iPONICS: IoT Monitoring and Control for Hydroponics

K. Tatas, A. Al-Zoubi
Frederick Research Center and
Frederick University,
Nicosia, Cyprus
{com.tk, st012422.stud}
@frederick.ac.cy

A. Antoniou
Adaptive Hydroponics Limited,
Larnaca, Cyprus
adaptivehydroponics@gmail.com

D. Zolotareva
Frederick University,
Nicosia, Cyprus
st014383@stud.fit.ac.cy

2021 10th International Conference on Modern Circuits and Systems Technologies (MOCAST) | 978-1-6654-1847-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/MOCAST52088.2021.9493387

***Abstract*-This paper presents the design, and implementation of an intelligent, low-cost IoT-based control and monitoring system for hydroponics greenhouses. The system is based on three types of sensor nodes: The main (master) node is responsible for controlling the pump, monitoring the quality of the water in the greenhouse and aggregating and transmitting the data from the slave nodes. Environment sensing slave nodes that monitor the ambient conditions in the greenhouse and transmit the data to the main node. Security nodes that monitor activity (movement in the area). The system monitors water quality and greenhouse temperature and humidity, ensuring that crops grow under the optimal conditions according to hydroponics guidelines. Remote monitoring for the greenhouse keepers is facilitated by monitoring these parameters by connecting to a website. The system is optimized for low power consumption in order to facilitate off-grid operation.**

Keywords—IoT, Wireless Sensor Networks, hydroponics, smart agriculture

I. Introduction

Water shortage and pollution are environmental threats affecting large parts of EU as well as the entire world. Climate change is expected to further increase water shortage and threaten food production. This problem cannot be solved by a single effort but must be addressed on many levels. Food security is dependent on several environmental conditions such as water quality and availability, soil condition, energy availability, etc. Due to the extensive and exhausting practice of agriculture during the last decades, several environmental issues have arisen. Climate change is expected to affect water quality and quantity already in decline [1] [2].

Smart farming is a capital-intensive and hi-tech system based on the application of modern ICT into agriculture. In IoT-based smart farming, a system is built for monitoring the crop field with the help of sensors and automating the irrigation system. Solutions for measuring environmental conditions and water quality based on electronic sensors and microcontrollers include ATLAS scientific [3], Libelium Waspmote [4] etc. A number of research papers demonstrate such solutions as shown in the recent surveys of [5] and [6].

In this paper, we present the design and implementation of a novel, low-cost hydroponics monitoring and control system based on IoT technologies. In particular, the system is composed of a specialized Wireless Sensor Network for monitoring the essential parameters for Hydroponics and control for the pump. It provides the user with a user friendly web-based tool to monitor his crops as well as being appraised by appropriate alarms and warnings. This greatly facilitates the observation of multiple hydroponics greenhouses with minimal effort and need for intervention.

The rest of the paper is organized as follows: Section II describes the background and Section III describes the proposed system in detail. The paper concludes with Section IV which summarizes the results and discusses future work.

II. Background and System Requirements

Hydroponics [7] relies on fertilized and aerated water which provides both nutrition and oxygen to a plant's root zone. It often involves sophisticated mechanization processes which can be daunting to casual hobbyists as well as small-scale commercial farms. Nutrient solutions must usually be below the temperature at which pathogen growth can begin, yet not so cool that root activity is suppressed. In hydroponics, as in conventional agriculture, nutrients should be adjusted to satisfy Liebig's law of the minimum for each specific plant variety [8].

As it can be seen from Table I, certain crops are tolerant to a wide range of conductivity and pH values, such as spinach, asparagus and tomato, while others, such as strawberry, are very sensitive especially to pH changes. This requires the close monitoring of the above variables to adjust the nutrients.

