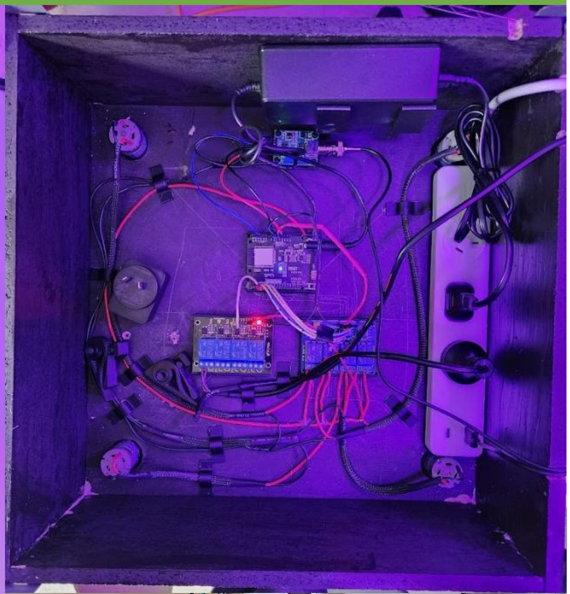
Designed Solution

Semi-automated

Vertical Hydroponics System



Stage 2 Electronics

2023

SACE ID: 268419R https://youtu.be/O1UjPc237cc

Contents

Design Brief

The Problem

The small-scale hydroponics system aims to develop a transportable and sustainable alternative to conventional agricultural techniques. Traditional agricultural practices face several difficulties, including a lack of available farmland, water shortage, dependence on chemical fertilisers, and a vulnerability to pests and illnesses. These issues are addressed by the small-scale hydroponics system, which provides a portable, lightweight method of growing plants without soil, integrated with IoT *(Internet of Things, which is an online web platform that can monitor and control the hydroponics system).* It maximises the utilisation of available space and uses the least amount of water by using regulated conditions and nutrient-rich water solutions. Because the plants get the nutrients they need directly, this strategy also lessens the need for artificial fertilisers. Fewer insecticides are required since the regulated atmosphere helps lower the danger of illnesses and pests. Moving farms closer to cities encourages sustainability and localised food production by reducing the costs and carbon footprint associated with long-distance food delivery.

Design Requirements

Portability: To provide adaptability and versatility, the system must be portable, allowing users to set it up and relocate it as required.

Space Efficiency: The system should incorporate vertical or compact designs to make the most of available space and allow for the development of many plants to be space efficient.

Water Efficiency: By utilising water recycling and effective watering techniques, lowering waste, and encouraging sustainability through water efficiency, the hydroponics system should be created to use as little water as possible.

Nutrient Delivery: The system should ensure that nutrient solutions are delivered to the plants effectively and accurately, maximising their health and growth.

Modularity and Scalability: Flexibility in the design should enable simple expansion or modification based on the user's requirements and the available area. It should also allow for different plant sizes and species.

User-Friendly Operation: Users of all levels of competence should be able to use and maintain the system successfully due to the system's intuitive design, user-friendliness, and clear instructions and controls.

Design Constraints

- To ensure cost and viability, the system must be constructed within a budget of $120 and up to $400.
- **Size and Weight:** There should be weight and size restrictions for the system to be portable and functional aiming to grow approximately 40 plants.
- **Power Supply:** Considerations like energy efficiency and the availability of power sources should be considered while designing the system to run under power sources.
- **Maintenance and Upkeep:** The system should be created with the simplicity of maintenance in mind, taking access to components, cleaning needs, and routine maintenance duties into account.
- **Safety:** The system must adhere to safety standards to safeguard users and reduce possible risks.
- Materials and Resources: Materials should be chosen with sustainability objectives, considering durability, recyclability, and availability.
- **Climate and Environmental Factors:** The system must be built to resist certain climatic conditions and environmental elements that may impact its performance, such as temperature extremes, humidity levels, and sun exposure.
- **Plant Varieties:** The system should consider the specific needs of the targeted plant kinds, considering things like light intensity, nutritional content, and compatibility with growth material.
- **Timeframe:** The design and execution phases of the project may need to consider any unique deadlines or time limitations.

Exploration of Existing Products

Product 1: Wick System

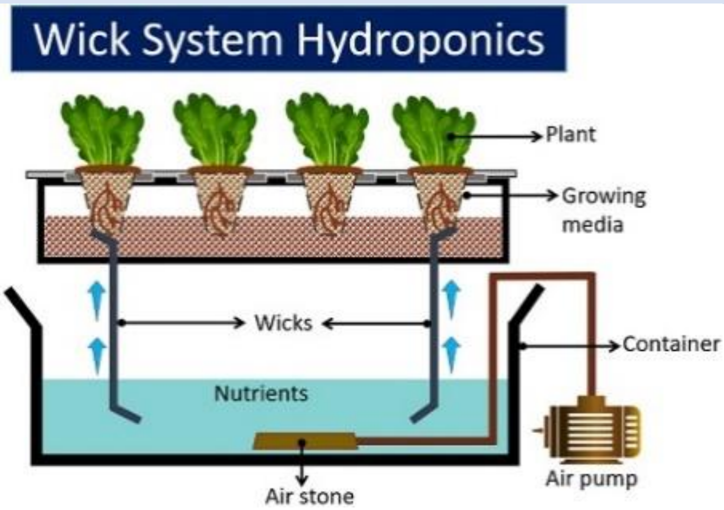


Figure 1: Wick System diagram (Supriya, 2022).

Figure 2: Wick System Prototype (Jeena, 2023).

The Wick system excels in **user-friendliness** with easy setup and low maintenance. Its **modularity** enables expansion and adaptability to different plants. **Nutrient delivery** via wick or nylon thread promotes plant health. **Water efficiency** is achieved through capillary action. **Compact design** optimizes space. **Portability** allows for easy setup and relocation.

Product 2: Ebb and Flow (Flood and Drain)

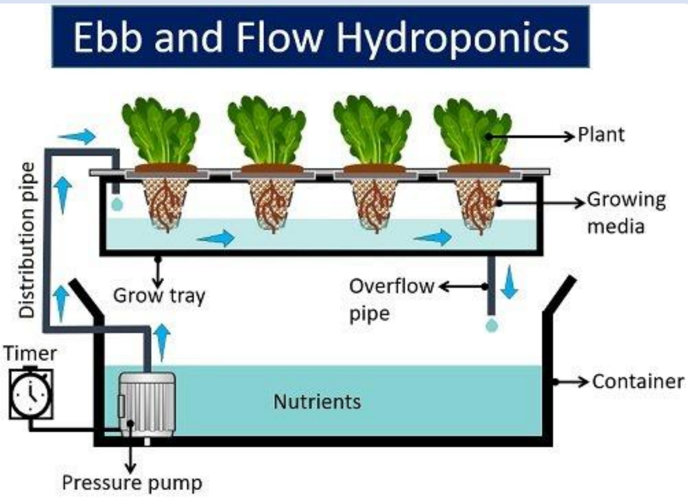


Figure 3: Ebb and Flow system diagram (Supriya, 2022).

Figure 4: Ebb and Flow system real life example (Lee, 2022).

The Ebb and Flow system are a **user-friendly** operation, as they require only the manual work of plantation and nutrients. **Nutrient delivery** is efficiently ensured as the water pump distributes nutrient-rich solution to the grow bed, promoting plant health and growth. Its **modularity** allows easy expansion or modification based on user needs, requiring more space, supporting various plant sizes and species. The Ebb and Flow system are portable on **smaller scale designs**, offering users the flexibility to set up and **relocate** it as needed. The system's **water efficiency** is demonstrated by a cyclic schedule controlled by a timer, minimising water waste.

Product 3: Drip Systems

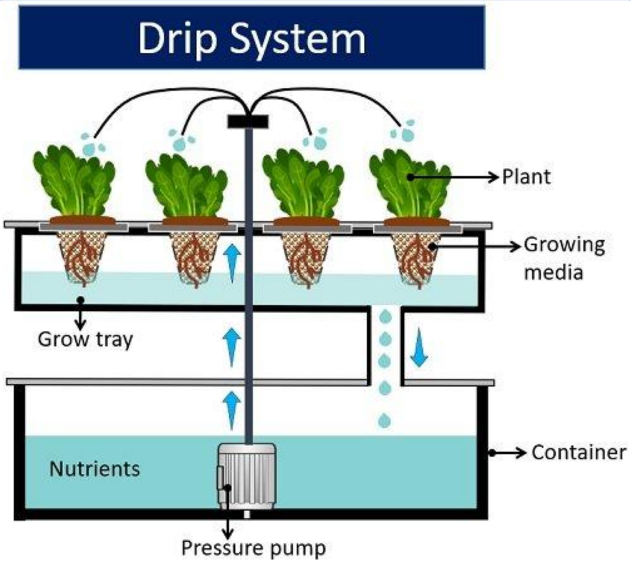


Figure 6: Drip system real life (Nadaraja, 2022).

Figure 5: Drip system diagram (Supriya, 2022).

It features an intuitive design that requires maintenance for the pipes and cleaning, still **user-friendly**. **Water efficiency** is achieved through precise dripping, minimising waste. The system's **space efficiency** allows for the cultivation of multiple plants in a compact footprint. Is **portable** in smaller scales for convenient setup and relocation, adapting to various settings. The system can be easily **expanded or modified** to accommodate different plant sizes and species, in vertical or horizontal designs. **Nutrients** are delivered effectively through a constant flow of water and nutrient solution.

Product 4: Aeroponics

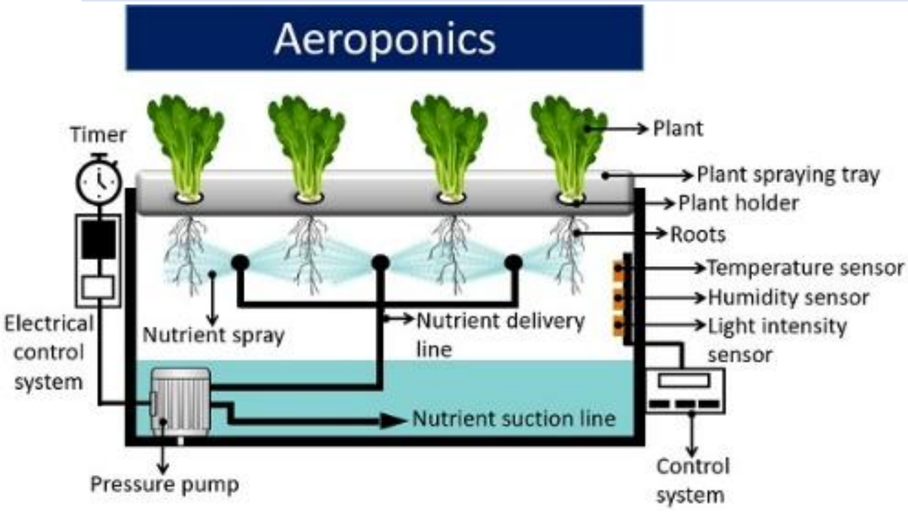


Figure 7: Aeroponics diagram (Supriya, 2022).

Figure 8: Drip system diagram (Barth, 2018).

It is the **easiest** type of hydroponic system even if it looks complicated. Here, plants and their roots are suspended in the air and a control system does all the work, the user just harvests the plants. The water pump in the reservoir comprises several **mist nozzles**. Misters give a fine spray of the **nutrient** solution onto the roots of each plant. The rest of the nutrient solution falls into the reservoir. To construct aeroponics, you must have an idea of the correct **dimensions** of the reservoir. For instance, a reservoir should be deep to grow larger plants. The aeroponics can be **expanded** vertically or horizontally and **portable** depending on the size of the hydroponics design.

Product 5: N.F.T. (Nutrient Film Technology)

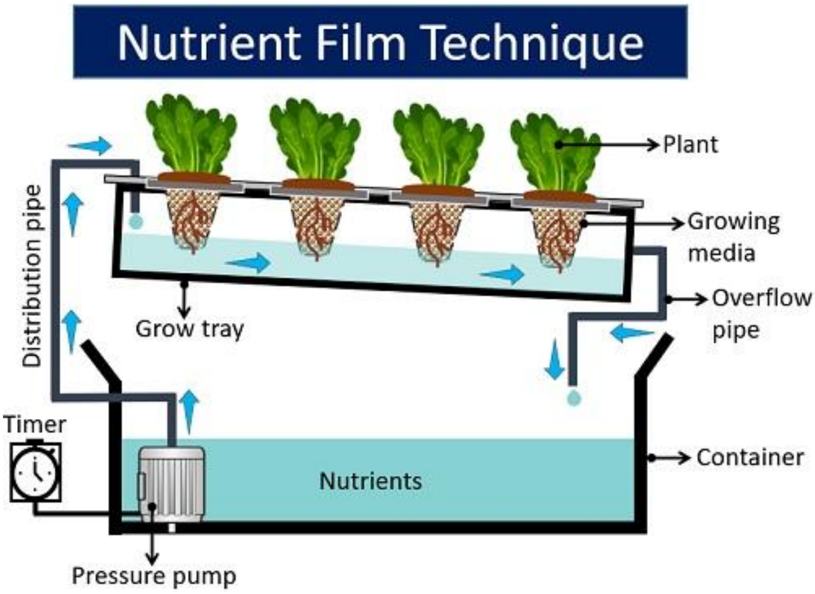


Figure 9: Nutrient Film Technique N.F.T. diagram (Supriya, 2022).

Figure 10: Nutrient Film Technique N.F.T. real life example (Trees.com Staff, 2017).

It is a **simple** method for commercial and home gardens. The reservoir of the nutrient film technique contains **nutrient-rich** water and a motor. The water pump drives the nutrient-rich solution upwards to the sloping channel. Here, the grow tray appears as a sloping channel which is **modular**. The plants are **put inside the net pots with** or without the growing media. When the pressure rises, the motor push water to the grow tray for a predetermined time and then the excess **falls back to the reservoir**. Can be **portable** with smaller designs.

Product 6: Deepwater culture

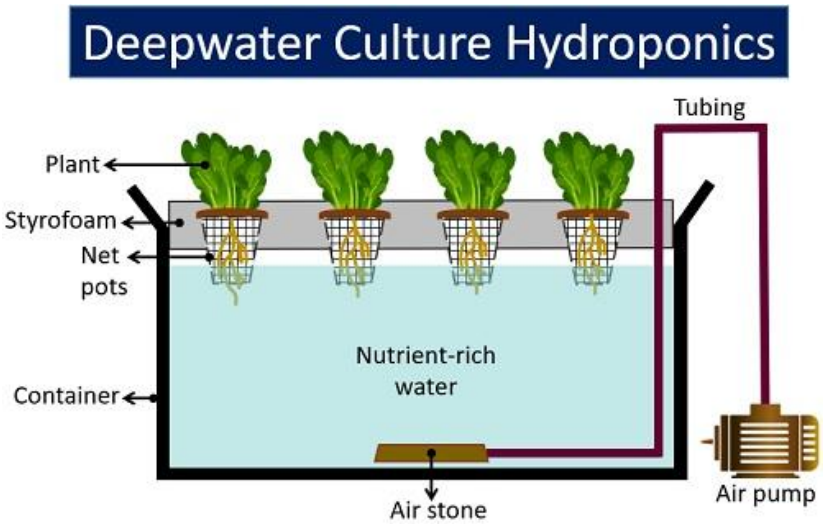


Figure 11: Deepwater Culture diagram (Supriya, 2022).

Figure 11: Deepwater Culture rea life example (Espiritu, 2014).

This is less **user friendly** as it is higher maintenance and more water usage. It is also called a reservoir system, containing **nutrient solution**. Here, the plant roots are in direct contact with the nutrient source. A diffuser or air stone allows uniform distribution of oxygen in the reservoir. The deepwater culture system uses net pots to **keep plants in their proper position** which is **modular**. **More water** is required meaning it is less **portable**.

Summary of Exploration of Existing Products

Table 1: *Summary of the different Hydroponics systems explored.*

Criteria	1 - Wick	2 - Ebb and Flow	3 -Drip	4 - Aeroponics	5 - N.F.T. (Nutrient Film Technology)	6 - Deepwater culture
Portability	B	C	A	B	A	D
space efficiency	C	B	B	A	B	C
water efficiency	C	B	C	A	A	C
effective nutrient delivery	C	B	B	A	A	A
Modularity and Scalability	B	C	B	B	A	C
user-friendly	B	B	B	A	B	C

Note: Criteria grading scale
A – Excellent
B – Good
C – Average
D – Below Average
E – Poor
F – Fail

The different hydroponic systems encompass various design principles and criteria for efficient plant growth. These systems include Wick, Ebb and Flow, Drip, Aeroponics, N.F.T. (Nutrient Film Technology), and Deepwater Culture. Each system is designed to provide plants with nutrients and water in specific ways, facilitating their growth without soil. Despite their differences, they all adhere to essential hydroponic principles, emphasising controlled nutrient delivery and root access to oxygen. These systems enable growers to cultivate plants indoors with increased efficiency and optimal resource utilisation.

Investigation and Analysis

This investigation and analysis focus on the nutrient delivery system, with a specific examination of two distinct pump types: peristaltic and centrifugal pumps. My measuring the flow rates of the pumps it can be determined which pump can be utilised for nutrient dosages as this study aims to provide valuable insights into optimising nutrient delivery in controlled agricultural environments.

Flow rate test

Materials:

- 1× 12v Peristaltic pump.
- 1 × 12v Centrifugal pump.
- 1× 3mm silicone tubes.
- 1× LCD shield.
- 1× Motor shield v2.
- 1× Arduino uno.
- 1× 500mL measuring cup.
- 1× 2L container.
- 1× Cloth.
- 1× Phone stopwatch.
- 2× plastic pegs.
- 1× Sellotape.
- 1× 12v power supply adapter plug.

Figure 12: Centrifugal pump with tube attached.

Figure 13: The peristaltic pump.

Variables:

Table 2: *Variables table for peristaltic pump.*

Variable	**Peristaltic Pump**
Control	• The volume of the cup (500mL). • The height of the cup from the pump output. • The liquid being pumped. • The temperature of the liquid. • The diameter of the tubing used in the pump. • The calibration of the pump.
Dependent	• Time it takes to fill the 500mL cup.
Independent	• The flow rate of the peristaltic pump *(Adjusted by the Duty cycle of the motor e.g., the percentage of flow rate through the Arduino IDE)*

Plan for pump test:

An experiment to evaluate the flow rate of the peristaltic pump will be investigated with a developmental board Arduino based system to compare the typical flow rate performance of a centrifugal pump to the flow rate of the peristaltic pump. The Arduino system must first be configured before connecting the pump to a water system to measure flow. Then the pump's data will be compared.

Method peristaltic pump test and centrifugal pump test:

1. Construct electronics by connecting LCD shield, Motor shield, and Arduino Uno to the peristaltic pump.
2. Fill a stable 2L container with room temperature water and place a cloth underneath.
3. Securely position the 500mL measuring cup on the cloth.
4. Attach the pump's silicone tubes to designated containers/cup, securing them with Sellotape and plastic pegs.
5. Set peristaltic pump duty cycle to 50% in Arduino IDE, start the pump, and timer as water fills the cup.
6. Stop the pump and timer when the cup reaches 500mL, recording the filling time for three trials.
7. Repeat steps 6-7 for duty cycles of 60% to 100%.
8. Drain peristaltic pump tubes, turn off the pump, and disconnect electronics.
9. Repeat steps 2-9 with the centrifugal pump, submerging it in the 2L container.
10. Analyse data by calculating average filling time for each duty cycle.

Evidence of peristaltic pump test:

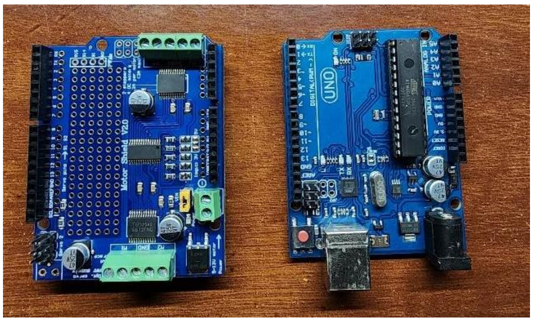


Figure 14: Motor Shield v2 & Arduino uno respectively.

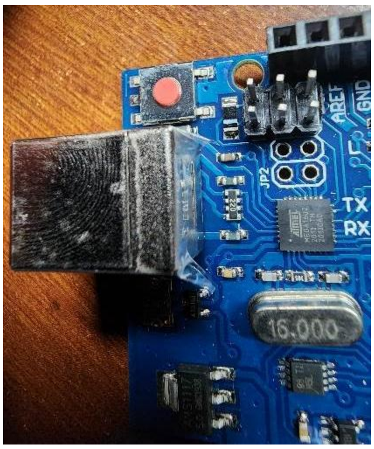


Figure 15: Sellotape on USB connector to stop conductivity from the Motor Shield that goes on top of the Arduino.

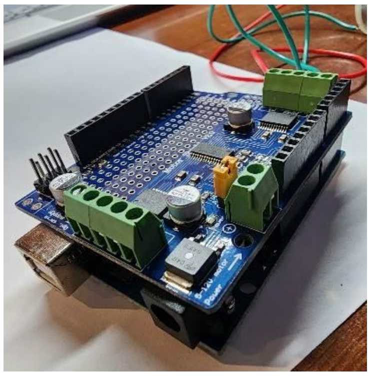


Figure 16: Motor Shield v2 & Arduino uno connected.

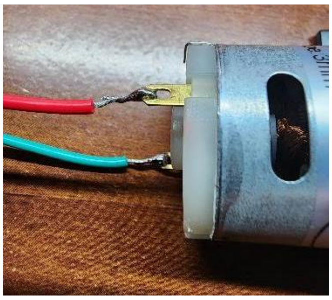


Figure 17: Soldered wire to peristaltic pump motor.

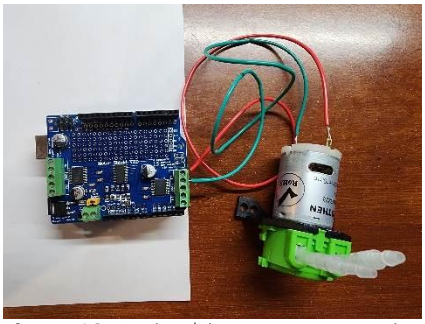


Figure 18: Peristaltic pump motor wired to circuit.

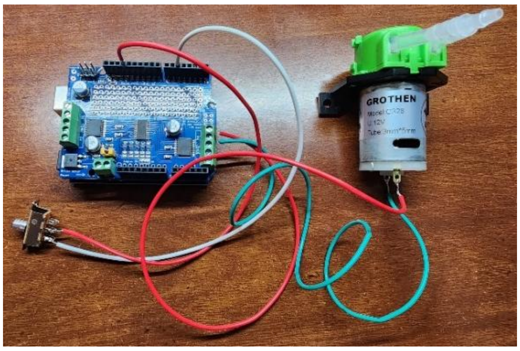


Figure 19: Initial plan to use a slide switch to turn motor on/off.

Figure 20: LCD shield to control motor with buttons and stopwatch. (Stopwatch was inaccurate due to the Arduino not having accurate timing)

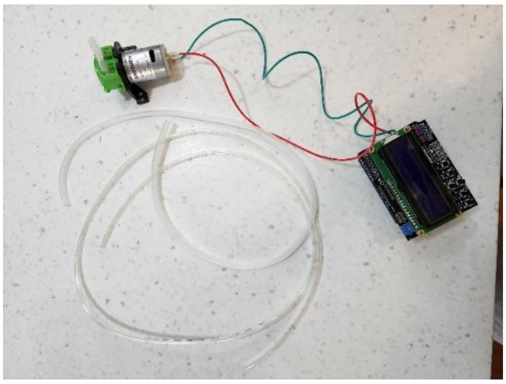


Figure 21: Silicon tubes being connected to the peristaltic pump.

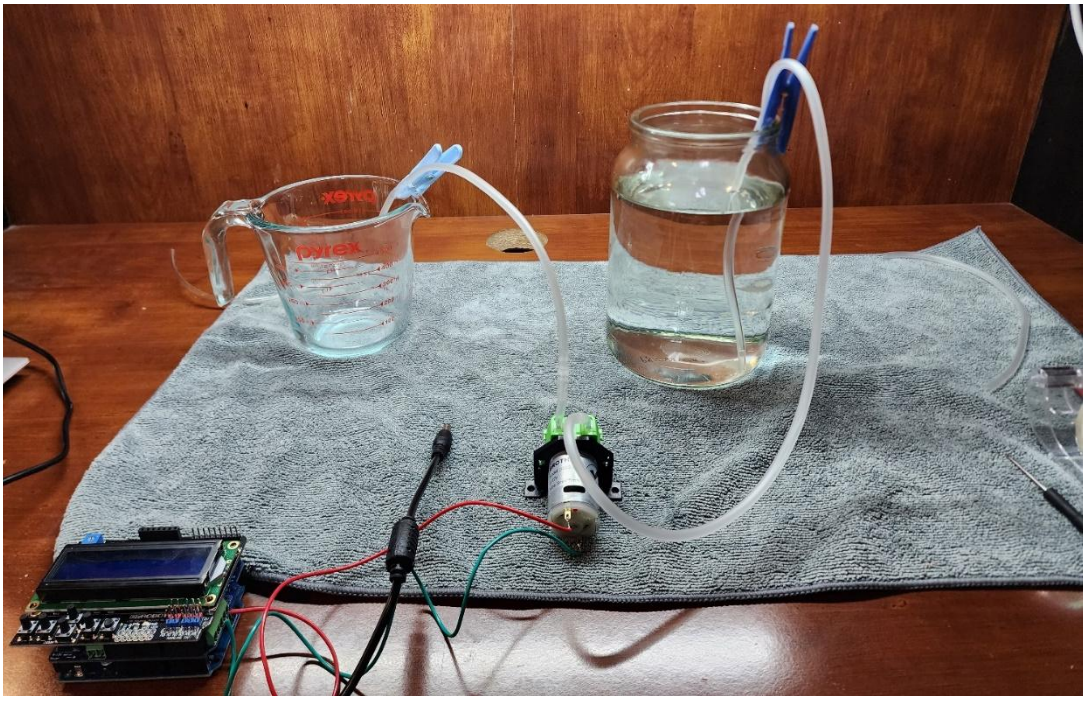


Figure 22: Full set up of the peristaltic pump flow rate test.

Evidence of centrifugal pump test:

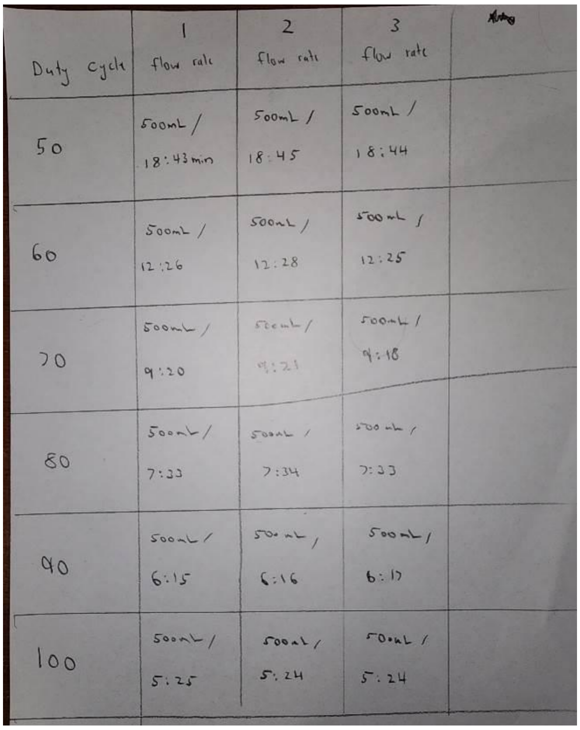
Figure 23: Tinning the write to fit better in the terminal blocks of the Arduino motor shield.

Figure 24: Centrifugal pump with tube and peg.

Figure 25: Full set up of the Centrifugal pump test.

