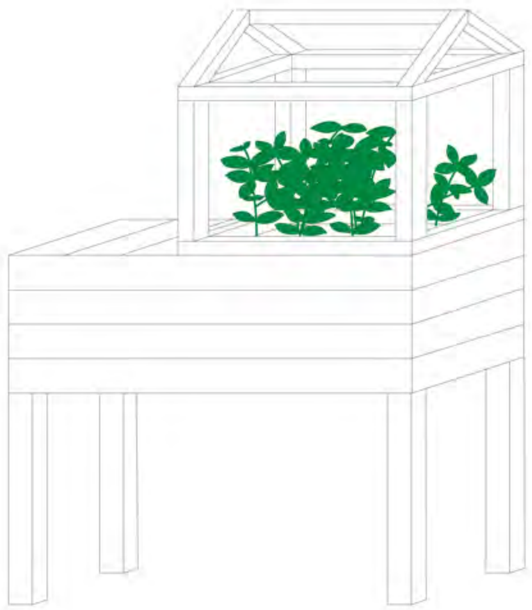
DEGREE PROJECT IN MECHANICAL ENGINEERING,
FIRST CYCLE, 15 CREDITS
STOCKHOLM, SWEDEN 2019

Automated Hydroponic system

JENS ORTNER

ERIK ÅGREN



KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF INDUSTRIAL ENGINEERING AND MANAGEMENT

Automated Hydroponic system

Jens Ortner, jortner@kth.se
Erik Ågren, eagr@kth.se

Bachelor's Thesis at ITM

Supervisor: Nihad Subasic
Examiner: Nihad Subasic

TRITA-ITM-EX 2019:61

May 2019

Abstract

This report includes research into how to automate a small scale system for hydroponics. Hydroponics is a growing technique which features a soil-less environment were the plants roots are exposed to a nutrient-enriched water solution.

The research focused mainly on how to regulate the pH and the level of nutrient in the water solvent and finding a system to automate that process.

In the research fully grown basil plants were used as test specimens, with the plants roots submerged in a water solvent. The water solvent had sensors that were connected to a micro controller making it possible to monitor the presence of nutrients and pH in the solvent.

If the micro controller deemed that the pH and/or the nutrient level was too high or too low, the micro controller would adjust the solvent by activating pumps adding pH down buffer solution and/or nutrient solution to the solvent.

The research proved that a way to automate a small scale hydroponics system is by building a computerized system consisting of:

- Micro controller.
- pH sensor.
- EC sensor (to measure nutrient level in solvent).
- Temperature sensor.
- Fluid pumps connected to pH- and nutrient reservoirs.

Keywords: Hydroponic system, Hydroculture, Automation, Mechatronics

Sammanfattning

I denna rapport följer en forskning om hur ett system för hydroponics kan automatiseras. Hydroponics är en odlingsteknik som utesluter nyttjandet av jord. Istället får plantorna näring och vatten via en näringsrik vattenlösning som dess rötter är i kontakt med.

Forskningen fokuserade huvudsakligen på hur man reglerar pH och nivån av näringsämnen i en vattenlösning och skapa ett system för att automatisera denna processen.

I undersökningen användes fullvuxna basilikaväxter som prover med plantornas rötter nedsänkta i ett vattenlösning.

Vattenlösningen hade sensorer som var anslutna till en mikrostyrenhet som gjorde det möjligt att övervaka nivån av näringsämnen och pH i vattemlösningen.

Om mikrokontrollern ansåg att pH- och/eller nivån av näringsämnen var felaktig så skulle mikrostyrenheten justera vattenlösningen. Detta skedde genom att mikrostyrenheten aktiverade vätskepumpar som tillsatte pH-buffer och/eller näringslösning.

Forskningen visade att ett sätt att automatisera ett hydroponicsystem är att bygga ett datoriserat system som består utav:

- mikrostyrenhet.
- pH mätare.
- EC mätare (används för att mäta näringsnivån i vattenlösningen).
- Temperaturmätare.
- Vätskepumpar anslutna till behållare innehållandes pH- och näringslösning.

Nyckelord: Hydroponicsystem, Hydrokultur, Automation, Mekatronik

Acknowledgements

We would like to show our gratitude to our supervisor Nihad Subasic for your consultancy and guidance. We would also like to thank Krister Ågren for supplying this research with material and Staffan Qvarnström for helping us with technical expertise.

Contents

List of Figures

List of Tables

Nomenclature

AC	Alternating current
API	Application programming interface
DC	Direct current
DWC	Deep water culture
EC	Electric conductivity
HTML	HyperText Markup Language
IoT	Internet of things
IP	Internet Protocol
LED	Light emitting diod
NC	Normally Closed
pH	Potential hydrogen
PWM	Pulse-Width Modulation
URL	Uniform Resource Locator
V	Volts

Chapter 1

Introduction

In this chapter follows a presentation on what this research will include.

1.1 Background

Unlike conventional growing, in a hydroponic system the plant is never in contact with soil. Instead the roots of the plants are in contact with a nutritional water solvent.

Research suggests that soil-less growing can result in crops growing faster, can take up 1/5th of the space and need 1/20th of the water compared to growing in soil [1]. Soil-less growing also has a smaller risk of diseases and pests, which allows the use of less pesticides [2].
In a society where the demand for farming is rising every year due to a growing world population, hydroponic farms form a possibility to make growing crops more efficient [3].

1.2 Purpose

This report includes research on how to automate a small scale hydroponic system and use this system to grow sweet basil.
The properties of the environment are of high importance in order for the plants to grow properly in a hydroponic system, and therefore, maintenance of the properties are important. Parameters that should be considered to maintain a fertile environment for the plants are for example pH-, nutrient-, oxygen- and water-levels along with sufficient exposure to sun light.

Questions that will be researched and answered in this report includes:

- How to automate a small scale hydroponics system?
- How to measure the parameters that are vital to properly regulate the system?
- How should the parameters be regulated?

1.3 Scope

The project will be centered around the creation of a small scale automated hydroponics system with the use of mechatronics. The system itself shall be designed to house and maintain six medium sized sweet basil plants. Sweet basil will be suitable for this research because it is easy to maintain and grows fast compared to other plants [6].

The system is intended to keep track of following parameters:

- pH concentration
- Nutrient solution concentration

If the system recognizes that the concentration of pH and/or nutrient level is not optimal, it will automatically compensate by adding more of either nutrient- and/or pH solution to the water solvent.

The system will be placed indoors by a window. This will give the system sufficient sunlight and ambient temperature, hence an external heater and light source will not be included in the system for regulation.

An air pump will be added to the water solution. This will provide the water solvent with a constant rate of oxygen. The air pump will not be directly regulated.

Chapter 2

Theory

This chapter will include the theory required to answer the research questions asked in the previous chapter (see section 1.2 Purpose). Additional information will be brought up to cover the theory required to maintain a sustainable hydroponic system.

2.1 Methods of hydroponics

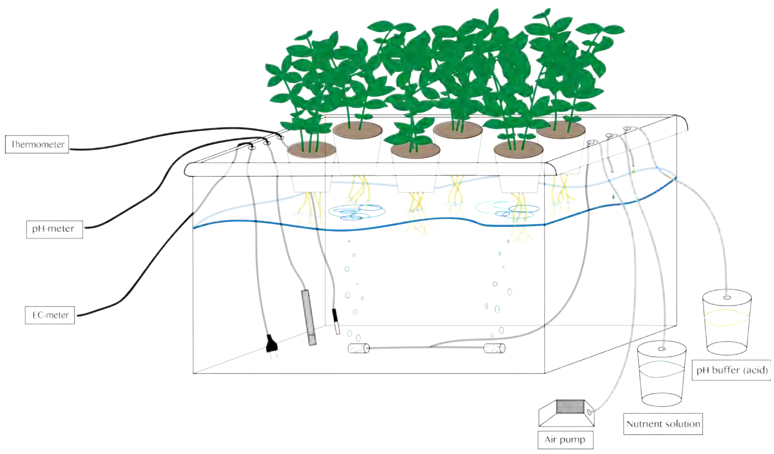


Figure 2.1: Schematic illustration of deep water culture, created in Adobe Illustrator.

There is a few ways to grow with hydroponics. This project will use a method called Deep Water Culture (DWC). In DWC the roots of the plants are constantly submerged in a water solvent. Therefore the plants will extract nutrients from the solution that its roots are continuously in contacts with.

It will also be of vital importance to keep the water solution oxygenated with an air pump. This will prevent the plant from suffocating.

The main advantages of DWC, compared to other hydroponics methods, are that the water is highly oxygenated, therefore the system will require less nutrients and generally need less maintenance [4].

2.2 Water solvent

The nutrients that are required for a plant is divided into two categories, macronutrients and micronutrients. Macronutrients consists of elements that the plant needs in larger quantities compared to micronutrients. Among macronutrients includes for example nitrogen, phosphorus, potassium, carbon etc. Some examples of micronutrients are iron, zinc, copper, nickel etc. These elements can be found naturally in the ground, but in order for the plants to get the crucial nutrients in a hydroponic system, a nutrient solution needs to be added to the water. There is a big variety of different premixed nutrient solution on the market that is suitable for different applications (plants, system, environments etc.).
A suitable water and ambient temperature for growing sweet basil is between 18°C and 35°C [6][8].

