A Hydroponic Planter System to enable an Urban Agriculture Service Industry

Akashi Satoh†
†Department of Communication Engineering and Informatics
The University of Electro-Communications, Chofu, Tokyo, Japan

***Abstract*— We have developed a compact hydroponic planter and are conducting cultivation experiments to realize an agricultural service industry that uses vacant spaces in urban areas. In order to provide a full-fledged cultivation method for fruit and vegetables to beginners in agriculture, a low-cost sensor module and a planter with an integrated remote-control system have been developed. MQTT, a lightweight protocol for IoT is used to monitor sensor data and to control a pump. Data is securely encrypted using TLS/SSL. In addition to monitoring sensor data, a USB-connected camera-based still-image photography function with motion detection is included. We are also enhancing the interfaces to support AI speakers as well as smartphones and PCs.**

Keywords—smart agriculture, hydroponic culture, sensor module, Arduino, Raspberry Pi

I. Introduction

We have proposed a new consumer-level urban agriculture methodology by developing and introducing hydroponic systems for use on rooftops and verandas, as opposed to traditional agriculture which cultivates the land for production of crops [1][3]. The systems can be used at office buildings, hospitals, and elementary schools to provide an oasis, a relaxing space, and a food education program. We also aim to create new sixth industries by not only linking producers and consumers, but by turning the consumers into the producers.

In this recreational urban agriculture, an enhancement of crop productivity is not important. However, some support systems for cultivation are required because the main users of our hydroponic systems are beginners in agriculture. It is not realistic for a service provider to check each system scattered in a city directly every day. A remote management system using IoT technologies, such as an agricultural sensor system for a large vegetable factory is an effective solution, but it is too expensive for a small facility in a city area or for personal use. The sensor system requires high accuracy and high reliability for optimal control of cultivation environment, which results in a costly product.

In order to provide a hydroponic system that enables anyone to casually enjoy cultivation, we have developed monitor and control system from low-end, high-performance microcomputer boards (Arduino and Raspberry Pi) and open source software. Our system is not for mass production, but for recreation where optimal control with high accuracy sensors is not required. Therefore our specialized sensor module for hydroponic cultivation was able to achieve low cost through the use of a simplified its circuit structure. In this paper, the hardware and software architectures of our hydroponic culture system are described.

II. Compact Hydroponic Planter

Fig. 1 shows our compact hydroponic planter and its basic structure, which is based on a vertical pipe for suspended cultivation [3]. The simple structure facilitates maintenance repair and installation, while full-fledged cultivation of fruit and vegetables can be enjoyed. Water in a tank is pumped up into two vertical pipes and is showered over the roots of plants put into the pipes. With no obstacles the roots extend freely in the air, and are free from root rot, which is common in hydroponic cultivation during the summer. Fig. 2 shows tomato cultivation using the compact planter on the rooftop. Limiting water is important to increase the sugar content of the tomatoes, and is easily achieved by controlling the pumping time.

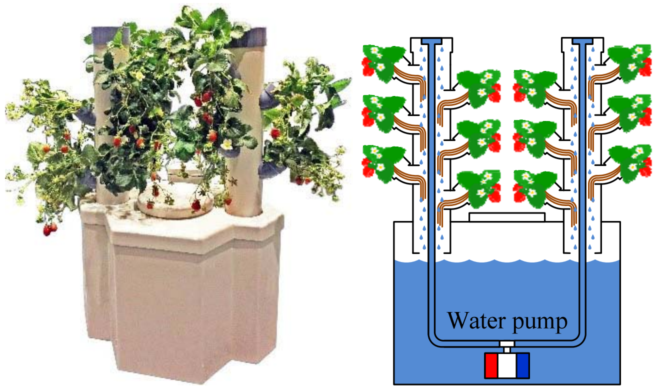

Fig. 1 A compact hydroponic planter for cultivating strawberries and its basic structure

Fig. 2 Rooftop cultivation of tomatoes.

III. Sensor Module

Fig. 3 shows our sensor module for the hydroponic planter. Liquid fertilizer concentration (based on electrical conductivity (EC)), water level, water temperature, luminance, temperature, and humidity are measured by attaching various sensor devices to the terminal connectors on the module. The water pump and a fertilizer control equipment are driven by two relay switches. Fig. 4 shows the compact planter equipped with the sensor module. The thin, green stick in the tank is a printed wiring board with electrodes to measure the EC value and water level. The water circulation pump with hoses and a ball tap for automatic water supply are placed at the center and on the left, respectively.

