aPLANT: A Low Cost Greenhouse Manager System

Dylan McKeever
Ben Neuendorf
Devon Simmons

Advised by: Aleksander Malinowski

Table of Contents

I. Statement of the Problem

Most small to medium sized greenhouses have a distinct disadvantage when it comes to the price of plant management devices. While a relatively new concept, plant management on an individual basis has a very valid and measurable impact on the production of crops. Indoor greenhouses are the future of increased yield in limited space. Current devices on the market, require the user to have a sensor per pot to get individual control over a variable in the “growing” equation such as moisture control, ph control, and temperature control. Some variables are easily managed with other techniques ex. temperature control. The immobile elements of growing such as planting and moisture control are vastly benefitted by precise control. The modules made by other manufactures for this purpose are usually separate, and typically don’t expand functionality easily to more than one growing surface.

The need for these systems is steadily growing. Changing global climate, Growing food scarcity, and needs for water conservation lend to the value of systems that guard against potential problems such as these. The problem we have chosen to tackle with our project is the fiscal aspect of these systems. While other genuine systems exist and do as advertised, the barrier to entry is too high sometimes in the range of thousands of dollars. We believe that the hardware being used is key to solving most of the issues with cost because there exist a wide variety of low power, feature rich, and inexpensive chips that do exactly what is needed. We chose the ESP8266 for 2 of those main reasons and a few tertiary ones. The ESP8266 is inexpensive coming in at about 5$ a module, and is very robust for this price point with multiple modes and fantastic connectivity. The tertiary reasons are that we are very familiar with the python enabled version of this chip because of the TCP/IP class that our department offers, and it has a embedded C capability.

