Is hydroponics organic? Is it better or worse?

There has been a battle raging during the past decade between soil-based organic producers and hydroponic growers, to figure out whether hydroponically produced crops can or cannot be considered for organic certification. The entire discussion centers around whether a hydroponically grown crop can in fact comply with the requirements of the USDA organic standard. Within this post, we are going to discuss why there is even a discussion, why a hydroponic crop could be considered organic, and what the arguments against such a designation currently are.

USDA Organic food coming from traditional organic based growing practices

All that is required for a crop to be considered “hydroponic” is the complete absence of soil. This means that all the nutrition required for the crop is going to come from the nutrient solution and the substances that are put within this solution can or cannot comply with the USDA requirements for the “organic” label. Some substances like heavy metal chelates, potassium phosphates, and most nitrates, are forbidden by the USDA organic designation due to their synthetic origin, and the environmental impact of their production and normal usage. However, the total impact of these substances also rests heavily on how the hydroponic crop manages them and how efficiently they are used.

A hydroponic crop could use a fraction of the water and fertilizer used by a traditional soil crop of the same area while capturing all fertilizer effluents, making it environmentally more sustainable than a traditional soil crop and probably worthy of some sort of designation to recognize this fact. A hydroponic crop grown with traditionally produced

synthetic fertilizers, that has absolutely no fertilizer dumping of wastewater to the environment and uses no synthetic pesticides on products has a low environmental impact and produces food of very high quality. Hydroponic crops can also use land that would otherwise be unusable by traditional soil-based methods, expanding the area that could be used for healthy and sustainable food production.

However, the defendants of the organic designation argue that it is not only about what is being produced and how it is being produced but where it is being produced. The argument is that the organic designation and requirements have specific provisions about soil sustainability and soil building, that a hydroponic crop could not possibly comply with. They argue that part of the spirit of the organic designation is to make growing in soil more sustainable and that hydroponically grown crops simply cannot do this because they completely lack any soil or any soil building process.

Both hydroponic and traditionally designated organic crops can produce food that is healthy, pesticide-free, and sustainable. Hydroponic crops can do this on land that is not traditionally arable and can do so at astonishingly high efficiencies. Therefore, it would be fair to provide hydroponic crops that are evaluated to be sustainable and grown over non-arable land, an organic designation, since they comply with the spirit of what, I believe, the people who buy organic want, which is to have foods that are produced in a sustainable manner, with little impact on the environment. If the use of synthetic fertilizers is a concern, a requirement to meet this designation could also be the use of the same array of inputs available to traditional organic growers. This is harder to achieve, but still viable within the hydroponic production paradigm.

Some hydroponic farms can be very sustainable. Farms coupling hydroponics with fish production – known as aquaponics – can make use of no synthetic fertilizers at all.

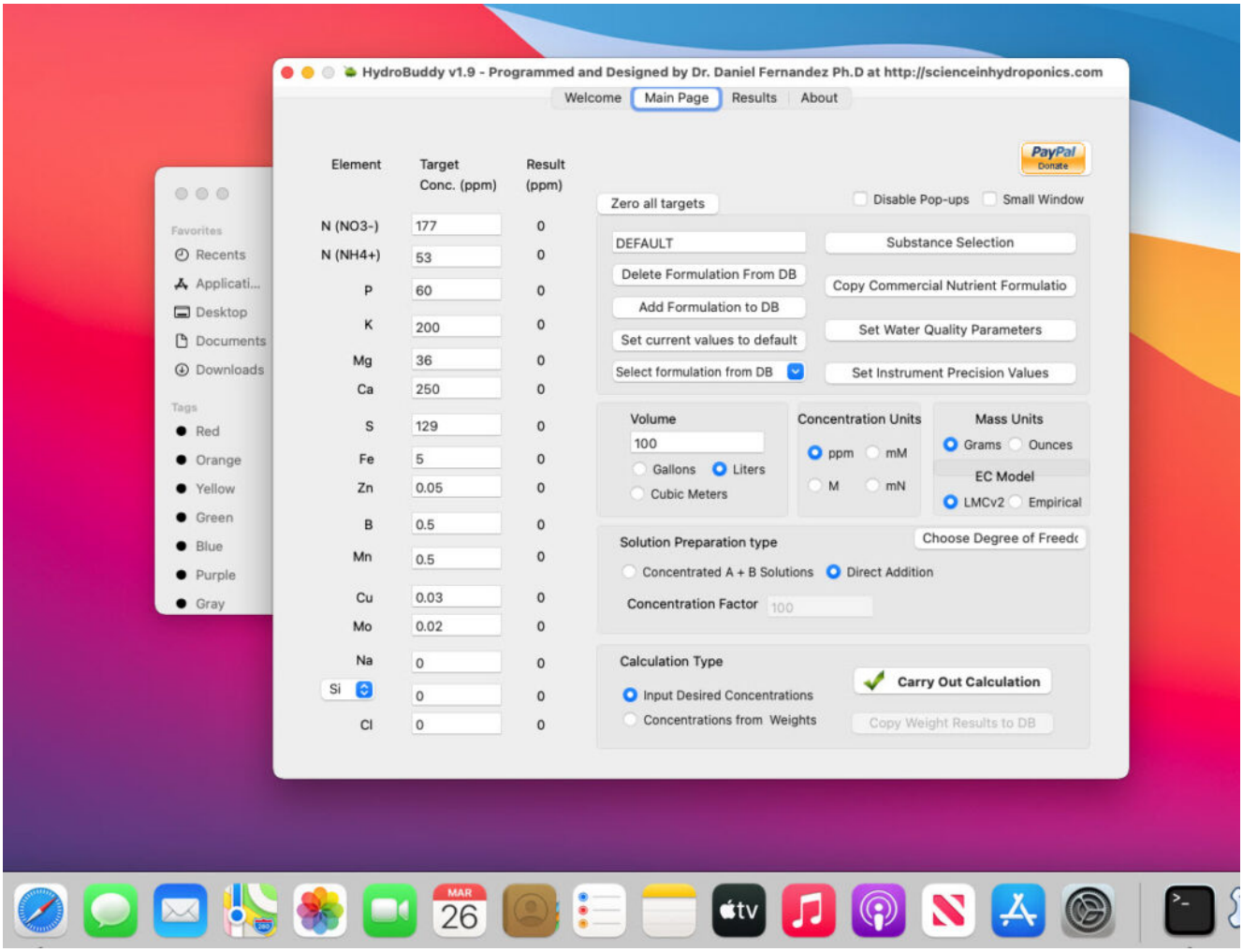
Recently, hydroponic growers have won battles in California about being granted an organic designation, however, because of the large amount of money that the organic designation carries – allowing growers to charge a big premium for items designated as organic – the organizations of soil-based organic growers are going to continue to fight this as much as they can. Organic grower organizations have even fought the potential for an independent "organic hydroponic" designation (see here), as they say, this might be confused with the normal "organic" designation and negatively affect their products.

Not all hydroponic crops are environmentally sound though. Many of them can be incredibly polluting and can make inefficient use of both water and fertilizer resources. For this reason, a designation is required to distinguish those that are sustainable from those that are not. If the USDA organic designation requirements are adjusted to accommodate for the potential for highly sustainable hydroponic crops grown on non-arable land, this would be a huge step in giving customers a clear way to tell products apart.

Hydroponic crops can be sustainable, have a low impact, and produce very high-quality, nutrient-rich food with no pesticides. They can make more efficient use of land, water, and non-synthetic fertilizers than soil-based crops do. However, the fact is that few of them really meet these criteria, because there is no designation they can achieve that would make this worthwhile from a commercial perspective. So while they are not organic at the moment, giving them the possibility to be organic would be a huge step towards a more sustainable future in agriculture. It could motivate hydroponic growers to become more sustainable and embrace a lot of the practices of soil-based organic growers.

HydroBuddy v1.9, MacOS binary, new EC model, many bug fixes and more!

Today I am releasing a new version of HydroBuddy (v1.9) which contains many suggested and needed improvements from the previous version of the software. In this post I want to discuss the changes within this release and how they will affect the way things are done in the program. Some big changes have been implemented so make sure you go through the list below if you want to use this new version. **Thanks to all of you who contributed your suggestions about HydroBuddy and/or reported bugs to me.**



One of the biggest changes in this release, the return of precompiled MacOS binaries.

