IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received May 9, 2018, accepted June 11, 2018, date of publication June 14, 2018, date of current version July 6, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2847440

Active Plant Wall for Green Indoor Climate Based on Cloud and Internet of Things

YU LIU[1], KAHIN AKRAM HASSAN[1], MAGNUS KARLSSON[1], OLA WEISTER[2], AND SHAOFANG GONG[1], (Member, IEEE)
[1]Department of Science and Technology, LiU—Campus Norrköping, 60174 Norrköping, Sweden
[2]Vertical Plants System Sweden AB, 60002 Norrköping, Sweden

Corresponding author: Yu Liu (yu.a.liu@liu.se)

This work was supported in part by The Swedish Environmental Protection Agency and in part by the Norrköping Fund for Research and Development in Sweden.

ABSTRACT An indoor climate is closely related to human health, well-being, and comfort. Thus, indoor climate monitoring and management are prevalent in many places, from public offices to residential houses. Our previous research has shown that an active plant wall system can effectively reduce the concentrations of particulate matter and volatile organic compounds and stabilize the carbon dioxide concentration in an indoor environment. However, regular plant care is restricted by geography and can be costly in terms of time and money, which poses a significant challenge to the widespread deployment of plant walls. In this paper, we propose a remote monitoring and control system that is specific to the plant walls. The system utilizes the Internet of Things technology and the Azure public cloud platform to automate the management procedure, improve the scalability, enhance user experiences of plant walls, and contribute to a green indoor climate.

INDEX TERMS Internet of Things, Azure cloud, IoT hub, remote monitoring and control, plant wall system.

I. INTRODUCTION

The term indoor climate reflects the thermal, atmospheric, acoustic, actinic and mechanical environments of residential, commercial or public buildings, according to the World Health Organization (WHO). The temperature, humidity, air quality and lighting conditions of an indoor space are highly related to human health, comfort perception, well-being and work productivity [1]. Numerous studies have been conducted to reveal negative influences on human-beings who live in contaminated indoor environments [2]. Therefore, the quality of an indoor climate has posed important concerns for people's lives and health. Many solutions and products have been developed in commercial markets to improve indoor climates, among which the vertical plant wall system is a prominent representative [3].

A plant wall system involves growing diverse types of green plants on a vertically supported system that is attached to an internal or external wall or is designed as a standalone product. A plant wall system consists of vegetation, growing medium and irrigation and drainage systems [4]. In addition to the initial aesthetic decoration, green plants make significant contributions to indoor environments via evaporation, air purification and water retention, which improves indoor climates and reduce energy use [5], [6]. Our previous research has proved that an active plant wall, i.e., a plant wall with an integrated fan to accelerate ventilation, can effectively reduce the main pollutants, such as particulate matter (PM) and volatile organic compounds (VOC), and stabilize CO_2 concentrations to a healthy level [3]. Plant walls have been deployed in public spaces, such as universities, airports, museums and offices. However, several constraints hinder plant walls from entering the household market. Regular management of massive plant walls is costly and time-consuming for plant wall suppliers, because vegetation on a plant wall needs professional knowledge of proper plant care. Regular service of plants is also limited by geography because suppliers are unable to serve all consumers worldwide. Thus, reliable and affordable remote management and monitoring solutions for plant walls are in high demand, which is the motivation to this study. In this type of systems, several key factors of an indoor climate, e.g., temperature, relative humidity (RH), CO_2 level, and VOC concentration, must be collected and monitored by both consumers and suppliers in real time. By analyzing collected data, suppliers with expertise in planting are able to provide professional feedback to consumers and remotely adjust the watering, lighting and ventilation functions to maintain thriving plants and guarantee a healthy indoor climate.

The evolving Internet of Things (IoT) and cloud technologies have become the key enablers to the digitalization of many traditional applications. A number of theoretical and analytic researches have been conducted to promote IoT and cloud technologies into industrial applications [7], [8]. We foresee that a remote management and monitoring system that is based on an IoT and cloud platform can be a viable solution to address the widespread deployment challenge of plant walls. From the integration of sensor networks and cloud infrastructures, remote monitoring solutions that are based on cloud services have been investigated in a broad range of fields, such as industrial monitoring [9], health care [10], [11] and pain monitoring [12], traffic monitoring [13], car parking [14], agricultural irrigation [15] and environment monitoring [16].

In this paper, we propose and implement a remote monitoring and management solution that is specific to a plant wall system based on the Azure public cloud platform and is aimed at contributing to the indoor climate monitoring and control in public or private buildings. The proposed system consists of both a local control part and a cloud service part. In the local part, a series of environmental parameters are monitored to perceive the indoor climate. The data are continuously fetched and sent to the cloud using the WiFi protocol to ensure security and availability. In our solution, the control functions of watering, lighting and ventilation in a plant wall system are considered. These functions are directly controlled by a local microprocessor according to pre-defined settings that are locally stored and remotely synchronized with the cloud. The cloud part takes advantage of the IoT Hub infrastructure and other services, such as functions, storages and web visualization offered by the Azure platform. Via a web-based user interface, administrators and end users are able to monitor an indoor climate in real time, check historic data from a database, and update the schedules of the pump, light and fan functions, as well as invoke actuators for management purposes. The proposed solution utilizes the Azure public cloud platform to address challenges in terms of security, reliability, scalability and cost. The contributions of this study are as follows:

- Proposed a complete solution for remote monitoring and management of plant wall systems.
- Implemented a local autonomous control unit for plant walls, which consists of both sensing and actuating functions.
- Developed a cloud solution that is based on the Azure cloud platform, including both back-end and front-end for data storage, real-time monitoring and historic data visualization.
- Realized remote management and control functions in both the local unit and the cloud via the IoT Hub infrastructure.

The remainder of this paper is structured as follows: Section II reviews currently existing solutions in the literature and analyzes system requirements. Section III provides a brief introduction to the public cloud platform, as exemplified by the Azure cloud. Section IV presents the architecture of the entire system, from the local part to the cloud server part. Section V describes the detailed system design, including the hardware and software solutions in the local part and the structures and services in the cloud. In section VI, we demonstrate our implementation and present the results of the proposed solution. The last section concludes the paper.

II. RELATED WORK

The motivation of this work originates from a practical challenge confronted by the plant wall industry. To tackle the long-existing pain suffered by plant wall suppliers, a remote management and monitoring solution that helps to accelerate the widespread deployment of plant walls is highly demanded.

Many efforts have been put into the research field of IoT-based monitoring solutions. In [28], a cloud-based environmental motion monitoring architecture was proposed, and the key components in the architecture were analyzed in theory. The work in [17]–[21] and [24]–[27] investigated ZigBee-based wireless sensor networks for monitoring of indoor air quality. Salamone *et al.* [19] proposed a control system for improving indoor air quality and lighting quality, in which CO_2 and illumination factors were collected and processed to control the air exchange system and the lighting system. As an isolated system, there is no cloud infrastructure implemented in this design. Therefore, data record, real-time monitoring or control functions are not supported. Bhattacharya *et al.* [20] collected distributed ZigBee sensor data with a sink device that was connected to a local computer via a USB port, and stored those sensor data to a local sensor database. A GUI application running on the local computer was designed to visualize categorized sensor data. A framework to integrate the control of heating, ventilation, and air conditioning (HVAC) system based on collected environmental data was also proposed but not implemented. Marques and Pitarma [17] and Pitarma *et al.* [18] developed an indoor air quality monitoring system utilizing ZigBee sensor networks. In this implementation, environmental data are continuously transmitted to a ZigBee gateway and then relayed to a private server using Restful web services where a MySQL database is deployed. A web and an Android App were developed to enable access to visualized sensor data and notifications. A similar solution was exploited by Yang *et al.* [24] by expanding the complexity of the private cloud server. The authors developed a cloud computing application named iEDMS in OpenStack, created a distributed computing environment based on Hadoop and processed environment data in HBase, which improved the reliability of the cloud server.

