ICIC Express Letters
Part B: Applications
Volume **7**, Number **4**, April **2016**

ICIC International ©2016 ISSN 2185-2766
pp. **819–826**

AN ENVIRONMENT MONITOR APPLICATION PROGRAM WITH IOT BRIDGE FUNCTION

Chung-Wen Hung*, Yuan-Hao Lai and Hong-Yu Chen

Department of Electrical Engineering
National Yunlin University of Science and Technology
No. 123, University Rd., Sec. 3, Douliou, Yunlin 64002, Taiwan
*Corresponding author: wenhung@yuntech.edu.tw

Received September 2015; accepted November 2015

Abstract. *For the past few years, environment monitor is widely applied in different kinds of fields like agriculture, industry or family. A complete environment monitor system includes hardware monitor device and software application program. In this paper, our purpose is to develop a multifunction environment monitor application program by using visual studio c# based on window OS, it can visually display the environment monitor data on application program, and support user different kinds of applications like warning list, history data research and device parameter setting. Otherwise, Modbus Protocol is implemented in our application program to strengthen the stability and universality of data transport. Cloud database connection is also implemented to let our application program with Internet of thing (IoT) function.*

Keywords: Environment monitor, Software application program, Internet of thing, Visual studio c#

1. **Introduction.** Environment monitor is a technology to monitor the quality and situation of the environment. It is widely applied in different fields such as agriculture, industry or family. Because the wireless sensor network (WSN) technology and Internet of thing (IoT) flourish these years, it is easier to implement environment monitor system by using WSN technology such as Bluetooth, Zigbee or Sub1G [1]. There are many researches of environment monitor systems discussed in [2-4], but these papers just only focused on hardware. However, besides hardware devices, a monitor application program is also important. A powerful application program not only provides observation of the each sensor node situation but also transfers sensing data to cloud database. In [5], an application program was proposed and it played a very import role. The program can provide real-time monitoring and display of the parameters of the engine. However, it could not send data to cloud database.

In this paper, environment monitor application program is proposed and it is coded in Visual studio c#. Visual studio c# is an object-oriented, visualization application programming language, which inherits advantages of visual studio's such as a full-featured code editor, compiler, project templates, designers, code wizards, a powerful and easy-to-use debugger. Programmers can more easily develop serial communication and Internet transmission with the transmission function in Visual c#. The Modbus Protocol, an industrial standard, is implemented in the WSN bridge function, and a cloud database is adopted for IoT function. And the multi-thread mechanism is used in Internet communication to avoid any program delay caused by network congestion.

The program architecture and function will be introduced in Section 2; and the technology which is implemented in program will be discussed in Section 3; next, the experimental results are showed in Section 4; finally, the conclusion will be discussed and future works are also described, here.

2. Application Program Architecture and Function.

2.1. Application program architecture. There are two versions in the proposed environment monitor application program. One is named main environment monitor application program (MAP), which can collect the environment data from bridge and update cloud database; and the other one is named branch environment monitor application program (BAP), which just only provides the display function for the environment data downloaded from cloud database. MAP can receive data from a bridge through serial communication if its communication follows Modbus Protocol. When MAP receives a new data from bridge, MAP will display on computer screen and transfer the data to the cloud database. The BAP could get the new data from cloud database and then show the new data on other computer screen. The application program architecture is shown in Figure 1.

