愛知工業大学研究報告
第 55 号　令和 2 年

Wireless Environmental Data Tracking System Design and Evaluation with a Multifunctional Device

多機能型デバイスを用いたワイヤレス環境センシングデータベースシステムの設計

福澤和久 †
Fukuzawa Kazuhisa †

Abstract Wireless environmental monitoring devices are in demand. These devices, which are also referred to as IoT (Internet of Things) devices, will be of vital importance in the next decade. The objective of this study is to propose a prototype system that can collect the environmental data using sensors, and transfers the data using a wireless network to a remote database for storage. The system was tested and verified by confirming that the monitoring data was stored in the remote database server. As a result, despite the set temperature of 20 degrees, the temperature fluctuated between 21 to 23 degrees. Our future work will involve using more sensors in the device so that it can collect multiple data, such as humidity, images, and the location. Furthermore, the author needs to deploy an actuator for physical control and improve the system's power supply so that it can be used outdoors.

1. Introduction

Wireless sensor network devices, which are also referred to as IoT (Internet of Things) devices, are emerging as devices of vital importance to various industries. As the number of transistors in a densely integrated circuit has doubled, the price of sensors has also reduced over time, which is Moore's law [1],. Consequently, IoT devices are becoming popular because of their low-cost and robustness.

IoT devices are used in many industries, like in agriculture, where they are used to improve productivity through monitoring environmental data, evaluating the suitable cultivating methods, and actuating the devices [2]. Manufactural industry is one of the most essential industry regarding the use of IoT devices.

† Aichi Institute of Technology, Toyota 470-0392

Many developments have been made in designing wireless sensor network systems using a Raspberry Pi [3]or Arduino [5] for environmental monitoring applications [4].

They incorporate sensors, actuators, wireless modules, and applications into the system.

There are challenges with using single-board devices, such as Raspberry Pi, to implement a database within the device [2] [4].

The author's approach is to deploy a system with an external database, where the device sends data through a wireless network connection, which is stored in the external database.

This paper describes the design of a wireless environmental sensor data tracking system with a multifunctional device comprising a Raspberry Pi and a temperature sensor. The objective of this paper is to propose a system that can collect the environmental data using a sensor and transfer and store the data through a wireless network to an external database.

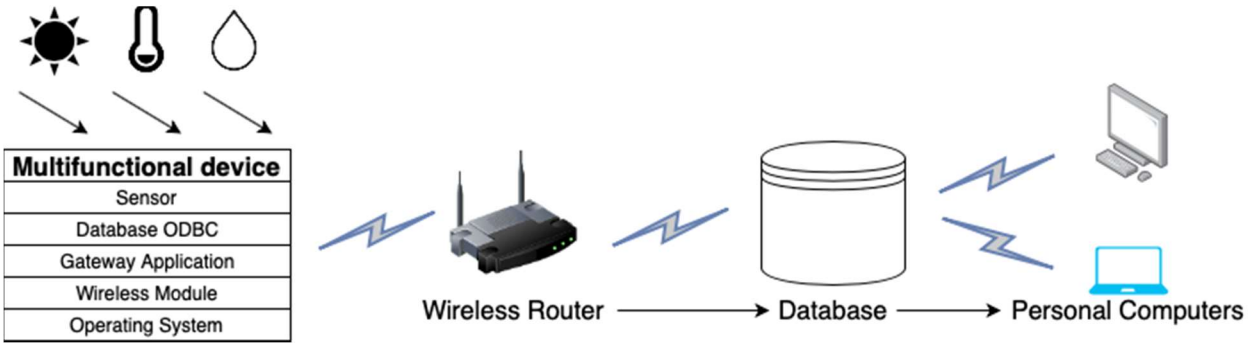

Fig. 1 Overall System Architecture

This paper consists of 5 sections. Section 2 describes the overall system architecture. Section 3 describes the design of the hardware and components. Section 4 shows the results of sensor data connecting with the external database through a wireless network, confirming if the sensor data was captured and uploaded to the database. Finally, the paper is concluded with a summary and suggestion for future work in Section 5.

