Application of Smart Home and Smartwatch to Assist Elders in Home Care Scenario

Handy Wicaksono[1,a)], Petrus Santoso[1,b)], Indar Sugiarto[1,c)], Dwi Kristiyono[1,d)], Jonathan Aditya Wijaya[1,e)]

Author Affiliations
[1]*Electrical Engineering Department, Petra Christian University*
Jl. Siwalankerto 121 – 131, Surabaya, Indonesia

Author Emails
[a)] Corresponding author: handy@petra.ac.id
[b)] petrus@petra.ac.id
[c)] indi@petra.ac.id
[d)] dwi@petra.ac.id
[e)] jonathanadityawijaya@gmail.com

Abstract. Many elders choose to live independently in their own houses. This research aims to apply home automation and wearable device technology to assist the elders in the home care scenario. Unlike many research in this field, we apply our home automation system in the real or physical house. There are four main features of our smart home: door control, lighting control, temperature control, and gas leakage detection. Six Wemos D1 minis are used as the local controllers for each sensor and actuator, while a RaspberryPi 4 is utilized as the main controller. NodeRed is chosen as the tool to integrate various components in our system, so a web dashboard can be used to monitor and control the smart house. We also developed a mobile sensing platform called SmartSens. Samsung Galaxy Watch is used as a smartwatch worn by the elder. We then conduct simple experiments where the elder controls our smart home in manual mode successfully. The automatic mode is also running well. Additionally, heart rate and step count data are also successfully collected from the smartwatch. By combining a smart house and a smartwatch, the elder who lives by himself can be monitored by caregivers via the internet.

INTRODUCTION

Recently, life expectancy for elders has been increasing. This means the needs of caregivers, ones who take care of the elders, are also increasing. Nowadays, a lot of people can not take care of their own parents or grandparents because they are very busy with their own jobs or family life. It is also not easy (and is expensive) to find a trustable and capable person to take care of the elders. One solution is putting elders in a retirement home, but some people (or the elders themselves) do not agree with this solution because it might make the elders feel less loved by their families. Many elders want to spend their retirement time in their own house [1]. However, it might be too risky for them, as many unwanted events related to their health and security might happen to them. In this research, we suggest combining home automation and wearable device technology to assist elders who live alone in their own house (home care scenario).

Majumder et al. [2] proposed a smart home system for elders equipped with a home gateway that connects various home appliances, sensors that are attached to the elders, and a data collection system. The smart house also must be connected to the internet so the elders can contact medical staff (doctor or nurse) or emergency health services. The sensors attached to the elders are crucial as they monitor the condition of elders directly. Wang et al. [3] mentioned three main things that should be supervised: the vital signs (e.g., temperature, heart rate, respiration rate, blood pressure, oxygen saturation), the accurate positions of elders at home, and activity recognition of elders (for example:

falling, running, laying). These monitoring results can be recorded into a database and sent to the caregivers or medical staff.

Mobile sensing platform (MSP) is a combination of wireless sensors, smartphones, and the web that provide a powerful ubiquitous computing platform [4]. The wireless sensor in this platform can be any kind of sensor that communicates directly to the web server or through a smartphone device. It can be a custom wearable device, internal phone sensor or smartwatch device. In the mobile sensing platform, a user-level application runs on the smartphone as a broker between the user and the web. The phone reads the internal phone's sensor or external sensor and sends data to the web server [5].

Triantafyllidis et al. [6] discussed a pervasive system, especially in the area of healthcare. They give a review on three important areas, namely mobile phone sensing with in-built or external sensors, self-reporting, and social sharing. They also discussed the shortcomings and directions of the system. Most developed systems do not reach their full potential, mainly because of design and validation issues.

In [4], an architecture of mobile sensing platform that supports the usage of a smartwatch as a sensing device has been proposed. The development of a sample application with the proposed architecture has been done. A development framework called SmartSens also has been built.

In this research, we propose an application of a smart home and a wearable device to assist elders in a home care scenario. We use NodeRed to integrate various components in our smart house as an internet of things system [7]. We also use SmartSens [4], a mobile sensing platform that we have been developed previously. Our research is unique as we implement our system in a physical house (not just miniature).

RESEARCH METHOD

In this section, we will describe the smart home design and the wearable device system framework.

