Journal of Engineering and Applied Sciences 13 (Special Issue 2): 2920-2925, 2018
ISSN: 1816-949X

Development of Open Source-Based Home Automation for Productivity Improvement

[1]Youngkwan Ju and [2]Hyung-Jin Mun
[1]Department of Software, Chungbuk National University, Gyeonggi-do, Republic of Korea
[2]Department of Information and Communication Engineering, Sungkyul University,
53 Sungkyuldaehak-ro, Manan-gu, Anyang-City, Gyeonggi-do, Republic of Korea

Abstract: The aging level of labor force in Korea is evaluated based on the percentage of workers above the age of 50 in the economically active population. While the percentage of elderly population was 25% in 2000 it is projected that it will reach 40% in 2020 and exceed 50% in 2050. The rapid increase of elderly population is leading to the decrease of productive population. It will also cause an aggravation of domestic productivity. Productivity decline due to the increase of elderly population can become visible within a few years. Information communication technology is finding ways to respond to this problem. Smart technology can enhance productivity through automation and intellectualization in a simple and repetitive labor field. Among smart technologies, the development of IoT technology has been carried out actively. Many examples that can increase productivity using IoT have been suggested widely. This study will describe the development of IoT using open source software and hardware that can minimize the development time of IoT technology. With examples of applying IoT technology to home automation for the development of a similar system in the future, this study seeks to suggest that productivity improvement using IoT communication to solve the problems of the decrease in productive population and productivity decline. In order to show that the issue of productivity decline due to aging can be solved using open source technology, this study suggested a direction by developing home automation using open source hardware and software.

Key words: Open source software, open source hardware, internet of things, machine to machine, home automation, suggested

INTRODUCTION

South Korea’s elderly population of age over 65 was 7,728,000 as of 2014 and it is projected to enter an aging society in 2018 and a post-aged society in 2026 (Min-Sin and Yoo-Ra, 2016). This is a serious issue because aging is expected to progress faster in South Korea than in advanced countries. Due to current aging, the increase of elderly population causes an aggravation of domestic productivity in South Korea which is quickly becoming a society of low-growth, high-cost (Anonymous, 2017; Boockmann *et al*., 2011; Yun-Young and Min-Seok, 2015; Chung *et al*., 2011). Therefore, it is necessary to find an alternative to respond to productivity decline caused by aging population.

In 2005, the ITU defined that IoT is a network of network that provides a network to all things (Gigli and Koo, 2011; Wang *et al*., 2013). IoT technology achieves automation in various fields using communication technology and embedded board technology. Recently, it has become a technology that anyone can realize easily using open source software and open source hardware (Gigli and Koo, 2011; Wang *et al*., 2013).

Home automation is a system for a user to control devices such as boiler, light, washing machine, surveillance sensor, surveillance camera, etc. in a home anytime, anywhere through IoT depending on an indoor temperature (Gigli and Koo, 2011; Wang *et al*., 2013; Gaikwad *et al*., 2015).

By building home automation using open source software and open hardware, this study aims to demonstrate that IoT technology can be applied in various fields and seek a way to improve productivity using this.

Literature review

Open source hardware: Open source hardware means electronic goods with open circuit diagram, parts and development environment that are needed to produce goods and it includes boards such as Arduino, Raspberry Pi, Beaglebone and Galileo as in Fig. 1 (OSHWA., 2017; Jonathan, 2013; Ryu, 2015; Doukas, 2012). Having multiple kBs of small memory and an 8 bit processor, Arduino is a computer that can make simple operations such as running a sensor and sending the information it measured to a server that collects it (Doukas, 2012). Raspberry Pi,

Corresponding Author: Hyung-Jin Mun, Department of Information and Communication Engineering, Sungkyul University, 53 Sungkyuldaehak-ro, Manan-gu, Anyang-City, Gyeonggi-do, Republic of Korea

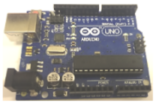
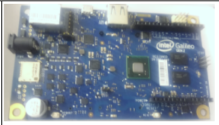
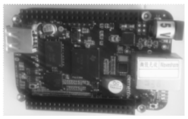
			
Model:Arduino Uno R3 CPU: ATmega 328P Memory: 2 kB USB ports- Ethernet	Model: Raspberry Pi2 CPU: Cortex-A7 Memory: 1 GB USB ports- 4 Ethernet: 10/100 Mbps	Model: Galileo CPU: Inter Qurak Memory: 256 MB USB Ports:- 2 Ethernet: 10/100Mbps	Model: Beaglebone black CPU: Cortex-A8 Memory: 1 GB USB ports- 1 Ethernet: 10/100 Mbps

Fig. 1: Types of open source hardware

Beaglebone and Galileo are high-performance processors that can process the collected information or act as a server based on a memory above 256 MB and high performance of 32 bit ARM processor.