2.2.1 Electric conductivity

To measure the nutrition level in the water solvent the system will use an electric conductivity sensor (EC-meter). The electric conductivity of a material represents the ability to transport an electric charge. The SI-unit for electric conductivity is $[S/m]$ (siemens per meter). Siemens is the same unit as the inverse of the electric resistance Ω^{-1} (ohm^{-1}). An EC-meter will not be able to measure the levels of each individual nutrient, but an increasing presence of nutrients and minerals in a water solution will increase the ability to transport an electric charge i.e. lower the resistance.
An EC meter will be built using an EU electric adapter plug. The plug will be submerged into the water solvent and be connected to the micro controller. The electrodes of the plug will be used to measure the resistance in the water solvent, and thus the EC can be calculated. To avoid polarization of the fluid that is to be measured, alternating current (AC) should be used to get as precise readings as possible. However, taking a quick measurement using direct current (DC) can be done without suffering a considerable effect of polarization [11].
The EC-value of a fluid has an approximate linear relation to the temperature with a roughly 2% increase of EC for every increasing degree Celsius.

Even though the the EC level may change depending on the temperature, the nutrient content in the solution will not. Therefore, an equivalent EC at 25°C is usually calculated to be able to define an interval of reference for the EC that is suitable for the particular plant of interest. The EC equivalent is calculated from:

$$EC_{25} = \frac{EC_t}{1 + a(t - 25)} \tag{2.1}$$

EC_t is the measured EC of the solvent, t is the measured temperature and a is a temperature compensation factor which is commonly set to 0.019 °C^{-1} for hydro-culture monitoring [9-12]

The EC-value of a fluid can be calculated from the following formula:

$$EC = K \cdot G \tag{2.2}$$

Where G is the conductance which is the same as the inverse of the measured resistance and K is the cell constant of the electric probe. The cell constant can be calculated from:

$$K = \frac{d}{A}$$

where d is the distance between the probes and A is the effective area of the probes. The effective area can be difficult to calculate exactly, therefore it is preferable to obtain the cell constant, K, by measuring the resistance in a reference fluid with known EC value. [13].

An ideal EC level for sweet basil is in the interval: 1.4-1.8 mS/cm, but a woody herb such as sweet basil can tolerate even higher levels of EC[14].

2.2.2 pH

The pH level of a solution represents the molar concentration of hydrogen ions (H^+) that is present in the solution. The more hydrogen ions present in the solvent the lower (more acidic) the solution is. A pH-level of seven at 25°C is said to be neutral, a pH below seven are considered acidic and a pH higher than seven are basic.
A common way to measure pH is with a glass-electrode probe. The probe measures difference in electric potential between the glass-electrode and a reference electrode to get the pH level of the solution [5].
A slightly acidic pH range between 5.6-6.6 is suitable for sweet basil[6].

Most formulas of nutrient solution, when added to the solvent, will initially lower the solvents pH to a level suitable for most hydroponics cultures. However, having mature plants growing in a hydroponics system, the pH tends to continuously increase. That increase can be adjusted by adding doses of an acidic solution to the solvent [7].

2.2.3 Fluid pump

To be able to control and regulate the water solution, both pH buffer and nutrient solution need to be added using a fluid pump. A peristaltic pump is deemed as optimal because it grants a constant flow and will not have issues with corroded parts from having acidic substances being pumped trough it [15]. The principle of a peristaltic pumps is that a hose or tube is getting squeezed in a linear motion which causes the fluid to move in the same direction. Electrical peristaltic pumps usually consists of a DC motor and an elastic rubber tube. A pump of this type operates at a slower flow rate compared to other types [15], but since the speed of which the solutions is getting added is not important, this disadvantage will be neglected.

2.2.4 Air pump

To properly oxygenate the roots of the plant, the DWC-system needs an air pump. Rubber tubes will transport the air from the air pump to two air stones that will be placed at the bottom of the water solvent container. Air bubbles will seep through the pores of the air stones and ascend to the surface. When reaching the surface, the bubbles will disrupt the surface and break the surface tension, which allows oxygen molecules to enter the water[16]. Therefore, it is important to generate enough water bubbles, so that a sufficient amount of oxygen molecules is present in the water solvent. A common rule of thumb when choosing an air pump for a small scale indoor DWC system is that they should produce at least 500-600 ml of air per minute[17].

2.3 Micro controller

This section includes an overview on how the micro controller Arduino Uno works.

2.3.1 Analog pins

The Arduino carries an analog-to-digital converter that has a resolution of 10 bits. This means that these pins can take voltage inputs (typically between 0-5V) and turn it into integer values between 0 and 1023 [19].

These input integers can then be converted to fit the different sensor readings. For example a pH sensor sends an input integer from a reading to the Arduino Uno. The micro controller can then run a script to compute a pH-value of that reading.

2.3.2 Digital pins

The digital pins on the Arduino Uno holds different functionalities. *Pinmode* is used to specify if you want the pin to be acting as input or output. *DigitalWrite* can make the pin set to an output value of 5V for HIGH and 0V for LOW [20].

2.3.3 Serial pins

Arduino Uno has two pins called RX, the receiving port, and TX, the transmitting port. These ports are used for communication between a computer or other serial devices using logic level signals. This means that it sends and receives signals at 0V or 5V that gets interpreted as zeros and ones that can make up code instructions [21].

2.4 Data storage and analysis

Here follows an explanation on how to store, analyze and visualize data using an Internet of Things (IoT) platform. The data flow is illustrated in Figure 2.2.

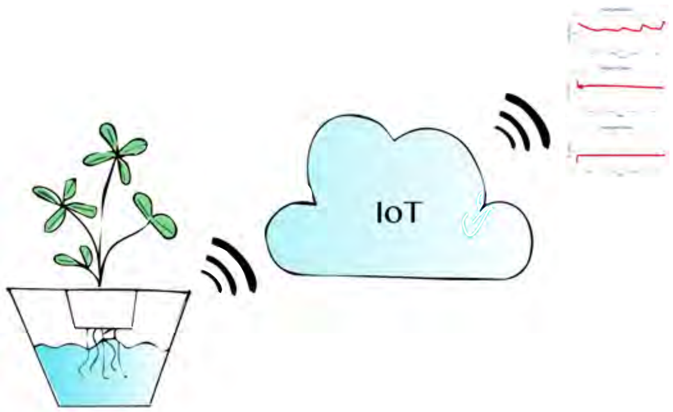


Figure 2.2: Figure of a IoT data-flow, created in Adobe Illustrator

2.4.1 ThingSpeak

ThingSpeak is an analytic tool developed as a cloud based platform for IoT devices. ThingSpeak has the ability to visualize and analyze live data and publish it on a live feed channel[18].

The ThingSpeak channels has unique Application Programming Interface (API) keys, that with the use of HTML commands, make it possible to read and write directly to a specific channel. The data this channel stores can then be remotely accessed using a web browser to plot the data in the form of graphs [18].

2.4.2 ESP8266 WiFi module

The ESP8266 is a network solution adapted for Arduino among other micro controllers. In this project the ESP8266 shall be used as WiFi-adapter to allow the Arduino to connect to a WiFi-network and gain access to a ThingSpeak channel.

To establish an internet connection between a WiFi-module and a WiFi-network, a script uploaded to an Arduino board need to hold instructions for how the WiFi-chip shall operate. This includes which WiFi-network to connect to, but also instructions on how to transmit the data the Arduino gathers to a ThingSpeak-channel.

Chapter 3

Methodology

In this chapter follows the steps and methods on how to set up the hardware and software for this research.

3.1 Equipment

The equipment used in this research is specified in Table 3.1.

Table 3.1: List of equipment used for the research

Product type	**Quantity**
20L plastic container	1
Micro controller	1
pH probe	1
pH meter	1
WiFi-module	1
1000 ohms resistor	1
EU AC power cord	1
Peristaltic pump	2
Water thermometer	1
Air pump with air stones	1
Net pots for hydroponic	6
Breadboard	1
pH buffer (down)	1
Nutrient solution	1
Relay	2
Sweet basil plants	6

For this research, a premixed solution that is specifically made for hydroponic systems was used. The nutrient solvent was an aqueous solution (liquid), labeled as NPK 4-0.7-3. This indicates that the solution consists of 4% nitrogen, 0.7% phosphorus and 3% potassium. Other macronutrients were 0.3% magnesium, 0.3% sulfur and 1% calcium. Some of the micronutrients in the solvent were 0.02% boron, 0.005% copper and 0.05% iron.
The air pump used was specified with a flow rate of 50 litres of air per hour (or approximately 800 ml of air per minute). The hydroponic system was placed indoors, close to a window, where the temperature was about 22°C and the plants got the 10-12 hours of direct sunlight it required [6].