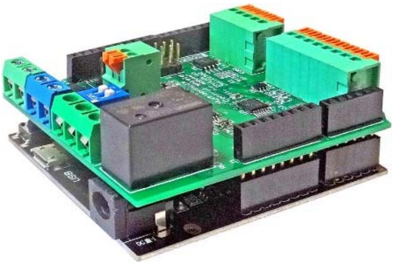

Fig. 3 A sensor module for the hydroponic culture system stacked on an Arduino-compatible board.

Fig. 4 A compact planter with sensor module.

Fig. 5 is an oscillator circuit for EC measurement whose frequency varies with solution concentration [2][4]. The frequency is captured by the frequency counter on the Arduino ATmega328 processor. In order to prevent buildup on the electrodes the circuit is active only during measurement. Even when deactivated, leakage DC current will flow if a voltage difference occurs between any of the electrodes in the same water tank. Therefore, a 3-state buffer with a high-impedance mode was inserted to block the DC flow.

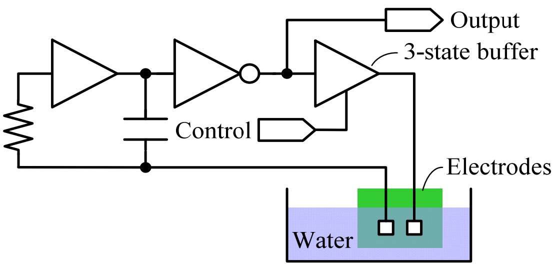

Fig. 5 EC measurement circuit.

Fig. 6 is the water level measurement circuit, where a 3.3-V source voltage is divided by nine serial resisters. The number of resisters connected to the GND voltage is changed with the water level, and the output voltage level is changed between 0 when all resistors are connected to GND and 3.3V when all resisters are open. The voltage is measured by a 10-bit AD converter on the ATmega328 processor. This circuit also has 3-state buffers to prevent DC current flow in the water when deactivated.

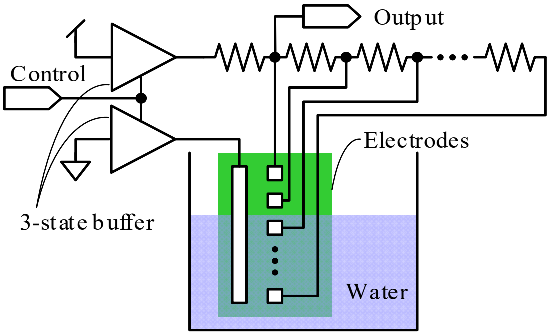

Fig. 6 Water level measurement circuit.

The water temperature and the luminance are measured based on the change of resistance of a thermistor and a photo diode, respectively. A low-cost sensor device [5] is used to measure atmospheric temperature and humidity. In order to monitor the status of the water pump and other devices connected to the sensor module, current monitors are attached to the two relays, which include a mechanical switch and a power MOSFET. The sensor module supports Arduino and other compatible boards as its main board, and we mainly use an Arduino-compatible board equipped with a micro controller, ESP-WROOM-32 [6]. The sensor data are sent to the Arduino-compatible board thorough the serial interface, and are transferred to a server (Raspberry Pi) by using the ESP-WROOM-32's Wi-Fi. In order to reduce sensor cost, the circuits have simple designs for just measuring frequency and voltage data, and thus the data needs to be converted to EC, water level and temperature. As described in the next section, this conversion is performed by Raspberry Pi, but not by Arduino.

IV. Remote Management System

The sensor data are transferred to the Raspberry Pi, which acts as a sever, over Wi-Fi. The data can be monitored by portable devices such as smartphones and tablet PCs from a remote location. One simple solution for this system is that of installing a Web server on the Raspberry Pi with a global IP address and transferring the data using the http protocol. If the Raspberry Pi and the sensor module are not place in the same LAN, this solution requires to assign another IP address to the module for the remote pump control.

Therefore, the MQTT (Message Queuing Telemetry Transport) protocol that does not require a global IP address to be assigned to the client is used for bidirectional communication. MQTT, supporting just the minimum functions for IoT devices, is lighter than HTTP, and thus a low-price data-only communication service without a global IP, such as MVNO, can be used. MQTT is more suitable for real-time operation than HTTP, because MQTT keeps the TCP session connected once it is established, and thus no hand shake protocol is required before each data transfer between server and client. MQTT supports QoS, and three levels 0-2 can be used depending on importance of data, such as the highest level 3 for the pump control and the lowest level 0 for temperature that does not change drastically.

MQTT does not support security functionality, such as encryption. However, cultivation environments of the planter placed at a private house, such as temperature, humidity, and illuminance, are private information, and the pump control should be prevented from being controlled by a malicious third party. Therefore, TLS/SSL is integrated to encrypt MQTT packets.

