Smart Hydroponic Greenhouse: Internet of Things and Soilless Farming

SARA BARDI, University of Padua, Italy

CLAUDIO E. PALAZZI, University of Padua, Italy

Population growth and rapid climate change are leading to a reduction in arable lands and this requires an immediate solution. Hydroponic cultivation is one of the best approaches developed in recent years since this method requires water as a means of nutrition rather than soil. Thanks to its characteristics, hydroponics can solve many problems not only with regard to food supply but also health diseases such as allergy to heavy metals. However, this type of cultivation requires a lot of effort to ensure a perfect environmental condition and a balanced nutritional solution. For this reason, IoT technology is usually applied to automate all these processes. In this work we present a prototype of a hydroponic greenhouse integrated with IoT technology and a Dashboard for viewing data in real time. The sensors applied can monitor the temperature and humidity of the greenhouse, the water level in the basin and the amount of light. In the prototype there is also the possibility to turn on or off some electronic components. The aim of this work is to build a basic system for an automatic hydroponic greenhouse that can be customized or amplified according to your needs.

CCS Concepts: • **Computer systems organization → Sensors and actuators**; • **General and reference** → *Design.*

Additional Key Words and Phrases: Hydroponic, IoT, Smart Farming

ACM Reference Format:
Sara Bardi and Claudio E. Palazzi. 2022. Smart Hydroponic Greenhouse: Internet of Things and Soilless Farming. In *Proceedings of the Conference on Information Technology for Social Good (GoodIT '22), September 07–09, 2022, Limassol, Cyprus.* ACM, New York, NY, USA, 11 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

With population growth, climate change and environmental pollution, arable lands around the world are decreasing year by year. Indeed, the United Nations Food and Agriculture Organization (FAO) has declared that the world population will increase to 9.73 billion by 2050 and to 11.2 billion by the end of 2100 [8]. We will see this increase especially in crowded urban areas which will expand causing the disappearance of arable lands. Not only this, climate change has induced desertification and deforestation by changing the surrounding landscape and the water has been damaged by salinization [10]. This poses a significant challenge for farmers that are unable to meet the future demand for food with such a significant reduction in natural resources. This is why a soilless farming method is needed in rural and urban areas to address these problems.

Among all the new methodologies available, hydroponic agriculture is one of the most effective. By hydroponics we mean a soilless cultivation technique in which the plants are irrigated with a nutrient solution consisting of water and compounds necessary to provide all the essential elements for normal mineral nutrition. Thanks to the fact that it no longer needs soil and that it can be done anywhere, for example in a greenhouse or even at home, the use of this technique has increased exponentially in recent years, becoming a revolutionary prospect for the future of agriculture.

Despite all these good characteristics, hydroponics also have some critical aspects to keep under control in order to obtain a perfect result. Since there is no soil, it is necessary that the nutrient solution for the plants is balanced and contains all the necessary elements. While it sounds like an easy task, it really is not. In general, maintaining the quality of hydroponic cultivation occurs by manually controlling aspects such as the pH of the water, the total dissolved solids, the electrical conductivity (EC) and the water temperature [12]. However, doing it manually requires a lot of effort and also a good knowledge of this technique. Using the Internet of Things (IoT) technology can make this task much easier.

The term IoT refers to a system in which numerous sensors, controllers and actuators are connected with *things* using the Internet and a cloud server. IoT can be used to monitor and control many aspects of crop life without human intervention, making farming smart. Hence the term *Smart Farming* to indicate IoT technologies applied to the agrifood sector [4]. As for hydroponic agriculture, the integration of the IoT into the system could add other advantages to this methodology, such as efficiency and simplicity, which will make it appreciated even more. In addition, the sensors can collect data and automatically send it to a cloud server that can be accessed remotely via a web application or a mobile application [11]. This means that farmers can monitor plants even if they are far away or not constantly present on the farm. Therefore, resources can be easily optimized with a fully automated and independent system.

In this project, we propose an implementation of a smart hydroponic greenhouse which makes use of some sensors to monitor the temperature, humidity, light and water level of the greenhouse. All the data collected by the sensor are sent to a cloud server by using WI-FI and it is then displayed in real-time in a dashboard with the additional possibility, for the final user, to control remotely the power on/off of some components.

