HYPLANTBOX!

Hydroponic system

ABSTRACT

HyPlantBox! project represents a user-friendly IoT-based hydroponic system controlled by an automation box. This system features a web-based user interface, enabling users to both monitor and adjust vital system parameters with ease.

Alissa Šubina, Andrei Lebedev

Contents

Introduction

The “HyPlantBox!” project is an IoT hydroponic system controlled by an automation box. The aim of this project was to build a minimalistic and efficient system, which can be easily adjusted by the user.

The control feature is initialized by CONTROLLINO PLC and Raspberry Pi, which are constantly exchanging the data. Besides the control, the user can also monitor the values read by the sensors and some other properties, such as text alerts, status of the pump and the ventilation capacity.

The “HyPlantBox!” was derived from the students’ innovation project and funded by UrbanFarmLab and Metropolia UAS for a paticular greenhouse installation, which is a mobile 2-storey shelf.

Getting Started

The system includes typical sensors used for water and air quality monitoring i.e pH, EC, temperature, and humidity meters. These in turn are responsible for triggering the activation of motor driven devices, such as pumps and fans. The significance of pumps lies not only in delivering water to the growing channels but also in dispensing the necessary nutrients automatically. The block diagram is demonstrated in Figure 1 below.

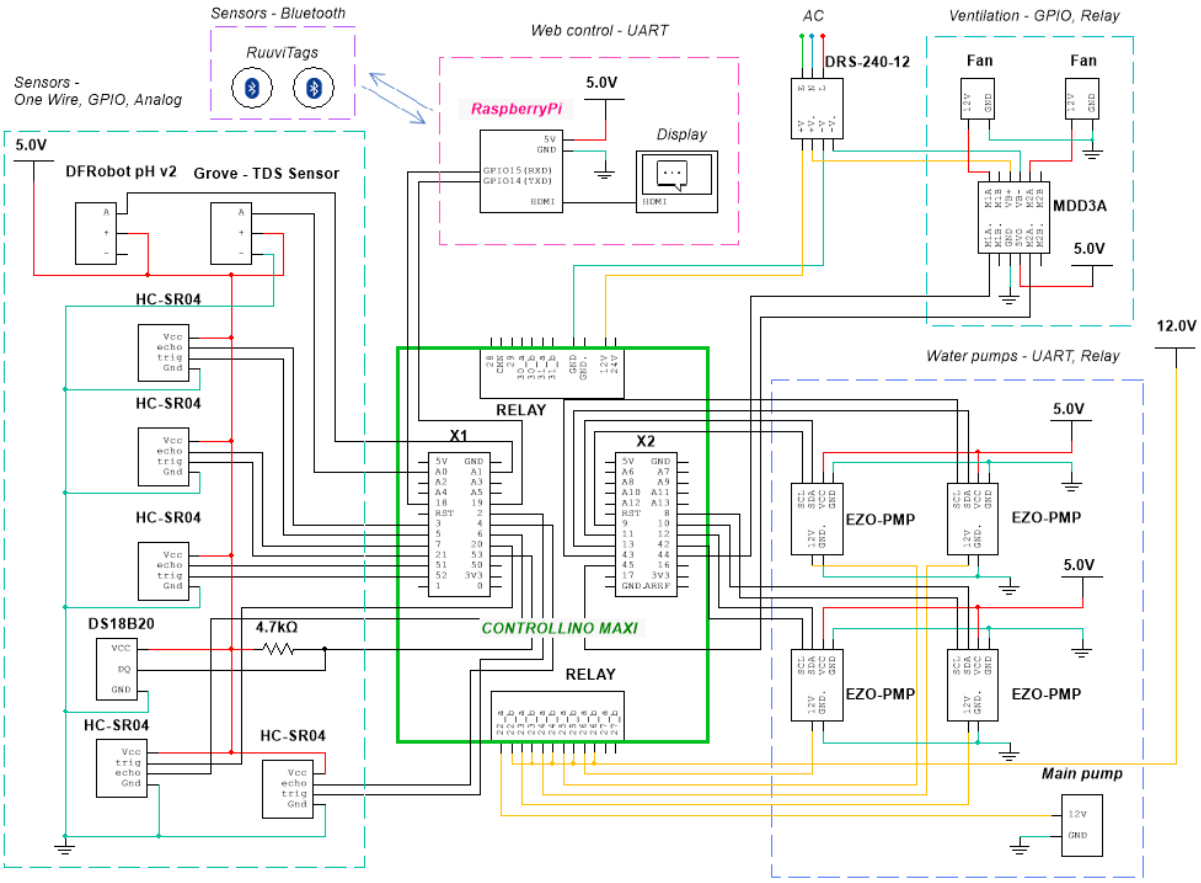


Figure 1. HyPlantBox! block diagram.