The iPONICS system facilitates remote monitoring of the values at low cost. Since it takes time for the plants to absorb the nutrients, a low sampling rate (as low as once per day) is sufficient for the electrical conductivity and pH. However, monitoring ambient temperature and humidity requires a higher sampling frequency since these vary significantly during the day and could lead to pathogen growth. Furthermore, we need frequent monitoring of ambient temperature to create an alarm in the case of fire. Furthermore, since the greenhouse can be situated at a remote location, it is required to add some security in order to be at least notified of, if not prevent, unauthorized entry. The most important iPONICS system requirements are summarized in Table II.

TABLE I. CROPS IDEAL EC AND PH VALUES [8]

Crops	EC(mS/cm)	pH
Tomato	2.0 - 4.0	6.0-6.5
Cuccumber	1.7 - 2.0	5.0 - 5.5
Strawberry	1.8 - 2.2	6.0
Banana	1.8 - 2.2	5.5 - 6.5
Spinach	1.8 - 2.3	6.0 - 7.0
Asparagus	1.4 - 1.8	6.0 - 6.8

TABLE II. IPONICS SYSTEM REQUIREMENTS

Requirement	Sampling period	Alarm/warning
Monitor water quality (pH, temperature, electrical conductivity, dissolved Oxygen)	daily	Warning depending on crops
Monitor ambient temperature	8sec, Record hourly	Alert (possible fire) Warning depending on crops
Monitor ambient humidity	Hourly	Warning depending on crops
Detect unauthorized entry	Event-driven (interrupt)	Alert
Put nodes to sleep to conserve energy	N/A	N/A
Provide simple user GUI (dashboard) for multiple sites	N/A	N/A

III. IPONICS SYSTEM DESIGN

Given the requirement of monitoring the temperature and humidity inside the greenhouse and the great variation in area between various types of greenhouses, the environment sensing system must be distributed across various nodes, the number of which will depend on the specific greenhouse dimensions. On the other hand, for monitoring the water quality and controlling the pump one central node is sufficient. The iPONICS system concept is shown in Fig. 1. The iPONICS sensing and control system is a Wireless Sensor Network with a star topology as shown in Fig. 2.

The system is composed of three nodes: The Sensing and Control Unit (SCU), the Environment Sensing Unit (ESN) and the Greenhouse Security Unit (GSN). The GHU contains one SCU which is the master node and a number of ESN and GSN "slave nodes" the number of which depend on the dimensions of the greenhouse.

A. Sense and Control Unit (SCU)

The SCU is the main (master) node as well as the central node in the star topology of the WSN. It is responsible for measuring the circulating water quality through four sensors: temperature, pH, water Electrical Conductivity (EC) and Dissolved Oxygen (DO). Furthermore, it is responsible for controlling the pump in an efficient manner: Conserving energy while maintaining the required water flow for crops growth and requesting and receiving data from the slave nodes.