The paper is organized as follows: Section 2 overviews related work; Section 3 provides a general description of our project; Section 4 and Section 5 describe the practical implementation of the prototype considering both hardware and software; Section 6 provides information on the functioning of our solution; Section 7 analyses the outcome of some tests we performed; Section 8 describe future directions for this work; Section 9 concludes the paper.

2 RELATED WORK

In this section an overview about the main solutions for smart hydroponic greenhouse which have been proposed in literature will be provided. Each of them implements a different approach to embed the hydroponic cultivation with IoT technology.

In 2016 Desta Yolanda et al.[6] implemented an hydroponic system based on NFT which uses a fuzzy logic to control the EC and pH condition of the system. The prototype consisted in an IoT environment with wireless sensors, data logger and actuators which can be remotely controlled.

The authors of [9] developed hydroponics in arid zone. With Arduino and Raspberry Pi microcontroller they detected different parameter of the environment around the plants to reduce the human intervention. A mobile application was developed to control and monitoring the growth of the plants.

Melchizedek I. Alipio et al.[3] used a different approach by implementing a Bayesian network to analyze the data coming from the sensors. Light intensity, pH, EC, water temperature and humidity sensors were attached to the hydroponic for this task. Then, the data gathered from these sensors were analysed by the Bayesian network and sent to a web interface to let the farmer monitors the hydroponic farm.

Mehra et al. [7] proposed a prototype for tomato plants which integrated sensors. Sensors interface with a Raspberry Pi 3 and an Arduino board to monitor the parameters. The relevant feature of this work is that authors used Artificial Neural Networks (ANN) to process multiple parameter (such as humidity, temperature, light intensity) to provide the appropriate control action.

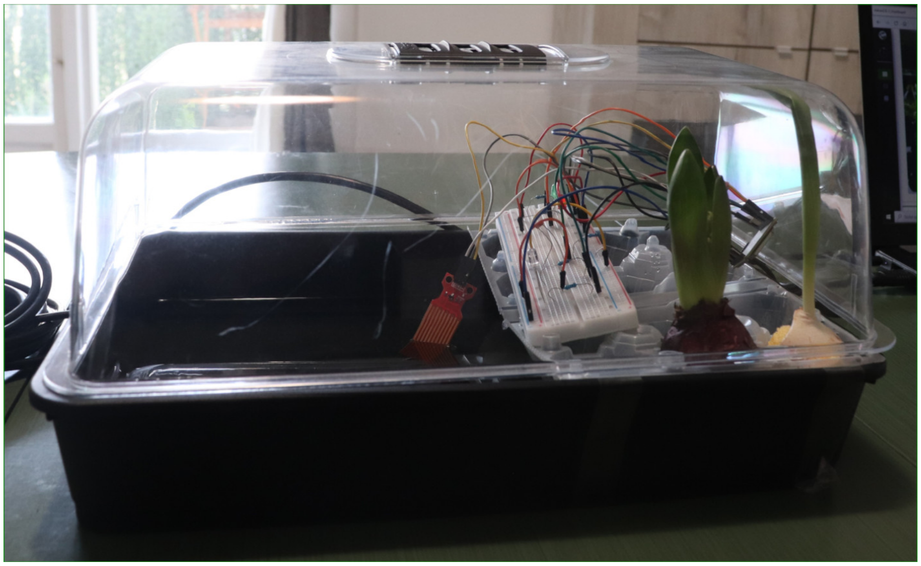

Fig. 1. Prototype of the smart hydroponic greenhouse.

In 2019, Demi Adidrana et al.[1] developed an hydroponic solution for lettuce and apply a k-Nearest Neighbor algorithm to predict the classification of nutrient conditions. Some sensor sensed the acidity or basicity of the water, the EC and the nutrient temperature. The prediction resulting from the KNN algorithm was then used to control a microcontroller to turn on or off the nutrition controller actuators.

As we can see, all the prototypes presented so far use sensors to monitor the environment around the plants in the hydroponic greenhouse. Each system collects some information from the sensors and uses it to remotely control actuators and plant growth through the use of an Android application or web interface. Some of them also use this data to build ANNs that can automatically control some microcontrollers. However, many systems are not that easy to customize or amplify. The goal of this project is to present a smart hydroponic greenhouse that can be easily modified to accommodate different crops but also manage the administrative and economic aspects of the farm without knowing too much the technical details behind the implementation.

