The University of Texas at Tyler

College of Engineering and Computer Science

Tyler, TX 75799

Final Design Report

For

IoT Leaf Patch

A design project to fulfill the requirements of Senior Design in the Departments of Electrical Engineering at The University of Texas at Tyler

The individuals whose names and signatures appear below certify that the narrative, diagrams, figures, tables, calculations, and analyses contained within this document are their original work except as otherwise cited.

Jared Carter	Signature	01-19-2022 Date
Karl Lorenz	Signature	01-19-2022 Date
Carlos Galvan	Signature	01-19-2022 Date
Rudy Montiel	Signature	01-19-2022 Date

Table of Contents

1. Project Description 1

2. Design Specifications 3

3. Design Solutions 10

4. Prototype Design and Fabrication 17

5. Testing and Validation 24

6. Broader Impacts of the Project 30

7. Conclusions 32

8. References 33

Appendices 34

Tables

Table 1 - Design Priorities 4
Table 2 - Evaluation Basis 6
Table 3 - Evaluation Results 8
Table 4 - Cloud Systems 9
Table 5 - Coding Languages 9
Table 6 - Hardware Components 14
Table 7 - Software Required 15
Table 8 - Bill of Materials 21
Table 9 - Equipment Needed 22
Table 10 - Shop-Man Hours 22
Table 11 - Testing Results 29

Figures

Figure 1 - Product Block Diagram....................10
Figure 2 - Multisim Simulation....................12
Figure 3 - 555 Timer Producing a 1kHz 5V Clock Signal....................13
Figure 4 - Code Block Diagram....................13
Figure 5 - Fusion 360 Drawing....................17
Figure 6 - Square Wave Produced by the 555 Timer....................21
Figure 7 - Voltage Booster/555 Timer Circuit....................25
Figure 8 - Oscilloscope Results....................26
Figure 9 - IoT Cloud System with Grafana....................27
Figure 10 - IoT Cloud System Raw Data....................27
Figure 11 - IoT Alert System....................28

1. Project Description

Methods for monitoring the health of plants are rare despite being exceptionally useful for things like growing crops or keeping that houseplant alive. Traditionally, the only way to measure the hormone levels in a plant is to take tissue samples, which could damage the plant, and then send them off to a lab to be analyzed. This creates a delay between the sampling of the tissue and the results coming back that could potentially be so large that the results are rendered irrelevant, or the plant died in the interim. Tissue sample analysis is valid for disease study and other things that further scientific knowledge but is not efficient for situations where the grower wishes to determine which plants are need tending. Moreover, to maximize growth and the potential of plants and crops, it may be advisable to monitor plant health in real time and provide an instant response to change.

There is already a current, or relatively recent, method to monitor plant health that achieves this goal. This is not a commercial product, but a proof of concept. Researchers from North Carolina State University have developed a leaf patch that can measure up to 13 different volatile organic compounds which are produced from various biological plant stress responses **[1]**. However, this method is more advanced than the average person might need: these compounds mean nothing to non-biologists and could become a source of confusion rather than help. Also, the NCSU patch lacks several key features. It has no Internet of Things (IoT) capability, which means no cloud storage or user-end applications and is more difficult to manufacture than the design proposed in this document as it requires the use of silver nanowires and ligand chemical bonding. These components also limit the durability and lifespan of the product, while also requiring a replacement of the patch rather than simply repairing it.

Therefore, the product goal is to create a non-damaging, non-invasive method to monitor plant health in real time. The most reliable indicator for plant health is the pH levels inside the leaves and, more specifically, inside each leaf’s xylem layer. Multiple plant stress indicators, such as salicylic acid, abscisic acid, amino acids, are acidic in nature, though other indicators are more basic. All these compounds can be found in the xylem layer and their concentrations regularly change as the plant detects or fixes problems, meaning the pH level is correspondingly affected. Building a device to measure pH levels is easier and gives more easily understood results than attempting to create something to measure one or multiple specific stress indicators. Ideally, these measurements would also be accomplished through a leaf patch, where small electrodes penetrate a plant leaf and provide electro-chemical responses to pH level changes. These small voltages can be stepped up through a voltage booster to provide a reasonable signal range and voltage strength. Using a microcontroller to analyze, sort, and upload the data through the IoT to some form of cloud storage allows one or more users to monitor that data whenever and wherever they wish. This data can be made even easier to understand by creating an end-user application that pulls the raw data from the cloud and visualizes it. For this product, the prototype will be tested by placing the patch on the leaf of a Maize plant, though the concept can be expanded for any plant by tweaking the programming of the microcontroller.

2. Design Specifications

This product contains 6 major components, and each part has its own requirements. The first component is the leaf patch. This patch houses two electrodes, one working/cathode and one reference/anode, which are themselves formed of an array of microneedles. Both are coated in the respective positive and negative conductive materials to read the pH level and generate a corresponding voltage signal. Since these specific voltage levels are relatively small, the signal strength needs to be increased by sending it through a voltage booster. This voltage booster is an intermediary device between the patch and the microcontroller, increasing the DC signal strength from the plant to an appropriate range for the microcontroller. Then, this microcontroller can analyze and process the incoming data and, through correct programming, extrapolate any incoming DC voltage to its paired pH level. Having done this, the microcontroller, having IoT capability, uploads the data to a cloud server. The server holds all past data and alerts the user. This data can be visualized by creating a user-end application to translate the data and display it in real time.

The amount of data stored and applied on the cloud system can be determined by the user through programming the system and specifying how many samples are plotted per second. There is a drop-down menu that can plot data from 10 seconds to 10 minutes. The microcontrollers stored program reads at 1 second intervals, meaning the software is continuously running, and the table on the cloud system is only storing the user's desired number of samples. This is the best option to get accurate results and for the system to function properly while the user does not have to fix anything on the device. The user can also set alerts to their desired method by clicking a drop-down menu and setting up how they want to receive those notifications; options include email or through the cloud system. These alerts can be programmed in a simple "If (sample) reaches (below or above) (fixed value) then send alert to (preferred notification)". This would allow the user to have full control of the data they receive. A more detailed explanation is in **Section 3**.

Precision of the system is determined by two things. First, how well the microneedle is constructed. The more flawless the patch creation process is, the higher the precision will be. Second, how much noise the signal carries. Of course, the plant will not produce a noiseless signal, but running the output through a voltage booster amplifies the noise already in the signal. It is expected that the pH level will be accurate to the tenth at the minimum. The pH range it detects depends on the calibration process, but being that plant pH is slightly acidic, the measured range will be approximately pH 3-9. The above-mentioned MCU and cloud system records all data points measured by the patch; the patch can take a measurement at any time interval desired by changing the MCU code. It is likely that the most optimal time interval is between 15 minutes and 1 hour, though any interval greater than 2 minutes is acceptable. The pH of a plant can change within 2 minutes, but only large changes need attention, and these changes take longer to appear. For example, a plant that is not being watered may take a day or two before a dangerous pH is reached.

Below, Table 1 provides more in-depth detail for each of the 6 major components

Table 1. Design Priorities

Table 1 - Design Priorities

Microneedle Patch	An array of microneedles on the patch’s base layer allows the electrodes to receive voltage readings from the plant to transfer to the microcontroller ports. These values will be small, necessitating a voltage booster. • Measurements are determined by leaf depth to ensure no full punctures occur. • Different chemical coatings are needed as a conductor to ensure accurate results. • Needles need to be durable to ensure they don’t break and hurt the plant. Strength test required.
Voltage Booster	This is a compact circuit that is soldered onto a prototyping board to accept low input voltages from the plant and raise them to an acceptable range for use with the microcontroller. • Plants generally have a max output less than 0.5 volts; MCU shows accurate results with higher voltages (around 2-5V). • The device needs to be tested on electronics instruments to ensure desirable output values before creating final design. • Electronics parts will need to be reliable to withstand various climates.
Microcontroller	This is the brain of the operation, used for powering devices and uploading data through built-in Wi-Fi capability. It needs to be durable, compatible with various power sources, compact, and portable. • Battery with enough micro-ampere-hours (mAH) needed to last all week. Most 2,000mAH batteries have a lifespan of 100 hours, but since microcontroller runs on minimum 3V and 0.5 Amps, the battery will last longer.