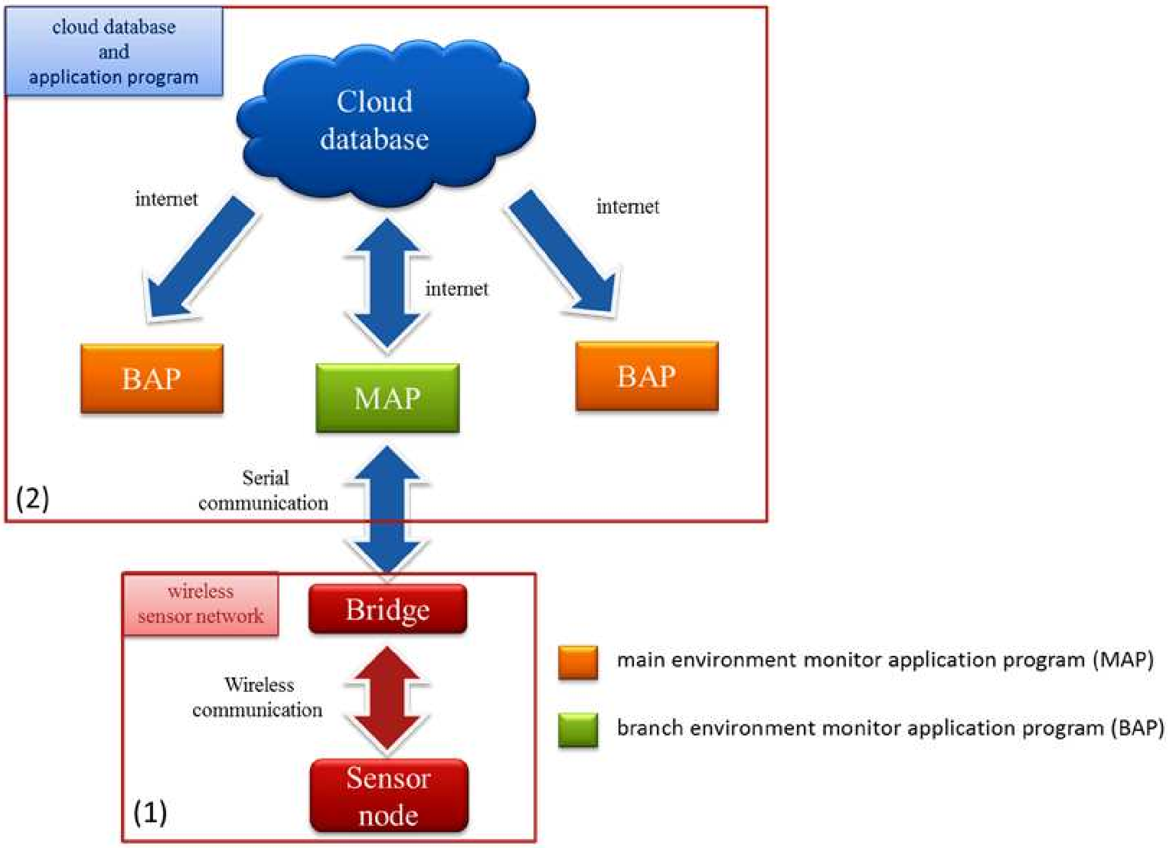

cloud database and application program
Cloud database
internet
internet
internet
BAP
MAP
BAP
Serial communication
(2)
wireless sensor network
Bridge
Wireless communication
Sensor node
(1)
main environment monitor application program (MAP)
branch environment monitor application program (BAP)

FIGURE 1. Application program architecture: (1) wireless sensor network and (2) cloud data-base with application program

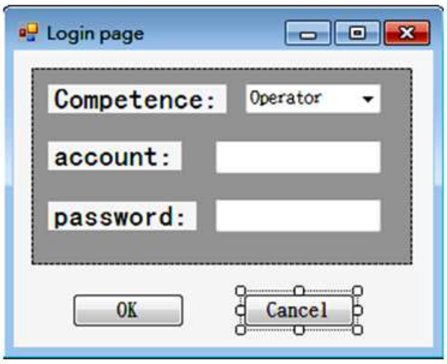
2.2. Application program function. Some software functions are included in both MAP and BAP, such as main page for environment monitor data display, warning list table, warning threshold parameter setting, data and warning history inquiry, and user login. However, serial communication setting and bridge connect setting are only implemented in MAP. The details of the functions are described below:

A. Main page: The main of the proposed application is used to show the measurement environment map and the sensor data of each node. The background picture file can be assigned according to user's requirement and real system installation. In the screen, the locations of sensor nodes and the popup windows which are used to display environment data also can be dynamically assigned. A user can switch to other function pages here, too.

B. Warning list page: Users will be informed, if the environment value is over or under the setting threshold. The limitation range is set in advance in the limitation range setting page, and the detail will be discussed later. It will display the ID number of sensor node, the occurrence time of an event, and the even value. This page also shows the message when the alarm frees.

