An IoT Platform for Smart Plant Care

Whai-En Chen[1](✉), Ming-Yih Chang[2], Kuan-Lin Chou[1], and Jin-Qiu Shi[2]

[1] Department of Computer Science and Information Engineering, National Ilan University, No. 1, Sec. 1, Shennong Rd., Yilan City 260, Yilan County, Taiwan
wechen@niu.edu.tw

[2] Department of Biomechatronic Engineering, National Ilan University, No. 1, Sec. 1, Shennong Rd., Yilan City 260, Yilan County, Taiwan

Abstract. The Internet of Things (IoT) technology is changing the agriculture to smart farming which enables farmers to face new challenges in agriculture. This paper develops an IoT platform for smart plant care where the sensors detect the environment parameters and report them to the server. The server invokes the fan and LED by calculating the parameters. In addition, the parameters are stored into the SD card when the connection to the server is failure. Once the emergency event occurs, the platform notifies the users by dialing the phone number instead of polling the system. This way saves the unnecessary messages exchange and power consumption.

Keywords: Arduino · Agriculture · IoT · Growth chamber · Smart farming

1 The Plant Care System

In this paper, we utilize a plant care system as an example to demonstrate a plant growth chamber. The plant care system is designed for hydroponic vegetables such as lettuce.

The plant care system consists of a cylinder [Fig. 1(a)], several electronic components and several sensors [Fig. 1(b), (d)], a Wi-Fi module and a micro-controller [Fig. 1(c)].

In the plant care system, we utilize ATmega328 (Arduino UNO board) as the micro-controller. The micro-controller connects to the sensors, the electronic components and the Wi-Fi module. The micro-controller receives the environment information (e.g., temperature and humidity) through the sensors, checks whether the value is within the threshold and sends the environment information to the server through the Wi-Fi module. For example, if the value is over the threshold (e.g., temperature is more than 25 °C), the micro-controller will activate the thermos electric cooling chip (i.e., the electronic component) to lower the temperature.

Growth cylinder [Fig. 1(a)] divided into two layers (upper and lower). The upper layer places the plant such as lettuce and devices, including a load cell [Fig. 1(d)], humidity and temperature sensors [Fig. 2(a)], five 3-watt white LED lights [Fig. 2(b)], a 150-watt thermoelectric cooling chip [Fig. 2(c)], an infrared sensor [Fig. 2(d)], a CO_2

Y.-B. Lin et al. (Eds.): IoTaaS 2017, LNICST 246, pp. 101–107, 2018.
https://doi.org/10.1007/978-3-030-00410-1_13

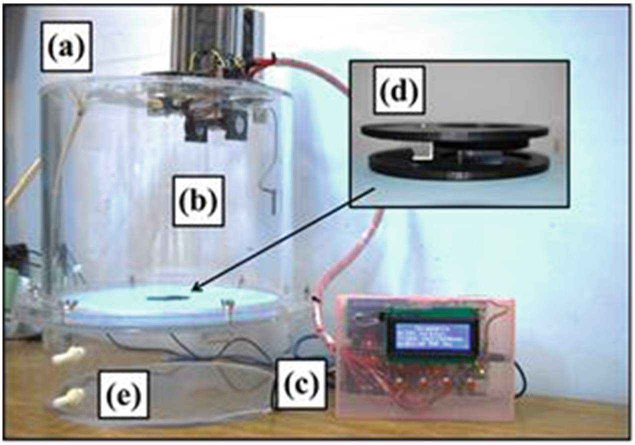

Fig. 1. Plant care system diagram

sensor [Fig. 2(e)] and a light sensor [Fig. 2(f)]. The lower layer sets drain hole [Fig. 1 (e)] used to easy to adjust the nutrient solution.