As shown on the diagram, all the sensors are located on the left and the mechanical equipment is on the right. In the center of the picture are the elements of the control panel. For a comprehensive pinout diagram specific to the CONTROLLINO MAXI, please refer to Appendix.

Note: CONTROLLINO provides power for sensors and devices ranging from 5V to 12V, while CONTROLLINO itself is supplied by 12V DRS-240-12 power supply.

As was mentioned earlier the system has certain dependencies. Due to DS18B20 waterproof temperature sensor, temperature compensation is applied to the DFRobot pH and Grove TDS meters, enhancing the accuracy of their readings. These readings, in turn, regulate the activation of the EZO-PMP dosing pumps. The specified threshold for pH and TDS triggers the pumps to either decrease or increase respective parameters in the main water tank by pumping up the appropriate solution (5 ml at a time by default).

Another noteworthy component is the MDD3A, serving as the motor driver within the system. This component enables precise control over fan speed by utilizing PWM signals. By changing the connection from M1A to M1B, the first fan can be made to rotate in the opposite direction. This same method can be applied to the second fan, providing versatile control over both fan units.

Interacting with the System

Once the box is powered and the Raspberry Pi operating system has successfully loaded, the user is greeted with the desktop interface. To launch the "HyPlantBox!" web application, the user needs to double-click on the "startup.sh" icon located on the desktop and choose the “Open in Terminal” option.

The key functionality of the web application consists of the following elements:

- Monitor - the main page used to monitor values measured by the sensors.

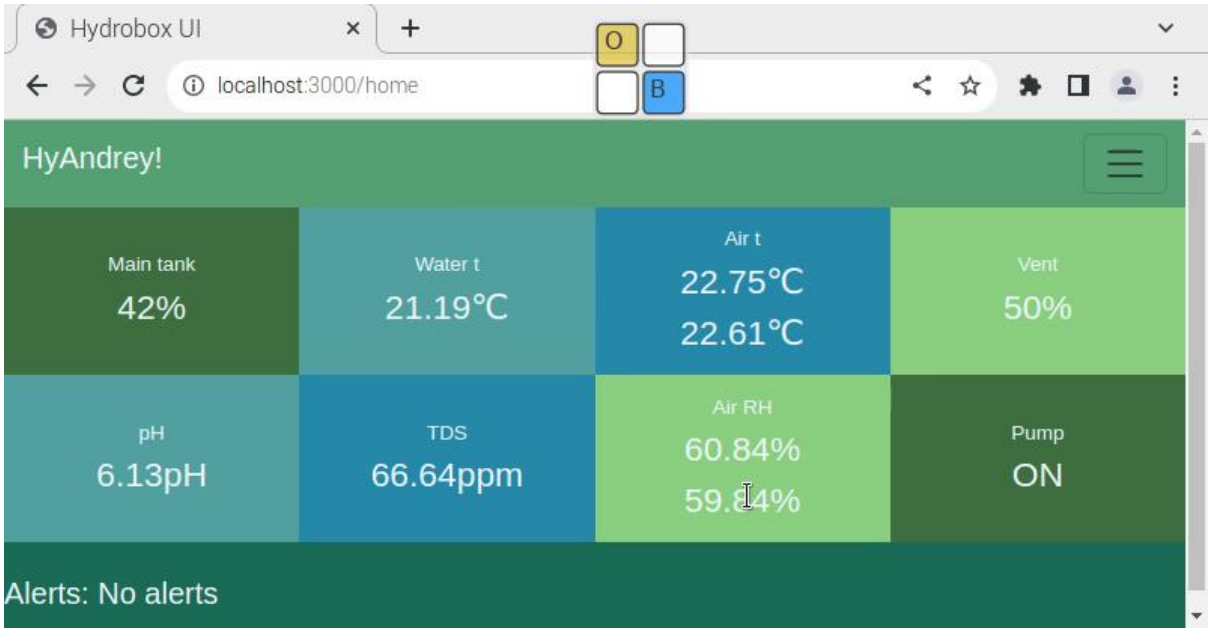


Figure 2. HyPlantBox! Monitor page.

In addition to the readings from the sensors, the monitor also displays information such as the status of the main pump and the fans’ performance. At the bottom of the page, the user will find “Alerts”, activated when the water level of any tank falls below 40% (thresholds can be changed in the source code).

- Controls – the page where the user sets up his/her preferences. Can be accessed by clicking on the burger button on the top right of the main page.