Solutions that utilize commercial visualization service are proposed in [21] and [23]. The work presented in [21] proposed a remote monitoring solution for winery and creamery environments. A series of environmental sensors, such as temperature, RH, VOC, CO_2 and particle sensors, are deployed. The sensor data are periodically read by a microcontroller and transmitted to a remote data server. The system relies on the IEEE 802.15.4 standard; therefore, an Ethernet gateway is needed to bridge between IEEE 802.15.4 and the Internet protocol (IP). All sensor data are graphically displayed via a commercial visualization platform named PI on a webpage that can be accessed from any web-browser-enabled device. Another design is presented in [23]; the authors proposed an indoor monitoring solution that is based on the Yeelink platform. They adopted a Linklt One board as a local hardware platform to collect sensed temperature, humidity, light intensity and volume fraction of the dust data and send them into the Yeelink cloud server via WiFi. The end users and administrators can check the indoor status with the Yeelink cloud service.

Remote monitoring solutions applied to healthcare are also referenced. Yang *et al.* [12] proposed an IoT and cloud-based remote pain monitoring system. In their implementation, a sensor node is embedded in a facial mask to monitor pain intensities of patients. The facial biosignal is locally sampled, digitalized and processed. A WiFi module is integrated to transmit sensor data to the remote cloud via a gateway using the User Datagram Protocol (UDP) or Transmission Control Protocol (TCP). In the cloud part, data are forwarded to a data channel that is connected to a cloud database for further signal processing and data mining. A mobile web application is built on an open source framework to provide caregivers access to streamed data and perform real-time analytics. Zhiqiang *et al.* [22] chose Bluetooth as the protocol to acquire medical equipment data through a mobile phone, and transmitted those data into the Ali public cloud platform via WiFi. A rich Internet application (RIA) was developed as a web client to query and visualize stored data from the Ali cloud database.

However, none of aforementioned solutions can provide remote management and control functions that are significantly important to a plant wall system. Our group has been investigating IoT and cloud based remote control solutions since early 2004. In the previous work [25]–[27], our group developed a remote climate control system for cultural buildings in Sweden, such as churches and museums. In this system, a wireless sensor network that is built on the ZigBee protocol is utilized to fetch temperature and humidity data, and heating is automatically controlled with an actuator according to the sensor feedback. The sensor readings are continuously sent to and stored in a local server via the ZigBee coordinator. This local server is periodically synchronized to a central main server via the Internet, which enables users and administrators to visualize sensed data and to change control settings from a website.

These previous studies have achieved their specific goals but cannot be applied to the case of plant walls due to three reasons. First, in [17]–[22] and [24]–[27], wireless sensor networks that is based on ZigBee, IEEE 802.15.4 or Bluetooth are exploited to satisfy the coverage and low power consumption requirement. However, in a plant wall system, low power consumption of sensors and microcontrollers is not critical at all, as compared to that for pumping, lighting and ventilation. In addition, the redundant gateway needed in these systems not only increases the complexity of the entire system but also greatly increases the deployment cost. Second, in [12], [17]–[21], [23], and [24], only uplink to the cloud is considered; downlink to the device is not considered, i.e., people can only passively monitor the status of actuators but cannot adjust or manage them, which is not suitable in a plant wall scenario. This is obvious, since the plantcare through control of watering, lighting and ventilation is as essential as the monitoring function to plant wall suppliers and customers. Third, the majority of existing solutions [12], [17], [18], [20], [24]–[27] are developed on private self-maintained cloud servers. The hardware and software development and maintenance cost is intolerable to small and medium-sized enterprises. The challenges in terms of throughput, security, scalability and reliability can incur critical issues when they are employed in commercial plant wall products. In [22] and [23], commercial visualization services are adopted. These single functional platforms are lack of flexibility to customize the required functions. Some key services such as device management, provisioning, application hosting, machine learning, analytics, etc. are also missing. The scalability is also geographically limited compared to big public cloud platforms. Therefore, in this work we aim to address these shortcomings by our own solution that is based on the commonly used WiFi protocol and the Azure public cloud platform. A comparison of our work to previous solutions is shown in Table 1. It is seen that this work aims to provide a complete remote monitoring and control solution requiring high reliability, scalability, easy deployment and, last but not least, low cost.

III. PUBLIC CLOUD PLATFORM

A. BENEFITS

The rapid advancement of cloud technologies has enabled the widespread deployment of cloud-based applications and services, whereas the gradually maturing commercialization of public clouds is enabling society to share the merits of cloud computing. The trend of commercial companies, public organizations, universities and private persons is to migrate their applications, services and data storages to public cloud platforms, such as the Azure cloud, Amazon cloud service, IBM Bluemix or Google cloud platform [29]. The integration with IoT is an inevitable feature of any public cloud platform, due to the emerging IoT era and the rapid annual growth of connected devices. Employing public cloud platforms instead of developing a private cloud in the IoT scenario has substantial benefits. A cloud platform is composed of diverse services, including services that are extensively employed in IoT applications such as device management, data storage, processing, analytics and visualization. With the aid of a public cloud platform, people are awarded the

TABLE 1. Review of remote monitoring solutions.

Solution	Field	Protocol	IP-Support	Gateway /Adapter	Monitoring Function	Offline Control	Command Control	Parameter Management	Cloud	User Interface	Alarm	Scalability
[17], [18]	Indoor Air Monitoring	ZigBee	No	Required	Yes	No	No	No	Private	Web&App-based	Yes	Low
[19]	Indoor Air Control	ZigBee	No	No	No	Yes	No	No	No	No	No	Low
[20]	Indoor Air Monitoring	ZigBee	No	No	Yes	No	No	No	Private	App-based	No	Low
[21]	Indoor Air Monitoring	ZigBee	No	Required	Yes	No	No	No	Public (PI)	Web-based	Yes	Limited
[12]	Pain Monitoring	WiFi	Yes	Not Required	Yes	No	No	No	Private	Web-based	No	Low
[22]	Mobile Healthcare	Bluetooth &WiFi	Yes	No	Yes	No	No	No	Public (Ali Cloud)	Web-based	No	High
[23]	Indoor Air Monitoring	WiFi	Yes	Not Required	Yes	No	No	No	Public (Yeelink)	App-based	No	Limited
[24]	Indoor Air Monitoring	ZigBee	No	Required	Yes	No	No	No	Private	Web-based	Yes	Low
[25]–[27]	Remote Climate Control	ZigBee	No	Required	Yes	Yes	No	Yes	Private	Web-based	No	Low
This work	Indoor Climate Monitoring &Control	WiFi	Yes	Not Required	Yes	Yes	Yes	Yes	Public (Azure)	Web-based, optimized for mobile device	Yes	High

flexibility to combine demanded services while avoiding the necessity to build a private platform. Benefiting from the massive deployment of hardware, the total cost of utilizing a public cloud is lower than the cost of implementing their own infrastructure from scratch. The professional knowledge and experiences of public cloud suppliers also contribute to the quality of services (QoS). The widespread use of data centers throughout the world enables people to freely expand their services and businesses worldwide without concerns about the scalability. Necessary redundancy of public cloud platforms guarantees high reliability since disaster recovery is usually a fundamental promise. In addition, in IoT use cases, security is always a prioritized concern because sensitive data leakage may incur severe consequences. The security aspect is another merit highlighted by the public cloud platform throughout the entire stack, including software, infrastructure, storage and network [30]. Considering the substantial investment in securing the infrastructures, timely upgrade to software and hardware, and accumulated expertise in dealing with myriad attacks with increased sophistication, the public cloud platform is more trustworthy than self-maintained data centers in terms of security and timely service upgrading.

Based on these merits, we consider the public cloud platform Microsoft Azure to build our remote monitoring and management solution for a plant wall system.

B. AZURE CLOUD PLATFORM

The Azure cloud is one of the most promising cloud computing service platforms; it offers services such as networking, application, computing, storage, container, analytics and artificial intelligence via distributed data centers throughout the world. IoT Hub is the infrastructure integrated in Azure; it handles the daily growing needs in the IoT scenario, which guarantees reliable and secure bidirectional communications between millions of IoT devices and solution back-ends [31].

