Journal
Appl Hort
Journal of Applied Horticulture, 22(1): 3-7, 2020
DOI: 10.37855/jah.2020.v22i01.01
Journal of Applied Horticulture
ISSN: 0972-1045

Implementation of a sustainable and scalable vertical micro-farm

Khushal Khan Liwal, Manohar Vohra, Hashir Sheikh, Obada Al-Khatib*, Nidhal Abdul Aziz and Czarina Copiaco

Faculty of Engineering and Information Sciences, University of Wollongong Dubai, Dubai, UAE.
**E-mail: ObadaAlKhatib@uowdubai.ac.ae*

Abstract

This paper summarizes the implementation of a vertical Micro-Farm that uses a WiFi network to communicate with sensors and actuators from multiple nodes. It addresses the issue of ordinary vertical farms, which require the user to monitor it occasionally to provide fertilizer and water. The system can be easily configured to automatically control supply of nutrients, water and light requirements for various plant types through a web enabled Interface. The web dashboard can further provide complex analysis of the whole system by collecting values from different sensors. The designed vertical farm system is power efficient, self-sustained, and can be setup easily by the user as each vertical rack acts as a single node or module. The user only needs to plant the seeds and fill up the tanks. Due to the modular approach, the system is also scalable without requirement of more complicated materials or wiring.

Key words: DHCP, microcontroller, Raspberry Pi, sustainability, vertical farming

Introduction

Vertical farming can be thought of as a method to counteract against the ever-growing urban areas around the world diminishing the availability of agricultural spaces (Despommier, 2009, 2013 and Al-Chalabi, 2015). It also opens a gateway that will allow people to easily grow herbs and vegetables indoors, well-assured that the harvest is completely organic and therefore much healthier (UN, 2019). In addition, vertical farms are also beneficial to the environment, having a smaller carbon footprint compared to traditional farming (UN, 2019).

With ordinary vertical farms however, users will still have to take out the time from their busy schedules to provide water and fertilizer, not to mention a source of light which can be sunlight or artificial. If sunlight is chosen, the user might have to find a spot which has optimum availability. Whereas, artificial lights must be efficient in two aspects: to provide the appropriate color for the specific plant-type (Olle *et al.* 2013), and power-efficient to avoid large electricity bills.

To address these drawbacks, this paper provides a solution in the form of an IoT-enabled vertical farm. We provide a unique, fully self-sustained, and power-efficient vertical farm which can be placed anywhere indoors, at the convenience of the user. The user simply needs to setup the shelf, place the soil containers onto the racks, place the seeds within, and fill-up the water and fertilizer tanks. From there on, the user can access the one-of-a-kind web-enabled dashboard provided. We call this novel ecosystem software the "Digital Farmer", which as the name suggests, takes responsibility of the farm on behalf of the user providing promising harvests. Here, the user can also configure as well as find vital real-time information on the farm.

The Digital Farmer kit comes with advanced software features that provide different configurations for growing a wide range of herbs and vegetables together with an automated watering and fertilizing system. In order to create a communication link between the shelves and the microcontroller-based server, on-board Wi-Fi modules have been utilized. Through this, the REST API is used to seamlessly communicate control signals to the server. On the other hand, data and status information of the farm is transmitted using HTTP. Further information about the system is provided in Proposed Solution Section.

With the expected increase in worldwide population, alongside the current climate change epidemic, the need for a better agricultural method is inevitable (Al-Chalabi, 2015; Despommier, 2009). Vertical farming poses as the perfect solution. The farm is basically multiple layers of a regular farm stacked on top of each other provided with light, controlled temperatures, water, and nutrients. Although plenty of research has been carried out within this field (Touliatos *et al.*, 2016; Ruengittinun, *et al.* 2017; Kalantari *et al.,* 2018 and Tsai and Liang 2018), there is yet to be a solution which is capable of providing a fully self-sustained, scalable, cost effective, power-efficient, and water-efficient farm available to the user as a plug-and-play product.

Tsai and Liang (2018) proposed a monitoring system for a vertical farm. The system however, is only a proof of concept without actually testing it practically on the farm. The components used are quite basic with the Arduino Yun being the only microcontroller and only a single temperature sensor being tested. The system design involves collection of sensory data, which is written to a file from which it is uploaded to a database and the dashboard.

