A reconfigurable IoT-based monitoring and control system for small-scale agriculture

Tran Nhut Khaın Hoan¹, Nguyen Cao Qui¹, Nguyen Phuc Truong², Luu ng Quinh Quóc Danh¹
¹College of Engineering Technology, Can Tho University, Can Tho City, Vietnam
²College of Electrical and Electronic Engineering, Vinh Long University of Technology Education, Vinh Long Province, Vietnam

*tnkhoan@ctu.edu.vn

Abstract. In this paper, we present the design of a low-cost remote monitoring and control network for household cultivation, in which peripherals connected to each sensor-node can be re-configured remotely from the internet via a master node without any changing on the system firmware. The feasibility of the proposed scheme is evaluated via experiments. Particularly, the experiments set-up consists of one master node communicating wirelessly with some sensor nodes (slaves) to collect sensing data or send commands to them to manage the operating of the system. The sensing data from slaves will be collected and updated periodically to Google Sheets by master node via the internet connection. Additionally, slaves are equipped with versatile ports for connecting with sensors. The port is designed to be compatible with most popular communication standards such as analog, digital, I2C, UART, SPI. The key design of these communication ports is that when plugging/unplugging a sensor into/from a port, the corresponding slave can be re-configured directly on Google Sheets or keypad connected to master node. The research results can be further extended and improved to be applicable to household cultivation as well as application models of smart agriculture.

1. Introduction

In recent years, modern technology applications have been contributing to the success and outstanding development of many countries’ agriculture [1][2]. In developing countries, farming management using modern technology to increase the quantity and quality of agricultural products while optimizing the human labor required has been emerging recently [3]. Armed with such tools e.g. sensors, connectivity, software, data analytics, present-day farmers can monitor conditions of their field without even going to the field and make strategic decisions for the whole farm or for a single plant [2][3]. However, since the cost for developing a smart farming system is still quite high [4], smart farming is only feasible for large-scale cultivation. Recently, there have been significant studies on monitoring and control systems for agriculture using intelligent sensor techniques; herein, we briefly summarize the most relevant papers to this work as follows:

In [5][6], the authors implemented an automated irrigation and crop field monitoring system in which the central unit was programmed to control the irrigation motor to pump water into the field when the monitoring values of temperature and moisture exceeds the pre-defined thresholds. The target is to optimize the use of water resource and to gather the monitoring database for improving the crop yields...
in future. Similarly, in [7][8], The authors presented an IoT based monitoring and control systems used in farm production processes in which autonomous sensor-devices gather sensing data for the monitoring system. Users can also access these devices directly via the internet to control the attached actuators such as DC motors or lights. In [9][10][11], the authors investigated the IoT-based monitoring systems for smart farming in which the central unit gathers sensing data from the cultivation environment and sends them to the farmers via GPRS connection for monitoring and control.

In short, works in the literature have paid great attention to monitoring and control systems - assisted smart farming. However, the drawback of the foregoing systems is that the operation mode of the central unit and sensor-nodes (also called firmware) is pre-programmed and fixed according to the peripherals and sensors connected to them. Once the firmware is installed, these peripherals and sensors cannot be changed since the system cannot recognize the new connected devices or fail to function normally. For instance, Figure 1 shows a commercial wireless - temperature - sensor node available in the market. We can see that farmers cannot modify this device for another purpose e.g. humidity-monitoring since the installed firmware and connection-port are fixed. Even if we can plug in a humidity-sensor to the connection-port, the firmware of the node must be re-programmed.

Motivated by the aforementioned, in this work, we propose a novel design for monitoring and control systems in which peripherals and sensors connected to the sensor nodes (also called slaves) of the designed system can be re-configured remotely via the internet without the need of re-programming the system firmware. Particularly, connection ports of each slave are designed to be compatible with most popular communication standards for sensors available in the market such as analog, digital, I²C, UART, SPI. Whenever a sensor is plugging into a port or unplugging from a port, the corresponding sensor node can be re-configured directly on Google Sheets [12] or keypad connected to central control unit (also called master node).

Therefore, in the proposed network, we can re-configure the peripherals and sensors easily and much faster without any changing on the firmware of the whole system. The feasibility of the proposed scheme is evaluated via experiments. The research results can be further extended and improved to be applicable to small-scale cultivation as well as application models of smart agriculture.

Figure 1. A wireless temperature-sensor node

The rest of this paper is organized as follows. Section 2 describes the considered system and analyzes the design in details. Then, the experiment setup, results and discussion are given in section 3. Finally, the paper is concluded in Section 4.