<table>
<tr><td>Programming</td><td>The microcontroller can be programmed using basic C to read analog input data, convert it to pH levels through function-mapping, and upload it through Wi-Fi.
Preparatory coding for cloud software is incorporated to allow all readings to be saved with timestamps for the user.
Declaration of values is priority since the only native function needed from the microcontroller is reading from the ports.
Performance must always work properly; if the programing fails, the data will stay on the microcontroller and not be uploaded to the cloud.</td></tr>
<tr><td>Cloud Storage</td><td>IoT storage needs to save the real time data from the MCU for safekeeping. This also helps the application display information over time to the user such that they can watch the plant health in response to new materials like fertilizers and vitamins.
The cloud service needs a hefty storage size due to the large amount of data that will be recorded from the patch.
It must perform contiguously with application to allow the user to change settings and continue to see values from previous data.</td></tr>
<tr><td>Application</td><td>This visualizes the data with well-organized graphs and allows the user to view results with labeled data points and warnings about the plant health, such as nutritional or water needs.
Data should be viewable on both mobile and computer devices, allowing the user to have an infinite spectrum of visibility.
Since it does run on a mobile application, performance must be optimized so that there will be no crashes or late uploads of data, even if the user accesses their mobile internet or Wi-Fi.
This app needs to be programmed to visualize the raw data, display history, and give meaningful feedback.</td></tr>
</table>

Table 2. Evaluation Basis

Table 2 - Evaluation Basis

Public Health	Product should adhere to Arduino voltage restrictions (min volts: max volts) and limitations, to prevent volatile behavior of fire or damage onto device. *NSPE Code of Ethics for Engineers [9], Section III Item 2-B: engineers “shall not complete, sign, or seal plans and/or specifications that are not in conformity with applicable engineering standards”. Leading to proceed with an unsafe product can harm or endanger a person’s life which is unethical and unacceptable. This project will adhere to the manufacturer specifications for the Arduino circuit and design all other components in accordance with the Arduino voltage level and not be volatile if left alone, damaged, or touched.
Safety and Welfare	A safe workplace and habitat for the person that designed, future maintenance personnel, engineers that work on updating the product or anyone who may come near it. *NSPE Code of Ethics for Engineers [9] mentions in Paragraph 7 Section A that an Engineer must not attempt to injure or discredit other Engineers' work. Relating it to this project: this design needs to be safe for all engineers working around you and the safety of the average consumer, by testing and correcting any errors with sufficient tests. Following this ethical consideration is taken in account when designing this product to ensure the safety of everyone.
Global Factors	Globally improvement is achieved by the efficiency of this product. Improving crop productivity per square footage area of production, can be improved by being able to detect problems sooner in your crops. By doing so there will be a detection of problems far sooner than regular that can enhance productivity. *NSPE Code of Ethics for Engineers [9], Section III Item 2-D: this point encourages engineers to “adhere to the principles of sustainable development”. With the constant sensing of crops, threats can be caught earlier and be dealt with sooner to prevent crop loss. This follows sustainable development as higher yield percentage means less land is required for the same output, which will reduce emissions and pollution.
Cultural Factors	Culture ethnicity who relies heavily on farm production to survive will be impacted heavily in this case. Different cultures require different types of crops for survival base. In example (Corn, Beans, Wheat, Fruits) are all important food bases regarding different cultures. Although some may not have the environment or habitat to grow some of these in with this product attempting to do so will become easier and have a higher probability of success. *NSPE Code of Ethics for Engineers [9], Section III Item 2-A

	states: “Engineers are encouraged to participate in civic affairs; career guidance for youths; and work for the advancement of the safety, health, and well-being of their community”. Knowing how the product affects different type of communities and cultures is important. In order to provide the best benefit to a specific culture that has different views from the rest, it is important to be incorporated in learning the culture first.
Social Factors	Having a product that can be outstanding to the public and specific chain of people in example farmers. Being able to explain your product design in a specific matter and manner about how it operates and all the benefits from it are important. *IEEE - 7.8 Code of Ethics [10] Section I, Paragraph 2 mentions that as an engineer you must explain any emerging questions by individuals and society. With this technology it can be beneficial to locations like farms who have seen droughts and severe damage as it helps to solve an issue that has hurt farmers for generations. But you must be ready to give a detailed explanation if problems arise. Being able to answer in situations shows the public that you can be counted on and will increase further growth in sales and a better reputation of your work ethic.
Environmental Factors	Destruction to the environment is not ideal whenever a new product is fabricated. *NSPE Code of Conduct [9] Section 2, Paragraph D mentions that an engineer must protect the environment for future generations, because the Ecosystem will not be affected or damaged by any electronic components. Harsh effects caused by the device will not alter wildlife preceding or during testing, neither when the device is manufactured. Damaging ecosystems and the environment does not influence the marketing of your product. In this case our design is to make or produce a better environment rather than to harm it.
Economic Factors	The product produced will cause attention from others in parallel fields. To understand the being of staying ethical it is important to stay ethical. A few people will come and try to convince you on how the product can benefit them and you at the same time. Sometimes not all the solution they will give are ethical. *IEEE - 7.8 Code of Ethics [10], Section I Item 4; NSPE Section II Items 5-A and 5-B [9]: engineers shall “avoid bribery” and “deceptive acts” such as offering “any valuable consideration to secure work” or “commission, percentage, or brokerage fee to secure work”. Members of this project will not accept any unfair compensation for their work nor attempt to give unfair compensation to complete the project easier or faster. It is important to know that no matter the offers that come, to consider them and know which offer is the best ethical offer.

Every part of this project has an important effect on the topics listed on table 2. With the information found in that table, the major elements of the project can be evaluated for several alternatives, with each being given a numerical value. Since there are 3 major pieces to the project, Tables 3 and 4 ranks each one on a scale from 1 to 3 (1 being the best choice and 3 being the worst choice) and 1 to 6 in complexity (1 being the simplest and 6 being the most difficult). These values carry multiple weights because each category is broad, and each component can favorably benefit or adversely affect, and sometimes both, individual aspects of each group of factors.

Table 3 - Evaluation Results

Microcontroller	Image	Speed / SRAM	Price	Rank
NANO 33 IoT		48MHz / 32kb	$18.40	3
NANO RP2040 Connect (raspberry-pi processor)		133MHz / 264kb	$25.50	2
HiLetgo ESP-WROOM-32 ESP32 ESP-32S		240MHz / 520kb	$10.99	1

The HiLetgo ESP-32 microcontroller works best for this project because of its small size and fast processing speed for uploading data. All contain the same power sources and use the same programming language, so that factor can be ignored in this table. The price of the HiLetgo ESP-32 is also low, which is an important factor for the consumer as they would get more efficiency for a more affordable price.

Table 4 - Cloud Systems

Cloud System	Price	Retention Days	Storage Capacity	Code	Complexity	Rank
Arduino IoT Cloud	$6.99 per month	3 Months	Unlimited	Arduino/IDE	2	2
Amazon Web Services	$39.00 per month	Depends on Package	25gb	Python	5	3
Grafana	$49.00 per month	13 Months	100gb	Python	4	1

The purpose of the cloud system is to store enough data such that any user can view the pH level as it changes over a substantial amount of time. The Arduino IoT cloud system best fits these criteria with its simplicity of use and unlimited amount of storage, but it contains no complex function. The final product we have chosen is Grafana, for the purpose of being able to send alerts and allow simple user interface dashboards for quick and simple functions.

Table 5 - Coding Languages

Name of Program	Complexity	Rank
Python	3	2
Arduino IDE	2	1
C	5	3

All microcontrollers can be programmed with these 3 coding languages. There are some issues, like the libraries and system requirements needed for C and Python, while the Arduino IDE contains most of the required settings and has simple call functions to allow for a smooth and steady process when coding. This is easily the best choice for this project.

3. Design Solutions

There are 2 sets of classifications into which the final product can be broken. The first set, as shown by the figure below, is a grouping based on the purpose of each component. The purpose of the hardware is to create analog data and make it usable. The purpose of the MCU is to create digital data and upload it. The purpose of the cloud is to create a database by which to read and analyze the original inputs. In the following paragraphs, these goals are also classified into two divisions: hardware and software. The hardware components are detailed in **Section 3.2** and its subsections while the software components are detailed in **Sections 3.3** and **3.4.**

3.1 Product Architecture

Figure 1 - Product Block Diagram

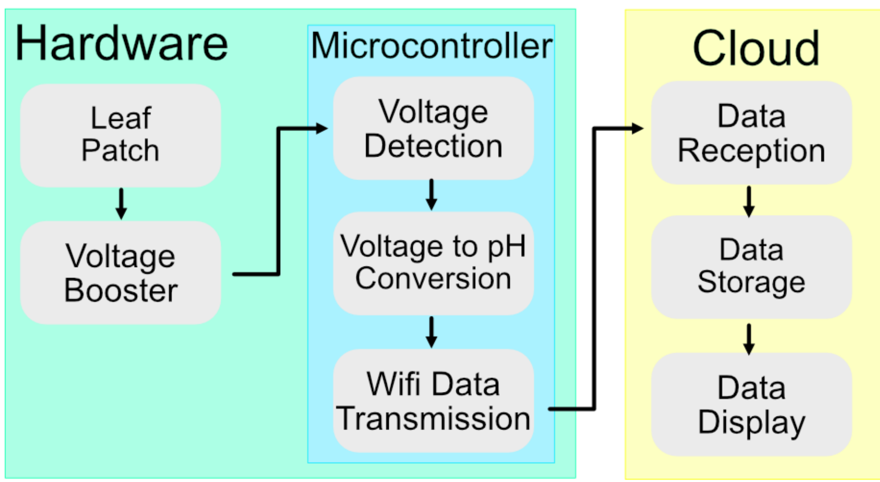


Figure 1 is block diagram representation of the functions that each major component does to achieve the goal of the project. Each level has a step to get to the next step; if an error is detected in any part, then the code in the microcontroller will save the current progress before it stops reading.