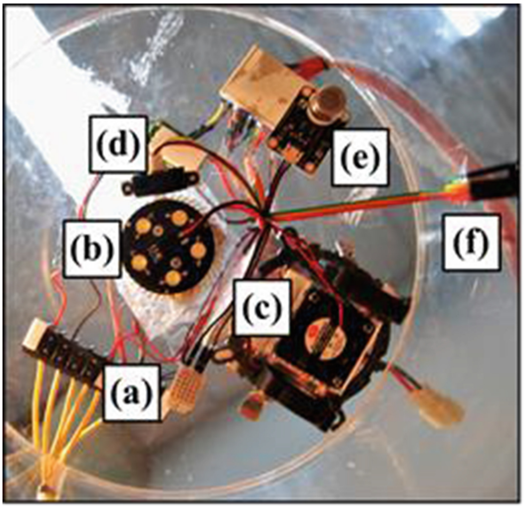

Fig. 2. Electronic components and sensors diagram

The humidity and temperature sensor [Fig. 2(a)] measures the relative humidity and temperature value. Take lettuce for example, the suitable relative humidity and temperature ranges are 70%–80% and 20 °C–25 °C. Based on the relative humidity value, the micro-controller adjusts the relative humidity through the thermoelectric cooling chip. When the temperature decreases lower than the dew point, the moisture condenses and the relative humidity is decreased.

The light intensity affects the growth of the plant. If the light intensity is not enough, the plant's growth will be slow. For example, the range 150–250 μmol/m^2/s is suitable for lettuce's growth. In plant care system, we use five 3-watt white LED lights [Fig. 2(b)] as the light source for plant illumination. The photoperiod are set to light/dark 16/8 h.

The temperature affects the growth of the plant, and thus the temperature should be controlled in a range that is suitable for the plant's growth. For example, the range 20–25 °C is the suitable temperature for lettuce's growth. In the plant care system, we use 150-watt thermoelectric cooling chip [Fig. 2(c)] to control the temperature.

The infrared sensor [Fig. 2(d)] measures height from growth cylinder's top to the plant's leaf crown. The plant height is the growth cylinder's height (22.5 cm) minus the measuring result.

CO_2 is also an important factor in plant growth. We use the CO_2 sensor [Fig. 2(e)] to monitor CO_2 concentration. Generally, we maintain the CO_2 concentration in the range 1100–1300 ppm. The CO_2 solenoid valve connects with a CO_2 cylinder. Through switching the CO_2 solenoid valve, we can adjust CO_2 concentration in growth cylinder.

The load cell (Fig. 1(d)) is used to measure the plant's weight.

The sensors monitor environment condition and transfer the result to the micro-controller. Then, the data will be transferred to the server. The server receives the data and executes the related procedures. One is used to store the data into the database, and the other is used to check the data is unusual or not. If the data is abnormal, the server will send the notification message to the user's handset. The users can check the data by the phone.

Figure 3 illustrates the data transmission architecture. The micro-controller connects to the server through an 802.11 Wi-Fi module. The module can be changed to Zigbee, Bluetooth or LoRa for different environments. The micro-controller invokes an HTTP (Hyper Text Transfer Protocol) library to generate the HTTP GET or POST requests to the server. The HTTP request contains the parameter names and values reported by the sensors. The user equipment (f) connects to the server through 3G (i.e., UMTS) or 4G. The server is developed by using apache, PHP and My SQL software. In addition, a phone is connected to the server through COM port (i.e., USB) and used to make a phone call to notify the system administrators.

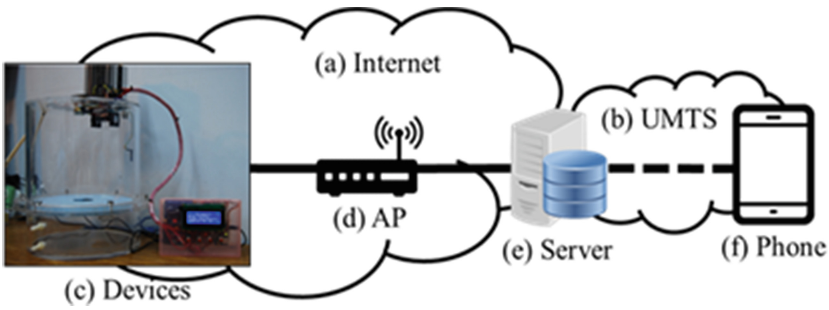

Fig. 3. Data transmission architecture

2 Solutions for Connection Failure and Emergency Notification

The server in the IoT platform stores the parameters reported from the micro-controller, calculates the optimized solutions and notifies the micro-controller taking actions. However, the wireless connection may not be always stable. This paper considers this situation and provides the solution for the connection failure. In addition, the emergency event (e.g., temperature is too high) may occur and the administrator should be notified. In this paper, we propose a PUSH mechanism instead of polling the server. The solutions are elaborated as follows.

