A Portable Node of Humidity and Temperature Sensor for Indoor Environment Monitoring

Trio Adiono
School of Electrical Engineering and Informatic, Institut Teknologi Bandung
Gd. Achmad Bakrie Lt. III, ITB campus, Jln. Ganesha No. 10, Bandung city, West java, Indonesia
tadiono@stei.itb.ac.id

Maulana Yusuf Fathany, Syifaul Fuada, Irfan Gani Purwanda, Sinantya Feranti Anindya
University Center of Excellence on Microelectronics, Institut Teknologi Bandung
IC Design Laboratory, Gd. PAU Lt. IV, ITB Campus, Jln. Tamansari No. 126, Bandung city, West Java, Indonesia

***Abstract*— Humidity and temperature are parameters those are commonly implemented for monitoring tasks, including monitoring within an indoor environment such as a smart home. In this study, we design, develop, and demonstrate the performance of a portable sensor system for indoor environment, which is connected to the smartphone-based user interface for monitoring humidity and temperature. This sensor system consists several components those are packed into a single small box, namely: a Zigbee communication module, STM32L100 microcontroller, DHT11 temperature sensor, BL-5C rechargeable battery, and a charger circuit. The sensor system and the application are then tested to gauge the performance. Based on the test, the system is able to communicate with the smartphone through the host. In addition, based on 24-hour battery testing, the system requires less than 0.53958 watts to operate, thus achieving the low-cost target.**

Keywords—DHT11; Humidity and Temperature; Indoor environment; Portable sensor system; Smart home

I. Introduction

Humidity and temperature are two of the most common monitoring parameters in Internet of Things research, whether for outdoor purposes (e.g. air quality measurement and irrigation) and indoor purposes (e.g. smart home and smart building). This is because of the fact that not only humidity and temperature often represent crucial aspects of the monitored object, but also due to the abundance and ease of use of the sensors. In usual, the monitoring is conducted through web-based or smartphone application-based user interface, with some of the examples can be observed in [1-5].

Despite the relative ease of use and implementation of the sensors, many of the deployment in the aforementioned literatures can be considered impractical especially for mass-production and integration into IoT network. In the aforementioned literatures, the sensors are deployed using Arduino boards, which are good enough to use for small-scale projects, but are impractical for larger-scale implementation due to the speed limitation and the boards' cost. Furthermore, it is desirable to combine the sensors and other components such as the microcontroller, power, and connectivity modules into one small package for lightweight implementation and rapid installation of the sensor system. For this modular plug-and-play approach, there have been some examples of the devices designed following this approach as exemplified in [6-8]. However, the aforementioned projects focused on sensor systems for outdoor deployment, which are then monitored using personal computer terminal that limits mobility aspect of the system. As such, for this project, the design and development is focused on sensor system for indoor deployment, which is then monitored using smartphone application.

To answer the aforementioned requirements, in this research a portable and low-cost humidity and temperature sensor system is designed. The designed sensor system consists STM32L100RCT7 microcontroller as main processor, DHT11 humidity and temperature sensor, Zigbee module for connectivity with Raspberry Pi-based host, and BL-5C battery and its charger circuit. The sensor system is monitored using Android-based application developed using MIT AppInventor 2, which is connected to the Raspberry Pi-based host using Bluetooth protocol. By designing the system, we aim to provide comfortable and practical way for user to monitor their home's condition anytime and anywhere.

II. Methods

A. System Description

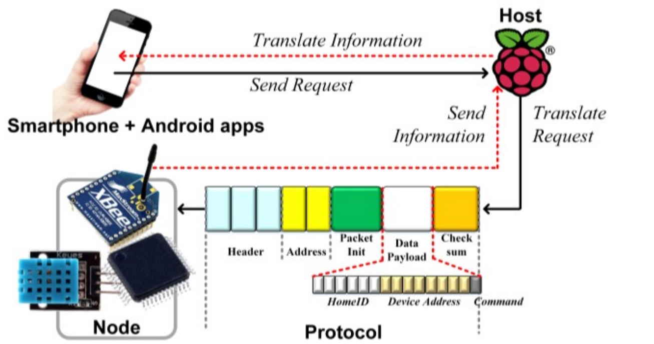


Fig. 1. System architecture of the indoor monitoring system, excluding additional features such as encryption and scheduling schemes

This work is related to the previous research as described in [9-16]. In the previous research, we propose the architecture

978-1-5386-0661-2/18/$31.00 ©2018 IEEE

of indoor monitoring system (Fig. 1) which is divided into three parts: host, user interface (smartphone), and sensor nodes. This work is part of the sensor node development, namely the sensor node for humidity and temperature monitoring.