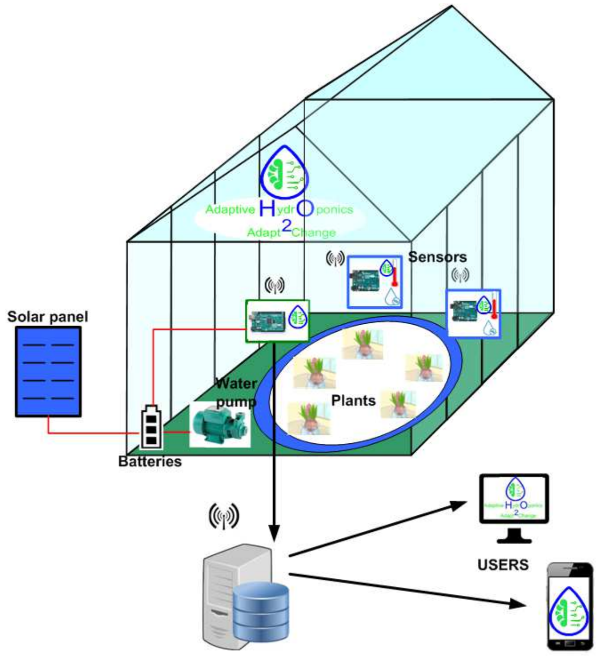

Fig. 1. iPONICS system concept

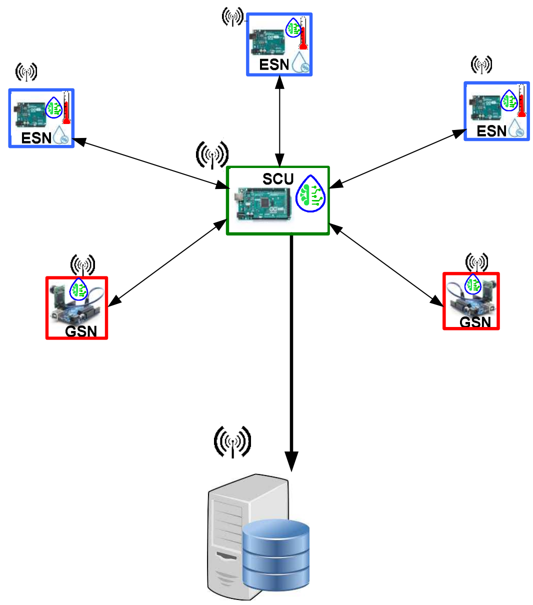

Fig. 2. iPONICS WSN topology

The sensors used were from Atlas Scientific [3]. The reasons for selecting the particular sensors were the following: Firstly, they require infrequent recalibration, which would be inconvenient while being deployed in a greenhouse during plant growth which requires weeks to months. Secondly, they are easily integrated in an Arduino-based microcontroller system, reducing development time and time-to-market. Thirdly, they are long lasting at a reasonable cost, essential attributes in order to keep the overall cost of the system an attractive investment for farmers.

The SCU processing unit is an Arduino Mega 2560, providing the required processing power, program memory and number of pins to support the following connections: the Xbee shield, the RTC shield, the Atlas-Scientific EZO circuits, the GSM shield, the Voltage sensor and, finally the Relay module control the ON/OFF switching of the water pumps. All the shield and modules are supplied by the power drawn out of the Arduino by the 5v and GND pins. Fig. 3 depicts the schematic of the SCU, while Fig. 4 illustrates its operation using a flowchart.

As shown in Fig. 4, the SCU starts by initializing the Serial and I2C communications and verifying all modules are connected. In order to preserve energy, the microcontroller is put to sleep along with the GSM shield since it's the most power consuming module. The microcontroller awakes on the following cases: First, an **Alarm_Interrupt** that signals an outstanding temperature/humidity readings, in which the SCU activate the GSM to send an **Alarm** SMS/Email and upload these readings to the server. Second, a **Security_Interrupt** that signals an unauthorized access to the site, in which the SCU activate the GSM to send a **Security** SMS/Email and upload the captured images to the server. Third, the SCU reaches the sampling time of water quality sensors, in which it collects those readings temporally to the EEPROM. Fourth, the SCU reaches the collection time of the environmental readings, in which it starts by activating the GSM shield and uploading the water quality readings, and then waking up each ESN at a time and upload its readings to the server. Finally, the SCU wakes up on fixed intervals for the pump control, in which the state of the battery along with the latest water quality readings dictates the powering decision.

Since the entire system is meant to be off-grid, it is crucial to minimize the power consumption of the units connected to solar charged battery whenever possible, to guarantee the uptime. The use of the sleep mode and intelligent control of the water pumps, ensures moderate usage of the power resources. As it can be seen in Table III, the SCU consumes around 1.25W while sleeping, where the GSM is off and the microcontroller is in deep sleep mode. Other connected modules are still powered including the Xbee shield, as Interrupts are Asynchronous. In the idle state, the Arduino is awake to regularly do the water quality readings or control the water pump, in which it consumes about 1.35W. The power consumption almost doubles when the GSM shield is activated for transmission, in which it consumes about 2.75W. This can rise up to 11.5W, given that the transmission is occurring under extreme weather conditions or weak network connection, which causes the GSM to draw higher currents. On the good side, without interruption, the GSM stays off until the upload time, which happens once a day. Fig. 5 shows an SCU prototype.