Here is the list of changes in this version:

- A MacOS binary compiled in Big Sur 11.0.1 has been released.
- Ability to make any formulation the “default” formulation. This selected formulation is loaded when the software is started.
- The LMC conductivity model has now been replaced with LMCv2 which is an important improvement. See here to learn more. The LMCv2 model now adjusts conductivity based on each specific ion’s charge and the overall ionic strength of the solution. It now includes no arbitrary terms.
- The treatment of liquids/solids in the program has now been changed. Instead of specifying liquid or solid (and the program having to make assumptions) users can now

select whether the percentages and substance amounts are going to be either in g and w/w% or in mL and w/v%. This should simplify the interpretation of results and the addition of substances.

- An additional column has now been added in the results page to specify the unit of the amount being calculated. When a user wants a substance's contribution to be calculated in mL, the appropriate unit will be shown here.
- When adding a new substance, all fields are reset to null values (previously the program kept the values from previously opened/updated substances).
- Density has now been eliminated as a variable used in the program since it is not needed if there is no cross between w/w% and w/v% calculations. It is only kept in the "Copy commercial nutrient formulation" dialogue.
- An error where P and K were mixed up in the product comparison window of the "Copy commercial nutrient formulation" function has now been fixed.
- The wording of options in the "Substance selection" dialogue has been changed so that the buttons better describe what they do. For example the "Delete" button has now been changed to "Do not use".
- Two buttons have been added next to the EC model prediction in order to allow users to increase or decrease the EC by adjusting all nutrient concentrations by +5%/-5%. This will allow you to see how nutrient concentration changes affect conductivity in a straightforward manner.

The above modifications are now committed to the github repository as well. Feel free to take a look if you're interested in how any of the above variations were coded into the program.

Improving on HydroBuddy's theoretical conductivity model, the LMCv2

Hydrobuddy's theoretical conductivity estimates have never been good. As I discussed in a previous post, the program uses a very simple model based on limiting molar conductivities to calculate the EC. The software knows how much each ion conducts when it's all by itself, so it adds all these conductivity values multiplied by the concentration and assumes there are no additional effects. The conductivity values resulting from this assumption are very large – because there are effects that significantly reduce the conductivity of ions at larger concentrations – so HydroBuddy just cuts the estimation by 35% hoping to reach more accurate values. This works great for some cases, but very badly for others.

The reason why this happens is that the actual conductivity contribution of some ions decreases more drastically as a function of concentration and due to the presence of other ions compared to others. This means that we need to account for these decreases in conductivity in an ion-specific way. One way to approach this, is to forget about theoretical approximations and just create an empirical model that uses experimental data. This is what I did when I created the empirical model that is present in HydroBuddy from v1.7. This model works really well, provided you are using the exact list of salts that were used to create the model and you stay within the boundaries of concentration values that were used to create it.

1) $$EC = \sum_i \Lambda^0_{m,i} (\gamma_i)^\alpha c_i$$

2) $$\alpha = \begin{cases} 0.6 / |z_i|^{0.5} = const & \text{if } I \leq 0.36 |z_i| \\ \sqrt{I} / |z_i| & \text{otherwise} \end{cases}$$

3) $$\ln \gamma_i = -(\ln 10) A z_i^2 \sqrt{I}$$

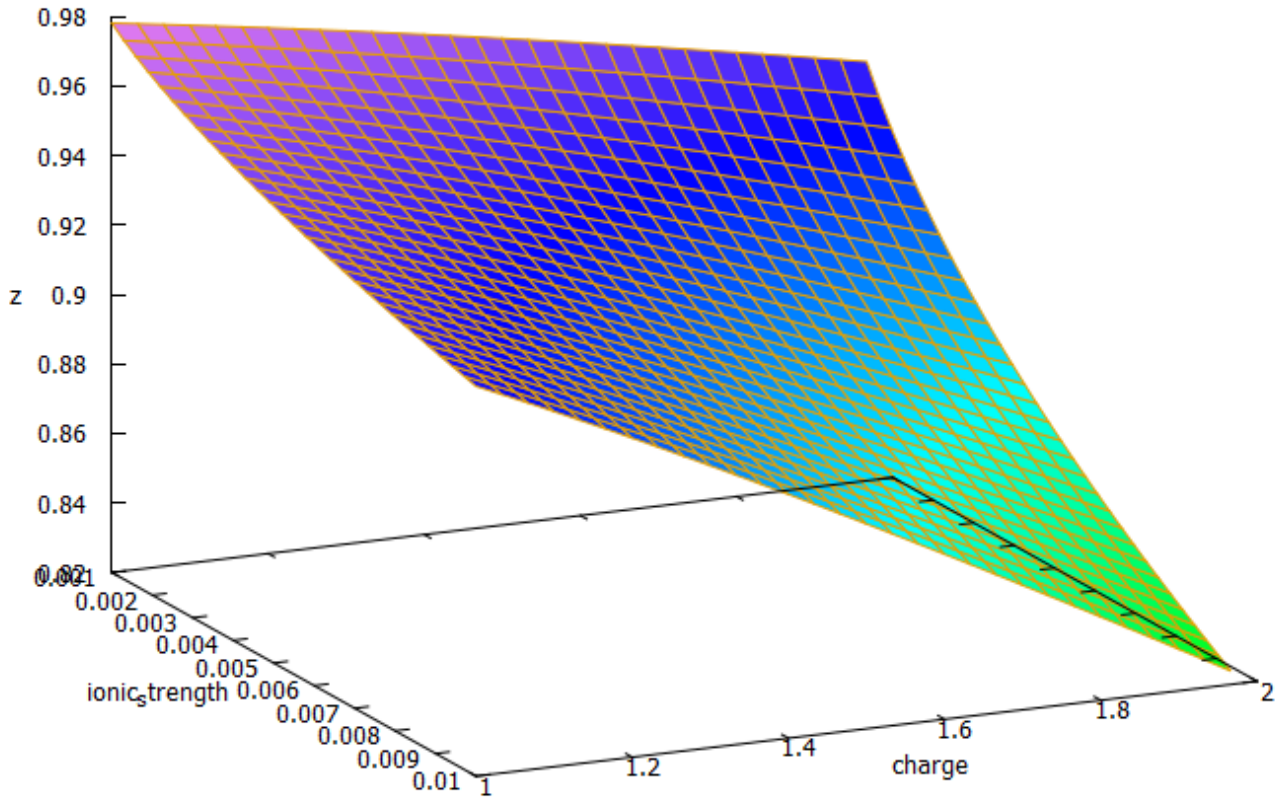
4) $$EC = \sum_i \Lambda^0_{m,i} \left(e^{-0.7025148 z_i{}^{1.5} \sqrt{I}} \right) c_i$$

Equations 1-3 were taken from here. I have then used these equations to derive equation 4, which is going to be the new LMCv2 model for HydroBuddy from v1.9. Where $\Lambda^0_{m,i}$ is the limiting molar conductivity of each ion, z_i is each ion's charge, I is the ionic strength of the solution and c_i is the molar concentration of each ion..

This experiment-based solution can be great. It is in fact, a technique I've used to create custom versions of HydroBuddy for clients who want to have high accuracy in their EC estimations within the salts that they specifically use. The process is however cumbersome and expensive, my wife and I – both of us chemists – do all the experimentation, and it generally requires an entire day, preparing more than 80+ solutions using high accuracy volumetric material, to get all the experimental data. It is also limited in scope, as any salt change usually requires the preparation of a substantial number of additional solutions to take it into consideration.

It would certainly be great if we could create a better, fully theoretical, conductivity model. Diving into the literature and programs used for conductivity-related calculations, I found a program called Aqion that implements a more accurate model compared with HydroBuddy's LMC model. You can read more about their approach here. They use the limiting molar conductivities but introduce additional terms to make ion-specific corrections that are related to both ionic charge and ionic strength. The ionic charge is the electrical charge of

each ion, for example, +1 for K^{+} and +2 for Fe^{+2}, etc. The ionic strength is the sum of the molar concentration of each ion times its charge.



3D plot of equation 4 showing the magnitude of the correction factor (z) as a function of charge and ionic strength.