The monitoring and control of the system is conducted through an Android-based application named 'mySmartHome v1.0'. The application works by interacting with the Raspberry Pi host using Bluetooth. The host serves to translate information and send instructions from the smartphone to the sensor nodes through XBee module based on the given identification address. The complete format of the message is described in section E.

B. System Specification

The sensor system is designed to measure indoor humidity and temperature within home environment, as well as to work within Internet of Things network. As such, the sensor doesn't need to be highly precise, but just enough to get the gist of the room's condition. On the other hand, because the system is to be connected to the network, the system requires fast processor and connectivity that is both fast and uses low power.

C. Hardware Design

The system consists an STM32L100RCT7 microcontroller, DHT11 humidity and temperature sensor Xbee Pro module for Zigbee-based connectivity with Raspberry Pi-based host, and BL-5C battery and its charger circuit. The diagram block depicting the system's structure can be observed in Fig. 2.

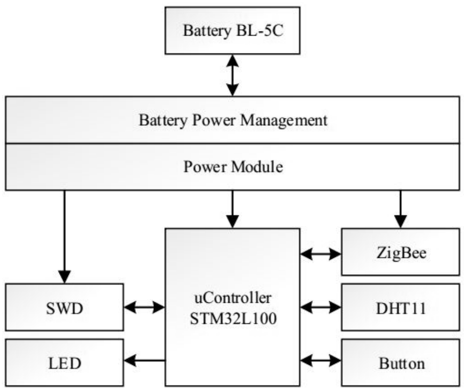


Fig. 2. Structure of humidity and temperature sensor system

While the DHT11 sensor is a low-cost sensor for reading humidity and temperature. The sensor allows humidity reading at 20-80% and 0-50 °C temperature, each with 5% and ±2 °C error respectively. The basic configuration of the sensor with STM32 microcontroller can be observed in Fig. 3. In addition to the sensor, this system uses Zigbee for communication with the host. The Zigbee protocol is chosen due to its low power usage and extensive coverage range, which also makes it suitable for power conservation purposes. This system is also equipped with microUSB port, which can be used for firmware update and charging the system's battery.

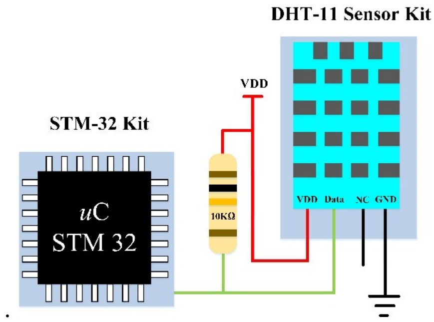


Fig. 3. Basic configuration of DHT11 sensor connected to STM32L100 chip

D. Software Design

The software design consists two parts, namely the design of the software within the microcontroller and the design of the Android application. The software in the microcontroller serves to process the message given by the Raspberry Pi host, while the Android application serves to control. The flowcharts for each software are depicted in Fig. 4 and Fig. 5.