2. Overall System Architecture

Fig. 1 shows the overall system architecture of the wireless environmental sensor data tracking database system that the author has developed. The system includes a multifunctional device that has an operating system, wireless module, gateway application, database ODBC, and a sensor. First, for detecting environmental data, the author uses a temperature sensor. Second, to extract the data, the author employs data extraction programming written in Python [6]. Third, the data is sent using a database ODBC (a driver that can connect to the database system) and a wireless module connecting to a wireless router. The router sends the data to an external database, which is developed using a relational database management system (RDBMS) PostgreSQL [7]. The device works as a wireless sensor node. The database system stores the sensing data uploaded from the sensor node. The author then verifies that the stored data from the node connected to the database.

The author choses Raspberry Pi Model B as the sensor node shown in Fig. 2. Raspberry Pi is a popular IoT device, which is a small single-board computer developed by the Raspberry Pi Foundation for learning basic computer science and electronics engineering. The author adopted Raspbian as the operating system of Raspberry Pi, which is the most popular operating system of Raspberry Pi, based on a Debian OS, a distribution of Linux.

A wireless module is already implemented in Raspberry Pi3, based on the Wi-Fi IEEE 802.11 wireless standard and operating in the 2.4 GHz b/g/n single band.

Fig. 2 Multifunctional device (Raspberry Pi Model B)

Fig. 3 shows DS18B20 temperature sensor. The sensor is a digital thermometer that provides 9-bit to 12-bit Celsius temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18B20 communicates over a 1-Wire bus that requires only one data line (and ground) for communication with a central microprocessor. Additionally, the DS18B20 derives power directly from the data line (“parasite power”), eliminating the need for an external power supply [8].

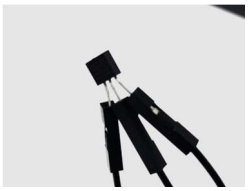

Fig. 3 Temperature sensor (DS18B20)

Fig. 4 is the database server in a remote location. The database server is connected using a wireless local area network (WLAN). In this study, the author adopted a PostgreSQL Database as a relational database management system. PostgreSQL is an open-source application, which is used in many situations. The server is operated by Windows Server 2019 and has other database instances running.

Fig. 4 Database Server (PostgreSQL)

3. Design of the Hardware and Components

3-1 Design of Hardware

The author designed the temperature sensor DS18B20 with the pin configurations shown in Fig. 5, and it was implemented as a sensor node by wiring the pins according to the GPIO (general-purpose input/output) pins of Raspberry Pi shown in Fig. 6.

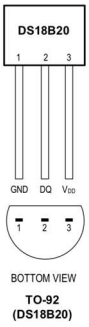

Fig. 5 DS18B20 Pin Configurations

Source: maxim integrated (maximintegrated.com)

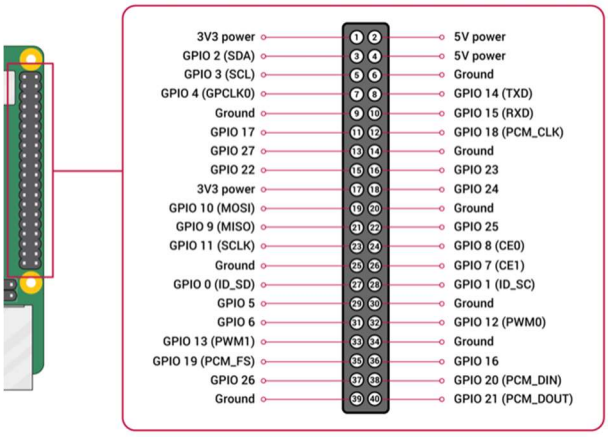

Fig. 6 GPIO pins

Source: Raspberry Pi Foundation

3-2 Design of Software

The software part was developed on Raspbian using the programming language Python version 3 series and the automatic task scheduling system crontab.

