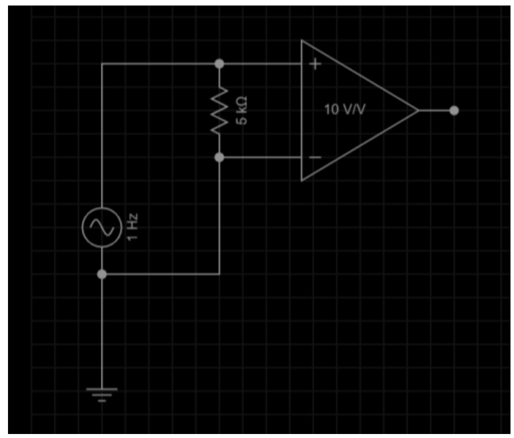

Figure 2. Schematic representation of AD622 operational amplifier with R_G and AC source representing the input pH signal.

C. Signal Analysis

The Arduino UNO will be used as a microcontroller to further analyze the data and output simplistic data to the user. The thermistor circuit and the pH electrode will be connected to the analog pins, and four digital pins will be connected to a switch gate, as well as three different LED lights.

The sensor takes readings based on a single momentary push button switch connected to the Arduino microcontroller. When the button has not been pushed, the Arduino will read the button's state as being LOW or connected to the Arduino's ground. In the event that it's pressed, the code associates the button state as HIGH, or as the 5 V of the Arduino power supply, and takes readings from both of the analog pins. These readings are saved in a pre-existing array, and each button click event is associated with another loop iteration. Once the button has been pressed twice, the two versions of measurements taken are calculated into a slope mapping the found voltages to a pH value. Using the slope, the pH 7 exact found value from the second buffer, and then the corresponding electric potential value, is used to establish the current sensitivity and linear relationship between pH and electrical potential and inform the final pH values calibration.

The Arduino code is given in Supplemental (1) section below.

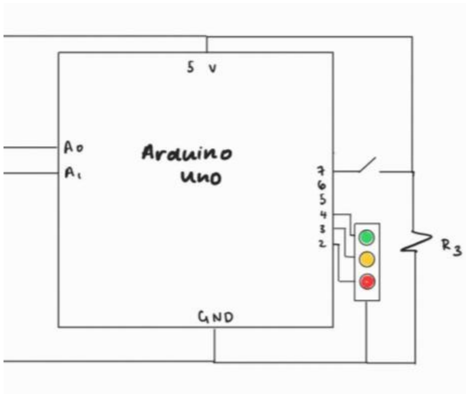

Figure 3. Arduino Uno microcontroller pinout diagram.

D. *Signal Output*

Upon taking three readings, the Arduino will compare the found pH sample value to the pre-programmed range of interest of approximately a pH of 5.5 -7. The Arduino will then light up either the red, yellow, or green LEDs if the pH value is in this respective range. This range is currently set to red: outside of the 5.5 and 7 range; yellow: outside of a value 1 and 14, which is assumed to be an erroneous measurement; green: within the 5.5 and 7 range.

III. Discussion

A. *Limitations*

- *Buffers expiring*: The buffer solutions act as calibration for the pH electrode, allowing the user to account for errors in deviations in the relationship between pH and electric potential amidst varying ambient temperatures. By being a known pH buffer, the Arduino microcontroller can better establish the linear relationship between pH and electric potential beyond the theoretical values. However, given that the buffers are subject to oxidation and reduction with continued exposure, the buffer solutions can lose their value as comparative measurements. It is significant to use buffers that list both their expiration dates and also their instructions detailing ideal storage conditions.
- *Electrolytic solutions become contaminated*: Although the given pH reference electrode design of being an open junction and gel electrolytic solution is ideal to prohibit the contamination of the pH electrode solution, it still is not an ideal system. Theoretically, particulate matter from soil solutions will still enter through the open junction into the gel electrolytic solution, causing ionic interference with potential readings. Thus, the pH electrode must be switched out after some amount of indeterminate uses, depending on the contamination potential of the sample of interest and the junction of the used pH electrode. Further field testing with the pH electrode must be done to see at what level of contamination the readings are no longer valuable measurements.
- *Users' steps must specifically match the Arduino code*: Given that the Arduino code is coded to count the number of button clicks as iterations, and also stores those values in an array for further mathematical analysis, it is significant that the user takes measurements and calibrates in the correct series of instructions. Slight deviations from initially calibrating with a pH 4.0 buffer, pH 7.0 buffer, and then finally taking a sample reading, will make the output's data useless. Possible solutions include the integration of more push buttons connected to different Arduino inputs, as this would allow for the storage of differing values based on which specific button is pressed, rather than simply the number of iterations.

