**RasPi + Arduino EC and Temp Sensors Report**

Jonah Gonzales

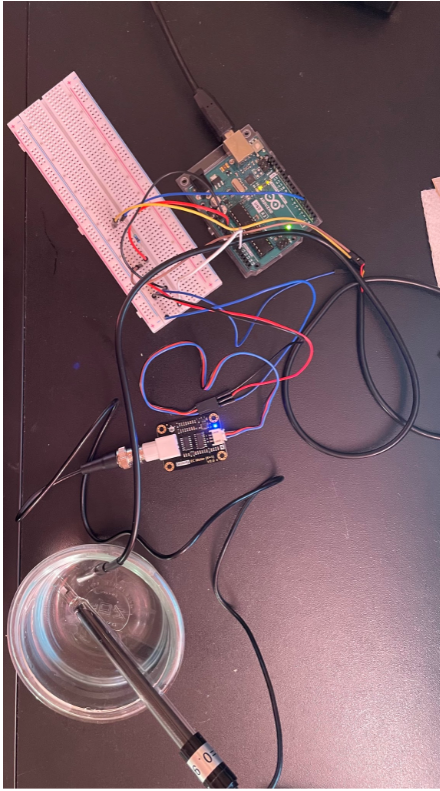
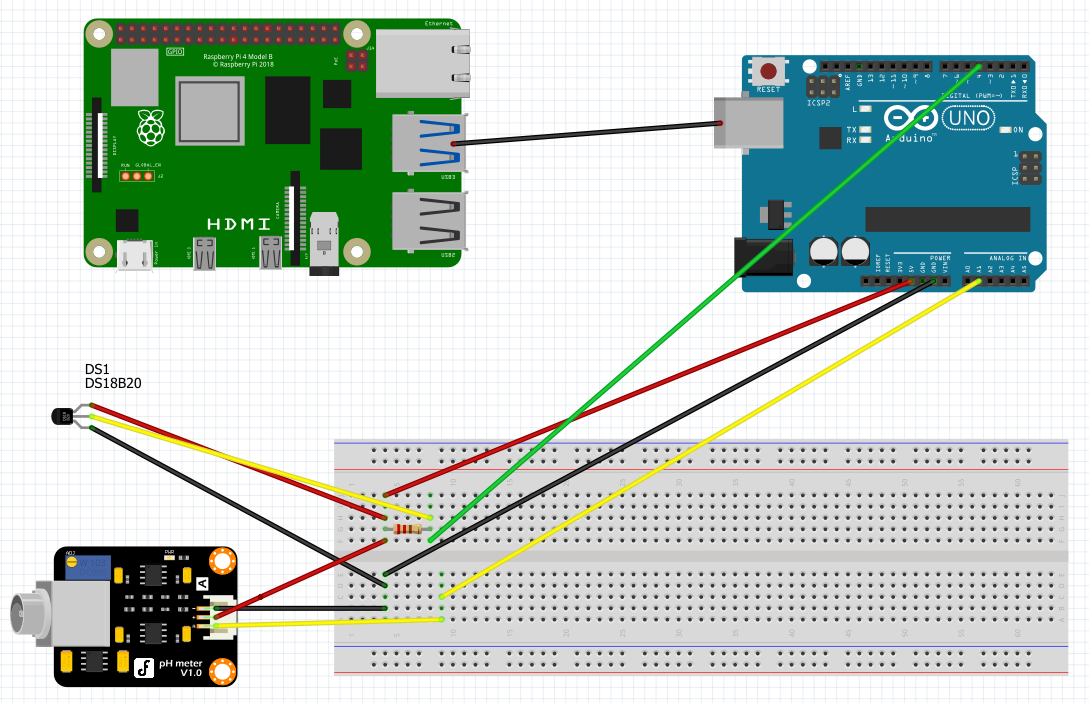
ANT (Automated Nutrient Technique)

November 8, 2021

**Opening:**

This module will allow us to process the data through the RasPi, which will in turn let us control the peristaltic pumps that will inject the system with the nutrients as needed to balance the system. It will be adjusted once we settle on a sampling rate for the measurements, and control of the peristaltic pumps will be added to the Python code. Currently, the Python code takes the average of the 5 most recent measurements. The average is taken to account for any fluctuations in the readings. We imagine we will take a sample over one minute, every 20 minutes, for example. Once the system is built and we get peristaltic pump control, we will adjust and set those times.

**Fritzing Schematic:**



**Figure 1. Left: Photo of built schematic. Right: Fritzing Diagram of RasPi + Arudino Temp/EC Sensor setup.**

**Code:**

**RasPi Python Code (Will be adjusted once we settle on sampling rate):**

**import serial**

**import re**

**import sys**

**import os**

**import string**

**import time**

**#open serial port**

**ser=serial.Serial('/dev/ttyACM0', 115200)**

**temp = [0,0,0,0,0]**

**EC = [0,0,0,0,0]**

**counter = 0**

**while True:**

**SensorData = str(ser.readline().decode("utf-8")).split(' ')**

**#print(SensorData)**

**temp[counter] = float(SensorData[0])**

**EC[counter] = float(SensorData[1])**

**TempCount = 0**

**ECCount = 0**

**for i in range(0,5):**

**TempCount = TempCount + temp[i]**

**ECCount = ECCount + EC[i]**

**TempAvg = TempCount/5**

**ECAvg = ECCount/5**

**print("Temp[]: ", temp, "EC[]: ", EC, "\n")**

**print("Temp: ", TempAvg, "EC: ", ECAvg, "\n")**

**counter = counter + 1**

**if counter > 4:**

**counter = 0**

**Arduino Code:**

#include <DS18B20.h>

#include "DFRobot\_EC.h"