Smart Home Desigin

A lot of previous work in the home automation field is only applied in miniature size. In this case, most sensors and actuators are wired directly to a local controller as their location are closed to each other. When we try to implement the system in the real house, the main problem that we deal with is the sensors, the actuators, and the main controller are far from each other. Wiring all of them is also not practical and not tidy. In order to overcome this problem, we decide to equip each sensor and/or actuator with a local controller that has wi-fi capability. We choose Wemos D1 mini as it is a low-cost IoT controller that has all the features we need. Some relay shields are used to connect the controller with light and fan, which operate in AC voltage.

To gather all of the data from various IoT controllers, we choose Node Red that is installed in a RaspberryPi 4. NodeRed is a flow-based visual programming environment that is suitable to integrate various IoT hardware and software. We developed a dashboard that can be accessed from an internet browser on any devices. When abnormal things happen, for example, when there is a gas leak, a notification email will be sent to the house owner. The block diagram is shown in Figure 1.

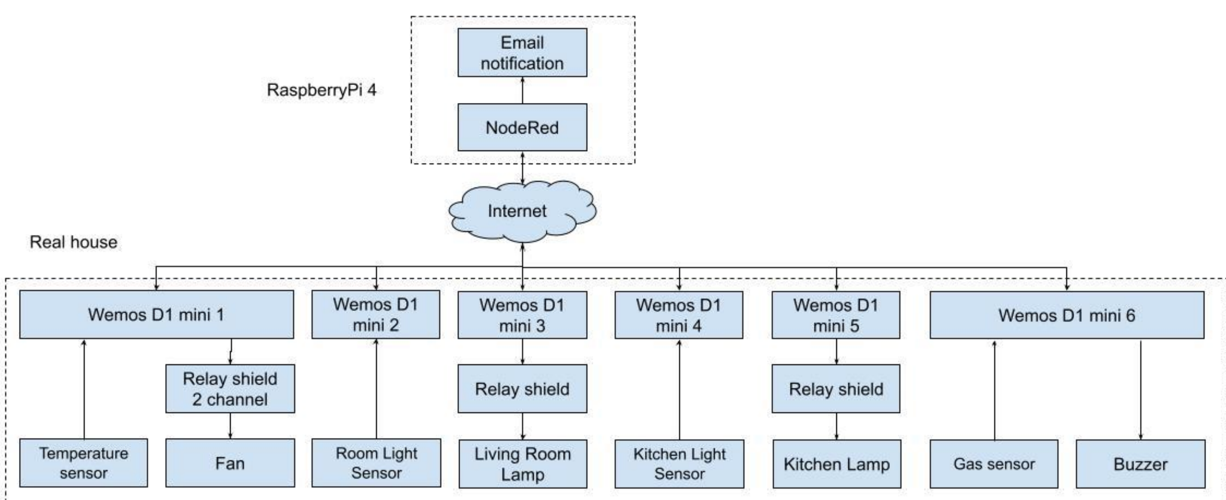
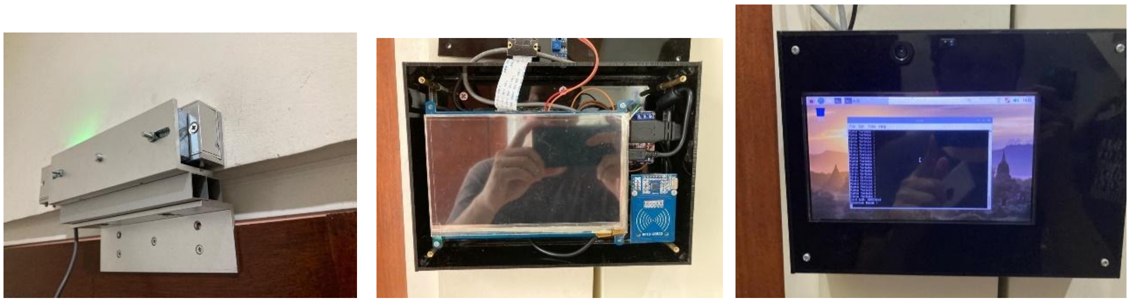


FIGURE 1. Block diagram of our system

There is one subsystem that has not been connected via the internet: the smart door system. Currently, it is only implemented locally using a magnetic sensor, an RFID sensor, and Raspberry Pi 3B as the local controller.