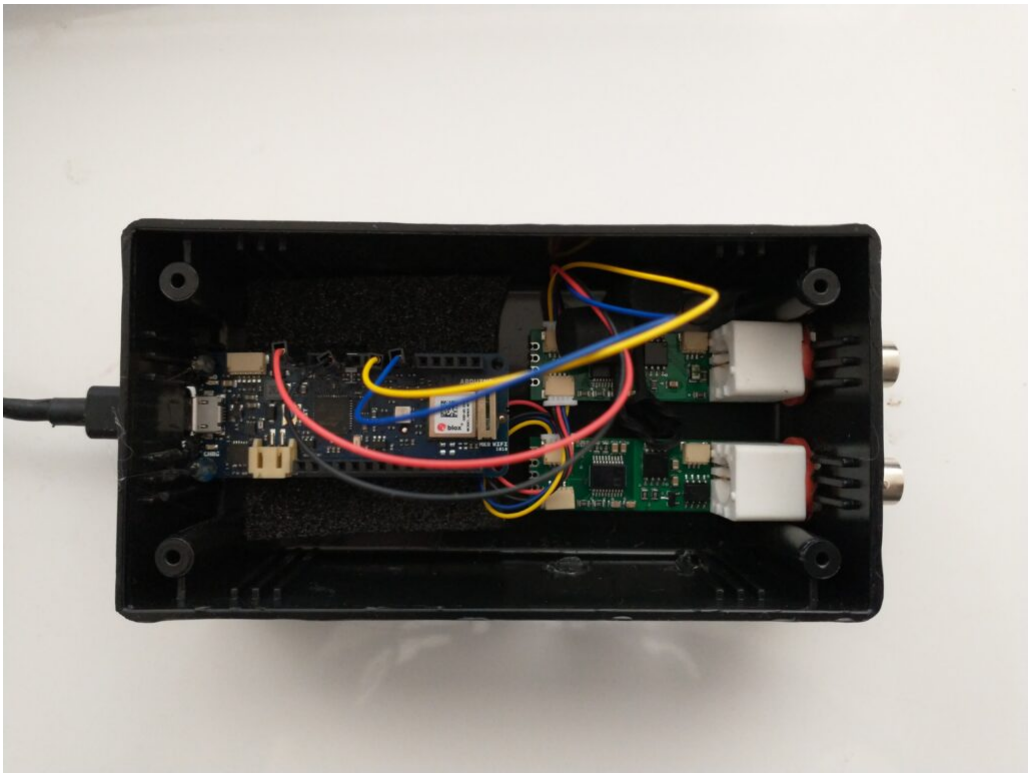
The plot above shows you how this correction factor affects a solution as the ionic strength and charge of the ions change. As a solution gets more diluted, the equation approaches the sum of the conductivities at infinite dilution. Conversely, as the solution becomes more concentrated or the ion charge becomes higher, the drop in the conductivity becomes more pronounced. These are both phenomena that are in-line with experimental observations and much better reflect how conductivity is supposed to change when different ions interact in solution.

The above equation provides us with a more satisfactory theoretical estimation of conductivity compared to the current HydroBuddy LMC model. The new model is able to implement

correction factors on a per-ion basis and also changes the magnitude of these corrections depending on how concentrated the solutions are. This new model will be implemented to replace the current LMC model in HydroBuddy v1.9, which will be released in the near future. This should provide significantly more accurate estimates of conductivity for the preparation of hydroponic solutions.

Creating a pH/EC wireless sensing station for MyCodo using an Arduino MKR Wifi 1010

There are multiple open-source projects available online for the creation of pH/EC sensing stations for hydroponics. However, all of the ones I have found use a single Arduino or Raspberry Pi to perform the measurements and store any data, making them unsuitable for applications where more flexibility is needed. For example, a facility using multiple different reservoir tanks for nutrient storage might require multiple pH/EC sensing stations, and single-board wired setups would be unable to accommodate this without a lot of additional development. In this post, I am going to show you a simple pH/EC sensing station I built with an Arduino MKR Wifi 1010 that can communicate with a MyCodo server using the MQTT protocol. Multiple sensing stations could be built and all of them can communicate with the same MyCodo server.



My Arduino MKR wifi 1010 based sensing station, using uFire pH and EC boards in a small project box.

This project makes use of the small pH/EC boards provided by uFire, which have a lower cost compared to those provided by companies like Atlas, but do have adequate electrical isolation to avoid problems in readings when multiple electrodes are put in the same solution. This is a substantial improvement over other low-cost boards where using multiple probes can cause heavy electrical noise and interference. In order to build this project you will require the following materials:

Note, some of the links below are amazon affiliate links. This means that I get a small commission if you purchase through these links at absolutely no extra cost to you. The links to other websites are not affiliate links.

1. Arduino MKR Wifi 1010
2. uFire pH probe
3. uFire EC probe
4. A rugged pH probe with a VNC connector
5. An rugged EC probe with a VNC connector
6. Two Qwiic-to-Qwiic connectors

7. One Qwiic-to-male connector
8. A project box to put everything inside (optional)
9. A micro USB cable

The code for the project is shown below:

```
#include <uFire_EC.h>
#include <uFire_pH.h>
#include <WiFiNINA.h>
#include <ArduinoMqttClient.h>

#define SECRET_SSID "ENTER WIFI SSID HERE"
#define SECRET_PASS "ENTER WIFI PASSWORD HERE"

//calibration solutions used
#define PH_HIGH_SOLUTION_PH 7.0
#define PH_LOW_SOLUTION_PH  4.0
#define EC_HIGH_SOLUTION_EC 10.0
#define EC_LOW_SOLUTION_EC  1.0
#define CALIBRATION_TEMP    20.0

// topics for the mqtt sensors
// Make sure all stations have different topics
#define EC_TOPIC       "EC1"
#define PH_TOPIC       "PH1"
#define CALIB_TOPIC    "CALIB1"
#define MQTT_BROKER    "ENTER MQTT SERVER IP HERE"
#define MQTT_PORT      1883

int status = WL_IDLE_STATUS;     // the Wifi radio's status
String message;

uFire_pH ph;
uFire_EC ec;
WiFiClient wifiClient;
MqttClient mqttClient(wifiClient);

void check_connection()
{
  if (!mqttClient.connected()) {
    WiFi.end();
    status = WiFi.begin(SECRET_SSID, SECRET_PASS);
```

```
    delay(10000);
    if (!mqttClient.connect(MQTT_BROKER, MQTT_PORT)) {
      Serial.print("MQTT connection failed! Error code = ");
      Serial.println(mqttClient.connectError());
      delay(100);
    }
    mqttClient.subscribe(CALIB_TOPIC);
  }
}

void setup()
{
  Serial.begin(9600);
  while (!Serial);

  // connect to wifi and mqtt broker
  check_connection();
  // coorectly initialize the uFire sensors
  // note the Wire.begin() statement is critical
  Wire.begin();
  ec.begin();
  ph.begin();
}

void loop()
{
  // mqtt keep alive
  mqttClient.poll();

  // read messages
  message = "";
  while (mqttClient.available()) {
      message += (char)mqttClient.read();
    }

  // execute calibration if requested
  Serial.println(message);
              if       (message       ==       "EC1_HIGH")
ec.calibrateProbeHigh(EC_HIGH_SOLUTION_EC, CALIBRATION_TEMP);
               if       (message       ==       "EC1_LOW")
ec.calibrateProbeLow(EC_LOW_SOLUTION_EC, CALIBRATION_TEMP);
```

```
            if      (message      ==      "PH1_HIGH")
ph.calibrateProbeHigh(PH_HIGH_SOLUTION_PH);
            if      (message      ==      "PH1_LOW")
ph.calibrateProbeLow(PH_LOW_SOLUTION_PH);

  // Measure EC
  ec.measureEC();
  Serial.println((String) "mS/cm: " + ec.mS);

  // Measure pH
  ph.measurepH();
  Serial.println((String) "pH: " + ph.pH);

  // Ensure the wifi and mqtt connections are alive
  check_connection();

  // post EC to MQTT server
  mqttClient.beginMessage(EC_TOPIC);
  mqttClient.print(ec.mS);
  mqttClient.endMessage();

  // post pH to MQTT server
  mqttClient.beginMessage(PH_TOPIC);
  mqttClient.print(ph.pH);
  mqttClient.endMessage();

  // ensure sensors are not probed too frequently
  delay(1000);

}
```

Once you get all the materials you should first assemble the components. Connect the pH and EC board together using the Qwiic-to-Qwiic connector, then use the Qwiic-to-male connector to hook up one of these boards to the Arduino (doesn’t matter which one). Connect the black cable to ground, red cable to 5V, blue cable to SDA, and yellow cable to SCL. Set up your board according to the instructions in the Arduino MKR wifi 1010 getting started page, modify the code above to properly include information about your wifi network, calibration solutions, and MQTT server, then upload the code. The Arduino

will connect to your Wifi and MQTT servers and automatically reconnect when there are connection issues.