Azure supports abundant IoT devices that are connected to and managed by IoT Hub, including native IP-enabled devices, existing IoT devices and low-power devices. IP-capable devices can directly establish a secure link with IoT Hub via the Internet, whereas other devices with different protocols (such as ZigBee) or resource-constrained devices are provided with a field gateway, namely, Azure IoT Edge, to perform protocol translation and data relay. A vast variety of communication protocols are supported by IoT Hub, such as the HTTPS, MQTT and AMQP protocols. In addition to these protocols, customized protocols are also supported by the field gateway [32].

A cloud gateway sits in the front of IoT Hub and performs management of data input and output, device authentication and authorization using information stored in the IoT Hub device identity store, where access rules, keys, and metadata are reserved. The core concept of IoT Hub is referred to device twins, which sit in the center of IoT Hub. It is a JSON format document with recorded device properties. IoT Hub manages devices by enforcing remote devices to synchronize desired properties and update device states to their device twins. For cloud to device (C2D) communication, IoT Hub enables C2D messages to send notifications and a direct method to order commands. For device to cloud (D2C) communication, messages are supported to send device telemetry to the cloud [31]. These natively supported communication mechanisms can offer significant assistance to users.

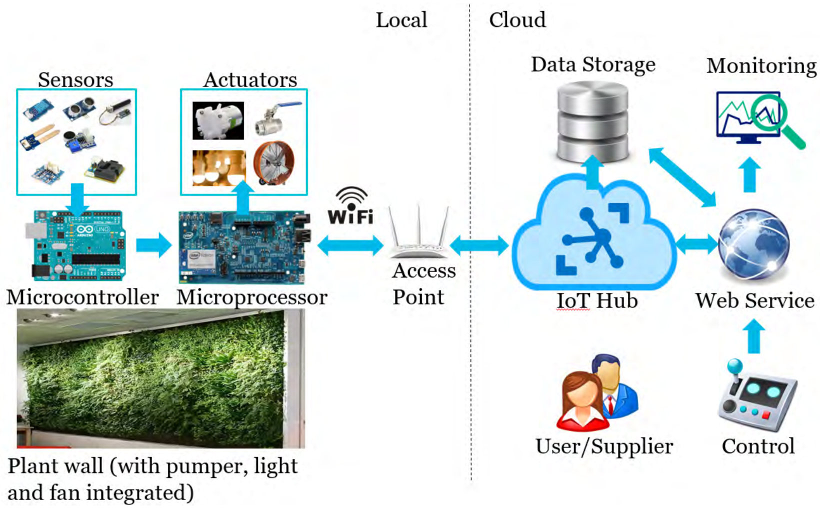


FIGURE 1. System overview of Azure cloud and IoT-based remote monitoring and management system for plant walls.

IoT Hub can be seamlessly integrated into other services in the Azure platform, such as data storage, analytics, web service, and machine learning. This integration offers users with sufficient freedom to customize their cloud applications according to real needs.

Azure IoT Hub and other cloud services can satisfy the reliability, scalability, security and cost requirements in most IoT use cases.

IV. SYSTEM OVERVIEW

An overview of the proposed remote monitoring and management solution for plant walls is shown in Fig. 1. The system is split into two parts, namely, the local monitor and control part, and the cloud server part. The local part is composed of a microprocessor, a microcontroller, sensors and actuators. Environmental sensors, including temperature, humidity, PM, CO_2, light and multichannel gas sensors are connected to and sampled by the microcontroller. The sensor readings are periodically sent to the microprocessor by query. A microcontroller is selected for sensor reading due to its deterministic timing, which is compatible with sensors, whereas a microprocessor with non-deterministic timing and an operating system is employed to execute complicated tasks. Separating the duty of the microcontroller and microprocessor also improves the robustness of the system.

Actuators, i.e., the water pump, the light and the fan installed in the plant wall, are under autonomous control of the microprocessor according to defined time schedules. These settings are locally stored as device properties and have an identical copy recorded in the device twin in IoT Hub. By updating the device twin, administrators are capable of managing the schedule and controlling the watering, lighting and ventilation systems in the plant wall. A WiFi module is integrated with the microprocessor chip to enable communication with the remote cloud via any available access point. Benefiting from the long term development and widespread adoption, the WiFi protocol has become a low-cost, mature and secure solution in indoor communications. Compared to other commonly exploited wireless communication protocols such as Bluetooth, ZigBee, and Z-Wave, etc., WiFi has native support to Internet Protocol (IP), which greatly takes advantage of existing Internet infrastructures of modern buildings instead of deploying redundant gateways. The local system can work in two modes: online mode and offline mode. When connectivity to the cloud is enabled, it operates according to the settings that are synchronized with the device twin. Once the Internet connection is off, it runs with pre-defined local settings. The switch between the two modes is transparent to users.

The cloud part relies on the infrastructure of Azure IoT Hub. The periodic messages sent from the local microprocessor will be processed by IoT Hub and then routed to different services, such as storage and web visualization. The website host in Azure cloud provides real-time data visualization to administrators, and can also query historic data from the database and graphically display them. Using the interface of the website, administrators and users are able to perform the management, by either updating the device twins to modify the time schedules or invoking functions to immediately execute according to their needs.

V. SYSTEM DESIGN AND IMPLEMENTATION

In this section, the design and implementation of the solution are presented. The architecture of the system is shown in Fig. 2, which details the connectivity of hardware, data stream between components and adopted services.

A. HARDWARE COMPONENTS

1) MICROPROCESSOR

The microprocessor in the local part is the central processing unit that is in charge of transmitting sensor data to and

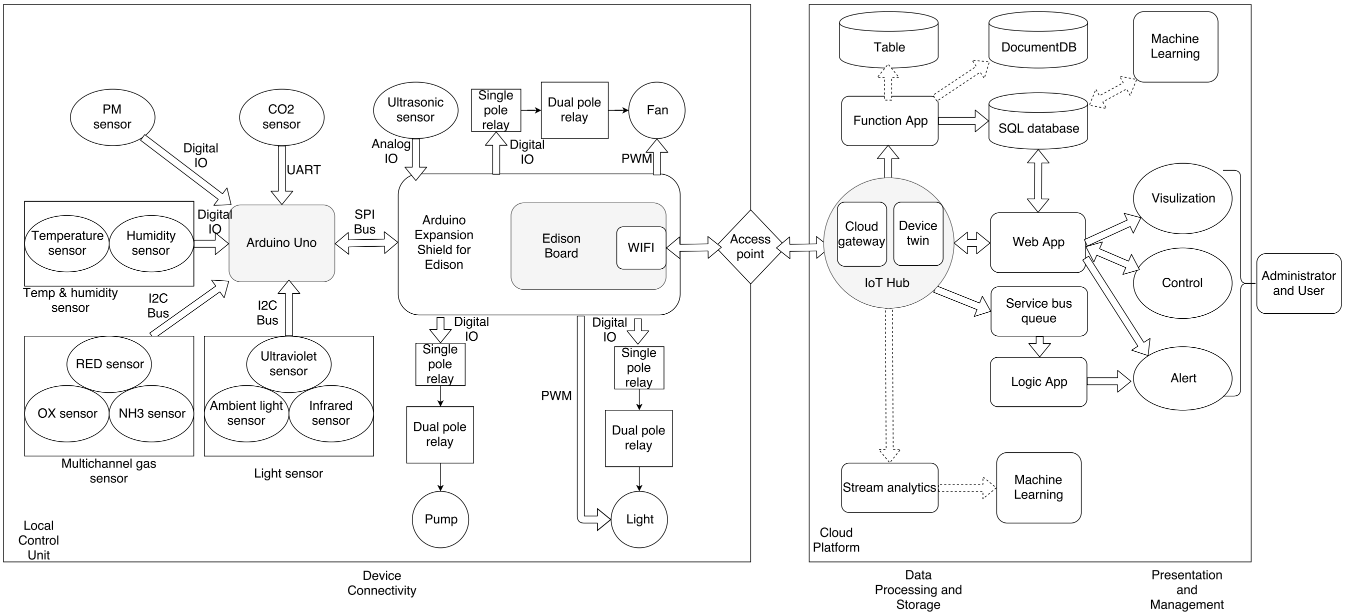


FIGURE 2. System architecture of proposed remote monitoring and management system for plant walls.