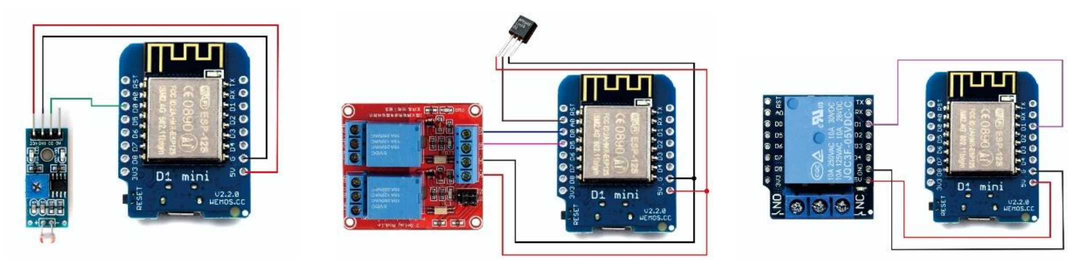
There are three automated sections in our house: the bedroom door, the living room, and the kitchen. The bedroom door is automated by giving a magnetic lock on the upper side of the door. Whenever a particular RFID card is scanned, the magnetic lock is released, and a person can enter the bedroom. RaspberryPi 3B is used as a local controller here. See Figure 2 for the detail.



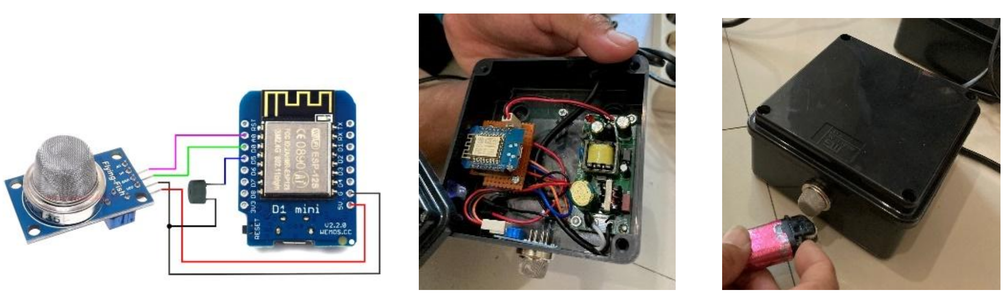
(a) The door magnetic lock (b) The opened controller box (c) The closed controller box

FIGURE 2. Smart door system

In the living room, we provide a light sensor (LDR, see Figure 3a.) and an AC lamp (driven by a relay, see Figure 3c.) as a lighting control system. A similar lighting control system is also set in the kitchen. We also provide a temperature sensor (LM 35) and a fan to build a temperature control system (see Figure 3b) in the kitchen. A gas leakage detection system is built here by providing an MQ2 gas sensor and a buzzer (see Figure 3d.). We put all components carefully in one sturdy black box so it looks tidy and professional (see Figure 3e and 3f). These black boxes are located around the house, which will be used for experiments.



(a) (b) (c)



(d) (e) (f)

FIGURE 3. Hardware components of our system

There are two control modes on our system: manual and automatic. The user should choose which mode he wants to use. In manual mode, all of the actuators (lights, fan, and buzzer) can be controlled manually from the NodeRed dashboard in the internet browser. While in automatic mode, the actuators are controlled based on these simple rules:

- Lighting rules: if the light sensor reading is below a particular threshold, then the AC lamp will be turned on automatically
- Temperature rules: if the temperature is more than the first threshold, then the fan will operate at a low speed. If the temperature is more than the second one, then the fan will move at high speed.
- Gas rules: if the gas concentration is more than a particular threshold, then the buzzer is turned on.

Figure 4 shows the flowchart of our smart house system.

The wearable device

SmartSens is a modular development framework for a mobile sensing platform. There are three main parts in the framework. The first part is for server development, the second part is for mobile apps development, and the last part is for smartwatch apps development. The server part adopts NodeJS as the main programming language. Mobile apps development support Android and iOS platforms with Java for Android and Swift for iOS. For the smartwatch part, the supported platform is Tizen and WearOS.

Figure 5 depicts the architecture of the mobile sensing platform. The platform consists of a server, a smartphone and a smartwatch. The server consists of Web API which communicates with smartphone and smartwatch, database backend, and processing backend. The smartphone has two kinds of modes: master smartphone and slave smartphone. For master smartphone mode, the communication is only with the server. This mode is used for the monitoring purpose. The slave smartphone mode is related with sensing through an external sensor (smartwatch) or its internal sensor. Smartwatch part dealing with sensing mechanism and simple interface to the user.

