# Abstract

HydroPWNics is an open source intelligent, connected, and autonomous hydroponic and dirt based garden control system. The goal is to design and develop an open source garden control system to facilitate sustainable independent farming. Existing systems for this purpose are mostly proprietary and expensive, HydroPWNics aims to be a fully open source and cost effective solution for everyone!

Overview

More specifically HydroPWNics is the system of hardware and software designed to control and monitor hydroponic gardens. The goal of such a system is to create a precisely controlled automated hydroponic garden. The benefit of such a garden would be autonomous food production, self-sustainability, and conserve water as a resource. The system consists of custom electronics and software integrated with and builtd on top of the already existing Neuron Robotics development kit. Hardware consists of the Neuron Robotics DyIO module, the HydroPWNics modules, and a Beaglebone Green. The DyIO is used in the AC power control box subsystem. The HydroPWNics modules are used for wireless plant and environmental sensing, and the Beaglebone Green runs the Java application directing all the hardware. The software consists of a Java application running on the Beaglebone Green for controlling the garden and a webui/website for viewing plant data and controlling the garden remotely. The Java application is written in the Neuron Robotics SDK and the HydroPWNics API, this application is what controls the garden, directing the hardware modules (DyIO, HydroPWNics modules) and sending plant data to the cloud (http://www.vivaplanet.io/).

## Problem

HydroPWNics was started with the idea of creating an autonomous hydroponic garden. The purpose of such a garden would be for the autonomous and ultra-efficient production of healthy food produce. Modern human beings have busy schedules, may lack the expertise for agriculture, or be in a location that makes it difficult. HydroPWNics aims to be a complete open source autonomous hydroponics garden using the nutrient film technique method to save water. Developing nations could possibly make use of such a system to maximize food production and save huge on fresh water, a scare resource in some nations. Thought not initially, HydroPWNics will evolve into a solar powered system, further extending its efficiency and making it more energy friendly. By virtue of being open source HydroPWNics and every subsystem it incorporates is available to anyone everywhere. Additionally the basic components of the system are designed from parts easily source-able at hardware stores aside from the electronics.

## Detailed Breakdown of The Whole System

## Electronics:

**Custom HydroPWNics Hardware** (Hardware designed and developed for the project)

The HydroPWNics Hardware is of a modular design consisting of a Module Hub as a master controller and Modules for function and sensing. The Module Hub interfaces with up to three modules, each module connect to a port on the module hub and communicates with it over UART. The Module Hub also contains an ESP8266 WiFi SoC for communication with cloud and Beaglebone Green. Currently the Modules consist of two models, the Analog Sensor Module and the Digitial Sensor Module. The Analog Sensor Module allows for interfacing with up to 9 analog sensors. The Digital Sensor Module can interface with up to 4 I2C based sensors. With this modular system additional modules can be created for additional functions and easily be swapped into a slot on the Module Hub. For now the focus will be on sensing for plants and environmental monitoring, but in the future modules for automating the other aspects of the garden will be created. The Goals is to have the whole system running off HydroPWNics hardware modules and the Beaglebone Green in the end.

**Sensor Modules**

Two Sensor modules two cover nearly all bases of plant sensing and monitoring. The analog module is designed to interface with 9 analog sensors via ADC on the ARM MCU. The digital sensor module is designed to interface with 4 I2C sensors, specifically combination temp/humidity sensors and other sensors that talk over I2C. Each module communicates with the module hub via a galvanically isolated UART serial port. In the future there might be different variants of the digital board, I'm thinking either a SPI or 1 Wire version, both or a mixed one, necessity will dictate which one gets made.

**Analog Sensor Module:**

Specs:

- MKL05Z16VFK4 ARM Cortex M0+ 48MHZ MCU

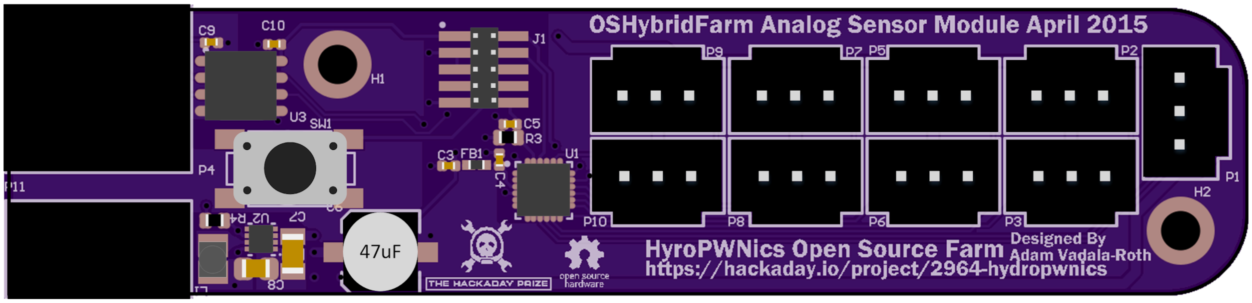
- 9 Analog Sensor Inputs with 3V3 DC Power

- On Board DCDC 3V3 regulator for 1A 3V3 DC Power

- ARM Cortex M JTAG Debug Connector

- Galvanic isolation

- M3 mounting holes

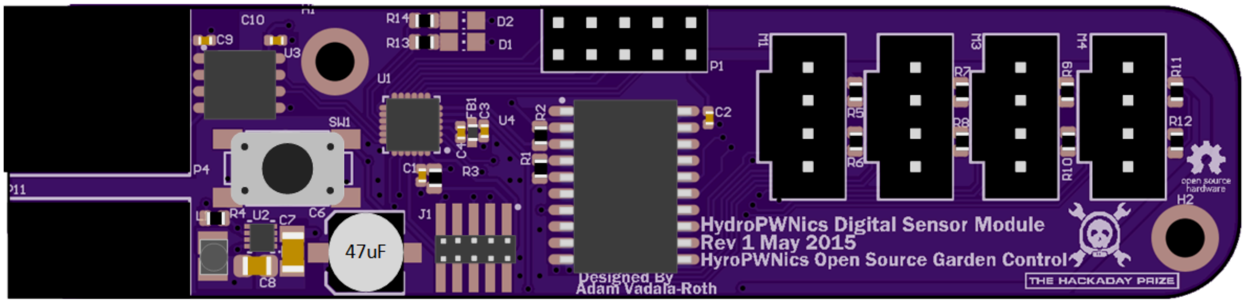


Hardware source for Analog Sensor Module:

https://github.com/HydroPWNics/AnalogSensorModule-Hardware

**Digital Sensor Module:**

- MKL05Z16VFK4 ARM Cortex M0+ 48MHZ MCU
- 4 I2C Sensor Inputs with 3V3 DC Power
- On Board DCDC 3V3 regulator for 1A 3V3 DC Power
- ARM Cortex M JTAG Debug Connector
- 8 GPIO pins, power, ground Expansion connector (SPI signals included, some ADC too)
- Galvanic isolation
- M3 mounting holes



Hardware Source for Digital Sensor Module:

https://github.com/HydroPWNics/DigitalSensorModule-Hardware

**HydroPWNics Module Hub**

The module hub is designed to interface with the sensor modules as well as any other module fitting the same formfactor. The hub can interface with 3 modules at once. Onboard the hub is another FreeScale ARM Cortex M0+ this time an MKV10 boasting a 75MHZ clockspeed and two UART peripherals. This chip was used because it has a high clock and two UARTs, one UART is used with an analog multiplexer to talk to the modules the other is used to for a Bowler serial port, allowing coms with other bowler devices at 5V. In additoin to interfacing with all the sensor modules, the hub also features three epxnasion headers, onboard power for modules, and an ESP-12E ESP8266 WiFi module. The module hub will act as the hardware interface between all sensing modules and the software situated in the firmware of the modules/hub and the Java API. Module Hub hardware source: https://github.com/HydroPWNics/ModuleHub-Hardware

**Module Hub**

Module Hub Spec:

- MKV10Z16VFM7 - 32 Pin QFN 5X5

- 3 Module Ports with 12VDC Power

- ESP8266 ESP-12E WiFi Module

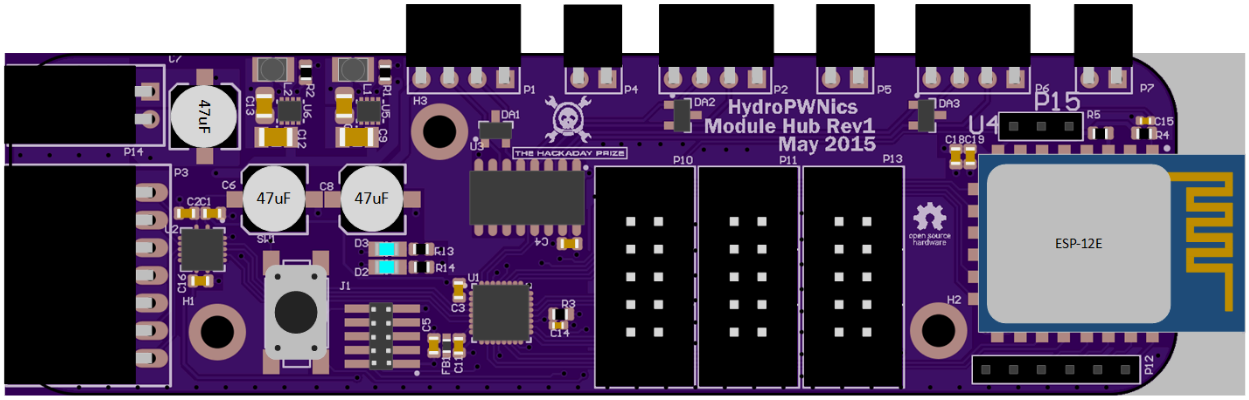
- On Board DCDC 3V3 regulator for 1A 3V3 DC Power