3.2 Hardware setup

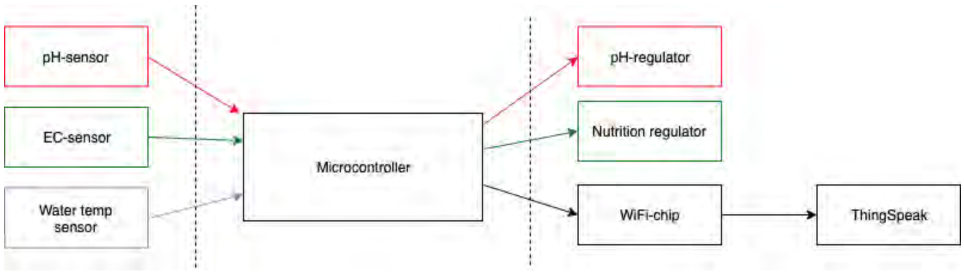


Figure 3.1: Schematic illustration of the hardware, drawn in Draw.io

Figure 3.1 shows a schematic illustration of the different hardware components used in the system. The arrows in the figure indicates the flow of information between the different hardware components. The system consisted of three different kinds of parts: sensors, a micro controller and actuators.

In figure 3.2, an image of how each different component were connected to the micro controller can be viewed.

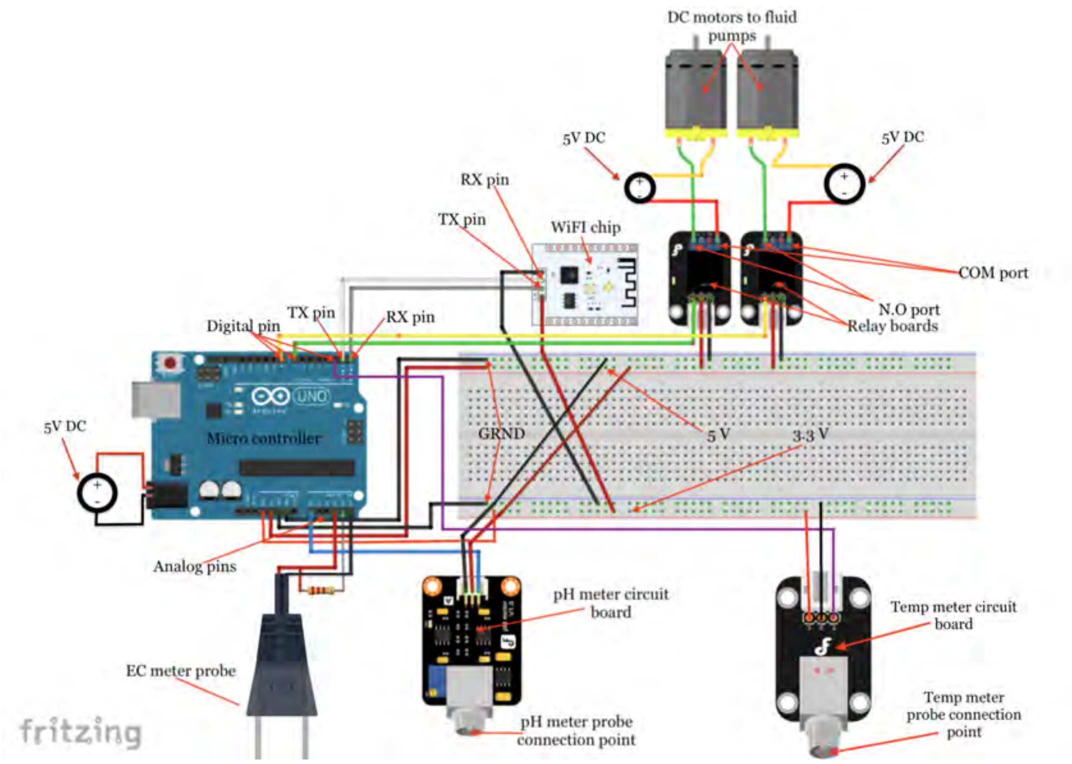


Figure 3.2: Coupling scheme of the system, drawn in Fritzing.

3.2.1 pH sensor

The pH sensor probe was connected to the Arduino chip via a pH meter circuit board that can be seen in Figure 3.2. The pH meter was connected to the Arduino chip through it's analog gates along with supplied power at 5V.

The specific pH probe used in this research is of the glass-bulb electrode type. The probe was also specifically made for long term submerging [22]. This means that it could monitor the water solvent for longer periods of time without needing frequent calibration.

A manual for the pH electrode can be found in Appendix B.

3.2.2 pH meter calibration

To get as precise readings of the pH level as possible, the pH meter was calibrated using 2 different pH calibration fluids with known pH value. One of the fluids had a pH of 4.01 and the other had a pH of 7.01. The probe of the pH meter was rinsed and then submerged in the calibration fluids. By observing the voltage measured by the pH probe (see Table 3.2), a linear approximation was made for the pH-value as a function of the measured voltage.

Table 3.2: Measured voltage of known pH

pH (at 22°)	**mV**
4.01	0.98
7.01	1.93

After the calibration was made, the pH probe was once again submerged into the calibration fluids and the following pH-values were measured:

Table 3.3: Measured pH after calibration

Actual pH (at 22°C)	**measured pH (at 22°C)**	**Relative error**
4.01	4.04	0.75%
7.01	7.03	0.29%

3.2.3 EC meter

The EC meter was built out of an EU power cord, acting as a probe, and a 1000 ohms resistor. The two poles of the cord was then connected to the analog A1, A2 and A3 analog pins on the Arduino micro controller. The A2 pin would act as a ground reference for one of the poles on the power cord. The A1 pin would supply power to create voltage between the poles of the power cord. The A3 pin would, along with a resistor, measure the voltage created between the A1 and A2 pin. The Arduino chip would then convert the voltage input to an analog value that could be calculated into an EC value for the water solvent.

3.2.4 EC meter calibration

To be able to use the electric probe as an EC meter it had to be calibrated to determine the cell constant of the probe. This was done by submerging the electric plug and a thermometer into an EC calibration fluid with an EC of 1413 μS/cm (at 25°C). After measuring the resistance the cell constant could be calculated to 1.44 cm^{-1}

The program that was used to calibrate the EC meter can be found in Appendix D.

3.2.5 Fluid pumps

The fluid pumps were connected in two separate circuits (see Fig. 3.2) to be able to control each of the two pumps individually.

The only control settings needed for these pumps were activation to full speed, to feed liquid, and total power cut off. Therefore, each of the two circuits had a relay board to enable such functionality.

The relay boards was connected to the digital pins on the micro controller and to 5 V for operating voltage along with a ground connection.

The DC motors for the pumps were then connected in series with a 5 V power source and then to the normally closed (NC) port on the relay boards to complete the circuits. This meant that pumps would be consistently off, unless a signal was sent to the relays from the Arduino instructing it to open and turn the pumps on.

3.2.6 WiFi module

The WiFi module's serial ports were connected to the micro controller's serial ports, RX and TX, to be able to establish serial communication between the WiFi module and the micro controller. For external power the WiFi chip used the 3.3 V port on the Arduino.

See Appendix C for the instructions used to connect the WiFi module to ThingSpeak.

3.3 Software setup

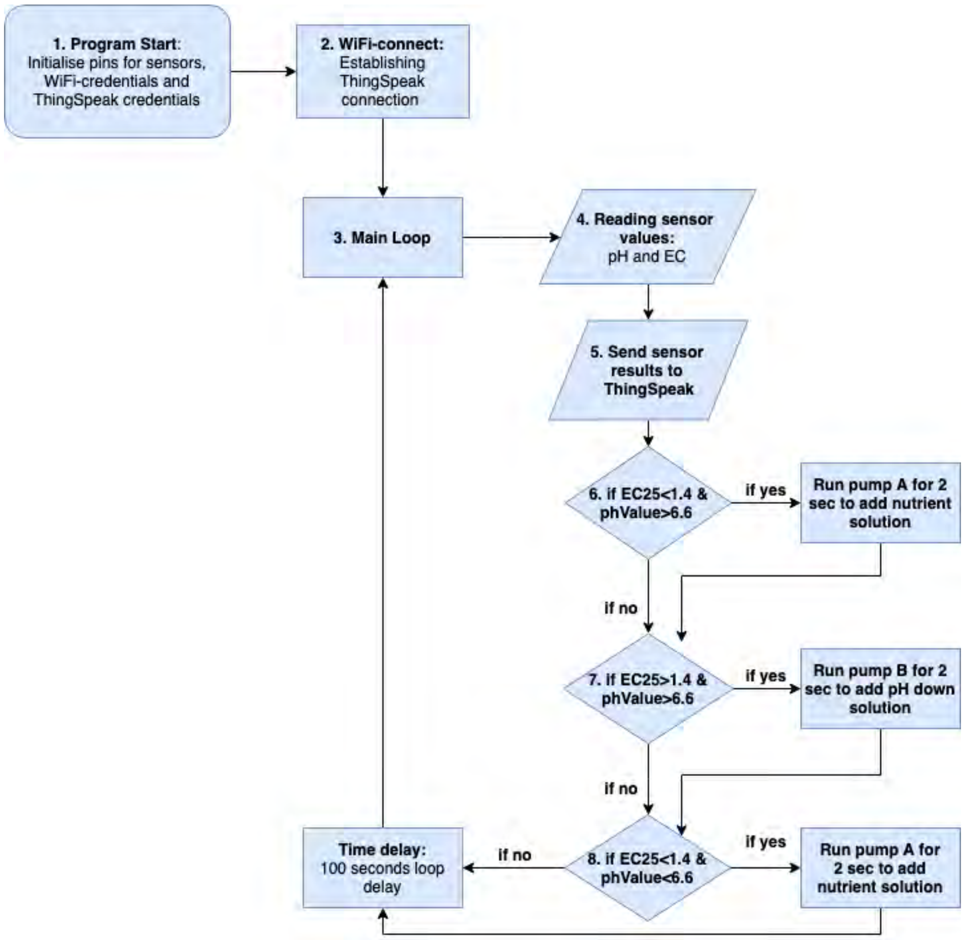


Figure 3.3: Schematic illustration of software, drawn in draw.io

The different purposes and states of the Arduino code are illustrated in Figure 3.1.