Although the work shown does have a real-time dashboard, and back-end database, using the Arduino Yun for this has its limitations. For example, it is an expensive microcontroller to be used for a shelf, where the up-scaling of the farm is simply impossible. In addition, the Arduino Yun has limited pins on-board, which will not only limit the number of sensors but also the number of vegetables or plants that can be grown within one shelf. All in all, the authors fail to mention how the collected data can be used for the farm, no readings on the power consumption as well as how the system can be fully developed to use on the farm.

While drip irrigation is one of the most efficient methods for watering, hydroponics is an alternative where plants or vegetables are grown without soil (Despommier, 2009). Ruengittinun *et al.* (2017) explore the use of hydroponics within small vertical farms. The system proposed by the authors is similar to the one presented in this work but oriented towards hydroponics. The system design once again has sensor information being collected and stored into a database through a lightweight messaging protocol called MQTT broker. A mobile application is used to represent the system dashboard with manual and automatic modes. Similar to Tsai and Liang (2018), an Arduino microcontroller is being utilized although the model is the cheaper Uno in this case. It is coupled with an Arduino WiFi shield, which handles the data transmission.

The control system in Ruengittinun *et al.* (2017) does not include any form of hysteresis to the decision-making process, which may lead to many on-off cycles of the motors. Furthermore, the proposed dashboard also does not take into account the plant type to automatically monitor the growth process. For instance, the user must enter the configuration which they might not know; therefore, still requiring some farming experience. As for the results, key attributes to the performance of the farm such as the power consumption, whether or not the automatic mode is able to grow plants or vegetables, and if it is scalable, is missing.

A conventional vertical farming approach was taken by Ismail *et al.* (2017). In their IoT-enabled vertical farm, a setup had an Arduino Mega with an Ethernet shield, a water supply which uses a DC motor, LEDs, and soil moisture and water level sensors. The way the system operates is, the user must log on to the web browser, go to the webpage of the dashboard, and see the current status of the farm. From there, extensive inputs from the user are required as it does not take any actions itself but rather expects the user to click on/off manually. The concept is that the user will be notified when there is less water, and based on judgement, an action must be taken. This is not convenient for the user as it is time-consuming.

In addition, the design stores information locally on the Arduino memory, which is limited and cannot be used for advanced analysis. Moreover, having to use Ethernet instead of wireless might be inconvenient since the farm must be placed near available ports. Moreover, the farm does not support a variety of LED colours, which is a drawback to some plant/vegetable types as it may not enhance growth to the maximum potential (Olle *et al.*, 2013). The design also includes one temperature sensor for all racks, as the room temperature does not vary within such a small distance across the shelf. As for the dashboard, it is limited to consist of only text and is not real-time as it runs on the Arduino locally. On the other side, the farm does good use of excess water by taking it back into the storage but this would not be needed if a drip irrigation system was put in place. Overall, the system is not suitable for a fully-automated vertical farm.

Aeroponics is another type of growing technique whereby the roots are upheld in air and nutrients are supplemented through vapour (Despommier, 2009). Belista *et al.* (2018) have proposed a scalable vertical farm, which utilizes aeroponics, and promises a product that is catered towards households. The farm is monitored by a water-level sensor, pH level sensor, and temperature and humidity sensors. Sensor data is stored on a network-attached storage (NAS) device. The microcontroller, which does the storage and monitoring, is the Raspberry Pi Zero W. This device comes with no analog-to-digital converter and no substitute has been mentioned in the paper. Moreover, the framework provided does seem ideal for scalability but lacks details on how this can be done. LEDs have been optimized for specific plants providing the correct light spectrum for growth; however, no information on how this will be included are provided. Since no implementation seems to have been conducted, the power efficiency and cost-effectiveness are not provided either. All in all, the design seems to have all advanced features but are not yet completed and so the question on its feasibility arises.