Fig. 7 shows an example network structure of the hydroponic system. A MQTT client library is installed on the Arduino, and the sensor data are "published" through a Wi-Fi router. A user can access the Raspberry Pi which acts as a "Broker" (MQTT server) using a smartphone and PC to monitor the sensor data, and to send control data to Arduino which acts as a "Subscriber" . Node-RED [7] installed on Raspberry Pi is used as a platform to monitor and control data. Node-RED is a flow-based programing environment developed by using Node.js, and provides various applications such as IoT and Web services by connecting Node modules as shown in Figs. 8 and 9.

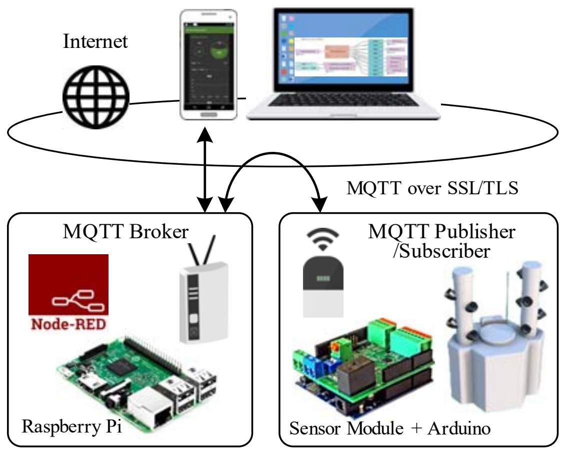

Fig. 7 An example network structure of the hydroponic system.

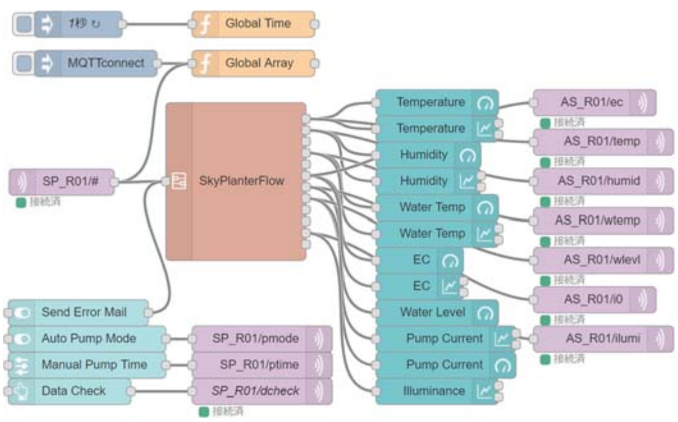

Fig. 8 A main flow graph of the hydroponic system.

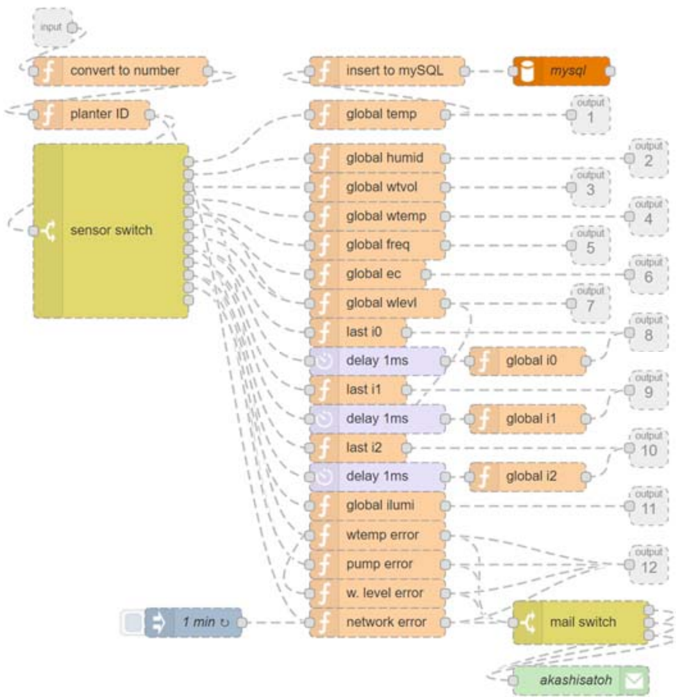

Fig. 9 Sensor data operation flow of the hydroponic system.

Dashboard [8], one of Node-RED libraries, is used to provide a GUI, and real-time data monitoring and pump control are performed on the screen shown in Fig. 10. The graph data on the Dashboard screen scrolls with time. The data is stored in the Raspberry Pi server by using the open source database software MySQL [9] to make them analyzable afterwards. When abnormal situations, such as the water being empty, pump failure, and network disturbance over a long period occur a warning e-mail is sent to a pre-selected address.

Fig. 10 Dashboard for dat monitoring and pump control.