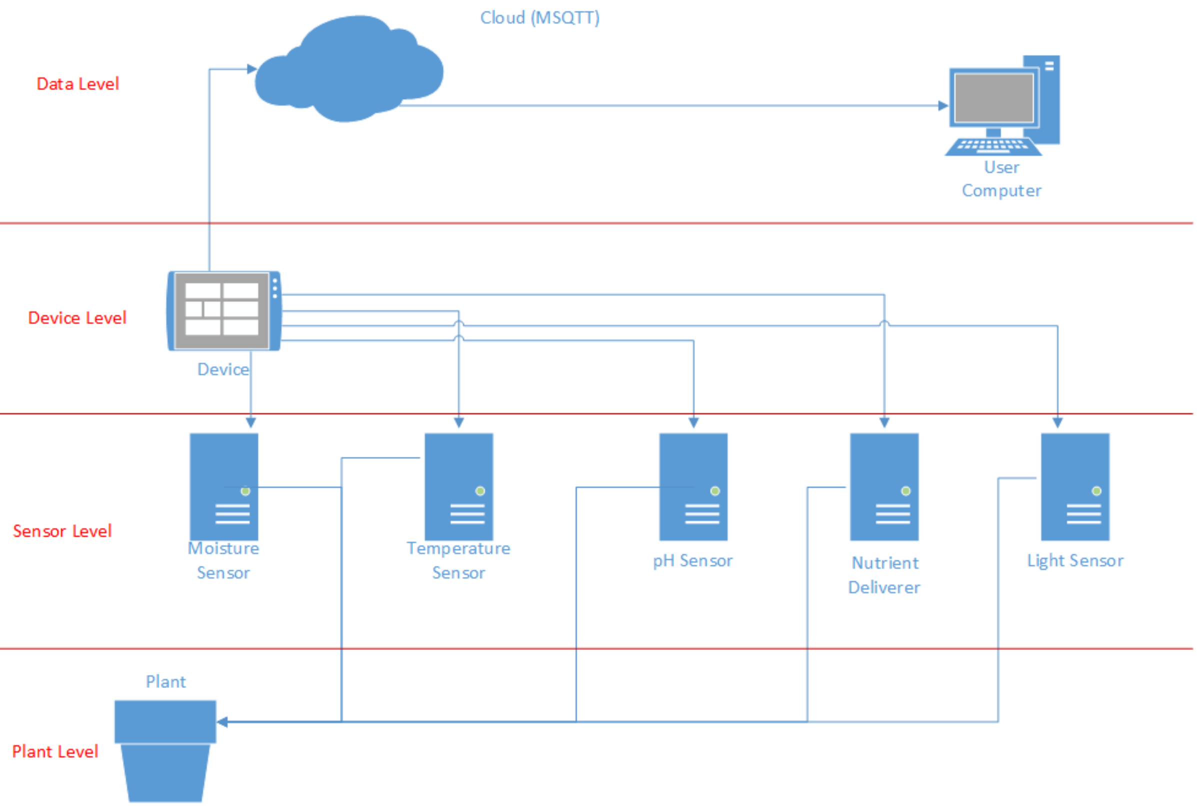

Figure 1: Flow Diagram

II. Functional Requirements

A. Purpose

We intend with this section to give the flow of the project and give a listing of the flow levels. This is not meant to show the physical configurations of the devices most of those will be done at the hardware level using the c environment that is provided by the manufacturer of the ESP8266. The labeling of Figure 1 is not to give a engineering description, but segment based on the users relationship to the system.

B. System Flow Levels

Data Level: The Data Level contains the cloud MQTT (Message Queuing Telemetry Transport) database that has the sensor data arranged into files per sensor type and further divided based on the specific pot from which the data was taken. It will be responsible for outputting the data to the user end to make evaluations from and set up the sensors.

Device Level: The Device Level contains the individual pots arranged by an esp8266 chip designated as the “Controller”. The controller serves as the vehicle for our modularity and maintains connections to the data going to the server, and the configuration requirements that the user wants to set. Our plan is to have a controller be an esp8266 device set solely for this purpose in this functional level.

Sensor Level: The Sensor Level contain all of the sensors and devices to be connected to the plant locally restricted to one esp8266 chip. The different sensors, from moisture to light, will be localized on one esp8266, but for the interest of showing how they will be initialized and implemented it is useful to show them as separate entities, as shown in the flow diagram in Figure 1 on page 3.

Plant Level: The Plant Level will only contain the plant itself, the ends of the sensors, and the connection to the esp8266 module that is interpreting the data and sending it upward to the Sensor Level.

III. Schedule of Work

<u>November 2018:</u>

Weeks 1&2

1. Write Project Proposal
2. Develop parts list and submit to Chris Mattus
3. Work on Website Homepage

Weeks 3&4

1. Develop Presentation Draft
2. Practice Presentation
3. Finalize Project Proposal
4. Present Project

<u>December 2018:</u>

Weeks 1&2

1. Finalize and release website with deliverables

Weeks 3&4

1. None (Winter Break)

January 2019:

Weeks 1&2

1. None (Winter Break)

Weeks 3&4

1. Configure Sensors and Connectivity

February 2019:

Weeks 1&2

1. Program Device-to-Device Communication

Weeks 3&4

1. Test Watering Features
2. Setup Database
3. Push Log Data to Database

March 2019:

Weeks 1&2

1. Register for Student Expo (by March 15th)

Weeks 3&4

1. Work on Final Report Rough Draft
2. Work on Senior Project Conference Abstract

April 2019:

Weeks 1&2

1. Work On Student Expo Poster
2. Practice Poster Presentation
3. Work on Final Presentation Draft

Weeks 3&4

1. Present Student Expo Poster on April 17th (setup April 16th)
2. Submit Presentation Draft on April 23rd (Hard Deadline)
3. Finalize and Practice Presentation

May 2019:

Week 1

1. Present at Senior Project Conference

IV. Division of Labor

Sensor and Baremetal Programming Controller Side - Devon
Light Sensor - Ben
Soil Level Humidity Sensor - Devon
Ambient Temperature - Ben
MQTT Connections - Dylan
Database - Dylan

V. References

1. Simon, M. (2017, November 20). The Hydroponic, Robotic Future of Farming in Greenhouses. Retrieved from https://www.wired.com/story/the-hydroponic-robotic-future-of-farming-in-greenhouses-at-iron-ox/
2. Thomas, P. A., Westerfield, R., & Pennisi, S. V. (2006, June 01). Growing Ferns. Retrieved from http://extension.uga.edu/publications/detail.html?number=B1318&title=Growing Indoor Plants with Success
3. CUBASCH, U. and WUEBBLES, D. (2018). *Fifth Assessment Report - Climate Change 2013*. [online] Ipcc.ch. Available at: https://www.ipcc.ch/report/ar5/wg1/.