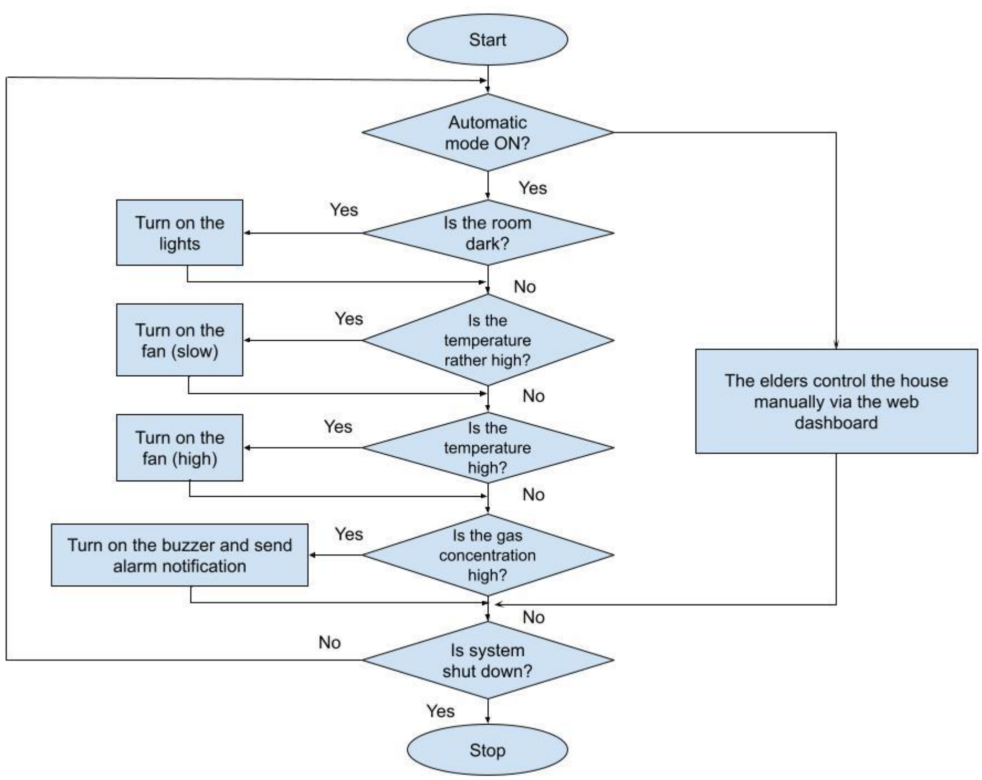


FIGURE 4. Flow chart of our smart house system

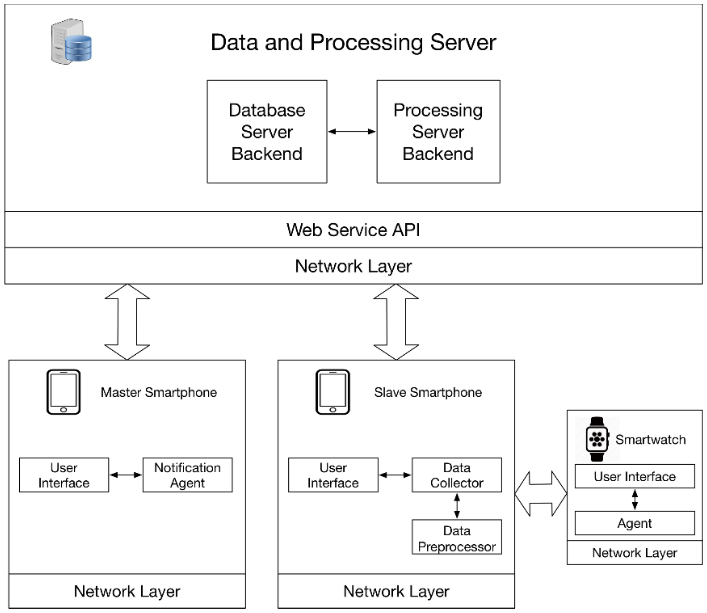


FIGURE 5. Mobile sensing platform architecture

The web API for the server conform to RESTful architecture. Data exchange uses JSON as the standard format. The server receives JSON requests from a client and gives a response. It can be a Success response or an Error response. Figure 6 depicts an example of such requests.

