Sensors and Materials, Vol. 27, No. 8 (2015) 727–732
MYU Tokyo

S & M 1109

Implementation of Body Temperature and Pulseoximeter Sensors for Wireless Body Area Network

Muhammad Udin Harun Al Rasyid*, Bih-Hwang Lee[1],
Amang Sudarsono and Taufiqurrahman

Politeknik Elektronika Negeri Surabaya (PENS), Electronics Engineering Polytechnic Institute of Surabaya (EEPIS), Jl. Raya ITS, Kampus ITS Sukolilo Surabaya 60111, Indonesia
[1]Department of Electrical Engineering, National Taiwan University of Science and Technology, No. 43, Sec. 4, Keelung Rd., Da'an Dist., Taipei City 10607, Taiwan

(Received July 2, 2014; accepted February 9, 2015)

Key words: wireless body area network, sensor, pulseoximeter, temperature, E-health

The E-health sensor is one of the technologies of the wireless body area network (WBAN), which has a wide range of sensors that can be used to check a person's health data. In this work, we implement an E-health sensor system that can be used to read health vital sign data and store it in a database server. The sensors used in this work are a body temperature sensor and a blood oxygen sensor. The method that we used enables us to show the results of the sensor readings to a desktop application, then read the printout of the series and create a visual in the form of tables and graphs, as well as store the data in a database and display the data via a website in the form of reports that can be accessed remotely. After doing some testing on the user, it can be concluded that this application can be used to access health data remotely.

1. Introduction

Health services have become more attractive with the advent of the need to collect health data remotely from a patient automatically. The wireless body area network (WBAN) is a biomedical sensor network node connected wirelessly to the communication devices, in and near the area of the body. The WBAN is composed of small devices and low-power biomedical nodes to prolong the lifespan of sensor nodes.[(1–8)]

The WBAN applications are very diverse, including health, athletic training, safety, electronics user, secure authentication, and security personnel in uniform. The WBAN can also connect to a local area network using a variety of wired and wireless communication technologies.[(1)]

*Corresponding author: e-mail: udinharun@pens.ac.id

ISSN 0914-4935

E-health Sensor V2.0 is a sensor that allows Arduino and Raspberry Pi users to conduct biometric and medical applications where body monitoring requires nine different sensors: sensors for oxygen in the blood (SpO2), air flow (respiratory) sensors, body temperature sensors, electrocardiogram (ECG) sensors, glucose sensors, galvanic skin response (GSR-sweating) sensors, blood pressure (sphygmomanometer) sensors, sensors for determining the position of the patient (accelerometer), and muscle/eletromyography (EMG) sensors.

The current problem of health service is the lack of effective health care in hospitals, clinics, and health centers.[2] Thus, the application of the WBAN for remotely monitoring the effectiveness of health care on the basis of the patient's condition is necessary. The purpose of this work is to implement the protocol of WBAN E-health using body temperature and pulseoximeter libellium sensors to collect and monitor health data remotely (telecare). Data from sensor readings on the serial monitor will be visualized as graphs and stored in a database. The results of the two sensors can be viewed through a website, thus allowing the nurse or physician to monitor the patient's condition.

There are some related works on the WBAN. Reference 3 describes the architecture of the WBAN and design issues and focuses on how these technologies could help the military and aerospace industries. Reference 4 proposes a WBAN system that allows private medical applications developed using personal electronic devices coupled together with the sensor. Reference 5 describes the special features of wireless technologies along with their main advantages, weaknesses and applications in the WBAN. Reference 6 describes wireless systems using a medical path to obtain physiological data from sensor nodes to remote stations with a multi-hopping technique using the medical gateway wireless boards.

2. System Design

Figure 1 shows the system design architecture in this work. The E-health sensor device contains a temperature sensor, a pulseoximeter sensor, an E-health sensor board, and an Arduino board. The temperature and pulseoximeter sensors read the patient's body condition. The health data will be stored in a PC desktop server, then stored in a MySQL database server. Furthermore, the results from the database can be viewed via a website in the form of a report.