C. Trend chart page: The trend of environment parameters is displayed in trend chart page after assigns of sensors. Not as in warning list page, users only read the measurement value by number, but can trace the data changing by chart. The time coordinate axis will be adjusted automatically to show the data in suitable resolution.

D. Other function pages: The program also includes other function pages, such as:
As shown in Figure 2, in login page, a user can login and get the authority to set threshold parameters and access the cloud database, MySQL. In sensor and threshold setting page, a user can set the display sequence and threshold of sensors, as shown in Figure 3. A user can set the serial communication parameter in serial communication setting page, as shown in Figure 4. In Historical Query page, a user can get output data and warning history information in this page, as shown in Figure 5.

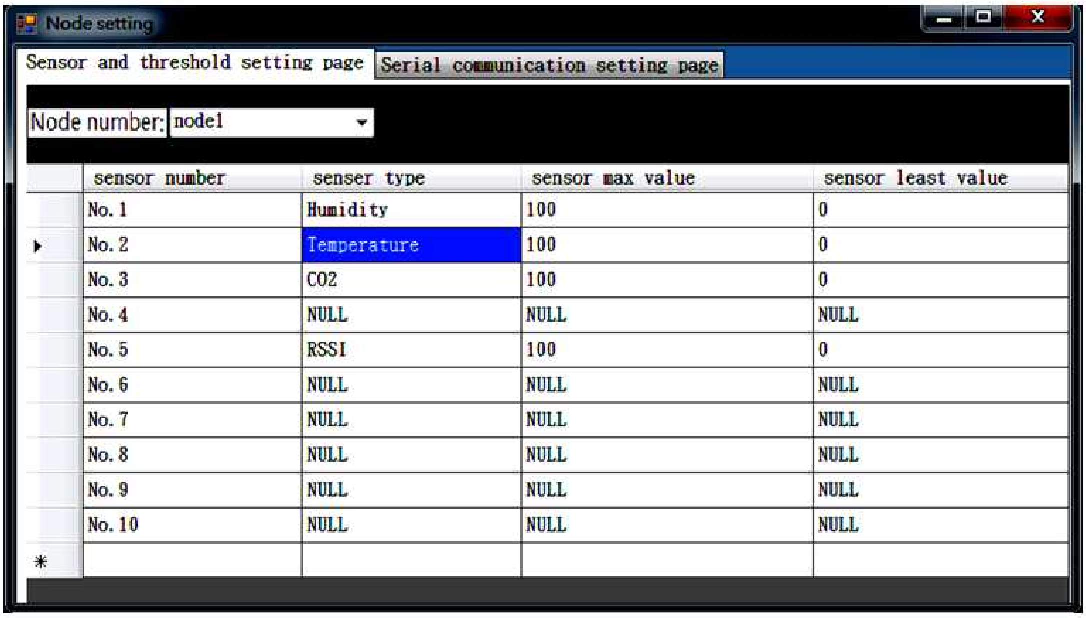


Competence: Operator

account:

password:

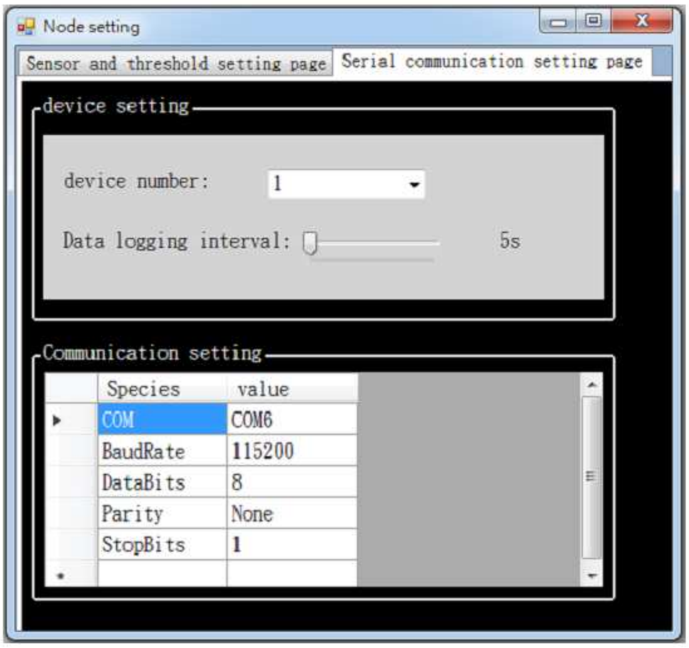
Figure 2. Login page


Node setting
Sensor and threshold setting page
Serial communication setting page

Node number: node1

	sensor number	senser type	sensor max value	sensor least value
	No. 1	Humidity	100	0
▶	No. 2	Temperature	100	0
	No. 3	CO2	100	0
	No. 4	NULL	NULL	NULL
	No. 5	RSSI	100	0
	No. 6	NULL	NULL	NULL
	No. 7	NULL	NULL	NULL
	No. 8	NULL	NULL	NULL
	No. 9	NULL	NULL	NULL
	No. 10	NULL	NULL	NULL
*				

Figure 3. Sensor and threshold setting page


Node setting
Sensor and threshold setting page
Serial communication setting page

device setting

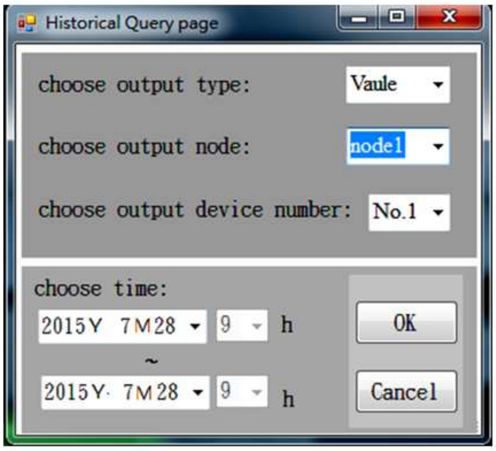
device number: 1

Data logging interval: 5s

Communication setting

Species	value
COM	COM6
BaudRate	115200
DataBits	8
Parity	None
StopBits	1

Figure 4. Serial communication setting page



choose output type: Vaule

choose output node: node1

choose output device number: No.1

choose time:

2015Y 7M28 9 h

~

2015Y 7M28 9 h

Figure 5. Historical Query page

3. **Application Program Technology Implementation.** In order to be compatible with other application and industrial devices, two technologies are implemented in this program. First is Modbus Protocol what is used in the communication between application program and bridge, and the other is the connection of cloud database. Both of them need to be developed in multi-thread, and details are discussed as follows.

3.1. **Modbus Protocol introduction.** Modbus Protocol is a communication protocol that is widely used in industry. It has two advantages, which are simple implementing and the stability of data transmission. Thus, Modbus is used in this program to communicate with a bridge. Modbus is a master/slave type protocol: the proposed application program is the master, and sensor nodes are slaves. Each slave has a unique ID address, and a normal Modbus command includes an ID address of the target device. When master sends a Modbus command, the sensor node which is indicated by the ID will execute the order and the response. All Modbus commands include CRC check code which is a function to make sure whether the command is lost or destroyed.