API Endpoint	JSON Request	JSON Response

/guardian/signUp	Request :
	{
	"email":"s",
	"password":"s",
	"name":"s",
	"fcmClientToken":"s"
	}
		Success Response:
		{
		"message": "Thanks! ",
		"guardianId": i,
		"name": "s",
		"email": "s"
		}
		Error Response:
		{
		"message": "..."
		}

FIGURE 6. JSON exchange example

Besides web API, the server also has a backend processing element. The backend processing element is modular. It is the core service element of the server. There are the database backend and the processing backend. We can add a processing logic module in the processing backend. By default, each raw data received from the sensor is sent to the database backend through the processing backend. In the processing backend, we can add a custom processing algorithm, such as fall detection, geofencing, activity detection, etc. The processing backend is also configurable to accept data from a single sensor or multiple sensors. The configuration is done through the web API.

In the master smartphone mode, the core part of the framework is the notification agent. Notification agent interacts with server through the publish-subscribe mechanism and gives notification for a predefined condition. The publish-subscribe and notification mechanism is configurable in the API.

For slave smartphone mode, the main function is to do sensing using an internal sensor or external sensor. The core part is the data preprocessor. The framework implements sensing architecture that consists of a sensing module and sensor manager. The sensing module provides access to the sensor. When new data is available, it is processed and structured in JSON format. The resulting object is then sent to the data collector. Sensor managers deal with the sensor availability detection, initialization, configuration, start and stop sensing process. It can be configured for continuous, periodic, or aperiodic sensing.

As for the smartwatch, the framework provides an agent to do the sensing process. The agent works as a sensing module and sensor manager in the slave smartphone. The configurable setting is how the smartwatch provides data to other entities in the system. It can communicate directly with the server or communicate with a slave smartphone. It can wait for a signal from another device to send data or actively send data to the other device.

RESULTS AND DISCUSSION

In this section, we elaborate on the home automation scenario, testing results for the smart house and the smartwatch.

Experiments on the home automation system

The sensors, actuators, and controllers of our system are located in three areas of a house: a bedroom door, a living room, and a kitchen (see Figure 7).

FIGURE 7. Location of sensors and actuators in the house

We successfully perform some experiments on the manual mode:

- The elder opens the door by tapping the RFID card on the controller (see Figure 8a. and 8b)
- The elder controls the AC lamp in the living room (see Figure 8c and 8d)
- The elder controls the AC lamp in the kitchen (see Figure 8e and 8f)
- The elder controls the fan in the kitchen (see Figure 8g and 8h)
- The elder operate the gas stove so the amount of gas detected by the sensor will be changed (see Figure 8i)

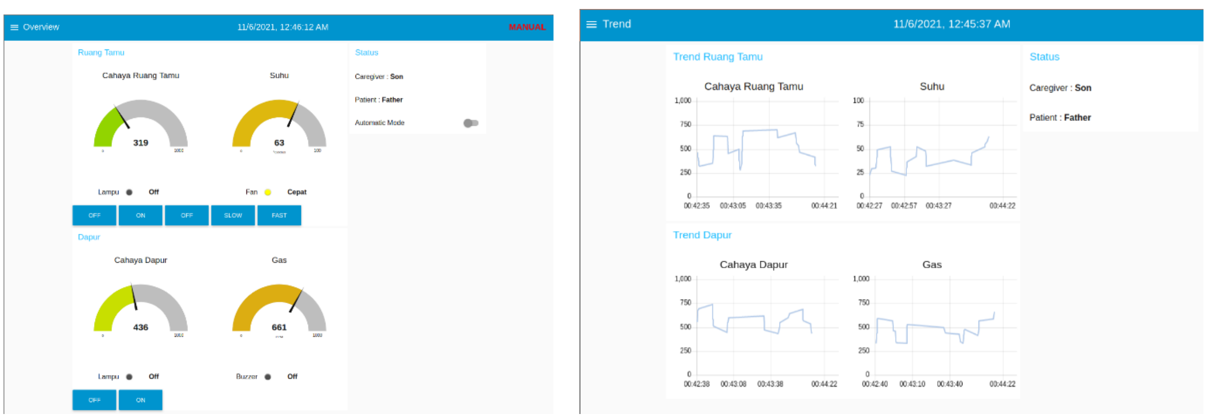
(a) (b)

(c) (d) (e) (f)

(g) (h) (i)

FIGURE 8. The elder performs manual control on the smart home

All of the analog sensor data (such as light sensor, temperature sensor, gas sensor) are displayed in gauges on NodeRed (see Figure 9a). They are also available in trend (see Figure 9b), so it is easier for a user to get the context of the data.