3.2 Hardware Subsystems

3.2.1 Microneedle Patch

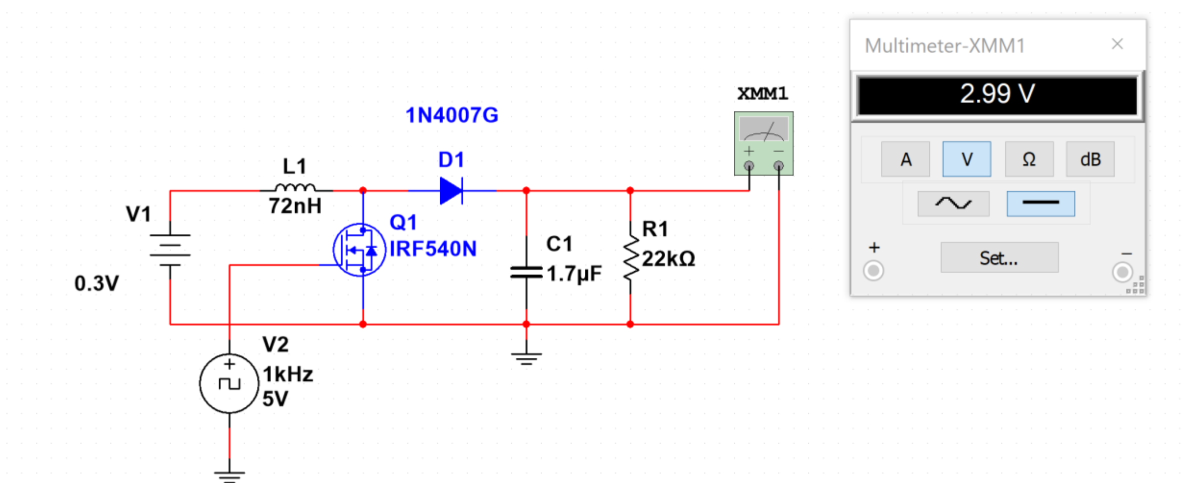
The objective of this component is to, at any given point, determine the instantaneous pH level of the plant to which it is attached. In essence, the microneedle patch can ‘read’ the current pH level of any plant, and these readings need to be requested. Requests can be done manually with a timer or with more sophistication through a microcontroller. When designing a microneedle patch, several key things need to be considered. First, the needles. Each needle needs to be small enough to prevent puncturing the plant from tearing, and from damage in general. Yet, they must also be big enough that they can be 3D printed with consistency in size and shape. Furthermore, the needles can be arranged in arrays to create an electrode of appropriate size such that errors between needles average out. Second, the arrays need a base on which to sit where they can be protected from wind, weather, and any other form of damage. Like the needles themselves, this patch base needs to be small enough and large enough; as small as effectively possible while providing a solid, stable base for the needle arrays. Third, and last, there needs to be room enough on the patch base for wire leads to be secured in such a manner that they won't become loose. The appropriate length depends on the gauge of wire and the flexibility of its insulation. Combining these three is what constitutes a good design, like the one found in the prototype section that follows.

But there is more to consider. The above guidelines only cover designing the microneedle patch model. This model is then 3D printed, after which comes the time to turn models into working prototypes. The needles themselves, being 3D printed, are not electrodes nor do they read pH levels. This is accomplished by coating the needle arrays, as one might a wall to paint it, with special compounds. These compounds depend on the usage situation. Here, the need is to create a pH sensor. For this case, there are 2 electrodes, and each gets a different material. The working electrode gets a coat of graphene and polyaniline, which combine well to create an electrode that generates a small voltage signal in response to pH changes. The reference electrode gets a coat of silver-chloride, which remains stable despite changes in the pH of its environment. Small wires are then attached to each electrode (their connection method is determined by the patch design) whereby the signal generated by the working electrode can be measured or otherwise used.

3.2.2 Voltage Booster

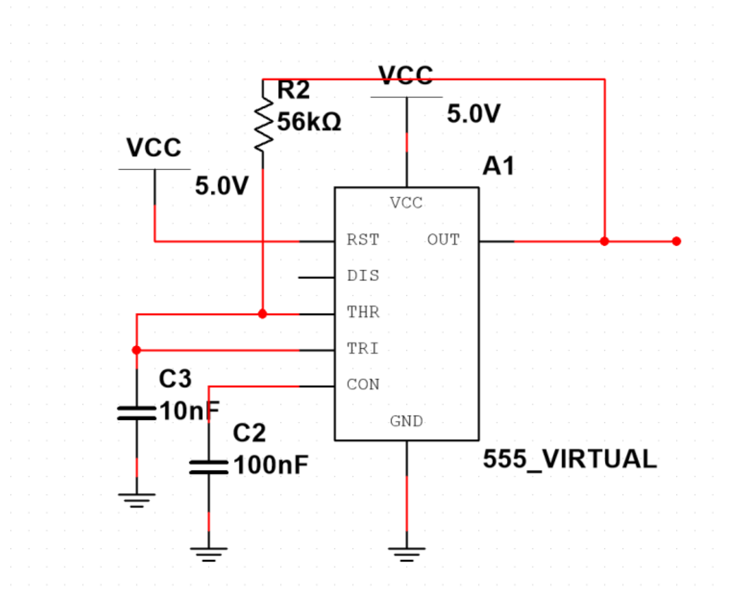
The readings from the microneedle patch (detailed in **section 3.4.1**) produce a small voltage, ranging anywhere from 0 to 0.5 volts. The microcontroller uses the analog port to receive such data, but since the values are a bit too small, a voltage booster is needed to raise the input. The figures below are from a virtual circuit simulator called Multisim that allows for editing, viewing, and testing circuit configurations before building with physical components. The results below come from a basic DC voltage booster design, where a small voltage is stepped up with each clock cycle to some maximum voltage. This maximum voltage is determined by the values of each component. The component values can be calculated knowing the input voltage, the desired output voltage, and doing some circuit analysis. So, the inductor, capacitor, and resistor values can be changed to create a semi-stable DC voltage. These are the components included: a 72nH Inductor, a 1N4007G Diode, a 1.7µF Capacitor, a 22K Ohm resistor, and an IRF540N MOSFET that is powered by a 1kHz 5v power supply. The 0.3V input is a simulation for the output of the microneedle patch.

Figure 2 - Multisim Simulation



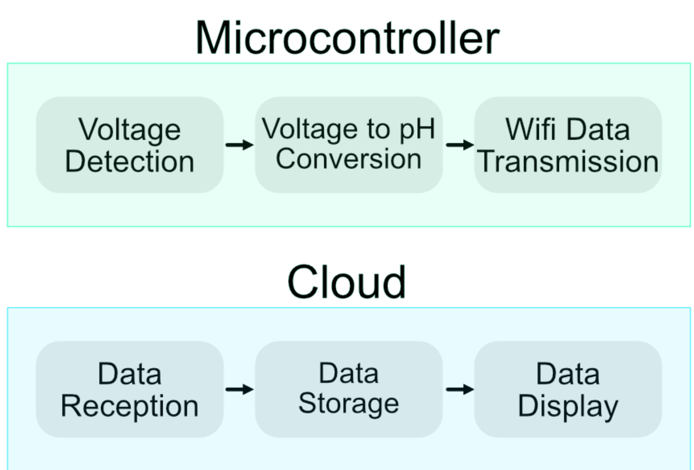
The circuit needs a clock voltage to power the MOSFET, so a 555 Timer can be used to produce a value relatively close to 1K frequency and 5 Volt output. This will be a separate circuit on the PCB Board. Like the voltage booster, the 555 Timer has its clock frequency determined by the value of the resistor and capacitors. These can also be calculated by circuit analysis and a desired frequency.

Figure 3 - 555 Timer Producing a 1kHz 5V Clock Signal



3.3 Software Architecture

Figure 4 - Code Block Diagram



The voltage detection code reads the analog signal from the voltage booster and converts it to a digital value. The analog-digital conversion code is intrinsic to Arduino IDE and is automatically called by the function analogRead(*pin*). Then, a code needs to be written to map the digital voltage data to its corresponding pH value. This can be done by running each reading through an arithmetic formula to extrapolate the original pH values. Finally, a transmission code must be written that will send the pH data from the Arduino over Wi-Fi to a cloud server for storage and display. The cloud server code will receive store and display the pH data to users. For mobile applications, Arduino provides a simple IoT app that can be programmed to do basic visualizations of the data on the cloud.

3.4 Software Modules

3.4.1 Arduino IDE

When compared to Java's script language, Arduino IDE has relatively similar, but simpler, executions. To achieve the goal of receiving voltage and displaying it on an application (Grafana), many of the various code parts must be explained. First, the library needed to connect the microcontroller to the WIFI is installed by downloading it through the Arduino library search function using influxDBClient.h library (). This then provides a template for applying the WIFI network connection information. The code for this is in **Appendix B 8.1**. Then, by using void loops and void setup tabs, this WIFI code can be applied after each microcontroller reset. The code for this is in **Appendix B 8.2**.

The process is then saved online into the IoT Cloud application through specifying the microcontroller's IP address. The results are then displayed on a graph. The complete code can be found in **Appendix B 8.3**

3.5 Hardware Off-The-Shelf

The table below shows the necessary hardware for accomplishing the goal of the project. Parts can be purchased from various websites, but the URLs linked in each section is accompanied by a data sheet for editing purposes.

Table 6 - Hardware Components

Part Name:	Part Number:	URL/Data Sheet:
HiLetgo ESP=WROOM- 32 2,4 Ghz Wi-Fi + Bluetooth Microcontroller	3-01-1287	Data Sheet
ELEGOO 120pcs Multicolored Dupont Wire	EL-CP-004	URL

Electronic Component Assortment 2200 pcs	Ultimate E-Kit	URL

3.6 Software Off-The-Shelf

The table below is a list of off-the-shelf software programs that are needed to accomplish the goal of the project. Each has a use for a certain component of and the reasoning is listed under “License Reasoning”.