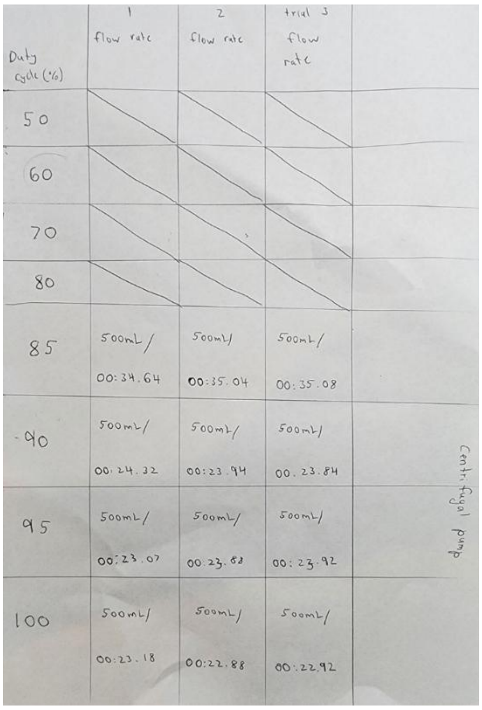
Figure 26: *Close look at Centrifugal pump test.*

Results for flow rate tests:



Duty cycle	1 flow rate	2 flow rate	3 flow rate	
50	500mL / 18:43min	500mL / 18:45	500mL / 18:44	
60	500mL / 12:26	500mL / 12:28	500mL / 12:25	
70	500mL / 9:20	500mL / 9:21	500mL / 9:18	
80	500mL / 7:33	500mL / 7:34	500mL / 7:33	
90	500mL / 6:15	500mL / 6:16	500mL / 6:17	
100	500mL / 5:25	500mL / 5:24	500mL / 5:24	

Figure 27: *Raw data noted on a piece of paper of* ***peristaltic*** *pump test. (mL/Minute).*



Duty cycle (%)	1 flow rate	2 flow rate	trial 3 flow rate	
50				
60				
70				
80				
85	500mL/ 00:34.64	500mL/ 00:35.04	500mL/ 00:35.08	
90	500mL/ 00:24.32	500mL/ 00:23.94	500mL/ 00:23.84	Centrifugal pump
95	500mL/ 00:23.07	500mL/ 00:23.[illegible]	500mL/ 00:23.92	
100	500mL/ 00:23.18	500mL/ 00:22.88	500mL/ 00:22.92	

Figure 28: *Raw data noted on a piece of paper of* ***centrifugal*** *pump test. (mL/Minute).*

Table 3: Calculated data for the average flow rate given three trails of the **peristaltic** pump, *sourced figure 14.*

Duty Cycle (Percentage of PWM signal)	**Flow rate trail 1 (mL/s) (5.D.P)**	**Flow rate trail 2 (mL/s) (5.D.P)**	**Flow rate trail 3 (mL/s) (5.D.P)**	**Average Flow (mL/s) (5.D.P)**
50	500/1123 = 0.44524	500/1125 = 0.44444	500/1124 =0.44484	0.44484
60	500/746 = 0.67024	500/748 = 0.66845	500/745 = 0.67114	0.66994
70	500/560 = 0.89286	500/561 = 0.89127	500/558 = 0.96057	0.91490
80	500/453 =1.10375	500/454 = 1.10132	500/455 = 1.09890	1.10132
90	500/375 = 1.33333	500/376 = 1.32979	500/377 = 1.32626	1.32979
100	500/325 = 1.53846	500/324 = 1.54321	500/324 = 1.54321	1.54163

Table 4: Collated data for the average flow rate given three trials of the **peristaltic pump**.

Duty Cycle (Percentage of PWM signal)	**Flow rate trail 1 (mL/s) (5.D.P)**	**Flow rate trail 2 (mL/s) (5.D.P)**	**Flow rate trail 3 (mL/s) (5.D.P)**	**Average Flow (mL/s) (5.D.P)**
50	0.44524	0.44444	0.44484	**0.44484**
60	0.67024	0.66845	0.67114	**0.66994**
70	0.89286	0.89127	0.96057	**0.91490**
80	1.10375	1.10132	1.09890	**1.10132**
90	1.33333	1.32979	1.32626	**1.32979**
100	1.53846	1.54321	1.54321	**1.54163**

Table 5: *Calculated data for the average flow rate given three trails of the* ***centrifugal*** *pump, sourced figure 15.*

Duty Cycle (Percentage of PWM signal)	Flow rate trail 1 (mL/s) (5.D.P)	Flow rate trail 2 (mL/s) (5.D.P)	Flow rate trail 3 (mL/s) (5.D.P)	Average Flow (mL/s) (5.D.P)
50	N/A	N/A	N/A	N/A
60	N/A	N/A	N/A	N/A
70	N/A	N/A	N/A	N/A
80	N/A	N/A	N/A	N/A
85	500/34.64 = 14.43418	500/35.04 = 14.26941	500/35.08 = 14.25314	14.31891
90	500/24.32 = 20.55921	500/23.94 = 20.88555	500/23.84 = 20.97315	20.80597
95	500/23.07 = 21.67317	500/23.88 = 20.93802	500/23.92 = 20.90301	21.17140
100	500/23.18 = 21.57032	500/22.88 = 21.85315	500/22.92 = 21.81501	21.74616

Table 6: *Collated data for the average flow rate given three trials of the* ***centrifugal*** *pump.*

Duty Cycle (Percentage of PWM signal)	Flow rate trail 1 (mL/s) (5.D.P)	Flow rate trail 2 (mL/s) (5.D.P)	Flow rate trail 3 (mL/s) (5.D.P)	Average Flow (mL/s) (5.D.P)
85	14.43418	14.26941	14.25314	**14.31891**
90	20.55921	20.88555	20.97315	**20.80597**
95	21.67317	20.93802	20.90301	**21.17140**
100	21.57032	21.85315	21.81501	**21.74616**

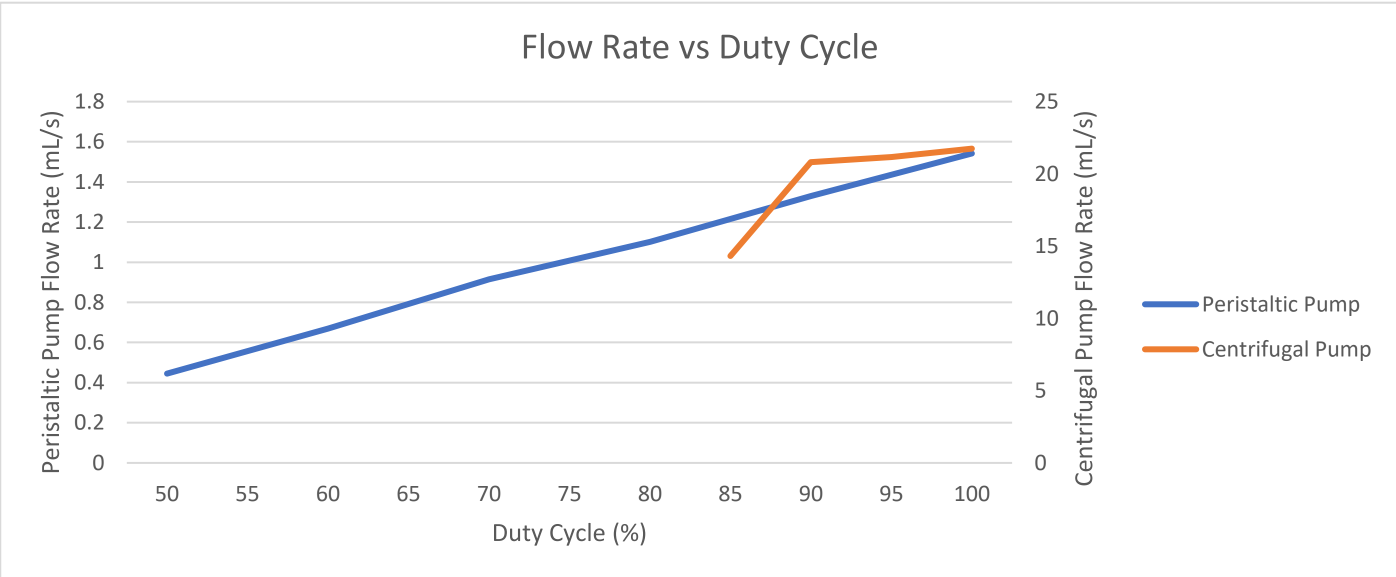


Figure 29: Graph of the average flow rate from ***peristaltic and centrifugal*** pump (Average Flow rate (mL/s) vs Duty cycle (%)

Note: The centrifugal pump capped at 85% duty cycle due to the pump needing a higher constant voltage to push the water up through the tube**.**

The data gathered confirmed that the peristaltic pumps maintain a constant and linear flow rate by squeezing fluid through the tubing at a rate directly proportional to the motor's rotational speed (*Figure 29*), which is helpful because I can predict the accuracy of the flow rate vs rpm (duty cycle). In contrast, centrifugal pumps have a non-linear relationship between impeller speed and flow rate. Therefore, a peristaltic pump is preferable in applications where height or flow rate variations could occur in the hydroponics system. The peristaltic pump(s) will be used for dosing the nutrients and pH in the system.

Design Development and Planning

Materials Cost analysis

Table 7: *Bill of materials for the project.*

Product Name	Quantity	Unit Cost ($)	Shipping Cost ($)	Total Cost ($)
Wire/Wire management				
Cable HU RND H/D 7.5A SLC RBR 10M RL BLK HP9558	1	9.95		9.95
Cable HU RND H/D 7.5A TINNED 10M RL RED WH3006	1	7.95		7.95
Cable HU RND 13X0.12MM L/D BLU 25M RL WH3004	1	7.95		7.95
Cable HU RND 13X0.12MM L/D YE; 25M RL WH3046	1	6.95		6.95
Cable HU RND H/D 7.5A TINNED 10M RL BLK WH5532	1	6.95		6.95
HeatShrink 3mm X 1.2m BLK	4	1.35		5.4
HeatShrink 1.5mm X 1.2m BLK	4	1.55		6.2
HeatShrink 2.5mm X 1.2m RED	3	1.75		5.25
HeatShrink 2.5mm X 1.2m CLR	1	0.3		0
Electronics				
For Arduino PH Value Detection Sensor Electrode Probe PH Module PH Regulator	1	18.92		18.92
Green 12V Peristaltic Pump Long Service Life Liquid Dosing Pump Gadget Supplies	4	17.42		69.68
TEEMO Aquarium Submersible Water Pump: 2500L/H 45W Quiet Adjustable with 1.9M Power Cord for Hydroponics | Garden Waterfall | Pond | Fish Tank | Fountain	1	30		30
WeMos D1 R32 (ESP32) Development Board	1	23.7		23.7
Adafruit Motor/Stepper/Servo Shield for Arduino v2 Kit	1	40.7		40.7
Nutrients/Hydroponics supply				
ISN Nutrients Supreme Grow - A&B (5L)	1	79.95		79.95
Liquid Science pH Lower Solution (1L)	1	19.95		19.95
Liquid Science pH Raise Solution (1L)	1	17.95	14.99	17.95
100pcs Hydroponics Plant Grow Net Cup Mesh Pot Basket Aeroponic Mesh Pots	1	7.2	7.99	15.19
100 Grow Starter Cubes Plug Hydroponic Rockwool Grow Media Cloning D	1	22.99	7.99	30.98
Sealants and Gap Fillers				
Selleys 475g No More Gaps Interior Multipurpose Gap Filler	2	5		10
Selleys 410g White No More Gaps Bathroom And Kitchen Gap Filler	1	10.99		10.99
Protek 125ml Type N Blue Cement Solvent	1	6.29		6.29
Safety, PVC Pipes and Fittings (Physical build)				
Storage Barrel Handy Storage 35L 460X420X420mm	1	30.54		30.54

Gloves Mechanix Wear SpeedKnit XL BUT-S2DE-58-010	1	16.15		16.15
Press PVC Pipe Class 12 Holman 40mm 1m PVP4012-1	1	15.99		15.99
Tube Clear Vinyl Pope 19mm X 90ck 1011921	1	7.35		7.35
Holman 150mm x 3m PVC DWV Pipe	1	83.1		83.1
Holman 90mm Storm PVC Finish Collar with Grate	1	14.5		14.5
Holman 50mm x 1m PVC DWV Pipe	1	12.95		12.95
Holman 15mm x 3m Class 18 PVC Pressure Pipe	2	8.95		17.9
Holman 15mm PVC Coupling	1	1.55		1.55
Total Cost:		**630.93**		

Design Schedule – Gantt Chart

Table 8: *Gantt Chart time management chart*

	1	2	3	4	5	6	7	8	9	10	Holidays		1	2	3	4	5	6	7	8
Preliminary Planning																				
Circuit Schematic Design																				
Initial Breadboarding Circuit (Arduino)																				
Initial Breadboarding Circuit (Wemos D1 R32)																				
Arduino IoT integration																				
Hydroponics Physical Build																				
Design Enclosure																				
Fabricate Enclosure																				
Order Materials																				
Prepare plants																				
Testing																				
Grow plants (Lettuce)																				

Product Design

Brainstorm – Lotus Diagram

***Table 9:** Lotus diagram brainstorm*

Machine learning (Advanced)	Regular updates	Water analysis	Continuous monitoring	Water recycling	Organic nutrients	Vertical farming	Sensor-driven irrigation	Irrigation timing	LED grow lights	Closed-loop systems	Drip irrigation	LED technology	Light spectrum control
Fine-Tuning Automation Algorithms	User feedback on components	Automated adjustments	Nutrient schedules	Precision dosing	Buffer solutions	NFT/DWC Size of reservoirs	Nutrient delivery	pH sensor EC sensor	LED, UVA UVB light	Lighting systems	Energy-Efficient Lighting	Automated lighting	Light reflectors Local
Biological controls	Sensor calibration	Addressing Nutrient Imbalances	Nutrient sensors	Grow beds	Reservoirs	Pumps and tubing	Sensors	Energy-Efficient Design	IoT Integration for Monitoring	Space Optimisation	Nutrient recycling	Vertical farming	Plant spacing
Optimising Light Intensity	Managing Pest and Disease	Solving pH Fluctuations	Innovative Growth Techniques	Sustainability and Eco-Friendly Practices	Smart Irrigation Solutions	Energy-Efficient Design	Plant Health Optimisation	System Components	Water Recycling and Conservation	Optimal Nutrient Usage	Minimising Environmental Impact		
Starter kits Tech partnerships	Promoting Sustainable Gardening	Sharing DIY Kits	Challenges and Solutions	Central Concept	Efficient Resource Management	Drip System Aeroponics System	Individualised control Mist or fog delivery	Precise nutrient delivery high-pressure pump					
Eco-friendly materials	Hands-on learning School initiatives	Educational Workshops	Education and Outreach	Making an automated Hydroponics system	Hydroponic Techniques	Wick System Deepwater Culture (DWC)	Passive system Buoyant platform	Capillary action Monitoring and maintenance					
Exhibitions at the school	Showcasing Technological Innovations	Community Involvement	Creative Enclosure Designs	Automation and Control	Plant Varieties and Cultivation	Ebb and Flow System Nutrient Film Technology (N.F.T.)	Controlled flooding Continuous flow	Timing flexibility leafy greens and herbs.					
Aesthetic Enclosure Solutions	Insulation	Sealed enclosures	Weatherproofing Techniques	Central control	Microcontroller Integration	pH and EC Regulation	Sensor Networks	Monitoring plant health	Flower and Herb Growth	Year-Round Cultivation	Space efficiency		
Space-Saving Enclosure Ideas	Shelving systems	Functional Enclosure Materials	Remote Monitoring	Programmability	Climate Control Integration with IoT	Timed Nutrient Delivery	Real-time data	Nutrient-Rich Produce	Modular designs	Vertical Gardening Ideas			
PVC pipe	Living walls	Steel frames	Alerts and notifications	Remote monitoring	LED grow lights	Solar power	Mobile apps and web interfaces	Alarms and alerts	Crop Selection	Crop suitability	Controlled environment		
Artistic installations	Compact enclosures	Wood shelves	Automated adjustments	Continuous monitoring	Synchronisation	Reduced human intervention	pH regulation	Precision scheduling	Culinary herbs	Indoor systems	Aesthetics		

Brainstorm – Sketches/images

Ai generated images:

Figure 30: *AI generated image of vertical hydroponics (clipdrop.co, 2023)* https://clipdrop.co/stable-doodle

Figure 31: *AI generated image of hydroponics multiple vertical hydroponics (clipdrop.co, 2023)*

Figure 32: *AI generated image of sketch of hydroponics, vertical hydroponics (clipdrop.co, 2023)*

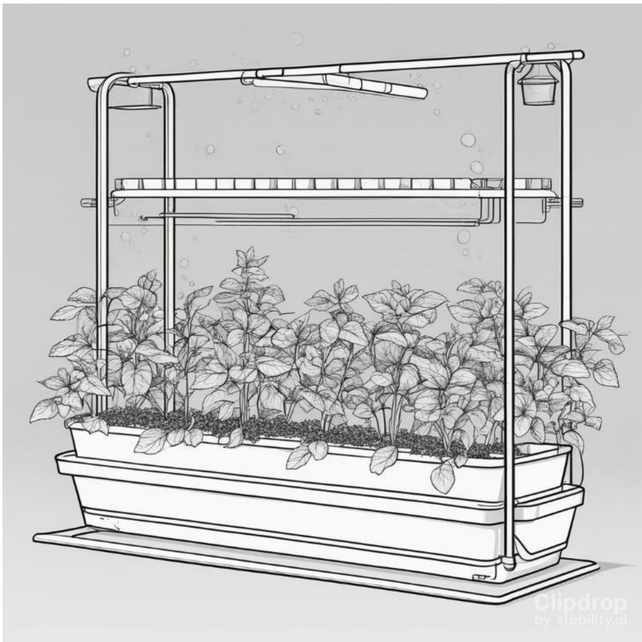


Figure 33: *AI generated image of sketch of hydroponics, horizontal hydroponics (clipdrop.co, 2023)*

Figure 34: *AI generated image of sketch of hydroponics, horizontal and vertical hydroponics in a home setting (clipdrop.co, 2023)*

Rough Design Exploration Sketches

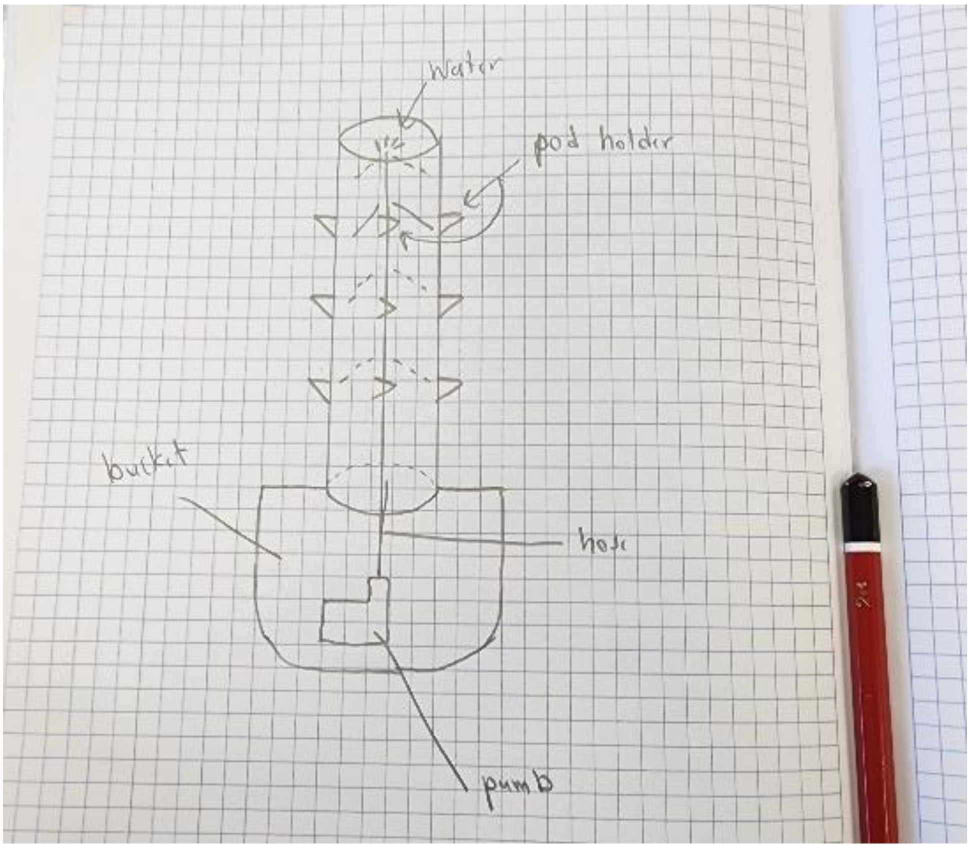


Figure 35:

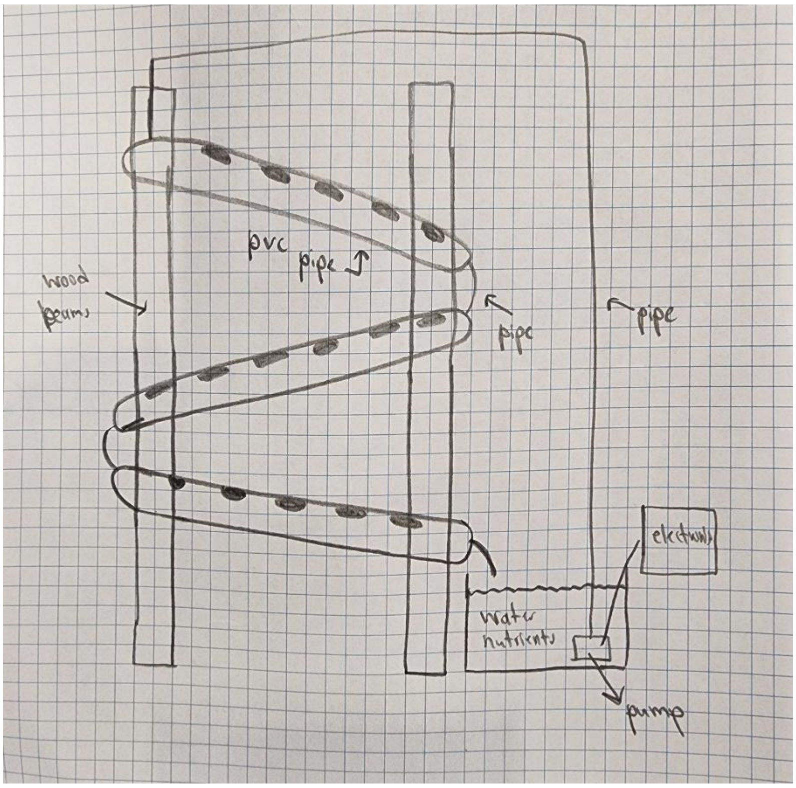


Figure 36:

Scaled Sketches Electronics Design

Adafruit Motor shield V2: (Did not work with the Wemos D1 R32

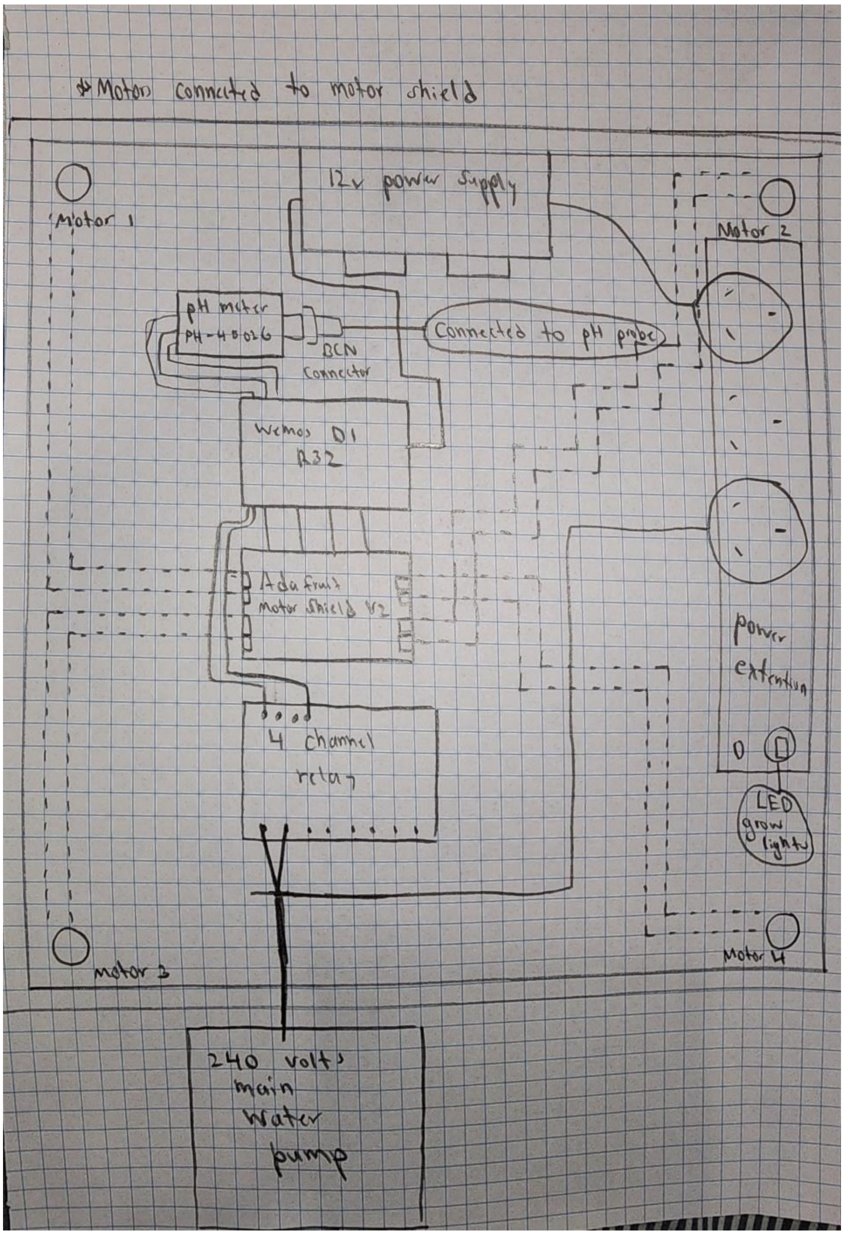


Figure 37: *Adafruit motor shield in a scaled diagram (1 square = 1cm)*

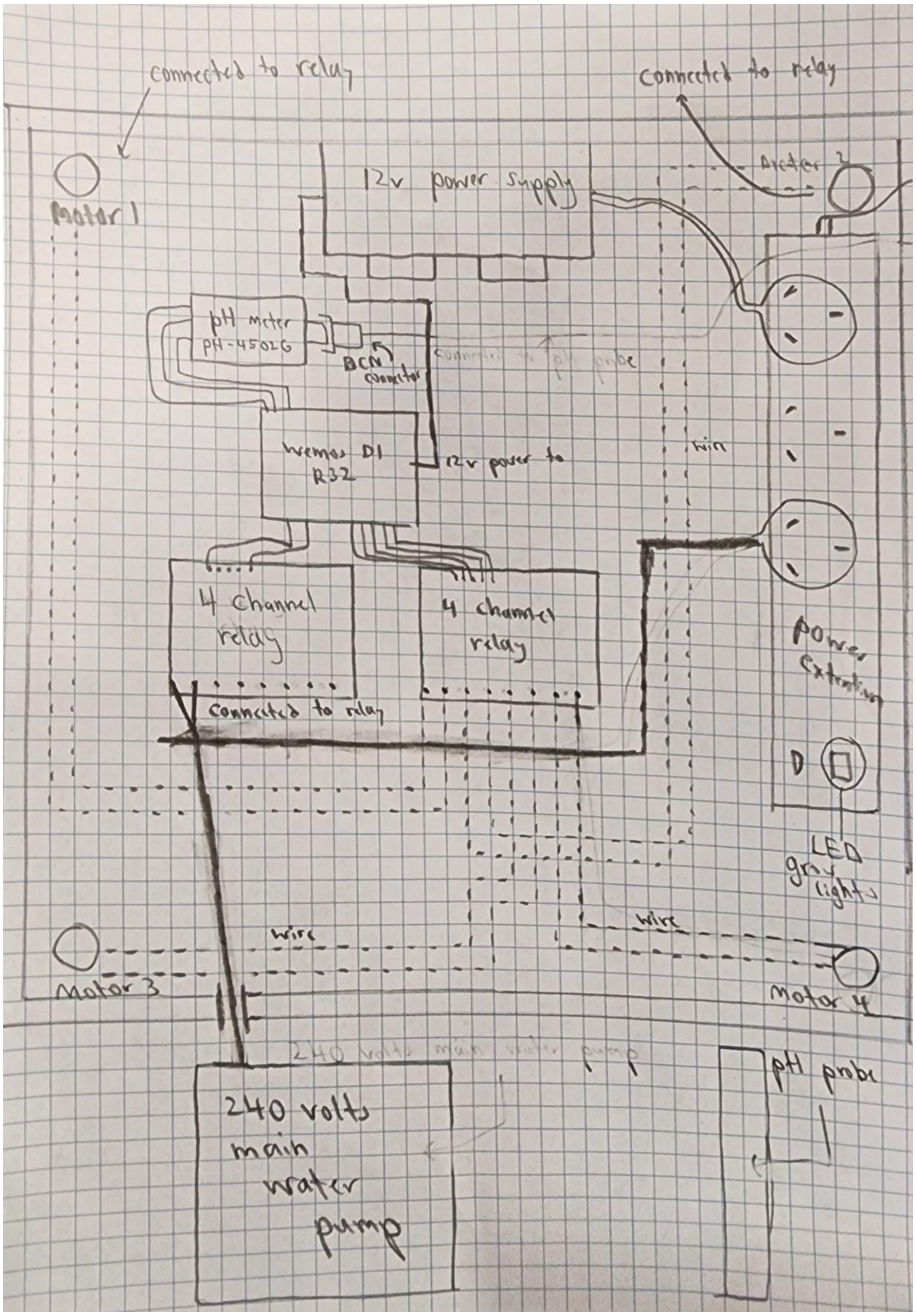


Figure 38: *Working system with 4 channel relays powering the motors (1 square = 1cm)*

*After the Adafruit was not compatible, had to re-solder and add 4 channel relays to make a working system.