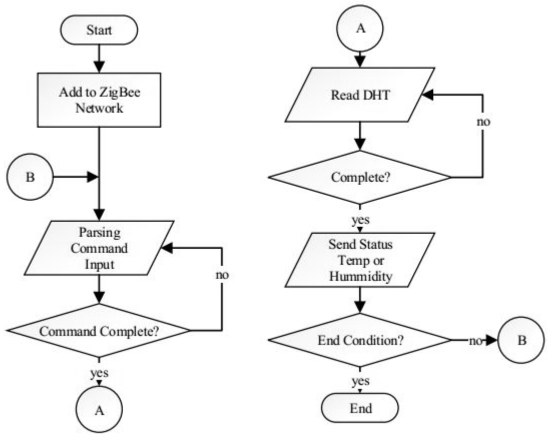


Fig. 4. Flowchart of humidity and temperature node software

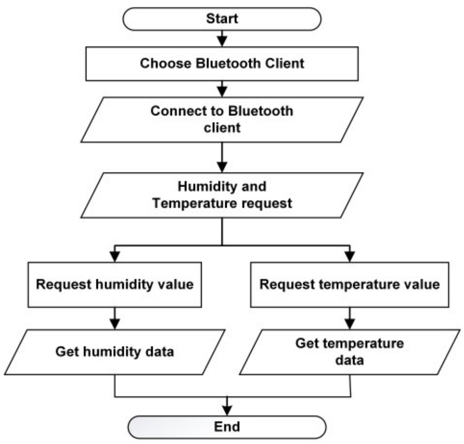


Fig. 5. Flowchart of Android application

E. Packet Data Design

The message sent between the host and the sensor node is arranged based on the data protocol from the previous work in [17-18]. The structure of the message (Fig. 6) consists 3 Bytes of header, 2 Bytes of address, 1 Byte for packet initialization, varying Bytes of data payload, and 1 Byte for verification using check sum. For this humidity and temperature sensor, the size of data payload is 1 Byte.

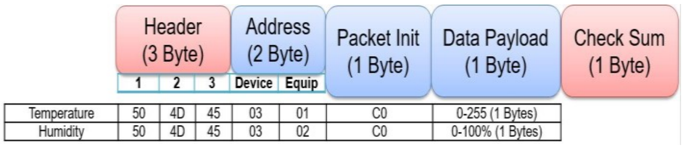


Fig. 6. Packet data structure for humidity and temperature sensor node

III. RESULTS AND DISCUSSION

A. System Implementation

Based on the established specification and design, the printed circuit board of system hardware is designed as shown in Fig. 7 for top view and Fig. 8 for bottom view. The PCB consists two layers and masking, with the components are mounted and soldered on both sides. The assembled board is then packaged as depicted in Fig. 9.