Table 7 - Software Required

Software Name:	URL:	License Type:	License Reasoning:
Multisim 14.2	URL	NILM	Circuit Simulation
Fusion 360	URL	Autodesk	Microneedle Sketches
PreForm 3D	URL	GPL	3D Printing Software
Arduino IDE	URL	GPL / LGPL	Programming Microcontroller
Arduino Cloud IoT	URL	GPL / LGPL	Cloud Storage/Application
Fritzing	URL	GPL 3.0	Virtual Sketches

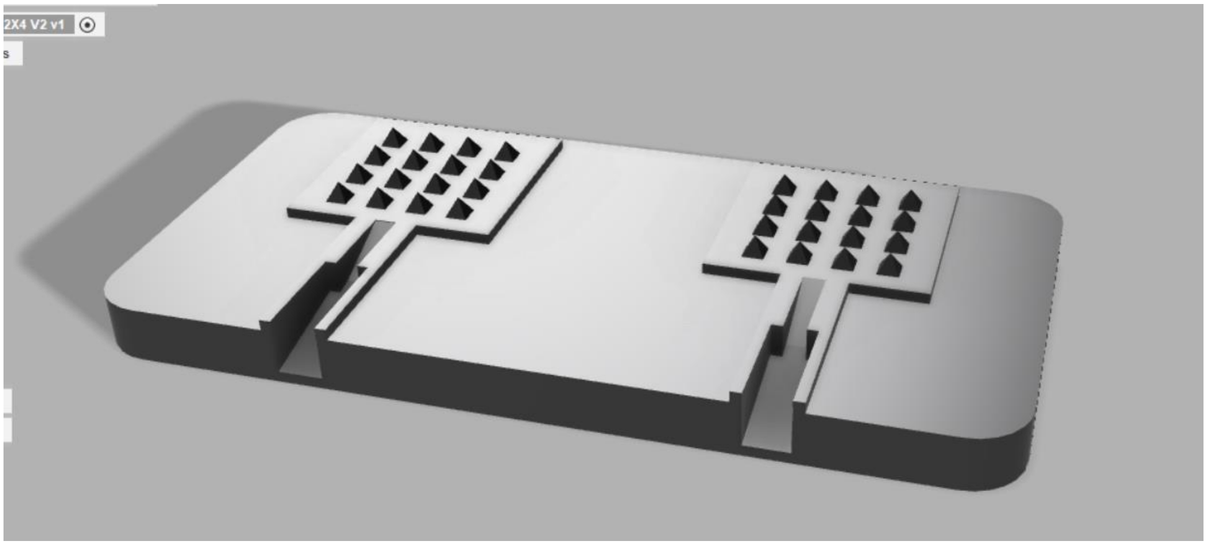
3.7 Schematics/Wiring Diagrams/Mechanical Drawings

The final Design is shown in **appendix B 8.4.** This figure shows all components that were explained in the previous chapters. This drawing is labeled to show how each component connects to the others. The ESP-32 microcontroller, voltage booster with 555 timer, microneedle patch, and battery are shown in their final arrangement. Any component that might need to be fixed or replaced can be easily accessed.

4. Prototype Design and Fabrication

Design of the patch

Figure 5 - Fusion 360 Drawing



The patch base has dimensions of 2cm in length and 4cm in width. The rounded edges shown in the picture above are filleted, or rounded, with a radius of 0.25cm. Sharp edges can potentially puncture the plant or any person handling the patch. Also, rounded edges can help decrease friction should the patch rub against other leaves, namely those closest to the patch but not the one on which it is attached. The needle arrays are on raised pads that are .05cm in height and 1cm x 1cm in base. These pads distribute the overall force being applied to the individual needles to reduce the risk of damage from puncture strain. On top of these pads are the 2 arrays of microneedles, each with 16 in a 4x4 arrangement. From these pads extend out channels, in which wires are embedded, to connect the needle arrays to the microcontroller. These channels have two sections of differing sizes. The larger channel's cross-section is 0.20cm x 0.20cm. The smaller one, connected to the elevated needle base array, has a cross-section of .10cm x .10cm. The lengths of each are not of any great importance. Each section serves a purpose: the thinner channel fits stripped 22-gauge wire, the size used in the Arduino, and the thicker channel fits the same wire while still wrapped in its insulation. The smaller channel ramps up at a 35° angle to the base of the needles because the exposed wire also needs to be coated in the same manner as the working electrode. This provides a solid support and prevents the wire from breaking when the patch is bent, twisted, dropped, or otherwise mishandled.

Fabrication of the patch

Fusion360 is a subset of the AutoCAD software suite and is used for modeling the microneedle patch in 3D. The microneedle patch, once modeled, is then 3D printed in a FormLabs 3D resin printer using a bio-compatible resin. Once the Fusion360 project with the patch model has been exported with the proper file extension (.STL), it can be opened in an application called PreForm. In general, all resin-based 3D printers print by forming layers and stacking them. So, PreForm splices the Fusion360 model into the individual layers used by the 3D printer and determines the order in which they print. Once PreForm finishes organizing the layers and uploading the instructions, the patch is ready for printing. But 3D printing a thin structure, such as a patch, sometimes requires a support structure to prevent any flexing or bending when handling the freshly printed model. A support structure is essentially a flat foundation with many equally spaced pillars to support the actual design. This can be generated in PreForm, rather than trying to create one in Fusion360. A patch of this size, with roughly a $4cm^3$ volume, takes about 40 minutes to print. After printing, the patch must be thoroughly rinsed (think swished in a cup) with IPA (isopropyl alcohol) for about 10 minutes to remove extra resin and smooth the surfaces, as if they were sanded. The curing process forever cements irregularities, so careful washing is necessary. Now that the patch is clean, it can be put into a UV curing chamber for hardening. For this case, each patch is heated at 60°C for 60 minutes. Only after the printed design is cured can it be handled with little risk of damage. Finally, the structural support can be removed by clipping the pillars with wire cutters, and the patch is now ready to be coated.

Coating the microneedles

As shown above, there are 2 arrays of needles forming 2 electrodes. Each electrode of the patch needs to be coated with conductive materials such that a voltage difference is generated. For the working electrode, these materials are graphene and polyaniline. Graphene is an excellent conductor, and polyaniline adheres better to it than other viable conductive materials. Polyaniline is an odd but useful substance: it generates a voltage signal, albeit small, in response to changes in the pH level of its surroundings. For the reference electrode, this material is only Ag-AgCl (silver - silver chloride). Ag-AgCl is a stable compound, at least in relation to its purpose here, as it does not react to changes in the pH level of its environment. Because of their size and the relatively low viscosity of some of these compounds, the coating is done using a pipette. The pipette is filled with the compounds from their respective bottles and dropped evenly over the surface of the electrodes. Ensure that the needles are well coated, especially the needle tips, to minimize measurement error. The channels need to be coated as well: each inner part of the channels will be coated, including the floor base, the side walls, and the ramp leading to the needle array. Take special care when coating the junction between the ramp and needle array base since the corners can be hard to reach. Coating the channel provides an outlet for the signals generated at the needles to be read by the wire leads. Once the coating process is finished, the patch will need to be cured again. This curing happens in a vacuum oven to prevent unwanted particles from creating interference or noise in the voltage signal. Place the product inside the vacuum oven for 30 minutes at boiling, or 212°F/100°C. Note: The product can stay in there while the oven heats up to the required temperature but only start the timer when the oven reaches the specified temperature. When the timer is up, remember to wear oven proof gloves when handling the cured patch.

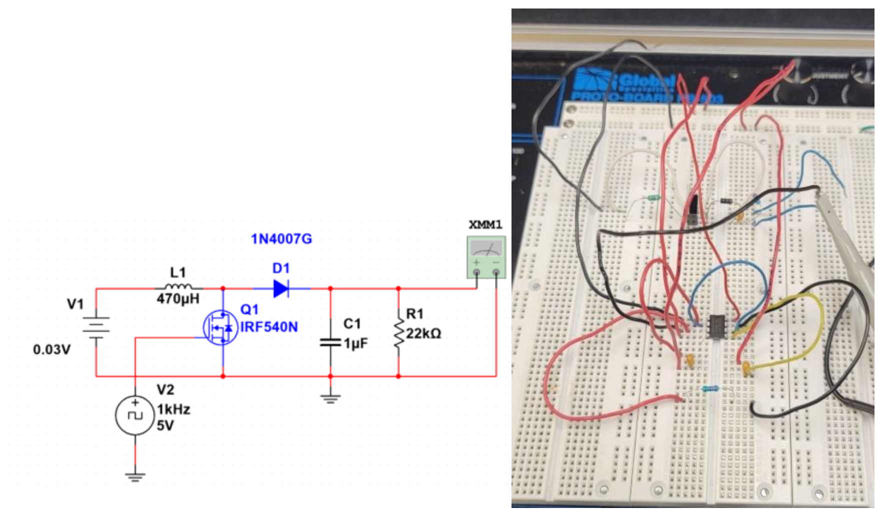
Connecting conducting wires

The wire leads for the electrodes will be Arduino compatible wires, for no reason other than their abundant availability to the team. These wires are 22 gauge (AWS). On one side, the wire will have 0.5cm stripped with wire strippers. Then, the 0.5cm bare wire will be placed in the smaller channel section and the larger channel will house unaltered, insulated wire through its length. The wire will be pushed towards the needle array until the insulated part of the wire contacts the smaller channel section. From there, the exposed wire tip must be flush with the needle base to ensure contact with the conductive coating. Note that failure to properly seat the wire can result in them coming loose or negatively affecting the accuracy of the readings. Thus, for the sake of accuracy, securing the wire in place with glue is not advisable as glue is not conductive and may absorb the tiny voltages produced. So, the wires are attached to the patch using copper conductive tape. This has a negligible effect on the readings; moreover, properly securing the wires prevents the ends from being exposed due to mishandling, wind, or anything else. The copper conductive tape should be cut so that the length of the tape runs parallel with the channels; ensure that the tape covers the channel completely and holds the wire into place. After the tape is applied, the patch is complete and ready to be connected to the voltage booster.