Another smart vertical farm system was proposed by Stevens *et al.* (2018). The system uses an Arduino Mega, which has a Grove Base shield that is a placeholder for all its sensors. Logically, the purpose of the Arduino is to collect sensor data and forward it to a Raspberry Pi v2 through a wired USB connection, which does not allow this system to up-scale easily. The Raspberry Pi is responsible for taking all actions based on the readings as well as storing this data on the cloud server. A dashboard allows the user to monitor the sensor data but also allows configuring what levels of sensor readings should be maintained. No details are recommended to the user based on the plant or vegetable type, and so the user must have some knowledge about the growth conditions, which is not convenient nor user-friendly. But, the system does provide a wide range of controls which can allow an advanced environment control over the farm. Although, there is no feature which allows changing the LED colors which, as mentioned above, can enhance the growth. The power consumption and cost of the system have not been stated. The paper also fails to mention the algorithm with which the automation is actioned, whether there is any hysteresis provided as actuators are activated based on sensor readings.

Kalantari *et al.* (2018) conducted an in-depth survey on some of the opportunities and challenges faced within vertical farming. The paper considers plenty of vertical farms, most of which are on a large-scale. Furthermore, the features of each farm are also outlined. Common elements of the farms are LED lighting, sensors, efficient watering, and automation. The paper states the various environmental, economic and social impacts of vertical farming most of which provide a great prospect to this field. The aim of our proposed solution is to be able to bring the high-end features in a convenient and affordable package in conjunction to our previously proposed design (Liwal *et al.*, 2017).

Proposed Solution: For the proposed solution, an IoT based system is used, which focuses on scalability and sustainability. In Fig. 1, a general overview of the proposed farm is provided. The IoT server can be considered as the master node for the whole operation with control over N racks or slave nodes in a single premise. Each slave node has its own sensor pack, light pack, water supply motor and a fertilizer supply motor, all integrated to better nurture the plant in that rack. From now on, we refer to a rack as the slave node and the IoT server as the master node.

Hardware Description: The IoT server has many core duties

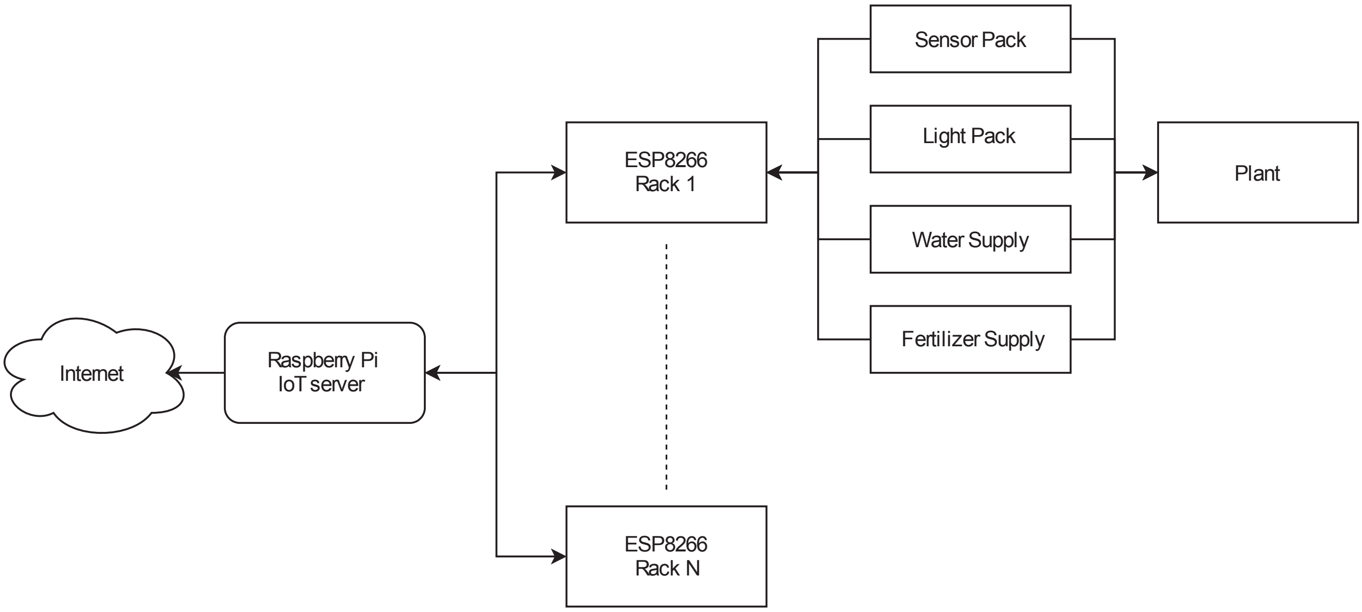

Fig. 1. General overview of the system with IoT server as the master node & ESP8266 as the slave nodes