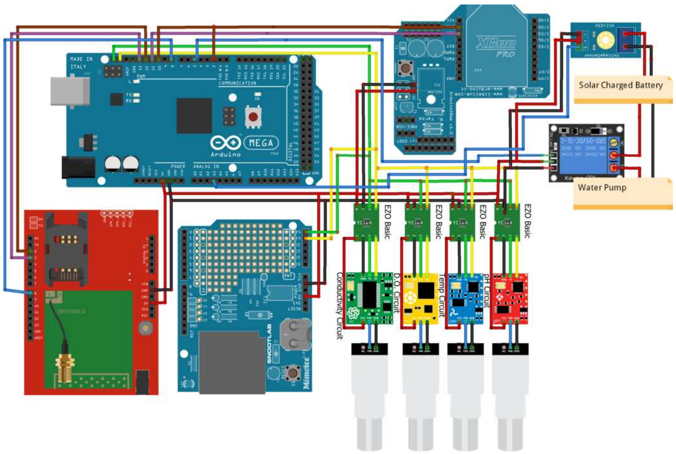

Fig. 3. SCU Schematic Diagram

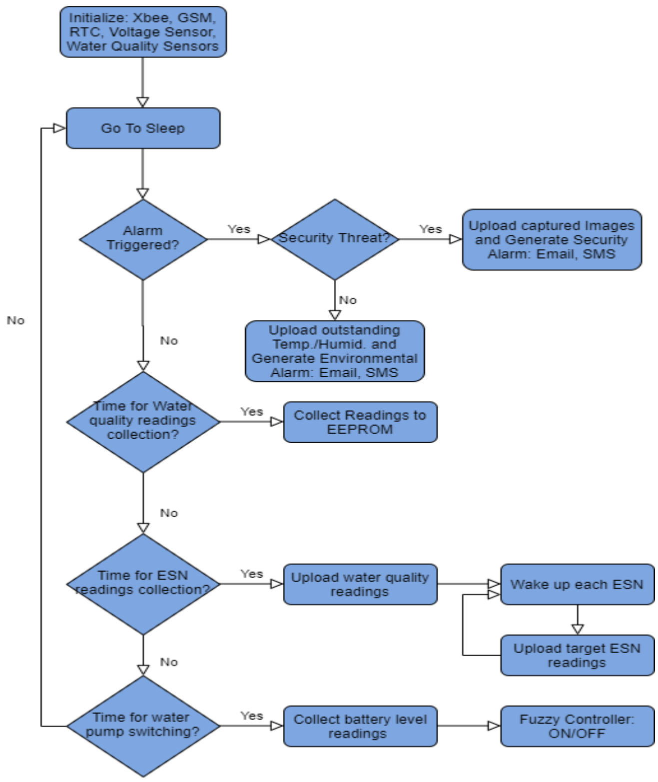

Fig. 4. SCU Flow-Chart

TABLE III. SCU POWER RATINGS

State	Current(A)	Power (W)
Sleep_mode	0.250	1.25
Idle	0.270	1.35
Transmission_idle	0.550	2.75
Transmission_extreme	2.300	11.5