Constructing the voltage booster

Voltage boosters are simple RLC circuits to raise the voltage of any small reading by using frequency step ups. To accomplish this, these are the materials needed: MOSFET, resistors, inductors, and capacitors. By using a 555 Timer, the frequency can be set to a desired value so that the voltage step up can be limited. For this project, the initial voltage reading from the Microneedle patch is 0-0.6 volts. Since that value is low, the objective is to raise the voltage to accommodate a much larger interval that can be converted into a pH level through programming the Arduino. So, the first thing to do is design the circuit on Multisim, followed by producing a physical version on a bread board. Below, the simulated circuit compared is to the physical one:

Figure 6 - Voltage Booster Simulation and Prototype



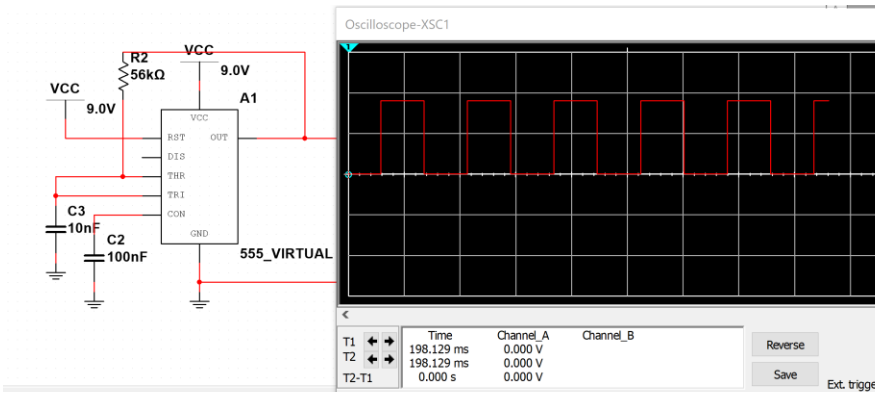
Now, when it comes to testing, there are various issues to be overcome, such as static noise, which is common on these step-up circuits. One way to fix this is filtering out the noise by using a bigger diode, as well as ensuring the components are properly grounded.

555 Timer

The timer is used because the MOSFET requires a step-up voltage to open the logic gate and raise it to the next level. By inputting a high voltage, like 9V, into the timer, an output of approximately 5V is generated. And, by configuring the capacitor and resistor values, the desired output frequency of 1kHz is achieved. This signal can then be input into the

MOSFET gate to continue the process of raising the voltage. The square function is output like this:

Figure 6 - Square Wave Produced by the 555 Timer



The voltage is now stepped up and produces the desired input for the analog port of the microcontroller, ready for A-D conversion and transmission to the cloud.

Bill of materials

Table 8 - Bill of Materials

	Item	Price
Leaf Patch	FormLabs BioMed Clear Resin (3g)	3.00$
	Graphene Ink (1ml)	1.00$
	Copper Tape (~2cm)	<0.01$
	2 Cables	0.05$
Voltage Booster	2 Resistors (22kΩ	1.00$
	1 Inductor (470µH)	1.00$
	3 Capacitors (1µF)	1.25$
	1 MOSFET	0.35$
	555 Timer	1.25$

Table 8 provides an estimation for the cost to produce a single patch and the voltage booster. If the microcontroller is added (value: ~$10), the total system cost, barring mistakes during the manufacturing process, is less than $20.

Equipment needed

Table 9 - Equipment Needed

Item	Price
Formlabs 3D Resin Printer	3,500.00$
Formlabs Form Cure	700.00$
pH Tester	500.00$
Keysight Oscilloscope	8,800.00$
Electronics Kit	70.00$

Table 9 gives the cost to purchase the equipment necessary to manufacture and test the microneedle patch. This 3D resin printer has an extremely large resolution and allows microneedles to be precisely and consistently manufactured. The pH tester is used on the plant to double-check the finished product. The oscilloscope is used to confirm the signal output from the patch is correct, and the electronics kit contains the components needed for the voltage booster, as well as several spares and variants of RLC parts in case the theoretical calculations were off.

Manhours

Table 10 - Shop-Man Hours

Time needed to recreate prototype	
Item	**Time**
Printing Patch	1h
Cleaning Patch	0.5h
Curing Patch	1h
Coating Microneedles	0.25h
Connecting Wires	0.25h
Assembling Voltage Booster	2h
Programming the MCU	1h
Coding the Cloud System	3h

Table 10 provides the estimated time to complete the manufacturing of a single patch, though the process must be understood, and no mistakes can be made. This also does not include the time to coat the patch with polyaniline, since this hasn’t been done yet.

Assuming a knowledgeable and experienced person, and that no testing needs to be done, the time to manufacture a total system is approximately 9 hours.

5. Testing and Validation

This section describes in detail the steps to assess the functionality outlined in previous chapters. There are different methods available, but only the chosen method will have detail with a paragraph following on how other methods could have helped achieve the goal and what could be done to build upon the final project/design. A simulated diagram of all the components combined on a 31x26 Printed Circuit Board can be found in **Appendix B 8.4.**

<u>Microneedle Patch:</u>

Testing the microneedle patch requires some extra steps after its creation to ensure it works correctly. The steps outlined in Section 4 describe the creation process, but not the validation process. Since the goal of the patch is measuring pH, it must be calibrated. This calibration must be done in accordance with the expected pH values of the plant. The best method for this will now be outlined. First, a sample of sap is extracted from the plant. The sample is analyzed to determine its specific composition and default pH value. Knowing the default value helps to estimate the probable range of pH values. Then, from the expected range and the sample's composition, calibration solutions are made. These calibration solutions are called "buffer solutions"; buffer solutions have a specific and pre-determined pH value, hence their use in calibration. Also, because they are created in partnership with the plant sample's composition, buffer solutions effectively tune the pH patch sensor to the plant whose sample was used for calibration. This means that the pH readings are likely to have a much higher accuracy than if random, unrelated sample solutions were used for calibration.

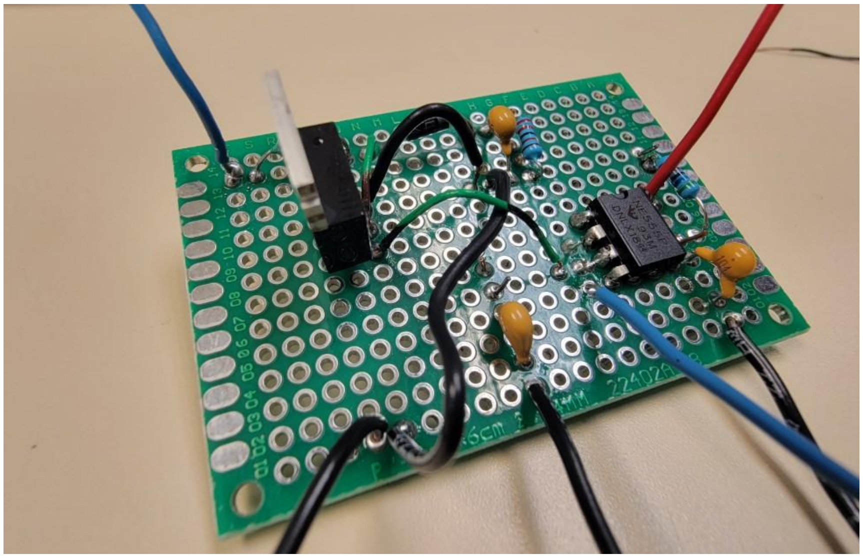
Now, as mentioned in Section 4, the microneedle patch requires 3 coatings. The first 2, graphene and silver-silver chloride, are easy enough to accomplish by the authors of this paper. However, the final coat, which is polyaniline, is a bit more involved than initial research indicated. As such, we (the authors) have been informed by our project mentor that a graduate student needs to assist us in applying the polyaniline coating. We will now explain the reason. As best as we understand, coating with polyaniline can be done in 2 ways: in-situ chemical oxidative polymerization **[2]** or electrochemical reduction followed by electro polymerization **[3].** We believed that polyaniline coating was simpler, requiring only electrolysis. But even supposing that electrolysis is possible, it is not straightforward and needs multiple set-up steps such as defining the correct electrolytes and current amperage necessary to apply the desired coating. This is not to say using polyaniline in our project is impossible, only that it will take longer than expected since the process itself we know is longer and we must also coordinate a time slot to work with a graduate student.

Thus, the validation and testing process of the microneedle patch is currently incomplete.

Voltage Booster:

The simulation design shown in **Section 4** gives details and results that are within the range of the goals stated in **Section 2**. Now, to ensure the microcontroller and voltage booster run on their own and are adaptable for users, the circuit is soldered to a printed circuit board (PCB) and then tested to see if the achieved results are the same as the simulation. Figure 7 below is the voltage booster laid out on a small piece of PCB; the external cables lead to the ESP32 microcontroller.