The specifications of the hardware and software used in this work are as follows: Compaq Presario CQ40, CPU 2.40 GHz (Dual Core), 2048 MB RAM, Arduino, E-health sensor, Windows 8 professional 64 bit, Visual Studio Premium 2012, and Arduino IDE.

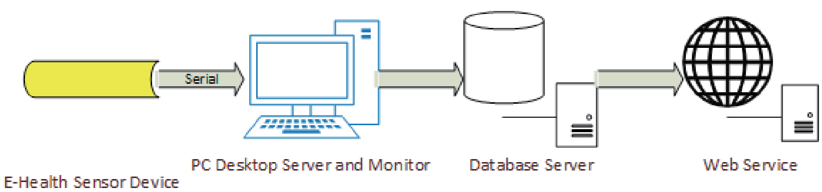


Fig. 1. (Color online) System design architecture.

3. Results and Discussion

3.1 *E-health experiment*

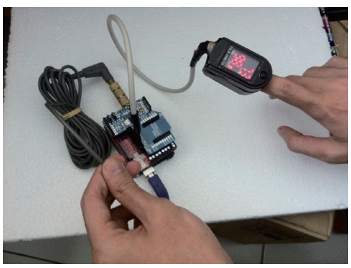
In this section, we explain the initial testing of the Arduino uno program to be able to perform and display the sensor readings in a serial monitor. Figure 2 shows the installation of the temperature and pulseoximeter sensors on the E-health sensor board and paired to Arduino uno as the microcontroller. The values of these sensors are then processed on a PC desktop and a database server.

Figure 3 shows the outputs of the body temperature and pulseoximeter sensors on a serial monitor. For the temperature sensor value, we determine two digits after the decimal to be displayed. For the pulseoximeter sensor, there are two parameters that are shown on this sensor, Bpm and Spo2. Bpm shows the heart beat rate per minute and Spo2 gives the return value of oxygen saturation in the blood.

3.2 *Application experiment*

After the sensor data is successfully displayed in the PC monitor, we carried out desktop visualization to take the value in the serial monitor and process it into a desktop application.

Figure 4 shows the design of the initial appearance of the application. This application has several features, among which, it can store user data, check the connection to the database, view the results in a database record, and perform serial monitor readings. The functions of each button in this application are as follows. Add button: this button is used to add a new user into the application; clear button: it acts to remove the user; save button: this button is disabled because the function keys are used to perform user data storage. This button will be enabled when the add button is pressed. Combobox port E-health: this combobox is used to determine which ports will be processed for reading; combobox baud rate: the baud rate function is used to determine the port that will be read from several baud rate options. This baud rate must be the same as that set on the Arduino. Combobox device: this functions to determine which device will be reading the series; start button: this functions to open a port that has been set on the combobox port E-health as well as to set the appropriate baud rate relative to that of the E-health; stop button: this works to stop reading the series; record table result:



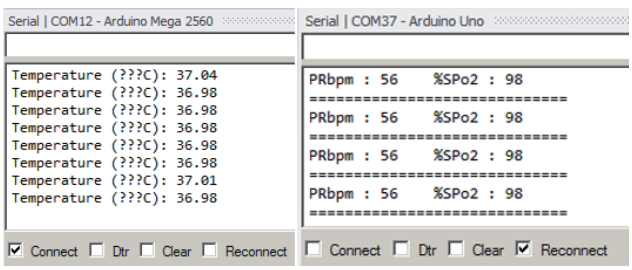


Fig. 2 (left). (Color online) E-health sensor device.
Fig. 3 (right). Body temperature sensor output.