and is the main point of access to the whole system. It provides a local web based interface which allows control of each individual node. It also requires an internet connection for security updates and remote access if required. Additionally, A WiFi hotspot is provided by the master node, so that each slave node can connect with the master node and automatically obtain an IP address through the Dynamic Host Configuration Protocol (DHCP) for seamless integration of new nodes. The master node schedules a fixed water and fertilizer supply of each node by sending the appropriate commands using HyperText Transfer Protocol (HTTP).

Secondly, the nodes themselves consist of hardware connections to many sensors including temperature, humidity and an analogue soil moisture sensor. Figs. 2 and 3 show the modules used in a single slave node. The microcontroller used here is ESP8266, which is a low-cost and low-power Wi-Fi microchip with full TCP/IP stack and microcontroller capability. The other components are WS2812B Addressable RGB LED Strip, pumps, 4-channel relay, soil moisture sensor, DHT22 sensor for temperature and humidity. The DHT22 sensor gives the normal temperature and humidity readings for the surroundings near the plant. The soil moisture sensor is used to measure the moisture level in the soil to avoid the soil from going dry. The RGB LED strip changes colors depending on the type of the plant and helps in growth of the plant. The 4-channel relay is used to power the pumps, the relay is controlled by using ESP8266. Two pumps are used, one to supply liquid fertilizer and the other to supply water. Each pump is connected to separate drip irrigation kit to conserve water and liquid fertilizer.

The ESP8266 handles all the modules connected and transmit data to Raspberry Pi to keep the system functioning. Fig. 3 shows the connection between the modules, everything is connected to the ESP8266. The water pumps are connected to ESP8266 using the relay module. The only analog reading here is from the moisture sensor. The ESP8266 acts as a hub that commands all the modules by sending the status of moisture sensor, LED Strips, DHT22 sensor and water pumps. The status of all the modules is sent to Raspberry Pi by sending it to a webpage, which evaluates data and then send instructions to the ESP8266 depending on the settings set.

Communications: The Master and slave nodes communicate using WiFi and basic HTTP requests in which each slave node only sends data when the master node requests it through an HTTP request that is sent directly to the IP of the slave node. The master node with its dashboard is accessed from http://gateway_ip/ and is usually the first IP in the subnet. The base URL of a slave node is http://node_ip/, where node_ip is the DHCP assigned IP of the node. This URL provides all sensor data as comma separated values including the status of the 3 actuators, *i.e.*, water