Refined Scaled Sketch Full Sketch

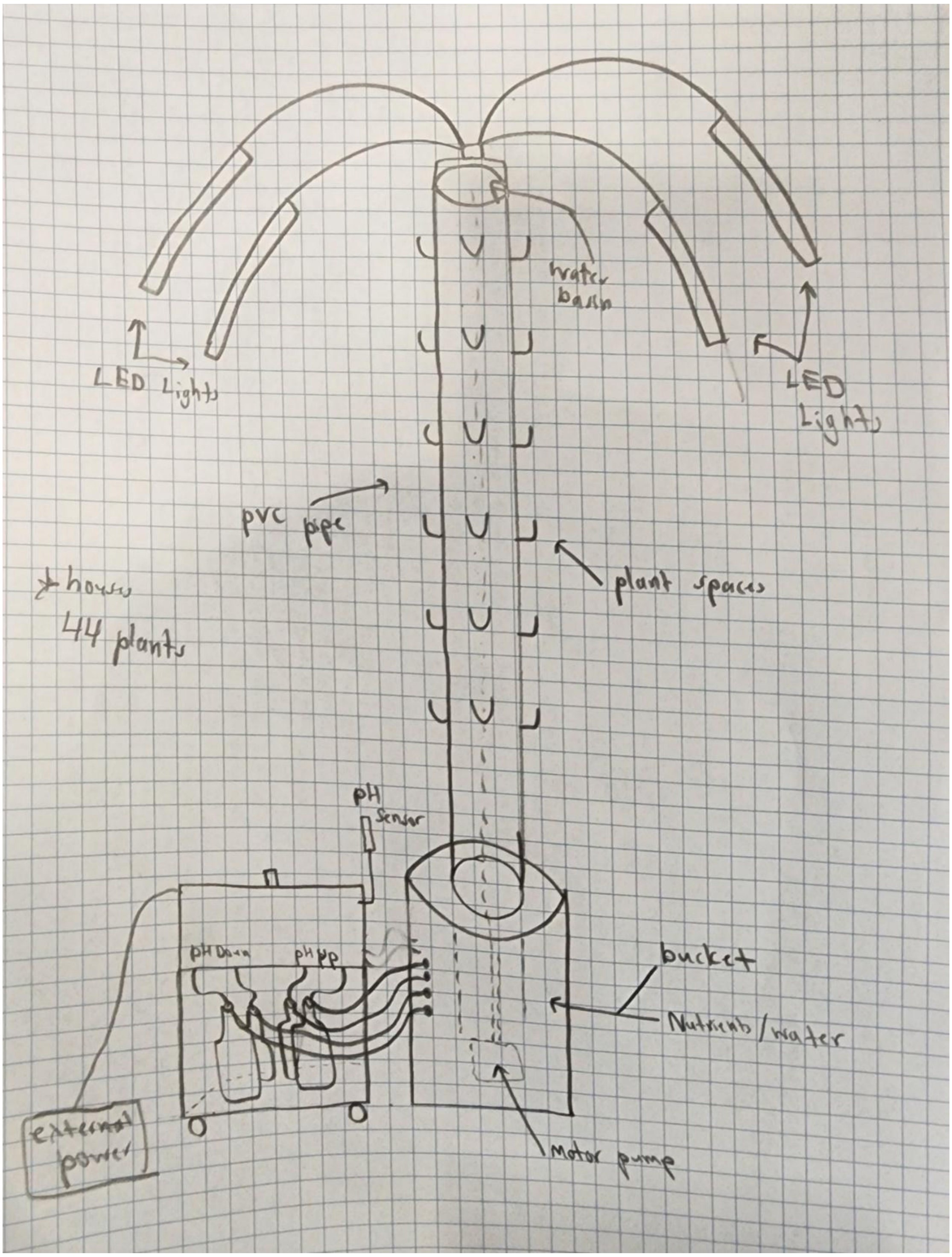


Figure 39: *Working system hydroponics (1 square = 5cm)*

Flow chart(s) of logic processes

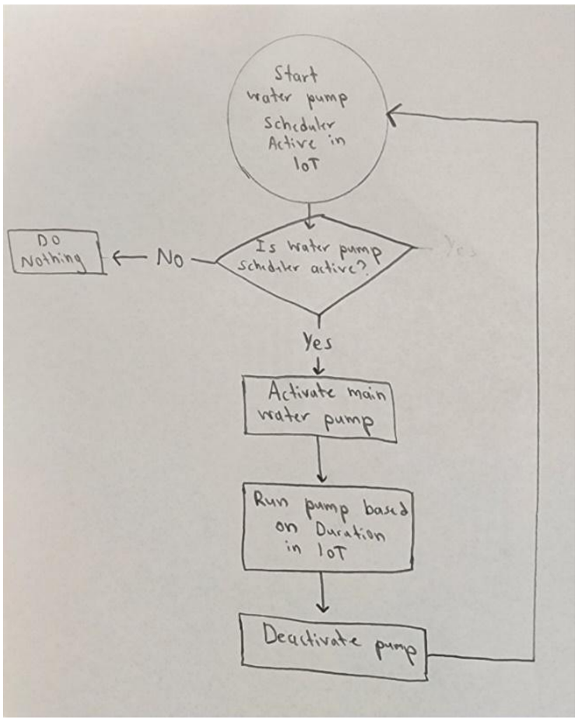


Figure 40: *Water pump flow chart*

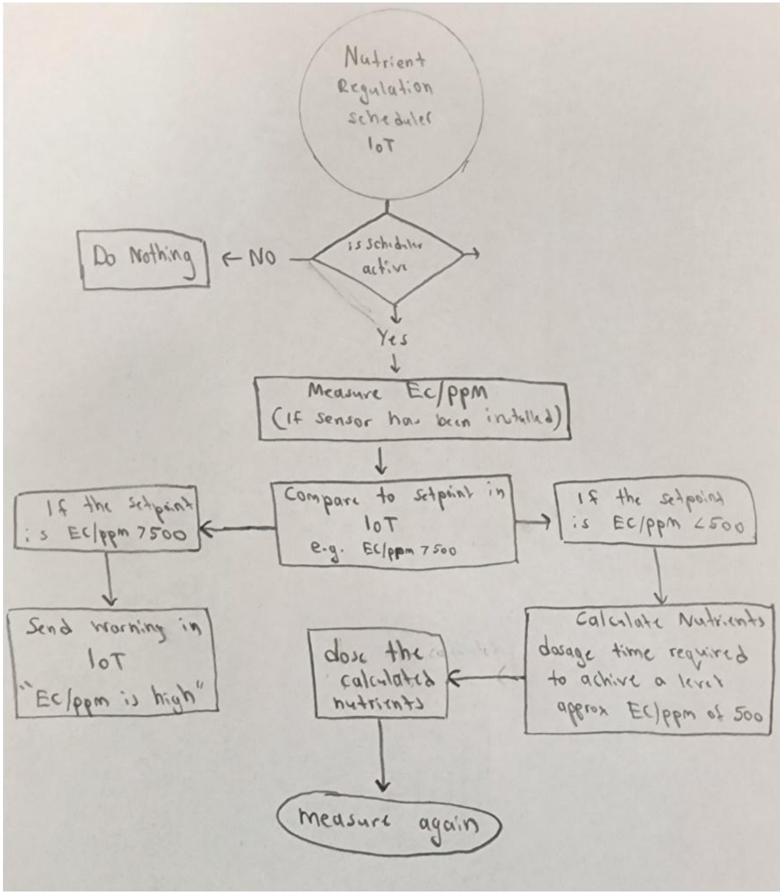


Figure 41: *Nutrients flow chart (coding logic)*

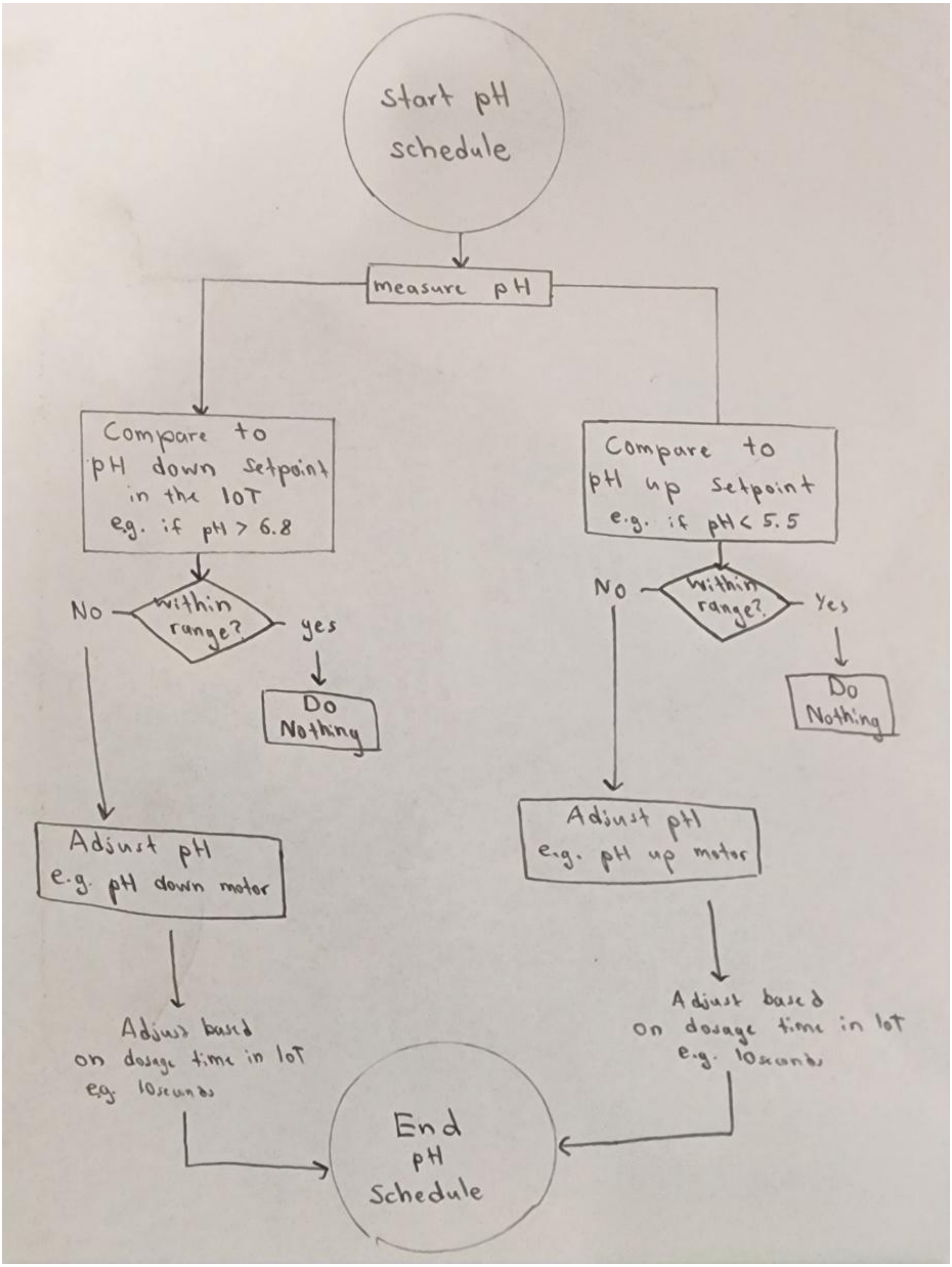


Figure 42: PH logic regulation flow chart (coding logic)

Final CAD design:

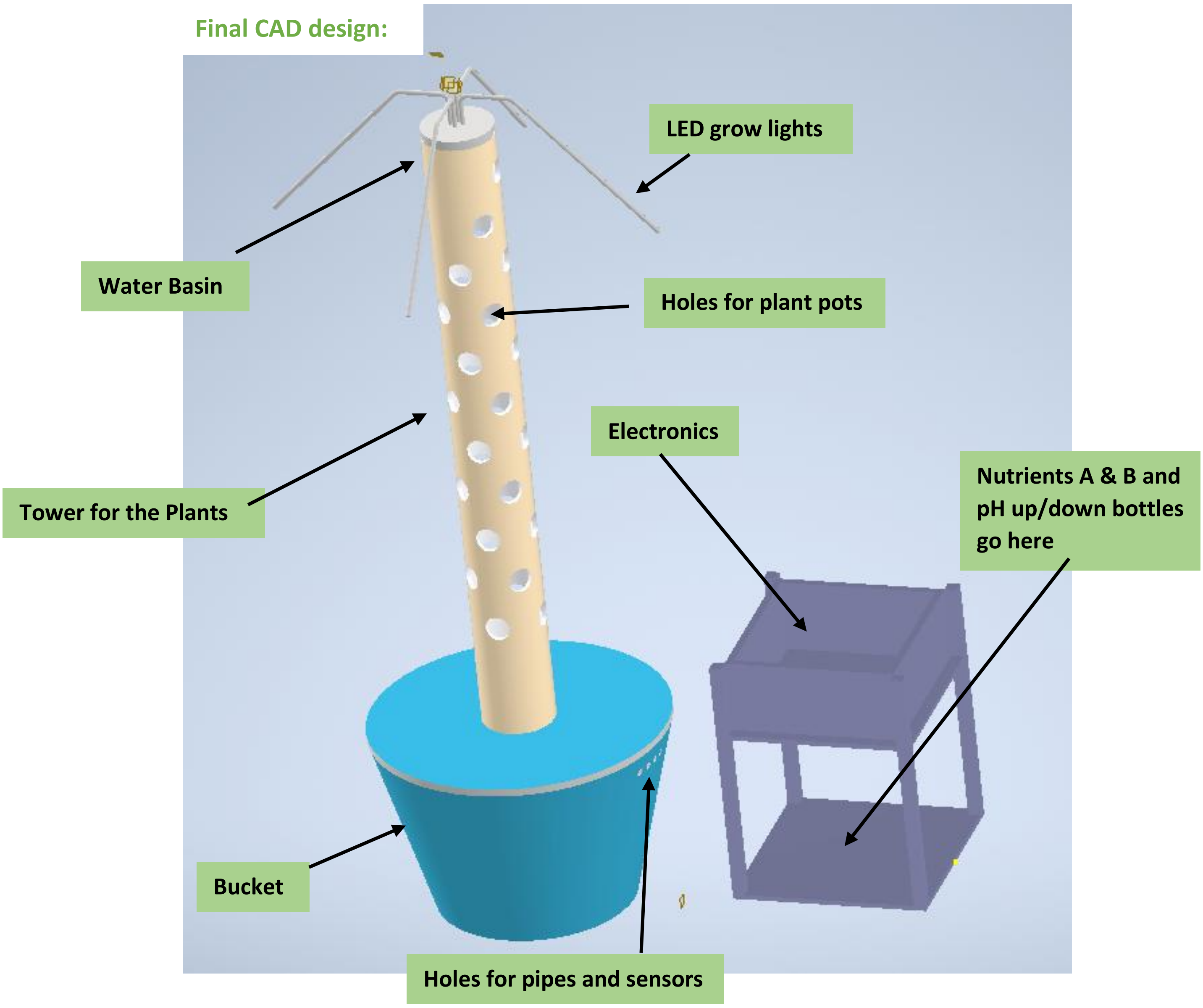


Figure 43: *Autodesk Inventor Pro CAD design of the hydroponics.*

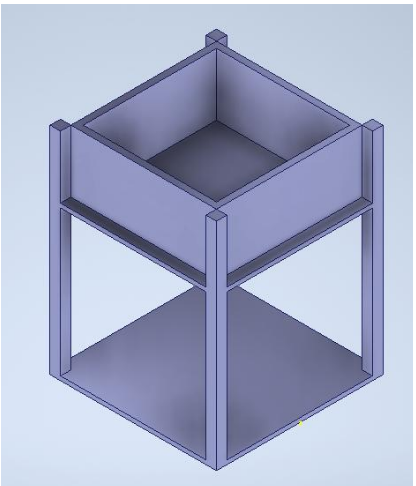


Figure 44: *Autodesk Inventor Pro CAD design of the electronics box hydroponics.*

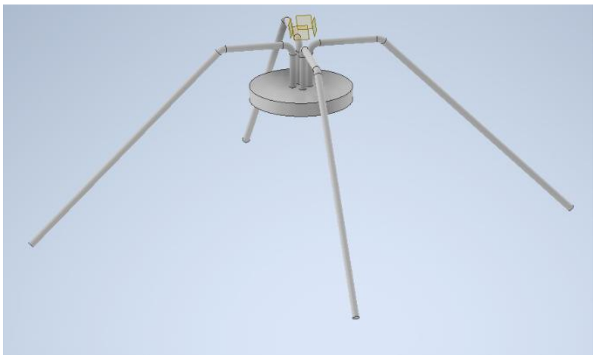


Figure 45: *Autodesk Inventor Pro CAD design of the LED Grow light lid.*

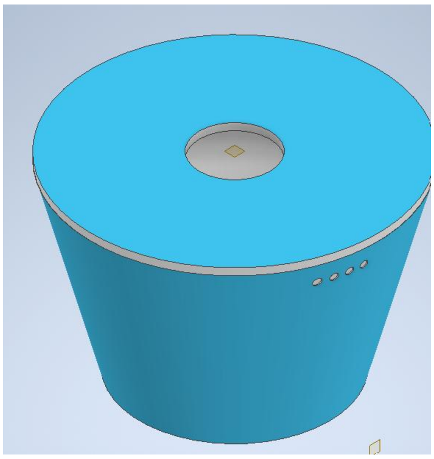


Figure 46: *Autodesk Inventor Pro CAD design of the bucket that waters the plant.*

IoT design (Internet of Things)

Brainstorm- possible IoT platforms

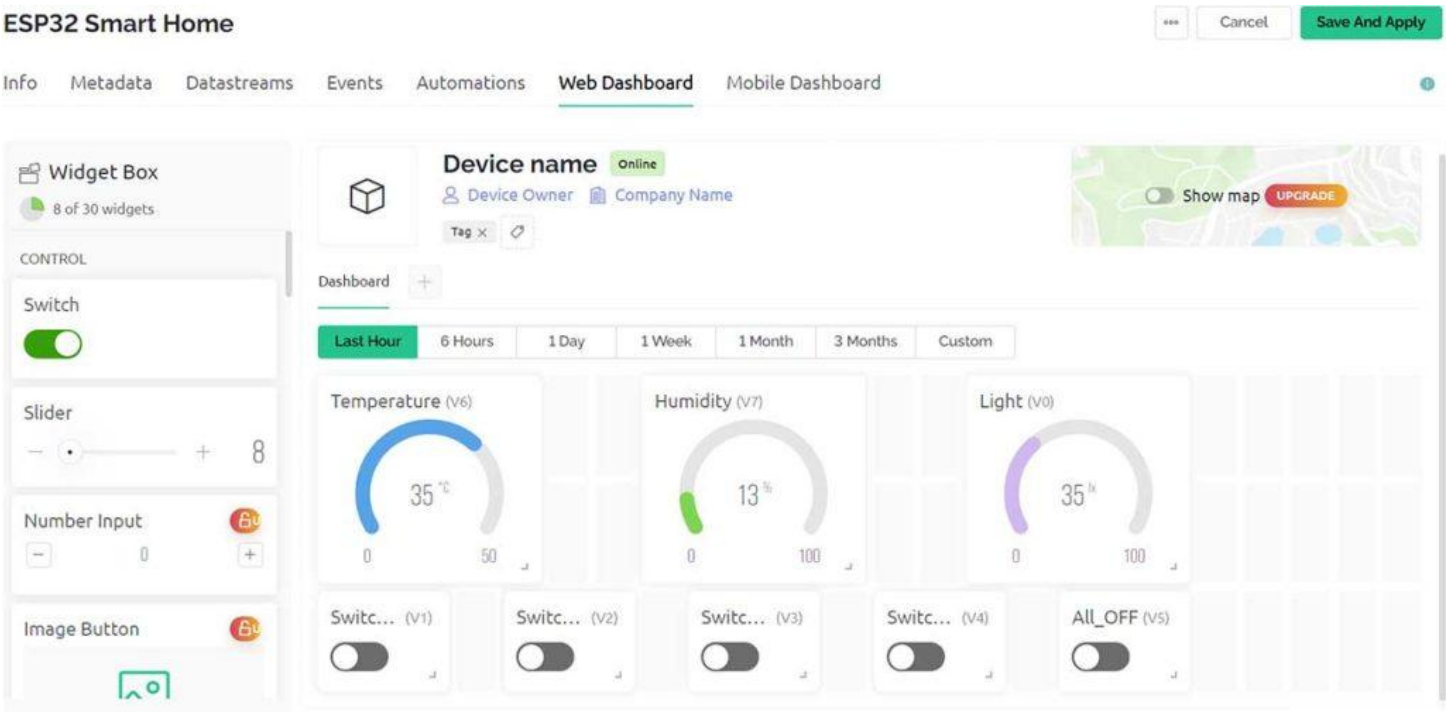


Figure 47: *Blynk Smart Home IoT example from web dashboard Blynk platform.* (Admin, 2022).

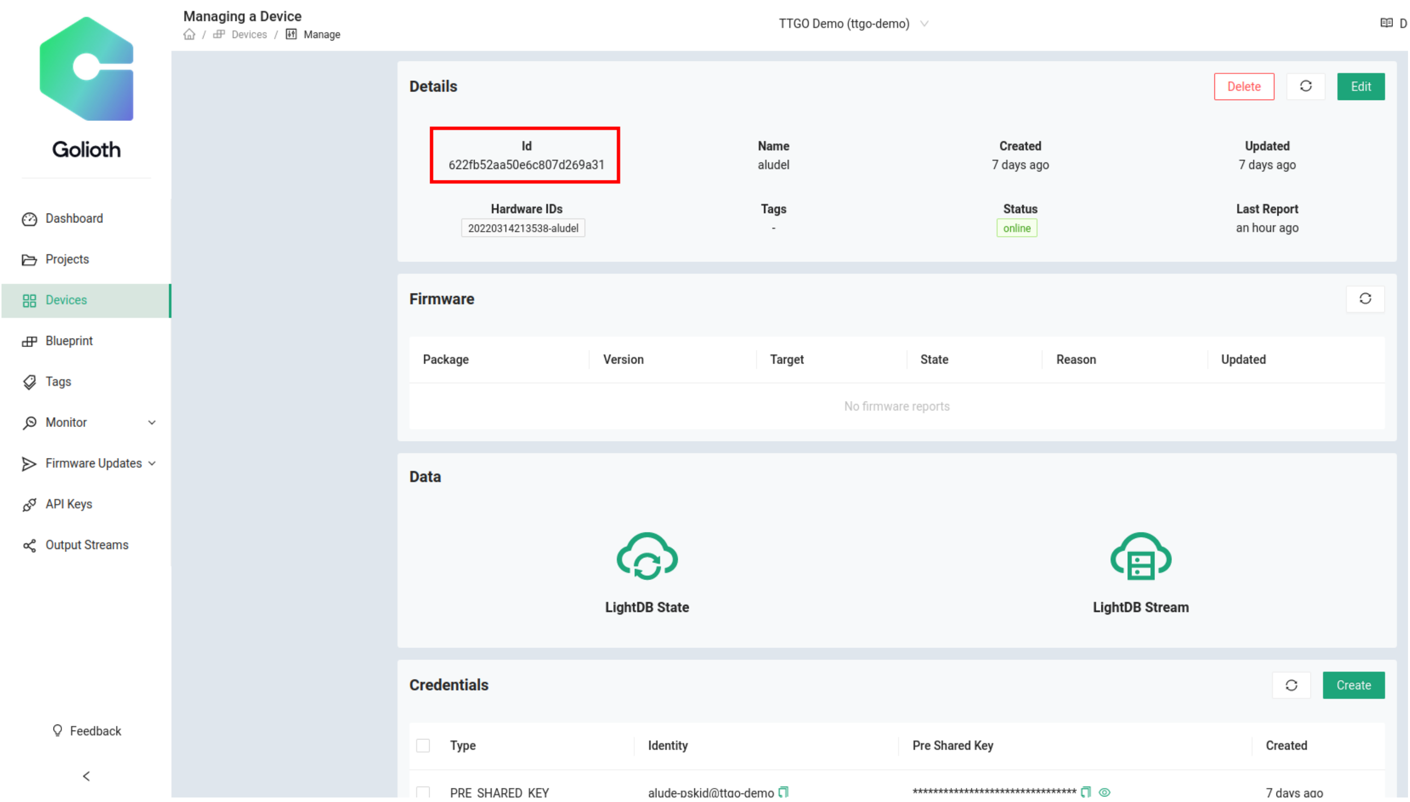


Figure 48: *Golioth IoT example from web devices platform able to make a dashboard.* (Szczys, 2022).

Figure 49: Arduino IoT example dashboard. (Söderby, *2023).*

IoT dashboard design for Arduino IoT cloud as chosen platform.

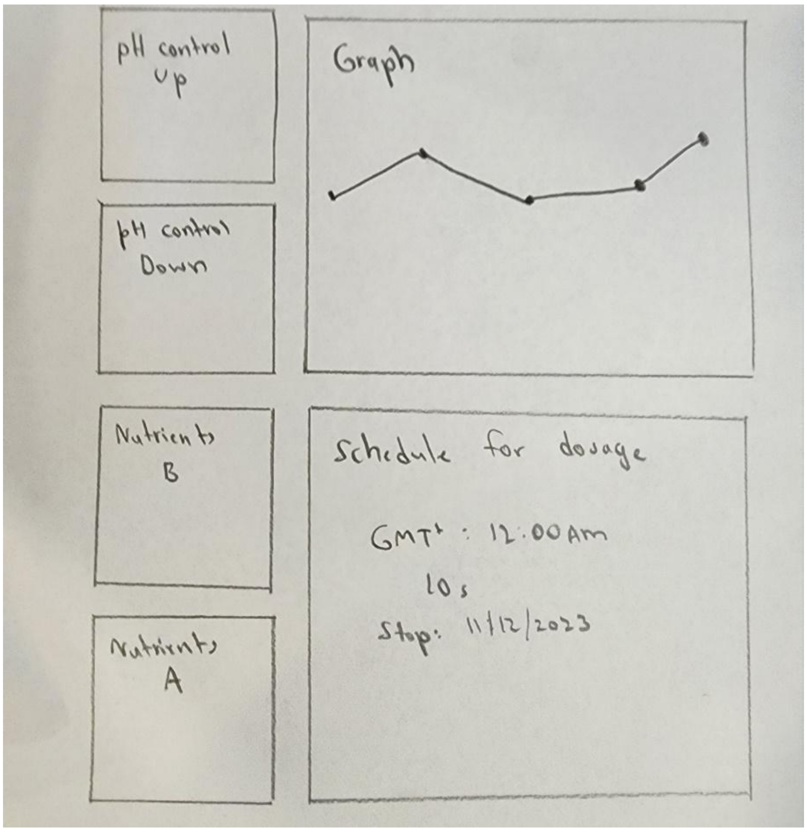


Figure 50: Arduino IoT dashboard sketch example dashboard.