2. System Description and Design

2.1 System Description and Assumption

The considered monitoring and control system consists of one master node and a number of sensor nodes (slaves) as shown in Figure 2. The master node communicates with slaves by using the LoRa [13] technology at the 433-MHz ISM band. The master controls the operation of the whole system e.g. searching for new slave, allocating the address for new slave, controlling the transmission of slaves to
collect monitoring data and re-configure peripherals of slaves. The master node can periodically upload the monitoring data to cloud storage e.g. Google Sheets via the Wi-Fi module ESP8266 [14]. The Google Sheets is also functioned as user interface that allows users (e.g. farmers) to manipulate the system more conveniently. Moreover, users can also control the system directly by using the keypad connected to the master.

![Figure 2. Structure of the designed system](image)

Each slave is equipped with versatile ports for connecting with sensors. Each port can be used to communicate with sensors and other peripherals by using most popular communication standards such as analog, digital, I²C, UART, SPI.

### 2.2 Hardware Design for Slave

The block diagram of a sensor node is shown in Figure 3. In experiments set-up, we design 4 communication ports in each slave. It is noted that the number of ports can be designed according to the applications. In this paper, we focus on the design of port’s pin-out in order that the port can be compatible with most popular communication protocols. The pin-out of port structure can be analyzed as follows.

![Figure 3. Block diagram of a sensor node](image)

In controlling applications, most of common sensors and peripherals often use one of the following communication protocols known as analog, digital, universal asynchronous receiver-transmitter (UART), Inter-Integrated Circuit (I²C), Serial Peripheral Interface (SPI). Signals’ specification and the most popular pin names of each protocol are summarized in Table 1.

| Protocol  | Number of signals | Signal specification |
|-----------|-------------------|----------------------|
| Analog    | 1                 | Analog               |
| Digital   | 1                 | Digital              |

Table 1. Signal specification of communication protocols
Table 1 shows that the first four protocols specify only 1 or 2 signals while SPI specifies 4 signals. To simplify the hardware design, the first 4 protocols can be integrated in the same port as shown in Figure 4; SPI protocol is placed in a separate port. In our experiments set-up, each slave is equipped with 4 ports (as shown in Figure 3) in which port 3 supports 4 communication protocols; port 1 and port 2 can be used for analog, digital and UART protocols; and port 4 is used for SPI protocol. We use the SX1278 LoRa 433 MHz module with UART interface [15] for communicating with master node and ATmega328P-AU for microcontroller. We note that the decision on how to select microcontroller much depends on the design of communication ports, supported communication protocols, applications and programming. Fortunately, most microcontroller-products in the market can support all popular communication protocols nowadays.

![Figure 4. Port pin-out for Analog/ Digital/ UART/ I²C Protocol](image1)

2.3 Hardware Design for Master Node
The block diagram of a master node is shown in Figure 5. The master controls the operation of the whole system as well as interacts with users. Users can control the system e.g. reconfigure slave’s hardware, remove slaves from system, mainly via the interface on Google Sheets. Additionally, users can also use the keypad connected directly to the master. In practical monitoring and control applications, the master node serves as a base station that is used to buffer the monitoring data sent from slaves, then upload the data to cloud storage. Thus, the master should be equipped with large program and data memories. In the experiments, we use the ATmega2560 chip (256 KB Flash, 8KB SRAM and 4 KB EEPROM) for the microcontroller in the master node. The SX1278 LoRa 433 MHz module with UART interface is also used for communicating with slaves. Additionally, the master node is equipped with a 0.9” OLED display to show the operating status of the system and support users when they configure the system using keypad.

![Figure 5. Block diagram of the master node.](image2)
2.4 User Interface
Generally, after setting up the system, to change or remove a passive device (e.g. a sensor) connected to a slave’s port or plug in a new passive sensor without re-programming the firmware of the system, the hardware specifications of the port should be designed in such a way that it can be redefined by software. Below, we suggest two approaches to define the port:

a. Physical specifications, parameters and communication protocol of each port are defined and stored in the slave’s EEPROM and this specification data can be synchronized with the data form designed on Google Sheets. By this way, users can modify the physical specification of each slave’s port according to the device plugged into that port.

b. Designers can build a library including the drivers for most popular sensors or devices that are available in the market and install this library in each slave. By this method, users can call the driver corresponding to the device plugged in to a slave’s port via Google Sheets.

We note that the first method seems more complicated. However, by using this method, we can even install any new device if the physical specifications of that device is compatible with the one users want to install. The second method is much easier, but it costs more memory installed at slaves to store the library. For quick implementation, we used the second method for our experiments.