Figure 7 - Voltage Booster/555 Timer Circuit



In the top left corner of the PCB, there is a blue wire which is the patch electrode input leading to the voltage booster. On the right side, the flat 8-pin component is the 555-timer circuit which controls the frequency input of the MOSFET. For this project, the frequency is set at 1kHz. The black wires will be connected to the ground on the microcontroller to complete the circuit.

Figure 8 - Oscilloscope Results

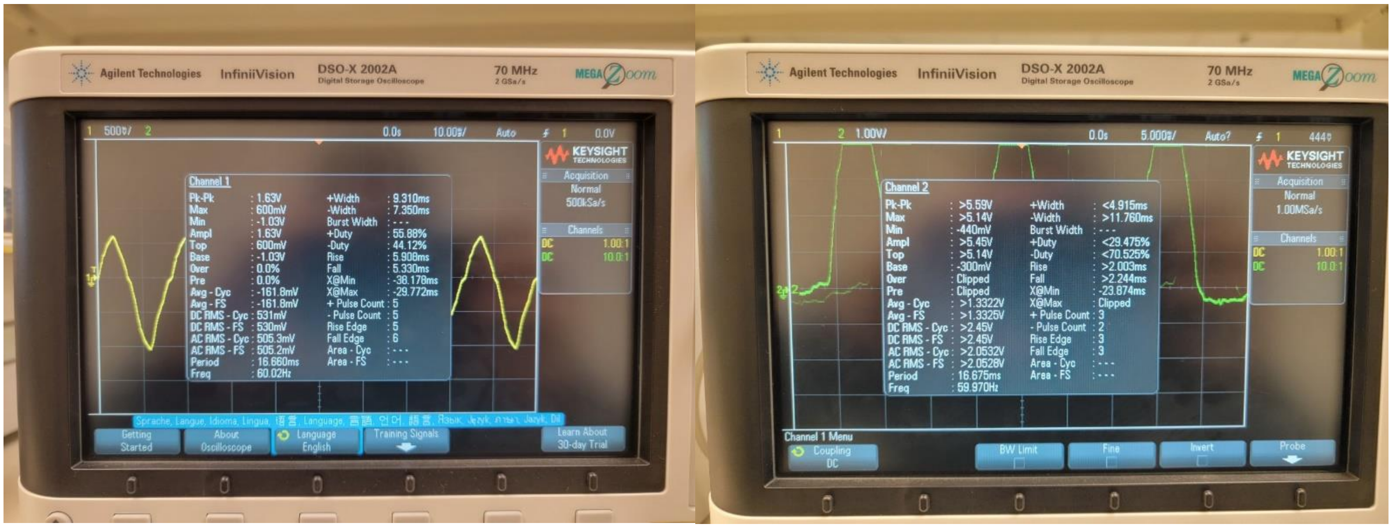


Figure 8 above shows the oscilloscope readings for simulated low-voltage inputs. The left image shows an initial 1.63 volt reading and the right oscilloscope shows that it increases over time to cap at 5.60 volts. Low voltages are expected from the plant; it is likely that the range of voltages read will be between 0.1V and 0.5V and almost definite that 1V will not be exceeded. So, even though 1.63V is not a reading likely to come from the patch, it is used as an example to prove that any provided voltage can be raised. One error that has been detected already is a small amount of noise coming from the booster. This noise, since the circuit inputs are being simulated, is coming from the diode, and can be addressed by using a better diode, adjusting the circuit to account for this noise, or, if it's bad enough, adding a filter.

Different methods for boosting voltage will create different PCB layouts, and even within the same design, different engineers might create different variants. Since this project was completed step-by-step, each piece is on its own PCB board. One obvious design improvement would be designing a layout for them all to be one the same piece of PCB.

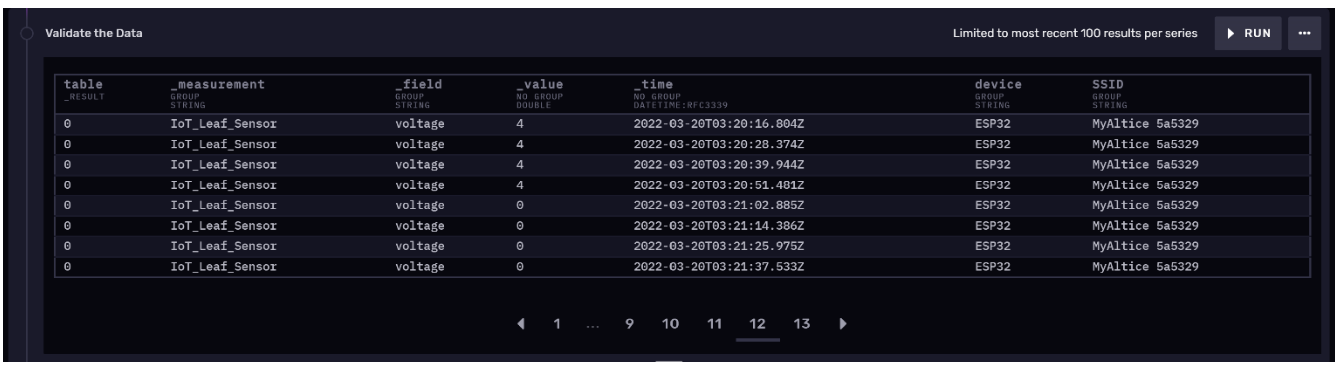
<u>Microcontroller/Code/Cloud:</u>

The code provided in **Appendix A** is simple enough to test with just a Wi-Fi connection and the microcontroller; testing it gives the expected results. Then, the important thing is testing the new cloud system (Grafana). The web cloud system of Grafana can be accessed through external links which can be shared between anyone. InfluxData, its cloud system, allows for different settings to edit in different plotting services. Figures 9 and 10 below show the real time data being stored and plotted on a graph.

Figure 9 - IoT Cloud System with Grafana

This is the graph displaying simulated pH levels. The vertical axis is in pH, from 0 to 14, and the horizontal axis is in time. For testing Grafana, different pH levels are applied to prove the theoretical design explained in the previous section.

Figure 10 - IoT Cloud System Raw Data



Validate the Data

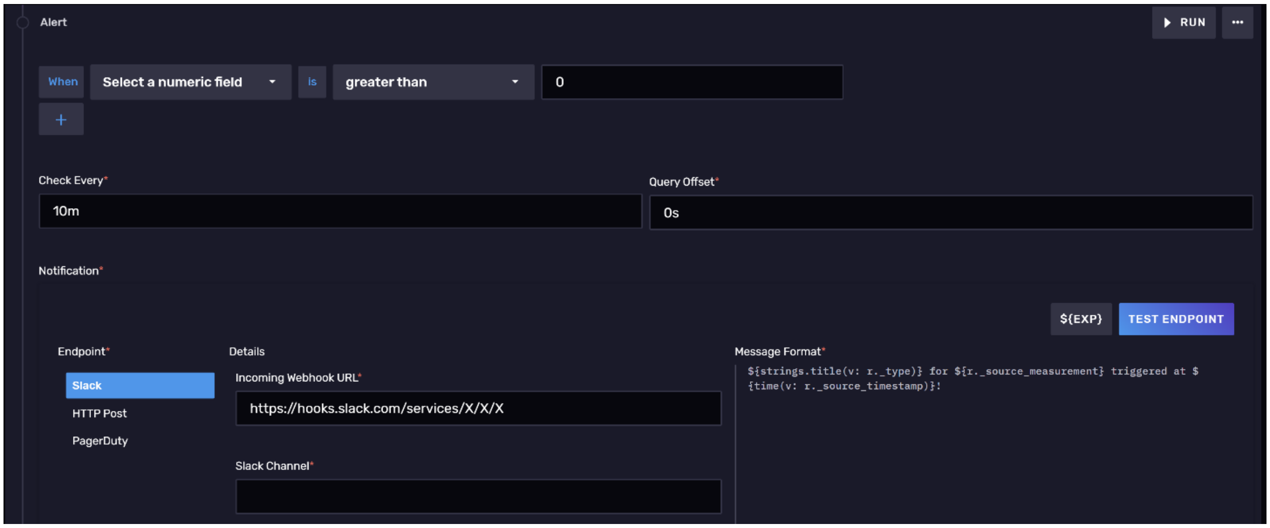
Limited to most recent 100 results per series

table _RESULT	_measurement GROUP STRING	_field GROUP STRING	_value NO GROUP DOUBLE	_time NO GROUP DATETIME:RFC3339	device GROUP STRING	SSID GROUP STRING
0	IoT_Leaf_Sensor	voltage	4	2022-03-20T03:20:16.804Z	ESP32	MyAltice 5a5329
0	IoT_Leaf_Sensor	voltage	4	2022-03-20T03:20:28.374Z	ESP32	MyAltice 5a5329
0	IoT_Leaf_Sensor	voltage	4	2022-03-20T03:20:39.944Z	ESP32	MyAltice 5a5329
0	IoT_Leaf_Sensor	voltage	4	2022-03-20T03:20:51.481Z	ESP32	MyAltice 5a5329
0	IoT_Leaf_Sensor	voltage	0	2022-03-20T03:21:02.885Z	ESP32	MyAltice 5a5329
0	IoT_Leaf_Sensor	voltage	0	2022-03-20T03:21:14.386Z	ESP32	MyAltice 5a5329
0	IoT_Leaf_Sensor	voltage	0	2022-03-20T03:21:25.975Z	ESP32	MyAltice 5a5329
0	IoT_Leaf_Sensor	voltage	0	2022-03-20T03:21:37.533Z	ESP32	MyAltice 5a5329

1 ... 9 10 11 12 13

Another option is allowing the user to always view data and be able to read raw information while using the same dashboard. Of course, visualizing is better but having access to the data entry logs is a benefit, since they carry timestamps.