3 PROTOTYPE DESCRIPTION

The project consists in a prototype of a smart hydroponic greenhouse. The prototype will be composed by a basin which will contain the water and the nutrient solution, a support structure for plants and a plastic shield as shown in Fig. 1.

The project is composed by a hardware and a software part. Sensors are used to monitor the hydroponic greenhouse behaviour and the surrounding environment to promote optimal growth conditions. A dashboard allows the user to monitor the hydroponic greenhouse in real-time and send remote commands to the sensors. In general, our dashboard provides these functionalities:

- Real-time data visualization: some charts to display temperature/humidity values and water/light percentages.
- Remote control: toggle button to let the user turns on/off the led for violet light.

- Some additional features: inbox mailbox, to-do list widget, interactive calendar, possibility to integrate a video surveillance plugin.
- Some additional information regarding the cultivation: number of plants, how many days has the greenhouse been active, last check of the correct functioning and an estimate of the incoming traffic from the sensors.

After a careful planning of all the necessary requirements for correct operation, all the necessary parts, both hardware and software, have been implemented.

4 HARDWARE IMPLEMENTATION

Regarding the hardware components, we employed:

- ESP32 board;
- Photoresistor;
- Water level sensor;
- HTU21D temperature/humidity sensor;
- RGB LED;
- Red LED;
- Green LED;
- Resistors of 100 Ω.

The general idea is that the ESP32 board coordinates the sensors and, after receiving the data from all of them, sends the obtained values to an open source broker. The HTU21D sensor measures the temperature and humidity inside the greenhouse, the photoresistor estimates the amount of light present and the water level sensor monitors the water level inside the basin. As for the two LEDs, the red one and the green one, they work with the water level sensor. If the water level is too low, the red LED starts flashing until the level reaches the optimal level. If, on the other hand, the water level is greater than or equal to 20% of the total water that can be contained in the basin, the green LED lights up, indicating that the hydroponic greenhouse is functioning correctly. Finally, the RGB LED was set to produce a magenta color useful for plant photosynthesis. However, by default, this LED is off and it is the user who can control it remotely. The electronic links among sensors and ESP32 board are shown in Fig. 2.

Among all the sensors of the prototype, the HTU21D sensor is the most complex to set since it needs to exchange more data than a single bit. For this purpose a serial communication is used and, in particular, I2C (Inter-Integrated Circuit) protocol has been implemented. I2C is a synchronous, half-duplex, master-slave with shared lines protocol. In synchronous protocols, a clock is used in the transmissions and it requires one more line which is the SCL (Serial CLock) line Fig. 3.

Half-duplex protocol means that the same physical line is used for both transmitting and receiving data and in I2C it is the SDA (Serial DAta) line. Finally, in a master-slave protocol, only one device (the master) can start the communication and, in this case, the master is the ESP32 board.

It is worth noting that the outputs from the sensors are mainly digital but there are some of them which are analog such as the values from the photoresistor and the water level sensor. In particular, analog signals are signals which measure physical quantities that vary continuously. However, we are not able to process such quantities as they are and, for this reason, we need to convert them in a digital signals. The ESP32 board has a built-in ADC (Analog-to-Digital Converter) module which can perform this operation.