The author developed a process for the temperature data while transforming the data using Python, and upload the data to database. Furthermore, a process was developed that uses crontab to execute the program at regular intervals (in this study, every 10 minutes).

3-3 Database system

The database was designed by PostgreSQL. Since this study concerns the development of a prototype, the author used the minimum configuration shown in Table. 1. Data acquisition date (created_on) in column A is the primary key, and column B is the temperature.

Table. 1 Database table design

redtdb	
PK	created_at
	temperature

4. Experimental Results

4-1 Experimental methodology

The author conducted an experiment of sensing the room temperature using the device that the author designed. The experiment was conducted from 10:00 am to 17:00 pm in the laboratory of the Aichi Institute of

Technology on December 12, and the air conditioner was set to 20 degrees for the duration of the experiment. The device was placed in the center of the room.

The experimental objectives are as follows: whether the temperature can be measured, whether data can be acquired every 10 minutes, and whether data has been uploaded to the database.

4-2 Sensing and connection data result

Fig. 7 shows the execution confirmation every 10 minutes using crontab in Raspbian. From the output of the execution log, the connection was successful.

Fig. 8 shows the connection to the database from another computer. From (a), the author can confirm that the connection was successful. Furthermore, the author confirmed that the sensing data from the multifunctional device was sent via the wireless router to the database from (b).

```
pi@raspberrypi:~ $ tail -f /var/log/cron.log | grep python
Dec 12 10:30:01 raspberrypi CRON[12079]: (pi) CMD (/home/pi/.pyenv/versions/
3/bin/python ~/dev/temp_cap/insert_temp.py)
Dec 12 10:40:01 raspberrypi CRON[12830]: (pi) CMD (/home/pi/.pyenv/versions/
3/bin/python ~/dev/temp_cap/insert_temp.py)
Dec 12 10:50:01 raspberrypi CRON[13595]: (pi) CMD (/home/pi/.pyenv/versions/
3/bin/python ~/dev/temp_cap/insert_temp.py)
Dec 12 11:00:01 raspberrypi CRON[14372]: (pi) CMD (/home/pi/.pyenv/versions/
3/bin/python ~/dev/temp_cap/insert_temp.py)
Dec 12 11:10:01 raspberrypi CRON[15124]: (pi) CMD (/home/pi/.pyenv/versions/
3/bin/python ~/dev/temp_cap/insert_temp.py)
Dec 12 11:20:01 raspberrypi CRON[15912]: (pi) CMD (/home/pi/.pyenv/versions/
3/bin/python ~/dev/temp_cap/insert_temp.py)
Dec 12 11:30:01 raspberrypi CRON[16749]: (pi) CMD (/home/pi/.pyenv/versions/
3/bin/python ~/dev/temp_cap/insert_temp.py)
Dec 12 11:40:01 raspberrypi CRON[17563]: (pi) CMD (/home/pi/.pyenv/versions/
3/bin/python ~/dev/temp_cap/insert_temp.py)
Dec 12 11:50:01 raspberrypi CRON[18351]: (pi) CMD (/home/pi/.pyenv/versions/
3/bin/python ~/dev/temp_cap/insert_temp.py)
Dec 12 12:00:01 raspberrypi CRON[19109]: (pi) CMD (/home/pi/.pyenv/versions/
3/bin/python ~/dev/temp_cap/insert_temp.py)
Dec 12 12:10:01 raspberrypi CRON[19932]: (pi) CMD (/home/pi/.pyenv/versions/
3/bin/python ~/dev/temp_cap/insert_temp.py)
Dec 12 12:20:01 raspberrypi CRON[20696]: (pi) CMD (/home/pi/.pyenv/versions/
3/bin/python ~/dev/temp_cap/insert_temp.py)
Dec 12 12:30:01 raspberrypi CRON[21476]: (pi) CMD (/home/pi/.pyenv/versions/
3/bin/python ~/dev/temp_cap/insert_temp.py)
```