The above code will also post the readings of the pH and EC sensors to topics PH1 and EC1 respectively if you add an input in MyCodo to capture these readings you should be able to store them and take control actions using the MyCodo interface. Additionally, the Arduino code will respond to calibration requests published to the topic “CALIB1”. For example, if you want to calibrate your EC sensor with a two-point calibration method with a standard solution with an EC of 10mS/cm, you would put the electrode in the calibration solution, then send the message “EC1_HIGH” to the CALIB1 topic and the Arduino will perform the task as requested. The code assumes you will want to do 2 point calibrations for both EC and pH, with the calibration events triggered by EC1_HIGH, EC1_LOW, PH1_HIGH, and PH1_LOW. Note that the definition of the EC and pH values of the calibration solutions should be changed to the solutions you will be using within the code. The high/low values in the code, as is, are 10mS/cm|1mS/cm for EC and 7|4 for pH.

A simple cheatsheet for macro nutrient additions in hydroponics

In hydroponic growing, we are often faced with the need to adjust the nutrient concentrations of a fertilizer reservoir or foliar spray directly, in order to increase the quantity of some nutrient by a specific amount. Although you can use a program like HydroBuddy in order to quickly calculate these

values, it is often the case that these calculations need to be done in the field or in a growing environment, and a computer to calculate things is not at hand. For this reason, I have created a small “cheat sheet” that you can use in order to figure out the amounts of salts that you would need to add to a solution to increase any of the macronutrients by 10 ppm.

Salt Name	**ppm**	**Element**	**ppm**	**Element**	g/L	**g/gal**
Calcium nitrate (ag grade)	10	N (NO3-)	13.19	Ca	0.0694	0.2629
MAP	10	N (NH4+)	22.1	P	0.0821	0.3108
Ammonium Sulfate	10	N (NH4+)	11.4	S	0.0472	0.1785
Gypsum	10	Ca	7.99	S	0.0430	0.1626
Calcium Chloride	10	Ca	17.69	Cl	0.0277	0.1048
Magnesium Nitrate Hexahydrate	10	N (NO3-)	8.67	Mg	0.0915	0.3463
Epsom Salt	10	Mg	13.19	S	0.1014	0.3839
Magnesium Chloride	10	Mg	29.16	Cl	0.0392	0.1483
AgSil 16H	10	Si	10.9	K	0.0411	0.1554
MKP	10	P	12.62	K	0.0439	0.1663
Potassium Nitrate	10	N (NO3-)	27.87	K	0.0730	0.2763
Potassium Sulfate	10	K	4.10	S	0.0223	0.0844
Potassium Chloride	10	K	9.067	Cl	0.0191	0.0722

Cheatsheet for macronutrient additions in hydroponics

With the above cheatsheet, you can quickly evaluate some of the most common options you would have to increase all the different macronutrients in a hydroponic or foliar solution by 10 ppm and which secondary elemental contributions you would get from these additions. For example, if you add 0.0694g/L of Calcium Nitrate, this would add 10ppm of Nitrogen as nitrate plus 13.19ppm of Calcium. Careful consideration of secondary contributions need to be taken into account, especially when using salts that contain elements that can be toxic, such as

chlorides.

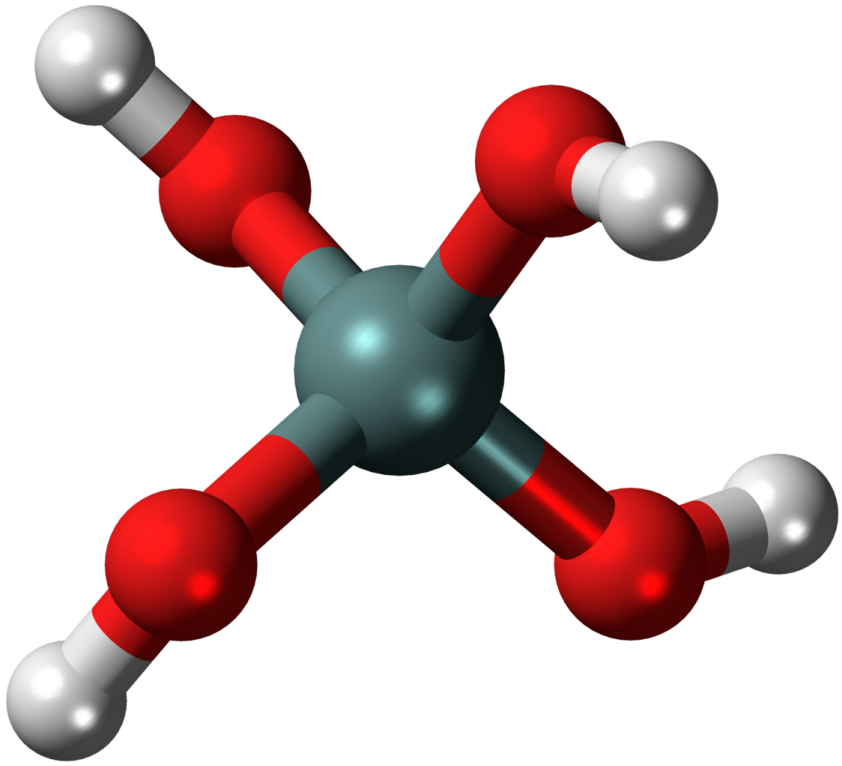
How to make your own stabilized mono-silicic acid for use in hydroponics

Please follow this link, for an updated an easier process for the synthesis of mono/ortho-silicic acid.

During the past several years, there have been a lot of “mono-silicic acid” products being marketed for their use in hydroponic culture. These differ from the traditional potassium silicate based products in that they are very acidic in their concentrated form and are stable in solution for longer periods of time at the pH values used in hydroponics. While a hydroponic nutrient solution that has potassium silicate added to it and the pH adjusted to 5.5-6.5 will generally see extensive polymerization of the silicon-containing molecules within 24 hours, these stabilized mono-silicic acid products will be stable for far longer periods of time. They are therefore ideal for use in recirculating systems, where potassium silicate additions can be less effective.

If you watched my youtube video on silicon in hydroponics, you’ll know that the most common way to produce these stabilized products is quite complicated and involves the use of silicon chloride and very specific reaction conditions. These are unavailable to hydroponic growers, reason why it is not easy to produce a home-made version of choline stabilized ortho-silicic acid (ch-OSA). However, choline is not the only reagent that can be used to stabilize silicic acid in

solution, and research in industry has shown us that it is actually possible to form stabilized silicic acid products starting from potassium silicate itself.



Model representation of orthosilicic acid, also called mono-silicic acid.

This patent describes how to prepare mono-silicic acid solutions that are stabilized by carnitine and several other additives, in the region from 0.01 to 8% silicic acid by weight. The great thing about this process is that we can start from potassium silicate, which is readily available. The concentration of Si obtained will be significantly lower than what is possible when generating ch-OSA – where solutions can reach 40% mono-silicic acid – but the fact that we can prepare it from readily available materials might compensate for this to some extent. **It is also worth noting that this process comes from an unexpired patent, so it should not be used commercially without licensing the technology from the owner of the intellectual property**.

Extrapolating from the contents of the patent and the examples given, we can come up with a process to brew our own mono-

silicic acid at an 8% concentration. Here are the things you will need:

Note the amazon links below are affiliate links. This means that, if you choose to purchase through these links, I get a commission for your purchase, at no extra cost to you.

1. Potassium silicate (at least 32% K as K_2O)
2. Carnitine hydrochloride
3. Phosphoric acid (85%)
4. Propylene glycol
5. Distilled water
6. Scale to weight the materials (precision of at least +/- 0.1g, max at least 500g)

To prepare around 425g of stabilized mono-silicic acid, you could follow this process.

Before continuing please make sure you understand what you're doing. Wear adequate eye and body protection, carry this out in a place with enough ventilation and make sure you read the material safety data sheet (MSDS) of all the materials used. These instructions are provided for educational purposes only, follow them at *your own risk.*

1. Add 10g of carnitine hydrochloride to a clean 1000mL beaker
2. Add 65g of distilled water to the mix.
3. Stir until all the carnitine hydrochloride dissolves
4. Add 10g of propylene glycol.
5. Add 240g of 85% phosphoric acid.
6. Put the mixture on an ice bath with ample ice.
7. Wait for 15 minutes, so that the mixture cools down.
8. During the course of an hour, s*lowly* add 125g of potassium silicate to the mixture with constant stirring. Add more ice to the ice bath if needed to keep the solution cool. Note that predissolving the silicate in 150mL of distilled water and adding it as a liquid

makes this process easier, although KOH additions might be required to complete its dissolution.