- On Board DCDC 5VDC regulator for 1A 5V DC power (level shifter/expansion)

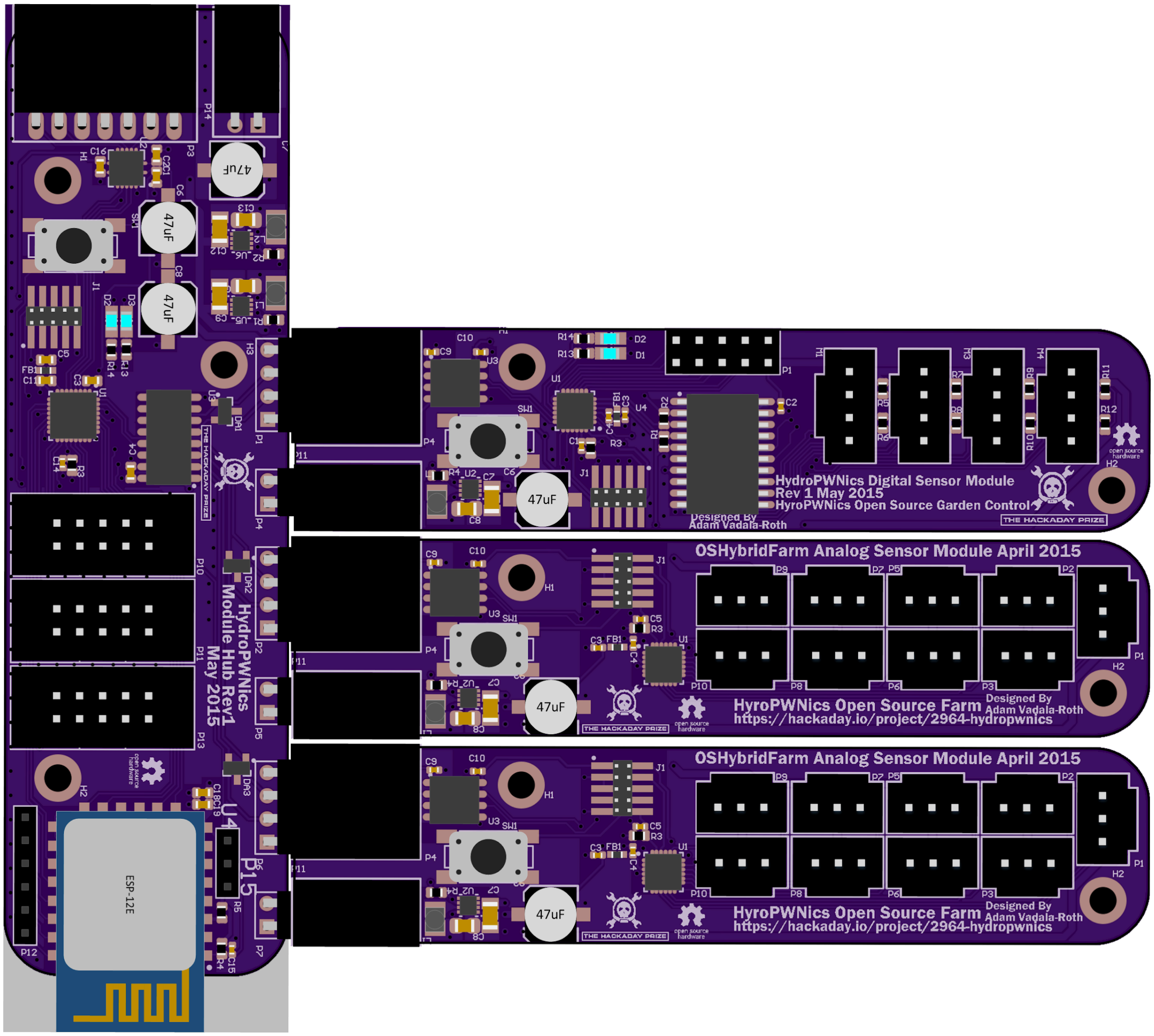
- ARM Cortex M JTAG Debug Connector

- 74HC4052 Analog Multiplexer Demultiplexer for UART module Interfacing

- 3 M3 Screw Mounting holesModule Hub:

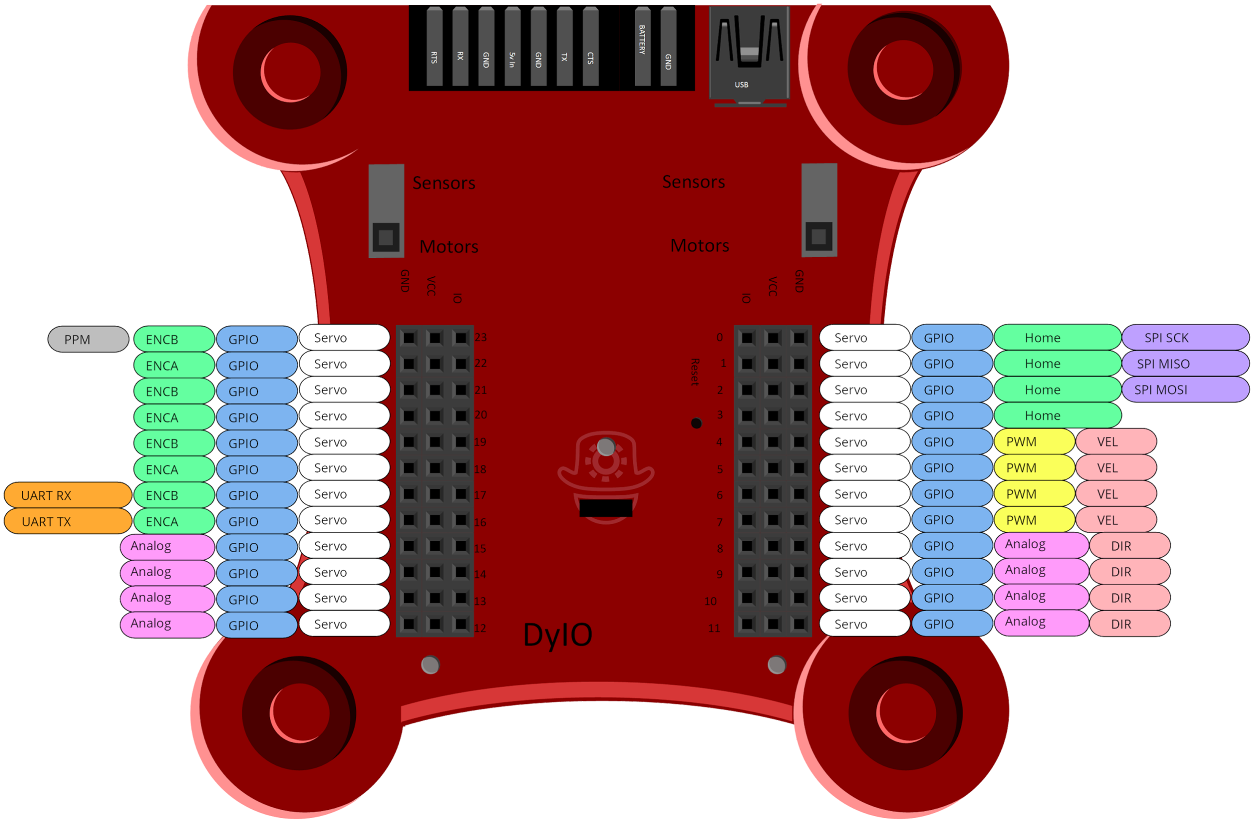


Module Hub with modules hooked in place.


HydroPWNics
Module Hub Rev1
May 2015
ESP-12E
HydroPWNics Digital Sensor Module
Rev 1 May 2015
HyroPWNics Open Source Garden Control
Designed By
Adam Vadala-Roth
OSHybridFarm Analog Sensor Module April 2015
HyroPWNics Open Source Farm
https://hackaday.io/project/2964-hydropwnics
Designed By
Adam Vadala-Roth
OSHybridFarm Analog Sensor Module April 2015
HyroPWNics Open Source Farm
https://hackaday.io/project/2964-hydropwnics
Designed By
Adam Vadala-Roth
THE HACKADAY PRIZE

### DyIO Module

The DyIO shorthand for Dynamic Input Output, is an open source electronics/software/control/robotics platform based around the DyIO controller module and an advanced software API that runs on the PC in Java with other languages on the way. You can learn more about it on the Hackaday Projects page for it here: https://hackaday.io/project/3185-dyio-dynamic-input-output-controller The DyIO module will be used primarily for driving high current AC solid state relays controlling all AC current devices within the system.