Figure 11 - IoT Alert System



The last feature is allowing the user to set their own alert. For example, if the pH reading is too high for a plant, then this can be set as a variable to send a notification through online or to an email address. Figure 11 shows the easy steps for adjusting these features, like using If-Else statements.

The program can be confusing, but the application contains default settings that can be applied to its own dashboard. That means the manufacturer can create a set up for a customer, enter this program, and fix it to the customer's values.

Some improvements that could be done is finding a program that has easier-to-use settings or maybe design some better pre-made functions. For future use, the IoT system can be edited in many ways, whether it's the code or the cloud system.

Table 111 - Testing Results

Specification tested	Protocol (HOW the test was performed)	Results Discussion
Voltage Booster	For this test an oscilloscope and multimeter were used to measure the inputs and outputs of the voltage booster. Applying a voltage allowed for reading test in real time to the cloud system.	Initial: 1.63 volts Final: 5.60 voltage
Arduino code for voltage detection and Transmission	For this test, set voltage inputs were applied to the microcontroller and processed	The microcontroller code read the voltages perfectly and sent the results to the web server and is shown on figures.
Ph detection with patch	For this test we used know pH solutions and added them to the sensor on the patch to find what voltage reading the patch will produce at that pH	(Relationship between pH and Voltage)

6. Broader Impacts of the Project

The most important ethical roles that this design encompassed are the abilities to account for the safety of the people and provide for the welfare of the people. In other words, the ethical consideration of safety in this design is to ensure that any person coming into close contact with the product will be protected from all potential harms, such as electrical shocks, battery leaks, or cuts from handling the microneedles. Ethical standards are provided throughout engineering history, and the safety of the people has always been among the strongest responsibilities. This product, as outlined in **Appendix A**, adheres to electrical engineering ethical considerations.

But welfare also matters: this product helps to create a higher standard of living wherever it is used by ensuring that field crops and produce plants have a consistent and higher yield, excepting extreme circumstances such as natural disasters. Thus, this project advances the main priority of engineers, which is to make the world a better place, progressing toward the best future, and to ensure the safety of the people while doing it. This design has the potential to affect larger scales of society by increasing the amount of food that can be produced within a country or state. This design aims to address current farming failures and transform farming such that product loss is less frequent through providing the status of the health and vitals of crops to the farmer. This improves how the farmer distributes water and fertilizer to his crops. If all farmers had the possibility to own this product and thusly learned how to control their crops effectively, then this could improve world health drastically. Produce imports would decrease and produce exports would increase, eventually reaching a point where most first world countries can become a supplying country and help improve worldwide food distribution. This project doesn’t stop at just plants, though. The process could be applied to develop a human diagnosis patch. The patch would be able to read a patient’s vitals and alert them or a nurse to an upcoming tragedy before it happens. This design could extend into the medical field and attract sponsors who are willing to help broaden this research. In today’s world, science and engineering have been better able to discover how things work in the world than ever before. The use cases for this design are many, and now it can now be incorporated into something people use daily. The ability to incorporate this design into the modern world is an attractive feature and makes for ease of marketing and product development.

Moreover, engineers have personal ethical responsibilities they must consider. Whenever we (the authors) were asked by our mentor how much we had completed of this project, we were honest and accepted whatever she needed done. We had a certain development schedule to follow and certain deliverables to create for each deadline. The sponsor reflects future clients who will want to be treated with honesty and respect. If these ethical considerations are met, then the demand for our design product will grow and word will get around about our integrity as an engineer. Our integrity is reflected in our reputation, which is talked about by peers and customers alike, and our reputation earns us both jobs and friends. Since the product has several different components, each member of the team was given a different task to complete, though each member helped the others as time allowed. All the separate parts relied on each other to create a complete and functioning system. Trusting and being able to count on the other members to hold their own was important to us, so that we don't all try to do the entire project on our own.

7. Conclusions

Overall, each function completed its objective and integrated smoothly with the others. Accurate readings from the Arduino IDE code and Grafana were achieved, displaying everything needed for a user to view the status of their plant. It should be evident from the initial design that much testing and simulation needed to be done, whether it was the patch or the programming, and the route from patch electrode to cloud smoothly carried the data. The voltage booster is necessary to increase the small voltage value from reading a leaf. The code stabilizes at the 7 (base) pH level and recognized rises and drops when different voltages were applied. The ability to store continuous information is accomplished, and being able to configure tables, charts, and alerts allows for a user to have a customized experience.

By no means is this a finished product: there are various tests yet to be done to figure out how it will function in different climates or temperatures. In these environments, there might be minor off-sets which would make the plotted results deviate from expected. This can be mitigated by redoing the calibration in the new environment. Another issue is determining the accuracy of the patch readings. As mentioned in **Section 5**, the project cannot be completed without graduate help. Since the process involves electrochemical theory, the final product's accuracy has yet to be determined.

For future uses, this design could be incorporated into bigger facilities with multiple patches to create an ecosystem with multiple IoT nodes. Instead of using only one of these systems, multiple individual systems can be used to create arrays of patches which the code would us to create a matrix system. Then, when there is a plant that needs nutrition, the microcontroller and cloud system can precisely pinpoint its location. This can be taken even further: with the capabilities that a microcontroller carries, it is easy to imagine functions like a self-watering system or even greenhouse windows that automatically increase or reduce sunlight. The possibilities are endless, and this project is just the first step in helping agricultural farmers increase the lifespan and production of their crops.

8. References

[1] Shipman, M., Wei, Q., & Zhu, Y. (2021, July 7). Plant Patch Enables Continuous Monitoring for Crop Diseases. doi:10.1016/j.matt.2021.06.009

[2] Ahirrao, D. J. (2021). Nanostructured Porous Polyaniline (PANI) Coated Carbon Cloth (CC) As Electrodes For Flexible Supercapacitor Device. Journal of Materials Science & Technology, 88, 168-182. doi:10.1016/j.jmst.2021.01.075

[3] Vacca, A. M. (2014). Coating Of Gold Substrates With Polyaniline Through Electrografting Of Aryl Diazonium Salts. Electrochimica Acta, 126, 81-89. doi:10.1016/j.electacta.2013.08.187

Appendices

Appendix A: Codes, Standards, and Constraints

Organization	Engineering Standards
IEEE	IEEE 2857-2021 - IEEE Standard for Wireless Smart Utility Network Field Area Network (FAN) This standard includes a safe and secure communication network security. With different network meshes come different parameters of security that need to be followed. With this standard, it is clearly stated that the basic security procedures are to be followed while making the communication sensor.
Safety	IEC 60590-1 and its adopted version UL 60590-1 Information Technology Equipment - Safety - Part 1: General Requirements; This code indicates that a technological piece of equipment is liable to hold and not exceed a max voltage of what the device in size is meant for. The device should be safe when communicating and the operating voltage will not exceed a maximum voltage which the specification calls for in the certain equipment. In our case, it should not exceed 5V.
Government	OSHA-1910.303 For General Electric standards. OSHA does not currently have standards on radio-frequency radiation as would be present in IoT (Wi-Fi, Bluetooth, RFID).
International	ISO/CD 5231 Extended Farm Management Information Systems Data Interface (EFDI) Indicated within is the protection of agriculture and farm-based population crops. The device being used should not be liable for destroying or damaging the crops that it within its range. The interface will be made where it can’t be compromised, and the data won’t implicate crop life or production.

Technical Descriptions	
Cost	The budget for this project shall not exceed 1000$. Each microcontroller (1 per patch) would be approximately 20-30 and the patch itself would be relatively inexpensive per unit, being mostly 3D printed. Most of the tools and equipment needed to construct are already provided in the lab.
Speed	Variable data polling frequency means an update speed ranging from every hour to every minute. Because data is not going to change that quickly, there must be a significant difference to let the user know what should be done, which can be edited in the code.
Reliability	The reliability of the Arduino will likely be better than the patch itself with it having fewer delicate components. The patch itself is expected to have a lifespan in the months or years, though testing will be necessary. And, to ensure accurate readings, several patches can be paired with one Arduino on the same plant, which also provides redundancy.
Power	Power will be mostly consumed by the microcontroller which runs on a Lithium-Ion rechargeable battery. The code must be read every hour, so not much power consumption will take place. But testing needs to be done to ensure that a battery lifespan can last for multiple days.

Complexity	The microneedles are on the micrometer scale, but the 3D printer can print at that level; the complexity of the patch itself is dependent on creating a good 3D rendering for the printer to use. The governing mathematics of electronics will be absolutely present in order to connect the patch to the Arduino and convert voltage differences to pH levels.
Manufacturability	Arduino microcontrollers can be sourced, and with enough printing material, patches can be made continuously. The struggle of fast manufacturing will be mapping the proper code to every Arduino before use and syncing each one with the cloud.
Sustainability	Patch will be reusable, able to be attached multiple times along with the microcontroller, but the patch would inevitably have to be replaced due to wear. The wear and tear period will depend on the 3D ink used in the microneedles, how rough or gentle each patch is handled as the needles are very small, and the weather conditions during the time the patch is attached and monitoring pH levels.