Fig. 7 Execution confirmation every 10munites using crontab in Raspbian

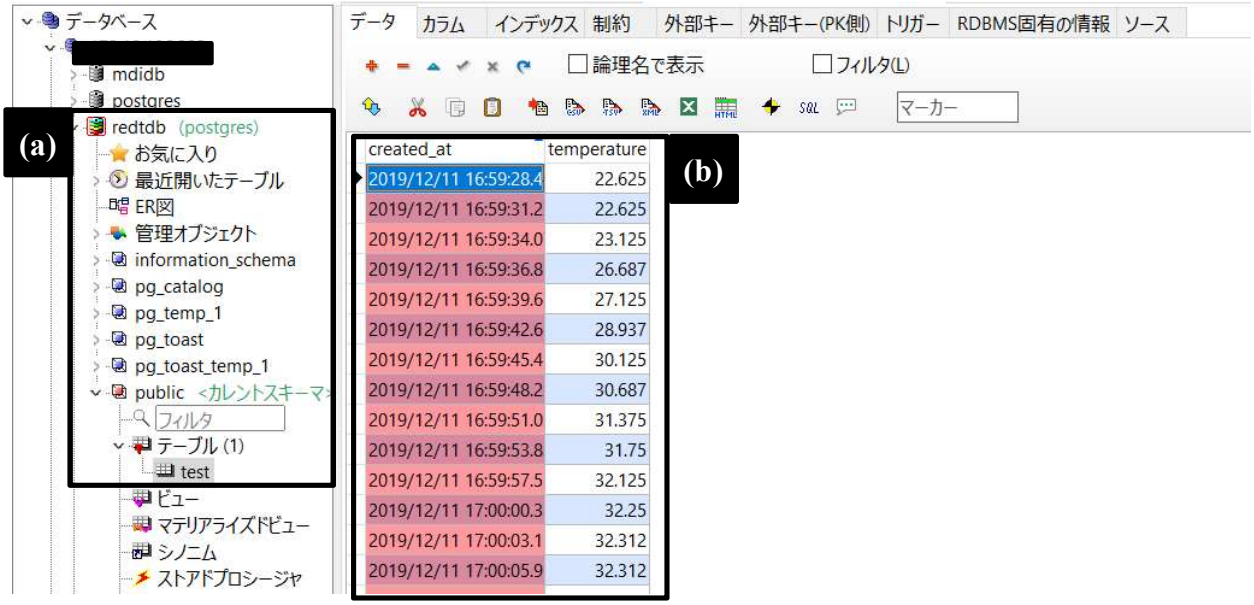

Fig. 8 Connecting to database system

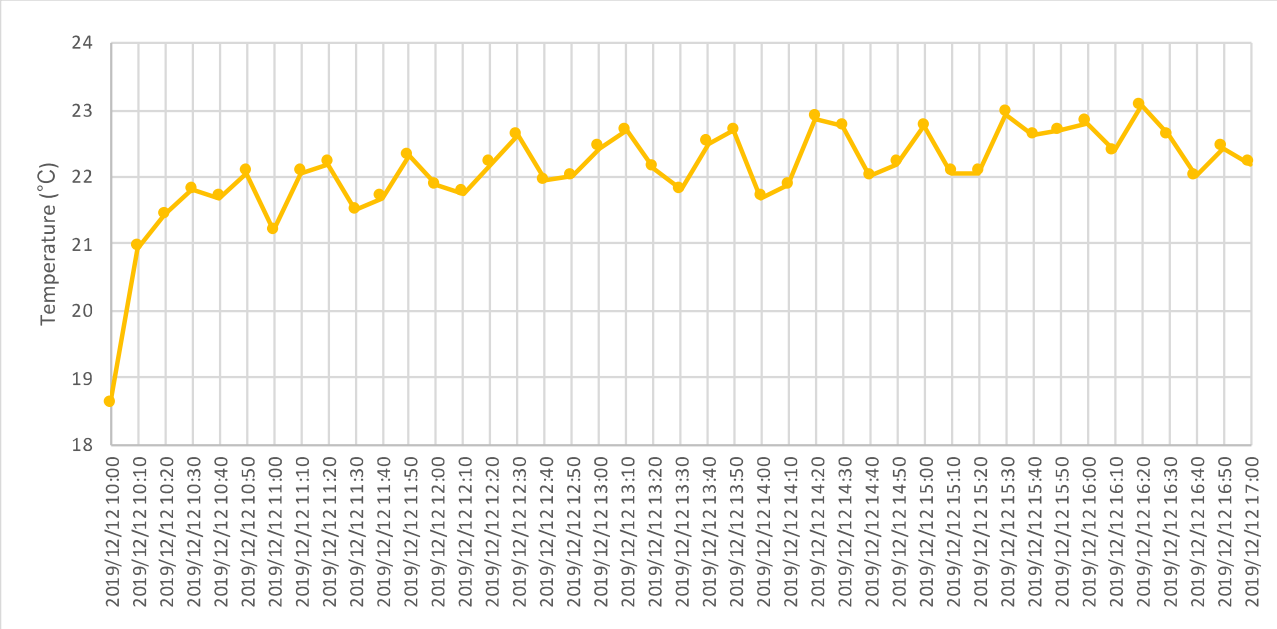

Fig. 9 room temperature sensing data every 10 minutes in 7 hours

Fig. 9 is the result of the experiment of temperature sensing. The temperature gradually increases from 18 degrees to approximately 22 degrees for 30 minutes from the start of the experiment at 10:00 am. The temperature fluctuates every 20 to 40 minutes. Furthermore, despite the set temperature of 20 degrees, the temperature fluctuated between 21 to 23 degrees.

5. Summary and Conclusions

This study proposes a prototype system that can collect environmental data using a sensor and transfers the data using a wireless network to an external database for storage. The system was tested and verified as working by confirming that the monitoring data was stored in the remote database server.

Regarding the fact that the actual temperature is 1 to 2 degrees higher than the set temperature, it is necessary to conduct another experiment and consider the cause.

Future work are as follows. First, Implementing more sensors. The device will involve more sensors in the device, so that it can collect multiple data, such as humid sensor for humidity, camera for images, CO2 sensor for CO2 density, illuminance sensor for illuminance. The author needs to deploy an actuator for physical control and improve the power supply, so that it can be used outdoors. Second, Inventing methodology for food adopting sensor information and AI.