B. *Future Improvements*

Currently, the Arduino is hardcoded to output signals to the user based on a set pH range of approximately a pH of 5.5-7. However, there is a very diverse range of pH values corresponding to various agricultural products [2]. Ideally, future improvements involve allowing integration of more switches and momentary push buttons, or an LCD to allow for user inputs specifying the range of interest

Additionally, for the instrumentation to have the actual functionality to aid in agricultural analysis, the device needs to improve its portability. By further investigating the scalability of the Arduino microcontroller and possibly further combining the pH sensor and thermistor into a singular signal acquisition system, the device has more theoretical usefulness for actual field use.

IV. Conclusions

As global conditions continue to change, there is an increasing concern surrounding food production and safety. By creating an alternative pH measuring device that is cheaper, easier to use, and just as effective, farmers can maximize crop production and nutrient density by adding the necessary soil additions to maintain the ideal pH value when alerted with one of the three LEDs.

The pH sensor in this report uses a standard pH electrode, LM35 temperature sensor, AD622 operational amplifier, Arduino UNO, and three different colored LEDs to collect the pH concentration and temperature, filter the signal, and send it through the Arduino, which analyzes the data and turns on the respective LED. The sensor is operated by a single button, which is connected to the Arduino, and with specific inputs can record and store data for future use.

The limited lifespan and design of the current model of electrode used can be more specified and improved upon as the sensor begins testing. Once a final model is complete, other iterations of the sensor can be used for water and other mediums that may require pH measurements with a variety of set pH levels.

ACKNOWLEDGMENT

We would like to thank Dr. Cauwenberghs and our BENG 186B TA's Vikrant, Adyant, and Samira for their guidance through this course and project, as well as our BENG 152 instructional team, Dr. Taylor, Carter, and Jeffery, who guided us in use of the Arduinos and circuits.

REFERENCES

[1] Oshunsanya, S. O. (2018, December 18). *Introductory chapter: Relevance of soil ph to agriculture*. IntechOpen. https://www.intechopen.com/chapters/64810

[2] Blakey, Dustin. " Adjusting Soil pH in California Gardens." *UC Agriculture & Natural Resources Yard and Garden*, https://doi.org/https://escholarship.org/uc/item/7419v4x5.

[3] "Ph Electrodes." *Chemistry LibreTexts*, Libretexts, 29 Aug. 2023, chem.libretexts.org/Bookshelves/Analytical_Chemistry/Supplemental_Modules_(Analytical_Chemistry)/Analytical_Sciences_Digital_Library/Courseware/Analytical_Electrochemistry%3A_Potentiometry/05_pH_Electrodes.

SUPPLEMENTALS (1)

The following supplemental information details the Arduino code to be associated with the detailed circuit:

```
//input pins
constant int thermistorpin = A0;
constant int switchpin = D7;
constant int signalpin = A1;

// constants and variable initialization
float measurementEV[3];
float measurementpH[3];
float measurementThermistor[3];
float Vo, RT, sensorValue, logRT, T, phValue;
int R1 = 10000;
int lastswitchState = 0;
buttonCounter = 0;
float c1 = 1.009249522e-03, c2 = 2.378405444e-04, c3 = 2.019202697e-
07;

void setup()
{
  Serial.begin(9600);
pinMode(thermistorpin], INPUT_PULLUP);
pinMode(signalpin, INPUT_PULLUP);
pinMode(switchpin, INPUT);
pinMode(0, OUTPUT);
pinMode(4, OUTPUT);
pinMode(3, OUTPUT);
pinMode(2, OUTPUT);
}
 // read the state of the pushbutton value:
 buttonState = digitalRead(switchpin);

  // check if the pushbutton is pressed. If it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  } else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}

void loop()
{
while buttonCounter < 3 {
//see if button/switch changed
switchState = digitalread(switchpin)
if (switchState != lastswitchState) {
If (switchState == HIGH) {
//calculating temperature
Vo = analogRead(thermistorpin);
 RT = R1*( 1023.0 / (float)Vo- 1.0 );
logRT = log(RT);
T=( 1.0 / (c1 + c2*logRT + c3*logRT*logRT*logRT ) )- 273.15;
//Steinhart-Hart equation, T in C
measurementThermistor[i] =  T;

//calculating pH
sensorValue = analogRead(signalpin);
pHValue = map(sensorValue, 0, 1023, 0, 14.00);
measurementEV[i] = sensorValue;
measurementpH[i] = pHValue;
}
buttonCounter = buttonCounter + 1;
}

//Calibrating slope
float slope_calib = (measurementEV[2] -
measurementEV[1])/(measurementpH[2] - measurementpH[1]);

//pH reading
float pHsample = measurementpH[2] + (measurementEV[3] -
measurementEV[2])/slope_calib;

}

if(pHsample < 5.5 || pHsample > 7) {
digitalWrite(2, HIGH)
}

if(pHsample > 5.5 && pHsample < 7) {
digitalWrite(4, HIGH)
}

if(pHsample < 0 && pHsample > 14) {
digitalWrite(3, HIGH)
```