2.1 Connection Failure Solution

Consider the wireless connection may be failure, we add an SD module to the micro-controller. Before sending the data to the server, the micro-controller checks whether the connection is successful. If yes, the micro-controller uploads the parameter names and values to the server through the HTTP requests. In addition, the micro-controller checks if there is any buffered data stored in the SD card. If the file exists, the micro-controller reads the parameters and uploads the parameters to the server. The example codes are shown as follows.

```
If(client.connect(SITE_URL, 80))//Check the connection
{
  if(LSD.exists(picName))//Check if the file exists
  {
    LFile dataFile = LSD.Open("dataSD.txt", FILE_READ);
    if(dataFile) //Open File Successful
    {
      dataFile.seek(0); //Move to Starting Point
      len = dataFile.size();
      dataFile.read(bufFile, len); //Read Seneor Data
      dataFile.close();
      befFile[len] = '\0';
    } else
    {
      //report error
    }
}
```

On the contrary, if the connection is failure, the micro-controller stored the parameters into a file in the SD card.

2.2 Emergency Notification

In traditional solution, the handset periodically sends the request to the server to check whether there is an emergency event. Since the emergency event does not occur frequently, most of the query messages are unnecessary that consume the wireless bandwidth and the battery power of the handset.

Figure 4 illustrates the emergency notification flow that is proposed by this paper. When the emergency event is detected and reported by the micro-controller [1], the server utilizes the **adb** command to invoke the android phone to make a phone call to the administrator's handset [2]. Upon receipt of the call, the application on the handset checks the caller identifier. If the caller identifier comes from the server, the application terminates the call and notifies the user by the sounds, vibration and light [3].

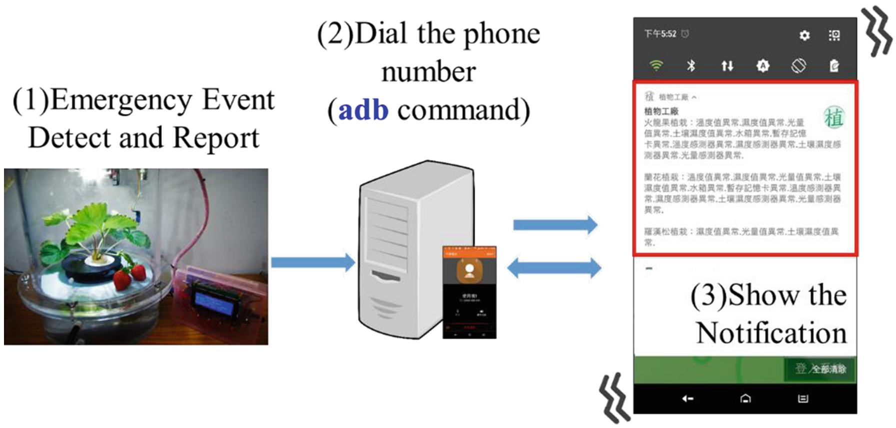

Fig. 4. Emergency notification flow

The script file (i.e., a batch file on Windows) is shown as follows. The script makes a call, waits **20** s, and terminates the call. Note that the call can be terminated by the called party. If the call is not terminated, the sever should terminate the call. Otherwise, the call enters the voicemail system. In this example, the script waits **20** s.

```
@echo on
adb shell am start -a android.intent.action.call -d
tel:%phonenumber
TIMEOUT 20
adb shell input key event 6
echo finish
```

On the android handset, the application utilizes **PhoneReceiver** to receive the incoming phone call information (i.e., the caller identifier).