(a) (b)

FIGURE 9. NodeRed dashboards for the smart home

The simple automatic control mode is also successfully applied in our system. By implementing JavaScript program in NodeRed functions, the simple rules mentioned in Section Research Design are applied.

We also implemented the alarm notification feature. If the gas concentration is high, our system might trigger an alarm notification email to the house owner. Whenever the gas concentration is normalized, another notification email will be sent. Figure 10 shows the email received from our system.

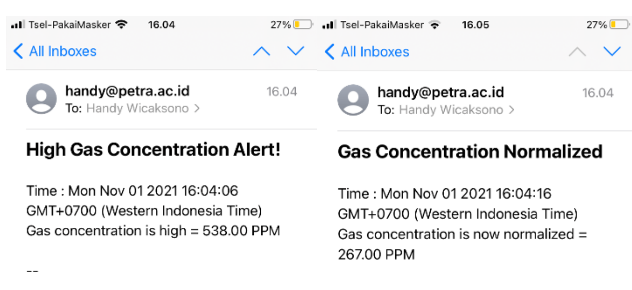


FIGURE 10. Alarm notification email for the houseowner

Experiments on the wearable Device

An experiment is performed to ensure the working mechanism of data acquisition in the framework. For this experiment, a particular subject wears the smartwatch for several days (Figure 11 shows the elder who is wearing the smartwatch).

FIGURE 11. The elder wear the Samsung Galaxy Watch

The smartwatch collects the data and periodically sends it to the slave smartphone. There are two kinds of information collected in the experiment, namely heart rate and step count. Figure 12 and 13 shows the result of the experiment. We are still doing preliminary experiments, so we just test the ability of our system to read that information.

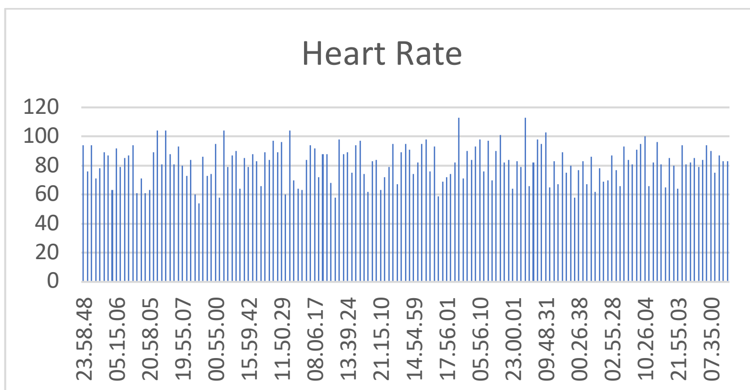


FIGURE 12. The heart rate information

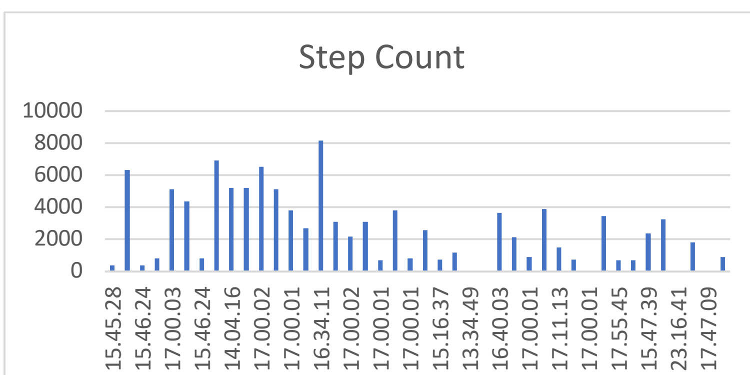


FIGURE 13. The step count information

Another experiment is performed to check the performance of the framework in the case of battery usage. There are three defined scenarios in this experiment, namely continuous usage, periodic heavy usage, and periodic light usage. In continuous usage, the testing device is configured to read various sensors continuously whenever the data is ready. In periodic heavy usage, the sensing rate is limited for a single data reading per minute. For periodic light usage, data reading is limited to every 5 minutes.