receiving messages from the cloud and executing control functions to take care of the plant wall. In our solution, Intel Edison is used as the head of the local unit. The Edison module is a system-on-chip (SoC) integrated with a dual core, dual-threaded Intel Atom 22 nm CPU that runs at 500 MHz. The module is powered by a 7-15 V power supply. It has an integrated on-chip WiFi module and a Bluetooth 4 module, which enables an out-of-box connection to the Internet. Edison supports many external interfaces, including SD card, UART, I2C, SPI and I2S protocols. Edison also has abundant GPIO pins for digital reading, writing or communication, of which four pins provide PWM output. In the SoC, Edison is also embedded with 1 GB RAM and 4 GB eMMC flash storage. With the Arduino shield for the Edison module, it is compatible with various Arduino peripherals but with greater powerful processing ability. A modern embedded Linux operating system, namely, Yocto Linux, is natively supported by Edison to enhance its ability and simplify the embedded development. Edison can be easily accessed by either virtual serials using the USB port or SSH via WiFi.

2) MICROCONTROLLER

The microcontroller in our solution is Arduino Uno due to its popularity and broad support from the public. It can be powered either by an AC-to-DC converter with a 6-20 V power supply or an USB cable. The Arduino Uno board is based on the ATmega328 chip. The board has six analog input pins and 14 digital input/output pins, of which six pins can be employed as PWM outputs. ATmega328 supports UART serial communication and I2C and SPI bus. The rich I/O and communication interface makes Arduino Uno the reliable choice for controlling the sensor readings.

3) SENSORS

a: TEMPERATURE AND HUMIDITY SENSOR

Temperature and RH are measured by the Grove DHT11 sensor, which provides pre-calibrated digital output and guarantees long-term reliability. The humidity is measured by the capacitive sensor element, whereas temperature data are fetched by a negative temperature coefficient (NTC) thermistor. The measurement range is 20-50% for RH and 0-50 °C for temperature, which is sufficient in an indoor scenario.

b: LIGHT SENSOR

The SI1145 sensor from Silicon Lab is utilized to detect the light luminance to determine its suitability for plants to thrive. The device has a visible light sensor and an infrared light sensor integrated in the device. The ultraviolet index is approximated by measurements of visible light and infrared light. This sensor has a wide spectrum support, which ranges from 280 to 950 nm. The sensor is connected to Arduino via the I2C bus.

c: CO_2 SENSOR

The CO_2 sensor that is adopted in our case is MH-Z16, which is a low-power sensor with high sensitivity and high resolution. This sensor uses nondispersive infrared to detect the CO_2 concentration in the air. The sensor can operate at 0-50 °C and 0-90% RH levels; the measurement range is 0 to 20 000 parts per million (PPM). The output mode can be set to UART, PWM wave or an analog signal.

d: PM SENSOR

A Grove dust sensor is used to measure the PM level in our solution. The mechanism counts the low pulse occupancy time (LPO) in a given time unit since the LPO time is proportional to the PM concentration. This sensor can detect particles with the diameter as small as 1 μm, and the detectable range is 0 to 28 000 pieces per liter. Digital output is enabled by the device.

e: GAS SENSOR

We employ a Grove multichannel gas sensor for the measurement of gas concentration. This sensor is built on MiCS-6814, which is a robust MEMS sensor that is used to detect vehicular exhausts and agricultural or industrial odors. Three channels are available on the devices and eight different gases can be detected: carbon monoxide (CO), nitrogen dioxide (NO_2), ethanol (C_2H_6OH), hydrogen (H_2), ammonia (NH_3), methane (CH_4), propane (C_3H_8) and iso-butane (C_4H_{10}). An ATmega168 microcontroller is embedded in the device; it reads sensor data from MiCS-6814 via an AD converter and transmits the results using the I2C bus.

f: ULTRASONIC SENSOR

An ultrasonic sensor is installed at the bottom of the plant wall to measure the water level in the tank. Real-time monitoring of the water level can be very helpful because it reminds customers when to fill the tank with water. In the future, a smart algorithm can be applied to efficiently control the pumping function to save energy. We deploy the UNAM 18U6903/S14 sensor from Baumer in the system because it promises 0.5 mm accuracy within the range 100-1 000 mm. The range is adjustable by a Teach-in pin in the device that requires 15-30 V DC input; the results are reflected via a 0-10 V analog output.

4) ACTUATORS

The actuators in the system consist of a pump, lights and fans. The pump is operated with a 230 V AC power supply, whereas the lights and fans are supplied with 24 V DC. Considering safety, we connect dual-pole relays to the actuators and utilize single pole relays to switch on or off the dual-pole relays. The light and fan accept a standard 0-10 V input to control the brightness and fan speed.

5) HARDWARE CONNECTIVITY

Fig. 2 depicts the connectivity of the deployed hardware in the local unit. The multichannel gas sensor and light sensor are linked to the Arduino board via the I2C bus. The PM sensor, and the temperature and humidity sensor are sampled using digital IO according to their specifications. UART is employed to fetch data from the CO_2 sensor. The communication between Arduino Uno and Edison is based on the SPI bus. Arduino will transmit the latest data to Edison on query. Three single-pole relays are directly switched on or off by the GPIO ports in Edison. To control the brightness of the lights and the speed of the fans, two PWM ports in the Edison board are occupied to generate varying voltage signals.

The ultrasonic sensor is directly connected to Edison because the water level is critical to the system; thus, the latency of data-fetching should be minimized based on the demand for precise control of the pumping function and reasonable preparation for the potential tank-autofill function in the future.

The ultrasonic sensor has the maximum output of 10 V, and the lights and fans need 0-10 V input control signals. Edison works at 5 V in its IO ports. The power supply for the ultrasonic sensor is distinct with Arduino and Edison. Therefore, a necessary circuitry is designed to enable the connectivity. The circuitry has a uniformed 20 V DC power supply for one 7815 regulator and three 7812 regulators. The 15 V output from the 7815 regulator drives the ultrasonic sensor, whereas the function of the 12 V outputs from the 7812 chips is to power Arduino, Edison and a LM324 operational amplifier chip. The signal from the ultrasonic sensor is passed through a voltage divider circuit and then a voltage follower based on an operational amplifier. The signal can be read by the Edison board. The 0-5 V PWM signal from Edison passes through a similar voltage divider and then enters a 4X amplification circuit to reach the 0-10 V output range to control the lights and fans.

B. SOFTWARE DESIGN

In this subsection, the software implementation of the local control unit and the services that run in the cloud are presented.

1) MICROCONTROLLER

The program that runs in the Arduino microcontroller is responsible for obtaining the sensor readings and transmitting the data to Edison during query. Benefiting from the resourceful support of the Arduino community, the majority of the sensors deployed in our solution have had a library-supporting Arduino board, which significantly eases our work in integrating miscellaneous sensors with the microcontroller.

During program startup, the necessary ports are initialized for additional communication with the sensors and enable a hardware interrupt, which is invoked by Edison for data query. The main loop in the program is to periodically fetch every sensor reading. The reading interval is set to five seconds, and the results are locally stored and updated in a buffer. When the interrupt is triggered by Edison sending a byte through the SPI bus, an SPI interrupt service routine is called to return the buffered sensor data. We use a flag to point to the position in the buffer for the next data transfer because SPI can only transmit one byte at a time. After all data have been transferred, the flag will be cleared by a rising signal from the Edison board via another interrupt service routine.