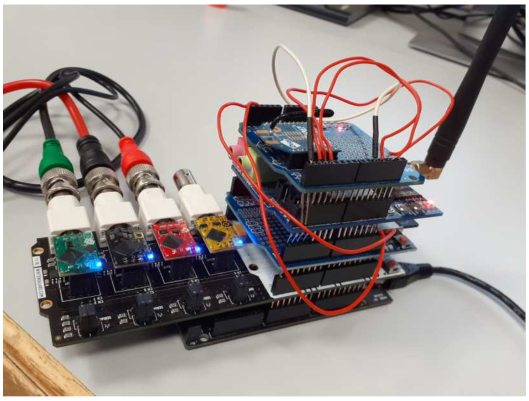

Fig. 5. SCU prototype

B. Environment Sensing Node (ESN)

The ESN is a slave node responsible for monitoring greenhouse ambient temperature and humidity and transmitting them to the SCU. The temperature is monitored every few minutes to detect any abnormal condition (like fire) early, but temperature and humidity readings are permanently stored in an hourly basis in an SD card. Data are transmitted to the SCU using Zigbee once a day unless there is an alarm (high temperature) or warning (faulty sensor or SD card). This allows effective monitoring of the greenhouse environment while minimizing data transmission and therefore power consumption. A number of ESNs are present in each greenhouse, depending on greenhouse dimensions. Each ESN has its own ID number and Zigbee address. Fig. 6 depicts the schematic diagram of the ESN circuit.

The ESN starts by initializing the Serial and SPI communication (Fig. 7), and verifying the connection to the modules. It awakes on fixed intervals, to collect the temperature and humidity readings to the SD card, and if no outstanding readings are present the ESN remains asleep until interrupted by the SCU for data collection. On the occurrence of the outstanding readings, the ESN interrupts the SCU with a Zigbee packet payload **"Alarm_Interrupt_XX"**. The **"XX"**, is either **"FIRE"** or **"Connection"** to distinguish whether the alarm is due to high temperature due to a possible fire in the greenhouse or just a connection problem.

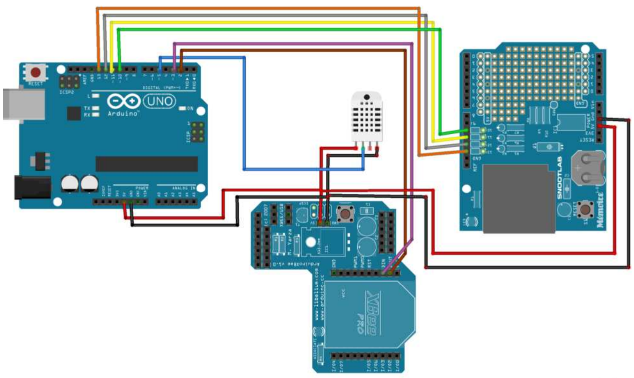

Fig. 6. ESN Schematic Diagram

The ESN consumes about 0.4W in the **sleep mode**, while it consumes about 0.55W during transmission. However, as mentioned earlier with SCU, the transmission only occurs once a day, unless there is an alarm. In both modes, the Xbee module stays active as it is an end device to be interrupted by the SCU.