#include <EEPROM.h>

#include <DallasTemperature.h>

#define EC\_PIN A1

#define Temp\_Sensor 4

float voltage,ecValue,temperature = 25;

DFRobot\_EC ec;

OneWire oneWire(Temp\_Sensor);

DallasTemperature sensors(&oneWire);

void setup() {

Serial.begin(115200);

ec.begin();

sensors.begin();

}

void loop()

{

static unsigned long timepoint = millis();

if(millis()-timepoint>1000U) //time interval: 1s

{

timepoint = millis();

voltage = analogRead(EC\_PIN)/1024.0\*5000; // read the voltage

//temperature = 25; //Temporary value until temp sensor added

temperature = readTemperature(); // read your temperature sensor to execute temperature compensation

ecValue = ec.readEC(voltage,temperature); // convert voltage to EC with temperature compensation

//Serial.print("temperature:");

Serial.print(temperature,1);

Serial.print(" ");

//Serial.print("^C EC:");

Serial.print(ecValue,2);

Serial.print("\n");

//Serial.println("ms/cm");

}

ec.calibration(voltage,temperature); // calibration process by Serail CMD

}

float readTemperature() {

//Serial.print(" Requesting temperatures...");

sensors.requestTemperatures(); // Send the command to get temperature readings

//Serial.println("DONE");

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//Serial.print("Temperature is: ");

temperature = (sensors.getTempCByIndex(0)); // Why "byIndex"?

// You can have more than one DS18B20 on the same bus.

// 0 refers to the first IC on the wire

delay(1000);

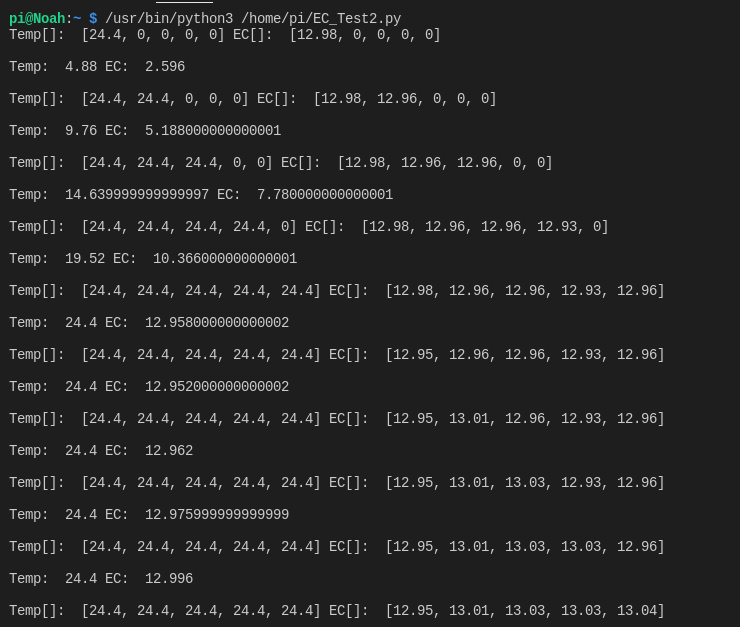
return temperature;

}

**Results Screenshots: (Calibrated as in previous EC Sensor Report)**

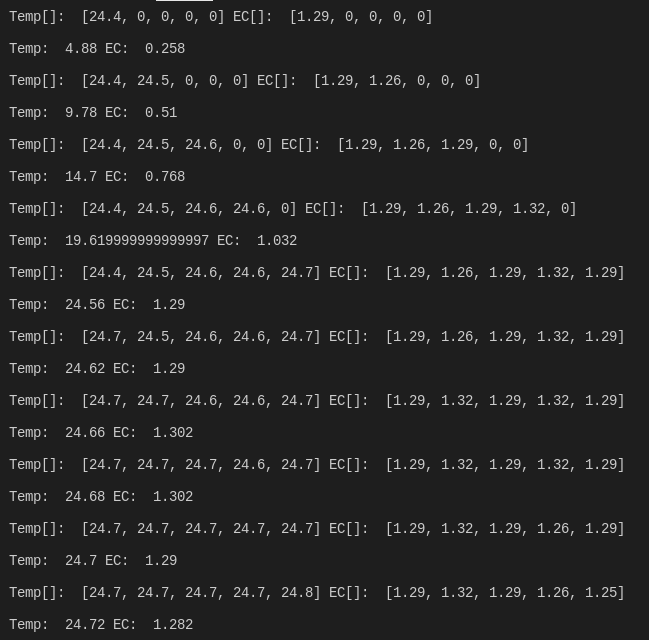
**Temperatures verified by food thermometer**

**12.88 ms/cm Solution**



**Figure 2: Terminal results of Python Code when testing the 12.88 ms/cm solution**

**1413 us/cm Solution:**



**Figure 3: Terminal results of 1413 us/cm solution test**

**References:**

<https://yantraas.com/send-sensor-data-from-arduino-to-raspberry-pi/#Setting_Up_Serial_Communication_On_Arduino>