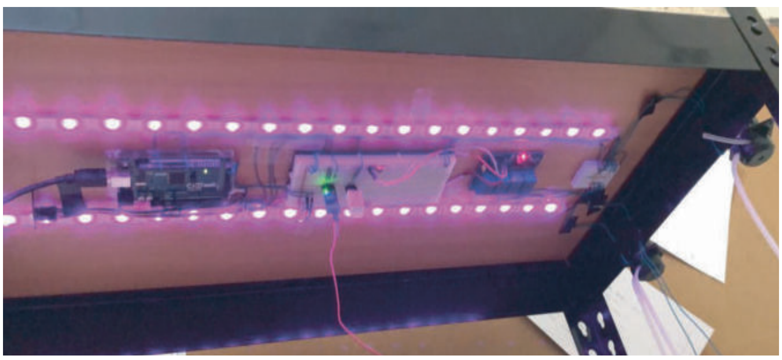

Fig. 2. A typical slave node

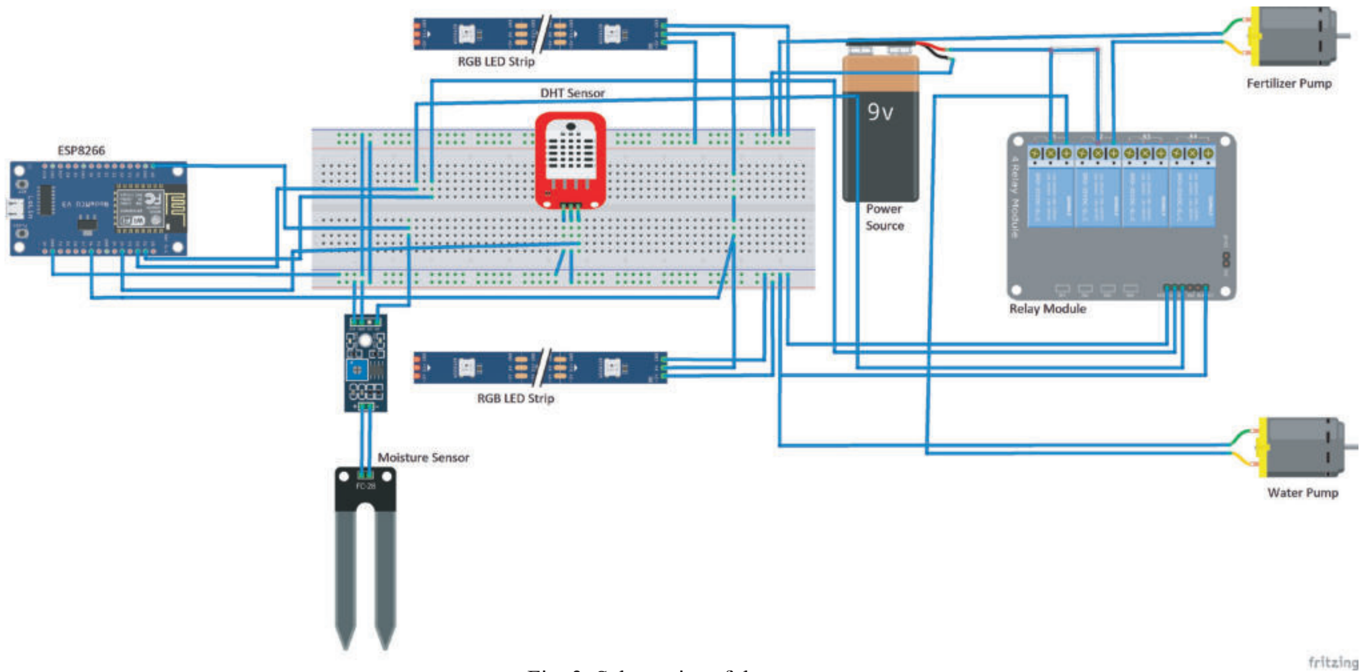

Fig. 3. Schematics of the system

supply motor, fertilizer supply motor, and the LED. To control the actuators, such as the water supply motor, slight variations in the following URL are used http://node_ip/actuator_id/ on or http://node_ip/actuator_id/ off where actuator_id in the URL is replaced with the required ID of the actuator.

As an example, to switch the LED ON in slave node 1, an HTTP request is sent to the URL https://protect-za.mimecast.com/s/gz1XCDRx8DHzWQEtWZ89l?domain=192.168.12.2 with GET request method by the master node. The node responds with a status code of 200 indicating a successful response and provides a comma-separated list such as "22.1, 140.5, 45, on, off, off". The first 3 numbers correspond to temperature, soil moisture, and humidity readings, respectively. The next three values represent the state of actuators, such as the LED, water supply motor, and fertilizer supply motor, respectively.

Software Description: The system has a number of software components which reside in both the master node and slave node. Most of the software components in the master node have dependencies on well-established third party libraries and software technologies, which include an HTTP Server with a Fast Common Gateway Interface (F-CGI) interpreter and process manager, WiFi AP library based on IEEE 802.11 access point management and IEEE 802.1X/WPA/WPA2/EAP Authenticators, a DHCP server and for storage Sqlite3 databases.

Most of the server logic lies in the CGI scripts and consists of subcomponents, such as the Login script, Dashboard, Node Commander and Master Sync. Derivative of their names, the login and dashboard provide a simple HTTP interface as shown in Fig. 4 for authenticating to the system and securely accessing the system. The Node Commander, which is accessed through the dashboard, sends manual commands to a selected node on demand, such as turning on the fertilizer or water supply if manual control is required. Two Sqlite3 databases, sensor and rack, are maintained to avoid write locks.