The program script can be found in Appendix E.

1. Program start: This section of the software included code library, pin setup for the sensors and the relays actuating the pumps. The information the WiFi chip needed to establish connection to ThingSpeak is also defined in the beginning of the code. This includes the log-in credentials for the WiFi-network, the IP address for ThingSpeak's API and a URL-Get parameter associated with the ThingSpeak channel for data upload.

2. WiFi-connect: Here the Arduino chip started to communicate directly to the WiFi-chip trough the serial ports telling it to connect to the previous defined network. In case of failing to connect to the network, the program would continue to the main loop.

3. Main Loop: This was the cycle that handles the automation process, making the program continuously take readings from the sensors, upload the data for the readings to a ThingSpeak channel. The data readings would be the basis for the system whether or not to add pH buffer and/or nutrient solution trough the pumps.

4. Reading sensor values: Every time the pH was to be measured the pH-sensor took ten sensor samples which were then sent to the analog port of the Arduino in the form of an array consisting of ten integers. By sorting the array by size and picking out the six middle entries, an average of the readings could be calculated. That average was then converted by the Arduino into a pH value.

The temperature sensor software worked similar to the pH-sensor software. The Arduino would instruct the temperature sensor circuit board to take a series of measurements. These measurements would then get uploaded to the Arduino chip through a digital port. The Arduino would calculate an average of the samples taken and convert it into a Celsius reading.

By setting the power supply to the EC sensor to 5 V, the voltage drop between the poles of the probe got picked up by the Arduino as an analog input. The algorithm then converted the input to a value corresponding to how much the voltage had dropped between the poles of the probe. The voltage drop could then be related by the Arduino as a corresponding EC-value.

5. Sending sensor data to ThingSpeak channel: When the sensors had produced a set of readings, that data got sent away to be stored at the ThingSpeak channel. The Arduino would send commands to the WiFi-chip through the serial ports. These commands would instruct the WiFi-chip to establish a connection to the ThingSpeak channel with an IP-address.

The Arduino would then create a URL Get-parameter consisting of the pH, EC and temperature readings and send it as a text string. The text string then got passed to the ThingSpeak channel through the WiFi chip.

6, 7 & 8. pump states:

In this research there would not be any way to counter a water solvent with a rising nutrient level and/or to low pH.

Therefore, it was required to only add a small amount of the liquids at a time to avoid excessive compensation. This was achieved by running the pumps during short time periods to create a slow decrease of the pH and a slow increase of the EC value of the water solvent.

The algorithm would decide on which solution to add to the solvent based on the following three states:

State 6:

If the EC level is measured below 1.4 mS/cm and the pH value is measured above 6.6 the system would only compensate by adding nutrient solution.

State 7:

If the EC level is measured above 1.4 mS/cm and the pH level is measured above 6.6 the system would add pH down buffer to make the water more acidic.

State 8:

If the EC value is below 1.4 mS/cm and pH-value below 6.6 the nutrient solution would be added.

This state is mainly for adding nutrient solution if the EC value would drop.

Time delay:

If solution was added to the water solvent during the loop cycle it was important to let the water solvent mix properly. Faulty sensor readings could then be avoided for the up coming loop cycle. Therefore a 100 second time delay was added upon the completion of each Main Loop cycle.

Chapter 4

Results

In this chapter raw data is presented in the form of graphs. The data is gathered as a result of the different tests.

4.1 Reaction test

A test was conducted to analyze how the system reacts when nutrient- and pH buffer solution was added to the water solvent. The test was performed by filling up the plastic container with 15 liters of tap water. Six sweet basil plants were placed on top of the container with the roots submerged in the solvent. Initially the pH of the water solvent was 7.48 and the EC was 0.31 mS/cm. At the start of the test one of the peristaltic pumps added nutrient solution for two seconds. After 150 minutes, another peristaltic pump added a pH down buffer solution for two seconds. The test was conducted during a time period of 22 hours. Data was gathered every 100 seconds throughout the test period. The result from this test is shown in Figure 4.1 on the next page.

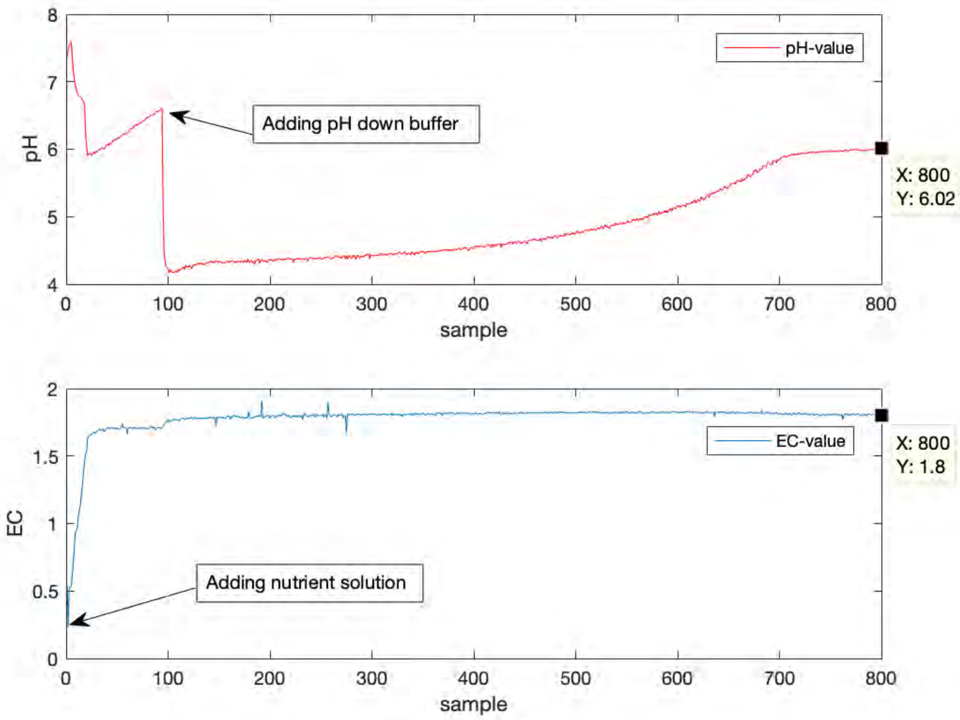


Figure 4.1: Data collected during the reaction test, plotted in MATLAB.

4.2 System behavior test

To test the system's ability to keep the different parameters inside desirable intervals, a test was conducted during a period of 16 days. The test was carried out by using 6 fully grown sweet basil plants in a 20 liter plastic container. The container was then filled with 15 liters of tap water and all the electrical components were turned on. The system initially had a pH value of 7.56 and an EC-value of 0.70 mS/cm. After 7 days an additional 2 liters of water was added to the water solvent to compensate for water loss due to evaporation. Data was gathered every 100 seconds throughout the test period. The result from this test is shown in Figure 4.2.