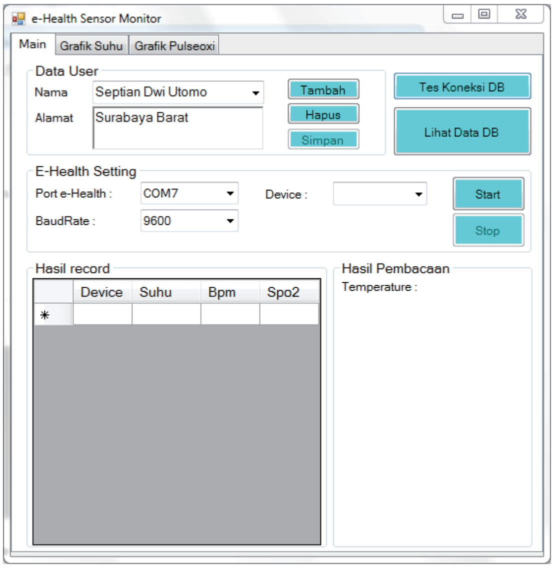


Fig. 4. (Color online) Main application.

this shows the results of a table to store temporary data that has been read (health data can be read on the serial output); and tab graphs: these show the temperature charts and pulseoximeter result in a graph.

We display the result on the console first to find out which would then be subjected to data processing as shown in Fig. 5.

Figure 5 shows the output on the console version. From the console output that we have tried, we finally take the data per line, then break it down into an array. From these results, we determined the sequence of the data as follows: device number | temperature data | data Bpm | data Spo2 | separator.

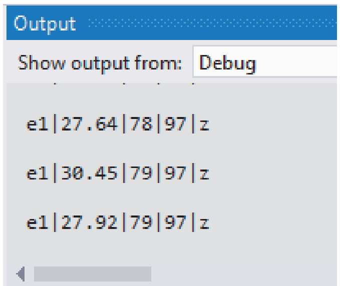
Thus, after reading per line, we take the data and break it into an array. After the data is saved in the table, the next process is carried out in the form of a graph. For any data that was originally in the form of an array of strings, we changed it first into a double-data type, which will be incorporated into a chart, as in Fig. 6.

Figure 6 is the graph that shows the change in temperature in Celsius (y-axis) relative to the time in seconds (x-axis). The range of the y-axis changes every 30 s.

3.3 *Web report experiment*

In this section, we test the web application report to display data based on a user and a specific time range. Figure 7 shows the initial view of the web report.

This figure shows the website report form that contains a combobox name and period. This combobox contains the name of an existing user in the database, whereas the period is the period of user data to be displayed in the report. The period indicates the date between when and for how long the data will be displayed in the report. If the



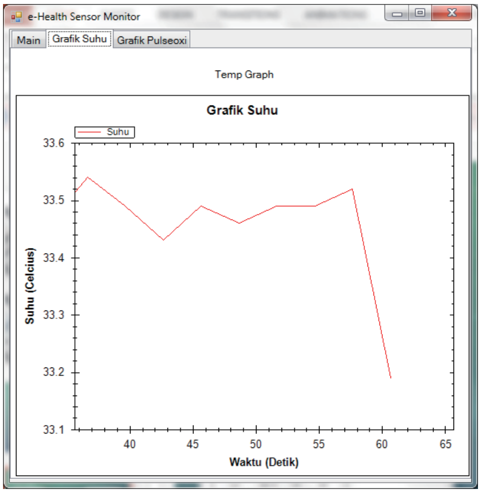
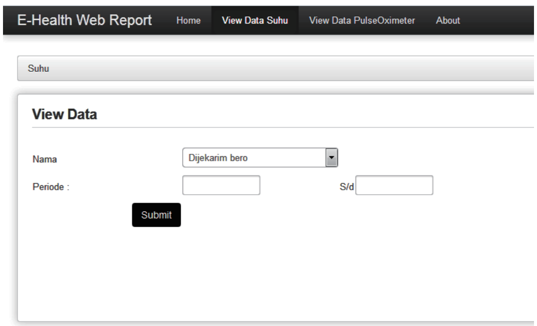
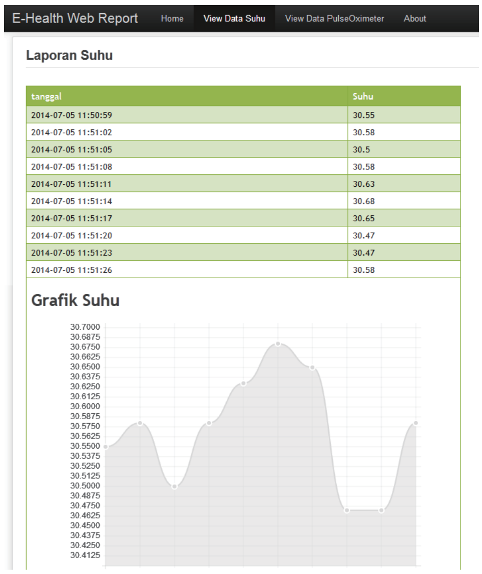