3 Performance Evaluation

In this system, we would like to understand the performance of the proposed notification method. We compare the proposed method with the SMS (Short Message Service) which is also a PUSH service.

Figure 5 demonstrates the test environment which includes a server and two android handsets. The server is connected to one handset through the COM port (USB cable), and the handset is used to send the SMS message or make a phone call. The server and the handsets synchronize their clocks through the same NTP (Network Time Protocol) server. In the SMS method, the server sends the SMS messages with the sending timestamp to the called handset. The called handset can calculate the latency. In the proposed method, the server stores the sending timestamp in the notification while making the call. The application sends the request to retrieve the notification (with timestamp) after receiving the phone call from the server. The handset then obtains the latency.

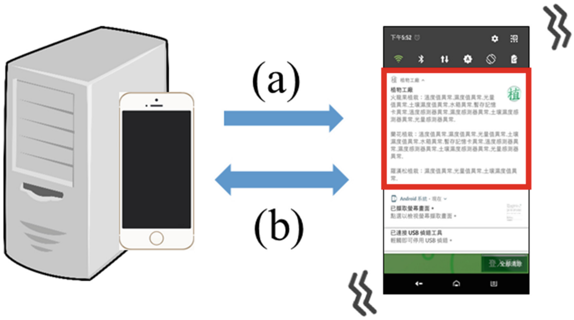

Fig. 5. Performance test environment

The comparison of the SMS and the proposed method are listed in Table 1. We observe that the latency of the SMS and that of the proposed method are very similar. However, the SMS is one-way communication from the server to the client. The proposed method a bi-directional communication. Using the proposed method, the server can record the timestamps that the notification arrives the handset and the user.

Table 1. The comparison of the SMS and proposed notification methods.

Notification method	SMS	Proposed method
Average latency	5.23 s	5.62 s
Variance	0.1501	0.2376
Communication	One-way (server → client)	Bi-direction

4 Conclusion

The Internet of Things (IoT) technology is changing the modern agriculture to smart farming which enables farmers to face new challenges in agriculture. This paper develop an IoT platform for smart plant care where the sensors detect the environment parameters and report them to the server. The server invokes the fan and LED by calculating the parameters. Specifically, we utilize a plant care system as an example to demonstrate the growth chamber. The plant care system is designed for hydroponic vegetables such as lettuce.

Besides the development of the sensor platform, we also consider the wireless connection failure. In this case, the parameters are stored into the SD card. Once the connection is recovery, the buffered data are retransmitted to the server. In this way, the data are not lost due to wireless link failure. We design the local control system instead of the cloud server to handle the environment control during the wireless failure. This part is not discussed in this paper.

When the emergency event occurs, the proposed IoT platform notifies the system administrators by dialing the phone number. This way saves the unnecessary messages exchange and power consumption. Through the experiments, the notification is almost the same fast (about 5 s) as the SMS and the proposed method provides bi-directional communications.

To sum up, the proposed IoT platform can provide reliable data records and effective emergency notification system which can be applied to the smart framing applications.

Acknowledgements. The work is supported by MOST projects 105-2218-E-197 -001 and 105-2218-E-007 -029.

References

1. Bing, F.: Research on the agriculture intelligent system based on IOT. In: 2012 International Conference on Image Analysis and Signal Processing, pp. 1–4 (2012)
2. Kapoor, A., Bhat, S.I., Shidnal, S., Mehra, A.: Implementation of IoT (Internet of Things) and image processing in smart agriculture. In: 2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), pp. 21–26 (2016)
3. Nakutis, V., et al.: Remote agriculture automation using wireless link and IoT gateway infrastructure. In: 2015 26th International Workshop on Database and Expert Systems Applications (DEXA), pp. 99–103 (2015)
4. Shenoy, J., Pingle, Y.: IOT in agriculture. In: 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), pp. 1456–1458 (2016)
5. Foster, I., Kesselman, C.: The study and application of the IOT technology in agriculture. In: 2010 3rd International Conference on Computer Science and Information Technology, pp. 462–465 (2010)