Circuit Pin Maps

Arduino Uno

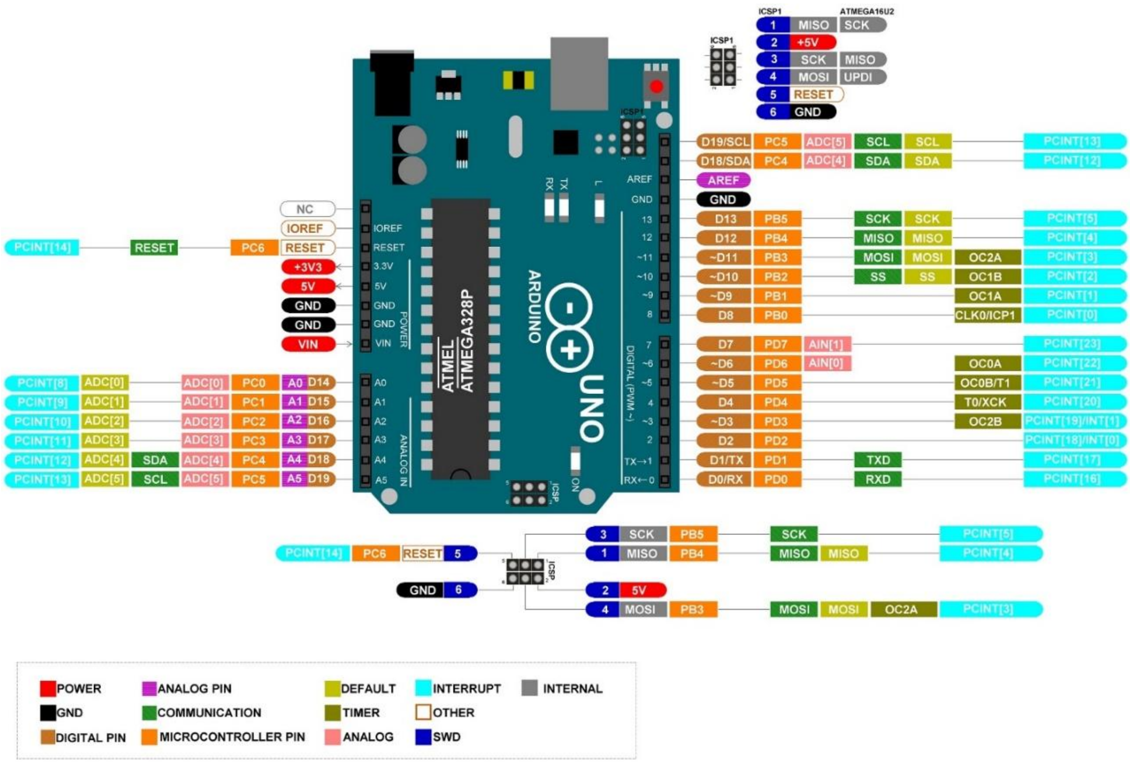


Figure 51: *Arduino UNO Pinout, Specifications, Board Layout, Pin Description (Teja, 2021)*

Wemos D1 R32

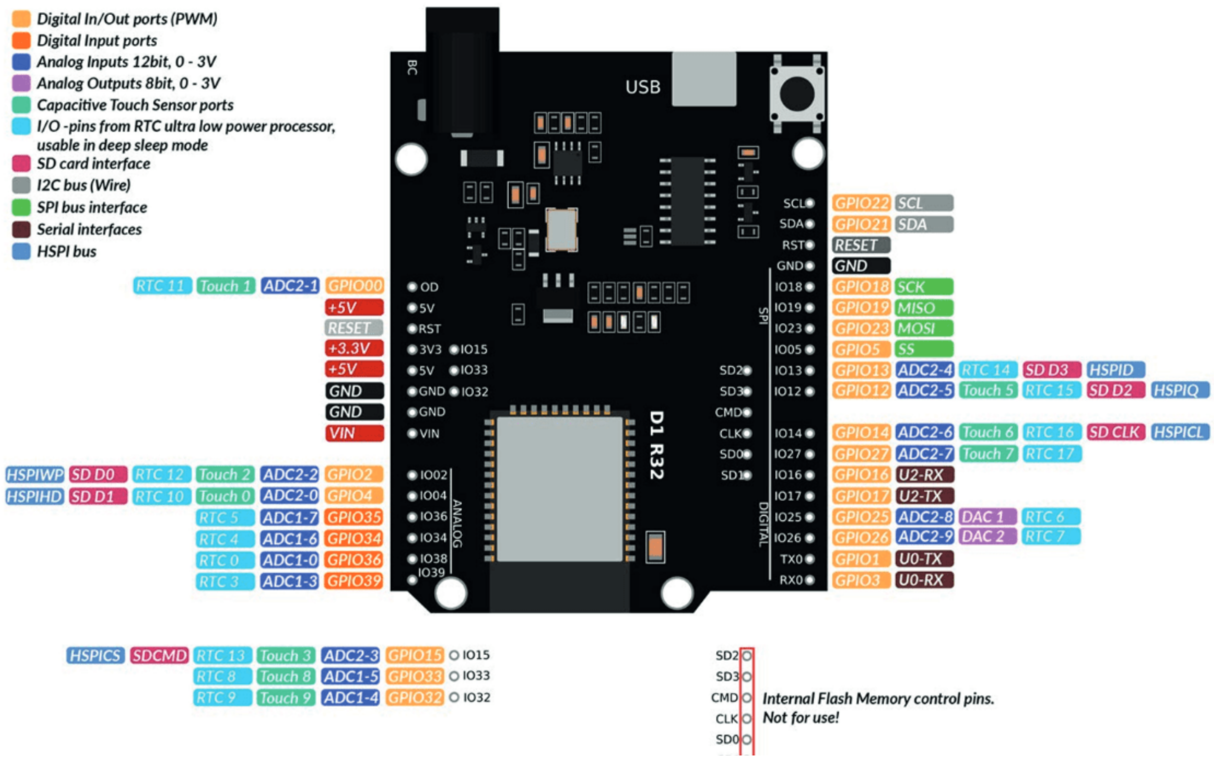


Figure 52: *Wemos D1 R32 Pinout, Specifications, Board Layout (C, 2022)*

Wemos D1 R32 circuit diagram

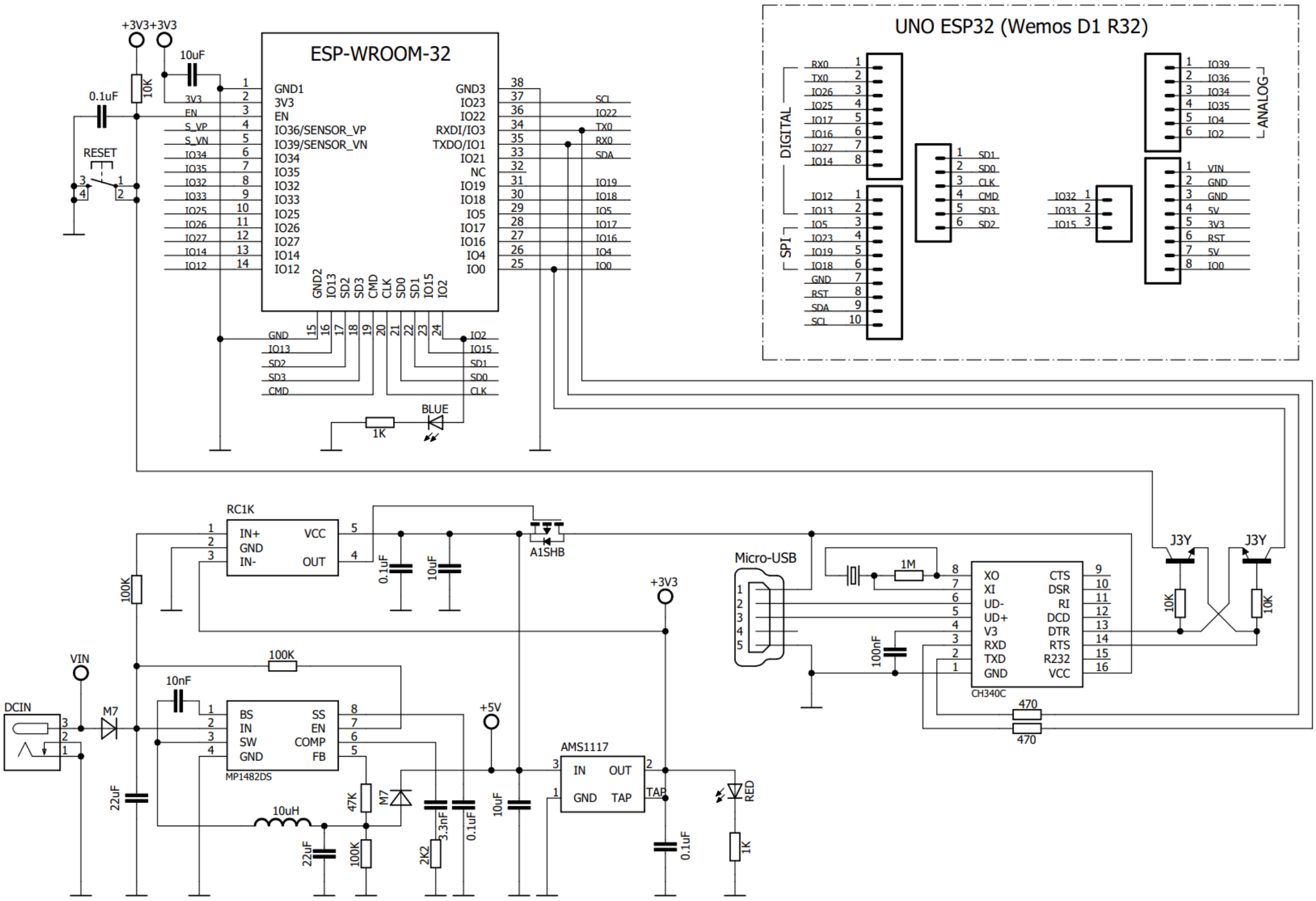


Figure 53: Wemos D1 R32 – IO pin assignment for Arduino UNO compatibility (ROBERT, 2022)

Circuit Schematics

pH sensor and LED light schematic

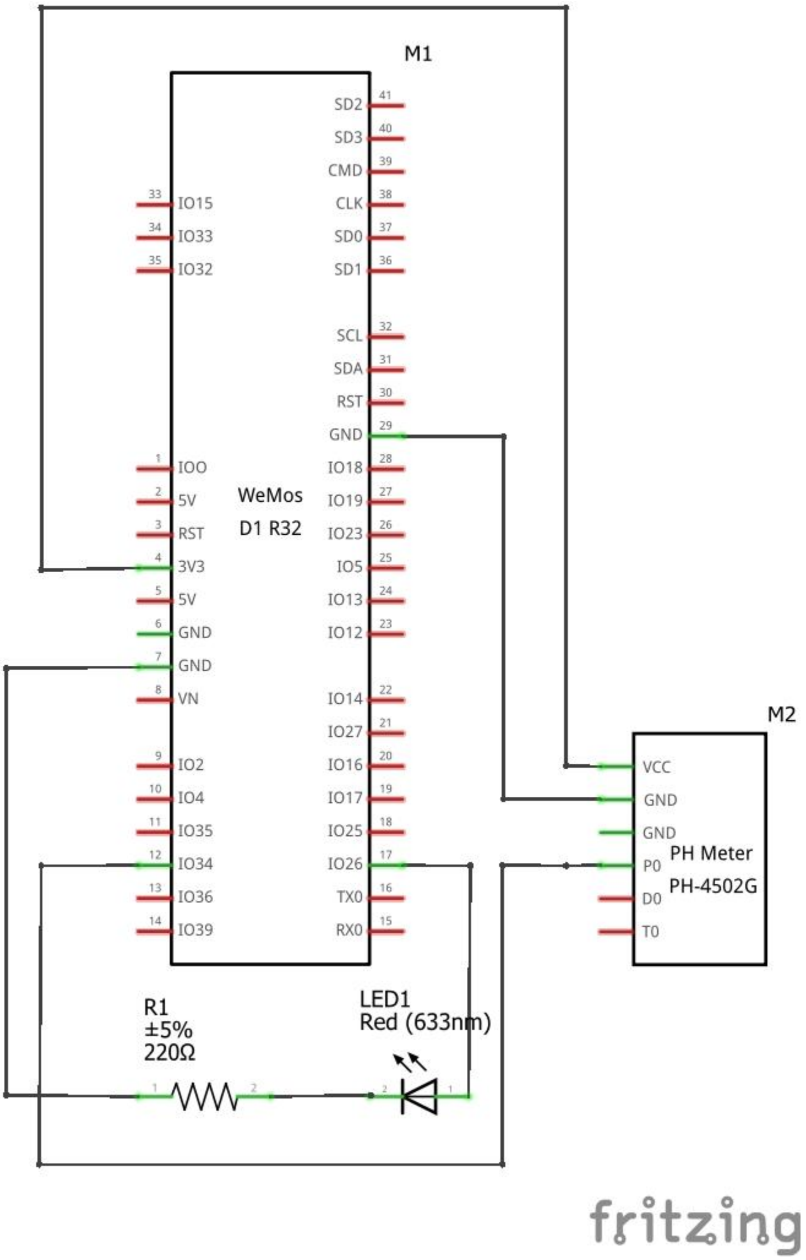


Figure 54: A schematic of the pH sensor (pH meter PH-4502G) and LED wired to the main circuit board (Microcontroller WEMOS D1 R32)

Wemos, Motor shield ‘w’ 4 motors, pH sensor and LED light schematic

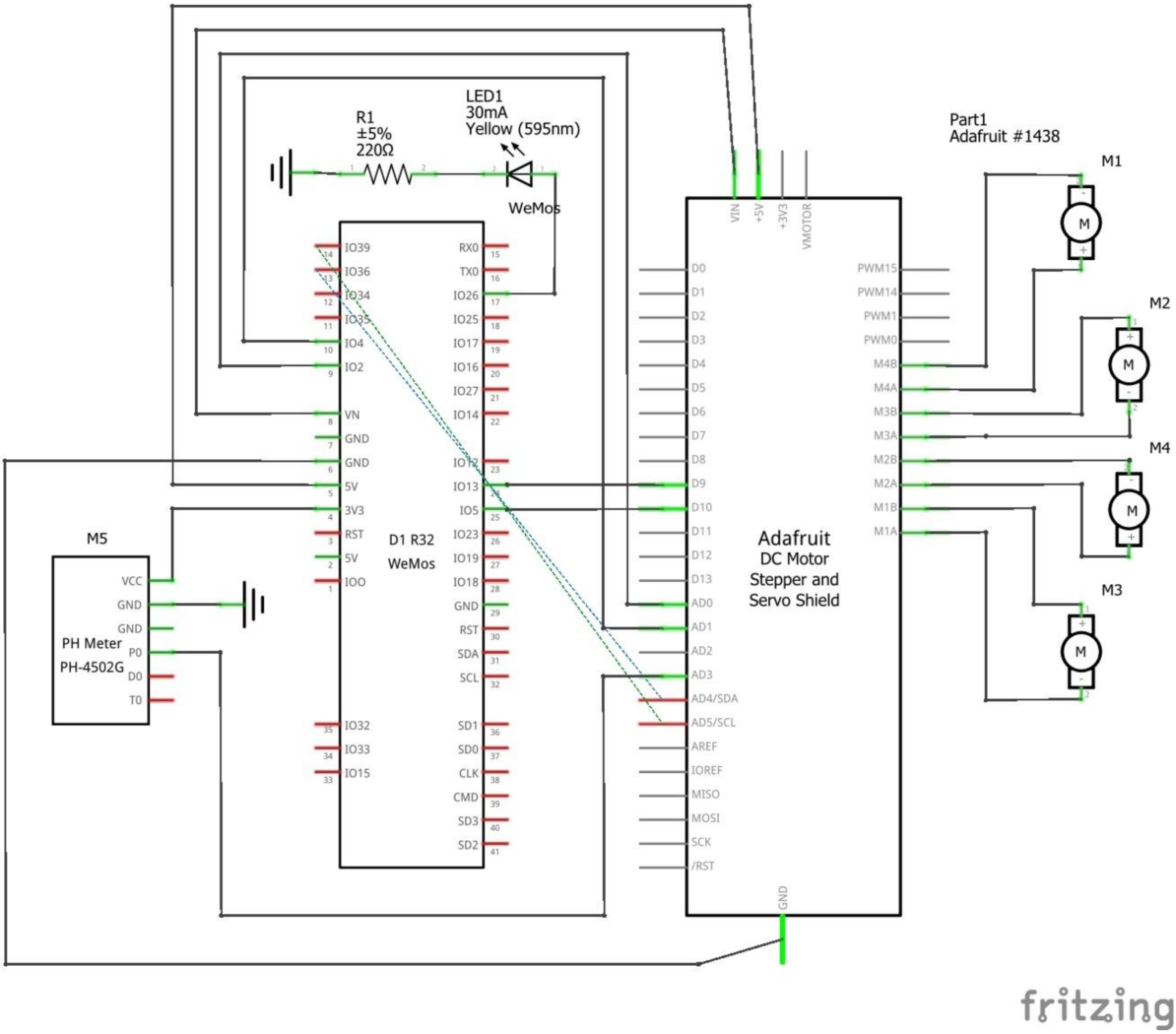


Figure 55: A schematic of the Wemos, Motor shield ‘w’ 4 motors, pH sensor and LED light schematic

Prototyping

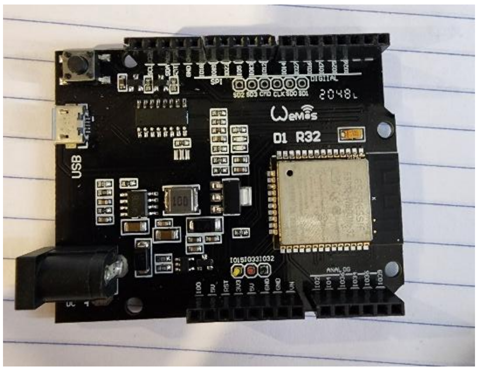


Figure 56: *Wemos D1 R32 micro-controller board based on the WROOM ESP32 and has the Arduino Uno form factor.*

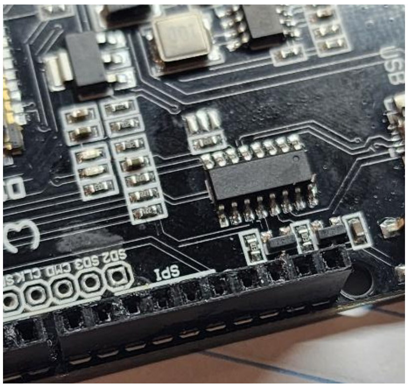


Figure 57: *Wemos D1 R32 micro-controller had* ***troubles*** *with the pin connections (Had to be replaced).*

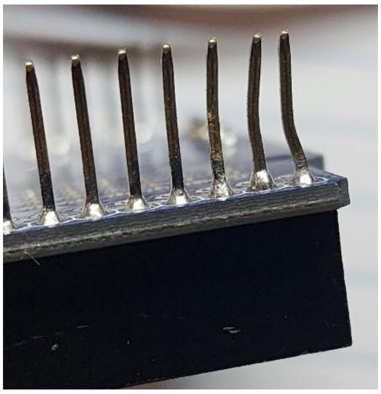


Figure 58: *Wemos D1 R32 micro-controller replaced female header pins.*

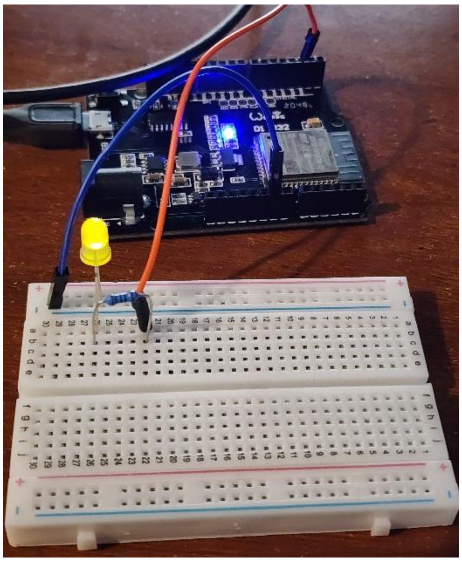


Figure 59: *Breadboard of led working with IoT (Internet of Things, Online integration)*

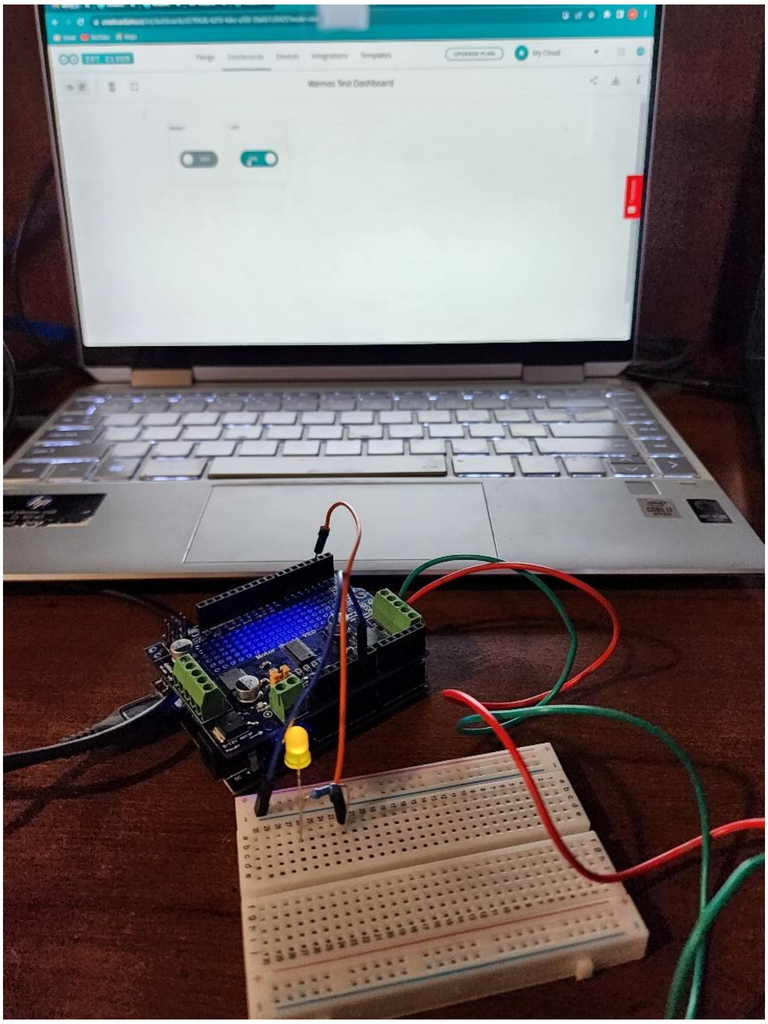


Figure 60: *Breadboard of led working with IoT, stacked Adafruit motor shield V2 (Internet of Things, Online integration)*

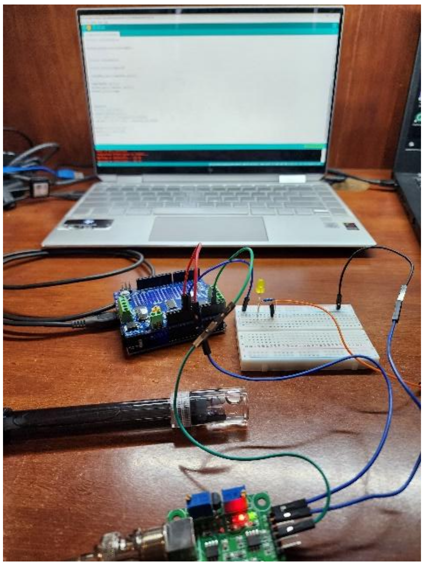


Figure 61: *Breadboard of pH sensor working Arduino IDE*

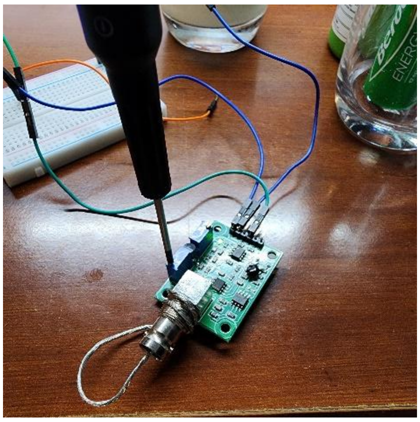


Figure 62: *Calibrating pH sensor by the adjustable Potentiometer.*

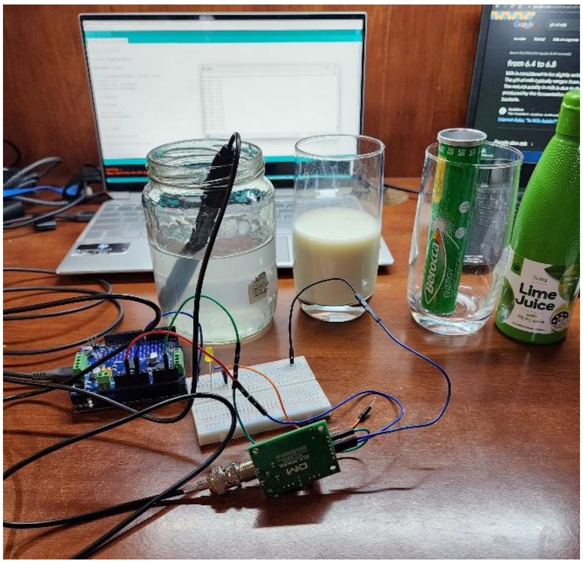


Figure 63: *Testing of the pH of water, milk, and lime juice.*

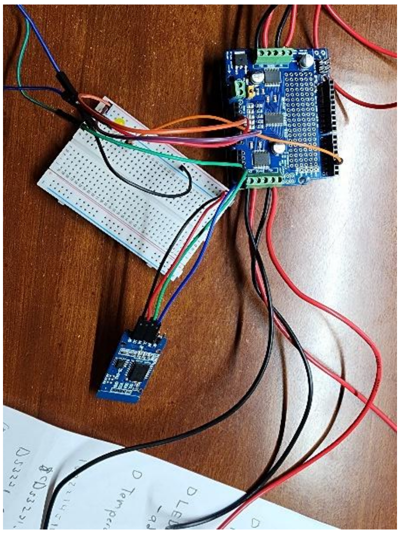


Figure 64: *Real time Clock (RTC) integration (Turns out the IoT has build in RTC, component not needed)*

Figure 65: *Soldering Setup Followed safety procedures with DIY extraction fan.*

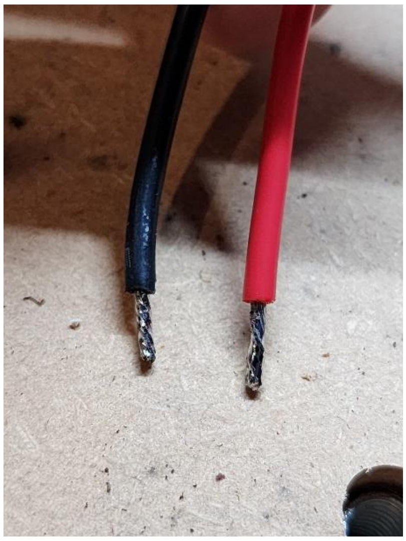


Figure 66: *Soldering wires.*

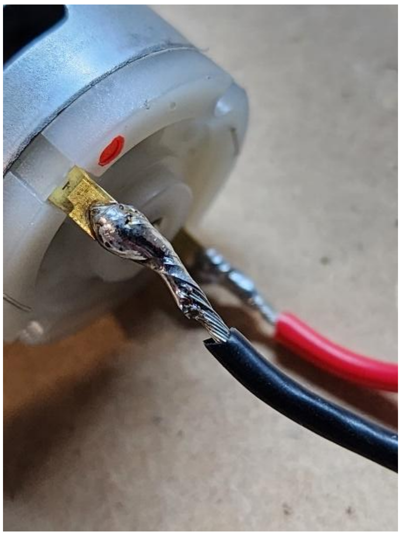


Figure 67: *Soldering wires to the dosage motors (Peristaltic DC motors)*

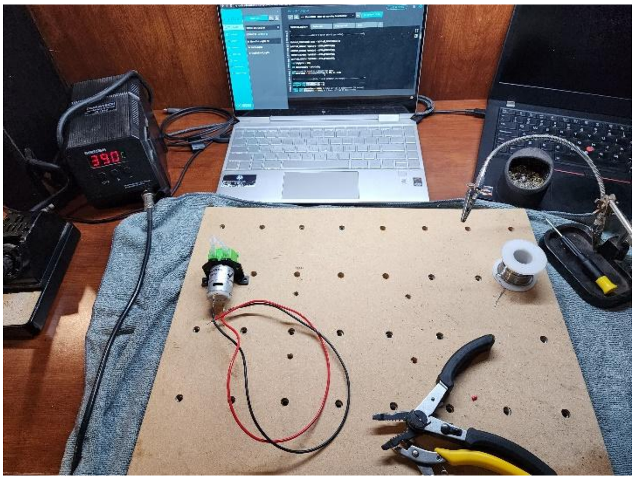


Figure 68: *Further testing and soldering*

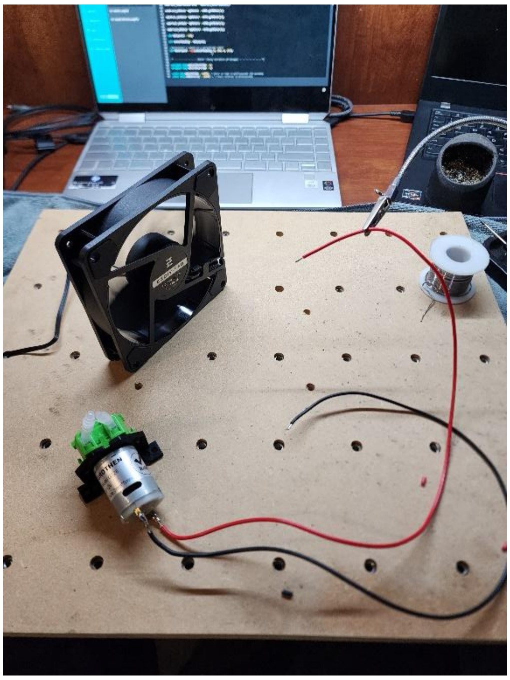


Figure 69: *Integrating with IoT soldering.*

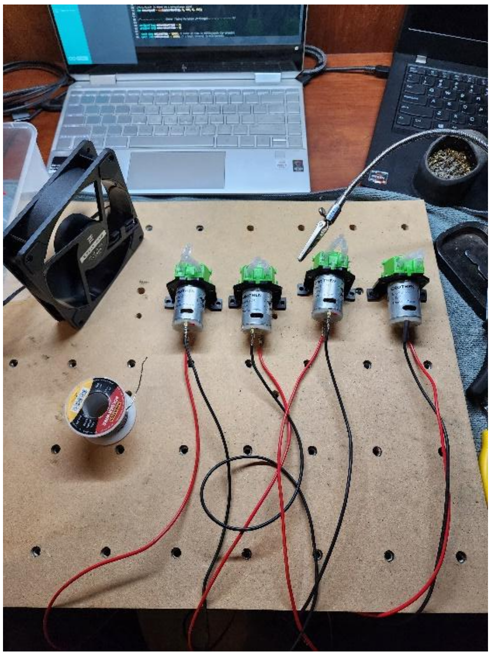


Figure 70: *Integrating the motors with IoT (Internet of things)*

Prototyping and IoT integration in the project required an innovative approach of blending hardware and software to create a functional system to drive multiple components to successfully grow plants simi-autonomously.