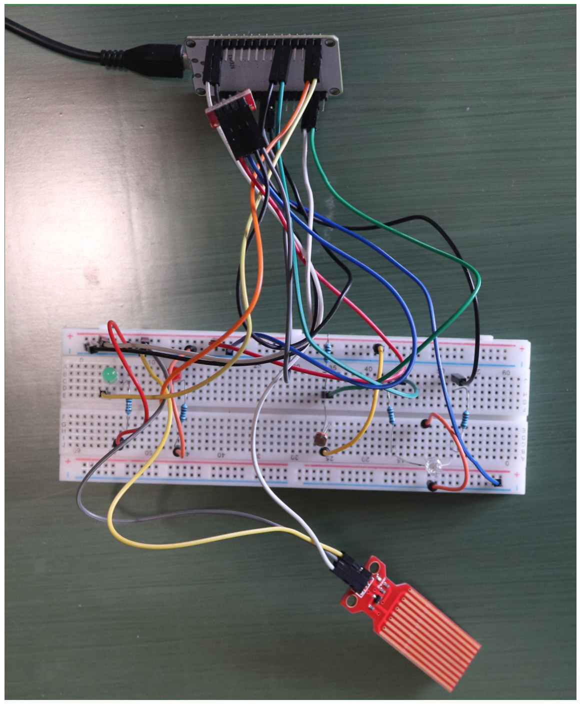

Fig. 2. Electronic links among sensors and ESP32 board.

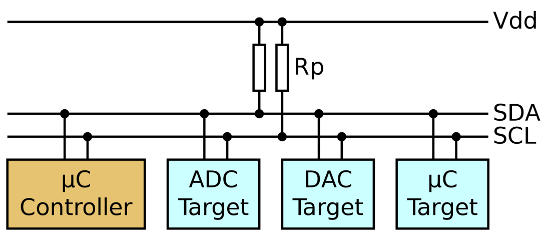

Fig. 3. I2C communication protocol [5]

All collected data is sent to the broker using the WiFi module of the board where they can be retrieved to create the dashboard.

5 SOFTWARE IMPLEMENTATION

The software implementation consists of a dashboard implemented in python which shows in real-time the sensors' data. The dashboard has been built using Flask, which is a python web framework, and AdminLTE template which is an open source web application template for admin dashboards and control panels.

5.1 MQTT

The building block of this project is the way sensors and the web application communicate with each other. This is not an easy task as a publish/subscribe paradigm is required. The general idea is that some entities can publish data with certain names, known as topics, and other entities can subscribe to updates of that data. In this protocol, the word "topic" refers to a string that the broker uses to filter messages for each connected client. Topic can consist of one or more levels and each of them is separated by a forward slash. Usually a topic is of this type:

`mainTopic/SubTopic/SubSubTopic/...`

An example can be:

`myhome/kitchen/temperature`.

In this project there are three entities: the sensors which are controlled by the ESP32 board, the web application written in python and the broker (`test.mosquitto.org`). To allow all these actors to communicate, the MQTT (Message Queue Telemetry Transportation) protocol has been adopted.

MQTT is a lightweight protocol for IoT communications: it is reliable, simple, has minimal overhead and efficient performance, and most importantly, it supports the publish/subscribe paradigm. The general behavior is that the sensors connect to the broker and publish the data in some topics using MQTT and, on the other hand, the web application also connects to the broker, subscribes to the same topics and retrieves the published data.

MQTT has three QoS (Qualities of Service) types for message delivery: QoS0, QoS1, QoS2. We have evaluated all three of them and assessed their performance in Section 7. We can already anticipate that QoS0 resulted the fastest one for message delivery.

The topics created for this project are the following:

`wnma_bsara/temperature`,

`wnma_bsara/humidity`,

`wnma_bsara/light`,

`wnma_bsara/water`,

`wnma_bsara/led`.

The publisher for the first four topics is the board while for the last it is the web application that publishes based on the request of the end user.

Actually, letting the ESP32 board communicate with the broker is not that difficult: there is a python library that can handle callbacks for publishing and subscribing. The non-trivial part was to retrieve the data to be displayed in the dashboard. In fact, the data should flow from the python back-end to the user interface written in JavaScript and HTML.

Furthermore, this request/response system must be asynchronous: once the main page of the dashboard has been loaded and the first set of values is displayed in the appropriate graphs, there should be a different mechanism whose task is to update the display of data every time received new messages from the broker. In general, this part of the project was the most complex as the only way to make this whole process working correctly was to build a WebSocket.

A WebSocket is a communications protocol providing full-duplex communication channels over a single TCP connection [13]. It enables interaction between a web browser (in this case the dashboard) and a web server (in this case the broker), facilitating real-time data transfer. Thanks to this mechanism, data is loaded in a dynamic and asynchronous way allowing the dashboard to be fully compatible with the real-time requirement.