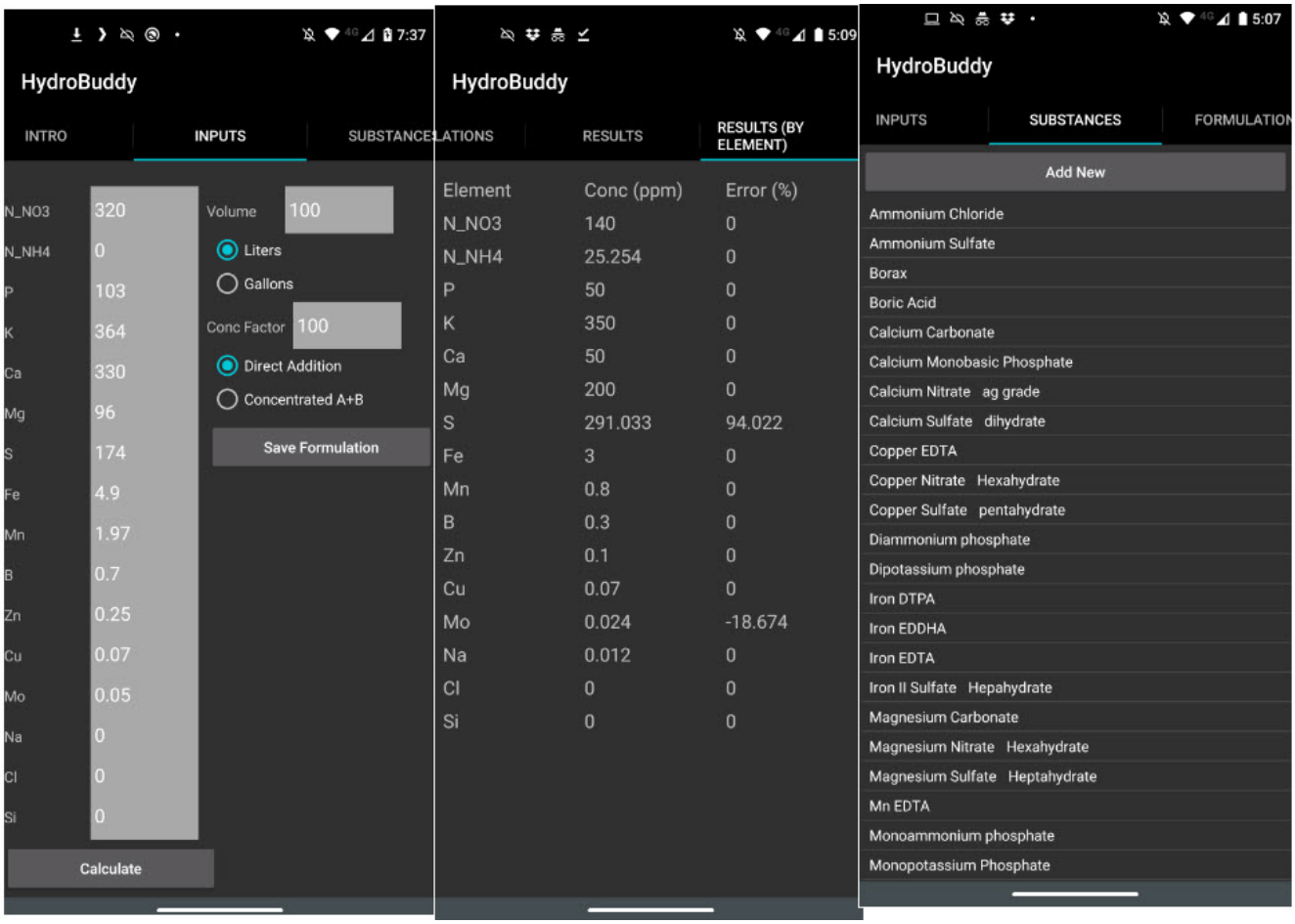
After this, you should be left with an acidic, completely translucid, carnitine and propylene glycol stabilized mono-silicic acid solution that should be around 7-8% w/w of Si as elemental Si. If there's any precipitate in the solution then the stabilization process did not go well and the silicic acid formed polymerized into silica. **This solution should then be used at around 1g/gal, which will provide ~18-20ppm of Si as elemental Si in your hydroponic solution**. When using this solution,. the silicon present at the pH used in hydroponics should be much more stable than when derived from direct addition of potassium silicate.

If you go through the above process, leave a comment and let us know how it went.

HydroBuddy coming to Android, free and open source!

The Hydrobuddy open source hydroponic nutrient calculator – which has been used extensively by both professional and amateur growers for the past 11 years – is finally making the leap to the Android platform. This is a big leap, as many growers – especially in developing countries – lack access to a PC but have easy access to Android phones. Thanks to the effort of the LAMW development team – who made the development of Android apps using Lazarus possible – I have been able to recode and port HydroBuddy to mobile and will release it for free and with NO ads within the next couple of weeks within the Google Playstore. The source code is already available on github and will be updated as the application development

process continues.



Some screenshots of the current testing release of the HydroBuddy Android app.

The HydroBuddy Android application will include most of the features present in the desktop application. It will include all the same substances and formulations that are included in the desktop application and will allow you to perform the same types of calculations you do on the regular application. It is however true that some functionality will be missing initially. The “Copy Commercial Nutrient Formulation” section has not been implemented yet, neither have the “Water Quality Parameters”, EC estimations, or calculation of instrumental errors. The “Concentration from weights” and “set weight” calculations are also missing at this point in time.

Right now, the HydroBuddy application is in its internal testing stage, which means that only people who I add to the testing group are able to download and use the application. **If**

you want to participate in this testing phase, you can download HydroBuddy in your Android device through this link. If you participate in this process, please share meaningful feedback about the application with me. *This feedback can be left as a comment in this post as well*.

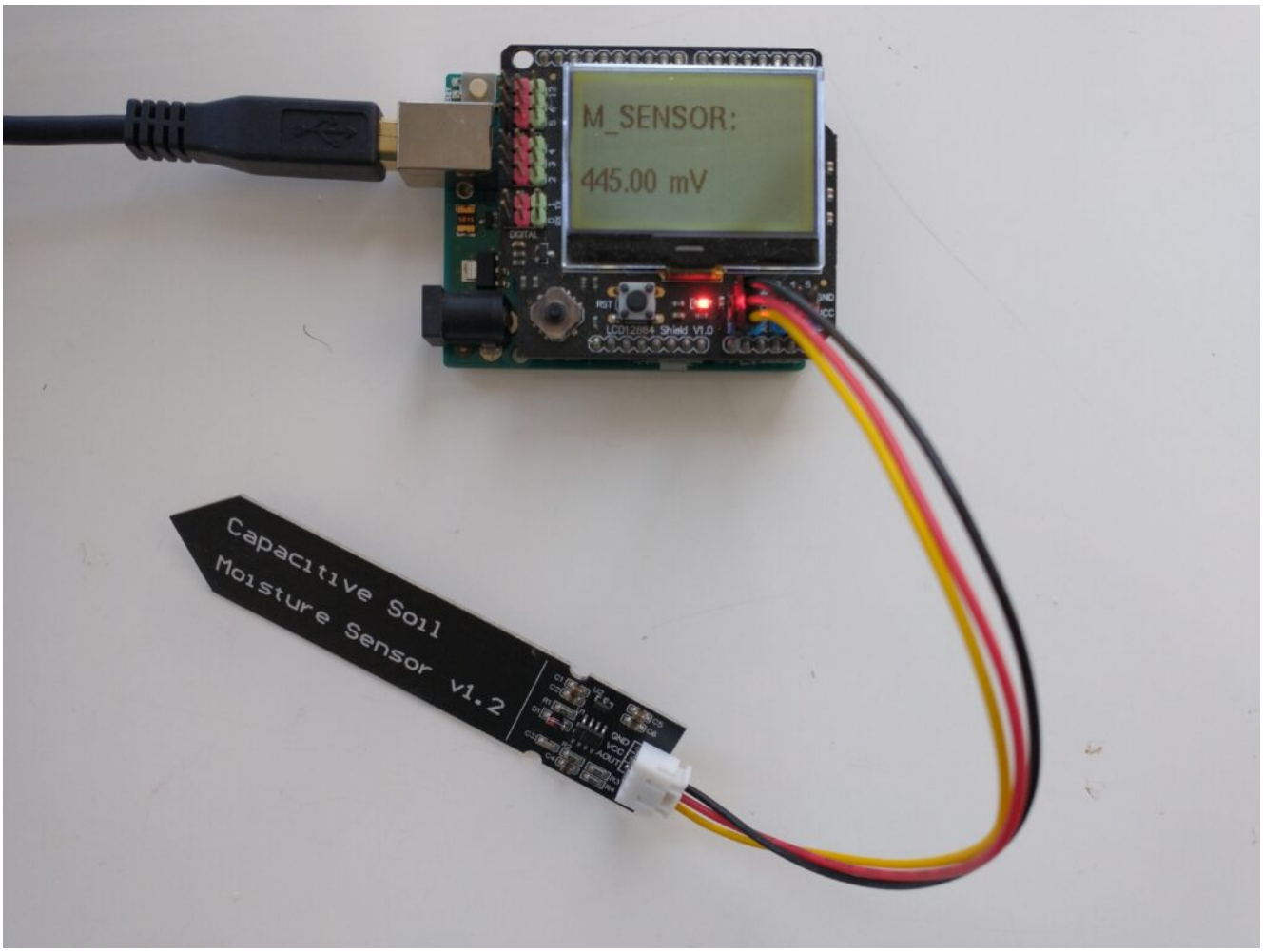
The port to Android is expected to be a stepping stone towards HydroBuddy v2.0, which will include a complete recoding of several portions of the calculator. This will be done in order to implement a more flexible database structure that allows for more effective saving and sharing of the inputs and outputs of the program. The aim of this is to allow growers of all origins: hobbyists, industry professionals, or researchers, to exchange complete calculation sets and allow for a much more profound use of the calculator as a community-building tool.