2) MICROPROCESSOR

The Edison board is running a Yocto Linux operating system that provides sufficient support for embedded development. The program in our solution is written in the C programming language due to its high execution efficiency. The main implementation of the program includes the following tasks:

- initialization and de-initialization,
- SPI communication with microcontroller,
- reading ultrasonic sensor,
- control of actuators,
- periodically transmitting messages to the cloud,
- listening to incoming messages from the cloud,
- monitoring and executing scheduled events.

a: LIBRARIES
Intel provides an MRAA Linux library for low-speed IO communication to simplify the embedded development process, which has native support for the Edison platform. Our program relies on the MRAA library to initialize and de-initialize necessary IO pins, fetch analog input voltages and operate digital input and output signals. MRAA also enables PWM function APIs; with its assistance, we can easily modify the control signal to the lights and fans by changing the pulse width and period so as to adjust the duty cycle.

To facilitate the IoT development based on Azure IoT Hub, Azure provides an SDK, namely, Azure-iot-sdk to the public, which is available in several popular languages. The SDK consists of device SDK parts and service SDK parts for the purpose of device communication with the cloud and the IoT Hub management, respectively. The device SDK enables different types of IoT devices to connect to IoT Hub in a uniform and secure manner. The device SDK provides support for three transport protocols: HTTPS, MQTT and AMQP. The message exchanges, commands and property update functions are also simplified by the device SDK for developers. In our program, which runs on Edison, we take full advantage of the device SDK to implement the continuous transmission of the sensor data, device twin updates and management commands from administrators, which significantly eases the engineering work and improves the program soundness.

b: MULTITHREADING
Considering the various tasks in the program, we employ the multithreading technique to enable several functions to be concurrently executed. Data fetch from Arduino is a constant and periodic task that runs on the Edison board. Therefore, a child thread is created after the platform initialization to execute the designate job. The thread attempts to send a data fetch command to Arduino and trigger its interrupt using the SPI bus and then waits for the feedback until all sensor data are transferred. A checksum is utilized to guarantee the data completeness during the transfer. The interval between two successive data fetches is set to five seconds. The renewed sensor data are flexibly written to a local buffer that can be read by the main thread. To protect concurrent reading and writing, a semaphore lock is employed in the program.

Another task that is enforced by multithreading is the pumping function. We decide to create a child thread to perform the job, because it may take 10 to 15 minutes to water the plant wall each time. This child thread is not attached to the main thread; thus, it can exit silently after pumping without any influence on the main program.

c: PROPERTIES
As mentioned in III-B, IoT Hub uses device twins to realize device management. A device twin records all necessary properties of an IoT device, which is used to describe the device parameters and status. The properties are composed of desired properties and reported properties. The administrators/users can manage the plant wall by modifying the desired properties in the cloud, whereas the changes are informed to the device. After receiving the notification, the device would change the local settings according to the desired properties and then update its value to the reported properties to ensure that the desired and reported properties are synchronized in the cloud.

In the plant wall system, we need to control the pump, lights and fans at given times. Thus, the status of the actuators, such as on/off, brightness, speed, and execution time, are considered to be properties. We have pre-defined values for these properties in the case that a device loses connectivity to the cloud to ensure that the plant wall is still taken care on a restricted level.

Fig. 3 lists the properties that we employed in our system. The desired/reported interval property is to adjust the time period that Edison reports sensor values to the cloud. The reboot number property in the reported property is used to check if a reboot has been completed.

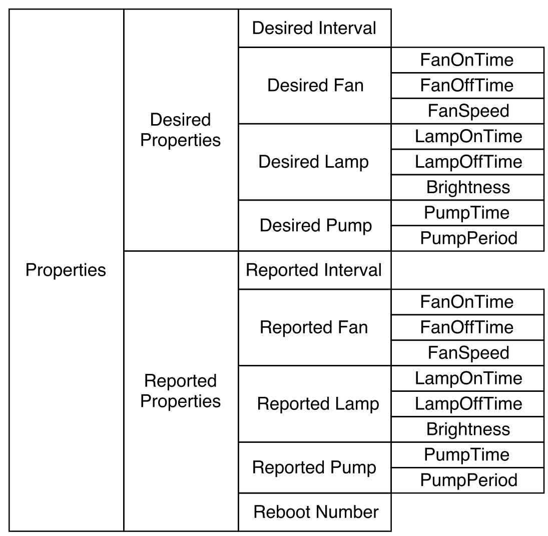


FIGURE 3. Properties used in the plant wall system.

d: DIRECT METHODS
The direct method is one of the C2D communication types supported by Azure IoT Hub. It is used to order commands that need immediate execution. In the plant wall system, we also create several direct methods that administrators can employ to enable direct control over actuators and system behavior.

- *Pump turned on method*
- *Fan turned on method*
- *Fan turned off method*

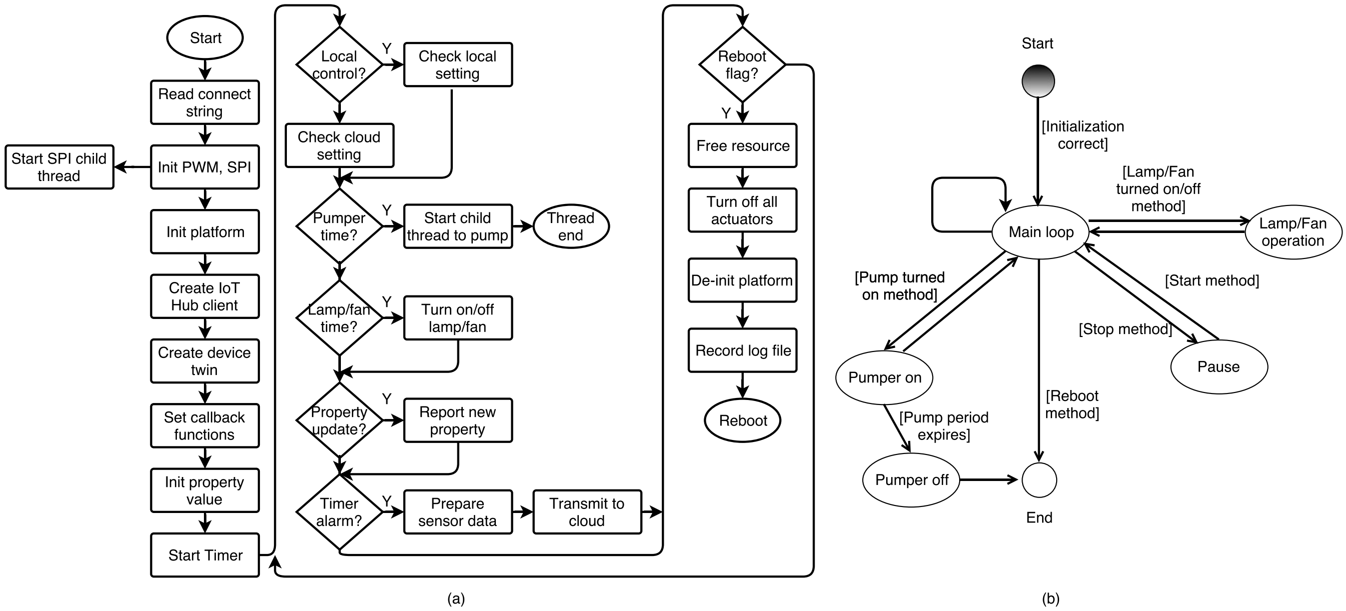


FIGURE 4. **Flowchart and state machine of the program. (a) Flowchart. (b) State machine.**

- *Lamp turned on method*
- *Lamp turned off method*
- *Program start method*
- *Program stop method*
- *Reboot method*

As the names suggest, the first five methods are called to operate the pump, lights and fans. Program start/stop methods are applied to temporarily pause and restart the program in case an abnormal event occurs,whereas the reboot method is used to resolve software failures and maintain the long-term reliability of the system. Corresponding handlers are linked with these methods in the program.

e: FLOWCHART AND STATE MACHINE

The main thread starts by reading the connection string from a local file that contains the domain and credential information needed to connect to IoT Hub. After a PWM port and an SPI bus are initialized, the child thread that is in charge of communicating with Arduino is immediately created and keeps running until the main thread is suspended. With the aid of Azure IoT SDK, an IoT Hub client is created with its device twin to perform the communication with IoT Hub. The callback functions are the key to enabling commands and property updates from the cloud and should be appropriately linked to their implementations. We use a timer to trigger the periodic message transmission. In the main loop, the device will check different settings according to the online or offline mode in which it operates. By comparing the current timestamp with the actuator working time, the device decides whether or not to execute the corresponding functions. The timer is set to send a signal to the main thread, which will notify the device to perform a sensor reading and transmission. When the client receives an update to the desired properties, it must renew the local properties and report to IoT Hub. The main thread would be kept in a loop unless it detects that a reboot method is invoked. In this case, it shall properly release the resource, turn off all actuators for safety, record useful logs and execute the reboot. When receiving direct methods from IoT Hub, the device would perform relevant callback functions without delay, i.e., to turn on/off lights and fans. The pump can only be turned on and would automatically end because it runs on another thread. A detailed flowchart and state machine of the program are shown in Fig. 4.