AdminLTE [2] is a open source WebApp template for admin dashboards. It is a responsive HTML template based on the CSS framework Bootstrap 3, meaning that it can adapt itself automatically to different size of screen width (computer, tablet, smartphone). Bootstrap 3 is a free and open-source framework which is widely used in building web applications. It contains HTML, CSS and JavaScript libraries which simplifies the development of web pages.

From Fig. 4, it is possible to notice that some customization has been made in order to address the requirements of the smart hydroponic greenhouse: first of all, the main page of the dashboard displays an interactive line chart inside which the temperature/humidity values are shown. This chart was made by using the Highchart JS plugin. Furthermore, there are two pie-progress components to indicates the water and light percentages. They were built with the jquery-asPieProgress plugin. Next to these, there are some additional feature that can be implemented as future work. In the main page, there is also a toggle button to let the user decides whether to turn off/on the RGB LED in the greenhouse. Finally, there are also some widgets like the "To Do List" with draggable tasks, the inbox mailbox made with jQuery UI and a Full Calendar instance with draggable events and the option of adding new ones. Once the user interface was built, it has been integrated with the Flask app.

5.2 AdminLTE

6 PROTOTYPE FUNCTIONING

After a description of each block necessary for the prototype, it is now possible to illustrate the general framework that is depicted in Fig. 5.

The ESP32 board collects data from the sensors and sends it to the broker (`test.mosquitto.org`) using MQTT. At the same time it subscribes to the topic to receive updates regarding the RGB LED. On the other hand, there is the Python Flask app which is capable of building the dashboard in real time. The communication between broker and dashboard takes place via WebSocket which is composed of two main blocks: the first, written in python, connects to the broker and subscribes to the correct topics using the MQTT protocol; the second, written in JavaScript, listens on localhost waiting for data from the python back-end that will be displayed in the user interface using JavaScript as well.

Finally, when the end user clicks the toggle button to turn the RGB LED on/off, this event is handled by the jQuery UI and a message is sent to the python back-end from the JavaScript coded part. The message is then published in the corresponding topic that the ESP32 has subscribed to.

The user interface is built on top of AdminLTE which has been customized by modifying some HTML, CSS and JavaScript files in order to obtain a dashboard suitable for a hydroponic greenhouse. Then, the model was integrated into the Flask app thanks to the Jinja template engine.

7 ANALYSIS AND RESULTS

Many tests have been performed to ensure the actual efficiency of the smart hydroponic greenhouse. First of all, all the sensors have been tested for accuracy. Regarding the HTU21D, an indoor environmental thermometer was used to check if the reported temperature and humidity were exactly those measured. As we can see from Fig. 6, the sensor deviates by 0.3 degrees for the temperature and 3% for humidity with respect to the target values. This shows that the sensor values are quite accurate.