The testing process is performed for ten cycles. For each cycle, the smartwatch is used until 5% battery capacity is left. The specification of the tested smartwatch device is shown in Table 1. The testing result is shown in Figure 14. In a continuous sensing scenario, on average, 18.4 percent battery capacity is used per hour. Maximum usage is 19.1 percent per hour, and minimum usage is 17.5 percent per hour. In a periodic-heavy scenario, on average, 14.2 percent battery is used per hour. Maximum usage is 15.3 percent per hour, and minimum usage is 12.1 percent per hour. In the periodic-light scenario, on average, 8 percent battery is used per hour. Maximum usage is 8.5 percent per hour, and minimum usage is 7 percent per hour.

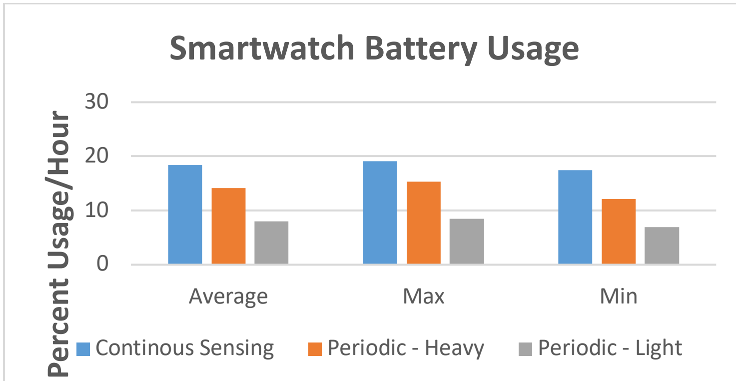


FIGURE 14. The smartwatch battery usage information

CONCLUSIONS

We integrate a smart home system and a smartwatch to assist elders in a home care scenario. We successfully developed and tested the smart home that can be controlled in the manual (via the internet) or automatic (based on sensor readings) mode. Our smart home has four main features: door control, lighting control, temperature control, and gas leakage detection. A buzzer will be turned on, and an email notification will be sent to the house owner if the gas concentration is high. We also test the data collection of heart rate and step count of the elder via smartwatch. We also evaluate the battery usage of the smartwatch in various conditions. Further design and experiments still need to be done to integrate these two components into one system so it can assist the elder effectively.

ACKNOWLEDGMENTS

This research is fully funded by by Hibah Penelitian Terapan Unggulan Perguruan Tinggi 2021 (contract number: 313/E4.1/AK.04.PT/2021) from DRPM Kemristek/BRIN.

REFERENCES

1. J. V Hoof, G. Demiris, and E. J. M. Wouters, *Handbook of smart homes, health care and well-being* (Springer International Publishing, 2017).
2. S. Majumder, E. Aghayi, M. Noferesti, H. Memarzadeh-Tehran, T. Mondal, Z. Pang, and M. Deen, "Smart homes for elderly healthcare—Recent advances and research challenges", in *Sensors* 17, no. 11 (2017)
3. Z. Wang, Z. Yang, and T. Dong. "A review of wearable technologies for elderly care that can accurately track indoor position, recognize physical activities and monitor vital signs in real time." In *Sensors* 17, no. 2 (2017)
4. P. Santoso, F. Pasila, I.H. Putro, "Mobile sensing platform using smartwatch", in *Electrical Power, Electronics, Communications, Controls and Informatics Seminar* (EECCIS), (IEEE, Malang, Indonesia 2018), pp. 430-433.
5. E. Macias, A. Suarez, and J. Lloret, "Mobile sensing systems", in *Sensors*, vol. 13, no. 12, pp. 17292–17321 (2013).

6. A.K. Triantafyllidis, C. Velardo, D. Salvi, S.A. Shah, V.G. Koutkias, L. Tarassenko, "A survey of mobile phone sensing, self reporting, and social sharing for pervasive healthcare", in *IEEE Journal of biomedical and health informatics* 1:218-227(2015).
7. H. Wicaksono, P. Santoso, I. H. Putro, I. S. Hutomo, and P. Alvina, "Towards Integration of Heterogeneous Controllers in an IOT-based Automation System", in *E3S Web of Conferences* Vol. 188 (EDP Sciences, 2020).