3) CLOUD PART

The cloud part has the essential role of enabling the remote monitoring and management system for plant walls, since the data processing, storage and interface for administrators are hosted in the cloud. Our cloud solution is built on top of a combination of Azure cloud services, including IoT Hub, Function Application, SQL Database, Storage Service, Web Application and Logic Application, as displayed in Fig. 2. IoT Hub is the core service that sits in the center of the cloud solution. IoT Hub performs two fundamental functions, device management and message routing. A single IoT Hub supports the maximum number of 500,000 devices, which enables it to govern massive broadly distributed plant walls.

a: DEVICE MANAGEMENT

Device management involves device registration, authentication, device twin maintenance and communications with devices. Each device has a unique device ID that is stored in IoT Hub as an index. The key used for encryption is decided and contained in a connection string when a new device is added to IoT Hub. Devices use their own connection strings to authenticate themselves to the cloud. To govern the connected devices, we use the Azure IoT Hub service SDK to develop our back-end code in the JavaScript language, to realize the update of the desired properties and invoke the direct methods. As illustrated in Fig. 5, when IoT Hub receives the request from the back-end to invoke a direct method, it relays this request to a specific device. Immediately, the device gives a response to notify the successful reception of the request. After taking corresponding actions, the device reports to IoT Hub about the results of the execution. By querying the reported properties, the back-end is aware of the status of execution.

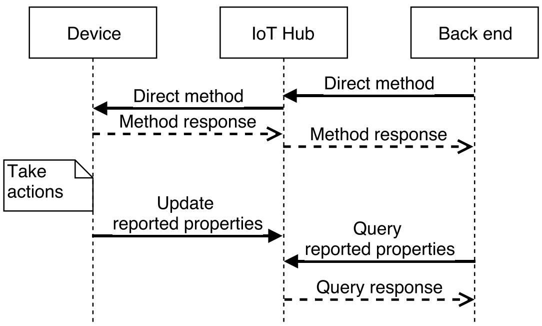


FIGURE 5. Message flow of direct method.

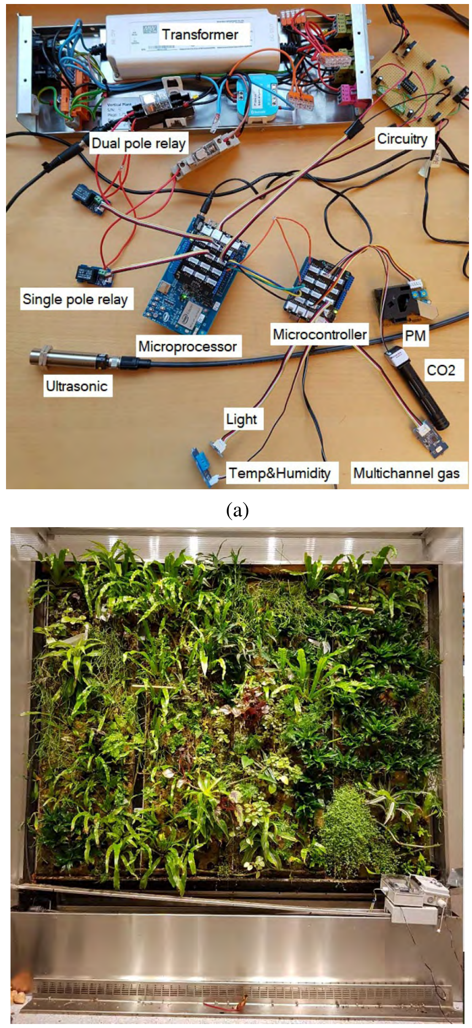
Using a property update to modify the device parameters has a similar message flow. When the device receives the notification from IoT Hub about a desired property update, it will not provide a response but will directly change the local property values to apply the update.

b: MESSAGE ROUTING

IoT Hub provides native support for a message routing service, with which an SQL-like query language can be applied to filter the received messages. IoT Hub exposes several endpoints that can be accessed by the consumers of the message. In our solution, we assign messages to data and alert two types at transmission. Messages are directed to different services according to their type properties that are stamped in their headers. Regular sensor data is stamped with the data type, whereas an alert message is transmitted when an abnormal event occurs, e.g., the water level in the tank is too low or the CO_2 level is too high. Data messages are routed into the default endpoint for additional processing while alert messages are guided into another endpoint that is linked to a queue service. A logic application service is subscribed to periodically monitor the queue. Once alert messages are detected, an email is sent to the administrator for additional probing.

c: STORAGE

In the plant wall system, continuously generated data enable users to have a comprehensive view of the indoor climate fluctuations. Detailed historic data also afford the possibility of future data analysis that is based on novel data mining and machine learning. Flexible data storage is another merit offered by the public cloud platform. In our solution, an MS SQL database hosted in Azure is chosen to store historic data from the plant walls. A function application that bridges IoT Hub to the database is deployed. The function application is written in the JavaScript language and set to execute when triggered by the IoT Hub messages. The application parses the sensor data from the JSON format messages and inserts them into the SQL database. According to previous research experiences [25]–[27], the indoor environment tends to change in a gentle way and drastic fluctuations rarely happen. Therefore, the insertion operation interval is set to 15 minutes to attain a trade-off between data storage cost and data precision. We also reserve the possibility of storing data into a non-relational database due to its increasing popularity among industries and table storage considering the economic cost.



(a)

(b)

FIGURE 6. System setup and deployment. (a) Setup. (b) Deployment.

FIGURE 7. **Updating parameters of fan on and off time. (a) Updating properties. (b) Relay: fan on (c) Relay: fan off.**

d: INTERFACE

The essential part of the cloud solution is the user interface for both administrators and end users. We have developed a web interface using the popular Django framework and deployed it in the Azure web application service. The web application performs live stream and historic data visualization and provides a control interface. It subscribes to the IoT Hub default endpoint to fetch the continuously updated sensor data and display them in a real-time manner. If some abnormal values are detected, the live stream will be switched to a highlighted color to alert people. When historic data are requested, the interface queries and visualizes stored data from the SQL database. The graphical interface for control of the pump, lights and fans and modification of the system properties is available on the website as a convenience to the management.

e: RESERVED FUNCTIONS

Machine learning (ML) and artificial intelligence (AI) have gained global attention during the last few years and become a driven force for digitalization of many traditional industries. Foreseeing this inspiring trend, we have reserved the possibility of performing data analysis based on the machine learning module in Azure to take full advantage of stored historic data and real-time data stream and explore the value of the data.

VI. RESULTS

The system setup is shown in Fig. 6(a). It is deployed in the vertical plant wall, see Fig. 6(b), installed in our university lab as a verification to the proposed system. The Edison board is connected to Azure IoT Hub via the eduroam WiFi network.