Product Design

Product Record

Figure 71: *Measuring lid for hydroponics PVC pipe to go in. (Photo extracted from a timelapse video I took of the construction)*

Figure 72: *The measured lid for hydroponics water bucket.*

__Figure 73:__ DIY Dremel cutting for a perfect round circle.

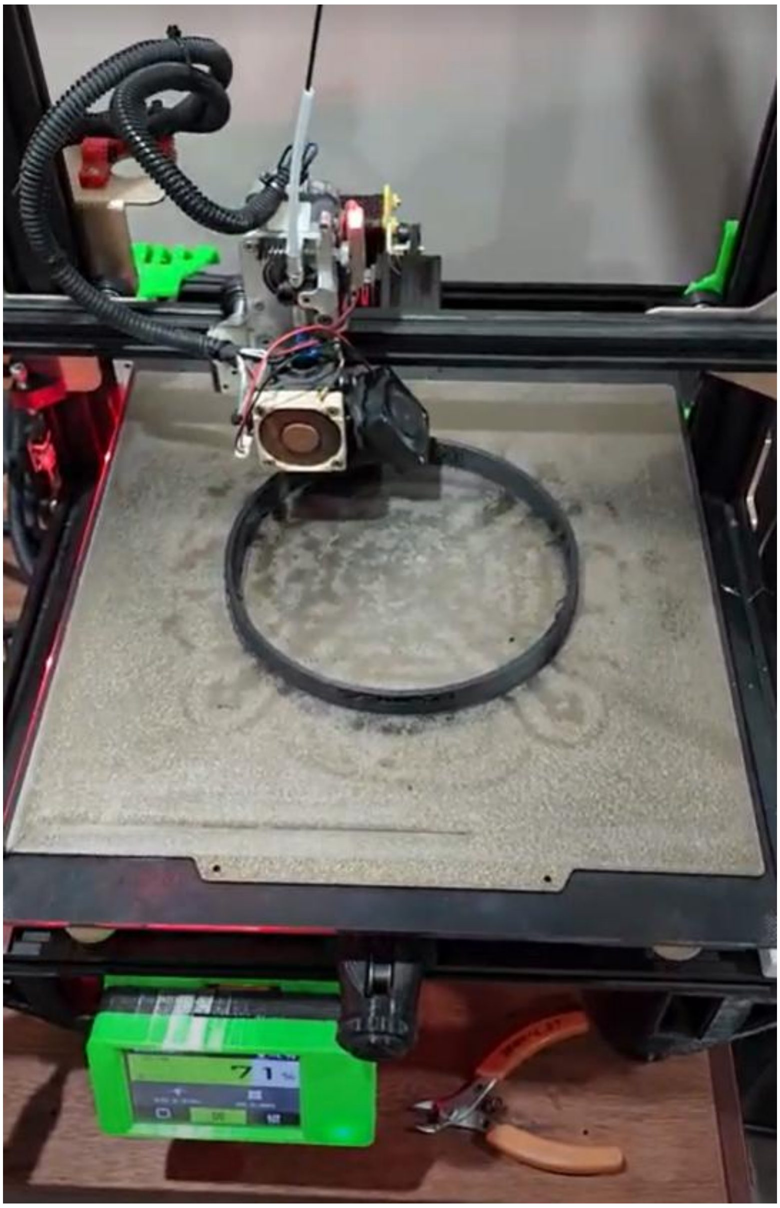


__Figure 74:__ Homemade 3D printer setup that will be used for printing the parts required to make the hydroponics.

__Figure 75:__ PVC pipe fits perfectly in the bucket.

__Figure 76:__ PVC pipe measuring where the hole is with masking tape, and cutting slits with Miter saw.

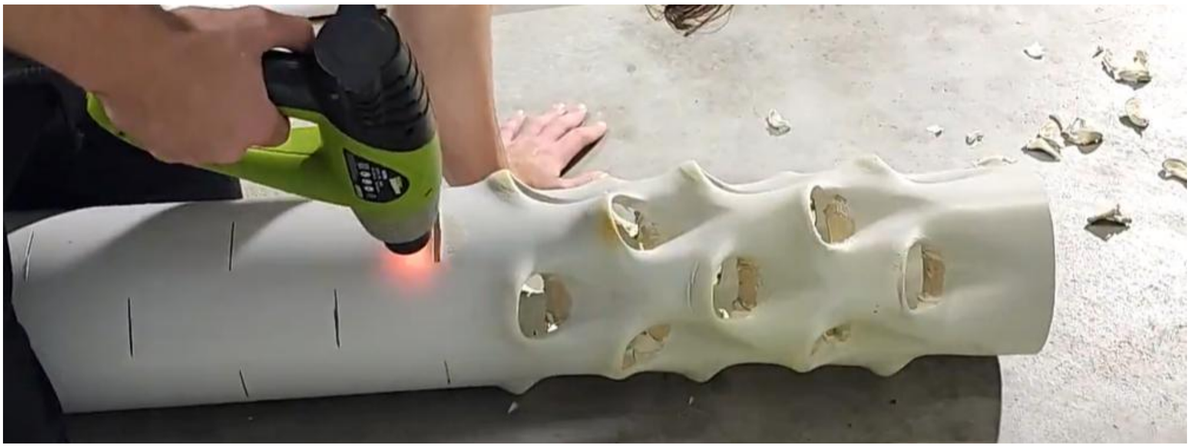


Figure 77: *PVC pipe shaping the holes with a heat gun by the measured and cut areas.*

Figure 78: *PVC pipe shaping the holes with another PVC pipe that is the size of the hydroponic net cups.*

Figure 79: *The hydroponic net cups.*

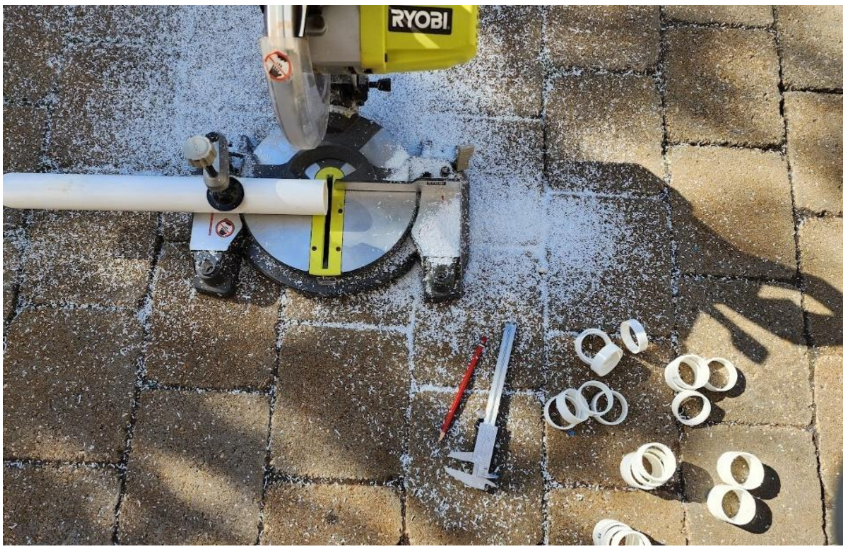


Figure 80: *Cutting rings for the hydroponic net cups to be secured with (cut with the Miter saw).*

Figure 81: *Finished heat shaping the holes of the PVC pipe.*

Figure 82: *Gluing the cut-out rings to the heat shaped holes.* (Glued with *Blue Cement Solvent)*

Figure 83: *The ring for the water basin at the top of the hydroponics tower.*

Figure 84: *Glued (Blue Cement Solvent) The ring for the water basin at the top of the hydroponics tower.*

Figure 85: *Preparing the tube for the main water pump.*

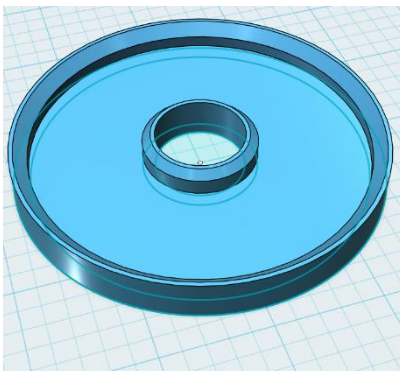


Figure 86: *CAD design of the water basin attachment at the top of tower.*

Figure 87: *Printed the water basin attachment at the top of tower.*

Figure 88: *Glued (Blue Cement Solvent) the tube for the main water pump extending its length.*

Figure 89: *Glued (Blue Cement Solvent) the water basin attachment and added the tube for the main water pump.*

Figure 90: *Preparation of the tube with the main water pump.*

Figure 91: *Testing the water flow of the main water pump (Centrifugal pump).*

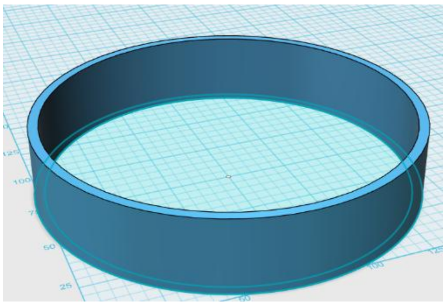


Figure 92: *CAD design for the water basin wall extension.*

Figure 93: *Gluing the 3D printed water basin wall extension with a weight to hold in place.*

Figure 94: *Testing the water flow with the water basin wall extension.*

Figure 95: *Re-using a piece of composite wood for enclosure.*

Figure 96: *Preparing the composite wood.*

Figure 97: *measuring the composite wood.*

Figure 98: *The measured lines for the rack of the enclosure design.*

Figure 99: *Setting the appropriate depth for the handheld Circular Saw.*

Figure 100: *Finished cut for the shelf of the electronics enclosure.*

Figure 101: *Re-purposing old garage shelving units, shelf brackets.*

Figure 102: *Shelf brackets, setup for enclosure, (holds the nutrients and electronics).*

Figure 103: *Test fitting the wood shelving unit.*

Figure 104: *Cut more pieces for enclosure, preparing to paint.*

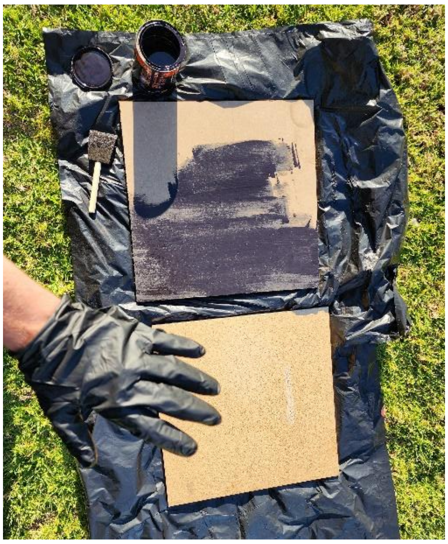


Figure 105: *Painting the wood shelf units.*

Figure 106: *Final painted wood shelf units.*

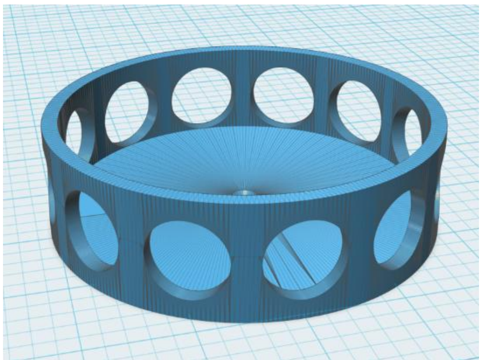


Figure 107: *CAD design for the nutrient holding brackets.*

Figure 108: *3D printed and preparing the holding brackets for the nutrients, sanded the supports.*

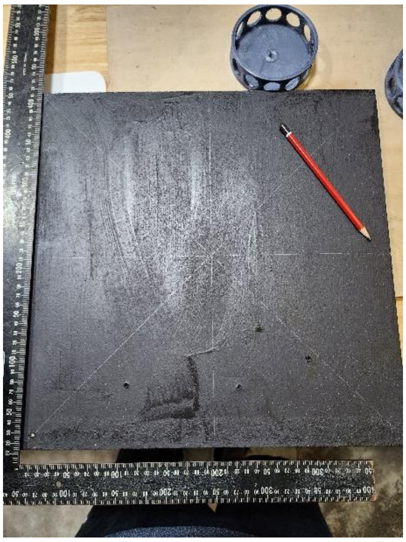


Figure 109: *Measuring where the holding brackets will be screwed.*

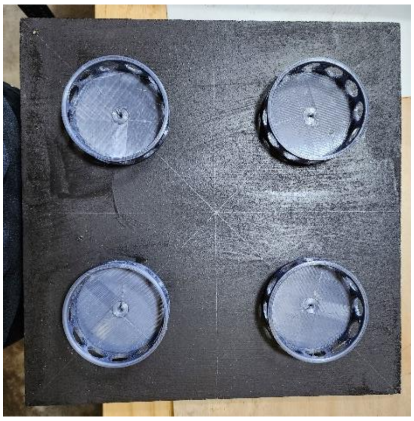


Figure 110: *Screwed on the holding brackets for the nutrient glass bottles.*

Figure 111: *Test fitting the glass bottles that with hold the nutrients and pH up/down.*

Figure 112: *Drilled holes in the water basin.*

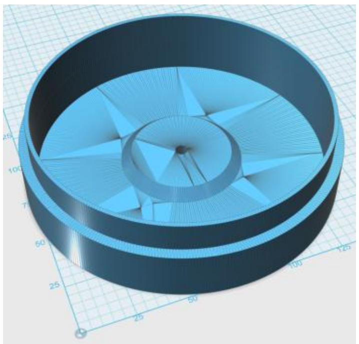


Figure 113: *CAD design for top lid of water basin, also for LED lights to be glued on.*

Figure 114: *Printed the top lid of water basin. (Hols the LED Grow lights on the top).*

Figure 115: *Test fitting the top lid of water basin.*

Figure 116: *3mm screw (too long)*

Figure 117: *Cutting the 3mm screw.*

Figure 118: *Smoothing the 3mm screw.*

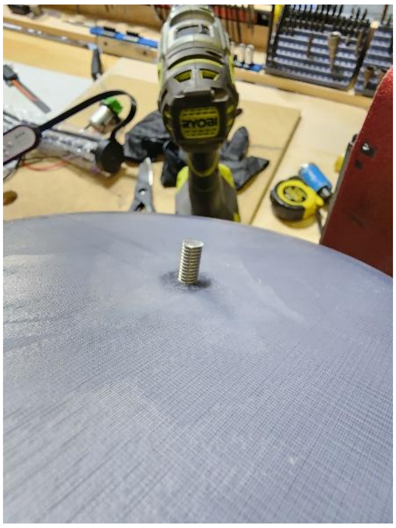


Figure 119: *Close look at the 3mm screw.*

Figure 120: *Bottom look at the 3mm screw.*

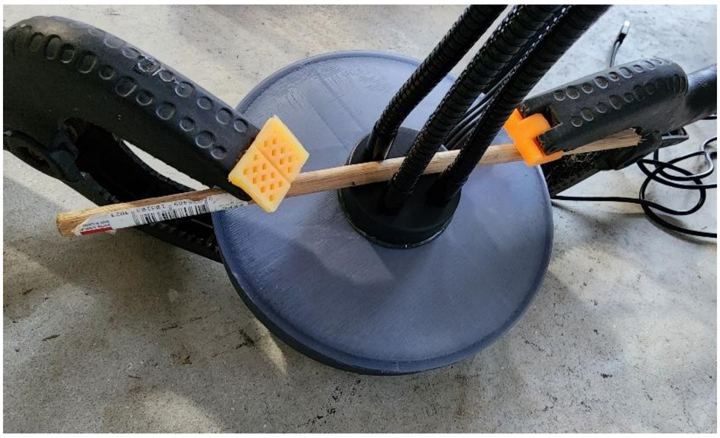


Figure 121: *Gluing the LED grow lights to the top lid.*

Figure 122: *Finished glued on LED grow lights to the top lid.*

Figure 123: *Finished glued on LED grow lights to the top lid added to the hydroponics tower.*

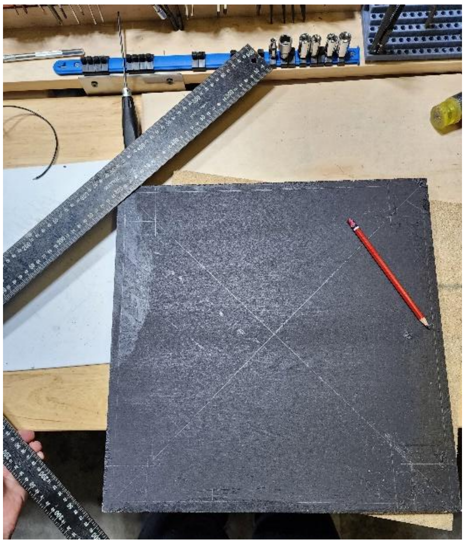


Figure 124: *Measuring hole placement for dosage motor pumps.*

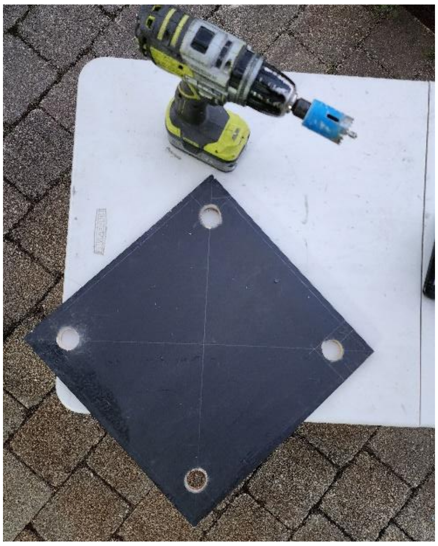


Figure 125: *Drilling holes for dosage motor pumps.*

Figure 126: *Securing the metal shelving unit with screws.*

Figure 127: *Gluing with techGrip glue.*

Figure 128: *Weights to hold shelves in place to dry.*

Figure 129: *Adding wheels to be more portable.*

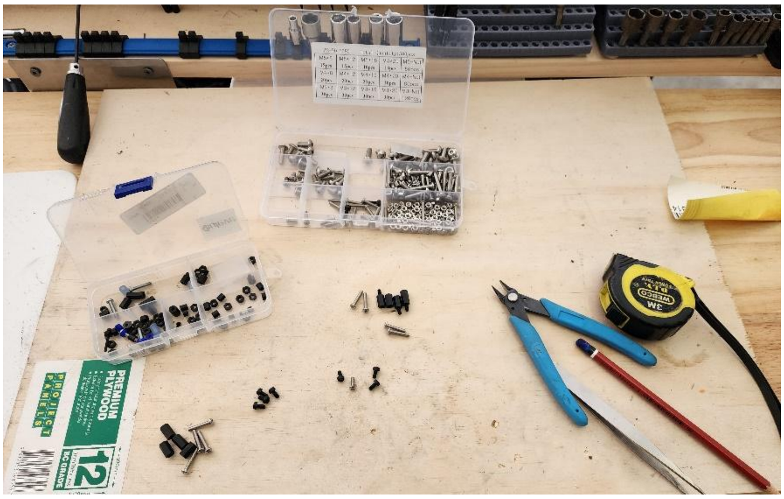


Figure 130: *Organising pins for wire management.*

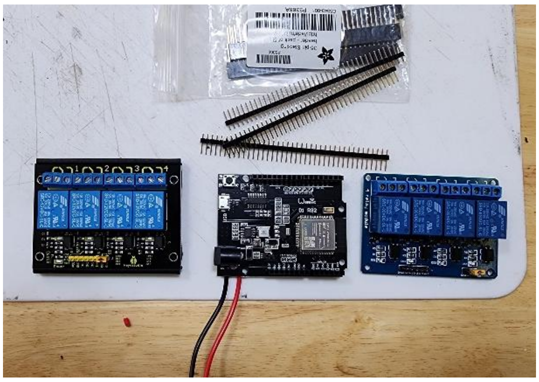


Figure 131: *Switching to 4 channel relays (Adafruit motor shield was not compatible, Check Challenges).*

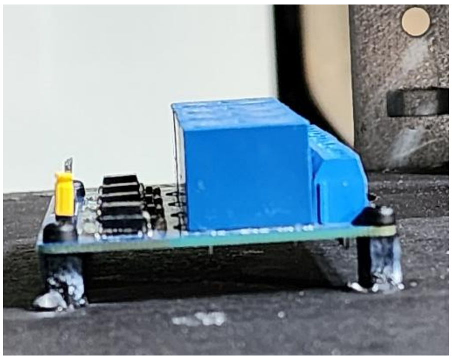


Figure 132: *4 channel relays added to wood enclosure with the pins.*

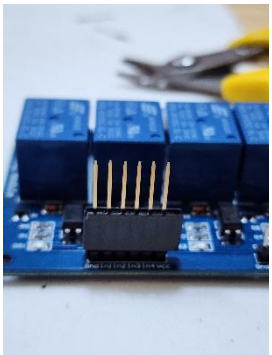


Figure 133: *Other 4 channel relays with header pins attached.*

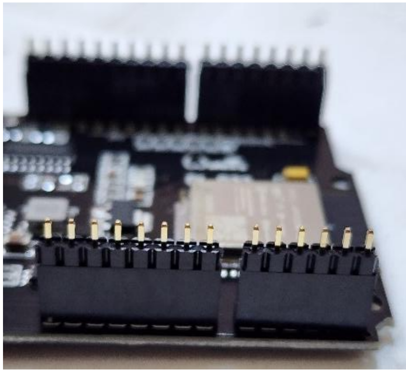


Figure 134: *Header pins attached to Wemos D1 R32.*

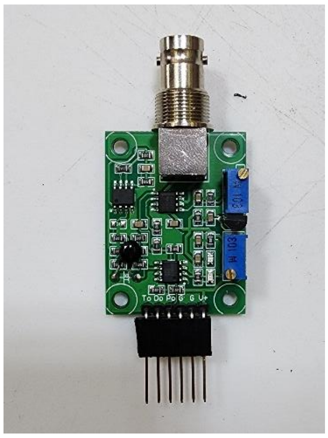


Figure 135: *Header pins attached to pH meter/sensor PH-4502G*

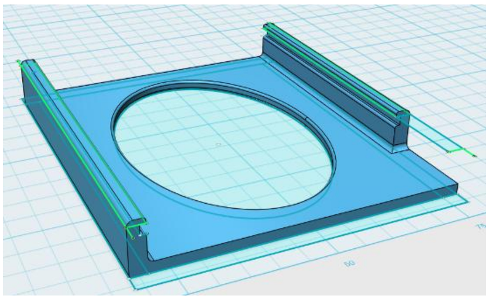


Figure 136: *CAD design of 4 channel relay holders.*

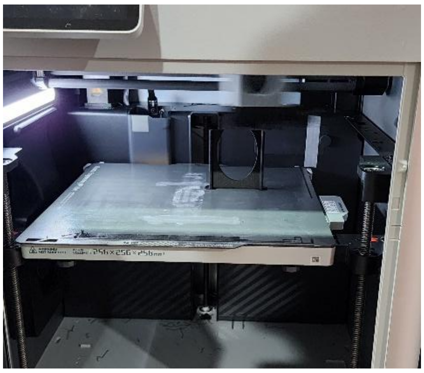


Figure 137: *3D printing the 4 channel relay holders.*

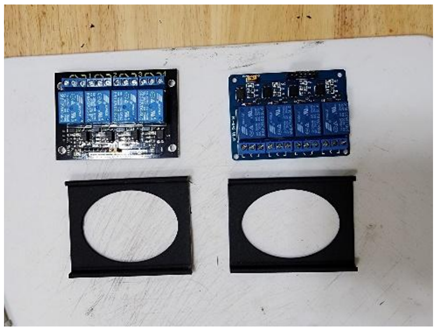


Figure 138: *Finished 3D printing the 4 channel relay holders.*

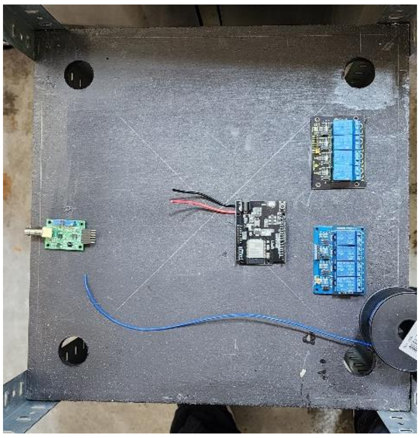


Figure 139: *Adding the components to enclosure.*

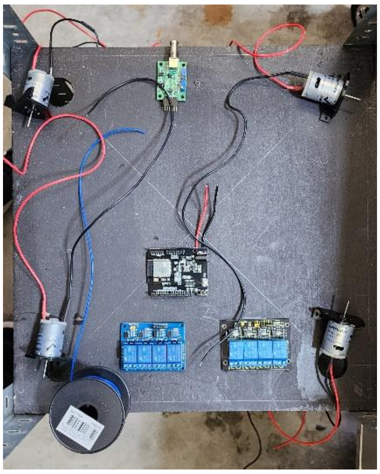


Figure 140: *Laying out the components to enclosure.*

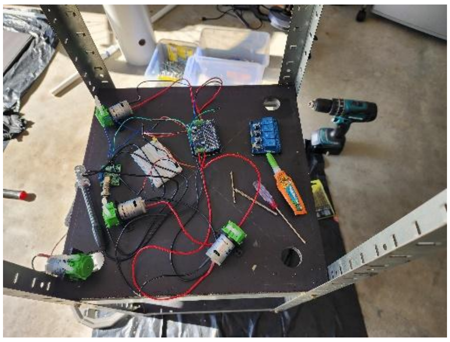


Figure 141: *Gluing and arranging the components, holders to enclosure.*

Figure 142: *Soldering of the wires.*

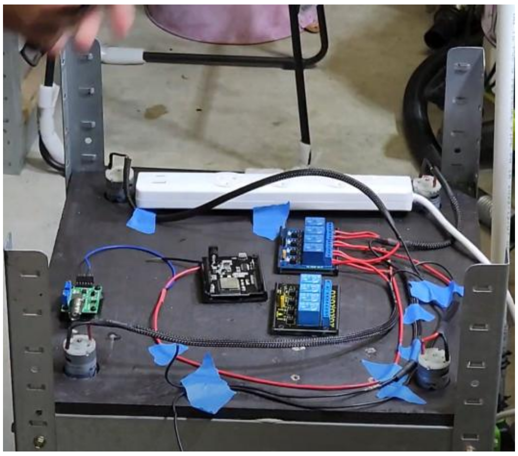


Figure 143: *Wire management.*

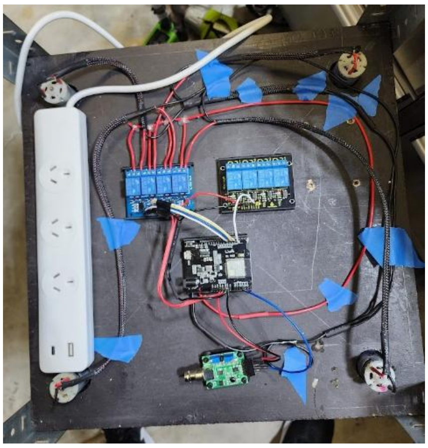


Figure 144: *Top view Wire management.*

Figure 145: *Side walls of enclosure painted left to dry.*

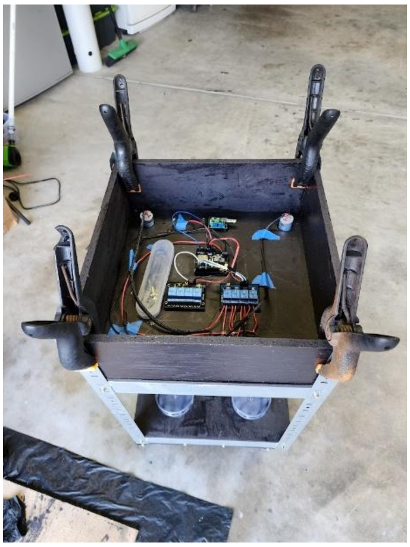


Figure 146: *Gluing of the side walls for enclosure.*

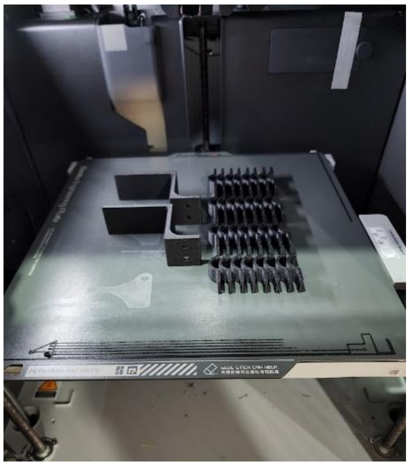


Figure 147: *3D printed bracket for 12-volt power supply and wire management clamps.*

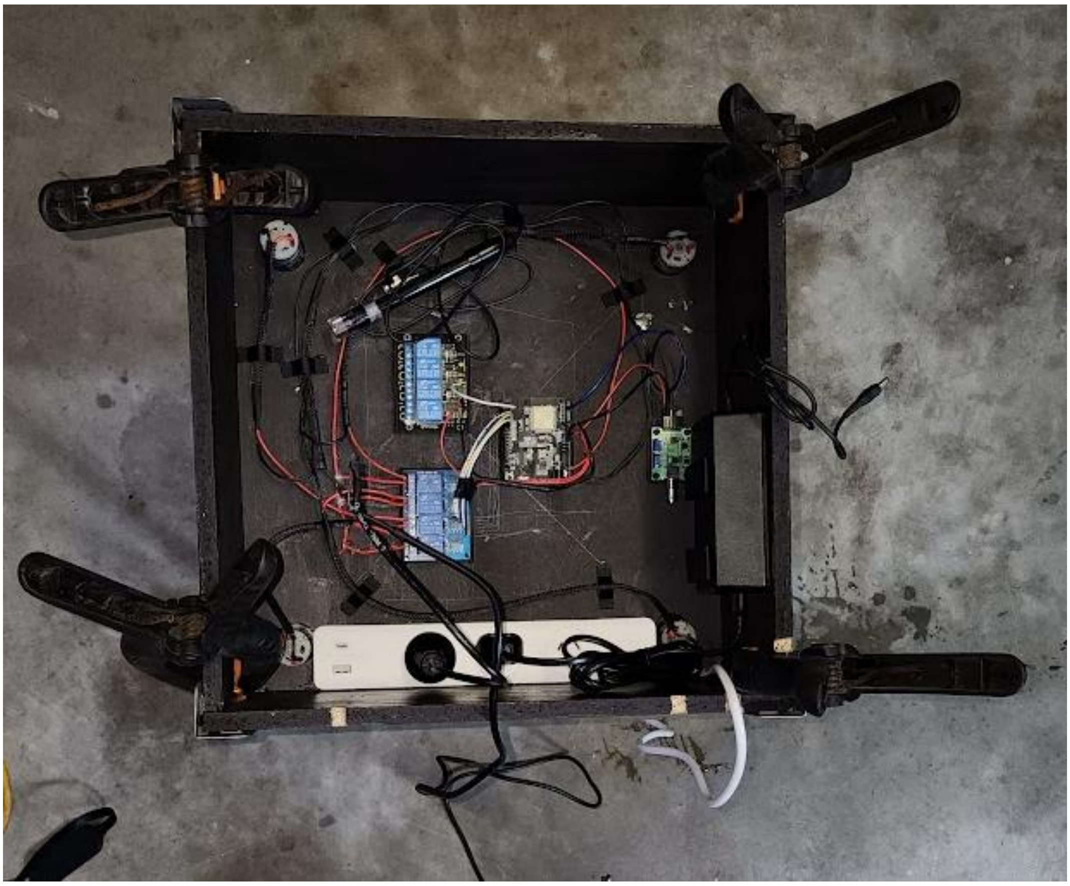


Figure 148: *Top view of electronic wire management.*

Figure 149: *Testing the IoT connection with the hydroponics system.*

Figure 150: *After some fixing the IoT integration works seen by the water coming out from the top water basin.*

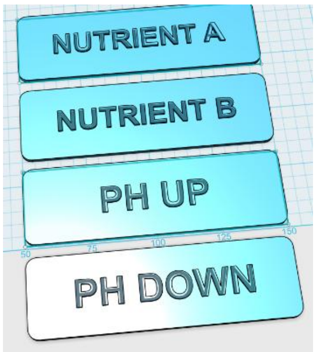


Figure 151: *CAD design of the labels.*

Figure 152: *3D printed labels, (Printed on the Bambu Lab X1-carbon)*

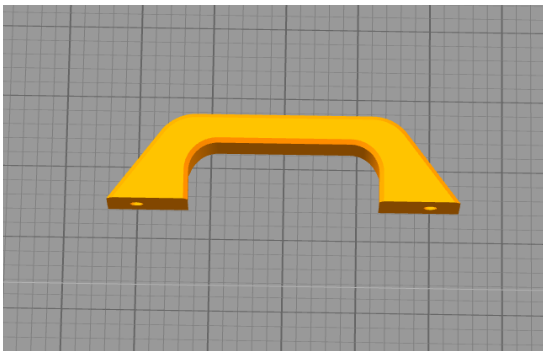


Figure 153: *CAD design of handle.*

Figure 154: *3D printed handle for lid of the enclosure.*

Figure 155: *Testing of the portability of the hydroponics system.*

Challenges Overcome

IoT Integration Problems

- **Connectivity Issues:** Struggled to maintain a stable internet connection. Issues such as poor Wi-Fi signal disruptions or updates.
- **Compatibility:** Ensuring the IoT devices (Wemos D1 R32) and sensors are compatible with the platform.
- **Scalability:** Adding more IoT devices or variables to the network, managing and scaling the system can become complex, logical issues resolved by troubleshooting.
- **Power Management:** Some IoT devices may run on batteries, requiring efficient power management, gave direct power from power outlet from the house.
- **Latency:** Depending on the application, latency in data transmission can be problematic. Use of schedules to dose nutrients based on a discrepancy between the latency.
- **Device Management:** Keeping track of numerous devices, updating their firmware, and managing configurations.
- **Interoperability:** Ensuring that devices from different manufacturers work together, the Wemos and Adafruit did not work together had to switch to relays.
- **Complexity:** IoT integration can be complex and may require expertise in hardware, software, and data analytics, which was learnt to control the hydroponics system.
- **Firmware and Software Updates:** Kept the IoT device firmware and software up to date.
- **User Training:** Needed to go though lots of forums to understand how to code and troubleshoot the software.