Figure 6 shows the user interface in our experiments that allows users to control or configure the peripherals equipped at slaves. We recall that, in the experiments, we build 3 slaves equipped with 4 communication ports as listed in Figure 6. Herein we build drivers for 5 different sensing devices and install them on each slave. The operation of the system can be described as follows.

Each slave in the system can be turned on or off anytime by pressing the corresponding power button on the user interface. Similarly, the device connected to any port can be plugged or unplugged by pressing the ON/OFF button corresponding to that port. The physical specification of each slave’s port consists of 3 basic parameters: port’s status (ON or OFF), device’s name and the collected data length in bytes. The values of these parameters are stored in the EEPROM of the slave (called Configuration table). Based on the information in the table, the slave will select the right driver from its library for the corresponding port. Any changing on the slave’s hardware configuration can also be done via user interface as shown in Figure 6.

For instance, to change a sensing device connected to a specific port, after plugging the new device in, users (e.g. farmers) can reconfigure that port by modifying the device’s name and data length fields of the device of the corresponding slave’s port on the user interface. After finishing the configuration process, master node will collect the configuration parameters on the Sheets; subsequently, it will synchronize them with the corresponding slave’s configuration table. Likewise, users can change and reconfigure the device connected to any other port of any other slave without re-programing the firmware of the system.
2.5 System Communication Protocols

This section will briefly describe the communication protocol among master node and slaves. Figure 7 shows the transmission frame of master node consisting of 3 fields. The command field is always 1 byte in length that is used to identify the frame types defined as searching for new slave (called S-Frame), monitoring and control (called M-Frame), and slave's peripheral reconfiguration (called R-Frame). The contents of the command field are detailed as follows.

a. The S-Frame only contains Command field 1 byte used to search and assign address for a new slave;
b. The M-Frame is related to monitoring and control commands e.g. collect monitoring data, send a request to a specific slave, In the M-Frame, the next field contains the address of targeted slave. The last field indicates the command to the corresponding slave. So, the M-Frame are 3-bytes in length;
c. The R-Frame includes a set of commands that allows users to reconfigure the slave’s hardware. This frame is 4-bytes in length in which the last field indicates the sensor-name with its connection port number (1 byte) and collecting-data length (1 byte) of that sensor.

| Command field | Address field | Command’s specification field |
|---------------|---------------|------------------------------|
| 1 (Byte)      | 0 - 1 (Byte)  | 0 - 2 (Bytes)               |

Figure 7. Transmission frame of master node

| Address field | Information field | Block Check Character (BCC) |
|---------------|-------------------|-----------------------------|
| 1 (Byte)      | 1 – 28 (Bytes)    |                             |

Figure 8. Transmission frame of slave

The communication in the system is the transmission-frame-exchange process between master node and a specific slave node under the control of the master. We end this section by presenting a short communication flowchart between master and a slave as shown in Figure 9. In the first frame sent from master, the field start is used to identify the command from master to the slave #1 addressed in the second field. The last field is the detail that requests the slave to collect sensing date from its sensing devices. Slave #1 acknowledges by sending its address to the master and collects the requested data. The second frame from master is the request slave #1 to send the collected data; then, the slave sends its data frame to the master.

Figure 9. The communication flowchart between master and slave #1
3. Result and Discussion

In experiments, we implement the network consists of one master node and 3 slave nodes that is the same as the system model shown in Figure 2. Physical specifications of each slave and master node were described in Section 2.2 and Section 2.3, respectively. The sensing devices connected to each slave’s port were implemented as described in Figure 6. We recall that only Port 4 of slave 1 can be used for SPI protocol. The implemented system is shown in Figure 10. The connection among master node and slaves is based on the LoRa SX1278 433-MHz RF modules, while the master connects to the Google Sheets via Wi-Fi connection.

![Figure 10. The implement of sensing devices at slaves](image)

First of all, we connect the sensing devices to the corresponding slave’s port. Then, on Google Sheets, we enter the name of each item, the status and the data length of the sensing device to the corresponding spot. After finishing hardware set-up, the master will fetch the configuration specifications to data storage; then deliver this configuration to the corresponding port of each slave. Then, the master will periodically control the system to collect the sensing data and upload that data to store on Sheets.

![Figure 11. The record of uploading data on Google Sheets](image)
shows the recorded data uploaded to Google Sheets. We can see that slave 1 has 4 sensing devices connected to 4 ports, so 4 columns Port 1, Port 2, Port 3 and Port 4 all have data uploaded. Meanwhile, slave 2 and slave 3 only have data uploaded at Port 2 and Port 1, respectively. The results indicate that the operation of the system were well-matched with the configuration on the Sheets. Likewise, based on the foregoing implementation of slaves’ sensing devices, we also verified the scenarios when we unplug a sensor from a specific port and plug that sensor into another port; then the hardware specifications can be modified very quickly and easily via the interface on Google Sheets.

