References:

<https://kylegabriel.com/projects/2020/06/automated-hydroponic-system-build.html#comments>

**Background info**

Currently we work with light devices that have ESP32 (wifi) that can connect to AWS, using MQTT from AWS IoT core, and a Front web UI that uses json messages to AWS. But for hydroponics we need more than that. Hence we want to make a hydroponic system.

We make lighting ourselves. Also we can find the Hydroponic NFT channels easily. For all the other parts (raspberry pi based) we would need your suggestion.

Milestones for Hydroponic control system:

## Main Milestones:

1. Interface of pi with hardware

2. Sending data to the web

3. Design of admin panel (multi users)

4. Communication and control between hardware, web and admin panel

5. Testing and deployment on live server

**Coding languages:**

Hardware: python

Web back-end: php and codeignitor

Web front-end: HTML5/ CSS3 and javascript or react?

Protocol hardware and web: JSON based REST APIs

Database: mysql

User roles:

Super admin: understand usage of the app, understand what features are most frequently used (technical data to monitor app stability)

Admins: admins specific for a group of users/ account (they can do the initial setup and change settings, approve requests)

Operators: users at the lowest level, they can see things and turn things on/ off, but cannot change settings or remove devices or add devices

## Milestone 1 Interface of pi

Things Todo:

1. List of components, make basic schematic of system
2. Interface pi with all sensors and actuators
3. See output of sensors and actuators

**Hardware Sensors to capture data:**

Water sensors: Temperature, pH, EC, Flow, Level, Dissolved oxygen

Air sensors: Temperature, Humidity, Vapor Pressure Deficit (VPD), CO2,

Light sensor: quantum sensor (400-730nm)

Various sensors: DB (noise), Microphone, Weight scale

Power: measure power consumption

Camera: simple camera for still photo’s, DSLR or iPhone for time-lapses, infrared camera for photosynthesis

**Hardware Actuators that can be controlled:**

Water actuator through 4 Peristaltic pumps: pH down, pH up, Nutrient A, Nutrient B, air pump and air stone, Water heater

Air actuator: Humidifier, Dehumidifier, Fan, Aircon (on/off)

Light actuator: Brightness control (0-255 for max 8 channel)

Various actuator: Thermal pad, Speaker

## Milestone 2 Sending data to the web

JSON based REST APIs

Authentication of users/ location?

## Milestone 3 Design admin panel

Based on material design principles with HTML5/ CSS3 and javascript or react

For UI use MyCodo as a reference <https://github.com/kizniche/Mycodo>

1. Login/Registration
2. Dashboard
   1. Water Temperature, pH, EC, Flow, Level, Dissolved oxygen, Ambient Temperature, Relative Humidity, Air pressure, Vapor Pressure Deficit (VPD), CO2, Light amount, DB, Weight, Power consumption, still photo, timelapse, infrared image [value, real-time graph with daily/weekly/yearly view]
3. All Locations
   1. Set target values [manual adjustment or auto]
4. All devices
5. Setting rules (App live/ off, approvals, notifications, format of Time, C or F)
6. Account
7. Visible for super admin only: app analytics: Google analytics, mixpanel (see location, browser, which actions or buttons are used most etc)

## Milestone 4 Communication and control between hardware, web and admin panel

Is this also JSON based REST APIs? Or MQTT?

AWS IoT Core?

Logical control based feedback

Automatically adjust water to target pH range by dispensing acid/base solutions from pH pumps.

Automatically adjust water to target electrical conductivity range by dispensing nutrient solutions from nutrient pumps.

Automatically adjust DO in the water system by air pump and air stone

Automatically adjust water temperature in cold environments by water heater

Automatically adjust air to target humidity, temperature, and CO2 concentration ranges by modulating an exhaust fan, humidifier, dehumidifier, .

Automatically adjust vapor pressure deficit in high humidity environments with a dehumidifier

Automatically regulate air vapor pressure deficit (VPD) with a humidifier and exhaust fan, using a PID controller.

Automatically adjust thermal pad temperature to regulate a consistent temperature for germinating trays of seeds in a nursery area

Use timers to schedule grow lights, air exhaust, and a DSLR camera shutter for high quality time-lapse photography.

E-mail alert notifications if select measurements fall outside acceptable ranges (e.g. temperature too high, water level too low, water flow has stopped, etc.).

Email approval or notification approval to admin if frequency of the pH or nutrients adjustment exceeds a certain number

Measure electrical energy usage with a transformer to automatically calculate operating costs.

Use DSLR or iPhone Camera to monitor plants with a live video stream and conduct time-lapse photography of plant growth.

## Milestone 5 Testing and deployment on live server

Testing/ debugging

Target domain: hortipower.app

There are many sub-tasks in each those are not mentioned here.

Others

Source-code will be handed over in zip or github.

Use-cases currently **excluded** but maybe part of the system in the future:

* Super admin to manage pricing of the app
* Billing functionality

Further comments:

Here are login details for our light app [**http://lb.veztan.com:8000**](http://lb.veztan.com:8000/)/ login: jille@hortipower.com password: [jille@hortipower.com](mailto:jille@hortipower.com) 🡪 Lets not integrate with this platform right now. but see how we can use api’s to communicate now or in future.

And fyi: i also like this dashboard (not ours) [**https://panel.growflux.com**](https://panel.growflux.com/)/ email: jk@hortipower.com password: tryout