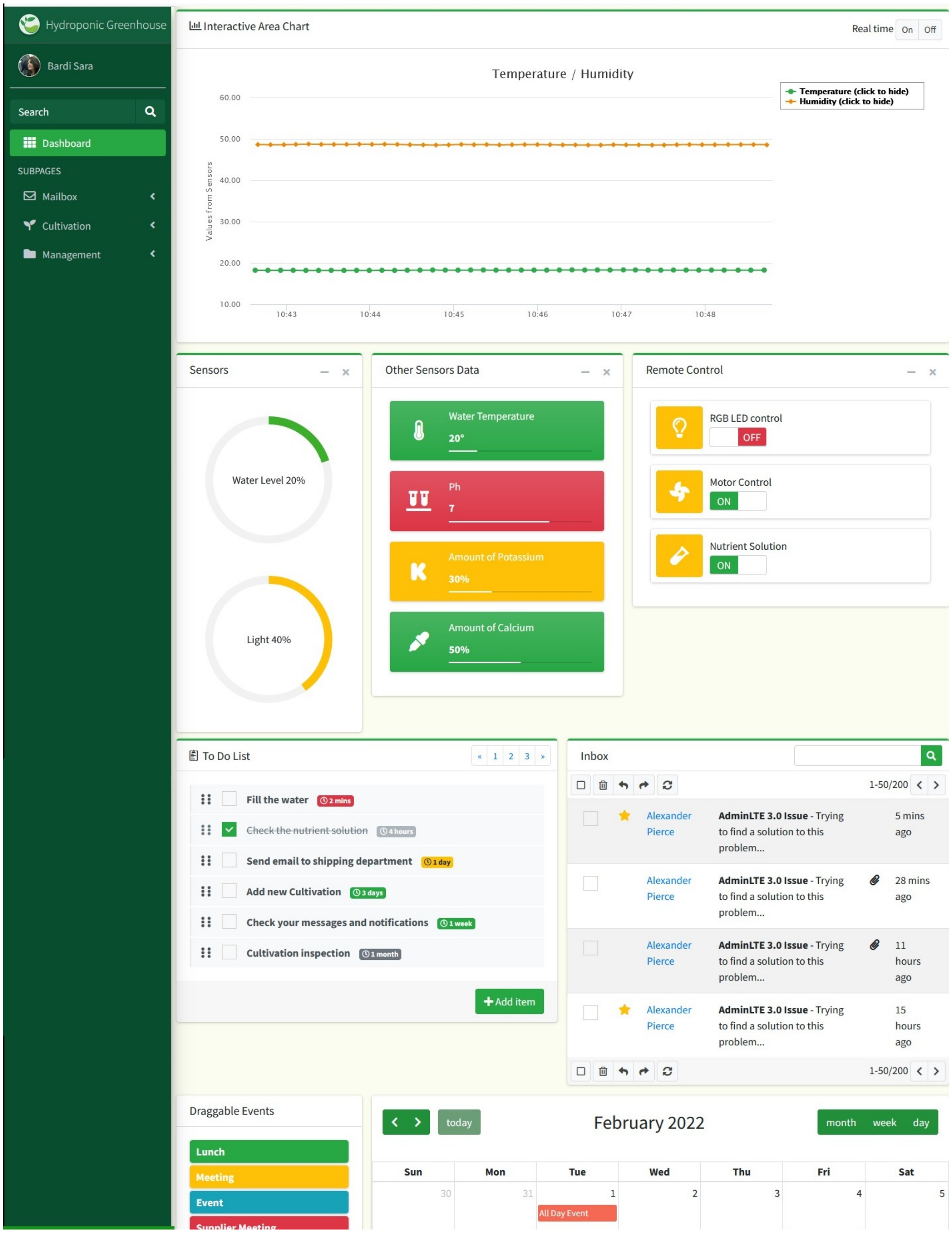

Fig. 4. A view of the dashboard.

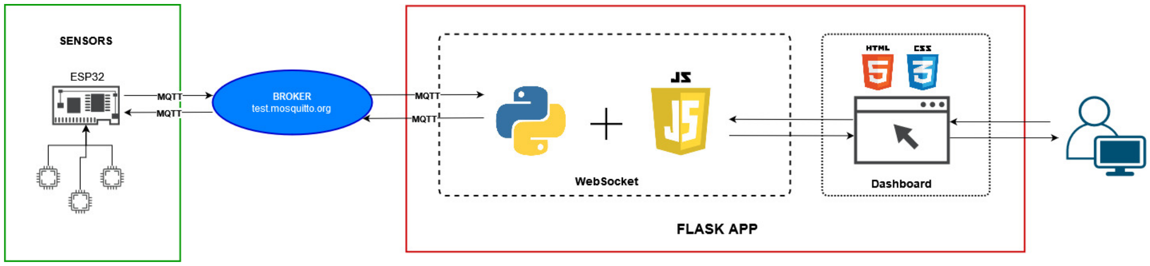

Fig. 5. The general framework of the prototype.