4. Conclusion
In this paper, we proposed a new design for a low-cost remote monitoring and control network. The key design of the proposed method is that the peripherals connected to each sensor-node can be reconfigured remotely from the internet via a master node without any changing on the system firmware. Since the reconfiguration is very simple and quickly, this can even be done by normal users without any technical support. The feasibility of this work is evaluated via experiments. The results show that this study can be further extended and improved to be applicable to small-scale cultivation in future.

5. Acknowledgments
This study is funded in part by the Can Tho University Improvement Project VN14-P6, supported by a Japanese ODA loan. This is also partially supported by Science Research Program of Can Tho University. We’d also like to acknowledge Ms. N.H.M. Nhi and Mr. T.T. Kha for their help to set-up the experiments.

6. References
[1] Ayaz, M., Uddin, A., Sharif, Z., Mansour, A., Aggouné, el-Hadi. “Internet-of-Things (IoT)-Based Smart Agriculture: Toward Making the Fields Talk.” IEEE Access. PP. 1-1, 2019, doi = {10.1109/ACCESS.2019.2932609}
[2] Imran Ali Lakhiar, Gao Jianmin, Tabinda Naz Syed et al. “Monitoring and Control Systems in Agriculture Using Intelligent Sensor Techniques: A Review of the Aeroponic System”, Journal of Sensors, Vol. 2018, Dec 2018, DOI = 10.1155/2018/8672769
[3] Navulur, S. and Sastry, A.S.C.S. and Prasad, M.N. “Agricultural Management through Wireless Sensors and Internet of Things.” International Journal of Electrical and Computer Engineering. 7. 3492-3499. Dec 2017. doi = {10.11591/ijece.v7i6.pp3492-3499}
[4] https://www.letsnurture.com/how-much-it-will-cost-to-develop-an-iot-based-proof-of-concept-for-smart-farming-system.html
[5] Dr. M.Yuvaraju, K. J. Priyanga, “An IOT Based Automatic Agricultural Monitoring and Irrigation System,” International Journal of Scientific Research in Computer Science, Engineering and Information Technology, Vol. 4, 5 pp. 58-65, April 2018. 10.1109/IECON.2008.4758346
[6] Rajalakshmi.P, Mrs.S.Devi Mahalakshmi, “IOT Based Crop-Field Monitoring And Irrigation Automation.” 2016 10th International Conference on Intelligent Systems and Control (ISCO), Coimbatore, 2016, pp. 1-6.
[7] Markovic, Dusan & Koprivica, Ranko & Pešović, Uroš & Randić, Siniša. (2015). "Application of IoT in monitoring and controlling agricultural production." Acta agriculturae Serbica. Vol. 20. pp. 145-153. 10.5937/AASer1540145M.
[8] Patil, Dr & Jadhav, Madhuri. (2019). “Smart Agriculture Monitoring System Using IoT.” *International Journal of Advanced Research in Computer and Communication Engineering (IJARCE).* Vol. 8. 4. pp. 116-120. 10.17148/IJARCE.2019.8419.

[9] S. R. Prathibha, A. Hongal and M. P. Jyothi, "IOT Based Monitoring System in Smart Agriculture," *2017 International Conference on Recent Advances in Electronics and Communication Technology (ICRAECT)*, Bangalore, 2017, pp. 81-84.

[10] R. Mythili, Meenakshi Kumari, Apoorv Tripathi, Neha Pal, "IoT Based Smart Farm Monitoring System," *2019 International Conference on Recent Advances in Electronics and Communication Technology (ICRAECT)*, Vol. 8, Issue 4, 10.35940/ijrte.D8806.118419

[11] T. Baranwal, Nitika and P. K. Pateriya, "Development of IoT based smart security and monitoring devices for agriculture," *2016 6th International Conference - Cloud System and Big Data Engineering (Confluence)*, Noida, 2016, pp. 597-602.

[12] https://docs.google.com/spreadsheets

[13] "LoRa  Modulation Basics", *Semtech*, 2019, https://web.archive.org/web/20190718200516/https://www.semtech.com/uploads/documents/an1200.22.pdf

[14] ESP8266 Technical Reference: https://www.espressif.com/sites/default/files/documentation/esp8266-technical_reference_en.pdf

[15] “The SX1278 LoRa 433 MHz module with UART interface”, *2015 Semtech Corporation*, https://www.elecrow.com/download/sx1276.pdf