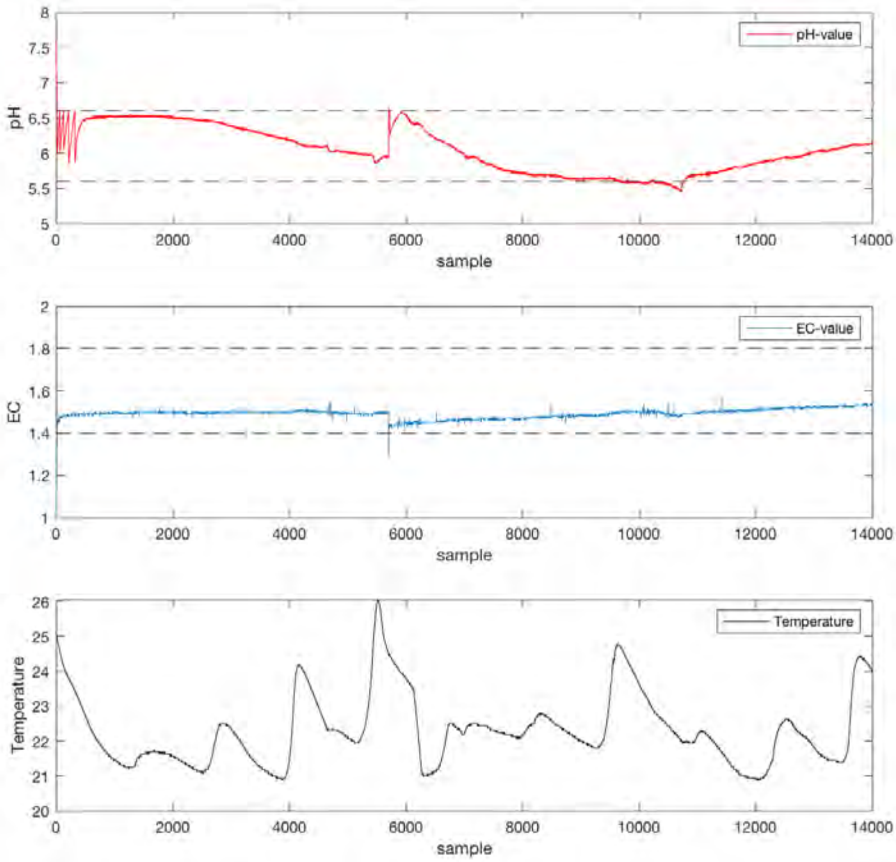


Figure 4.2: Data collected during the system behavior test, plotted in MATLAB.

The dashed lines in the pH- and EC-value plot indicates the upper and lower limit of the desired interval.

Chapter 5

Discussion

In this section the significance and validity of the results will be discussed and analyzed.

5.1 Analysis of the reaction test

From the result of the reaction test (see Figure 4.1), it can be seen that when adding nutrient solution, the pH-level decreases. Adding pH down buffer solution however, does not seem to influence the EC-value significantly. After both the pH buffer- and nutrient solution was added, the EC-value was stable at a constant rate. However, the pH value kept increasing for about 600 readings (approximately 16.5 hours) until it converged to a pH of roughly six. This confirms what was stated in section 2.2.2 about matured plants increasing the pH-value in a hydroponics system.

5.2 Analysis of the system behavior test

From Figure 4.2 it can be seen that the system managed to keep both the pH- and EC level within their desired interval for most of the test period. At around sample number 6000, two liters of tap water was added to the water solvent. This increased the pH value and lowered the EC level below the acceptable limit which forced the system to add more nutrition.

It can be seen that the pH level gradually was lowered around sample number 500 to 5500 and also around sample number 6000 to 11000. This might have been due to the water source used in this experiment. Tap water usually holds chemicals and elements that will cause pH instability in a hydroponics system. Other possible causes could have been because of different chemical reactions occurring when the nutrients breaks down in the water solvent. Also, the presence of microorganisms such as bacteria and algae can affect the pH by making it more acidic [23].

5.3 Error sources

The data collected during this research may deviate from the actual values due to a variety of different error sources. A faulty reading may originate from a bad calibration of the pH- and/or EC-meter due to contamination of the calibration fluids.

To get more precise monitoring of the EC, an industrial standard EC-meter would have been preferable.

The EC-meter used in this research was connected with DC which may have caused polarization of the ions in the fluid, which would have affected the resistance and given inaccurate readings.

When submerged in a fluid for a long period of time the probes of the sensors can get contaminated by a coating of charged ions. This especially is a likely source of error for the EC-meter, where the probes will be electrically charged while making a reading. This will attract negative and positive charged ions which will create a coating on the probes and affect the measured resistance of the fluid. The probe of the pH-meter is protected by a glass bulb which can get polluted by different salts and minerals.

A bad circulation of fluid may cause an uneven concentration of nutrients and hydrogen atoms which will result in a different distribution of pH and nutrient content in the container. Since the sensors are positioned at the bottom of the container, this may cause the system to regulate the parameters even though the levels at the surface close to the roots may be within acceptable range or vice versa.

Chapter 6

Conclusion

6.1 Addressing the purpose

In this section the questions stated in the purpose (section 1.2) will be answered.

6.1.1 How to automate a small scale hydroponics system?

The research proved that sensors to measure the pH, temperature and EC of the water solvent along with two pumps, to feed pH down buffer and nutrient solution, connected to a micro controller were needed for automation.

The micro controller needed an algorithm instructing it to continuously monitor the pH and EC level and add pH down buffer or nutrient solution to the solvent if needed.

6.1.2 How to measure the parameters that are vital to properly regulate the system?

The parameters that were most vital to regulate the system were the total level of nutrients and pH in the water solvent.

The pH level was measured with a glass-electrode probe that was submerged in the water solvent. The pH sensor was calibrated with specifically designed pH calibration fluids to obtain more accuracy of the pH readings.

The total level of nutrients were measured with a EC-meter measuring the electrical conductivity of the solvent. The electrical conductivity of the solvent was dependent on the temperature. Therefore, a temperature sensor was added to the system to make up for misreading due to temperature fluctuations. The EC-meter was calibrated, in the same fashion as for the pH meter, to get more accuracy of the EC readings.

6.1.3 How should the parameters be regulated?

The research proved that a way to regulate the pH and nutrient level in the water solvent is by having a computerized system including:

- Micro controller.
- pH sensor.
- EC sensor (to measure nutrient level in solvent).
- Temperature sensor.
- Fluid pumps connected to pH- and nutrient reservoirs.
- pH down buffer solution
- Nutrient solution

With the above setup it is possible to continuously monitor and regulate these parameters in a water solvent and adjust these parameters on demand.

The research showed that a way to keep the pH- and nutrient levels stable at an acceptable interval, was by considering the following states and actions:

State	Action
EC level below lower limit, pH level above upper limit	Add nutrient solution
EC level below lower limit, pH level inside the limits	Add nutrient solution
EC level inside the limits, pH level above the upper limit	Add an acidic pH solution

Upper and lower limit indicates the boundaries of the acceptable range.

6.2 Recommendation for future improvements

To further automate a hydroponics system there are several features that would be advisable to add.

To be able to counter a falling pH level in a water solvent an extra pump connected to a fresh water reservoir should be added. This feature could also be used to dilute the nutrient level in a water solvent if deemed necessary.

To avoid large fluctuations of the pH- and EC level due to excessive addition of pH buffer and/or nutrient solution to the water solvent, a function that could calculate the amount of pH buffer and/or nutrient solution to add.

To make the system less location dependent, and for instance be able to be stored in a cellar, a LED growing light with a timer should be added to mimic the sunlight the plants need.

The system could also be completely closed by having a green house enclose it. This would reduce the need of adding more water due to evaporation. To control the climate within the green house a humidity sensor and a servo motor can be used to regulate a hatch to air the plants.

Bibliography

[1] Deorao Sardare and Shraddha V. Admane ”A review on plant without soil - hydroponics”, *MIT Academy of engineering Alandi Pune*, Vol. 3, no. 3, 2013, p. 303

[2] AdvancedNutrients, *How To Start Growing With Hydroponics For Beginners.* [online] Advanced Nutrients, 15 December 2016, Available at <`https://www.advancednutrients.com/articles/easy-hydroponics-beginners-guide/`> [received May 30 2019]

[3] Food and Agriculture Organization of the United Nations, *The future of food and agriculture: Trends and challenges.* [online] 15 December 2016, available at <`http://www.fao.org/3/a-i6583e.pdf`> [received May 30 2019]

[4] M.F. Saaid, N.A.M. Yahya, M.Z.H. Noor and M.S.A. Megat Ali. ”A development of an automatic microcontroller system for Deep Water Culture (DWC).” *2013 IEEE 9th International Colloquium on Signal Processing and its Applications, CSPA 2013*, pp. 328-332

[5] Vanýsek, Petr. *The Glass pH Electrode.* [online] The Electrochemical Society, summer 2005, Available at <`https://www.electrochem.org/dl/interface/sum/sum04/IF6-04-Pages19-20.pdf`> [received May 30 2019]

[6] Stone, Amy. *Growing Hydroponic Basil? Read This First.* [online] Upstartfarmer, 22 December 2016, <`https://university.upstartfarmers.com/blog/hydroponic-basil`> [received February 14 2019]

[7] Morgan, Lynette. *Perfecting the pH of Your Hydroponic Nutrient Solution.* [online] Maximum Yield, 1 December 2016 Available at <`https://www.maximumyield.com/perfecting-ph/2/1212`> [received April 29 2019]

[8] Saha, Subhrajit, Monroe, Amber and R.Day, Martin. ”Growth, yield, plant quality and nutrition of basil (Ocimum basilicum L.) under soilless agricultural systems”, *Annals Of Agricultural Science*, vol. 61, no. 2, 2016, pp. 181-86