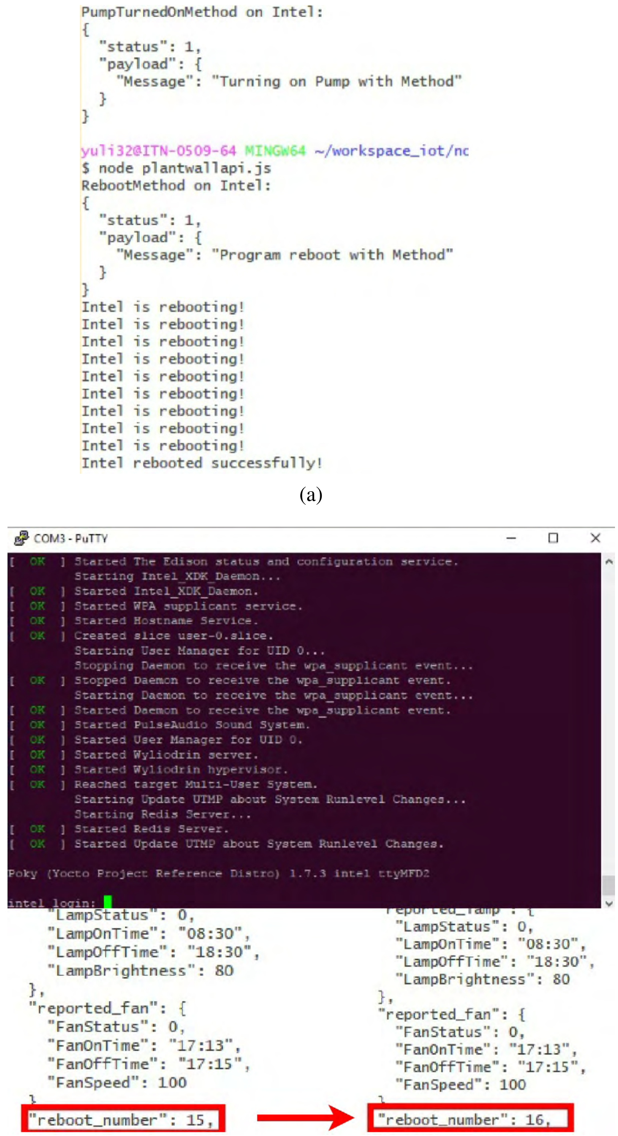


FIGURE 8. **Invoking direct method. (a) Invoking method from back-end. (b) Updating reboot property.**

We create a device in the IoT Hub portal and assign a device ID to the board, namely, "intel". When the system is booted up, it starts to report pre-defined parameters to IoT Hub to initialize the device twin properties.

Parameter updating is essential for remote management. We remotely update the fan on and off time values to "17:13" and "17:15", respectively. The updated device twin properties are displayed in Fig. 7. These modified time schedules are rapidly synchronized to the device. The single pole relay that is used to control the fans is punctually switched on and switched off two minutes afterwards.

Fig. 8 depicts the results after remotely invoking direct methods. First, the pump function is called via IoT Hub from the back-end. The device sends a notification to the back-end to confirm the reception of the method and then turns on

FIGURE 9. Web interface.

the pump. The reboot method is also verified using the same back-end. During reboot, the back-end keeps querying the device twin about the reboot number property. When reboot finishes, the device updates the property by increasing it by one so that the back-end is notified.

The web interface is implemented, as shown in Fig. 9. Different tabs exist to switch from live stream to historic data. On the live stream page, all sensor data are visualized in a real-time manner, and abnormal values are highlighted using alerting color. The control buttons and parameter update functions are placed on the same page for convenience. When switched to historic data, the app would query the SQL database for the stored data of a specific period and then display the data.

VII. CONCLUDING REMARKS

This study addresses the critical challenge of digitalization of green technology, e.g., plant walls. The need for remote monitoring and management has become a bottle neck that blocks the widespread distribution of the plant walls and their massive production. In this study, a remote monitoring and management solution that is based on IoT and a public cloud platform is proposed. The system has been completely developed from hardware to software, and from the local control unit to the cloud end. It is capable of performing fundamental plant care functions, such as watering, lighting and ventilation according to user scheduling. Several environmental sensors are integrated and the data are continuously transmitted to the public cloud for real-time visualization and data storage. Via Azure IoT Hub and a web-based interface, administrators and users are able to perform remote monitoring and maintenance.

The system has been tested and verified by an experimental deployment. The results indicate that a cloud and IoT-based remote monitoring and management system can be a significant merit to plant walls, in terms of its reliable performance, real-time monitoring, timely feedback and convenient remote control. This solution may greatly benefit plant wall suppliers by simultaneously improving the maintenance efficiency but reducing the cost. The system enables massive and broad deployments of plant walls in public and private buildings and contributes to a green indoor climate in the long term.

The IoT and cloud-based solution attains the study goal, i.e., to endow the plant wall system with a digitalized soul. Benefiting from its forward-looking design, this research can also be the starting point towards an intelligent plant wall system in the near future by applying novel data mining and machine learning technologies to the saved historic data to realize adaptive maintenance for the indoor climate. This framework is also applicable to other applications that need remote monitoring and control services.

ACKNOWLEDGEMENT

The authors thank Gustav Knutsson for his work maintaining the plant wall lab and supporting our study. Dejan Ilic and Thibault Durand at LiU-IT are acknowledged for their support concerning the Azure cloud and database.

REFERENCES

[1] M. Luo, W. Ji, B. Cao, Q. Ouyang, and Y. Zhu, "Indoor climate and thermal physiological adaptation: Evidences from migrants with different cold indoor exposures," *Building Environ.*, vol. 98, pp. 30–38, Mar. 2016.

[2] I. Paciência, J. Madureira, J. Rufo, A. Moreira, and E. de Oliveira Fernandes, "A systematic review of evidence and implications of spatial and seasonal variations of volatile organic compounds (VOC) in indoor human environments," *J. Toxicol. Environ. Health B, Crit. Rev.*, vol. 19, no. 2, pp. 47–64, 2016.

[3] A. Bondarevs, P. Huss, S. Gong, O. Weister, and R. Liljedahl, ''Green walls utilizing Internet of Things,'' *Sensors Transducers*, vol. 192, no. 9, pp. 16–21, 2015.

[4] J. Francis, G. Hall, S. Murphy, and J. Rayner. (Feb. 2014). *Growing Green Guide*. [Online]. Available: http://www.growinggreenguide.org/

[5] M. Köhler, M. Schmidt, F. W. Grimme, M. Laar, V. L. de Assunção Paiva, and S. Tavares, ''Green roofs in temperate climates and in the hot-humid tropics–far beyond the aesthetics,'' *Environ. Manage. Health*, vol. 13, no. 4, pp. 382–391, 2002.

[6] Y. Wang, F. Bakker, R. de Groot, and H. Wörtche, ''Effect of ecosystem services provided by urban green infrastructure on indoor environment: A literature review,'' *Building Environ.*, vol. 77, pp. 88–100, Jul. 2014.

[7] S. Abdelwahab, B. Hamdaoui, M. Guizani, and A. Rayes, ''Enabling smart cloud services through remote sensing: An Internet of everything enabler,'' *IEEE Internet Things J.*, vol. 1, no. 3, pp. 276–288, Jun. 2014.

[8] L. Zhou, D. Wu, J. Chen, and Z. Dong, ''When computation hugs intelligence: Content-aware data processing for industrial IoT,'' *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1657–1666, Jun. 2018.

[9] I. D. Peake and J. O. Blech, ''A candidate architecture for cloud-based monitoring in industrial automation,'' in *Proc. IEEE Int. Conf. Softw. Qual., Rel. Secur. Companion (QRS-C)*, Jul. 2017, pp. 548–554.

[10] S. B. Baker, W. Xiang, and I. Atkinson, ''Internet of Things for smart healthcare: Technologies, challenges, and opportunities,'' *IEEE Access*, vol. 5, pp. 26521–26544, 2017.