In remote cultivation, monitoring image data as well as sensor data is important. It is very hard to send hi-resolution video data in our system, because a low-cost low-bandwidth network service is used in our system to enable easy enjoyment. However, plants do not grow rapidly over a short period, and sending one still image every several minutes to several tens of minutes is sufficient. On the other hand, it is not favorable if accidents happening during the interval go unnoticed. In order to cope with this issue, motion detection by an inter-frame difference calculation is implemented, and if significant movement is detected during the regular interval, an additional image is taken (Fig. 11).

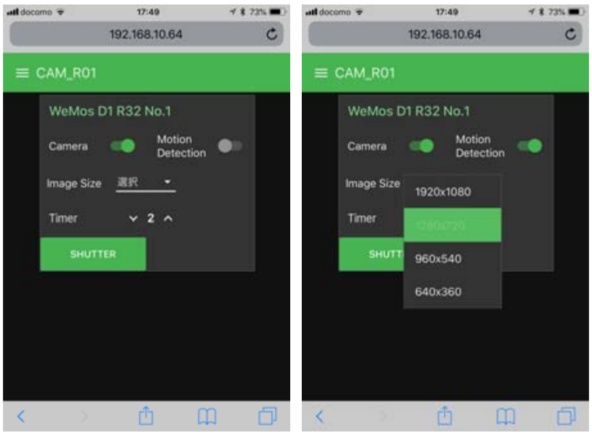

Fig. 11 Camera control menu.

Motion detection and JPEG compression are rather CPU-heavy operations for Arduino, and thus are performed by the Raspberry Pi with a general USB camera or an specialized camera module. A Raspberry Pi 3 equipped with a 64-bit processor is used for the MQTT server, but performance of the low-cost Raspberry Pi Zero is also good enough for the image processing. Currently, captured images are time stamped and uploaded to a cloud server to be accessed from around the world.

Many pictures are uploaded when the interval time is short or many motions are detected, and it is difficult to check each image one by one. Therefore, a method of compressing still images to a time lapse movie and playing it on a web browser is being implemented. The movie generation can be executed on the server, or the Raspberry Pi with the camera can create and transfer the movie to the server at night when the camera is not used. In the latter case, it is required to set up a Web server with a global IP address on the Raspberry Pi to check the video during the daytime. Therefore, we are also considering using a free DDNS (Dynamic Domain Name System) service that connects a host name to a dynamic IP address.

An AI speaker as well as a smartphone and PC are introduced as an interface of the hydroponic system.

A microphone, a speaker, and LED button of AIY Voice Kit [10] are added to the server Raspberry Pi to form the AI speaker, and Japanese speech recognition and voice synthesis are performed by Julius [11] and Open JTalk [12], respectively. Currently, basic management functions are implemented such as reading sensor data and notifying about abnormal states. We plan to enrich these functions not only for hydroponic cultivation but also for the user's day-to-day lives.

V. Conclusion

We developed a sensor module for a hydroponic cultivation system to expand a new type of enjoyable agriculture into urban areas as a service industry. In order to reduce costs of the system, general purpose microcomputer boards, Arduino and Raspberry Pi, are used as an MQTT client and server, and a remote management system was built using open source libraries and low-bandwidth communication services. We are also developing various service businesses for urban smart agriculture, and expect to report on them in near future.

References

[1] Satoh Laboratory, "Urban Smart Agriculture," http://satoh.cs.uec.ac.jp/en/research/hydroponics/index.html

[2] T. Nishimura, Y. Okuyama, A. Matsushita, H. Ikeda, and A. Satoh, "A Compact Hardware Design of a Sensor Module for Hydroponics," IEEE 5th Global Conference on Consumer Electronics (GCCE2017), OS-ICE(1)-4, Oct. 2017.

[3] Ueno Nursery Ltd. , "Easy and Enjoyable Farming" http://www.uenoengei.com/eng_rakuraku.html

[4] T. Nishimura, Y. Okuyama, and A. Satoh, "High-Accuracy and Low-Cost Sensor Module for Hydroponic Culture System," IEEE 4th Global Conference on Consumer Electronics (GCCE2016), OS-IOT-6, Oct. 2016.

[5] Adafruit: DHT11, DHT22 and AM2302 Sensors. https://learn.adafruit.com/dht/overview

[6] Espressif Systems: ESP-WROOM-32 https://www.espressif.com/en/producttype/esp-wroom-32

[7] Foundation: Node-RED - Flow-based programming for the Internet of Things. https://nodered.org/

[8] node-red-dashboard. https://flows.nodered.org/node/node-red-dashboard

[9] MySQL. https://www.mysql.com/

[10] AIY Voice Kit. https://aiyprojects.withgoogle.com/voice

[11] Julius. http://julius.osdn.jp/

[12] Open JTALK. http://open-jtalk.sourceforge.net