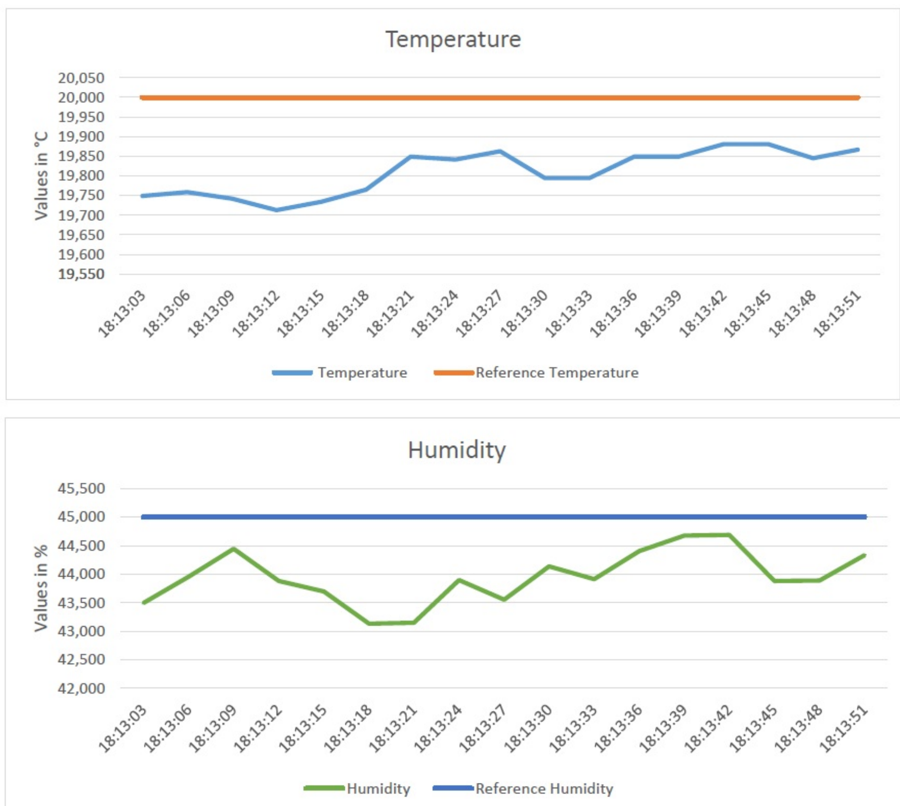

Fig. 6. Values measured by HTU21D

For the water level sensor, the values have been set taking into account the dimensions of the basin so they are consistent with the quantity of water contained in it. The less accurate sensor is the photoresistor. Actually, photoresistor is not designed to perceive minimal variations of light but it is more precise when the environment passes from total light to total dark. For this reason the measured values were influenced by some oscillations.

Table 1. Experimental assessment

	Payload Bytes	time QoS0	time QoS1	time QoS2
Temperature	19	3,031s	3,084s	3,096s
Humidity	17	2,962s	2,995s	3,025s
Light	2	2,013s	2,041s	2,088s
Water	2	2,002s	2,008s	2,032s

After testing the sensors, some experiments were made to verify the correct functioning of the prototype. The dashboard correctly displays the values that are measured by sensors. To investigate deeper in the possible delays that may happen, the prototype was evaluated with different QoS for MQTT.

As we can see from Table 1, the largest packets are those for temperature and humidity, 19 bytes and 17 bytes respectively. In average, the time it took to transmit these packets from the sensors to the Python back-end was approximately 3 seconds. On the contrary, the packets for light and water are much smaller, around 2 bytes, and the time required fluctuates around 2 seconds. This is exactly what we expected as the HTU21D needs to transmit larger packets than the other two sensors. However, this does not affect the real-time requirement of the dashboard as 3 seconds does not influence the user experience.

In addition to all these aspects, there is one important detail that should be mentioned. A lot of tests have been run in order to figure out which QoS is best in this scenario. Actually, none of the three appear to cause significant delays in transmissions. QoS1 and QoS2 seem to take a few extra milliseconds to transmit packets but the delay is not that significant. A possible explanation may be that the dashboard is hosted on localhost or perhaps there have not been so many duplicates or packet losses. For this reason QoS0 has been kept for this project.

8 FUTURE WORK

This project is an open project, meaning that many additional features can be added to make the prototype more powerful. Thanks to the flexibility of the implementation, new hardware components can be inserted in the greenhouse by just adding few lines of code.

Regarding the dashboard, it has been developed with the idea of keeping the modularity of the whole framework. New functionalities may be added to improve the user experience: for example, a video surveillance plugins, different report systems, other ways to remotely control the greenhouse.

One part that is missing in this project is a database solution, for example using SQLlite or MySQL or even some cloud like Firebase. It would be very interesting to implement this part since it would allow the prototype to really work in practice. With the support of a database, it is possible to implement a real and dynamic mailbox or even a complete system that tracks products, sales, human resources and the different locations for the greenhouse. For example, having a database can open up to the possibility of adding a mobile application that would work in parallel with the Web App.