[11] G. Neagu, S. Preda, A. Stanciu, and V. Florian, ''A cloud-IoT based sensing service for health monitoring,'' in *Proc. e-Health Bioeng. Conf. (EHB)*, Jun. 2017, pp. 53–56.

[12] G. Yang *et al.*, ''IoT-based remote pain monitoring system: From device to cloud platform,'' *IEEE J. Biomed. Health Informat.*, to be published, doi: 10.1109/JBHI.2017.2776351.

[13] A. Celesti, A. Galletta, L. Carnevale, M. Fazio, A. L. Ekuakille, and M. Villari, ''An IoT cloud system for traffic monitoring and vehicular accidents prevention based on mobile sensor data processing,'' *IEEE Sensors J.*, vol. 18, no. 12, pp. 4795–4802, Jun. 2018.

[14] T. Pham, M. Tsai, D. Nguyen, C. Dow, and D. Deng, ''A cloud-based smart-parking system based on Internet-of-Things technologies,'' *IEEE Access*, vol. 3, pp. 1581–1591, 2015.

[15] S. Salvi *et al.*, ''Cloud based data analysis and monitoring of smart multi-level irrigation system using IoT,'' in *Proc. Int. Conf. I-SMAC (IoT Social, Mobile, Anal. Cloud) (I-SMAC)*, Feb. 2017, pp. 752–757.

[16] Y.-T. Wang, Y.-P. Chiang, C.-H. Wu, C.-T. Yang, S.-T. Chen, and P.-L. Sun, ''The implementation of sensor data access cloud service on HBase for intelligent indoor environmental monitoring,'' in *Proc. 15th Int. Symp. Parallel Distrib. Comput. (ISPDC)*, Jul. 2016, pp. 234–239.

[17] G. Marques and R. Pitarma, ''An indoor monitoring system for ambient assisted living based on Internet of Things architecture,'' *Int. J. Environ. Res. Public Health*, vol. 13, no. 11, p. 1152, 2016.

[18] R. Pitarma, G. Marques, and B. R. Ferreira, ''Monitoring indoor air quality for enhanced occupational health,'' *J. Med. Syst.*, vol. 41, no. 2, p. 23, Feb. 2016.

[19] F. Salamone, L. Belussi, L. Danza, T. Galanos, M. Ghellere, and I. Meroni, ''Design and development of a nearable wireless system to control indoor air quality and indoor lighting quality,'' *Sensors*, vol. 17, no. 5, p. 1021, 2017.

[20] S. Bhattacharya, S. Sridevi, and R. Pitchiah, ''Indoor air quality monitoring using wireless sensor network,'' in *Proc. 6th Int. Conf. Sens. Technol. (ICST)*, Dec. 2012, pp. 422–427.

[21] N. Madrid, R. Boulton, and A. Knoesen, ''Remote monitoring of winery and creamery environments with a wireless sensor system,'' *Building Environ.*, vol. 119, pp. 128–139, Jul. 2017.

[22] G. Zhiqiang, H. Lingsong, T. Hang, and L. Cong, ''A cloud computing based mobile healthcare service system,'' in *Proc. IEEE 3rd Int. Conf. Smart Instrum., Meas. Appl. (ICSIMA)*, Nov. 2015, pp. 1–6.

[23] H. Jiang, Y. Li, and D. Li, ''Indoor environment monitoring system based on linkit one and yeelink platform,'' in *Proc. 2nd IEEE Int. Conf. Comput. Commun. (ICCC)*, Oct. 2016, pp. 933–937.

[24] C.-T. Yang, S.-T. Chen, W. Den, Y.-T. Wang, and E. Kristiani, ''Implementation of an intelligent indoor environmental monitoring and management system in cloud,'' *Future Gener. Comput. Syst.*, to be published.

[25] J. Zhang, A. Huynh, P. Huss, Q.-Z. Ye, and S. Gong, ''A Web-based remote indoor climate control system based on wireless sensor network,'' *Int. J. Sensors Sensor Netw.*, vol. 1, no. 3, pp. 32–40, 2013.

[26] J. Zhang, A. Huynh, Q.-Z. Ye, and S. Gong, ''Remote sensing system for cultural buildings utilizing ZigBee technology,'' in *Proc. 8th. Int. Conf. Comput., Commun. Control Technol. (CCCT)*, Apr. 2010, pp. 71–77.

[27] J. Zhang, A. Hyunh, Q. Ye, and S. Gong, ''Design of the remote climate control system for cultural buildings utilizing ZigBee technology,'' *Sensors Transducers*, vol. 118, no. 7, pp. 13–27, 2010.

[28] S. Balampanis, S. Sotiriadis, and E. G. M. Petrakis, ''Internet of things architecture for enhanced living environments,'' *IEEE Cloud Comput.*, vol. 3, no. 6, pp. 28–34, Nov. 2016.

[29] J. Peng, X. Zhang, Z. Lei, B. Zhang, W. Zhang, and Q. Li, ''Comparison of several cloud computing platforms,'' in *Proc. 2nd Int. Symp. Inf. Sci. Eng.*, Dec. 2009, pp. 23–27.

[30] S. Singh, Y.-S. Jeong, and J. H. Park, ''A survey on cloud computing security: Issues, threats, and solutions,'' *J. Netw. Comput. Appl.*, vol. 75, pp. 200–222, Nov. 2016.

[31] Microsoft. (2017). *Azure IoT Hub Guide*. [Online]. Available: https://docs.microsoft.com/en-us/azure/iot-hub

[32] Microsoft. (2016). *Microsoft Azure IoT Reference Architecture*. [Online]. Available: https://azure.microsoft.com/en-us/updates/microsoft-azure-iot-reference-architecture-available/

YU LIU received the B.Eng. degree in electronics science and technology from the Harbin Institute of Technology, Weihai, China, in 2014, the M.Sc. degree in computer science from the University of Trento, Trento, Italy, in 2016, and the M.Sc. degree in innovation in information and communication technology from the Technical University of Berlin, Berlin, Germany, in 2017. He is currently pursuing the Ph.D. degree with the Department of Science and Technology, Linköping University, Sweden. His research interests include cloud-based Internet of Things solution, embedded systems, and wireless sensor networks.

KAHIN AKRAM HASSAN received the B.Sc. and M.Sc. degree in media technology and engineering from the Department of Science and Technology, Linköping University, Sweden, in 2015, where he is currently pursuing the Ph.D. degree in information visualization of temporal multi-dimensional data sets.

MAGNUS KARLSSON was born in Västervik, Sweden, in 1977. He received the M.Sc., Licentiate of Engineering, and Ph.D. degrees from Linköping University, Sweden, in 2002, 2005, and 2008, respectively. In 2003, he joined the Communication Electronics research Group, Linköping University, where he is currently a Senior Researcher and a Lecturer. His main work involves wideband antenna-techniques, wideband transceiver front-ends, and wireless communication.

OLA WEISTER became interested in and began his education within the horticultural trade in early years. In 1992, he received the degree in horticultural engineering from SLU University, Alnarp, Sweden, with a focus on plant production in greenhouses. He was a Faculty Member with SLU University, where he was involved in landscape architecture, horticulture, and crop production science. Since 1992, he has been an entrepreneur with four different companies all within the horticultural trade. From 2013 to 2017, he was a member of the European Interior Landscaping Organisation Board. His research interests include interior landscaping and plant walls, landscape design, orangeri plants, urban farming, and plant light.

SHAOFANG GONG received the B.Sc. degree in microelectronics from Fudan University, China, in 1982, and the Licentiate of Engineering and Ph.D. degrees from Linköping University, Sweden, in 1988 and 1990, respectively. From 1991 to 1999, he was a Senior Researcher with the Research Institute RISE Acreo. From 2000 to 2001, he was the chief technology officer at a spin-off company from the research institute. In the meantime, he had an Adjunct Professorship at Linköping University. Since 2002, he has been a Chair Professor of communication electronics with Linköping University. His main research interest has been communication electronics, including radio frequency and microwave system design, high speed data transmissions, and wireless sensor networks towards Internet of Things.