The dashboard, (a snapshot is shown in Fig. 5) writes configuration data for each slave node in the rack database, which is then read by the master sync script. The master sync is enabled on startup and, by using the configuration in the database for a slave node, sends HTTP commands that configure a particular node and polls it for sensor data at regular intervals,

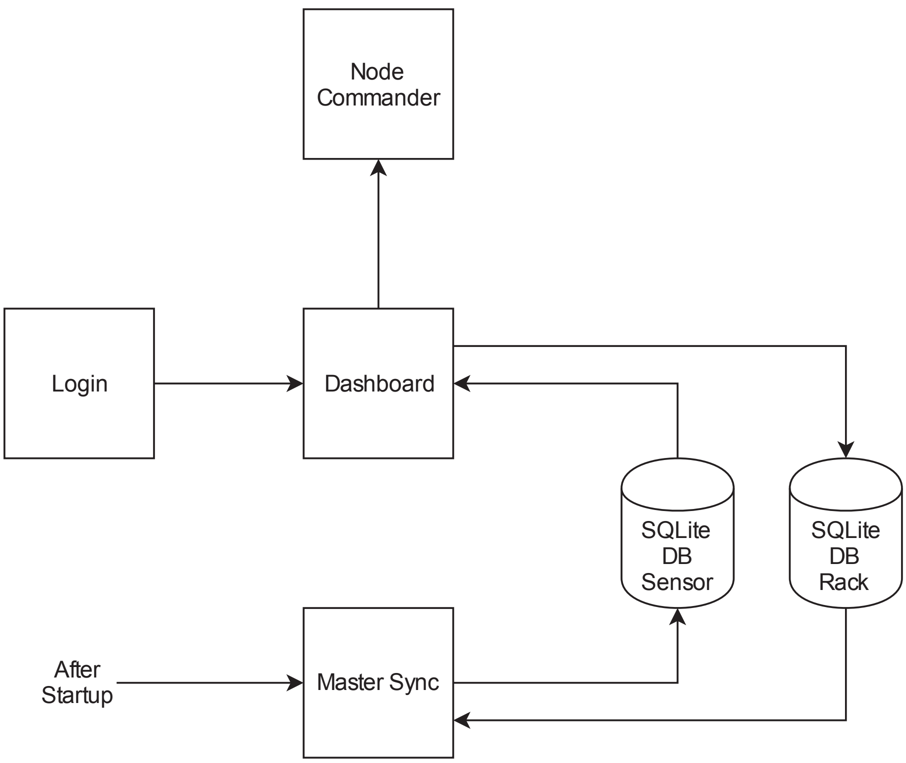

Fig. 4. Master node software general flowchart

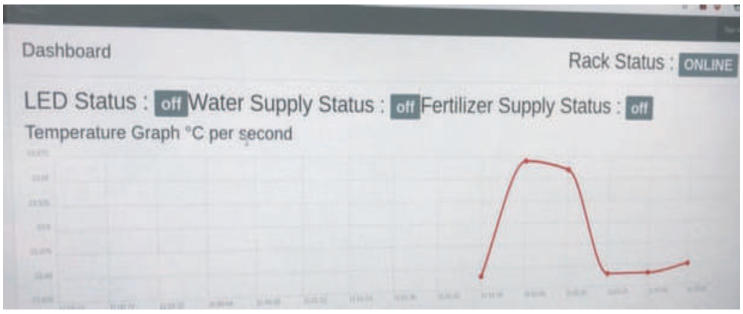

Fig. 5. HTTP Dashboard showing sensor readings over an interval of time

which is finally stored in the sensor database. The master sync also contains a scheduler for the water supply. Finally the dashboard is configured to read sensor data from the database and then plot a graph, which can be updated on re-running the dashboard script.

The master node itself holds the dashboard and hence can be accessed by any device that connects to the hotspot hosted by the master node. To remotely access the master node, it can be connected to an internet connected router by using an Ethernet cable that is configured with a port forward and Domain Name System (DNS).