# Sensors

### Environmental Monitoring

For environmental monitoring of the garden temperature, humidity, and light will be sensed/monitored. For sensing humidity and temperature the Adafruit HTU21D-F sensor will be used. The HTU21D-F interfaces over I2C and provides temperature and humidity data. Form optimal sensing the garden will be divided into 4 quadrants with a dedicated HTU21D-F in each quadrant. These HTU21D-F sensors will interface with the Module Hub via the Digital Sensor Module on one of the three Module Hub ports. For light sensing the Adafruit TSL2561 sensor will be used, it also communicates over I2C and will be interfaced with Module Hub in the same way. Again for optimal sensing the light sensors will be assigned to quadrants in the same way the temp/humidity sensors are.

### pH Sensing

For pH sensing an aquarium pH probe and processing board will be used. For processing the Atlas Scientific pH board will be used, until I have time to design my own solution. The sensor will interface with with the module hub over UART.

## Electrical Subsystems

### Control Box

The control box contains a Beaglebone Green, USB wifi adapter, USB hub, and powersupply. The Beaglebone Green runs the control software for the system, a java program written in the Neuron Robotics SDK. The USB hub is for connecting the wifi adapter and multiple external USB devices, such as DyIO modules and FTDI serial converters. The power supply will be an of the shelf 12V switching PSU, typically found on ebay and alibabba, and some pololu DCDC converter boards for lower voltages, the power supply is designed to power the BBB and USB hub, eventually this system will be designed to run off a solar power battery system.

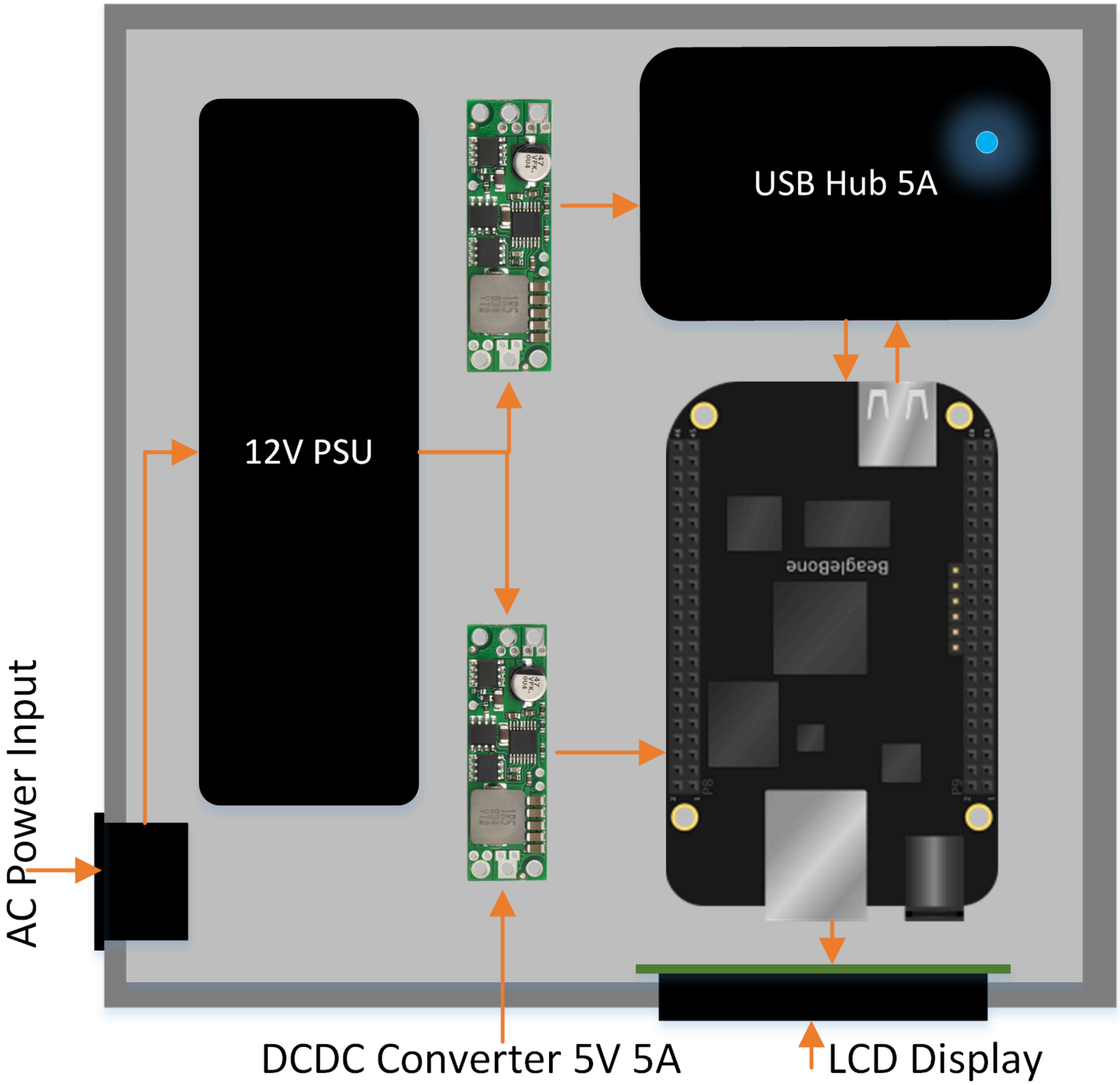
Specs:

- Beaglebone Green

- Two Pololu 5A 3v3/5v DCDC Converters

- 12V PSU

- 5A USB Hub



**AC Power, Pump, and Water Control Box**

The purpose of the AC Power box is to control AC power electronics within the system currently this includes the water pump and the grow lights. This box/subsystem is simple in its design, it consists of a DyIO module, solidstate relays, current sensors, and various connectors.

Specs:

- DyIO Module for control

- Solid state relay for each AC socket

- Current sensing on each AC socket

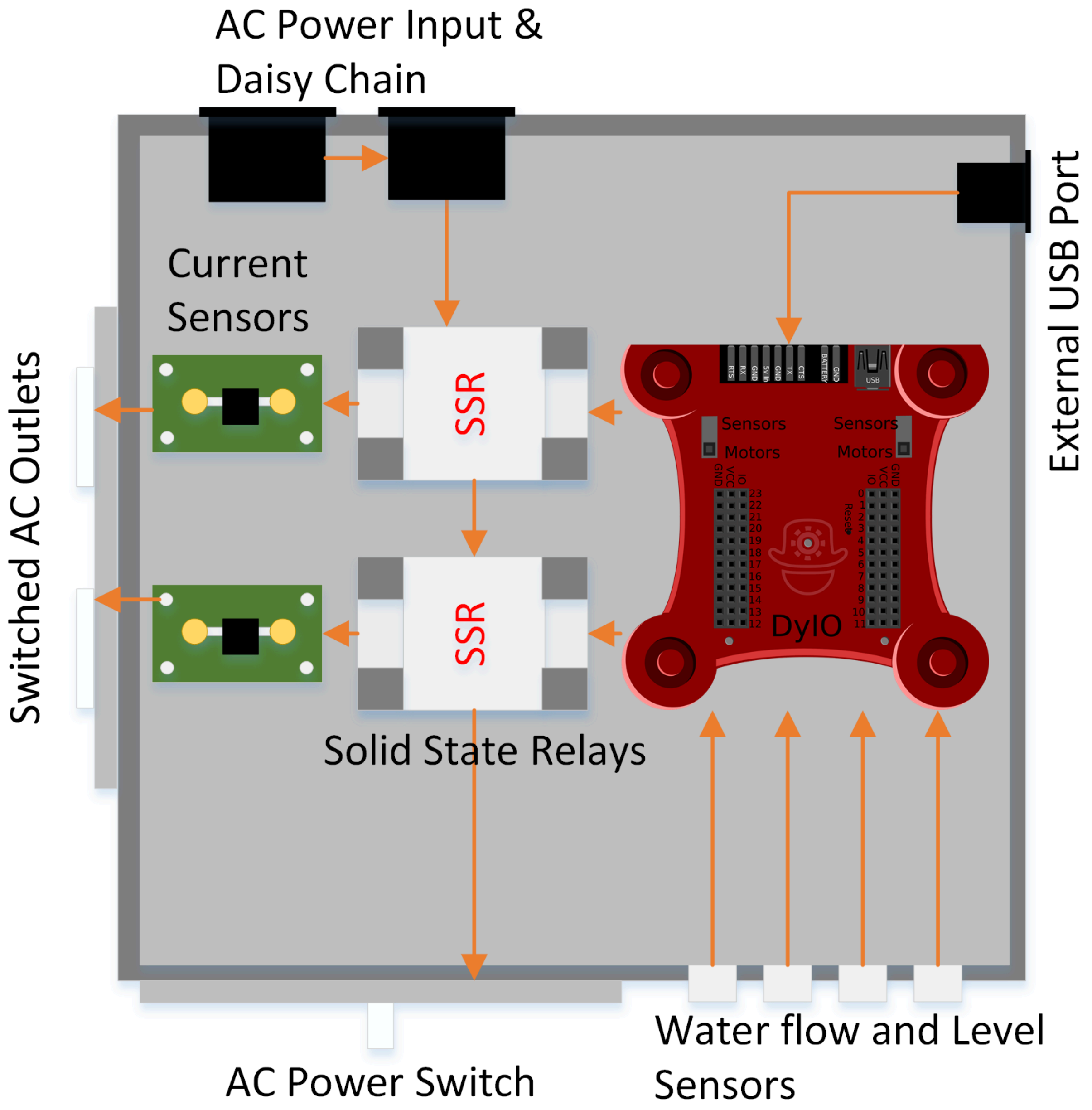
- Two water flow rate sensor inputs via 5 Pin Aviation Connectors

- Two water level sensor inputs via 4 Pin Aviation Connectors

- External USB port w/ optional cap.