Open source software: Open source software satisfies open source license for the public to use a source code without any limitations by making a source code open to the public (Anonymous, 2017; Feller and Fitzgerald, 2002). Open source software such as a source code or production method of various goods is distributed through Github and other sharing sites in many different fields including editor, development tool, game, network, security, system, research and publishing (Ray *et al*., 2014).

IoT communication protocol: IoT communication protocol includes AMQP (Advanced Message Queuing Protocol) which is a middleware-level message protocol, MQTT (Message Queuing Telemetry Transport), a connecting protocol of message pattern and an XMPP (Extensible Messaging and Presence Protocol) which is an XML-based open source IM protocol. On the other hand, the use of MQTT protocol is increasing as it has been selected as the standard of OASIS consortium (Mun *et al*., 2016). MQTT (Message Queue Telemetry Transport) is a light message protocol optimized for telemetry equipment and mobile device. As a protocol designed by considering the restricted communication environment of mobile products it is optimized for the evolution of the mobile field (Mun *et al*., 2016).

MATERIALS AND METHODS

Home automation design based on open source: To show that IoT can be built using open source hardware and software, this study will introduce its development process in home automation which is the most closely related field to our daily life. Because home automation using IoT technology has a wide range, the range will be set to be MQTT an IoT protocol to build a motor control module used in home automation server and electric control.

Home automation is a system that enables a user to check the status of indoor device and control it using IoT. It consists of a home automation server controlling indoor sensors or actuators or modules controlling indoor operations and information. This study will design a home automation server module needed to develop home automation. Figure 2 shows the composition of home automation monitoring and control system.

As in Fig. 3, the home automation server module consists of mobile view page, message processing module and database module. The home automation server module classifies and stores the data transferred to this server and manages the data needed to be displayed to the user or be put as accumulated data in a separate table. In addition, the home automation server module classifies accumulated data received from the sensor module and monitors whether the sensor value caused by a device error is entered. If a signal that goes beyond the set value is detected in the signal sent from the sensor module, a warning is sent to a monitoring view page. The user can check the information on the error of set value in real-time.

Monitoring view page: Shows the change depending on the data stored in the server to the user. The control can be set by an authorized user through the monitoring view page regardless of time and place.

Message processing module: Process messages of home automation server, converts them into messages to send and receive, receives a message requested by the monitoring view page, analyzes them and interprets and responds to the requested command.