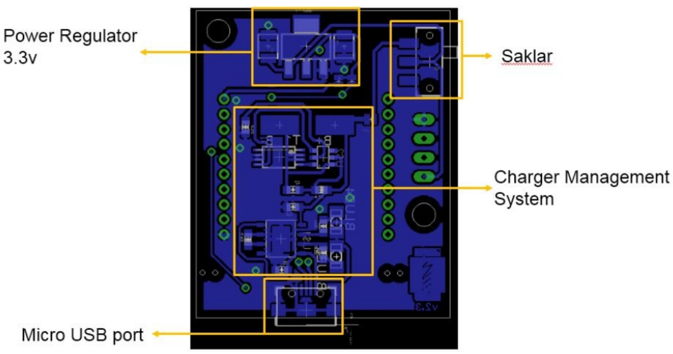


Fig. 7. Top layer of PCB for humidity and temperature node

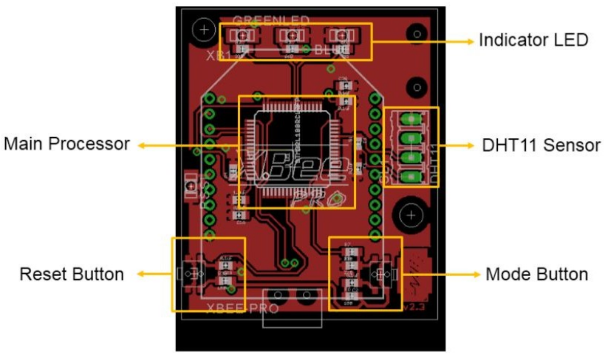


Fig. 8. Bottom layer of PCB for humidity and temperature node

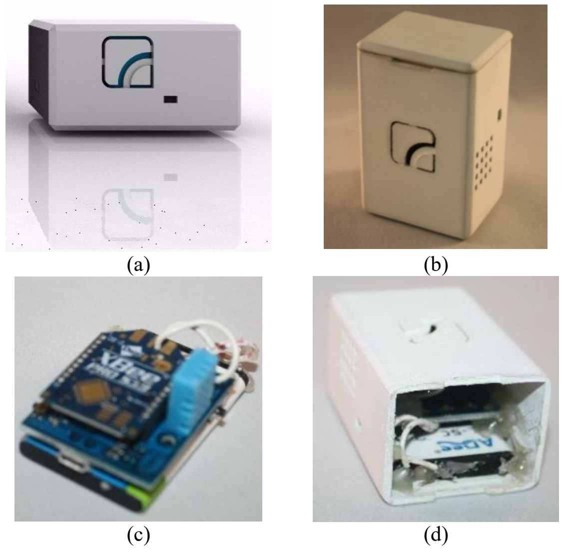


Fig. 9. Packaging of the humidity and temperature sensor node

```
if (flagDHT==1)
{
        DHT11_RawRead(buf_dht);
        res_dht = DHT11_Humidity(buf_dht);
        sprintf(strDisp, "H%02d", buf_dht[0]);
        kirim3_string(strDisp);
        flagDHT=0;
}
else if (flagDHT==2)
{
        DHT11_RawRead(buf_dht);
        res_dht = DHT11_Temperature(buf_dht);
        sprintf(strDisp, "S%02d", buf_dht[2]/29*25);
        kirim3_string(strDisp);
        flagDHT=0;
}
```

Fig. 10. Microcontroller code for humidity and temperature node

```
uint8_t DHT11_RawRead(uint8_t *buf);
float DHT22_Humidity(uint8_t *buf);
float DHT22_Temperature(uint8_t *buf);
uint8_t DHT11_Humidity(uint8_t *buf);
uint8_t DHT11_Temperature(uint8_t *buf);
uint8_t DHT11_pwm_Read(uint8_t *buf, uint32_t *dt, uint32_t *cnt);
```

Fig. 11. List of functions within dht11.c library

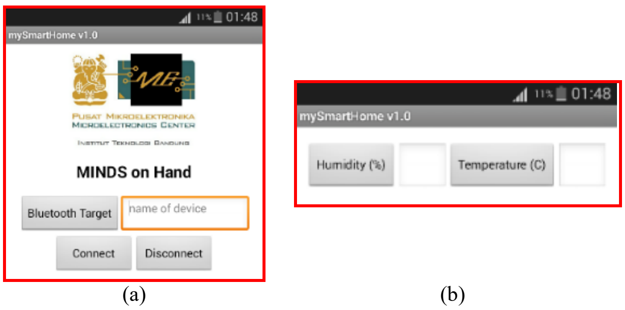
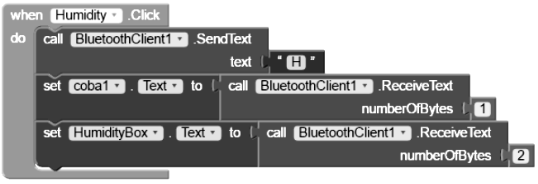
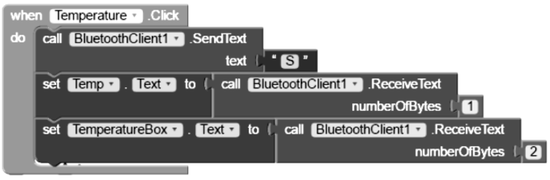


Fig. 12. GUI of the developed android application: (a) Bluetooth connectivity to the host; (b) retrieve humidity and temperature data

The software implementation for the microcontroller is depicted in Fig. 10 and Fig. 11, while the implementation for the Android application is depicted in Fig. 12 and Fig. 13. The software implementation utilizes the *dht11* open library for the microcontroller and MIT AppInventor 2 for the Android application, respectively.



(a)



(b)

Fig. 13. Screenshot of logic blocks for the: (a) humidity; (b) temperature reading in MIT App Inventor 2

B. *Performance test*

To gauge the performance of the system, three tests are conducted, namely connectivity test, power usage measurement, and battery life measurement. The connectivity to the system is tested using smartphone running Android 4.4.2 (KitKat). Based on the testing conducted, the system is able to communicate with the host and user interface, with the test result showing 34% humidity and 28 °C room temperature.