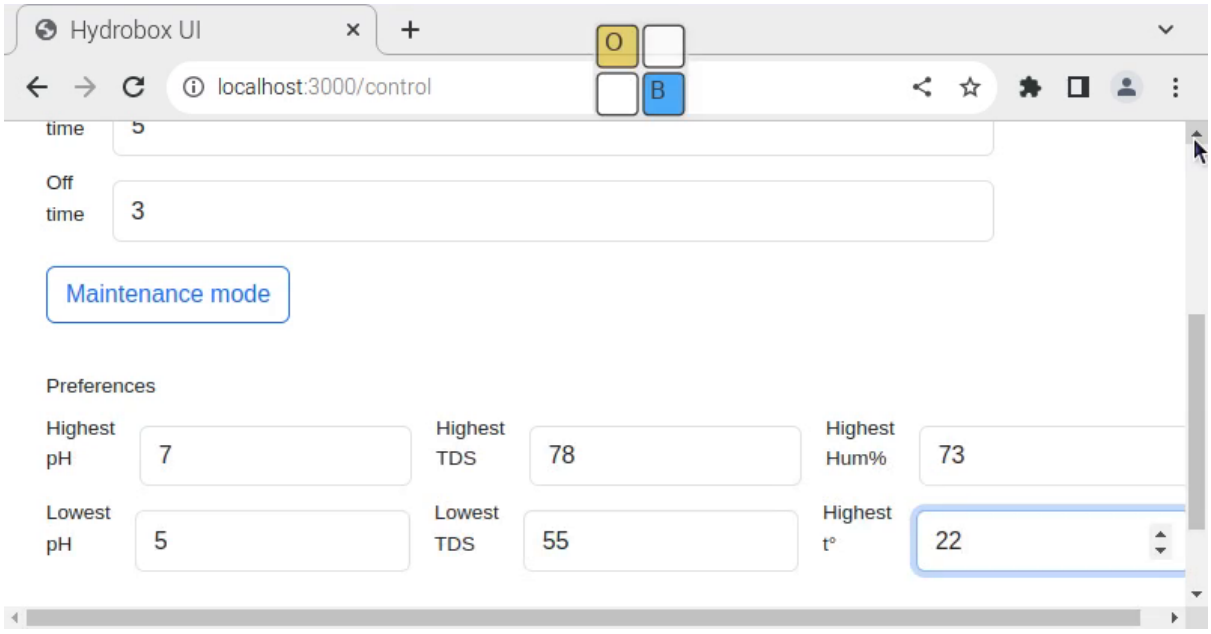


Figure 3. HyPlantBox! Control panel page.

The user must save the chosen settings by clicking the "Save" button. This action suspends Ruuvi tags data transmission, ensuring reliable communication. Additionally, the controls display the changes made on the last interaction with the panel, allowing to check the presets.

- Maintenance mode - a special mode designed for greenhouse maintenance. When activated and saved, this function turns off all pumps and fans allowing the user to perform essential tasks such as water replacement and calibration. This mode ensures a safe and convenient environment for maintenance activities.

- Login and Registration – the procedure which provides personalized access to the web application.

Calibration

The project in its current form doesn't include many components that need calibration. Calibration is necessary when there are any inaccurate readings. Nevertheless, the device manufacturers suggest calibrating equipment every month.

The calibration procedure for the dosing pumps is done by uploading the original code from the Atlas Scientific website. To save time, the code has been added to the GitHub page dedicated to this project. The dosing pump code is located in the "calibration" folder and is named "EZO_PMP_UART.ino". To start the calibration process, the appropriate pins must be specified in the SoftwareSerial line, which defines the receiver and transmitter pins. To make sure that the connection is established, open the Serial Monitor in Arduino IDE, and type "Status", the reply should be printed within seconds. Next, set the pump to dispense 10 millilitres via typing "D,10" to the command line and

measure the liquid dispensed. Finally, send the command "cal, (the amount that was actually dispensed)" and the calibration is done.

Note: only one pump can be calibrated at a time. Don't forget to comment out extra SoftwareSerial lines.

Calibrating the DFRobot pH sensor might involve a slightly more complex process. First, locate the code for the pH sensor in the "calibration" folder, named "DFRobot_pH_v2.ino." The corresponding pin must be specified. After the code uploading, the sensor must be immersed in two solutions, 7 and 4 pH. Start with the 7 pH and follow the values on the screen until they are stable. When the values have stabilized, type "enterph" and "calph". The Serial Monitor should respond, indicating successful calibration. To save the calibration send "exitph". Repeat the same steps with the 4 pH solution.