- AC power input connector with fuse/switch

- AC Power daisy chain connector



**Custom LED Grow Light**

HydroPWNics will be using custom made LED grow lights. LEDs are low power and energy efficient and run of DC voltage making conversion to solar power later easier. The lights are a simple design made of birch plywood, the LED strip is mounted inside and the PSU will be bolted on the end. The lights consist of two kinds of strip, a red, red, blue, LED grow strip and a mixed LED purple LED strip. The lights mostly consists of the red, red, blue variety, with only a few being of the mixed purple variety.

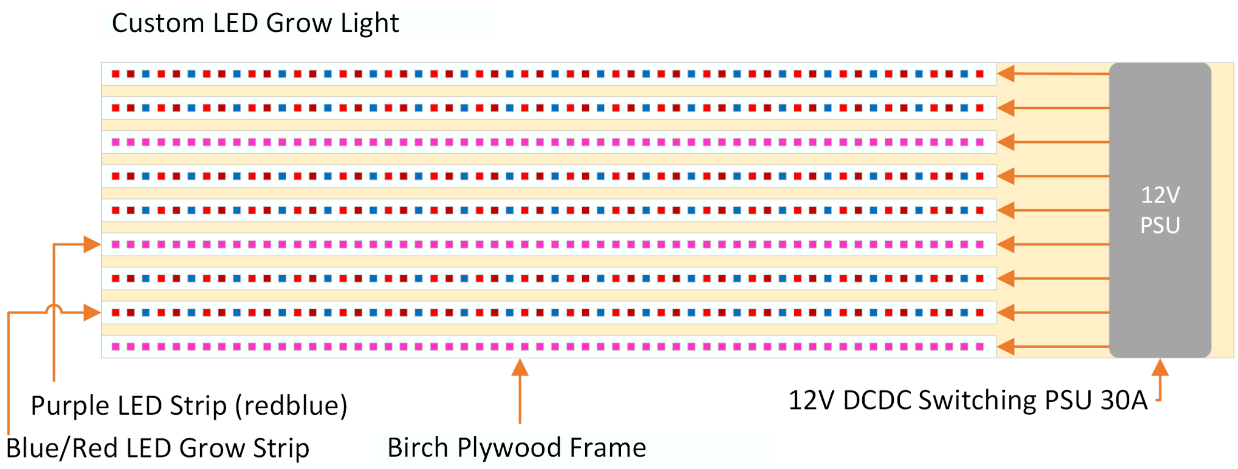
Specs

- DCDC 12V 30A LED Strip PSU (common PSU folks use)

- 5050 Red/Red/Blue LED Grow Strip 16FT (2x)

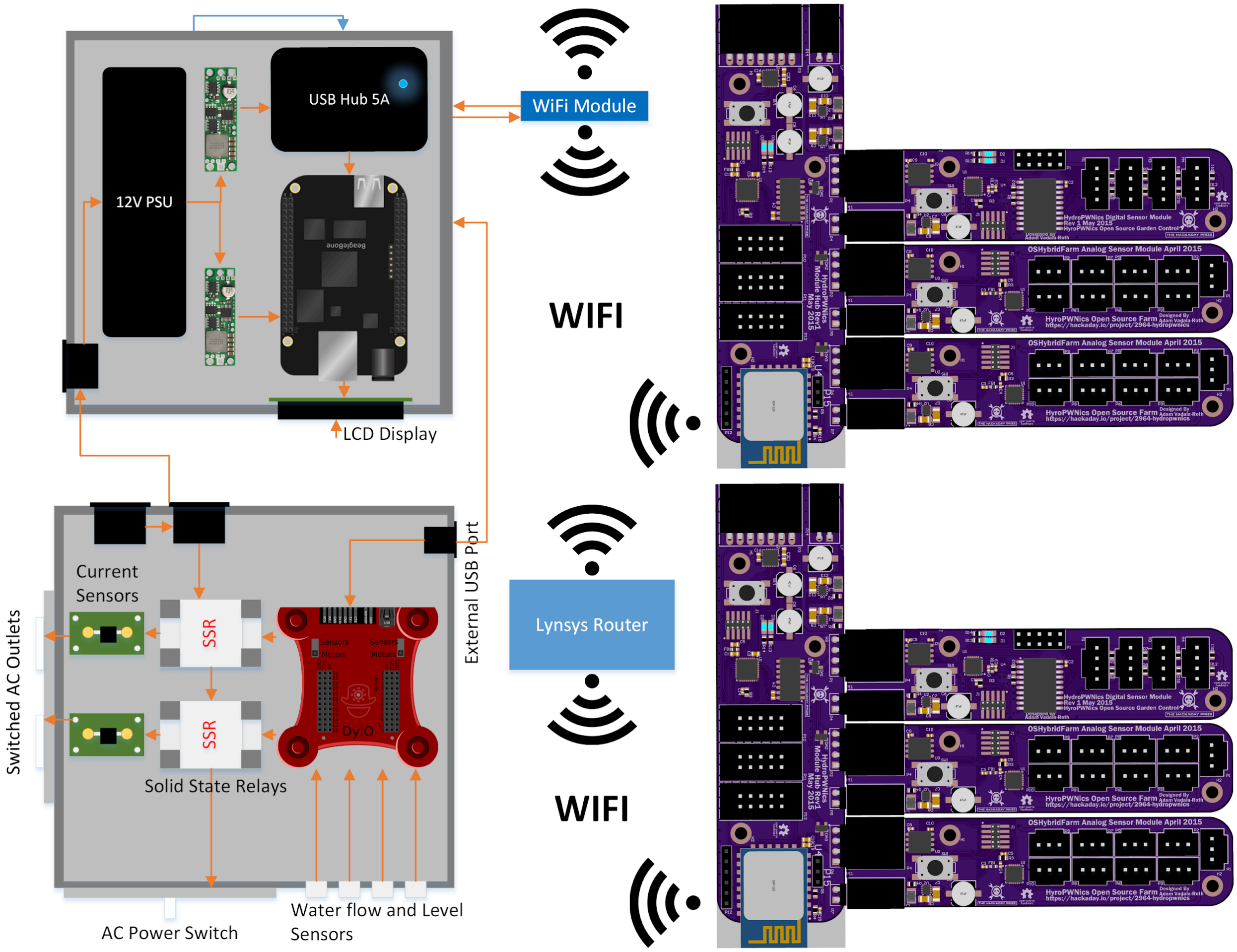
- 5050 Purple Mixed LED Strip 16FT

- Plywood frame



## Overview of Hardware Systems

Now that all the subsystems have been described in detail a breakdown of how they all interact with in the system is in order. Follow along with the diagram below for a stronger understanding of what's to follow. The diagram is based around the control box so I'll start there. The control box interfaces with the DylOs and serial converters via USB ports on the USB hub, the Module hubs interface with the control box over WiFi. The WiFi networking is provided by a dedicated WiFi router running off the main house network via ethernet. Once all the hardware modules are built and running the system should be relatively easy to setup the goal is to really make it as simple as the diagram.



# NFT Hydroponic Garden Grow System

As part of the HydroPWNics project a simple cost effective NFT hydroponic grow unit was designed and developed as a means to test the HydroPWNics garden control system.

Learn how you can build your own Here:

https://hackaday.io/project/7116-low-cost-nft-hydroponic-garden-under-200

Specs:

- 4 Rows made from 5X5" vinyl fence post

- 4 Plants per row in 3" netted pots, total of 16 plants in the unit.