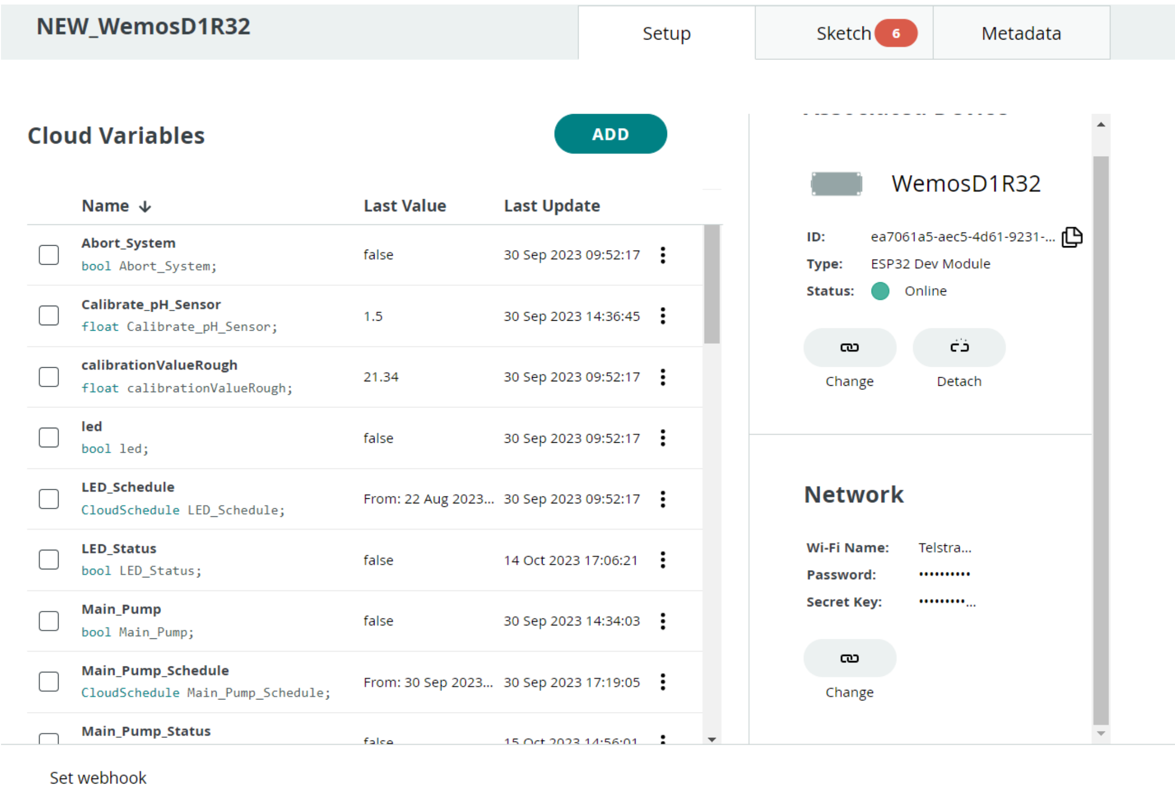


Figure 156: The Arduino IoT integration.

Adafruit Motor shield compatibility

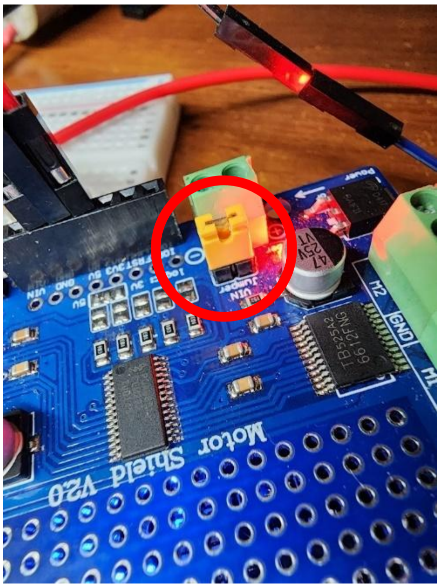


Figure 157: Yellow power supply selection Jumper.

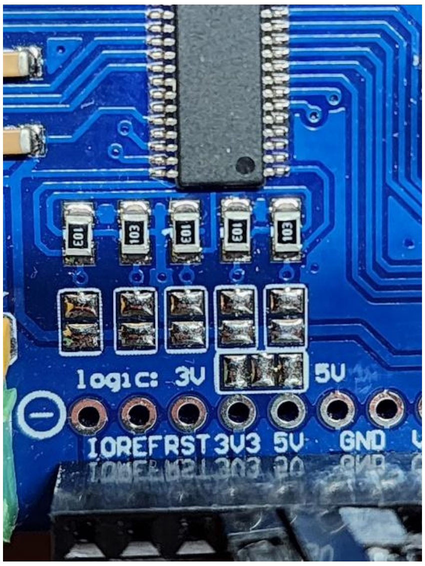


Figure 158: Changing of the 5v logic to 3v logic.

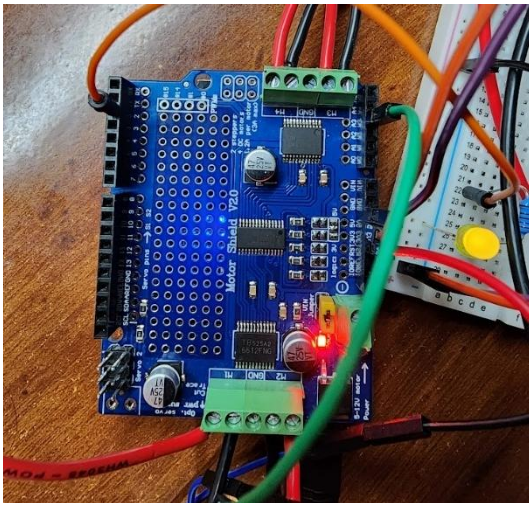


Figure 159: Testing the motor shield with the changes. (Not working)

I initially encountered issues while troubleshooting the Adafruit Motor Shield, beginning with an extensive evaluation of the system's coding logic. After rigorous testing, the root of the problem became evident: the disparity in logic voltage levels between the Adafruit Motor Shield (which operated at 5 volts) and my Wemos D1 R32 microcontroller (which utilised 3 volts). To address this, I adjusted the logic voltage on the motor shield to match the microcontroller's requirements. However, despite this seemingly compatible setup, the re-testing phase revealed that the problem persisted. Faced with the challenge of making the system functional, I eventually decided to switch to relays as an alternative solution, which proved effective in ensuring smooth motor operations and overcoming the logic compatibility issue. This shift allowed me to move forward and achieve the desired results in my motor control system.

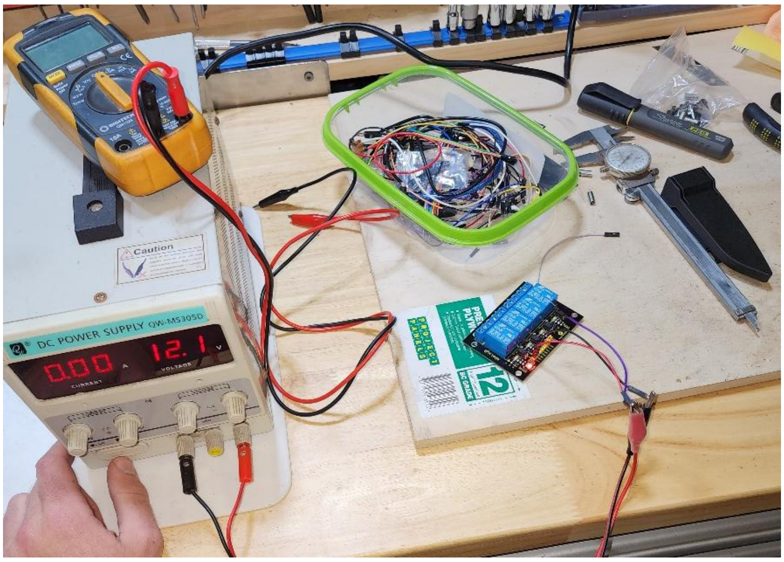


Figure 160: Testing the 12v 4 channel relay module. (Switched to relays to control the dosage motors)

To address hardware limitations, I switched to relay modules, which brought its own set of challenges. I had a 4 channel relay module crucial for my project but only had a 12-volt and a 5-volt relay. To use the 12v relay effectively, I had to establish a direct connection to the 12v power supply, essential for powering both the relay and the Wemos D1 R32 microcontroller. This solution resolved compatibility issues and enabled me to continue with my project.

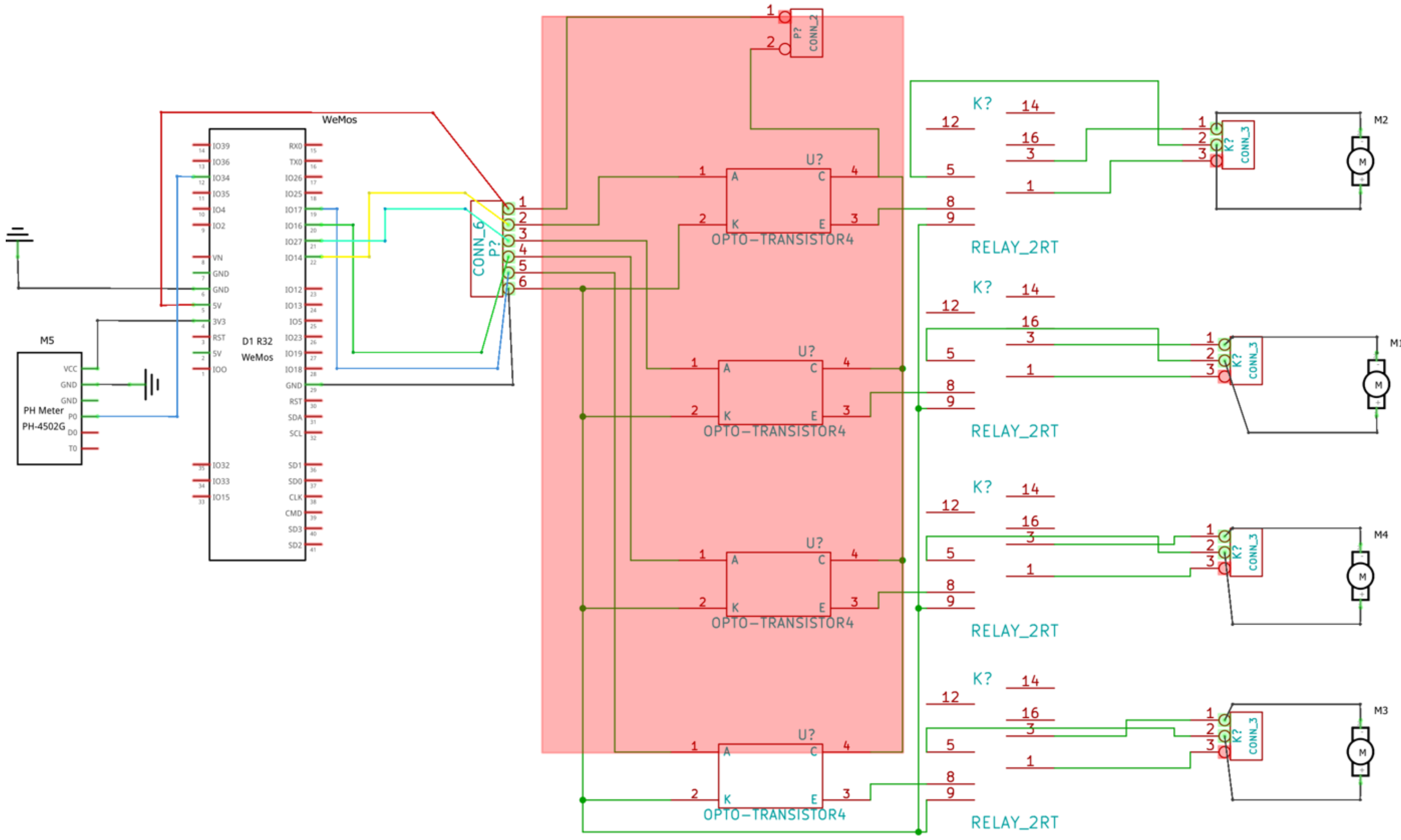


Figure 161: Circuit diagram due to the changes, switching to 4 channel relay modules to control the motors (M1, M2, M3, and M4).

Figure 162: With switching to relays, flow restrictors had to be printed to adjust dosage flow.

Figure 163: 4 3D printed flow restrictors to adjust dosage flow.

Switching to relays meant that the flow rate cannot be adjusted through code as the Adafruit motor shield can adjust the speed of the motors. Switching to the 3D printed flow restrictors allows for precise adjustment of the dosage flow. Also, one way valves where placed on the ends to ensure that there are no air pockets in the pipes.

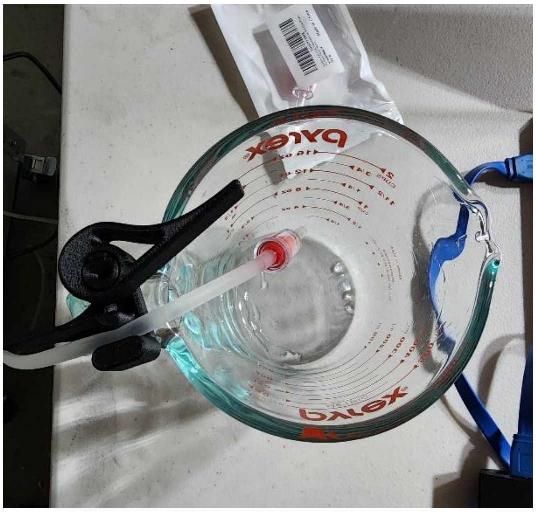


Figure 164: 4 3D printed flow restrictors to adjust dosage flow.

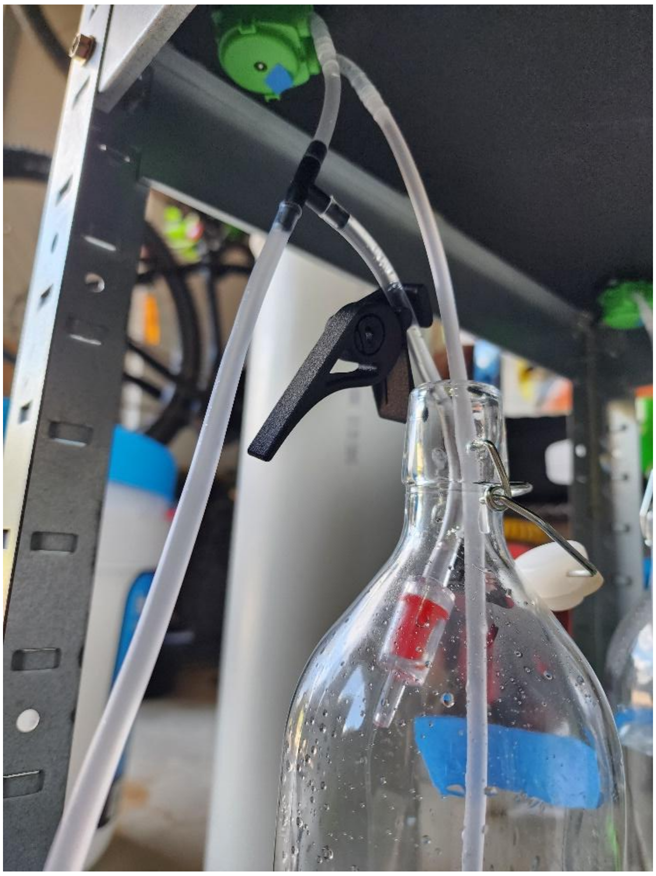


Figure 165: Positive displacement pump with flow rate restrictor for manual adjustment of flow.

Through testing the flow rate, it was determined that the flow can be adjusted from 0.1mL/s with not clamped to 1.1mL/s fully clamped. The peristaltic dosage motors are positive displacement pumps, so to limit flow, the clamp is open letting the nutrients to flow back into the glass as seen in *Figure 155*.

Figure 166: Holes in bucket for dosage pipes.

Figure 167: Dosage pipes through holes.

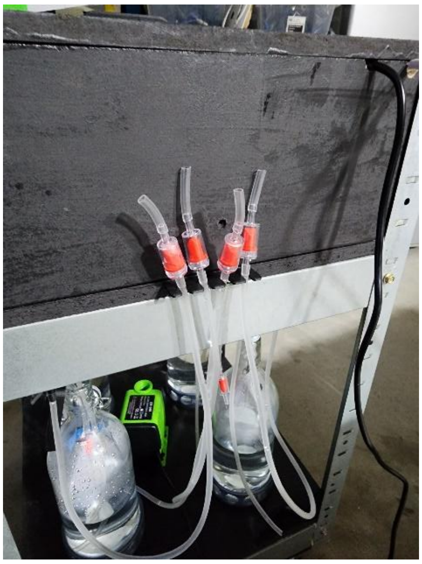


Figure 168: Dosage pipes bracket holder for the one way valves.

Sensor Problems

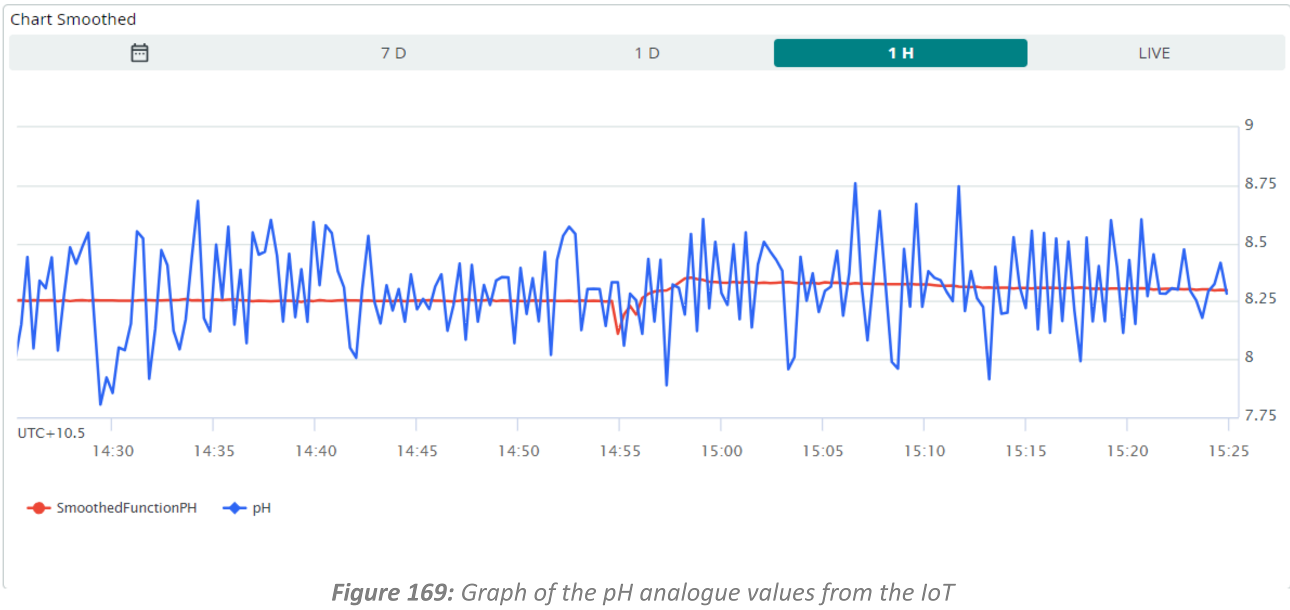


Figure 169: *Graph of the pH analogue values from the IoT*

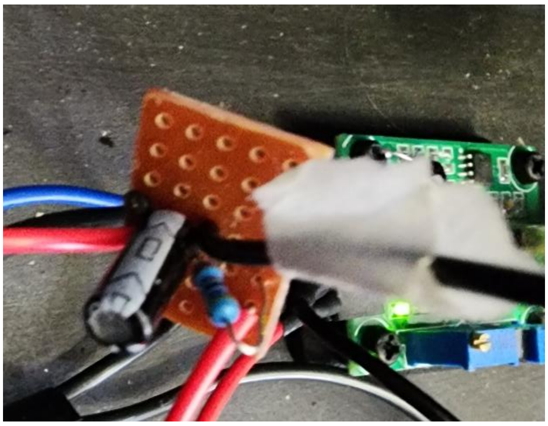


Figure 170: *Low pass filter (Resistor-capacitor type filter)*

A RC (Resistor-Capacitor) filter was used at first to smooth the blue line in the graph, where the resistor and capacitor are used together to achieve the desired smoothing effect on the analogue signal (pH). This filter is often used to remove high-frequency noise or ripples from a signal, leaving a smoother, more stable output shown by the red line.

A more effective solution was to add a library of code to the IoT called Smoothed.h which smooths the analogue value straight in the IoT.

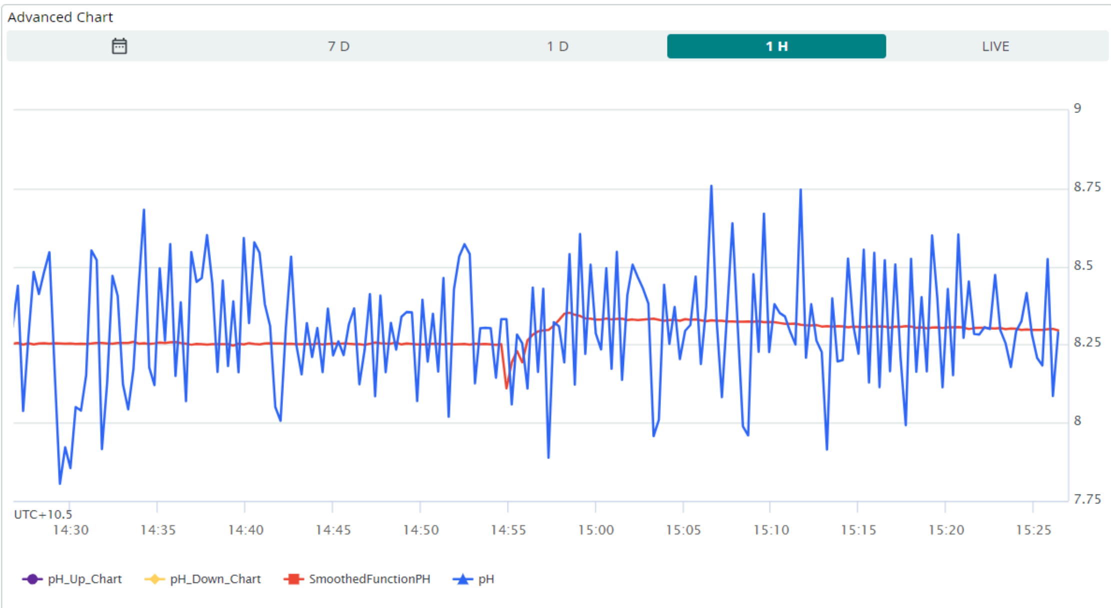


Figure 171: *Better graph of the pH analogue values from the IoT, also maps the dosages of pH down/up*