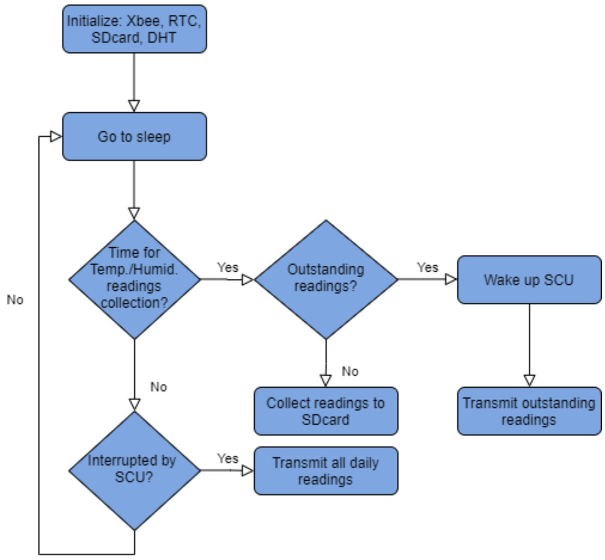

Fig. 7. ESN Flow-Chart

C. Greenhouse Security Node (GSN)

The GSN is a basic security node comprising a microcontroller with a motion sensor, camera, SD card and Zigbee transmitter/receiver. The reason such a node is required is that a greenhouse could be in an isolated area, and we wish to prevent unauthorized entry. The node sleeps to conserve energy, until woken up by an external interrupt from the motion sensor, in which case it activates the camera to take pictures and sends an alarm to the SCU through Zigbee. GSNs can be deactivated and reactivated when receiving a command from the SCU. Similar to the ESN, the GSN is built on top of the Arduino UNO. Fig. 9 depicts the schematic diagram of the node's circuitry.

As shown in Fig. 10, the node starts by initializing the I2C and SPI communications and ensure proper wiring of the modules. Then, the node falls into sleep until a motion is detected, in which the node expects an authorized ID swap, without any verification to the identity the GSN will interrupt the SCU and signals a **"Security_Interrupt"** inside the first payload, followed by the images data captured by the node. The GSN draws power of about 0.53W in sleep mode, while about 0.68W during transmission depending on distance.

D. Deployment and Evaluation

The iPONICS system is currently being installed in two experimental hydroponics greenhouses in order to grow crops, one in Nicosia and one in Larnaca, Cyprus and has been operating in a lab environment since September 2020. The monitoring website is shown in Fig.10. The user can monitor multiple greenhouses at a glance. There is a dashboard for monitoring the water quality and a sidebar for alerts and warnings such as sensor and SD card errors. The user can generate temperature and humidity graphs, with the default being the past 24 hours. The water quality sensor readings are shown as doughnut graphs color-coded the same way as the sensors (Fig. 10) for ease of monitoring.

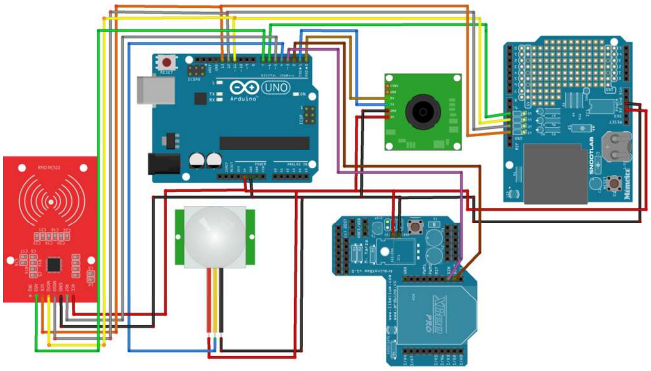

Fig. 8. GSN Schematic Diagram

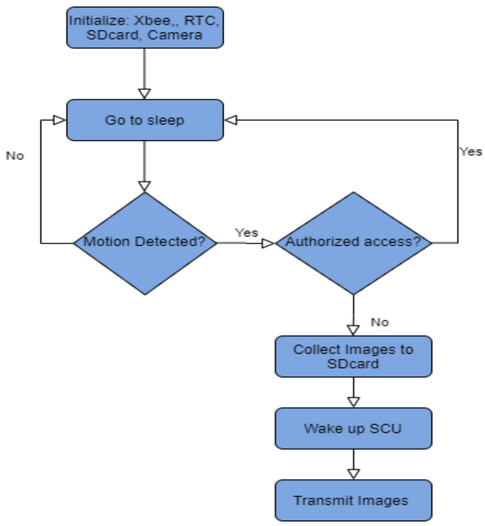

Fig. 9. GSN Flow-Chart

IV. Conclusions and Future Work

A novel hydroponics monitoring and control system based on IoT technologies was introduced. In particular, the system is composed of a specialized Wireless Sensor Network for monitoring the essential parameters for Hydroponics and control for the pump. It provides the greenhouse keeper with a user friendly web-based tool to monitor his crops as well as alarms and warnings allowing the observation of multiple greenhouses with minimal effort and need for intervention.