- Water level of each row adjustable with valve

- 16 Gallon reservoir

- 158GPH Submersible Pump

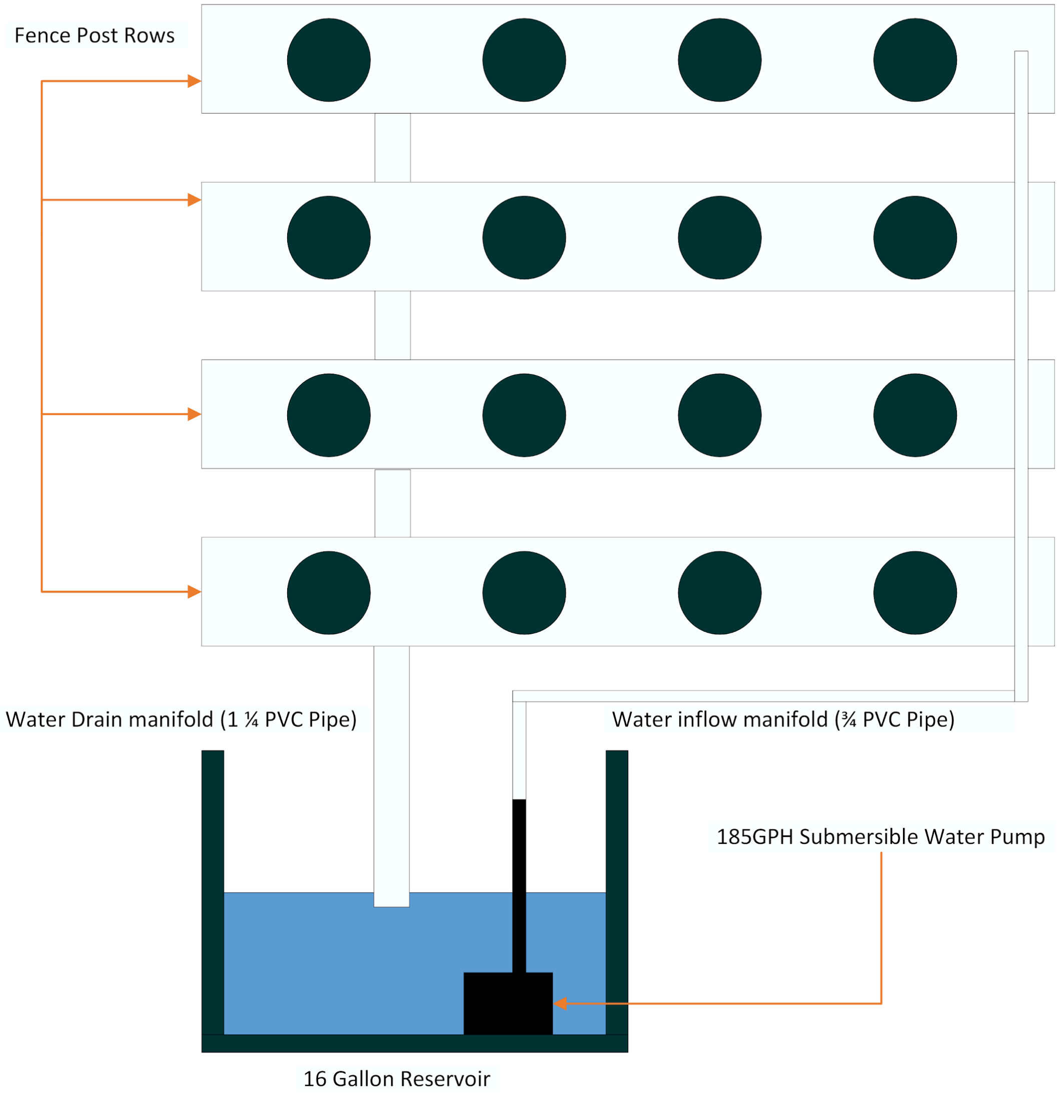
- Cheap wooden frame

- Water aeration in reservoir

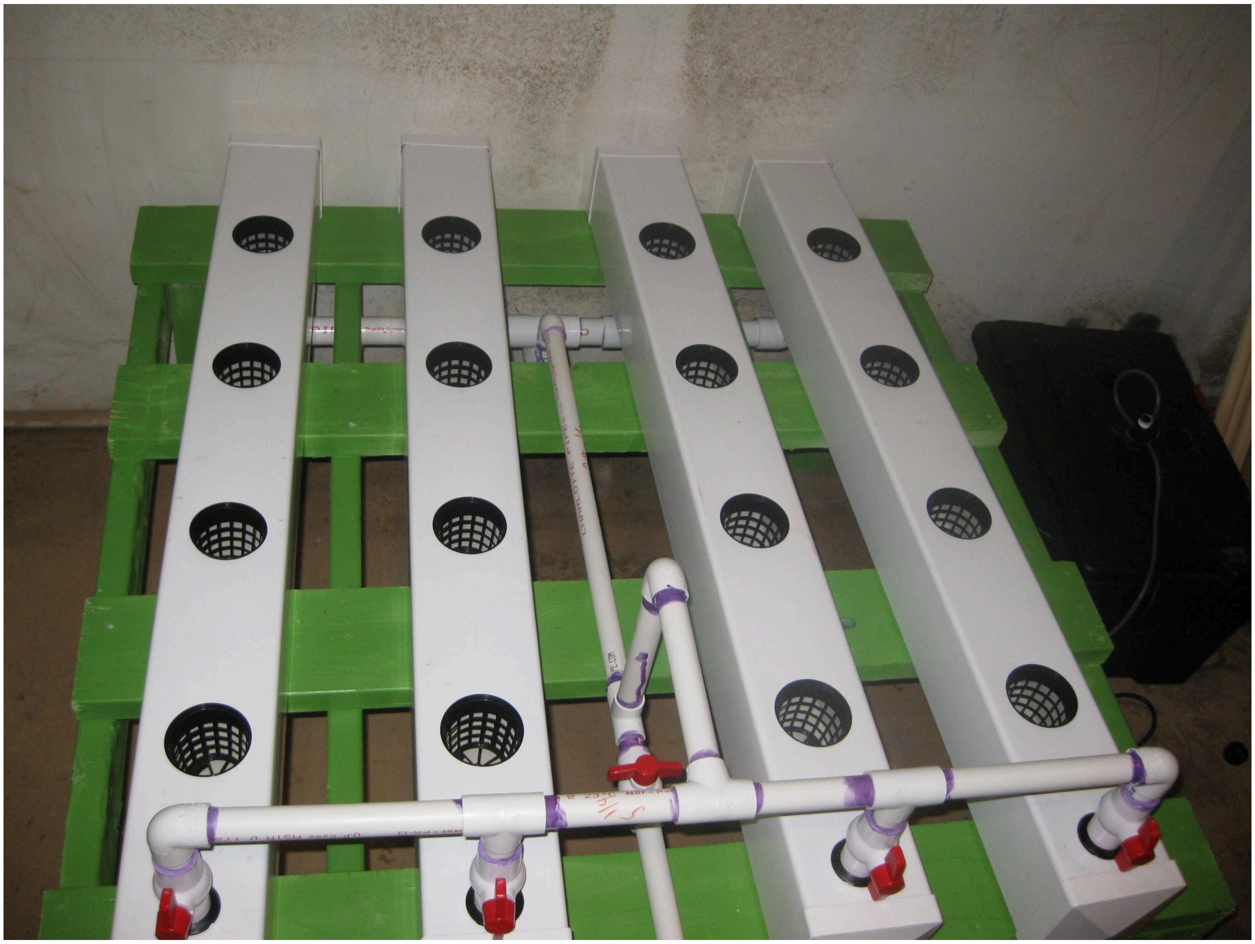
- Support up to 16 plants with capacity for large root systems

How it Works:

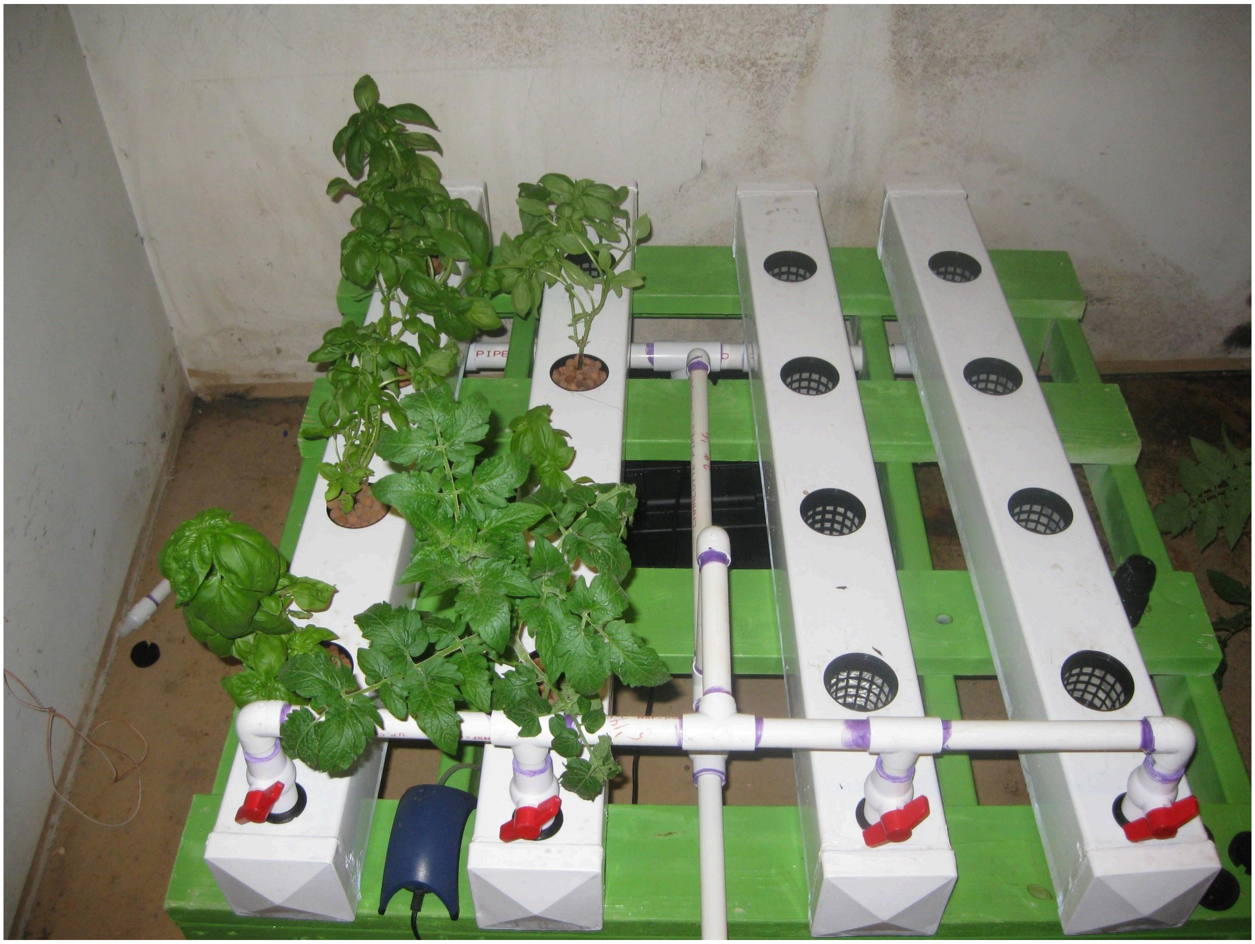
# 16 Site NFT Hydroponic Grow System


Fence Post Rows
Water Drain manifold (1 ¼ PVC Pipe)
Water inflow manifold (¾ PVC Pipe)
185GPH Submersible Water Pump
16 Gallon Reservoir

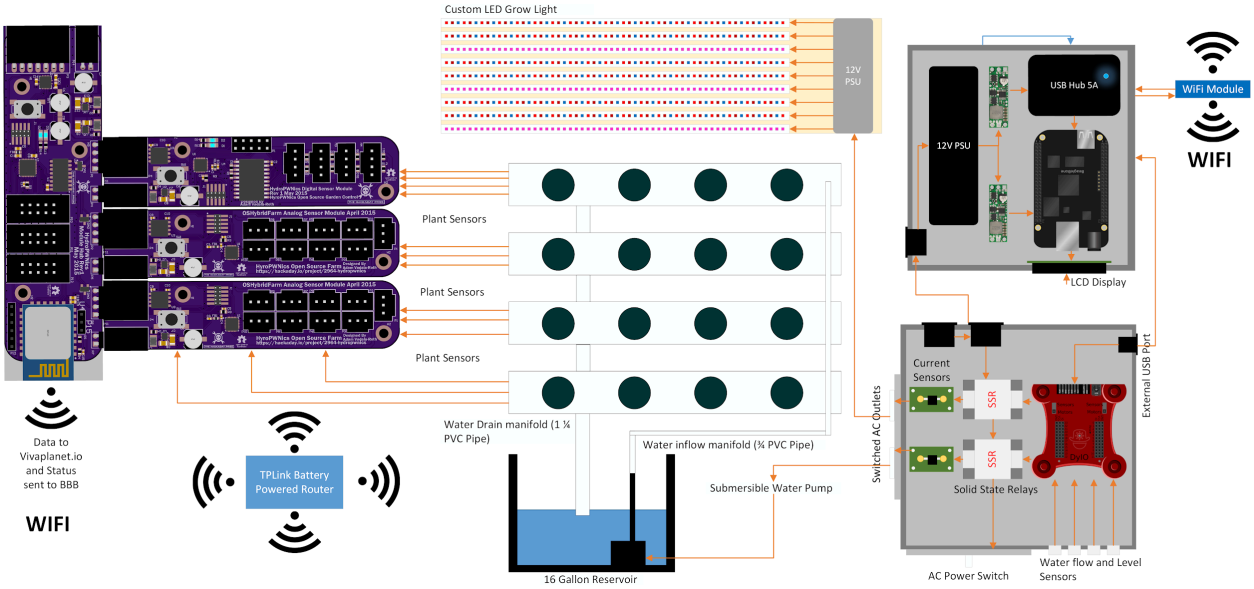
Completed Prototype:



Running and Plants transplanted (9/18/15)



# Complete Integration:

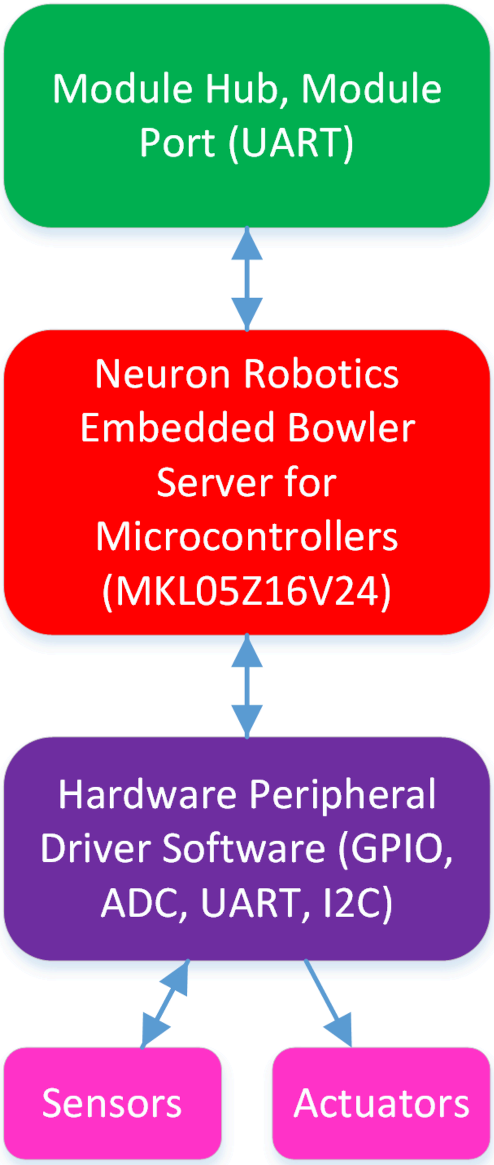

Custom LED Grow Light
12V PSU
USB Hub 5A
12V PSU
WiFi Module
WIFI
LCD Display
HydroPWNics Digital Sensor Module Rev 1 May 2015
HyroPWNics Open Source Garden Control
Designed By Adam Vadala-Roth
OSHybridFarm Analog Sensor Module April 2015
HyroPWNics Open Source Farm
https://hackaday.io/project/2964-hydropwnics
HydroPWNics Module Hub Rev1 May 2015
Plant Sensors
Plant Sensors
Plant Sensors
Data to Vivaplanet.io and Status sent to BBB
WIFI
TPLink Battery Powered Router
Water Drain manifold (1 ¼ PVC Pipe)
Water inflow manifold (¾ PVC Pipe)
Submersible Water Pump
16 Gallon Reservoir
Switched AC Outlets
Current Sensors
SSR
SSR
Solid State Relays
DyIO
External USB Port
AC Power Switch
Water flow and Level Sensors