Sadly there is no iOS version planned, since I do not own, or plan to own, any Apple devices, and therefore cannot program/compile/test code in this platform. However, the licensing terms of both the desktop and mobile versions of the application do allow anyone to port and publish the application in iOS, provided it is non-commercial and released for free, with no ads and under the same licensing terms (GPLv2).

Calibrating a capacitive moisture/water content sensor for hydroponics

As I've mentioned multiple times in my blog, moisture sensing is one of the most important measurements in a hydroponic crop

that uses a significant amount of media. It allows you to properly time irrigations and avoid over or under watering your plants. Capacitive sensors are the lowest cost initial approach to adequate moisture monitoring of your media. In this post, we are going to learn how to use an Arduino with a gravity shield and a low-cost capacitive moisture sensor in order to accurately monitor the water saturation of our media.



A capacitive sensor plugged into an LCD shield and an Arduino UNO. The measurements are very easy to track with this setup.

An analogue capacitive moisture sensor like this one does not expose any metallic parts to the media and can be used for the monitoring of moisture content. This sensor is powered by 3.5-5V and gives you a voltage signal that is proportional to the dielectric constant of the media it is in. As the dielectric constant of media changes with the presence of water, so does the signal of the sensor. However, no specific voltage corresponds to a specific water content measurement by

definition, so we need to calibrate the sensor in order to interpret the voltage values we read from it. In order to get readings from this sensor, I use an Arduino UNO, the above-linked capacitive sensor, and an LCD shield from dfrobot that I can use to easily get the device readings. The arduino code used for this device is also shared below.

```
#include <U8glib.h>

#define XCOL_SET 0
#define XCOL_SET_UNITS 50

U8GLIB_NHD_C12864 u8g(13, 11, 10, 9, 8);
float capacitance;

void draw() {

  u8g.setFont(u8g_font_helvB10);
  u8g.drawStr(0,21,"M_SENSOR:");
  u8g.setPrintPos(XCOL_SET,51);
  u8g.print(capacitance);
  u8g.drawStr( XCOL_SET_UNITS,51,"mV" );
}

void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(4, OUTPUT);
  Serial.begin(115200);
  analogReference(DEFAULT);
  Serial.println("MOISTURE");

  u8g.setContrast(0);
  u8g.setRot180();
}

void loop() {

  draw();
  capacitance = analogRead(1);
  Serial.println(capacitance);
```

```
  u8g.firstPage();
    do  {
      draw();
    }
      while( u8g.nextPage() );

  delay(1000);
}
```

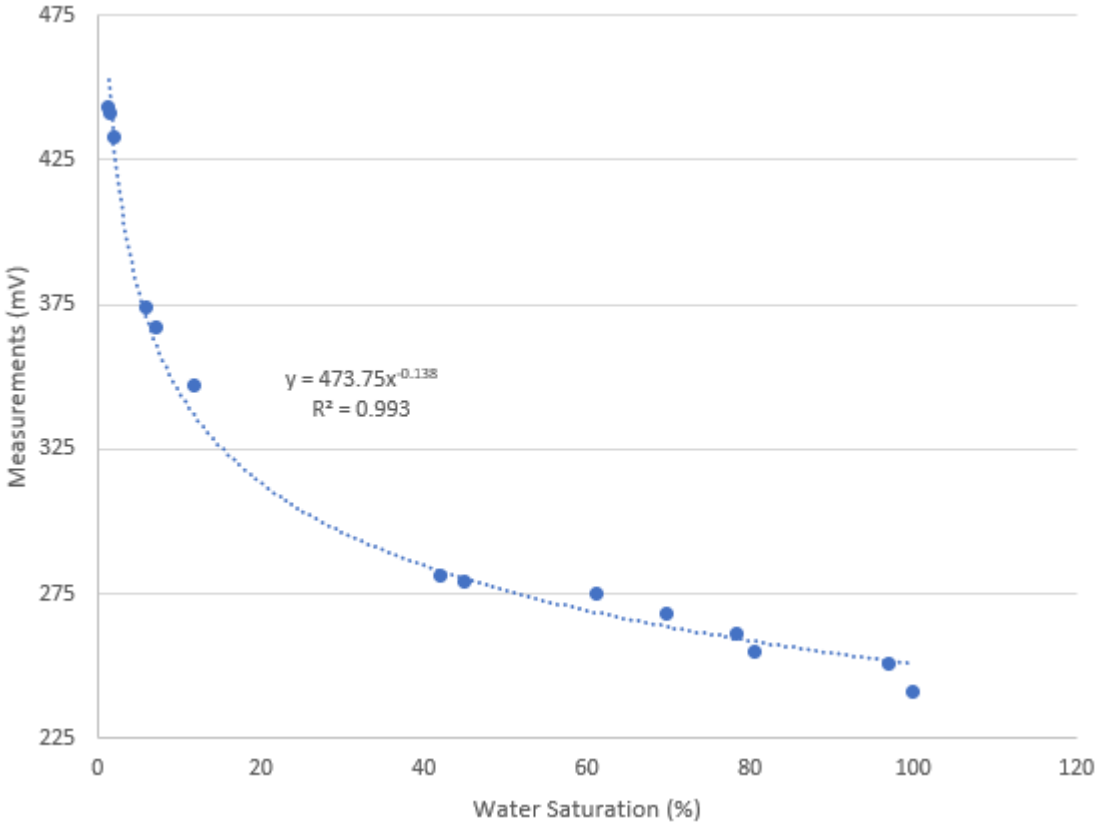
The calibration of a moisture sensor usually requires the creation of what soil scientists call a “water retention” curve, which is a curve where you plot the sensor’s signal as a function of a fixed volume or weight of water added to the media. However, this approach involves the use of many different containers and the addition of water to the media followed by oven drying steps in order to accurately determine how much moisture was actually in the media for every measurement carried out.

In order to do this procedure in an easier manner, losing as little accuracy as possible, I have created a calibration procedure that makes use of natural drying and only requires one single oven drying step. The procedure is as follows:

1. Heat the media that will be used for an experiment in an oven at 110°C (this drives out all water).
2. Wait for the media to cool to room temperature.
3. Fill the container that will be used for the test (this can plastic but has to have holes at the bottom). Put the sensor in the media, make sure the sensor is driven into the media until the top line is reached.
4. Take a reading, this is the “completely dry” media reading.
5. Disconnect the sensor from the Arduino.
6. Weigh the container+media+sensor. This will be the “dry weight”.
7. Take the sensor out.
8. Add water to the media until there is ample runoff coming out of the bottom.

9. Wait until no more runoff comes off.
10. Put the sensor in the media, making sure you drive it in until the top line is reached. The sensor won’t be taken out for the remainder of the experiment.
11. Connect the sensor and take a reading. Take note of this value.
12. Disconnect the sensor.
13. Weigh the container+media+sensor. Take note of this value. The first reading you take is the “max saturation” weight.
14. Repeat steps 11 to 13 every 1-6 hours (time is not very important as long as you gather enough data points) until you reach close to the dry weight. This can take several days depending on the media used. The more points you measure, the more accurate your curve will be.