Arduino IoT Cloud

Software

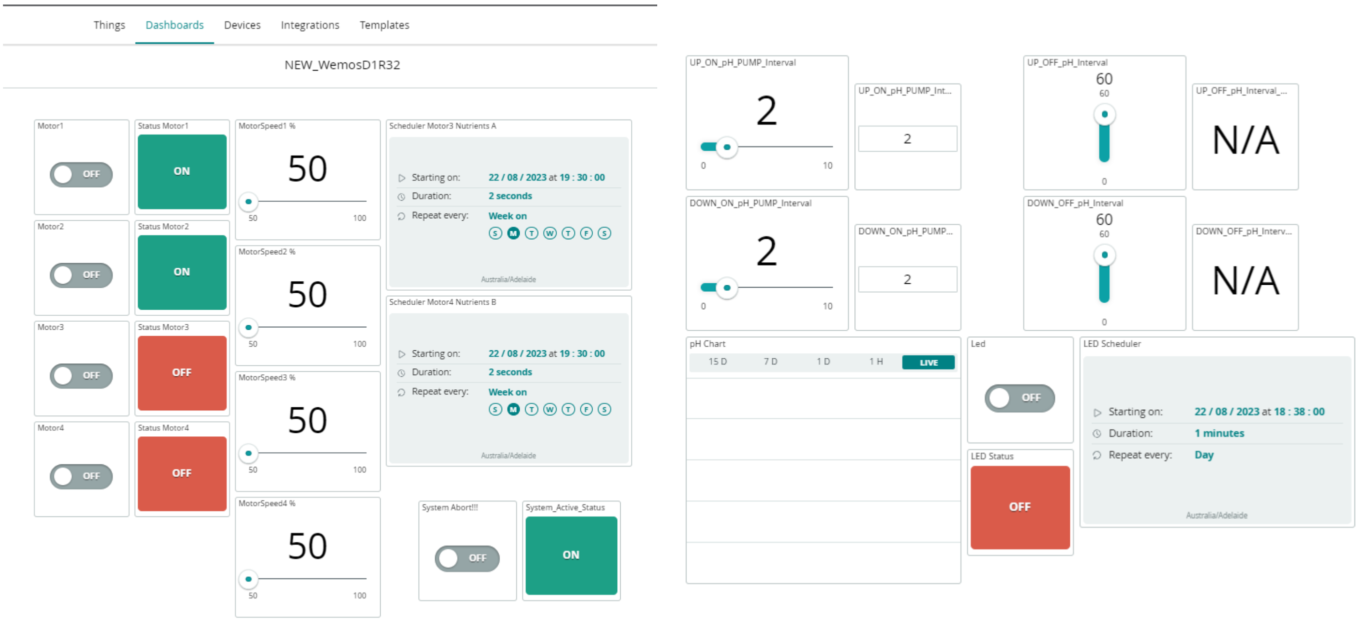
```
/*   Sketch generated by the Arduino IoT Cloud Thing "Untitled"
  https://create.arduino.cc/cloud/things/c549c9db-085d-42e1-91db-cc801dc094de

  Arduino IoT Cloud Variables description

  The following variables are automatically generated and updated when changes are made to the Thing

  float Calibrate_pH_Sensor;
  float calibrationValueRough;
  float pH;
  float pH_Down_Value;
  float pH_Up_Value;
  float SmoothedFunctionPH;
  int Nutrients_A_Chart;
  int Nutrients_B_Chart;
  int pH_Down_Chart;
  int pH_Up_Chart;
  int smoothedAverageCalibrate;
  CloudSchedule LED_Schedule;
  CloudSchedule Main_Pump_Schedule;
  CloudSchedule pH_Down_Schedule_Check;
  CloudSchedule pH_Up_Schedule_Check;
  CloudSchedule ScheduleMotor3;
  CloudSchedule ScheduleMotor4;
  bool Abort_System;
  bool led;
  bool LED_Status;
  bool Main_Pump;
  bool Main_Pump_Status;
  bool Motor1;
  bool Motor1_Status;
  bool Motor2;
  bool Motor2_Status;
  bool Motor3;
  bool Motor3_Status;
  bool Motor4;
  bool Motor4_Status;
  bool OverRide_Motor1;
  bool OverRide_Motor2;
  bool OverRide_Motor3;
  bool OverRide_Motor4;
  bool System_Active_Status;

  Variables which are marked as READ/WRITE in the Cloud Thing will also have functions
  which are called when their values are changed from the Dashboard.
  These functions are generated with the Thing and added at the end of this sketch. */
#include <Smoothed.h> // Smoothed - Version: Latest
#include "thingProperties.h"
#include <Wire.h>
/*--------------------(Motor & Motor Shield Variables)--------------------*/
Adafruit_MotorShield AFMS = Adafruit_MotorShield();
Adafruit_DCMotor *myMotor1 = AFMS.getMotor(1);
Adafruit_DCMotor *myMotor2 = AFMS.getMotor(2);
Adafruit_DCMotor *myMotor3 = AFMS.getMotor(3);
Adafruit_DCMotor *myMotor4 = AFMS.getMotor(4);
/*--------------------(millis)--------------------*/
int time_variable;
unsigned long previousMillis = 0;
/*--------------------(pH sensor Variables)--------------------*/
float calibration_value = 21.34 + 0.95;
int phval = 0;
unsigned long int avgval;
int buffer_arr[10], temp;
float ph_act;
float pH_Down_Value_Change = 6.5;
float pH_Up_Value_Change = 5.5;
float CalibrationValue = 0.95;
float CalibrationValueRoughSAVE = 21.34;
/*--------------------(LED)--------------------*/
int MYled = 26;
/*--------------------(Motor Variables)--------------------*/
int M1 = 14;
int M2 = 27;
int M3 = 16;
int M4 = 17;
int MainPump = 25;
/*--------------------(Smoothed)--------------------*/
Smoothed <float> mySensor;
```

```
/*====================[void setup]====================*/
void setup() {
  // 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);
/*--------------------(IoT Cloud)--------------------*/
// Defined in thingProperties.h
  initProperties();
  // Connect to Arduino IoT Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);
  /*
     The following function allows you to obtain more information
     related to the state of network and IoT Cloud connection and errors
     the higher number the more granular information you'll get.
     The default is 0 (only errors).
     Maximum is 4
 */
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();
/*--------------------(Smoothed)--------------------*/
mySensor.begin(SMOOTHED_AVERAGE, 1000);
/*--------------------(Dosage Motor Setup)--------------------*/
  AFMS.begin();
  myMotor1 ->setSpeed(0);
  myMotor2 ->setSpeed(0);
  myMotor3 ->setSpeed(0);
  myMotor4 ->setSpeed(0);
/*--------------------(LED)--------------------*/
  pinMode(MYled, OUTPUT);
/*--------------------(Motor )--------------------*/
pinMode(M1, OUTPUT);
pinMode(M2, OUTPUT);
pinMode(M3, OUTPUT);
pinMode(M4, OUTPUT);

digitalWrite(M1, HIGH);
digitalWrite(M2, HIGH);
digitalWrite(M3, HIGH);
digitalWrite(M4, HIGH);
/*--------------------(MainPump )--------------------*/
pinMode(MainPump, OUTPUT);
digitalWrite(MainPump, LOW);
}
/*====================[void loop]====================*/
void loop() {
  ArduinoCloud.update();
/*--------------------(Millis Function/Timer Function RTC)--------------------*/
  unsigned long currentMillis = millis();
/*--------------------(pH sensor)--------------------*/
//CalibrationValue is changed in the IoT /THE Capacitance was 21.34 now that the RC filter was added it was changed
calibration_value = CalibrationValueRoughSAVE + CalibrationValue;
  for (int i = 0; i < 10; i++)  {
    buffer_arr[i] = analogRead(34);
  }
  for (int i = 0; i < 9; i++)  {
    for (int j = i + 1; j < 10; j++)    {
      if (buffer_arr[i] > buffer_arr[j])      {
        temp = buffer_arr[i];
        buffer_arr[i] = buffer_arr[j];
        buffer_arr[j] = temp;
      }
    }
  }
  avgval = 0;
  for (int i = 2; i < 8; i++)
    avgval += buffer_arr[i];
  float volt = (float)avgval * 3.3 / 4095 / 6;
  ph_act = -5.70 * volt + calibration_value;
  pH = ph_act;
/*--------------------(Smoothed)--------------------*/
mySensor.add(pH);
float smoothedSensorValueAvg = mySensor.get();
SmoothedFunctionPH = smoothedSensorValueAvg; //SmoothedFunctionPH is the cloud variable, smoothedSensorValueAvg is
the smoothed average value
/*--------------------(Abort)--------------------*/  //if abort system not switched on
if(!Abort_System) {
System_Active_Status = false;
/*--------------------(pH Down _ myMotor1)--------------------*/
if (Motor1 || pH_Down_Schedule_Check.isActive()) {
 // if (Motor1) {
   if(SmoothedFunctionPH>pH_Down_Value_Change || OverRide_Motor1) {
//   Serial.println("Motor 1 ON.");
digitalWrite(M1, LOW);
    Motor1_Status = true;
     pH_Down_Chart = 6;
}
} else {
```

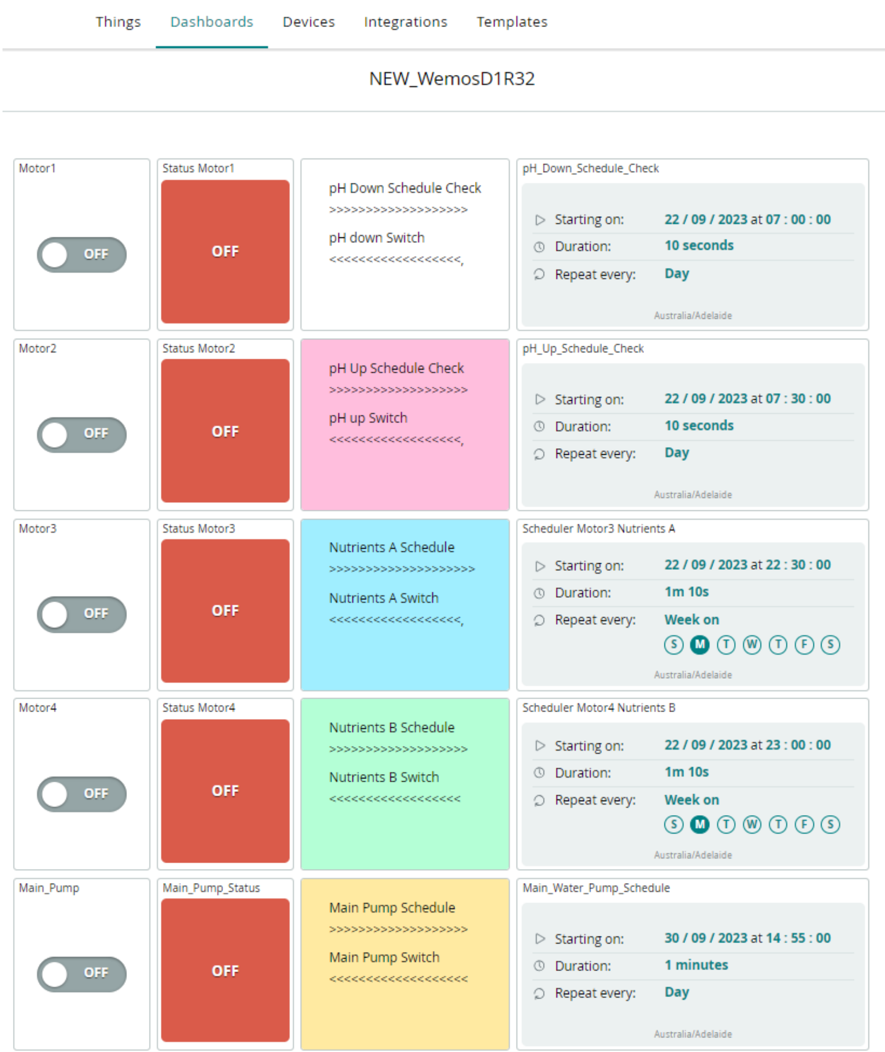
```
//    Serial.println("Motor 1 OFF.");

  digitalWrite(M1, HIGH);
    Motor1_Status = false;
pH_Down_Chart = 0;
}
/*--------------------(pH Up _ myMotor2)--------------------*/
if (Motor2 || pH_Up_Schedule_Check.isActive()) {
//  if (Motor2 ) {
   if(SmoothedFunctionPH<pH_Up_Value_Change || OverRide_Motor2) {
//    Serial.println("Motor 2 should be ON.");

  digitalWrite(M2, LOW);
    Motor2_Status = true;
      pH_Up_Chart = 5;
}
} else {
//    Serial.println("Motor 2 should be OFF.");

  digitalWrite(M2, HIGH);
    Motor2_Status = false;
pH_Up_Chart = 0;
}
/*--------------------(Schedule Nutrients A Motor 3)--------------------*/
if (Motor3 || ScheduleMotor3.isActive()) {
 //if (Motor3) {
//    Serial.println("Motor 3 should be active.");
  digitalWrite(M3, LOW);
    Motor3_Status = true;
     Nutrients_A_Chart = 1;
} else {
//   Serial.println("Motor 3 should be OFF.");
  digitalWrite(M3, HIGH);
    Motor3_Status = false;
Nutrients_A_Chart = 0;
}
/*--------------------(Schedule Nutrients B Motor 4)--------------------*/
if (Motor4 || ScheduleMotor4.isActive()) {
//    Serial.println("Motor 4 should be active.");
  digitalWrite(M4, LOW);
    Motor4_Status = true;
      Nutrients_B_Chart = 1;
} else {
//    Serial.println("Motor 4 should be OFF.");
  digitalWrite(M4, HIGH);
    Motor4_Status = false;
   Nutrients_B_Chart = 0;
}
/*--------------------(Schedule led)--------------------*/
  if (led) {
    digitalWrite(MYled, HIGH);
        LED_Status = true;
  }
  else if (LED_Schedule.isActive()) {
    digitalWrite(MYled, HIGH);
        LED_Status = true;
  }
  else {
    digitalWrite(MYled, LOW);
        LED_Status = false;
  }
/*--------------------(Main_Pump Schedule)--------------------*/
if (Main_Pump || Main_Pump_Schedule.isActive()) {
//    Serial.println("MainPump active.");

  digitalWrite(MainPump, HIGH);
    Main_Pump_Status = true;
} else {
//   Serial.println("MainPump OFF.");
  digitalWrite(MainPump, LOW);
    Main_Pump_Status = false;
}
/*--------------------(Abort!!! Turn motors and led off)--------------------*/
} else {
    myMotor1->run(RELEASE); // Stop motor 1
    myMotor2->run(RELEASE); // Stop motor 2
    myMotor3->run(RELEASE); // Stop motor 3
    myMotor4->run(RELEASE); // Stop motor 4
   Motor1 = false;
    Motor2 = false;
      Motor3 = false;
        Motor4 = false;
  Motor1_Status = false;
    Motor2_Status = false;
      Motor3_Status = false;
        Motor4_Status = false;
  digitalWrite(MainPump, LOW);
```

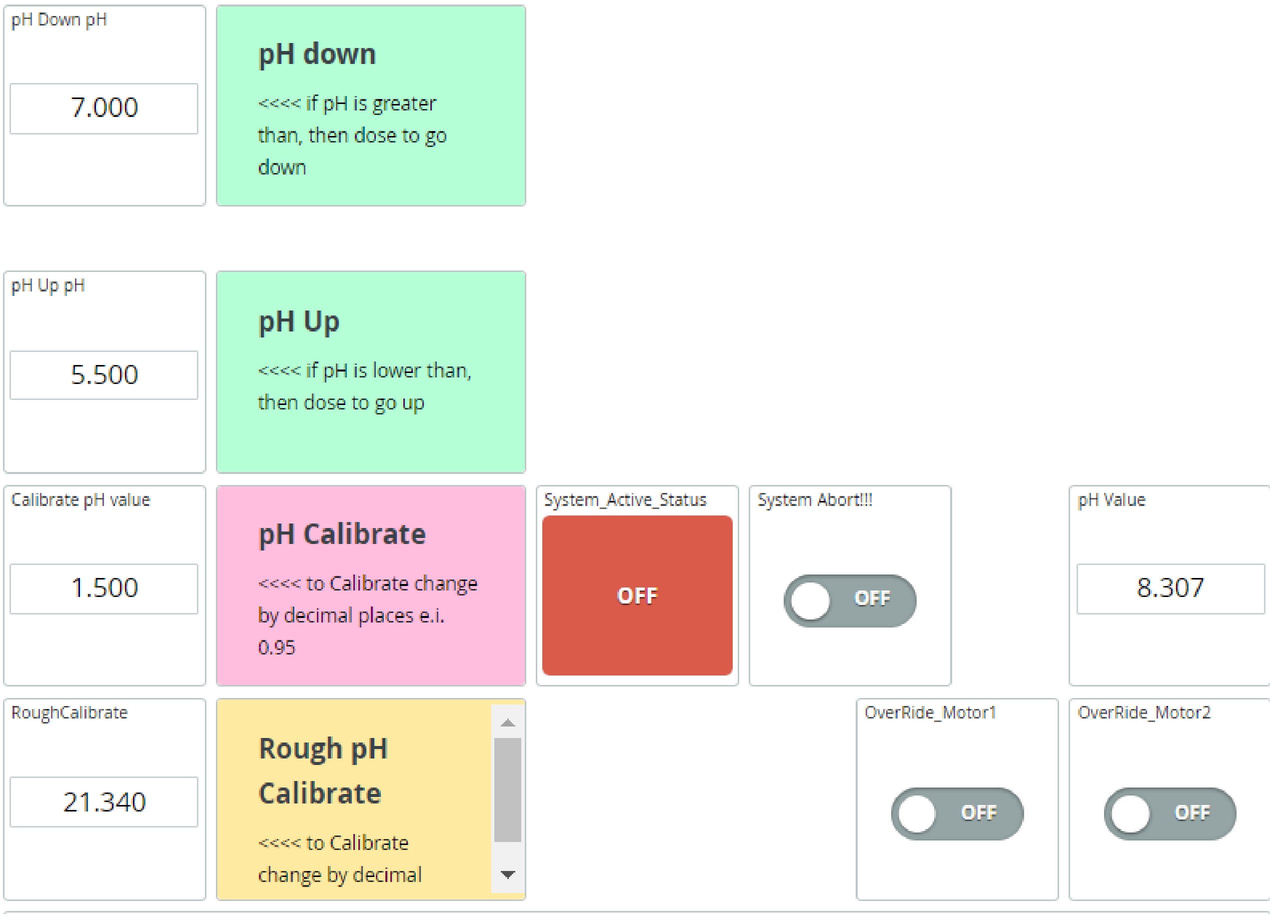
```
   digitalWrite(MYled, LOW); //turn led off
   led = false;
   LED_Status = false;
Serial.println("System Aborted!!!");
System_Active_Status = false;
}
}
/*====================[end of loop]====================*/
/*====================[void Functions]====================*/
    /*--------------------(void onMotor1Change())--------------------*/
/*
  Since Motor1 is READ_WRITE variable, onMotor1Change() is
  executed every time a new value is received from IoT Cloud.
*/
/*--------------------(void onPHChange())--------------------*/
/*
  Since Led is READ_WRITE variable, onLedChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onLedChange()  {
  // Add your code here to act upon Led change
  if (led || LED_Schedule.isActive()) {
    digitalWrite(MYled, HIGH);
    LED_Status = true;
  } else {
    digitalWrite(MYled, LOW);
    LED_Status = false;
  }
}
/*--------------------(void onLEDScheduleChange())--------------------*/
/*
  Since LEDSchedule is READ_WRITE variable, onLEDScheduleChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onLEDScheduleChange()  {
  if (led || LED_Schedule.isActive()) {
    digitalWrite(MYled, HIGH);
    LED_Status = true;
  } else {
    digitalWrite(MYled, LOW);
    LED_Status = false;
  }
}
/*--------------------()--------------------*/
/*
  Since PHDownValue is READ_WRITE variable, onPHDownValueChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onPHDownValueChange()  {
  // Add your code here to act upon PHDownValue change
  pH_Down_Value_Change = pH_Down_Value;
  Serial.println(pH_Down_Value_Change);
  UpdatePHValueChange(); }
/*
  Since PHUpValue is READ_WRITE variable, onPHUpValueChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onPHUpValueChange()  {
  // Add your code here to act upon PHUpValue change
  pH_Up_Value_Change = pH_Up_Value;
  Serial.println(pH_Up_Value_Change);
  UpdatePHValueChange(); }
void UpdatePHValueChange(){

  pH_Up_Value_Change = pH_Up_Value;

  pH_Down_Value_Change = pH_Down_Value; }
/*--------------------()--------------------*/
/*
  Since CalibratePHSensor is READ_WRITE variable, onCalibratePHSensorChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onCalibratePHSensorChange()  {
  // Add your code here to act upon CalibratePHSensor change
  CalibrationValue = Calibrate_pH_Sensor;
  UpdateCalibrationPH(); }
/*
  Since CalibrationValueRough is READ_WRITE variable, onCalibrationValueRoughChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onCalibrationValueRoughChange()  {
  calibrationValueRough = CalibrationValueRoughSAVE;
  UpdateCalibrationPH(); }
void UpdateCalibrationPH(){
    CalibrationValue = Calibrate_pH_Sensor;
    calibrationValueRough = CalibrationValueRoughSAVE;
    Serial.println(CalibrationValue);
    Serial.println(calibrationValueRough); }
```

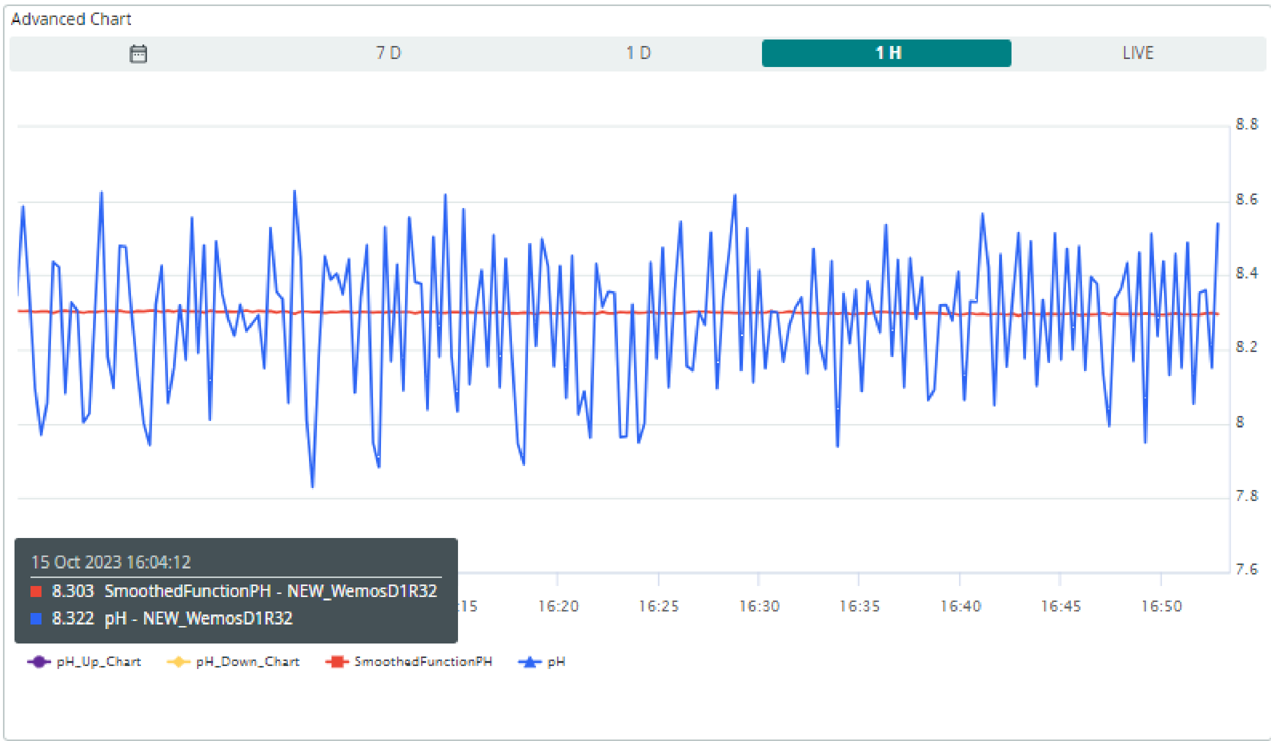
OLD IoT Cloud Dashboard



Improved IoT cloud Dashboard




pH Down pH
7.000
pH down
<<<< if pH is greater than, then dose to go down
pH Up pH
5.500
pH Up
<<<< if pH is lower than, then dose to go up
Calibrate pH value
1.500
pH Calibrate
<<<< to Calibrate change by decimal places e.i. 0.95
System_Active_Status
OFF
System Abort!!!
OFF
pH Value
8.307
RoughCalibrate
21.340
Rough pH Calibrate
<<<< to Calibrate change by decimal
OverRide_Motor1
OFF
OverRide_Motor2
OFF


Advanced Chart
7 D
1 D
1 H
LIVE
8.8
8.6
8.4
8.2
8
7.8
7.6
15 Oct 2023 16:04:12
8.303 SmoothedFunctionPH - NEW_WemosD1R32
8.322 pH - NEW_WemosD1R32
16:20
16:25
16:30
16:35
16:40
16:45
16:50
pH_Up_Chart
pH_Down_Chart
SmoothedFunctionPH
pH

Final Product

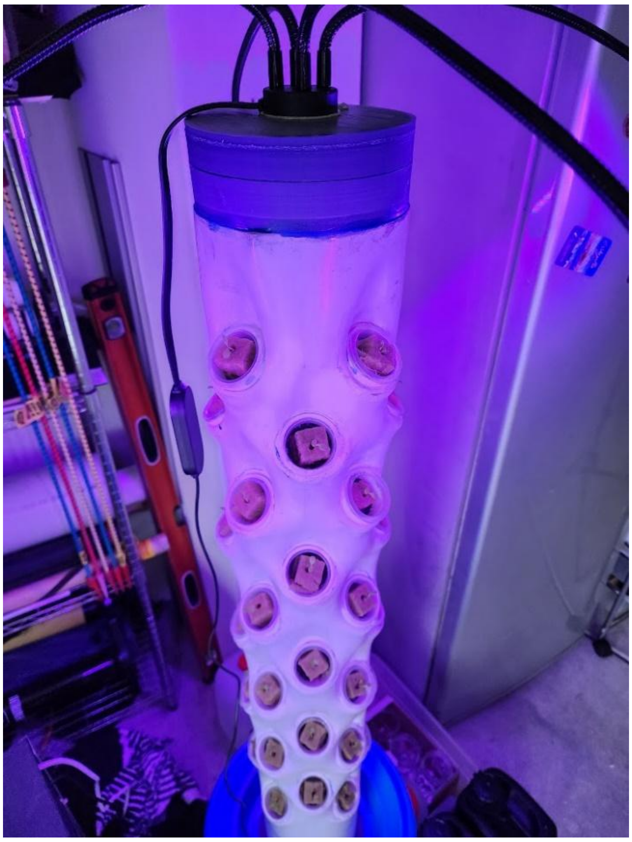


Figure 172: *Finished hydroponics tower*

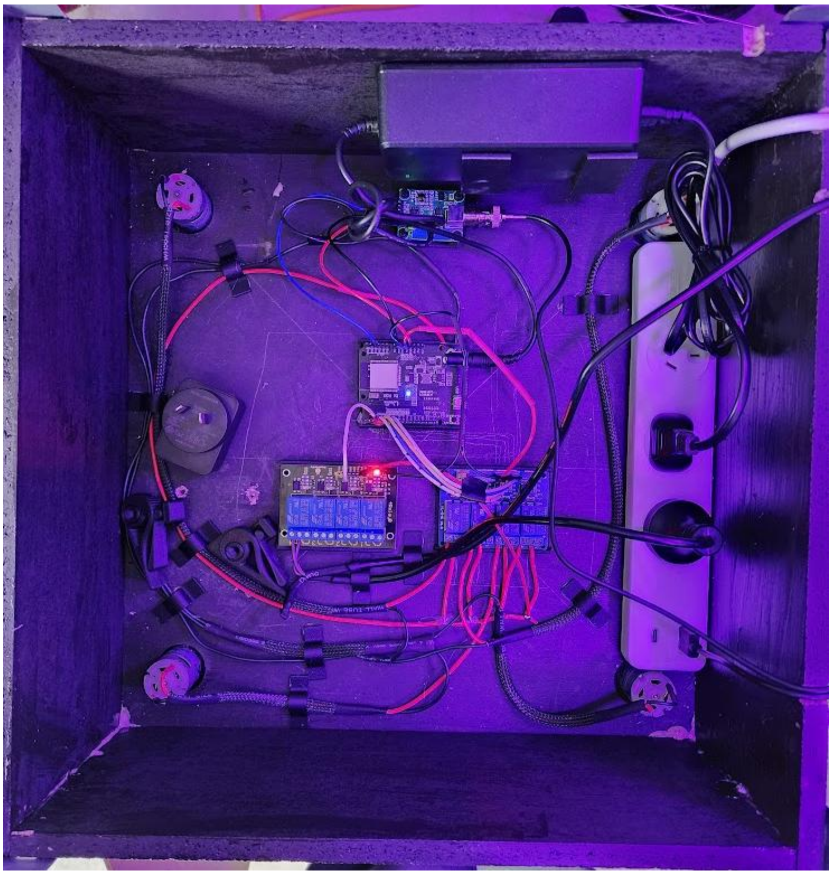


Figure 173: *Finished hydroponics electronics*

Figure 174: *Started growing plants.*

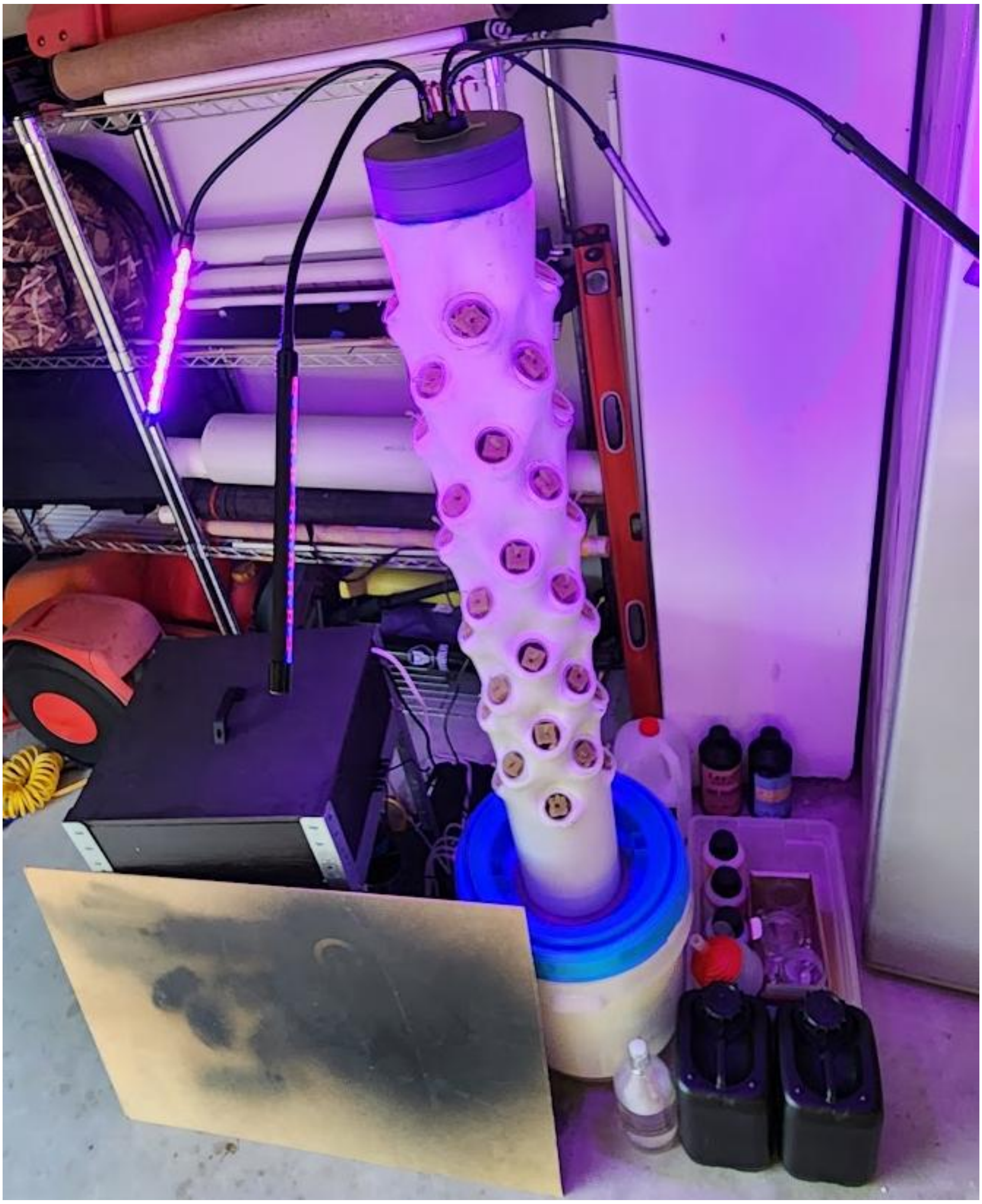


Figure 175: *Finished working setup of the hydroponics system.*

To watch the video of process of making, IoT cloud Explanation and final product go to:
https://youtu.be/O1UjPc237cc

Evaluation:

The focus of the automated hydroponics system had design requirements that were followed throughout the project. The project was designed and created for a quality functional design, the end product encompasses the following features:

- **Control via Arduino IoT Cloud**: Integration with the Arduino IoT Cloud allows the user to monitor and control various aspects of the hydroponic system, enhancing accessibility and convenience.
- **Large Plant Capacity**: With the ability to house 44 plants, your system is suitable for substantial crop cultivation, making it efficient for larger-scale farming with large plant varieties.
- **Automated pH Control**: The system includes pH control, ensuring that the pH level of the nutrient solution is maintained within the desired range for optimal plant health.
- **Nutrient Dosing**: Nutrient delivery is automated based on a schedule. This feature ensures that plants receive the right amount of nutrients at the right times, promoting growth.
- **Scheduling**: The system is capable of scheduling various tasks, such as nutrient dosing and potentially other operations like lighting and irrigation.
- **Sensor Integration**: The system includes a pH sensor to monitor pH level. The system can be expanded to more sensors such as nutrient sensors, and environmental sensors for data collection and control.