[9] Hayashi, Masaki. ”Temperature-Electrical Conductivity Relation of Water for Environmental Monitoring and Geophysical Data Inversion”, *Environmental Monitoring and Assessment*, vol. 96, 2004, pp.119–28

[10] Wittman, Mark. *Understanding EC, TDS and PPM*, [online] Maximum Yield, January 1 2016 Available at <https://www.maximumyield.com/understanding-ec-tds-and-ppm/2/1343> [received February 14 2019]

[11] Ratcliffe, Michael. *Three Dollar EC - PPM Meter [Arduino]*, [online] Hackaday.io 9 April 2015 Available at <https://hackaday.io/project/7008-fly-wars-a-hackers-solution-to-world-hunger/log/24646-three-dollar-ec-ppm-meter-arduino?fbclid=IwAR2EGrnjFIaaTx7wato6vD8fSn7QzYYs3TPmqfmd82f1Ejhcvw44uDpN8Rs> [received February 14 2019]

[12] Down, Randy D. and Lehr, Jay H. *Environmental Instrumentation and Analysis Handbook*, John Wiley & sons, inc, 2004

[13] Radiometer Analytical SAS. *Conductivity Theory and Practice*, France, 2004, p. 20

[14] Weller, Thomas. *The Best of The Growing Edge International*, New moon publishing, 2005, p. 108

[15] Treutel, Chuck. *Peristaltic Pumps: A Simple Solution for Abrasive and Corrosive Chemicals.* [online] Watson-Marlow Pumps Group, Available at <https://www.chemicalprocessing.com/assets/wp_downloads/pdf/Watson_Marlow_090507.pdf> [received March 15 2019]

[16] Gray, Nicholas. *”Water Science and Technology: An Introduction”* CRS press, 4th edition, 2017.

[17] Espiritu, Kevin. *The Best Air Pumps for Hydroponics*, [online] Epic Gardening 5 September Available at <https://www.epicgardening.com/best-air-pump-for-hydroponics/>[received March 27 2019]

[18] Pratim Ray, Partha. ”A survey of IoT cloud platforms”, *Future Computing and Informatics Journal*, vol. 1, no.1-2, 2016, pp. 35-36

[19] Arduino. *Analog Input Pins*, [online] Arduino.cc Avalable at <https://www.arduino.cc/en/Tutorial/AnalogInputPins> [received March 26 2019]

[20] Arduino. *digitalWrite*, [online] Arduino.cc Avalable at <https://www.arduino.cc/reference/en/language/functions/digital-io/digitalwrite/> [received March 26 2019]

[21] Arduino. *Serial*, [online] Arduino.cc Avalable at <https://www.arduino.cc/reference/en/language/functions/communication/serial/> [received March 26 2019]

[22] DF Robot. *Industrial pH Electrode - Armor Casing*, [online] dr-robot.com Available at <https://www.dfrobot.com/product-1074.html> [received May 2 2019]

[23] Hopper, Eric. *Top 4 Reasons Why Your Garden's pH is Out of whack*, [online] Maximum Yield June 1 2017, Available at <https://www.maximumyield.com/the-top-4-reasons-why-your-gardens-ph-is-out-of-whack/2/3240> [received May 30 2019]

Appendix A

Images

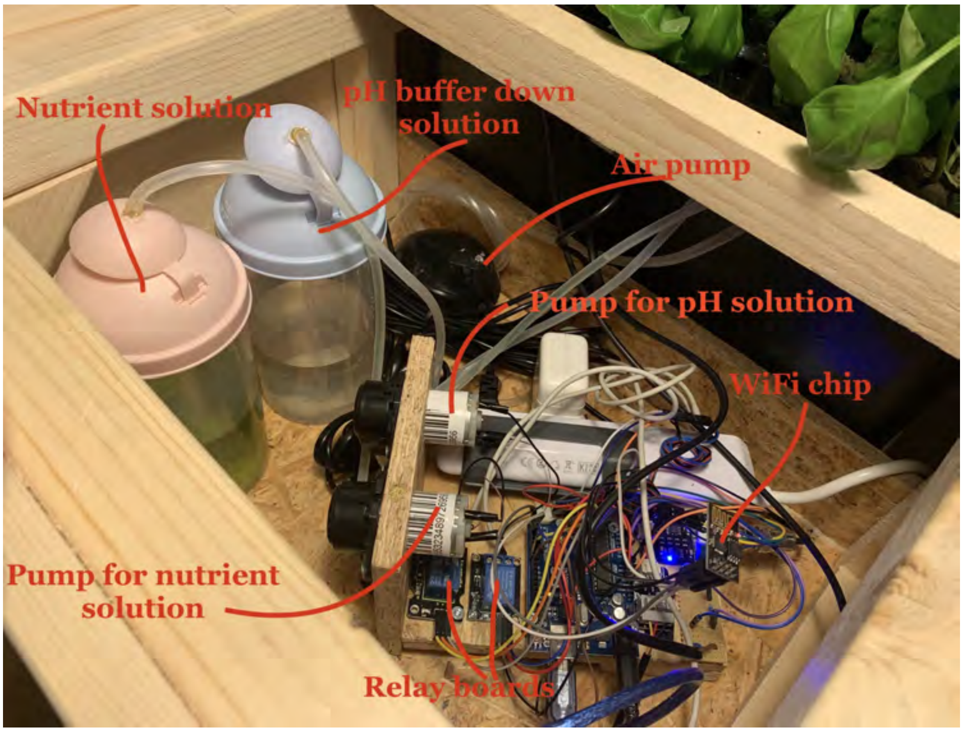


Image of the electronics

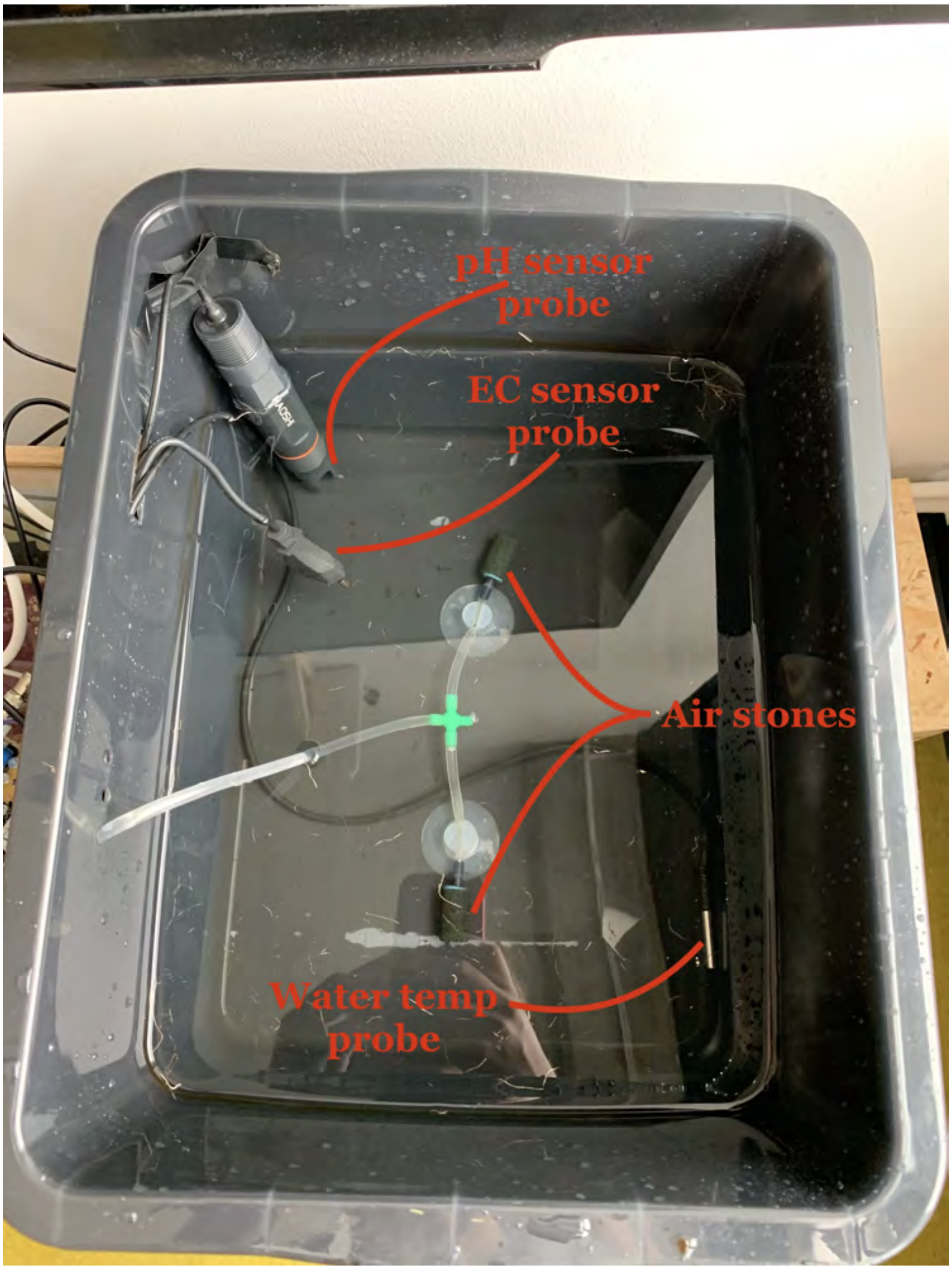


Image of the sensor

Side view of the system

Overhead view of the plants and growing tray

Appendix B

Product manual of pH probe

Introduction

- This pH combination electrode is made of sensitive glass membrane with low impedance.It can be used in a variety of PH measurements with fast response, good thermal stability.It has the good reproducibility, difficult to hydrolysis, and basically eliminate the alkali error .In 0 to 14pH range,the output voltage of the electrode is linear.The reference system which consist of the Ag/AgCl gel electrolyte salt bridge has a stable half-cell potential and excellent anti-pollution performance.The ring PTFE membrane is not easy to be clogged, so the electrode is suitable for long-term online detection.
- **This product is only a part. Its usage is same with the pH meter (SKU:SEN0161), and the Industrial one is suitable for long-term monitoring.**

Applications

The electrode is suitable for all kinds of printing, circuit board factory, wastewater containing chromium and other industrial and domestic sewage pH monitoring.