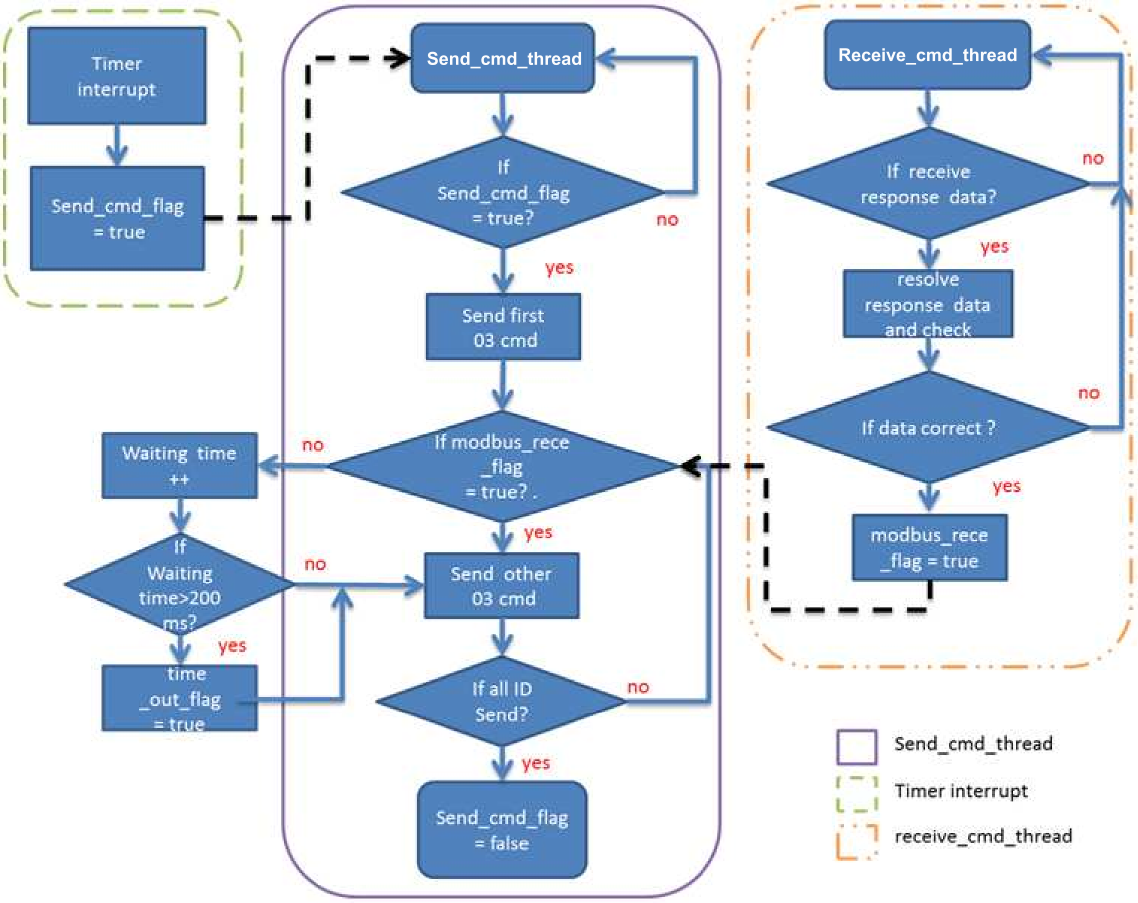
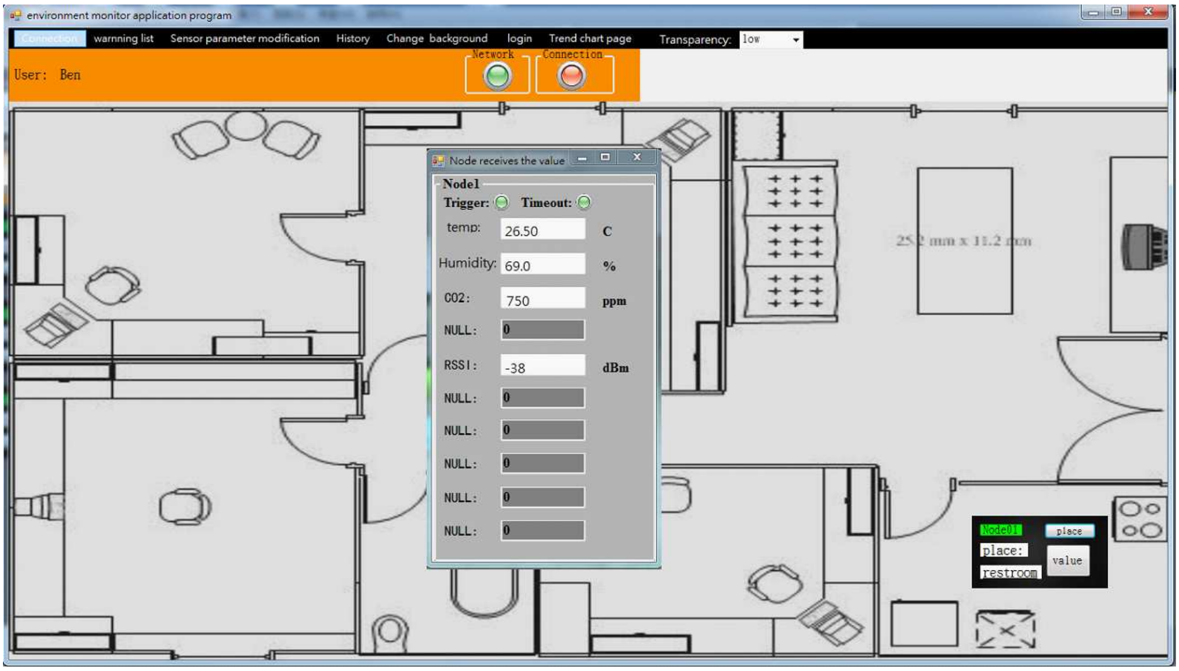