Discussion

The proposed farm is very economical when compared to other solutions. First of all, ESP8266, which is used for every slave node, is the most economically viable WiFi enabled micocontroller available at the time of writing and much cheaper when compared to other solutions. Secondly, the master node only consists of a Raspberry PI which only consumes roughly 2-5 Watts when the on-board WiFi is ON. This consumption is minimal when compared to storing data in third party commercial servers that utilize power supplies rated well over 400 Watts.

Moreover, the communication protocol uses comma delimited lists to more efficiently send and receive data instead of using JSON as used in Tsai and Liang (2018). This is because the only redundant data sent is the comma. The communication protocol also avoids entities such as MQTT Brokers and consists of a simple architecture, where each slave node is simply polled by the master node, since the system functions only in the local network.

The proposed farm automatically controls environmental conditions required to grow plants in indoor vertical farms. It has major strengths in scalability and convenience, as new racks or slave nodes can be easily added without requiring any complicated wiring or configurations from the end-user. The included dashboard can be easily extended to perform complex analysis of the multiple values collected from the sensors including the use of a camera to automatically measure growth of plants. For future work, it would be beneficial to cover the rack from all sides with an insulator and include fans to control temperature and humidity. Another useful addition could be the use of solar panels to power the farm.

References

Al-Chalabi, M. 2015. Vertical farming: Skyscraper sustainability?. *Sustainable Cities and Society*, 18: 74-77. doi: 10.1016/j.scs.2015.06.003.

Belista, F.C.L., M.P.C. Go, L.L. Luceñara, C.J.G. Policarpio, X.J.M. Tan and R.G. Baldovino, 2018. A smart aeroponic tailored for IoT vertical agriculture using network connected modular environmental chambers. *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, Baguio City, 2018, p. 1-4.

Despommier, D. 2009. The rise of vertical farms. *Scientific American*, 301(5): 80-87. doi: 10.1038/scientificamerican1109-80.

Despommier, D. 2013. Farming up the city: the rise of urban vertical farms. *Trends In Biotechnology*, 31(7): 388-389. doi: 10.1016/j.tibtech.2013.03.008.

Ismail, M.I.H. and N.M. Thamrin, 2017. IoT implementation for indoor vertical farming watering system. *2017 International Conference on Electrical, Electronics and System Engineering (ICEESE)*, Kanazawa, 2017, p. 89-94.

Kalantari, F., O. Tahir, R. Joni and E. Fatemi, 2018. Opportunities and Challenges in Sustainability of Vertical Farming: A Review. *Journal Of Landscape Ecology*, 11(1): 35-60. doi: 10.1515/jlecol-2017-0016.

Liwal, K.K., C. Copiaco, M. Vohra, H. Rahman, N. Abdulaziz and M.F. Malek, 2017. Optimizing communications and automation in micro farms. *International Journal of Latest Research in Engineering and Technology,* 03(09): 97-101.

Olle, M. and A. Viršile, 2013. The effects of light-emitting diode lighting on greenhouse plant growth and quality. *Agricultural And Food Science*, 22(2): 223-234. doi: 10.23986/afsci.7897.

Ruengittinun, S., S. Phongsamsuan and P. Sureeratanakorn, 2017. Applied internet of thing for smart hydroponic farming ecosystem (HFE). *10th International Conference on Ubi-media Computing and Workshops (Ubi-Media),* Pattaya, 2017, p. 1-4.

Stevens, J.D. and T. Shaikh, 2018. MicroCEA: Developing a Personal Urban Smart Farming Device. *2018 2nd International Conference on Smart Grid and Smart Cities (ICSGSC)*, Kuala Lumpur, 2018, p. 49-56.

Touliatos, D., I. Dodd and M. McAinsh, 2016. Vertical farming increases lettuce yield per unit area compared to conventional horizontal hydroponics. *Food and Energy Security*, 5(3): 184-191. doi: 10.1002/fes3.83.

UN Officials, 2019. Technology brings positive change, but ‘collateral damage’ must be minimized: senior UN official. https://news.un.org/en/story/2019/08/1045071.

Tsai, C. and T. Liang, 2018. Application of IoT technology in the simple micro-farming environmental monitoring. *IEEE International Conference on Advanced Manufacturing (ICAM)*, Yunlin, 2018, p. 170-172.

Received: December, 2019; Revised: December, 2019; Accepted: December, 2019