After carrying out this procedure, you can create a curve like the one shown below. You can use the difference between each weight and the dry weight divided by the difference between the weight at max saturation and the dry weight in order to calculate the water saturation percentage. As you can see, the curves for these sensors are fairly linear in the 40-100% moisture range for the media I tested, while below 40% the regime changes and the measurement increases exponentially until it reached the “dry weight” sensor value. The entire curve can be described with a power-law equation. This behavior will not be the same for all media tested, reason why it’s very important for you to calibrate the sensor for the specific media you want to use.



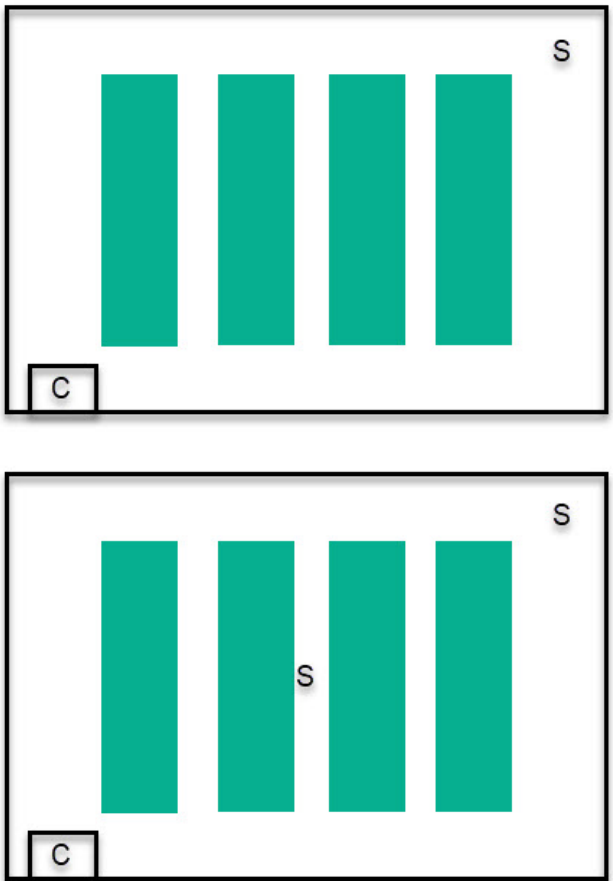
Calibration curve for a capacitive moisture sensor. In this case, the media was a mixture of 50% river sand and 50% rice husks.

Once you have your sensor calibrated you can know what a measurement in mV implies in terms of water saturation for a given media type. This allows you to time your irrigations at a given water saturation % effectively. Which water saturation percentage might be better, depends on the properties of your media and how the water potential changes as a function of the water saturation. However, this allows you to experiment with irrigations at different water saturation % values and figure out exactly where you need to water so that your plants are not under or overwatered.

It is also worth noting that the above sensor is probably not rugged enough for use in a hydroponic setup without a bit more hardening. In order to use these sensors in practical applications, you should apply conformal coating to the electronics at the top of the sensor and then use shrink tubing in order to fully protect these electronics from the elements.

Properly positioning temperature and humidity sensors in a hydroponic growing environment

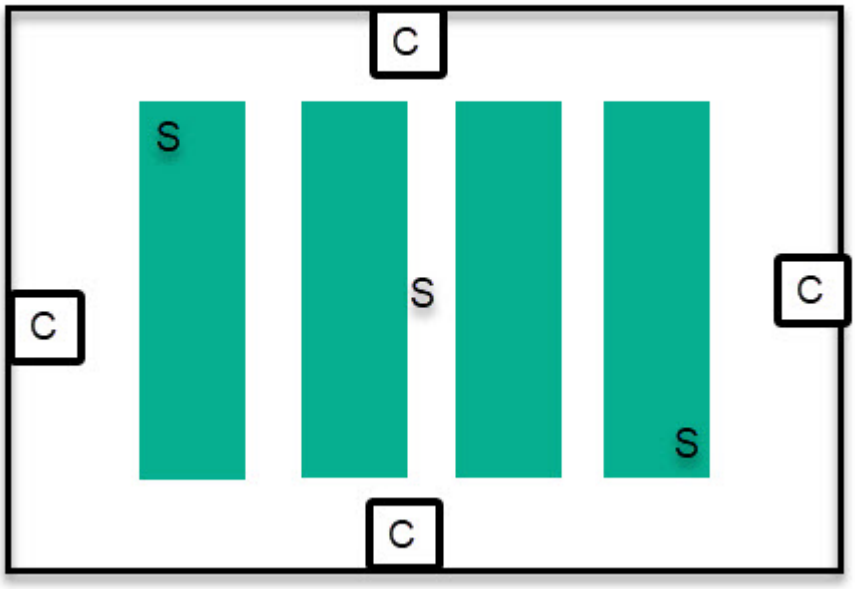
Temperature and humidity are two key variables you need to measure. They are important because they determine how your plants will transport water, and transpiration controls a lot of processes, with nutrient transport being one of the most important ones. However, the value of these variables in a growing environment – being that a greenhouse or a grow room – can change substantially depending on where they are measured. It is therefore critical to know where to place sensors and how to interpret their readings based on their location. In this post, we are going to discuss where it is best to measure these variables and what consequences it could have if these values are not measured properly.



Sensor placement relative to a control source (AC, humidifier/dehumidifier) for one or two sensors. Note that this setup assumes good circulation throughout the room, including middle of canopy.

Let’s start with the worst possible case, you only have one set of sensors and you need to control your environment with it. *In this case, place your temperature and humidity sensors at canopy height, as far away as possible from both the AC ducts and the humidifier/dehumidifier, make sure the sensors are hanging in the air and not stuck to a wall or tubing*. Then, make sure you use a hot wire anemometer to verify that your air movement speed is at least 0.3m/s across the entire room. This setup ensures that the worst controlled part of the room is at the correct value and it also attempts to minimize the gradient created from the control sources to the sensor using a good amount of air circulation. It is not perfect though and significantly different “climate zones” will be created close to and far away from the climate control devices.

The above setup can be used effectively in small growing environments, but can be problematic as both the number of plants and the size of the growing environment increases. At this point, using a single set of sensors is not an option if adequate climate control is desired. In these cases, multiple sensors need to be placed to ensure that climate control is being done properly. When using multiple sensors, place the second sensor at the place with the lowest air circulation inside the room, at canopy height, which is usually in the middle of the room, then place subsequence sensors as far away from either this or the first sensor in sequence. When doing climate control, the system needs to ensure all of the sensors remain within a “safety band”, making sure no sensor becomes too cold/hot, humid/dry, during control cycles.



Sensor placement for multiple control sources. Sensors are placed in order trying to always be as far away from sensors as possible but within the plant canopy.

When you implement a sensor system like this, you will realize pretty quickly that climate becomes very hard to control in a larger room when there is only one source of control (one AC, one humidifier, and one dehumidifier) because gradients become too big for effective control, so it takes too long for the AC

to be able to properly control the room while ensuring all sensors remain within proper boundaries. In this case, it becomes necessary to add multiple sources of control, so that the extent of gradients within the room can be minimized. This means adding multiple ducts for the output of an AC, multiple humidifier/dehumidifiers, etc.

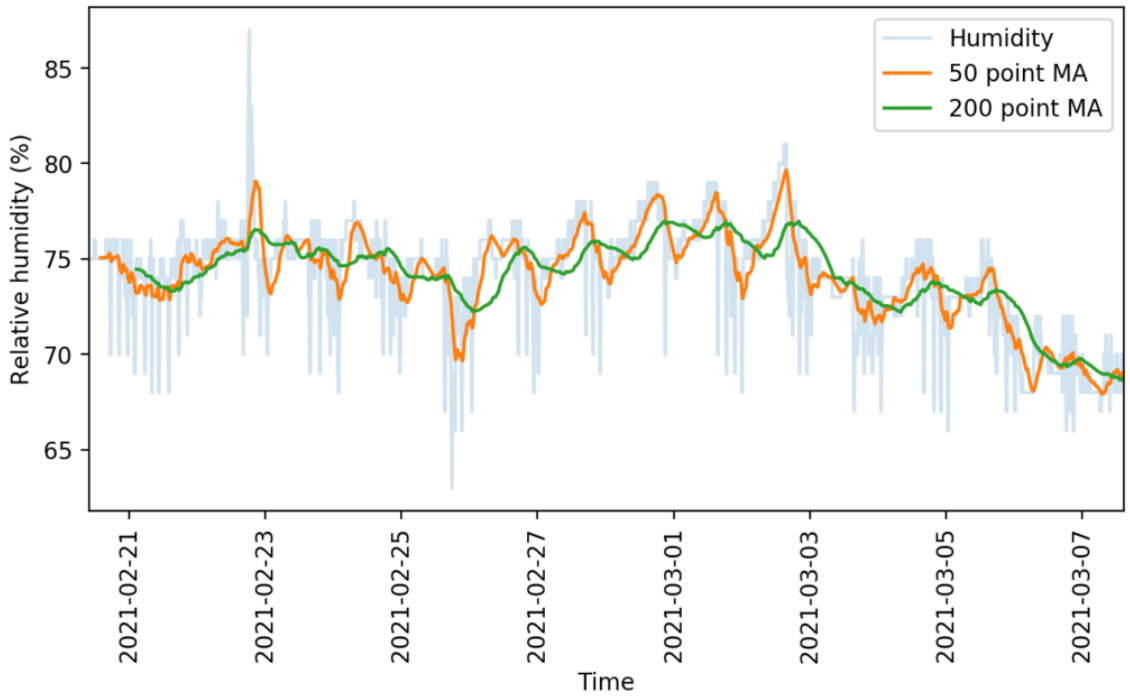
In these cases, sources of control are placed outside of the plant canopy to avoid plants being exposed to the flows from these control sources (which would expose them to very cold/hot/dry/humid air). Then the sensors need to be placed within the plant canopy, starting from as far away as possible from all sources of control – usually the middle of the canopy – and then to the corners of the growing environment.