Figure 6. Modbus thread operate processes

3.2. **Modbus Protocol software implementation.** Because of Modbus Protocol based on serial communication, the first thing is to declare serial communication manager object in order to handle serial communication. In the program, two threads and one timer interrupt is used to construct Modbus Protocol communication. One thread sends the Modbus command and the other one receives the message response from a bridge. Timer interrupt is used to send trigger command frequently. When the timer runs to updating limitation, timer interrupt will be triggered and raise a send_command_flag. Then the send command thread will start to send the Modbus command to bridge and wait for the receiving thread. In receive thread, it will distinguish different bridge response command with time delay and resolve the CRC and data by using switch function. If the CRC is correct, program will renew the data and rise modbus_send_flag again. Two threads repeat the same work until all sensor node ID addresses are sent. The Modbus thread operate processes are shown in Figure 6.

3.3. **Cloud database connection.** In this paper, the cloud database of the proposed program is based on MySQL database. It is necessary to include MySQLDriverCS library and function class in C#, if program needs to connect to the cloud database. A database named "environment" is created and a lot of tables in this database to store the environment data from bridge. Every table is mapped to one sensor node. According to Modbus Protocol, the max ID address is 255, and it means that there are 255 tables created in environment database. Besides, environment database also includes other tables to store else parameters such as the frequency of renew monitor data, and serial communication parameters. When program needs to send or get data from cloud database, it has to create a new thread for handling it, because it takes a little time to transmit information to cloud database. Without the thread, users may misunderstand that the system has shutdown. By thread scheme, the cloud database connection will work at background and resolve the above problem.

4. **Experimental Result.** An experiment is installed in 5m × 10m room, there are 3 sensor nodes and one bridge, and the Sub1G is used to connect nodes and bridge. And the proposed program is connected to the bridge via RS-232 and Modbus. Sensor nodes are dispersed around the room, and they will sense the environment data such as temperature, humidity, and CO_2 concentration. The application program sends a command through bridge to sensor nodes every 5 seconds, and then sensor nodes will respond environment data. In stability experiment, the proposed application program can steadily receive



