Embedded documentation

Based on 3 parts, programming, electronic and mechanical parts to be controlled.

**Tools** used for this part:

* ESP 8266 (controller)
* ATmega328 (Slave controller)
* Ultrasonic
* DHT sensor
* Temperature sensor
* PH sensor
* EC sensor
* LCD 20x4
* I2C module
* Push buttons
* LED strip
* Mini pump (Solutions)
* Medium sized pump (Water)
* Oxygen supplier
* Real time clock

Currently that is everything to work with and to be controlled in this part.

Software to be used for programming is “Arduino IDE”, where the AVR and ESP will be compatible with it. Next will be the description and how to use these components relating to the hydroponics project.

**Description**

It is wanted to control the flow of the project to meet the needs of plants, requiring nutrients in some specific times, refilling the water for mixing with nutrients, indicating the levels and the needed luminance for the plant. It is all related to the programming of the controller.

**ESP8266:** is main executioner for the project due to its high processing power, has many PWM pins which will be used for controlling the **LED strip luminance** according to the stage of the plant.

Will monitor the readings of the **Ultrasonic** to know the water level and decide whether it is needed to manually refill the container or not.

Taking readings from **DHT** (Humidity and temperature of the room) to ensure the environment of the plants is as required.  
Readings of **temperature** **sensor** for the solution (should be less the 35 degree).

Displaying the sensors value on an **LCD 20x4** to fit all the possible readings, it needs the I2C module, so we decrease the required pins from 6 to 2 pins.

Also, the **push buttons** to handle the LCD and user interface.

The ESP8266 has a Wi-Fi module, so it can reach the web application to upload the readings if needed and receive any critical command from the server.

**ATmega328:** The slave controller that does whatever the ESP asks or needs, it has more ADC pins which is good for the application as the ESP can provide only 1 ADC pin, connected to it the following devices:

**PH sensor**: which monitors the value of the PH in the solution and sends it periodically to the ESP.

**EC sensor**: for monitoring the Electrical conductivity of the solution, which is very critical to the application, also sends the value along with the PH.

**RTC**: the clock of the whole thing, that tells when to put each nutrient, when to fill the water or stop filling, even the time stamp of the uploads to the web application.

Then the control signals to the mechanical components when needed {Pumps and Oxygen supplier}.

**How it works**

* Choosing a certain plant on the web application to load a certain program with the plant values on the ESP.
* User puts the seeds in their place and press the start program.
* Water begins to fill the upper container until it reaches the grouping medium, where ultrasonic tells when to stop pumping.
* Oxygen supplier starts to work to provide the oxygen for seeds.
* Once everyday the LED strip goes on to take a picture of the plant to know if it is ready to the next step or not.
* If it is ready the server sends a command to start adding the nutrients.
* First pump pushes a certain amount of ‘A’ solution to the water.
* Waiting 10 minutes with RTC then starting to push the same amount but from ‘B’ solution.
* PH and EC sensors tells if it is the right values or not and the ESP takes the action, whether to add pure water (If PH is low) or add nutrients otherwise.
* The values would be monitored daily according to the time of RTC communicating with the web application to upload them.
* LED strip is controlled with RGB to give the plant the right luminance and wavelength, 12 to 14 hours per day actively, 12 to 10 sleeping.

**Steps taken.**

* Gathered all the modules needed “libraries & simulated”.
* Wrote the codes of each module individually.
* Integrated some of them in a single program.
* Ran the simulation on proteus for the possible modules and they worked.
* Tried the temperature sensor with hardwire.
* ESP works fine with the required applications; it has been tested.

**Next steps**

* Connecting the ESP with Wi-Fi and a web application for testing.
* Connecting ESP8266 with ATmega328 with UART.
* Receiving and sending data to the server and taking actions.
* Testing the modules “hardwire” individually.
* Integrating the program to test the application.