Specification

- Length (with protective cover)：17.7cm
- Diameter：2.74cm
- Wire Length：5m
- Connector：BNC
- Measuring Range：0-14pH
- Measuring Precision: ≦0.02pH
- Suitable Temperature: 0-60°C
- Response Time:10sec
- Drift：≦0.02PH/24hours
- Resistance of Sensitive Membrane: ≦200*10^6Ω
- Slope: ≧95 %
- Electrode's Equipotential Point:7±0.5PH

pH Electrode Characteristics

The output of the pH electrode is Millivolts,and the relationship between pH value and output voltage is shown as follows (25 °C):

VOLTAGE (mV)	pH value	VOLTAGE (mV)	pH value
414.12	0.00	-414.12	14.00
354.96	1.00	-354.96	13.00
295.80	2.00	-295.80	12.00
236.64	3.00	-236.64	11.00
177.48	4.00	-177.48	10.00
118.32	5.00	-118.32	9.00
59.16	6.00	-59.16	8.00
0.00	7.00	0.00	7.00

Dimensional Drawing

Use and Maintenance

- Electrode must be known before the measurement standard buffer solution PH value calibrated to improve the measurement accuracy , PH value of the buffer to be reliable, and the closer the measured value , the better , generally no more than three PH.
- Sensitive glass bubble ball electrode tip is not in contact with a hard object , any breakage and hair will rub the electrode failure.
- Electrode receptacle must be kept clean and dry height , if tarnished available medical cotton and ethanol to wipe and dry , definitely prevent the output ends of a short circuit , otherwise it will lead to inaccurate measurements or failure.
- Measurements should be taken before the bubble inside the glass bulb rejection to , otherwise it will cause measurement errors , measurement, should stir the electrode in the test solution still placed to accelerate response.
- Both before and after measurements and clean the electrode with deionized water to ensure measurement accuracy , the viscosity was measured in a sample , the electrodes need to use deionized water to remove the solvent.
- Passivated electrode will produce long-term use , the phenomenon is sensitive to low gradient , slow response , inaccurate readings , then you can lower end of the electrode bulb with 0.1M solution soak for 24 hours (preparation 0.1M dilute hydrochloric acid : 9ml hydrochloride diluted with distilled water to 1000ml), then use 3Mkcl solution soak for several hours to restore performance.
- Glass bulb liquid surface contamination or blockage , but also make the electrode passivation , at this time , should be based on the nature of the pollutant , the appropriate cleaning solution.
- Electrode cycle is about a year, after aging should be promptly replaced with new electrodes.

Appendix C

Program code

```
/*
 * University :Royal Institute  of Technology
 * KTHCourse :MF133X,  Degree Project  in Mechatronics
 * TRITA number :TRITA-ITM-EX 2019:61
 * Authors : Jens Ortner and Erik Ågren
 * Name of the program : CMAST16
 * Name of the project :  Autonomated Hydroponics
 * Finalized :2019-05-29
 *
 *
 * This code is used to control an Arduino Uno that has three
 * sensor connected to it:
 * 1) pH sensor.
 * 2) EC psensor.
 * 3) Water temperature sensor.
 * The Arduino also controls two 12V liquid pumps via two
 * separete relay boards.
 * The main idea of the setup is to make the feeding and controlling of
 * a hydroponics system. The code is designed to automaticly take pH, water
temp and EC
 * of a water solvent and add pH down buffer solution and/or nutrient solution
 * to the water solvent through one of the two pumps.
 *
 * The readings from the different sensor reading is uploaded every loop
iteration
 * to a Thinkspeak IoT channel via a WiFi chip.
 *
 *
 */

 //____________________________________________
 /* Here will the libraries needed are included

#include <OneWire.h>
#include <stdlib.h>


/* Below all the parameters and data types needed will be defined
 *
 *
 */

// Initializing pin for temperature sensor

int DS18S20_Pin = 2; // Signal pin on digital 2 on the Arduino Uno
OneWire ds(DS18S20_Pin);  // Creating a OneWire object

//Declaring Variables for temperature sensor
```

```
float TemperatureSum; // The avrage of a samples taken from one temp test will
be stored in this variable
byte data[12]; // Variable to temporary hold readings
byte addr[8]; // Variable to temporary hold the memory adress of the readings

// Initializing pins for the relay boards controlling the pumps

const int in_1 = 7 ; // Declaring pump to digital port 7
const int in_2 = 8 ; // Declaring pump to digital port 8

// Initilizing pins for EC sensor

const int ECPin= A1; // Assigning analog port 1 reference pin
const int ECGround = A2; // Assigning analog port 2 as ground level
const int ECPower = A3; // Assigning analog port 3 as power pin

//Declaring variables for EC sensor

int R1= 1000; // Declaring the internal resistance to variable R1
int Ra=25; // Declaring the powering pin resistance to variable Ra

float EC=0; // Declaring variable for EC-value
float EC25 =0; // Declaring variable for the equivialent EC-value at temp 25
degree celcius
float K=1.44; // trail measured coefficient
float TemperatureCoef = 0.019; // Temperature coefficient for water that is
estimated by trials.

float raw= 0; // The raw data from a EC-sensor reading is declared to this
variable
float Vin= 5; // The internal voltage supply from Arduino Uno to a analog pin
float Vdrop= 0; // The voltage drop measured from a EC-reading will be store
here
float Rc= 0; // The voltage of the water solvent

// Initilizing pin for pH sensor

const int pHpin = A0; // Declaring analog port 0 to the pH-sensor probe

//Declaring Variables for pH sensor

unsigned long int avgValue;  //Store the average value of the sensor feedback
float phValue; // Storing the calculated pH reading in assinged to this
variable
int buf[10],temp; // Temporary variable used to sort the 10 pH reading samples

// WiFi network setup
```

```
#define SSID "A WiFi network"     // The name of the WiFI network to connect
to is entered here
#define PASS "password"      // The password of the WiFI network to connect to
is entered here

//ThingSpeak channel setup

#define IP "184.106.153.149"// The ip-adress for thingspeak.com
String msg = "GET /update?key=94IADK04DP5YY184"; //A GET parameter associated
with the personal thingspeak channel
//____________________________________

/*
* In this section the setup for the pins are defined
*
*/

void setup(){

  Serial.begin(115200); // Setting the baudrate


  pinMode(in_1,OUTPUT); // Setting pin modes for the relay board controlling
the pump feeding pH down buffer solution

  pinMode(in_2,OUTPUT); // Setting pin modes for the relay board controlling
the pump feeding nutrient solution

  pinMode(ECPin,INPUT); // Setting pin mode for the input pin for the EC
sensor probe
  pinMode(ECPower,OUTPUT); // Setting pin mode for sourcing current
  pinMode(ECGround,OUTPUT); // Setting pin mode for sinking current
  digitalWrite(ECGround,LOW); // Setting ground level for the EC sensor probe
  R1=(R1+Ra);// Taking into account Powering Pin Resitance

  Serial.println("AT"); //Hayes command call for attention

  delay(5000); // 5000 ms delay

  connectWiFi(); //Calling on function to connect to WiFI
}

//__________________________________________

/*
* This is the main loop that will operate the sensor readings, control the
pumps, and call on
```

```
* the funtion to upload data to the ThingSpeak channel
*
*/

void loop(){

  //_____Checking water temperature___________

  if ( !ds.search(addr)) {
      //no more sensors on chain, reset search
      ds.reset_search();
      return -1000;
  }

  ds.reset(); // Reset the 1-wire bus
  ds.select(addr); // Selecting the adress of the device to operate
  ds.write(0x44,1); // Write a byte, and leave power applied to the 1 wire bus.

  byte present = ds.reset();
  ds.select(addr); // Selecting the adress of the device to operate
  ds.write(0xBE); // write to the temperature sensors RAM at this adress


  for (int i = 0; i < 9; i++) { // we need 9 bytes
    data[i] = ds.read(); // Reading the nine bytes stored in the temperature
sensors RAM
  }

  ds.reset_search(); //resetting the OneWire device search

  byte MSB = data[1]; //storing the temp reading
  byte LSB = data[0]; //storing the temp reading

  float tempRead = ((MSB << 8) | LSB); //using two's compliment
  TemperatureSum = tempRead / 16; // converting from hexa

 //_____Checking pH-value______________


  for(int i=0;i<10;i++)       //Get 10 sample value from the sensor for smooth
the value
  {
    buf[i]=analogRead(pHpin); //Taking ten analog readings from the pH probe

    delay(10); // 10 ms delay
  }

  for(int i=0;i<9;i++)        //sort the 10 analog readings from small to large
```

```
  {

    for(int j=i+1;j<10;j++)

    {

      if(buf[i]>buf[j])
      {
        temp=buf[i];
        buf[i]=buf[j];
        buf[j]=temp;
      }
    }
  }

  avgValue=0; // a temporary parameter for the pH readings

  for(int i=2;i<8;i++){  //take value of the 6 center samples
    avgValue+=buf[i];
  }

  avgValue=avgValue/6; //take avrage value of the 6 center samples

  phValue=(avgValue*5.0)/1024; //convert the analog readings into volt

  phValue=3.157895*phValue+0.9152632; //convert the millivolt into pH value.
Floats are coeffiocients from sensor calibration


 //______Checking EC level_______________

  // Estimates Resistance of Liquid

  digitalWrite(ECPower,HIGH); //Setting the power pin for EC sensor to high
  raw= analogRead(ECPin);
  raw= analogRead(ECPin);// First reading will be inconclusive due to low
charge in probe
  digitalWrite(ECPower,LOW); //Setting the power pin for EC sensor to low


  // Convert voltage to EC
  Vdrop= (Vin*raw)/1024.0; // The voltage drop measured
  Rc=(Vdrop*R1)/(Vin-Vdrop); // The resistance of the water solvent
  Rc=Rc-Ra; //acounting for Digital Pin Resitance
  EC = 1000/(Rc*K); // The calculate EC value


  // Compensating For the temperature in the water solvent//
```

```
  EC25  =  EC/ (1+ TemperatureCoef*(TemperatureSum-25.0)); //

  updateTS(); //Calling on funtion to update the ThingSpeak channel with new
data



  //_________DC motor control pumps___________



   if (EC25<1.4 && phValue>6.6) {   //if the nutrient level unsufficient, and
the pH value to high

         digitalWrite(in_1,HIGH) ; //Sending signal to relay to power up
nutrient solution pump
         delay(2000); // Running pump for 2 seconds
          digitalWrite(in_1,LOW) ; //Cutting power to pump
   }
   else if (EC25>1.4 && phValue>6.6){  //if the nutrient level sufficient and
the pH value to high

         digitalWrite(in_2,HIGH) ; //Sending signal to relay to power up pH
down buffer solution pump
         delay(2000); // Running pump for 2 seconds
          digitalWrite(in_2,LOW) ; //Cutting power to pump
   }
   else if (EC25<1.4 && phValue<6.6) {   //if the nutrient level sufficient
and the pH value sufficient

     digitalWrite(in_1,HIGH) ;//Sending signal to relay to power up nutrient
solution pump
     delay(2000); // Running pump for 2 seconds
     digitalWrite(in_1,LOW) ; //Cutting power to pump
  }

  delay(60000); //Wait 100 second before starting next loop iteration
}
  //______ThingSpeak channel update function___________
void updateTS(){

  String cmd = "AT+CIPSTART=\"TCP\",\""; // A serial command to intruct the
WiFi chip
  cmd += IP; //Adding the string containing the IP for ThingSpeak
  cmd += "\",80"; // The port to communicate with ThingSpeak through

  Serial.println(cmd); //Establishing connection with ThingSpeak

  delay(2000); // time delay
```