Transparency: low

User: Ben

Node1

Trigger: ● Timeout: ●

temp: 26.50 C

Humidity: 69.0 %

CO2: 750 ppm

NULL: 0

RSSI: -38 dBm

NULL: 0

NULL: 0

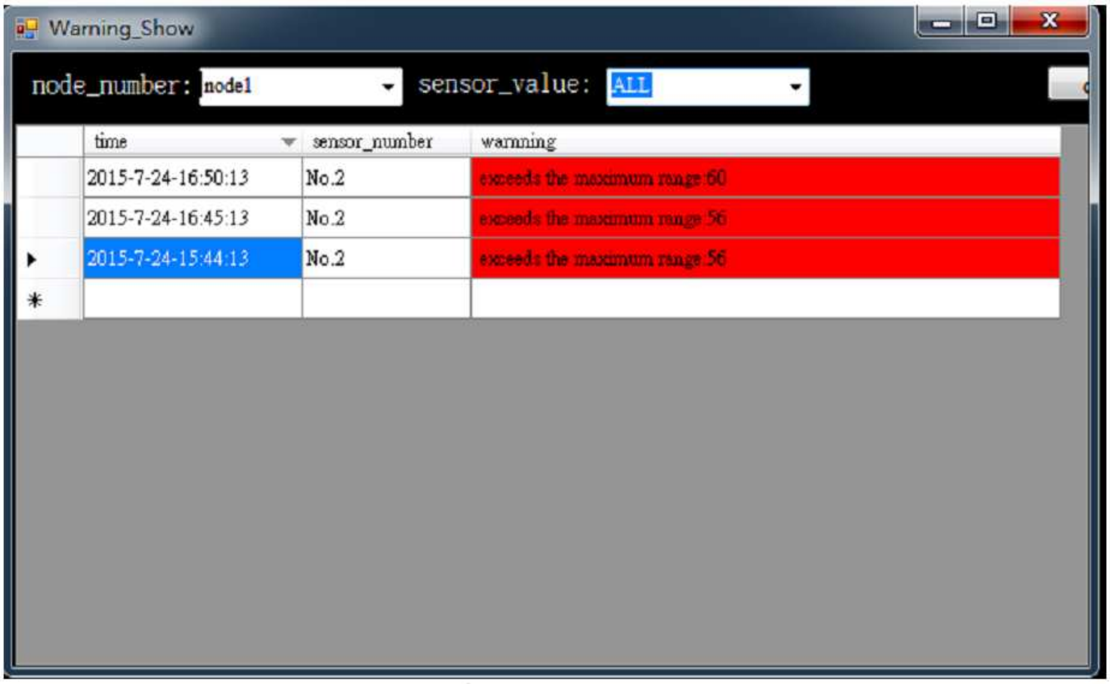
NULL: 0

NULL: 0

NULL: 0

place: restroom

FIGURE 7. Main page showing the monitor environment data



node_number: node1 sensor_value: ALL

	time	sensor_number	warnning
	2015-7-24-16:50:13	No.2	exceeds the maximum range:60
	2015-7-24-16:45:13	No.2	exceeds the maximum range:56
▶	2015-7-24-15:44:13	No.2	exceeds the maximum range:56
*			

FIGURE 8. Warning list page

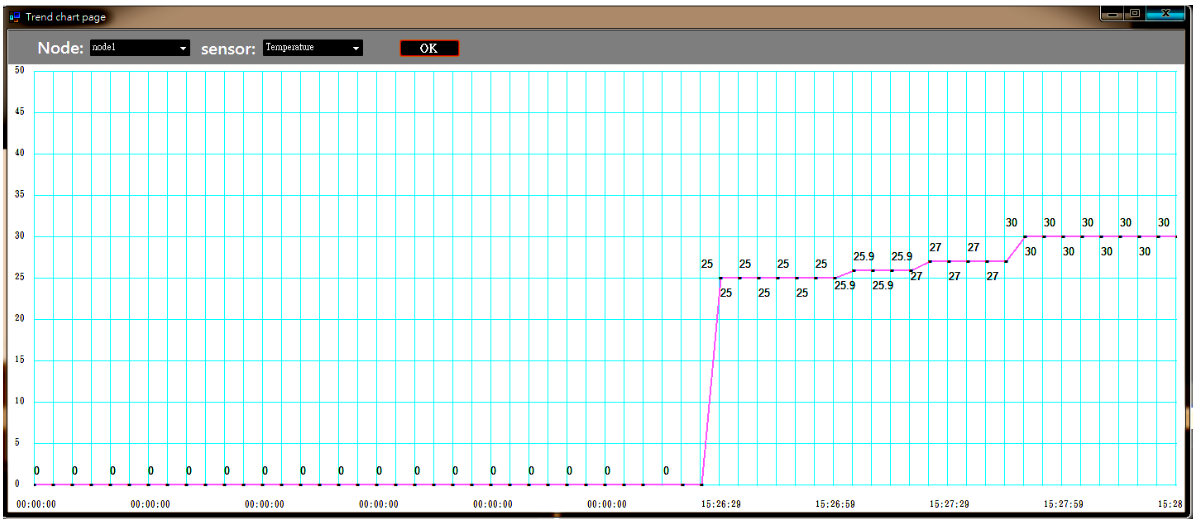
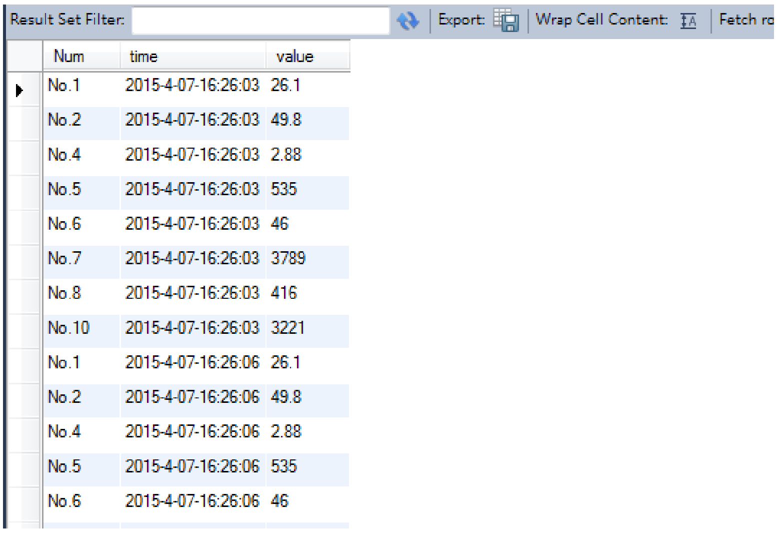