Our future work focuses on two directions: one is further and improved automation and the second is data analytics. The first approach includes more efficient water pump control using fuzzy logic and is already under development, while the second is predicting nutrient values based on the original concentrations and the water quality sensor values. In general, monitoring specific nutrients is a challenging problem for both terrestrial and space applications, even though several technologies exist [9]. Most of these technologies require costs that may be prohibitive to small scale farmers, who often add nutrients empirically. This could lead to either starved or even toxic plants. For now, we are using specialized nutrient monitoring equipment [10] to obtain data points daily. Accurately predicting the nutrient values would save the cost of such expensive equipment, while protecting from possible empirical errors leading to inaccurate addition of nutrients.

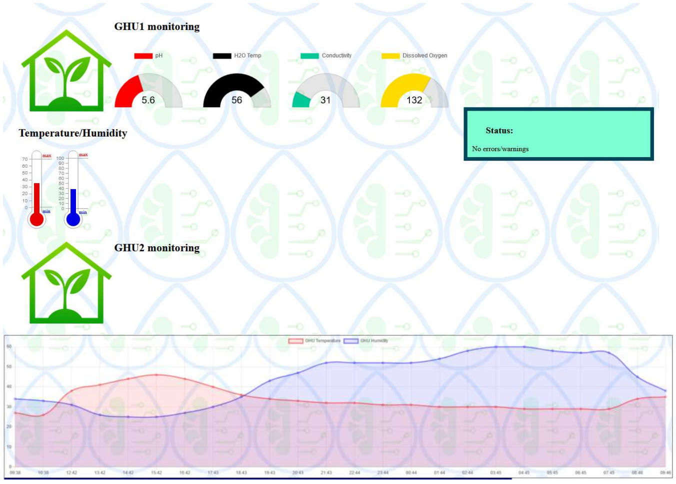

Fig. 10. Dashboard for monitoring greenhouse variables

Acknowledgment

This work was co-funded by the European Regional Development Fund and the Republic of Cyprus through the Research and Innovation Foundation (Project: START-UPS/0618/48 project title: "iPONICS: Smart Off- Grid System for Sustainable Hydroponics".

References

[1] N. E. Peters and M. Meybeck, “Water Quality Degradation Effects on Freshwater Availability: Impacts of Human Activities”, Water International, Volume 25, 2000 - Issue 2, pp. 185-193

[2] N. Khatri and S. Tyagi, “Influences of natural and anthropogenic factors on surface and groundwater quality in rural and urban areas”, Frontiers in Life Science Volume 8, 2015 - Issue 1, pp. 23-39

[3] https://www.atlas-scientific.com/, last retrieved: 28th November 2020

[4] http://www.libelium.com/products/waspmote/, last retrieved: 28th November 2020

[5] C. Verdouw, S. Wolfert and B. Tekinerdogan, “Internet of Things in agriculture”, CAB Reviews: Perspectives in Agriculture, Veterinary Science, Nutrition and Natural Resources 11 (2016).

[6] A. Tzounis et al., “Internet of Things in agriculture, recent advances and future challenges”, Biosystems Engineering Volume 164, December 2017, Pages 31-48

[7] J. D. Santos, et al., "Development of a vinasse nutritive solutions for hydroponics". Journal of Environmental Management. 114: 8–12, 2013

[8] Douglas, James (1985). Advanced guide to hydroponics: (soilless cultivation). London: Pelham Books.

[9] M. Bamsey et al., "Ion-Specific Nutrient Management in Closed Systems: The Necessity for Ion-Selective Sensors in Terrestrial and Space-Based Agriculture and Water Management Systems", Sensors (Basel). 2012; 12(10): 13349–13392.

[10] https://www.ionselectiveelectrode.com/pages/userguides, last retrieved: 28th November 2020