References

[1] G. E. Moore, Progress in digital integrated electronics, vol. Vol. 21, Electron Devices Meeting, 1975, pp. 11-13.

[2] K. O. Flores, I. M. Butaslac, J. E. M. Gonzales, S. M. G. Dumlao and R. S. J. Reyes, "Precision agriculture monitoring system using wireless sensor network and Raspberry Pi local server," 2016 IEEE Region 10 Conference (TENCON), 2016.

[3] The Raspberry Pi Foundation, "Raspberry Pi," [Online]. Available: https://www.raspberrypi.org.

[4] F. Sheikh and L. Xinrong, Wireless Sensor Network System Design using Raspberry Pi and Arduino for Environmental Monitoring Applications, vol. 34, Procedia Computer Science, 2014, pp. 103-110.

[5] Arduino Foundation, “Arduino,” [Online]. Available: https://www.arduino.cc.

[6] Python Software Foundation, "Python," [Online]. Available: https://www.python.org.

[7] https://www.postgresql.org, "PostgreSQL: The World's Most Advanced Open Source Relational Database," [Online]. Available: The PostgreSQL Global Development Group.

[8] Maxim Integrated, "DS18B20 Programmable Resolution 1-Wire Digital Thermometer," [Online]. Available: https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf.

（受理 令和 2 年 3 月 19 日）