Note that the control algorithm needs to ensure all of these sensors are within the proper control band and not attempt to control the average reading of these sensors. If you try to use the average of sensors to control a room, you might be left with a room where two extremes are present and the control system believes everything is ok while these extremes are maintained. The median is a better way to control a room, but it only becomes useful when 5 or more sensors are used. If only 2 or 3 sensors are used, ensuring all of these sensors are within adequate bands is fundamental to ensure that the room will have a lower chance of having humidity/temperature microclimates that will be detrimental to plant growth.

Making the most out of your hydroponic setup's logged

sensor and control data

If you have a hydroponic crop with a data logging and automated control solution, you probably have a lot of sensor and control data recorded that could be useful to take your crop's results to the next level. In this post, I am going to talk about some things that you could be doing with these stored data. You will see how the usage of this data opens up many possibilities and that even implementing the most basic of these suggestions could lead to important improvements in your understanding of your crop and its results.

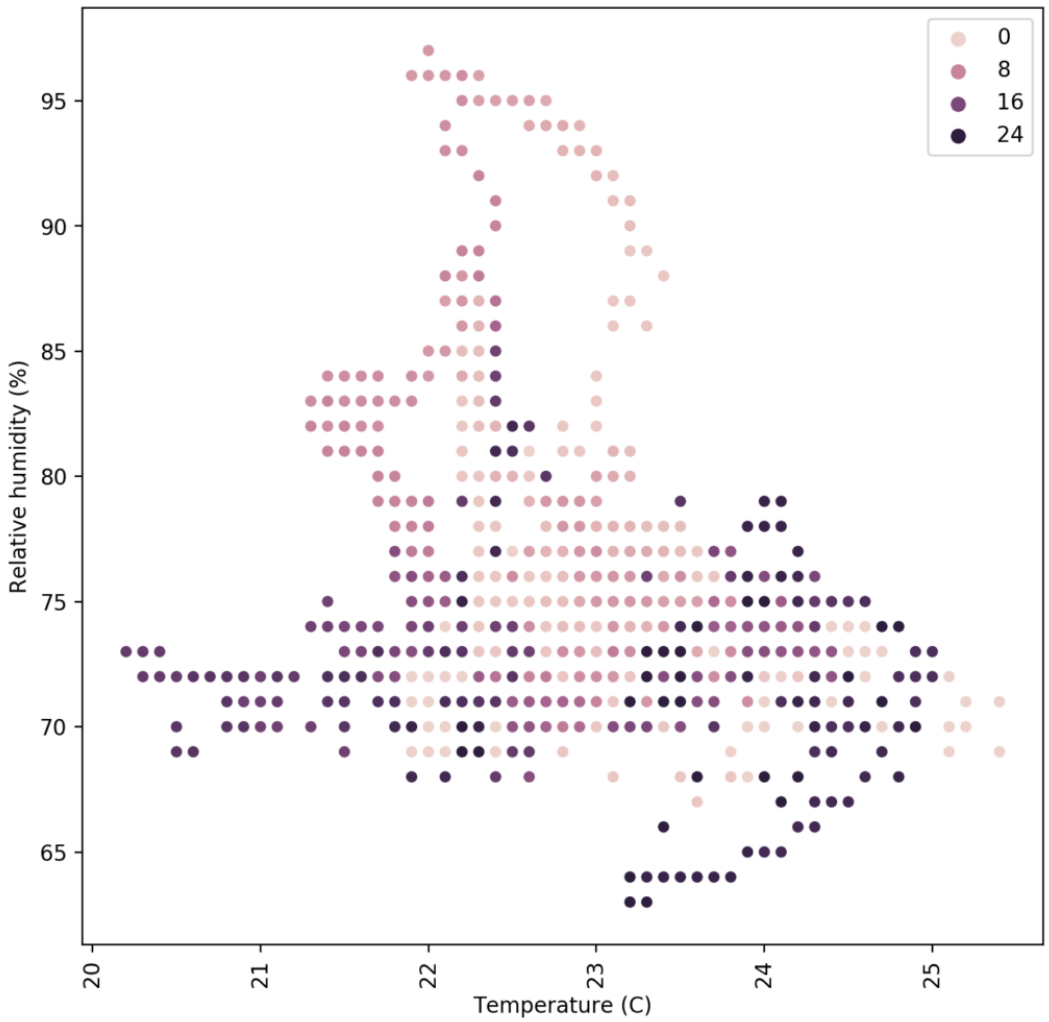


Use of different moving average to smooth out sensor readings.

The lowest hanging fruit to take advantage of logged data is to be able to download the data and put it into a database structure that you can properly query and search. Most data logging solutions record the data in either very simple structures, like csv files, or non-relational databases – like MongoDB – which are rather limited and do not allow for the degree of versatility that a true relational database engine offers. Having the data in a properly built database will allow you to start using it in a creative way. For example,

with the data in a proper database, it becomes possible to create a custom data visualization that can help you understand what’s going on inside your growing environment.

The images in this post show you some examples of this. The first one shows a simple example where a rather noisy humidity sensor is smoothed out using different moving averages, these averages can then be used to implement more effective control algorithms. The second image shows a detailed map of the temperature and humidity values experienced in a room, colored by the hour of the day. We can use this plot to easily locate where problematic times and VPD conditions might be, just by looking at when extreme readings happen. This behavior would be harder to observe and diagnose on a regular VPD Vs Time plot. Regular data logging web interfaces and platforms will not allow you to create plots of this sort, which is why putting your data into a proper DB and manipulating it to create custom visualizations can be very powerful.



Relative Humidity Vs Temperature map colored as a function of the hour of the day for a growing room being constantly monitored

The most powerful uses of the data come into play when you actually piece together your control and sensor data. Say you have an AC system coupled with a temperature sensor but you have a lot of other temperature and humidity readings and you also know the age of your plants at each point in time. Using this, you can create an advanced control algorithm where a system will use all of this additional information to know when to trigger AC systems and dehumidifiers to control the environment. Having a lot of logged data from a set point control system is a great starting point to train a reinforcement learning algorithm for climate control, since we know which control actions were taken at each point in time and we know the effect these had. Implementing such control mechanisms can lead to control systems that avoid spikes in

humidity and temperature across light on/off cycles, greatly smoothing out the environmental transitions for your crops.

Finally, there is also the potential to improve yields by gathering detailed mappings of yield data in a room and relating these yields with environmental sensors. If you have several different sensors in a room and you know the yield that you obtained on a per-plant basis, then you can create a map of all the yields in a room in order to see if there are important disparities in your yields because of differences in local humidity, temperature, light or air circulation levels. This can lead to important insights that can help better adjust climate conditions for the entire grow room. If multiple rooms are available, the information about environmental sensor data can be related to yields in order to stir all rooms towards more favorable conditions.

For example, after analyzing yield and temperature data from multiple growing cycles of one of my customers, we realized that the greenhouse with the lowest temperature standard deviations between sensors was giving the best yields, we then implemented better control algorithms on the other greenhouses to prevent this from happening, obtaining significantly better results across the board after that.

Data is a treasure. If you have been recording judicious sensor, climate control, and yield data through time, you're probably sitting on a gold mine that you haven't exploited yet. If you're interested in using my help to do so, please consider booking an hour of consultation time with me so that we can discuss your needs and how we could leverage your data to improve your growing results.