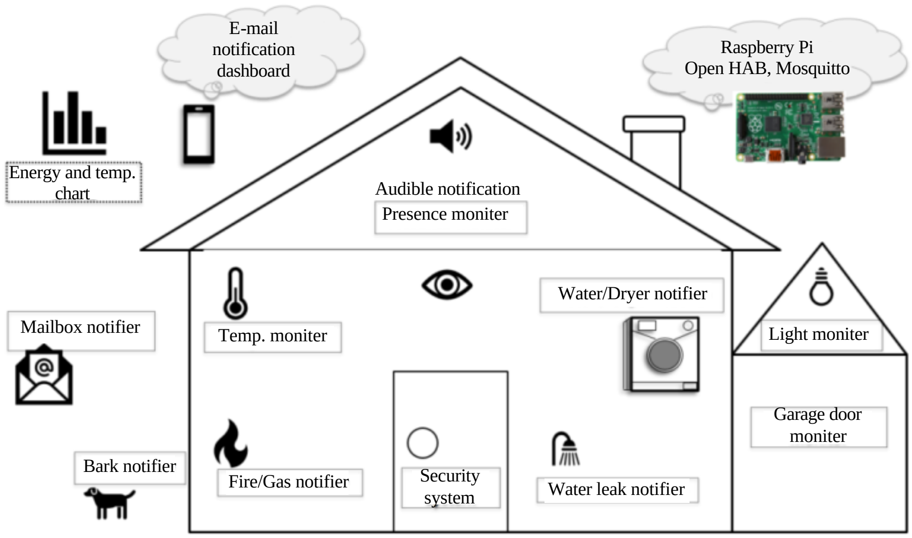


Fig. 2: Home automation

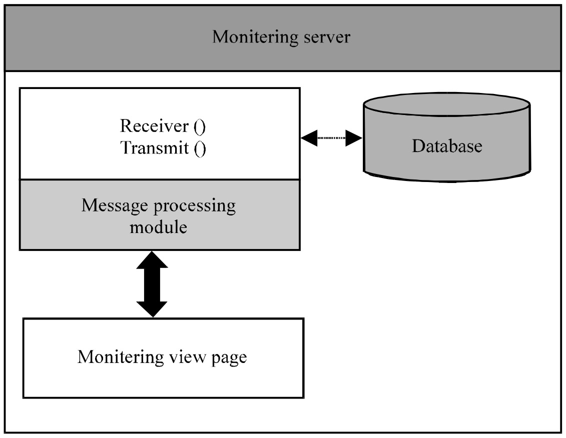


Fig. 3: Home automation server design

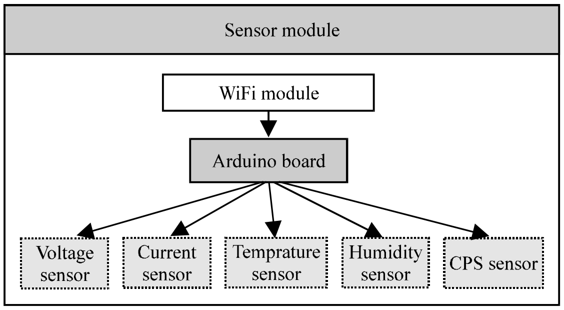


Fig. 4: Sensor module

Database module: Accumulates the sensor value of home automation on the mobile page and stores the data to display.

A sensor module controlling the operation and detects the indoor home automation will be designed. A sensor module installs a sensor in an appropriate place indoor, measures indoor temperature and humidity and converts them to data. The sensor data collected by sensor module is sent to home automation server through WiFi communication module. The sensor module sends the information measured by sensor at every certain time to the home automation server or sends the data commanded by the home automation.

In Fig. 4, the sensor module of home automation is a figure that created a sensor and communication module into an Arduino board. The sensor module was consisted to include temperature, humidity, voltage, current, light, etc. Which are necessary sensors for home automation. This sensor module controls light, temperature and cooling fan according to the set condition when the collected information is transferred to Beaglebone, the monitoring server of home automation. For home automation, all kinds of sensors and actuators are installed indoor to maintain the environment set by the user as well as check, control and monitor the outdoor situation when it is necessary.

RESULTS AND DISCUSSION

Implementation of home automation: A system was developed after selecting hardware and software needed for home automation. Beaglebone was selected as home automation server considering the memory and processing speed of open source hardware and a board controlling Arduino Uno sensor which is widely used was selected for the sensor module. The system is developed by applying open source Linux and sketch for the server and software used in the sensor.

In Beaglebone, the home automation server, MQTT server is installed after Ubuntu 14.04, a Linux OS is installed completely. As in Algorithm 1, mosquitto broker package that supports operations to MQTT server is installed. Next, run the mosquitto broker and develop the system to send (publish and subscribe) a topic to a broker address, the client.

Algorithm 1; MQTT server install:

```
1. Mosquitto broker install
       $ sudo apt-get install mosquitto mosquitto-clients python-mosquitto
2. Mosquitto broker execute
       $ sudo /etc/init.d/mosquitto start
3. MQTT publish & subscribe
       $ mosquitto_pub –h  broker addr –t  topic
```

Using a wired and wireless router, make sure that the home automation server and sensor module send and receive information. Send messages in a MQTT protocol form that supports 1:N connection from IoT protocol between the home automation server, sensor and actuator. Develop the remote control system of actuator and sensor monitoring through the monitoring view page. Algorithm 2 shows a server code in which the home automation server publishes and subscribes a message through sensor module and MQTT, the client. The server code consists of 6 stages.