# Software Systems

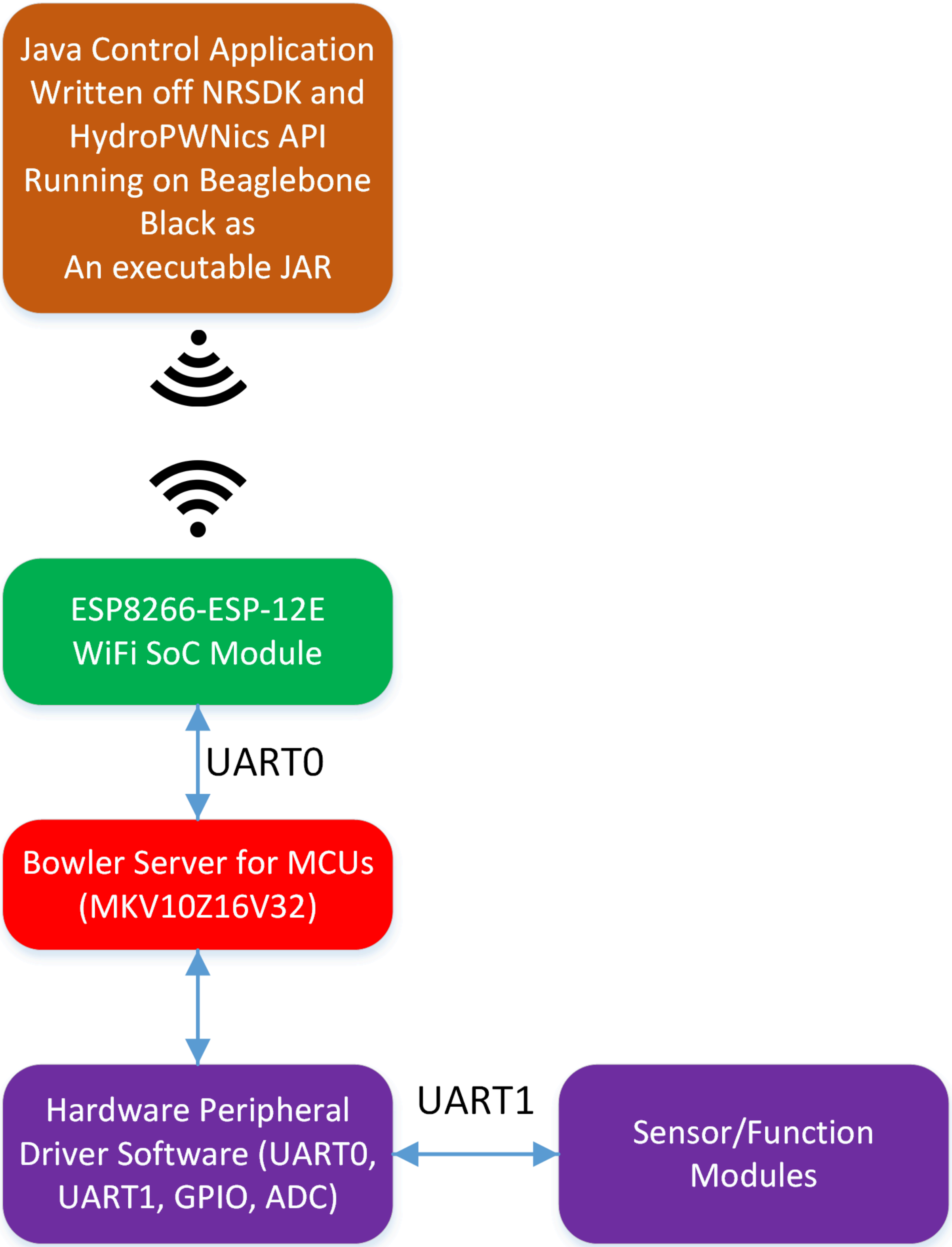
### Embedded Software

The embedded software pertains to the software running on the ARM Cortex M0+ MCUs on board the Modules and Module Hub. The software running on both the Modules and Module hub will be a custom port of the Neuron Robotics C-Bowler server (http://neuronrobotics.com/Bowler-Protocol/Overview/). This firmware will enable all the micro controller functions to be controlled via high level api (Java in this case) powered by the Bowler Communication system. This software setup was chosen because its an already working established system that is fairly easy to port over to any micro controller. The modules will not be directed the same way the hub will be by the java api, they will instead take simple commands from the module hub itself. Each module will identify itself to the hub as to what it is and what commands it can take, this is intended to keep things simple and abstract away complexity. The development plan for the emebbded software is to build working libraries for all functions of the two FreeScale CM0+s MCUs, and the ESP. The next step is to begin porting the C-Bowler server using the previous libraries. Next follows the never ending pursuit of developing the firmware and continually improving it! For development I will be using a student J-link and the FreeScale Freedom Board for the MKL05. Dev environment is probably going to be gcc on linux cli.

Sensor/Function Module Software:


Module Hub, Module Port (UART)
Neuron Robotics Embedded Bowler Server for Microcontrollers (MKL05Z16V24)
Hardware Peripheral Driver Software (GPIO, ADC, UART, I2C)
Sensors
Actuators

Module Hub Software:

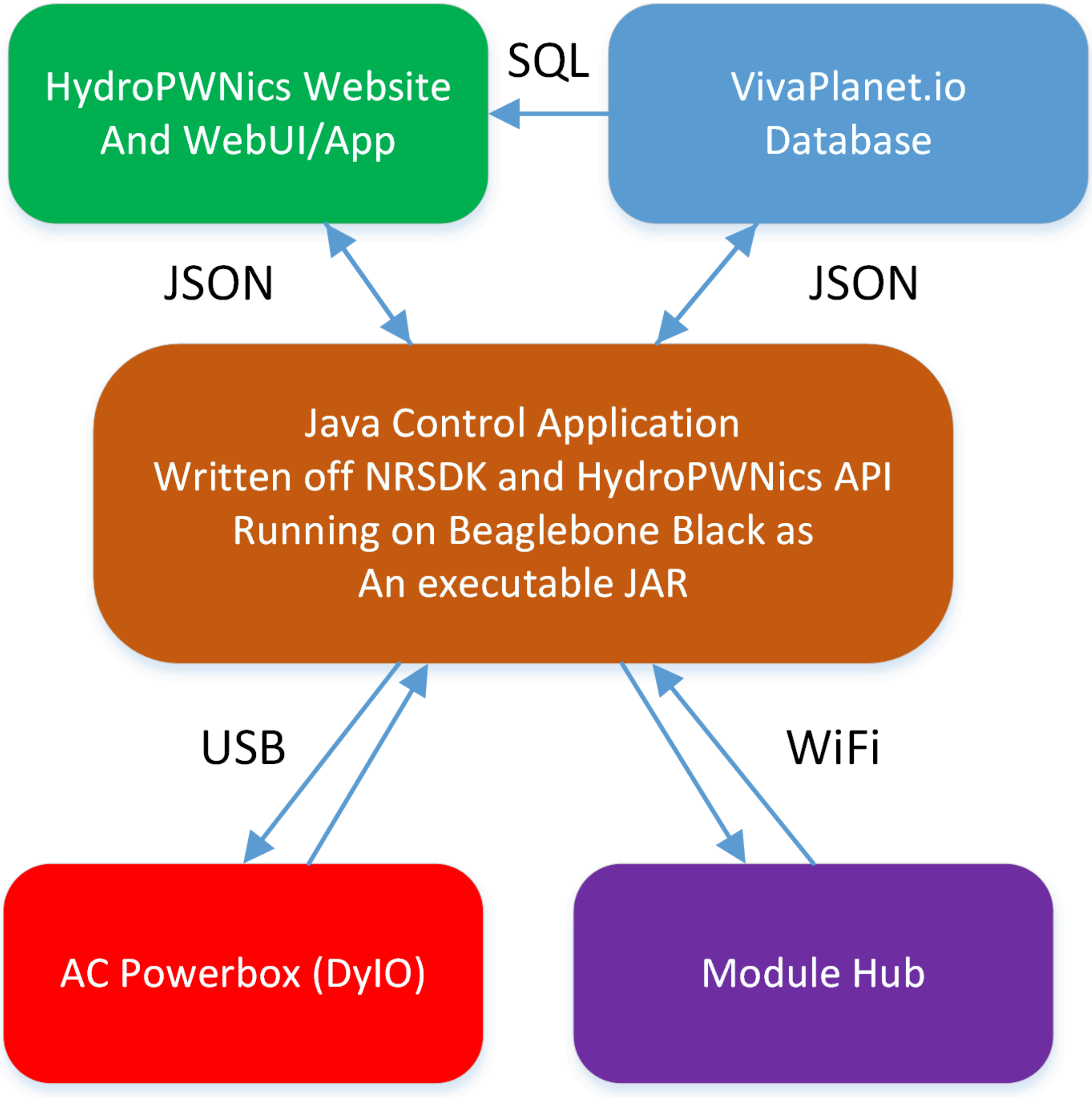

Java Control Application Written off NRSDK and HydroPWNics API Running on Beaglebone Black as An executable JAR
ESP8266-ESP-12E WiFi SoC Module
UART0
Bowler Server for MCUs (MKV10Z16V32)
Hardware Peripheral Driver Software (UART0, UART1, GPIO, ADC)
UART1
Sensor/Function Modules

**High level Software**

The high level software pertains to the software application running on the Beaglebone Green. The firmware described above running on the hardware only functions as a driver for the functions of the boards to be controlled from the high level program, the high level program is where all the application code will be written. In this case the application will be written in Java. The application software will communicate with the Module Hubs, polling them for data on the system and the plants, then packaging it to be pushed to the cloud. In addition to data collection and monitoring the application will be controlling ever aspect of the garden.

The development of the high level software begins with the development of an API, I call this HydroPWNIcs API. The API will be built off of the Neuron Robotics RDK (Robotics Development Kit). The RDK comes packages with BowlerStudio (https://hackaday.io/project/6423-bowlerstudio-a-robotics-development-platform ) This API is meant for controlling the HydroPWNics hardware in the same way that the Neuron Robotics RDK does with the DyIO as well as cloud communication adn JSON parsing. For remote control and remote monitoring of the garden the application will send data via JSON packet and also respond to JSON packets for making remote adjustments.

For a database for plant and garden data HydroPWNics project is a collaborating with the Vivaplanet.io project. vivaplanet.io is a database system for plants developed for the Hummingbird project (alson on Hackaday.io) . To learn more about Vivaplanet check out their project here: https://hackaday.io/project/6032-hummingbird . The system (app on BeagleBone Green) will be able to send data via JSON to the cloud in several ways. The HydroPWNics hardware is modular, so a given implementation on a garden is going to differ based on the style of hydroponic garden, its grow environment, control requirements, sensor requirements, etc. With the module nature of the hardware the software will be accommodating. One option for transmitting data to VivaPlanet.io will be the Module Hub first sending the data to the application program running opn the beaglebone green, then packaged into JSON and pushed to the cloud. This method is being chosen since its the simples to implement using the existing softwarelibraries I am using. The other option for pushing data to vivaplanet.io will be allowing the module hub to package and push JSON to the cloud directly. This implementation will require modifications to the Neuron Robotics RDK as well as the Bowler-c firmware, this feature will be implemented towards the end.