	Ethical and Professional Considerations
Public health	GMO’s are generally accepted by people, but there are worries surrounding modified foods. Some plant fertilizers aren’t too healthy, either. By monitoring plant health and thus maximizing the growth potential of all plants, it is possible that the number of chemicals used to grow produce will be reduced and the average human will consume less toxic material.
Safety and welfare	One major component of community welfare is having enough food. Unpredictability in food production could mean the difference between a full harvest and a poor one. By utilizing this patch, crop production can be stabilized through better management of water in times of drought, distributing it only when and where needed.
Global factors	As crops increase, more global trade occurs, increasing the prosperity of multiple countries. Not only that, but this patch can be adapted for use in third world countries, even if they don’t have Wi-Fi, and may increase food production where previously not thought possible.
Societal factors	Crop production will surge and flourish becoming more stable. This leads to more economic growth within cities, states, and countries. Also, with the ability to monitor the health of crops from anywhere with internet access, farming becomes vastly more sociable.
Environmental factors	Ecosystems will thrive in growth, therefore expanding and helping wildlife management and planetary wellness. A patch that monitors plant health is not only for agriculture. This can help in restoring or preserving precious ecosystems.

Economic factors	The cost of materials will be inexpensive; therefore, the finished product should be affordable for most people. A cheap solution to an expensive problem feels like giving out money for free.

Appendix B: User Manual/Instructions

8.1 WIFI set up

```
// Setup wifi
WiFi.mode(WIFI_STA);
wifiMulti.addAP(WIFI_SSID, WIFI_PASSWORD);

Serial.print("Connecting to wifi");
while (wifiMulti.run() != WL_CONNECTED) {
  Serial.print(".");
  delay(500);
}
Serial.println();

// Add tags
sensor.addTag("device", DEVICE);
sensor.addTag("SSID", WiFi.SSID());
```

8.2 Basic Arduino Code

```
int readPin=34;
int readVal;
//float voltage_test1 = 0;
float voltage_test1;
int delayTime = 500;
void setup() {
  pinMode(readPin,INPUT);
     //CODE
  // Initialize serial and wait for port to open:
  Serial.begin(9600);
  // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
  delay(1500);
}

void loop() {
  readVal = analogRead(readPin);
voltage_test1=(5./1023.)*readVal;
Serial.println(voltage_test1);
delay(delayTime);
}
```

8.3 Complete Arduino Code

```
#if defined(ESP32)
#include <WiFiMulti.h>
WiFiMulti wifiMulti;
#define DEVICE "ESP32"
#elif defined(ESP8266)
#include <ESP8266WiFiMulti.h>
ESP8266WiFiMulti wifiMulti;
#define DEVICE "ESP8266"
#endif

#include <InfluxDbClient.h>
#include <InfluxDbCloud.h>

float voltage_test1;
int readPin=34;                  // Analog pin input
int readVal;                     // Value being received that will change to comprehensive results

// WiFi AP SSID
#define WIFI_SSID "MyAltice 5a5329"
// WiFi password
#define WIFI_PASSWORD "green-993-296"
// InfluxDB v2 server url, e.g. https://eu-central-1-1.aws.cloud2.influxdata.com (Use: InfluxDB UI -> Load Data -> Client Libraries)
#define INFLUXDB_URL "https://us-east-1-1.aws.cloud2.influxdata.com"
// InfluxDB v2 server or cloud API authentication token (Use: InfluxDB UI -> Load Data -> Tokens -> <select token>)
#define INFLUXDB_TOKEN "MfwFpl2KZpMjQTWujXzxpKl9Xpe4rnq-CKfn6Q5Q7EcJfvnePvWGhNHHyZ69Aq3_7SL77oDnfo2f8Whiq-mW0w=="
// InfluxDB v2 organization id (Use: InfluxDB UI -> Settings -> Profile -> <name under tile> )
#define INFLUXDB_ORG "cgalvan5@patriots.uttyler.edu"
// InfluxDB v2 bucket name (Use: InfluxDB UI -> Load Data -> Buckets)
#define INFLUXDB_BUCKET "IoT_Leaf_Sensor"
```

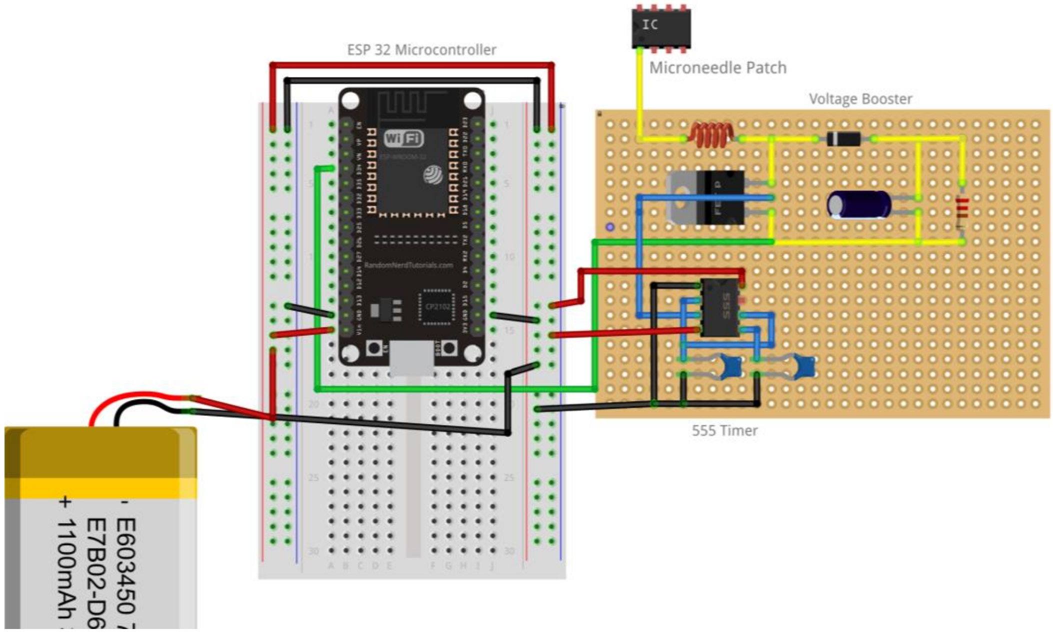
```
// InfluxDB client instance with preconfigured InfluxCloud certificate
InfluxDBClient client(INFLUXDB_URL, INFLUXDB_ORG, INFLUXDB_BUCKET, INFLUXDB_TOKEN, InfluxDbCloud2CACert);
// InfluxDB client instance without preconfigured InfluxCloud certificate for insecure connection
//InfluxDBClient client(INFLUXDB_URL, INFLUXDB_ORG, INFLUXDB_BUCKET, INFLUXDB_TOKEN);
```

```
// Data point
Point sensor("IoT_Leaf_Sensor");

void setup() {
  Serial.begin(115200);

pinMode(readPin,INPUT);        //IoT initiation with delays for proper connectivity
  Serial.begin(9600);            // Initialize serial and wait for port to open:
  delay(1500);                   // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
  // Setup wifi
  WiFi.mode(WIFI_STA);
  wifiMulti.addAP(WIFI_SSID, WIFI_PASSWORD);

  Serial.print("Connecting to wifi");
  while (wifiMulti.run() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();

  // Add tags
  sensor.addTag("device", DEVICE);
  sensor.addTag("SSID", WiFi.SSID());
```

```
  // Alternatively, set insecure connection to skip server certificate validation
  //client.setInsecure();

  // Accurate time is necessary for certificate validation and writing in batches
  // For the fastest time sync find NTP servers in your area: https://www.pool.ntp.org/zone/
  // Syncing progress and the time will be printed to Serial.
  timeSync(TZ_INFO, "pool.ntp.org", "time.nis.gov");

  // Check server connection
  if (client.validateConnection()) {
    Serial.print("Connected to InfluxDB: ");
    Serial.println(client.getServerUrl());
  } else {
    Serial.print("InfluxDB connection failed: ");
    Serial.println(client.getLastErrorMessage());
  }
}
```

```
void loop() {
  // Store measured value into point
  sensor.clearFields();
  readVal = analogRead(readPin);  // Gets value from analog pin and saves it under readVal = 0
voltage_test1=(1./1023.)*readVal; // Code is desplayed as binary so this converts it to floating numbers
Serial.println(voltage_test1);    // Print the result onto a graph
  // Report RSSI of currently connected network
 sensor.addField("voltage", voltage_test1);
  // Print what are we exactly writing
  Serial.print("Writing: ");
  Serial.println(client.pointToLineProtocol(sensor));
  // If no Wifi signal, try to reconnect it
  if (wifiMulti.run() != WL_CONNECTED) {
    Serial.println("Wifi connection lost");
  }
  // Write point
  if (!client.writePoint(sensor)) {
    Serial.print("InfluxDB write failed: ");
    Serial.println(client.getLastErrorMessage());
  }

  //Wait 10s
  Serial.println("Wait 10s");
  delay(10000);
}
```

8.4 Final Drawing



Appendix C: Knowledge and Skills acquired in previous coursework