Note: if calibration is unsuccessful, use the modified library for the DFRobot pH sensor, which is compatible with CONTROLLINO MAXI. Alternatively, replace the code in the "DFRobot_PH.cpp" file located in the originally installed DFRobot library on your computer with the corresponding file from the GitHub repository.

Possible problems and bugs

These following shortcomings are possible when using the system:

- Immediate activation of pumps and fans upon system startup.

  Upon plugging in the system, the pumps and fans start operating immediately, contrary to the expected behavior where they should remain off until the user configures the settings.
  Resolution of the problem may involve triggering the "Maintenance Mode" within the code right after the application starts.

- Inconsistent operation of Python script for Ruuvitags data retrieval.

  The Python script designed to retrieve the data from Ruuvitags exhibits occasional failures, which did not occur during standalone testing. The error indicates a connection timeout, possibly due to interference with the Bluetooth connection from other electronics near the Raspberry Pi within enclosure.
  Resolution of the issue would involve investigation and mitigation of the source of interference. Another option would be complete replacement of Ruuvitags for a different sensor with a different communication type.

- Occasional website crashes due to database Locking.

  Periodic crashes of the website resulting from database congestion caused by multiple concurrent script accesses.

Resolution of this problem would be to reboot the Raspberry Pi, or manual restart of the web application, but this is not recommended.

- Exposure to water may disrupt the accuracy of water level measurements.

  If the water exposure is unavoidable, it is recommended to consider replacing the HC-SR04 with the waterproof version.

Further work

Possible HyPlantBox! project improvements may include:

- Elimination of the above mentioned issues.
- Humidifier control. Implementation of the humidifier control with CONTROLLINO MAXI and solid-state (SS) relay.
- Control panel enhancements. Expansion of the capabilities of the web application by adding more functions to the control panel e.g., milliliters to dispense, variable height of the water tank.
- Automated calibration. Integration of an automated calibration process into the system by adding a dedicated button or function to the web application.

Safety

It is not recommended to use the “main pump” in its current setup for different purposes as it may affect the pressure of the pump and eventually the current consumption. For example, a higher greenhouse or viscous liquid may increase the risk of exceeding the current limit (6 A).

Furthermore, any interaction with the automation box contents should only be done with the power supply disconnected, as the high voltage presents a safety hazard.

References

GitHub:

1. "HyPlantBox!" repository
   https://github.com/aalicc/HyPlantBox

2. Innovation project repository
   https://github.com/aalicc/innoproject

3. Modified DFRobot pH v2 library repository
   https://github.com/aalicc/DFRobot_pH_v2

CONTROLLINO MAXI:

1. https://www.controllino.com/product/controllino-maxi/

Raspberry Pi 4B:

1. https://www.raspberrypi.com/products/raspberry-pi-4-model-b/

Sensors' datasheets and instructions:

1. HC-SR04
   https://www.amazon.co.uk/HC-SR04-Ultrasonic-Distance-Arduino-MEGA2560/dp/B07XF4815H

2. DS18B20
   https://www.analog.com/media/en/technical-documentation/data-sheets/DS18B20.pdf

3. DFRobot Industrial pH Sensor v2
   https://www.dfrobot.com/product-2069.html

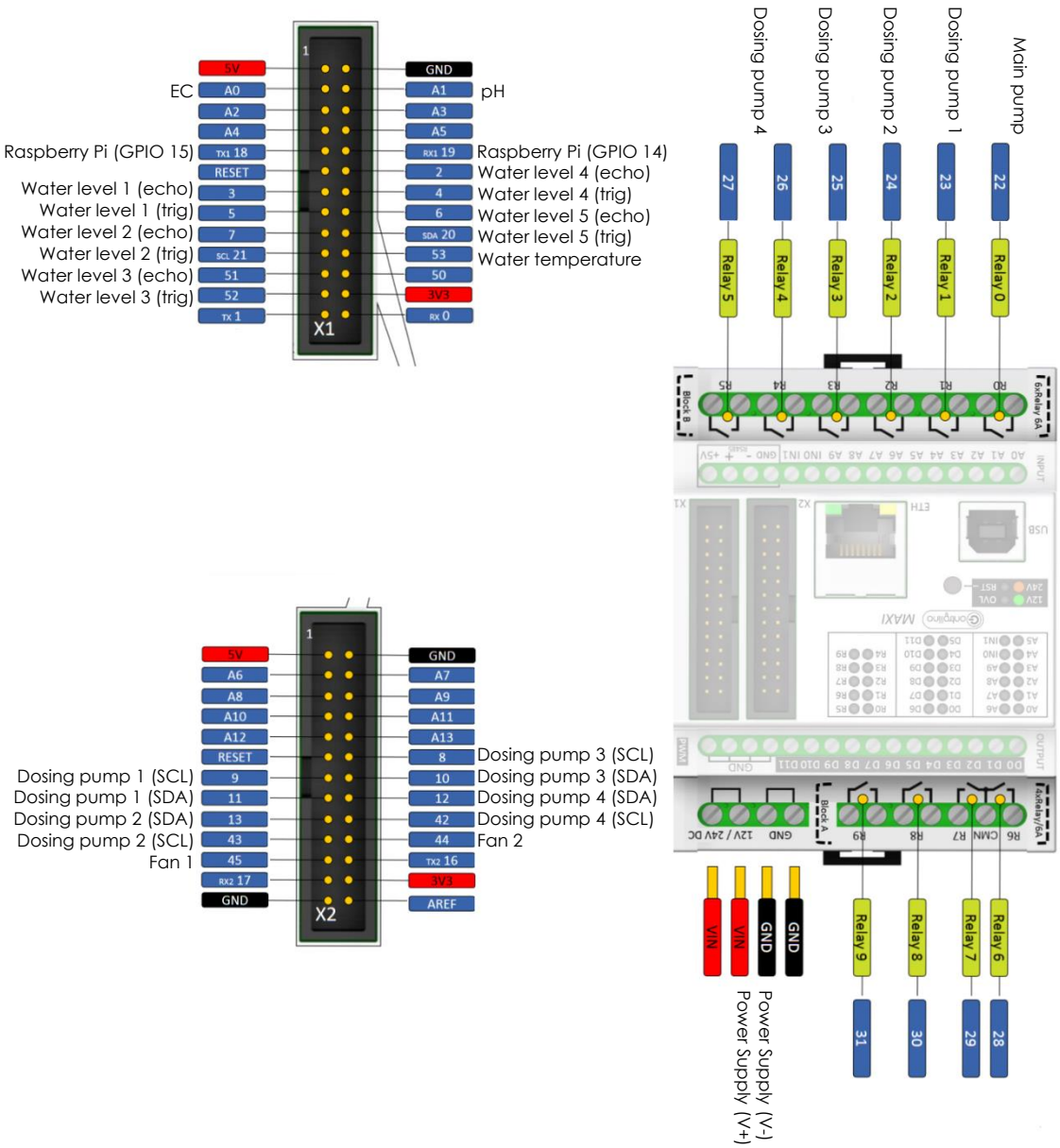
4. Grove – TDS Sensor
   https://wiki.seeedstudio.com/Grove-TDS-Sensor/

Motor driven devices and accessories:

1. EZO-PMP-BX
   https://atlas-scientific.com/peristaltic/sgl-pmp-bx/

2. MDD3A
   https://www.cytron.io/p-3amp-4v-16v-dc-motor-driver-2-channels

Appendix


X1
5V
A0
A2
A4
TX1 18
RESET
3
5
7
SCL 21
51
52
TX 1
GND
A1
A3
A5
RX1 19
2
4
6
SDA 20
53
50
3V3
RX 0
EC
pH
Raspberry Pi (GPIO 15)
Raspberry Pi (GPIO 14)
Water level 1 (echo)
Water level 4 (echo)
Water level 1 (trig)
Water level 4 (trig)
Water level 2 (echo)
Water level 5 (echo)
Water level 2 (trig)
Water level 5 (trig)
Water level 3 (echo)
Water temperature
Water level 3 (trig)
X2
5V
A6
A8
A10
A12
RESET
9
11
13
43
45
RX2 17
GND
GND
A7
A9
A11
A13
8
10
12
42
44
TX2 16
3V3
AREF
Dosing pump 1 (SCL)
Dosing pump 3 (SCL)
Dosing pump 1 (SDA)
Dosing pump 3 (SDA)
Dosing pump 2 (SDA)
Dosing pump 4 (SDA)
Dosing pump 2 (SCL)
Dosing pump 4 (SCL)
Fan 1
Fan 2
Dosing pump 4
Dosing pump 3
Dosing pump 2
Dosing pump 1
Main pump
27
26
25
24
23
22
Relay 5
Relay 4
Relay 3
Relay 2
Relay 1
Relay 0
Block B
6xRelay 6A
INPUT
USB
MAXI
OUTPUT
PWM
Block A
12V / 24V DC
GND
6xRelay/6A
VIN
VIN
GND
GND
Relay 9
Relay 8
Relay 7
Relay 6
31
30
29
28
Power Supply (V-)
Power Supply (V+)