Design Criteria key:

Portability: To provide adaptability and versatility, the system must be portable, allowing users to set it up and relocate it as required.
Space Efficiency: The system should incorporate vertical or compact designs to make the most of available space and allow for the development of many plants to be space efficient.
Water Efficiency: By utilising water recycling and effective watering techniques, lowering waste, and encouraging sustainability through water efficiency, the hydroponics system should be created to use as little water as possible.
Nutrient Delivery: The system should ensure that nutrient solutions are delivered to the plants effectively and accurately, maximising their health and growth.
Modularity and Scalability: Flexibility in the design should enable simple expansion or modification based on the user's requirements and the available area. It should also allow for different plant sizes and species.
User-Friendly Operation: Users of all levels of competence should be able to use and maintain the system successfully due to the system's intuitive design, user-friendliness, and clear instructions and controls.

Addressing the first design requirement, which concerns **portability**, the current design unfortunately, falls a bit short regarding portability. It necessitates the use of a trolley for the relocation of materials. However, the potential for enhanced portability remains. Future modifications could include a reduction in the number of plants, achievable using a shorter PVC pipe, which then will make the system more **portable.**

The **space efficiency** of the current design, accommodating 44 plants with a focus on small to medium-sized varieties, is a commendable achievement. The vertical and compact layout optimises the use of available **space efficiently**. However, for future enhancements, it might be worthwhile to explore ways to accommodate **larger plants**, such as tomatoes or taller herbs. This could involve adjusting the system's architecture to incorporate **sections with more vertical clearance**, creating a versatile setup capable of nurturing a broader range of plant sizes while maintaining its efficient use of space.

The water efficiency of the design is significant, primarily due to the system's water recycling of the vertical design. It cycles the water through the system effectively minimises waste and encourages sustainability by conserving water. To further enhance water efficiency, one possible improvement in the future could be the implementation of an advanced nutrient filtering system. This system would not only recycle water but also filter and recirculate the nutrient solution, thereby reducing nutrient wastage as the nutrients needs to be replaced every month.

The water is what distributes the **nutrients** to the plants and relies on scheduled dosing of the required **nutrients** per week. There is no way to measure the **concentration of nutrients** so the schedule doses weekly (By the Arduino IoT Cloud) the approximate amount needed. To further enhance nutrient delivery accuracy, the integration of an EC/PPM sensor could be considered for future improvement. This sensor would enable real-time monitoring of the **nutrient concentration** and allow for precise adjustments to optimise plant growth.

The design effectively achieves **modularity and scalability** as it can control multiple hydroponics systems using the same code. This expandability justifies the system's initial cost, making it a versatile solution for accommodating various plant sizes and species, meeting users' changing needs and available space.

The system excels in **user-friendliness**, with an intuitive online dashboard (Arduino IoT Cloud) and clear instructions for operation and maintenance. Given its **automated features, users** of all competence levels can easily manage the system. To further enhance **user-friendliness**, beta-testing with **random users** can provide valuable insights and suggestions for continual improvement, making it even more **user-centric.**

References

Admin (2022). Home Automation using ESP32 & Blynk 2.0. [online] IoT Projects Ideas. Available at: https://iotprojectsideas.com/home-automation-using-esp32-blynk-2-0/ [Accessed 7 Apr. 2023].

Barth, B. (2018). How Does Aeroponics Work? - Modern Farmer. [online] Modern Farmer. Available at: https://modernfarmer.com/2018/07/how-does-aeroponics-work/ [Accessed 7 Apr. 2023].

C, J. (2022). D1 R32 – ESP32. [online] Design Tech. Available at: https://designtech.blogs.auckland.ac.nz/d1-r32-esp32/ [Accessed 18 Jul. 2023].

clipdrop.co. (2023). Clipdrop - Stable Doodle. [online] Available at: https://clipdrop.co/stable-doodle [Accessed 4 Apr. 2023].

Espiritu, K. (2014). Deep Water Culture (DWC): What Is It And How To Get Started. [online] Epic Gardening. Available at: https://www.epicgardening.com/deep-water-culture-get-started/ [Accessed 13 Jul. 2023].

ForceTronics (2022). Controlling Water Flow with a Solenoid and Arduino IoT Cloud Part 2. [online] www.youtube.com. Available at: https://www.youtube.com/watch?v=v6e3KRIlFBM [Accessed 5 May. 2023].

Jeena, L. (2023). Hydroponic Wick System – The Ultimate Guide. [online] Gardening Heavn. Available at: https://gardeningheavn.com/hydroponic-wick-system/ [Accessed 9 May. 2023].

Last Minute Engineers (2018). Interface DS3231 Precision RTC Module with Arduino. [online] Last Minute Engineers. Available at: https://lastminuteengineers.com/ds3231-rtc-arduino-tutorial/ [Accessed 19 Jun. 2023].

Lee, A. (2017). Ebb And Flow Table Hydroponic System Rolling Bench For Medicine at Best Price in Zhengzhou | Henan Wisemax Agricultural Technology Co,.Ltd. [online] Tradeindia. Available at: https://www.tradeindia.com/products/ebb-and-flow-table-hydroponic-system-rolling-bench-for-medicine-5177535.html [Accessed 5 Jun. 2023].

Nadaraja, P. (2022). How To Set Up A Hydroponic Drip System - Hydroponic Way. [online] hydroponicway.com. Available at: https://hydroponicway.com/how-to-set-up-drip-hydroponics/ [Accessed 1 Apr. 2023].

NYH-workshop (2021). Wemos R32 with Arduino - Startup Guide! [online] Hackster.io. Available at: https://www.hackster.io/NYH-workshop/wemos-r32-with-arduino-startup-guide-7bc841 [Accessed 3 Apr. 2023].

ROBERT, E. (2022). Wemos D1 R32 – IO pin assignment for Arduino UNO compatibility – Emilian Robert Vicol. [online] Vicol Emilian Robert (byrev). Available at: https://robertvicol.com/tech/wemos-d1-r32-io-pin-assignment-for-arduino-uno-compatibility/ [Accessed 3 May. 2023].

Söderby, K. (2023). IoT Cloud Dashboards & Widgets | Arduino Documentation. [online] docs.arduino.cc.

Available at: https://docs.arduino.cc/arduino-cloud/getting-started/dashboard-widgets [Accessed 17 Oct. 2023].

Supriya, N. (2022). Types of Hydroponic Systems - Advantages and Disadvantages. [online] Biology Reader. Available at: https://biologyreader.com/types-of-hydroponic-systems.html [Accessed 3 May. 2023].

Szczys, M. (2022). Golioth and Datacake enable device data visualization with a few clicks. [online] Golioth. Available at: https://blog.golioth.io/golioth-and-datacake-enable-device-data-visualization-with-a-few-clicks/ [Accessed 9 Apr. 2023].

techiesms (2021). Using Arduino IoT Cloud with ESP32 | Is it better than New Blynk?. [online] www.youtube.com. Available at: https://www.youtube.com/watch?v=rcCxGcRwCVk [Accessed 7 May. 2023].

Teja, R. (2021). Arduino UNO Pinout, Specifications, Board Layout, Pin Description. [online] Electronics Hub. Available at: https://www.electronicshub.org/arduino-uno-pinout/ [Accessed 5 May. 2023].

Trees.com Staff (2017). What is the Nutrient Film Technique - NFT? How does it work? [online] Trees.com. Available at: https://www.trees.com/gardening-and-landscaping/nutrient-film-technique [Accessed 2 Apr. 2023].

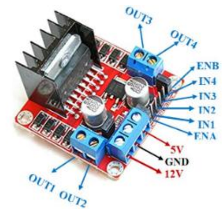
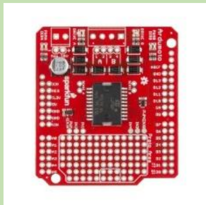
Appendix

Investigation and Analysis

Processor

Device	Processor	Board	Price (Approx)	Additional information	Picture
Raspberry Pi	Broadcom BCM2711	Raspberry Pi 4 Model B	$35 - $55	A versatile single-board computer with multiple models, ideal for various projects and applications.	 https://raspberry.piaustralia.com.au/products/raspberry-pi-4
Wemos D1 R32	Espressif ESP32	Wemos D1 R32	$10 - $70	A compact development board featuring the ESP32 microcontroller, suitable for IoT projects and prototyping.	 https://www.makerstore.com.au/product/elec-esp32-d1-r32/
Arduino Uno	Microchip ATmega328P	Arduino Uno Rev3	$20 - $50	A popular microcontroller board for a wide range of embedded system projects, known for its simplicity and ease of use.	https://core-  electronics.com.au/arduino-uno-r3.html

Motor drivers/shield/components

Product Name	Description	Motor Compatibility/ Key features	Voltage/ Current Ratings	Control Interface	Application / Availability	Picture
Adafruit Motor Shield v2	The Adafruit Motor Shield v2 is a versatile shield that can control up to four DC motors or two stepper motors. It's easy to use and comes with built-in PWM control for speed regulation. Suitable for various motor-based projects.	DC, Stepper. 4 DC motors or 2 stepper motors, PWM control	4.5V - 25V, 1.2A per motor	I2C and PWM control	Robotic projects, automation, robotics clubs. Adafruit, Online retailers	 https://learn.adafruit.com/adafruit-shield-compatibility/motor-slash-stepper-slash-servo-shield-v2
L298N Motor Driver	The L298N Motor Driver is a popular dual H-bridge module that can control two DC motors. It is capable of driving motors with higher current requirements and is commonly used in robotics applications.	DC, Stepper. Dual H-bridge design, adjustable motor speed	4.8V - 35V, 2A per motor	GPIO control	DIY robots, smart vehicles, educational projects. Various electronics suppliers	 https://components101.com/modules/l293n-motor-driver-module
SparkFun Ardumoto Shield	The SparkFun Ardumoto Shield is designed to control two DC motors with an Arduino. It offers a compact and simple solution for motor control, making it ideal for small to medium-sized projects.	DC. 2 DC motors, built-in voltage regulation	7V - 12V, 2A per motor	GPIO control	Robotics, remote-controlled vehicles, STEM projects. SparkFun, Online electronics stores	 https://core-electronics.com.au/sparkfun-ardumoto-motor-driver-shield.html
BTS7960B Motor Driver Module	The BTS7960B is a high-power H-bridge motor driver capable of driving heavy loads. It can handle high current and voltage requirements, making it suitable for larger robotic and automation systems.	DC. High current capacity, over-current protection	5V - 27V, 43A	PWM control (	Electric scooters, e-bikes, industrial automation. Various online retailers, electronics shops	 https://alselectro.wordpress.com/2021/11/08/6632/
Genuino Motor Shield / Motor Driver Shield for Arduino Clytron.	The Genuino Motor Shield is designed for use with the Genuino (Arduino) platform. It can control two stepper motors or four DC motors and offers a user-friendly solution for motor control with Arduino boards.	DC, Servo. 4 DC motors, 2 Servo motors, dual H-bridge design	7V - 12V, 1.2A per motor	GPIO control	Educational robotics, Arduino-based projects. Arduino Store, Online electronics shops	 https://my.cytron.io/p-1.2amp-7v-30v-dc-motor-driver-shield-for-arduino-2-channels

I2C (Inter-Integrated Circuit): I2C is a two-wire communication protocol for connecting and exchanging data between multiple devices, commonly used for sensors and integrated circuits.

PWM (Pulse Width Modulation) Control: PWM is a method of regulating power by rapidly turning it on and off, often used to control motor speed, LED brightness, and servo positioning.

GPIO (General-Purpose Input/Output) Control: GPIO pins on microcontrollers can be configured as inputs to read digital signals or outputs to send digital signals, making them versatile for interfacing with various components.

The Wemos D1 R32 is suitable for this project due to its compact design, efficient water and nutrient control, scalability, and user-friendly IoT integration, aligning well with the design criteria of portability, space efficiency, water efficiency, modularity, and user-friendly operation.

Growing Medium

Growing Medium	Physical Characteristics	Water Retention	Aeration and Drainage	pH Control	Nutrient Absorption	Cost Considerations	Picture
Rockwool	Rockwool is a lightweight and fibrous medium that excels at retaining moisture.	It retains water at a lower level, so less water is needed.	Offers excellent aeration and efficient drainage.	It usually requires a pH buffer for pH control.	Demonstrates excellent nutrient absorption.	It falls within a moderate cost range.	
Perlite	Perlite is a lightweight and porous medium, known for its low water retention.	It has a low capacity for retaining water, so it more water is needed.	Provides good aeration and reasonable drainage.	Maintains a near-neutral pH balance.	Shows less nutrient absorption capability.	It is cost-effective and budget friendly.	
Coconut Coir	Coconut coir is a fibrous medium that retains moisture well, with high water retention.	It retains water effectively and has a high water retention.	Offers moderate aeration and drainage properties.	Typically requires a pH buffer for pH control.	Provides moderate nutrient absorption capacity.	It falls within a moderate cost range.	
Specialised Substrate	Specialised substrates vary in characteristics, so specific attributes depend on the chosen product.	Characteristics can vary, which influences water retention.	Attributes differ based on the selected substrate.	pH control can vary depending on the substrate.	Nutrient absorption varies by product.	Costs vary according to the selected substrate.	

Rockwool was chosen for its excellent water retention properties, inert nature, and suitability for plant growth, aligning with the project's hydroponic requirements.

Grow Lights

Type of Grow Light	Suitable Growth Stages	Energy Efficiency	Customisation Options	Picture
LED Grow Lights	All growth stages	Highly efficient	Customisable spectra	
T5 Grow Lights	Seedlings and young	Energy-efficient	Limited customisation	
HID Grow Lights (MH and HPS)	MH for vegetative stage HPS for flowering stage	Variable	Limited customisation	
Full Spectrum Grow Lights	All growth stages	Balanced spectrum	Limited customisation	

Type of Pumps

Pump Type	Description	Common Use Cases	Photo
Submersible Water Pumps	These pumps are designed to operate underwater. They are often used for water circulation, irrigation, and nutrient delivery in hydroponic systems.	Circulating water in deep water culture (DWC), flood and drain, and nutrient film technique (NFT) systems.	 https://www.amazon.com.au/TEEMO-Aquarium-Submersible-Water-Pump/dp/B0BHN96N9B
Air Pumps (for Aeration)	Air pumps are used to infuse oxygen into the nutrient solution, promoting aeration and providing essential oxygen to plant roots. They work in conjunction with air stones or diffusers.	Aerating nutrient solutions, preventing root rot, and promoting healthy root growth in various hydroponic systems.	 https://sunshoweronline.com.au/pondmax-aeration-pump-kits/
Peristaltic Pumps	Peristaltic pumps use rotating rollers to squeeze a flexible tube, creating a flow of liquid. They are known for their precise and controllable flow rates.	Accurate dosing of nutrients, pH adjusters, and other solutions in hydroponics. Often used in automated dosing systems.	 https://www.tdapumps.com.au/product/l-series-peristaltic-pumps
Centrifugal Pumps	Centrifugal pumps use a spinning impeller to move liquid. They offer high flow rates and are suitable for large-scale hydroponic systems.	Large-scale water circulation and filtration systems, such as in commercial hydroponic farms.	 https://www.serfilco.com/products/pumps/magnetic-coupled-pumps/uc-centrifugal-pump/
Venturi Pumps	Venturi pumps operate by creating a vacuum through the Venturi effect, drawing in nutrients or air into the water flow. They require high water pressure.	Adding nutrients to the water stream, especially in nutrient reservoirs and circulation systems.	 https://www.thejettersedge.com.au/products/venturi-pump
Recirculating Pumps	Recirculating pumps are designed for closed-loop systems. They help maintain a consistent flow of nutrient solution, promoting even distribution and preventing stagnation.	Constantly circulating nutrient solutions in deep water culture (DWC), aeroponics, and other closed-loop hydroponic setups.	 https://www.bigbplumbing.com/blog/the-benefits-of-a-hot-water-recirculating-pump/

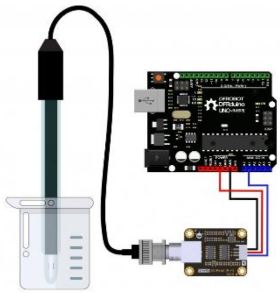
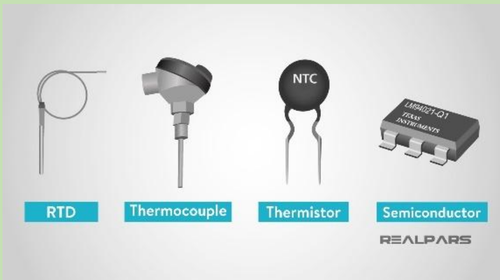
Different types of materials investigation

Material Type	Description	Applications	Strength and Durability	Cost Considerations	Environmental Impact
ABS vs. PLA	ABS (Acrylonitrile Butadiene Styrene) is a strong and durable thermoplastic known for its impact resistance. PLA (Polylactic Acid) is a biodegradable, plant-based thermoplastic with moderate strength.	ABS is suitable for functional parts in engineering.- PLA is commonly used for artistic projects, prototypes, and biodegradable applications.	- ABS is robust. - PLA is less durable.	- ABS is moderately priced. - PLA is generally affordable.	- PLA is biodegradable and environmentally friendly.
Plywood vs. Acrylic vs. MDF	- Plywood is an engineered wood product made from thin layers of wood veneer. - Acrylic is a transparent plastic material. - MDF is a dense composite wood product.	- Plywood is often used in crafts and furniture. - Acrylic is ideal for intricate designs and signage. - MDF is suitable for intricate projects, prototypes, and architectural models.	- Plywood is strong. - Acrylic is durable but can scratch. - MDF is dense but less durable than solid wood.	- Plywood is moderately priced. - Acrylic can be more expensive. - MDF is budget-friendly.	- Plywood and MDF vary in their environmental impact based on sourcing and adhesives used.
Solid Timber Varieties (Pine, Oak, Jarrah)	- These are natural, solid timber materials with unique textures and properties.	- Solid timbers are used for decorative elements, inlays, or artistic projects.	- They are known for their strength and durability, with variations between species.	- Costs can vary significantly based on the timber species.	- Sourcing from sustainably managed forests is important for reducing environmental impact.

Relays and Actuators

Component	Description	Applications	Advantages	Considerations	Photos
Relay Modules	Control various components such as pumps and lights in the hydroponics system.	Automation of key functions within the system.	Versatile and easy to interface with microcontrollers.	Proper setup and isolation are essential to prevent electrical interference.	 https://core-electronics.com.au/5v-2-channel-relay-module-10a.html
Solenoid Valves	Regulate the flow of water and distribution of nutrients within the system.	Precise control of water flow and nutrient delivery.	Reliable and durable for long-term operation.	Requires proper maintenance to prevent clogs or leaks.	 https://www.amazon.com.au/Solenoids-Applicable-Mechanical-Engineering-Automation/dp/B09M8NMX72
Servo Motors	Utilised for tasks like adjusting light fixtures, opening and closing vents, and other precision movements.	Fine-tuned control of specific mechanisms in the system.	High precision and repeatability in positioning.	May have limited torque for heavy applications.	 https://www.jaycar.com.au/arduino-compatible-9g-micro-servo-motor/p/YM2758

Sensors

Component	Description	Photo
pH Sensor	Measures the pH level of the nutrient solution.	 https://au.element14.com/dfrobot/sen0161/analog-ph-sensor-meter-kit-arduino/dp/2946120
EC Sensor	Monitors the electrical conductivity, which corresponds to nutrient concentration.	 https://core-electronics.com.au/analog-electrical-conductivity-meter-with-temperature-compensation.html
Temperature Sensor	Ensures the nutrient solution and environment are at the optimal temperature.	 https://realpars.com/temperature-sensor/
Humidity Sensor	Monitors humidity levels for plant health and disease prevention.	 https://au.element14.com/honeywell/hih-4010-002/humidity-sensor-40-to-85deg-c/dp/3408766

A temperature sensor is a device that measures the ambient temperature of its surroundings and converts that measurement into an electrical signal. These sensors are crucial components in hydroponics systems, helping to ensure that the growing environment is maintained at an optimal temperature range for plant growth. They come in various forms, including analogue and digital sensors, and are often integrated with microcontrollers to enable automated control of heating or cooling systems, ensuring that plants receive the ideal temperature conditions for healthy growth. The data from temperature sensors is also valuable for monitoring and making necessary adjustments to maintain a stable and conducive environment for hydroponic cultivation.

The Flow rate restrictor Testing flow rate:

Figure 162: *With switching to relays, flow restrictors had to be printed to adjust dosage flow.*

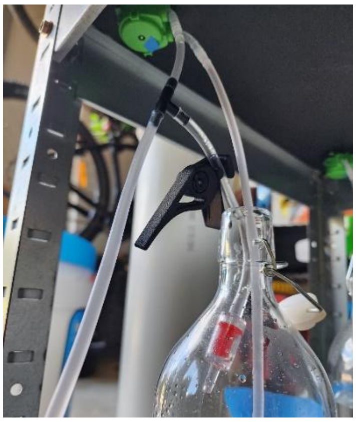


Figure 165: *Positive displacement pump with flow rate restrictor for manual adjustment of flow.*

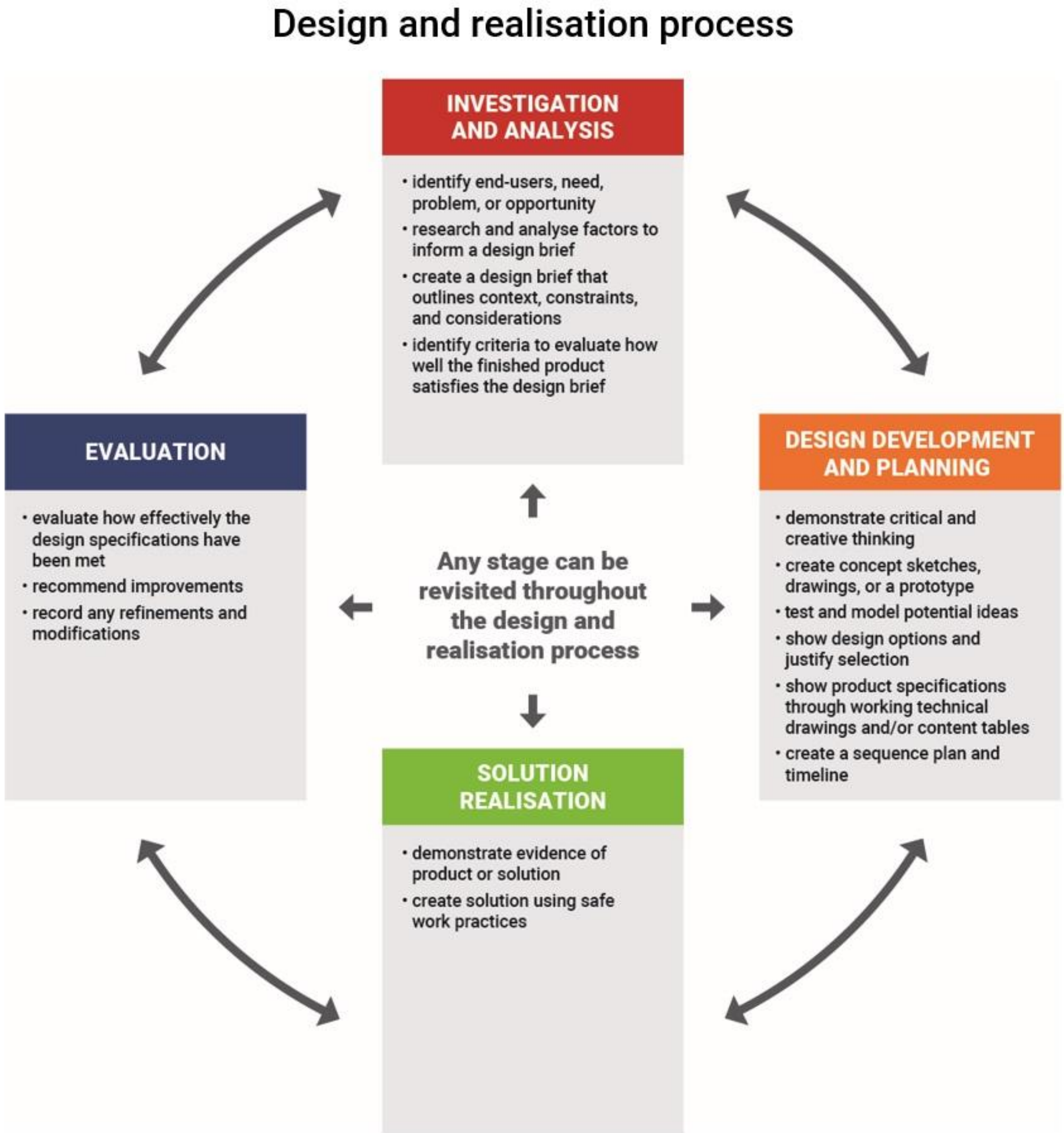
Flow restrictor	Flow rate trail 1 (seconds) (5.D.P)	Flow rate trail 2 (seconds) (5.D.P)	Flow rate trail 3 (seconds) (5.D.P)	Average Flow (ml/s) (5.D.P)
Placed on	100ml time: 13:02.93 100/783 = 0.12771	100ml time: 13:03.98 100/784 = 0.12755	100ml time: 13:02.84 100 / 783 = 0.12771	0.12766
Full Clicks	500ml time 07:18.47 500/438 = 1.14155	500ml time 07:18.12 500/438 = 1.14155	500ml time 07:17.42 500/437 = 1.14416	1.14242

The significant difference in flow rates between using flow rate restrictors with the peristaltic pump and without them can be attributed to the control mechanisms and principles of the peristaltic pump itself.

1. **Full clicks: Peristaltic Pump with Flow Rate Restrictors (1.14242 ml/s):**
    - A peristaltic pump operates by squeezing a flexible tube at various points along its length to create a series of controlled "pulses." These pulses propel the fluid within the tube forward.
    - Flow rate restrictors can be placed on the tube to regulate the timing and intensity of these pulses. By adjusting the restrictors, can control the rate at which the tube is compressed, which, in turn, controls the flow rate.
2. **Placed on: Peristaltic Pump Without Flow Rate Restrictors (0.12766 ml/s):**
    - When the peristaltic pump operates without flow rate restrictors, it delivers a consistent, slower flow rate. This is because the pump generates its pulses without external regulation, which results in a relatively constant and lower flow rate.

The flow rate restrictors allow the ability to fine-tune and increase the flow rate of the peristaltic pump by controlling how the pump compresses the tube and delivers fluid. Without them, the pump delivers a more consistent but slower flow rate. The choice between these two operating modes depends on the specific requirements of your hydroponics system, such as the desired nutrient dosage rate or irrigation frequency for your plants.

The Design and Realisation Process that was followed:



DESIGN, TECHNOLOGY AND ENGINEERING. (n.d.). Available at: https://www.sace.sa.edu.au/documents/652891/88457b34-845e-9f26-0685-360e6ad926b4#:~:text=The%20design%20and%20realisation%20process%20should%20begin%20with%20the%20identification [Accessed 1 Oct. 2023].