Appendix D

EC-meter calibration code

```
/*
 * University :Royal Institute  of Technology
 * KTHCourse :MF133X,  Degree Project  in Mechatronics
 * TRITA number :TRITA-ITM-EX 2019:61
 * Authors : Jens Ortner and Erik Ågren
 * Name of the program : CMAST16
 * Name of the project :  Autonomated Hydroponics
 * Finalized :2019-05-29
 *
 *
 * This code is used to control calibrate an EC meter with a
 * EC calibration fluid. The calibration fluid ha EC value at 1.413
[mSiemens*cm^-1]
 * at 25 degrees celcius
 */


/Including needed libraries

#include <OneWire.h>
#include <stdlib.h>






//______________________Defined Variables
______________________________________//



// Initilizing pins for EC sensor

const int ECPin= A1; // Assigning analog port 1 reference pin
const int ECGround = A2; // Assigning analog port 2 as ground level
const int ECPower = A3; // Assigning analog port 3 as power pin

// Defining varibles for EC-sensor

float CalibrationEC=1.413; //EC value of Calibration solution
float TemperatureCoef = 0.019; // Standard coefficient value for water
temperature compensation at 25 degrees
int R1= 1000; //Resistance of external resistor for
int Ra=25; //Resistance of powering Pins

float raw= 0; // The raw data from a EC-sensor reading is declared to this
variable
```

```
float Vin= 5; // The internal voltage supply from Arduino Uno to a analog pin
float Vdrop= 0; // The voltage drop measured from a EC-reading will be store
here
float Rc= 0; // The voltage of the water solvent
float K=0; // Calibration coefficient to be measured
float EC=0; // Declaring variable for EC-value

// Initilizing pins for temp sensor

int DS18S20_Pin = 2; // Dignal pin on digital 2 for temperature sensor
OneWire ds(DS18S20_Pin);  // Creating a OneWire object on digital pin 2 for
temperature sensor


//Declaring Variables for temperature sensor

float TemperatureSum; // The avrage of a samples taken from one temp test will
be stored in this variable
byte data[12]; // Variable to temporary hold readings
byte addr[8]; // Variable to temporary hold the memory adress of the readings


int ppm =0;

int i=0;
float buffer=0;


//________________________Setup__________________//

void setup()
{
  Serial.begin(9600); // Setting the baudrate

  pinMode(ECPin,INPUT); // Setting pin mode for the input pin for the EC
sensor probe
  pinMode(ECPower,OUTPUT); // Setting pin mode for sourcing current
  pinMode(ECGround,OUTPUT); // Setting pin mode for sinking current
  digitalWrite(ECGround,LOW); // Setting ground level for the EC sensor probe
  R1=(R1+Ra);// Taking into account Powering Pin Resitance


  delay(100);// gives sensor time to settlee



};
```

```
//_______________________Main Loop___________________//

/*
* This is the main loop that will operate the sensor readings
*/

void loop()
{



  i=1;
  buffer=0;

  //___________Checking water temperature___________

  if ( !ds.search(addr)) {
      //no more sensors on chain, reset search
      ds.reset_search();
      return -1000;
  }

  ds.reset(); // Reset the 1-wire bus
  ds.select(addr); // Selecting the adress of the device to operate
  ds.write(0x44,1); // Write a byte, and leave power applied to the 1 wire bus.

  byte present = ds.reset();
  ds.select(addr); // Selecting the adress of the device to operate
  ds.write(0xBE); // write to the temperature sensors RAM at this adress


  for (int i = 0; i < 9; i++) { // we need 9 bytes
    data[i] = ds.read(); // Reading the nine bytes stored in the temperature
sensors RAM
  }

  ds.reset_search(); //resetting the OneWire device search

  byte MSB = data[1]; //storing the temp reading
  byte LSB = data[0]; //storing the temp reading

  float tempRead = ((MSB << 8) | LSB); //using two's compliment
  TemperatureSum = tempRead / 16; // converting from hexa


//______________Estimating Resistance of water solvent___________//

while(i<=10){ //Taking 10 samples
```

```
digitalWrite(ECPower,HIGH); //Sending power to EC probe
raw= analogRead(ECPin);
raw= analogRead(ECPin);// Reading the analog input from EC probe
digitalWrite(ECPower,LOW); //Setting power back to 0 V
buffer=buffer+raw; // Gathering all the samples
i++;
delay(5000); //Adding delay in between the sample readings

};
raw=(buffer/10); //taking the mean of the samples



//___________Compensating EC For Temperaure____________________//

EC =CalibrationEC*(1+(TemperatureCoef*(TemperatureSum-25.0))) ;

//__________________Calculates Resistance relating to Calibration
fluid________________//

Vdrop= (((Vin)*(raw))/1024.0); // The voltage drop measured in the calibration
fluid
Rc=(Vdrop*R1)/(Vin-Vdrop); // The resistance in the calibration fluid
Rc=Rc-Ra; // Taking internal resistance into account


K= 1000/(Rc*EC); // The calibration coefficient for our EC probe
```

TRITA -ITM-EX 2019:61