Fig. 5 (left). (Color online) Console version output.
Fig. 6 (right). (Color online) Graph output.





tanggal	Suhu
2014-07-05 11:50:59	30.55
2014-07-05 11:51:02	30.58
2014-07-05 11:51:05	30.5
2014-07-05 11:51:08	30.58
2014-07-05 11:51:11	30.63
2014-07-05 11:51:14	30.68
2014-07-05 11:51:17	30.65
2014-07-05 11:51:20	30.47
2014-07-05 11:51:23	30.47
2014-07-05 11:51:26	30.58

Fig. 7 (left). (Color online) Website report form.
Fig. 8 (right). (Color online) Website report output.

data period is not selected, it will display all existing data for that user.

Figure 8 shows the tables and graphs on the temperature data of a user. The result of the report is the result of a SQL query made to obtain the output as shown.

4. Conclusions

In this work, we implement an application to create a monitoring sensor system for E-health, which can perform data reading and storage in a database. The sensors used in this study are body temperature and blood oxygen (pulseoximeter) sensors. The method that we used enables us to show the results of sensor readings to a desktop application, then read the printout of the series and create a visual in the form of tables and graphs, as well as store the data in a database and display sensor data via a website in the form of reports that can be accessed remotely. The data of temperature and oxygen in the blood of each user can be monitored remotely through the website report. For further research, we have a plan of changing the data transfer method not only serially but also using other connections such as ZigBee, Wi-Fi, and GPRS.

Acknowledgements

This research was supported in part by Ministry of Research, Technology and Higher Education of Indonesia, International Research Collaboration and Scientific Publication Scheme, under Grant No. 0094/E5.1/PE/2015.

References

1 B. Latre, B. Braem, I. Moerman, C. Blondia and P. Demeester: Wireless Networks **17** (2011) 1.
2 C. A. Chin, G. V. Crosby, T. Ghosh and R. Murimi: Proc. Workshop Comput. Networking Commun. (IEEE, 2012) pp. 99–103.
3 T. Farhana and I. M. Hassan: Proc. Int. Conf. Inf. Manage. Eng., eds. X. Zhou, R. Zainol and C.A. Aling (IEEE, 2009) pp. 656–660.
4 L. Minsoo, C. Okju, L. Seul-A and C. Nory: Proc. Int. Conf. Consum. Electron. (ICCE), ed. P. Corcoran (IEEE, 2011) pp. 549–550.
5 M. Samaneh, A. Perzila and A. Mehran: Proc. Int. Symp. Commun. Inf. Technol. (ISCIT), ed. E. Dutkiewicz (IEEE, 2012) pp. 42–47.
6 M. R. Yuce: Sens. Actuators **162** (2010) 116.
7 H. Cao, V. Leung, C. Chow and H. Chan: IEEE Communication Magazine (IEEE, 2009) pp. 84–93.
8 D. M. Barakah and M. Ammad-Uddin: Proc. Third Int. Conf. Intell. Syst., Model. Simul. (ISMS), eds. I. Saad, A. Pantelous and J. Yunus (IEEE, 2012) pp. 214–219.