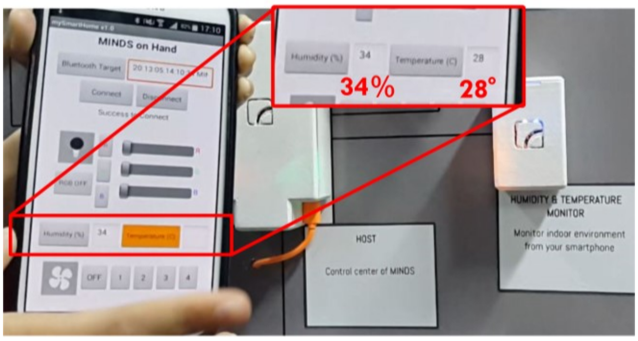


Fig. 14. Functional test of humidity and temperature sensor

The power usage measurement is conducted using power supply equipped using voltage and current display. Based on the conducted measurement, there is no difference between power usage in idle state and processing state. The result of the measurement is elaborated in Table I.

TABLE I. POWER MEASUREMENT OF SENSOR NODE

Input voltage	**Current**	
	Idle condition	***Process condition***
5 V_{DC}	42.6 mA	42.6 mA

The battery life is measured by having the system active for 24 hours. The BL-5C battery has 3.7 V_{DC} voltage and capacity 3500 mAH, from which it can be inferred that the battery has 12.95 watt-hours. Ideally, the testing should be conducted by completely exhausting the power, but in this research the system is only activated for 24 hours to estimate the power requirements of the system. If within 24 hours the battery runs out of energy, it means the device needs at least 0.53958 watt (145,833 mA).

However, in this testing, the system is still active after 24 hours, so it can be inferred the power requirement of the system is less than 0.53958 watt.

IV. Conclusion and Future Work

In this research, a sensor system to measure humidity and temperature is designed, implemented, and evaluated. The system is connected with an Android-based user interface through Raspberry Pi-based host, which utilizes Bluetooth for the former and Zigbee for the latter. Based on the testing conducted, the system is able to work with the designed Android application. Furthermore, the system uses low power (< 0.53958 watt), making it suitable for low-power Internet of Things application.

In the future, the system will be enhanced to support better security. Furthermore, additional sensors will be added for better measurement of air quality. The enhancement will hopefully make the system suitable not only for safety purposes, but also for maintaining health of the user.

Acknowledgment

This work is part of the "Internet of Things Devices for Smart Home System" project, which was funded by the Ministry of Research, Technology, and Higher Education of the Republic of Indonesia (Kemenristekdikti) for the decentralization scheme with Grant Number 009/SP2H/LT/DRPM/IV/2017
The sensor node along with the other end-devices have been exhibited in Centrum für Büroautomation, Informationstechnologie und Telekommunikation (CeBIT), on March 14-28, 2016 in Hannover-Germany
Site: (http://www.pme.itb.ac.id/microelectronics-center-itb-on-cebit-2016-and-ict-roadshow/

References

[1] T.D. Hendrawati and I. Lesmana, "Rancang bangun saklar lampu otomatis dan monitoring suhu rumah menggunakan VB. Net dan Arduino," *J. Teknologi dan Rekayasa,* Vol. 1(1), pp. 67-72, Desember 2016.

[2] X. Shi, et al., "Design of a smart home remote monitoring system based on C++ and Zigbee," *Advances in Intellegent Systems Research,* Vol. 133, pp. 166-169, 2016.

[3] P. Mandarani, "Perancangan dan implementasi *user interface* berbasis web untuk *monitoring* suhu, kelembaban dan asap pada ruangan dengan memanfaatkan jaringan *local area network*," J. TEKNOIF, Vol. 2(2), pp. 37-42, October 2014.

[4] H.H. Hadwan and Y.P. Reddy, "Smart home control by using Raspberry Pi & Arduino UNO,"*IJARCCE,* Vol. 5(4), pp. 283-288, April 2016.