FIGURE 9. Trend chart page



Num	time	value
No.1	2015-4-07-16:26:03	26.1
No.2	2015-4-07-16:26:03	49.8
No.4	2015-4-07-16:26:03	2.88
No.5	2015-4-07-16:26:03	535
No.6	2015-4-07-16:26:03	46
No.7	2015-4-07-16:26:03	3789
No.8	2015-4-07-16:26:03	416
No.10	2015-4-07-16:26:03	3221
No.1	2015-4-07-16:26:06	26.1
No.2	2015-4-07-16:26:06	49.8
No.4	2015-4-07-16:26:06	2.88
No.5	2015-4-07-16:26:06	535
No.6	2015-4-07-16:26:06	46

FIGURE 10. Monitor environment data saved in cloud database

environment data and send environment data to cloud database. Other functions can also work correctly. The experimental results are shown from Figures 7 to 10, and they indicate that the proposed program works well.

5. **Conclusion.** An environment monitor application program with IOT bridge function is proposed and implemented in this paper. Users can monitor the environment data clearly on computer screen, the background picture and sensor node location can be changed depending on real situation. Modbus Protocol is implemented in this paper, and is used to connect to a bridge. Besides, cloud database connection is also implemented in this paper. Moreover, the environment data history and trend cloud be searched and displayed, and also could be output in MS-Excel file format. The proposed program will be suitable for IoT systems. Some future research directions are considered: the program in the IOS or Android operation system is developed for portable devices, and a long distance remote controller is also studied for a smart home system.

Acknowledgment. This work is partially supported by the Ministry of Science and Technology, Taiwan, under contract No. NSC 103-2221-E-224-056-.

REFERENCES

[1] K. Piotrowski and S. Peter, Sens4U: Wireless sensor network applications for environment monitoring made easy, *The 4th International Workshop on Software Engineering for Sensor Network Applications*, pp.37-42, 2013.

[2] A. De Sarkar, An environment monitoring interface using GRASS GIS and Python, *The 3rd International Conference on Emerging Applications of Information Technology*, pp.235-238, 2012.

[3] V. Kotovirta, Building an application framework for monitoring the environment, *IEEE International Conference on Geoscience and Remote Sensing Symposium*, pp.2149-2152, 2006.

[4] O. Bjelica and S. Lale, Development environment for monitoring, data acquisition and simulation of PLC controlled applications, *The 21st Telecommunications Forum*, pp.912-915, 2013.

[5] D. Tan, J. Wang and H. Liu, The development of electronically controlled engine monitoring system based on VB, *2011 International Conference on Mechatronic Science, Electric Engineering and Computer*, pp.177-180, 2011.