Another idea that can be considered is to diversify the cultivation: for example we could have a lettuce cultivation, a tomato cultivation and an onion cultivation all monitored by the same system. Even if each of them needs different environmental conditions and a different nutritional solution, with this prototype it is not so difficult to expand the basic version to be able to manage them all.

This project is based on the general idea that everything should be customizable to the needs of the end user. Hence, everyone should be able to modify it as simply as possible and without knowing much about the framework behind it.

9 CONCLUSION

The *Smart Farmer* is the new frontier of a more eco-friendly and efficient way to cultivate. When it is intended for the hydroponic cultivation, it can be extremely useful in many different scenarios helping to fight against population growth and environment pollution. The possibility of incorporating this type of greenhouse with the IoT technology makes it even more interesting and desirable.

In this paper I presented a prototype of a smart hydroponic greenhouse which consists in both hardware and software implementations. Many technologies were used in order to achieve the best result possible: for instance Flask web application, the MQTT protocol, the WebSocket, AdminLTE template and JavaScript plugins.

The main challenge of this project was to create something which gives the possibility to interlace different type of skills: hardware, software and graphics competences were fundamental in order to meet the goal of designing a product that works correctly and can be easily expanded at will.

In conclusion, I can say that even if many other additional features can be added to the original project in order to make it more powerful, the final result meets all the requirements planned at the beginning of the project.

ACKNOWLEDGMENTS

This work has been partially funded by the Department of Mathematics of the University of Padua through the BIRD191227 project.

REFERENCES

[1] D. Adidrana and N. Surantha. 2019. Hydroponic nutrient control system based on internet of things and KNearest neighbors. In *IEEE International Conference on Computer, Control, Informatics and Its Applications*. 1–6. http://dx.doi.org/10.1109/IC3INA48034.2019.8949585

[2] AdminLTE. [n.d.]. https://adminlte.io/

[3] Melchizedek I. Alipio, Allen Earl M. Dela Cruz, Jess David A. Doria, and Rowena Maria S. Fruto. 2017. Smart Hydroponics Farming System Using Exact Inference in Bayesian Network. In *IEEE 6th Global Conference on Consumer Electronics*.

[4] C. Brewster, I. Roussaki, N. Kalatzis, K. Doolin, and K. Ellis. 2017. IoT in agriculture: Designing a Europe-wide large-scale pilot. In *IEEE Commun. Mag.*, Vol. 55. 26–33.

[5] Wikipedia contributors. 2022. I²C — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=I%C2%B2C&oldid=1070685598 [Online; accessed 13-February-2022].

[6] Yolanda D.and Hindersah H., Hadiatna F., and Triawan M. 2016. Implementation of real-time fuzzy logic control for NFT-based hydroponic system on Internet of Things environment. In *6th International Conference on System Engineering and Technology (ICSET)*. 153–159.

[7] M. Mehra, S. Saxena, S. Sankaranarayanan R. J. Tom, and M. Veeramanikandan. 2018. IoT based hydroponics system using deep neural networks. In *Comput. Electron. Agricult.*, Vol. 155. 473–486.

[8] United Nations. 2016. World Population Prospects (Key Finding and Advance Tables). https://esa.un.org/unpd/wpp/publications/files/key_findings_wpp_2015.pdf

[9] Vaibhav Palande, Adam Zaheer, and Kiran George. 2017. Fully Automated Hydroponic System for Indoor Plant Growth. In *International Conference on Identification, Information and Knowledge in the Internet of Things*. 482–488.

[10] N. Panwar, S. Kaushik, and S. Kothari. 2011. Solar greenhouse an option for renewable and sustainable farming. In *Renewable Sustain. Energy Rev.*, Vol. 15. 3934–3945.

[11] Rayhana R., Xiao G., and Z. Liu. 2020. Internet of Things Empowered Smart Greenhouse. In *IEEE J. Radio Freq. Identif.*

[12] K Roberto. 2004. How to Hydroponics. In *Futuregarden Press*.

[13] Wikipedia contributors. 2022. WebSocket — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=WebSocket&oldid=1069644845