[5] E. Salim and K. Tanjung, "Perancangan dan implementasi telmetri suhu berbasis Arduino Uno," *J. Singuda Ensikom*, Vol. 11(29), pp. 41-45, April 2015.

[6] W-Y. Yi, K-S. Leung, and Y. Leung, "A modular plug-and-play sensor system for urban air pollution monitoring: design, implementation, and evaluation," *Sensors,* Vol. 18(7), 2018. doi:10.3390/s18010007.

[7] T.S. Gunawan, et al., "Design and implementation of portable outdoor air quality measurement system using Arduino," *IJECE,* Vol. 8(1), pp. 280-290, February 2018.

[8] S-K. Noh, K-S. Kim, and Y-K. Ji, "Design of a room monitoring system for wireless sensor networks," *Int. J. of distributed sensor networks,* pp. 1-7, July 2013.

[9] B.L. Lawu, M.Y. Fathany, K. Afifah, M.H. Santriaji, R.V.W. Putra, S. Fuada, and T. Adiono, "Prototyping design of mechanical based end-devices for smart home applications," *Proc. of 2016 4th Int. Conf. on Information and Communication Technology (ICoICT)*, pp. 1-5, May 2016. DOI: 10.1109/ICoICT.2016.7571927.

[10] T. Adiono, R.V.W. Putra, M.Y. Fathany, K. Afifah, M.H. Santriaji, B.L. Lawu, and S. Fuada, "Prototyping design of electronic end-devices for smart home applications," *Proc. of The IEEE Region 10 Symposium* (TENSYMP), pp. 1-5, May 2016. DOI: 10.1109/ TENCONSpring.2016.7519415.

[11] T. Adiono, R.V.W. Putra, M.Y. Fathany, B.L. Lawu, K. Afifah, M.H. Santriaji, and S. Fuada, "Rapid prototyping methodology of lightweight electronic drivers for smart home appliances," *Int. J. of Electrical and Computer Engineering* (IJECE), Vol.6(5), pp. 2114-2124, October 2016.

[12] T. Adiono, M.Y. Fathany, R.V.W. Putra, K. Afifah, M.H. Santriaji, B.L. Lawu, S. Fuada, "Live Demonstration: MINDS-Meshed and Internet Networked Devices System for Smart Home," *Proc. of the 13th IEEE Asia Pacific Conference on Circuits and Systems* (APCCAS), pp. 736-737, October 2016. DOI: 10.1109/APCCAS.2016.7804031.

[13] T. Adiono, B. Tandiawan, M.Y. Fathany, W. Adijarto, and S. Fuada, "Prototyping Design of IR Remote Controller for Smart Home Applications," *Proc. of IEEE Region 10 Conf.* (TENCON), pp. 1304-1308, November 2017. DOI: 10.1109/TENCON.2017.8228059.

[14] T. Adiono, R. Marthensa, R. Muttaqin, S. Fuada, S. Harimurti, and W. Adijarto, "Design of database and secure communication protocols for internet-of-things-based smart home system," *Proc. of IEEE Region 10 Conf.* (TENCON), pp. 1273-1278, November 2017. DOI: 10.1109/TENCON.2017.8228053.

[15] K. Afifah, S. Fuada, R.V.W. Putra, T. Adiono, M.Y. Fathany, "Design of Low Power Mobile Application for Smart Home," *Proc. of Int. Symposium on Electronics and Smart Devices* (ISESD), pp. 127-131, November 2016. DOI: 10.1109/ISESD.2016.7886705.

[16] T. Adiono, S.F. Anindya, S. Fuada, K. Afifah, and I.G. Purwanda, "Efficient Android Software Development using MIT App Inventor 2 for Bluetooth-based Smart Home," *Unpublished.*

[17] T. Adiono, B. Tandiawan, and S. Fuada, "Device Protocol Design for Security on Internet of Things based Smart Home," *Unpublished.*

[18] M.Y. Fathany and T. Adiono, "Wireless protocol design for smart home on mesh wireless sensor network," *Proc. of ISPACS,* pp. 462-467, November 2015.