**Web Software**

The web software component of the project is the web page to which all the data will be published in the form of graphs and spreadsheets. The website will be written in HTML5 and Javascript, I will most likely grab some premade template and add javascript for securely connecting to the garden over the web. The website's primary function will be for remote viewing of garden status and remote monitoring data stats. If there is enough time the website will also feature a login page to reveal a dashboard to control/configure the garden over the web. The basic design of the website will be a simple HTML5/JS website from a template, a page for each kind of graph, graphs implemented using JSFreeChart and smoothiecharts, and the control dashboard being implemented in JS. The website will send/receive JSON packets to and from the garden for status updates and remote control/config functionality. As for the data in the charts, the web page will connect to Vivaplanet.io’s database using the user’s Vivaplanet.io login information and load the data from vivaplanet.io’s database into the charts.

## Nutrients, Plants, Ecology, etc

For the hydroponics component of the project I am using General Hydroponics grow nutrients. They seem to be a popular brand among hydro enthusiasts. I ran a small shallow culture last summer with these nutrients and I was impressed at how fast the plants grew, I'm hoping to get the same experience but better now that I am building a proper unit. These nutrients come in very high concentrations so the 3 bottle set will last you quite a while.

General Hydroponics Nutrient Kit:

Included the kit are three different bottles of nutrient solution. The three kinds are FloraGro 2-1-6, FloraMicro 5-0-1, and FloraBloom 0-5-4. FloraGro is for structural and foliar growth, the numbers 2-1-6 refer to the composition of the solution in percents, 2 percent total nitrogen, 1 percent phosphate, 6 percent soluable potash. FloraGro can be used on its own or combined with either of the other two (or both) for nutrient solutions tailored to a specific style of plant. Typically when I do cuttings (cloning) or start seeds I use a solution made of just water and FloraGro 2-1-6, for example. (Learn more about the FloraGro here: http://generalhydroponics.com/site/gh/docs/prod_labels/floragro.pdf ) FloraMicro is for providing plants with nitrogen, potassium, and calcium, its usually combined with FloraGro and given to plants well past seedling phase. The 5-0-1 refers the composition of 5 percent total nitrogen, the 0 refers to micro nutrients (see bottle label http://generalhydroponics.com/site/gh/docs/prod_labels/floramicro.pdf) , and 1 percent total soluable potash. FloraBloom is for the later stages of the plants life, it provides the necessary nutrients for reproductive growth e.g flowering, fruits, and seed production. FloraBloom is added after the plants have advanced vegetative growth to begin the fruit developing process (Yay food!!!) . The 0-5-4 pertain to the composition of 5 percent available phosphate, 4 percent soluable potash, 1.5 percent magnesium, and 1 percent sulfur. (Learn more about the FLoraBloom http://generalhydroponics.com/site/gh/docs/prod_labels/florabloom.pdf ) In my experiment last year I mixed the solution for my plants based on the ratios recommended on the bottles of the nutrients, I will continue with that strategy to start but I would like to experiment with different combintations. Also long after the GH kit is used up (or sooner) I plan to evaluate Fox Farm branded nutrients as well as adding homemade compost based nutrient solution.

Special Thanks to General Hydroponics for having such informative labels for thier products, a lot of the information was derived from their labels and I feel like I should credit them here, all information on the nutrients was pulled from their PDF labels:

http://generalhydroponics.com/site/gh/docs/prod_labels/floragro.pdf

http://generalhydroponics.com/site/gh/docs/prod_labels/floramicro.pdf

http://generalhydroponics.com/site/gh/docs/prod_labels/florabloom.pdf

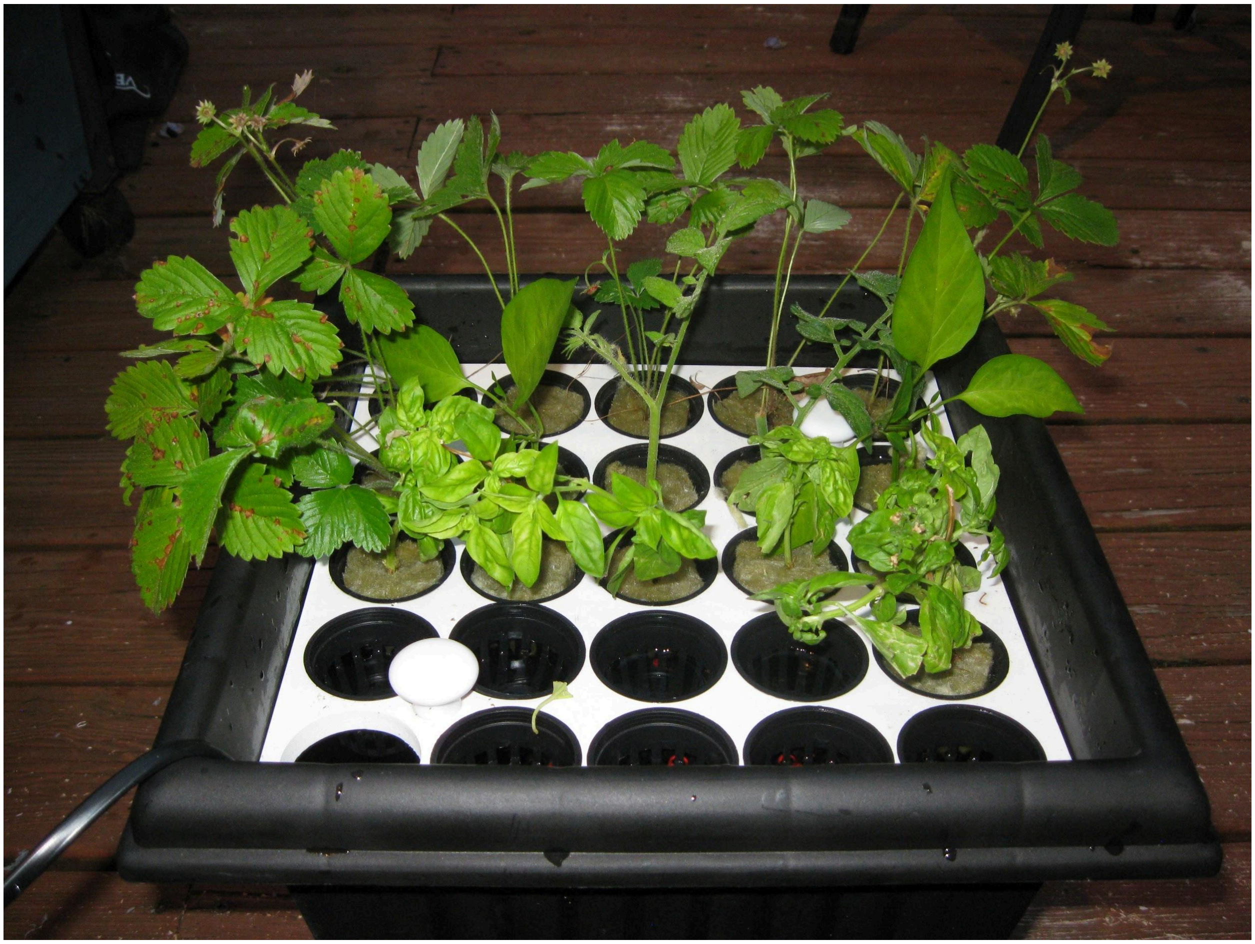
General Hydroponics Site:

http://generalhydroponics.com/site/index.php

**Plants and Creating Plants**

An important part of every garden no matter what kind are the plants that grow in the garden. There are a few ways to create plants for a hydroponic garden. You can start from seeds with special grow solution (time consuming), you can even order plants ready to go online, you can take cuttings of plants and clone/root them, and you can just take plants out of the ground clean off the roots and convert them over. I did a combination of pulling up plants and cuttings. To start the cuttings and convert the rooted plants over I got a small water culture rig. The setup is simple, just a pot with a water agitation pump and a simple foam core cutout to hold the pots. To keep the plants in the pots and have something for them to root into I used rockwool cubes. For plants I got a bunch of strawberries in a raised box in my backyard I just pulled the offshoot plants out of the dirty and rinsed them before stuffing them into rockwool cubes. I had some basil cuttings rooting on a window sill with just plain water in a glass I just took the rooted plants and put them in rockwool cubes and added them to the rig. The other plants are tomatoes and jalapeno peppers which are just cuttings stuck into rockwool, I'm hoping to root these.

In order to get these plants to root and have a lot o root growth I mixed up a solution for the water culture rig specifically for root grown and root boosting. Hopefully in a week or two these plants will be ready to load into the NFT Rig and be upgraded to bigger netted pots with terracotta stone growth media. For those interested, I am using General Hydroponics Rapid Start and Maxi Gro with this cloner, basically using the Maxi Gro as a booster for other sort of plant growth. More on the Rapid Start here: http://generalhydroponics.com/site/index.php/products/supplements/rapidstart/ More on Maxi Gro here: http://generalhydroponics.com/site/index.php/products/nutrients/maxi_series/maxigro/



Just before Transplant (9/18/15)