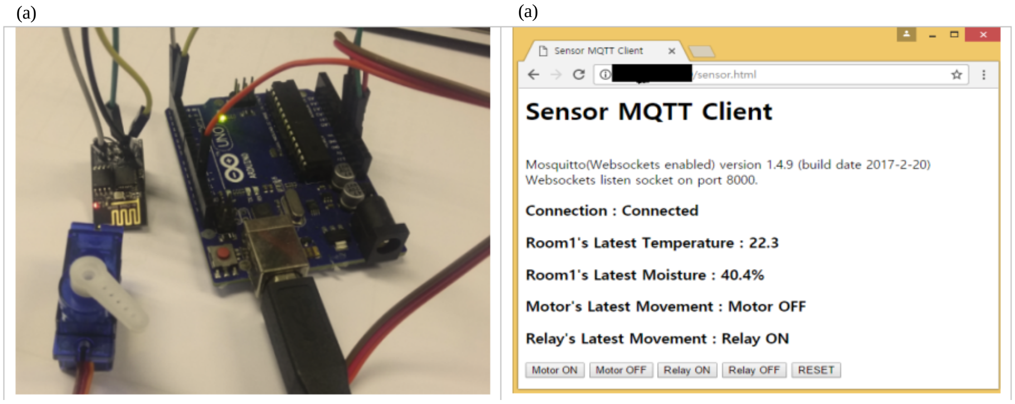
The server creates a client ID using the genId () function when a client accesses it. It displays a disconnected message when the client's connection is lost. If the client sends a subscription request message, the topic and payload received are written on the message study. After checking the topic of message to send, the message topic and payload to send are displayed. A new message object is created to the accessed client and the message is sent through MQTT broker.

Algorithm 2; MQTT server code:

```
1.  var client = new Messaging. Client (“192.168.0.16, 8000, “myclientid-
” + genId (), 10)); client. connect (options)
2.  client.on Connection Lost = function (responseObject) {
alert (“connection lost: “ + response Object. error Message)
 $ (‘#status’). text (“Disconnected”). remove Class (). add Class (“connect”)
 }
3.  client. on Message Arrived = function (message) {
$ ('#messages'). append ('<span>Topic: ' + message.destinationName + ' |
Pay load: '+ message. pay load String + '</span><br/>')
4. if (message. destination Name =’ send topic message ')
$("#send topic message").text(message.payloadString)
 }
5. var publish = function (payload, topic, qos) {
var message = new Messaging. Message (payload)
message. destination Name = topic
message. qos = qos
client. send (message)
}
```

An experiment set was built using Beaglebone for home automation server and Arduino board for sensor module. The sensor module is registered on the home automation server before its use. For the sensor module, DHT11 sensor that can measure temperature and humidity at the same time was used, the indoor temperature it can measure ranges from 40-100°C and the humidity ranges from 0-100%. SRD-05 relay and SG-90 servomotor were used for the control module and ES-8266 WIFI module was used to connect the sensor module and home automation server. In the set, a function was developed to operate a motor which is the actuator or to operate a relay that can turn on or off the power depending on the condition set by the user.

Figure 5 is a module that runs the servo motor as one of (a) sensor modules. This sensor module was developed by ES-8266 WIFI module, SG-90 servomotor, Arduino Uno 3 (b) This screen shows the execution of home automation and controls the server and relay operation of (a). The connect shows the access state of sensor module to the server as well as displays the recent actions of servomotor and relay. The view page was developed



[redacted]/sensor.html

Sensor MQTT Client

Mosquitto(Websockets enabled) version 1.4.9 (build date 2017-2-20)
Websockets listen socket on port 8000.

Connection : Connected

Room1's Latest Temperature : 22.3

Room1's Latest Moisture : 40.4%

Motor's Latest Movement : Motor OFF

Relay's Latest Movement : Relay ON

Fig. 5: Motor and relay control screen: a) Example of servo-motor module and b) Sensor control page

using HTML and the motor was made to be controlled immediately when requested by the home automation server.

Algorithm 3 shows the control code of relay attached on the sensor module. Relay control code is a core code that turns On/Off the relay by receiving a 'relay' topic through the WiFi connected to Arduino and by interpreting a command saved on the payload.

The code forms a domain address and port number for monitoring server access. A communication channel is formed by accessing a monitoring server. To prevent collision among sensors, a unique ID is given and a 'relay' message is transferred to a message client. The Callback () function extracts an actual operation from control code Payload according to the transferred message. The actuator has been made to operate according to the extracted message.

Algorithm 3; Relay control code:

```
# include <ESP8266 WiFi.h>
#include <PubSubClient.h>
      1. #define mqtt_server “192.168.0.100”
#define mqtt_port 18832
      2. const char* ssid = “HomeAutomation”
const char* password = “------password”
WiFiClient espClient
PubSubClient client(espClient)
void setup() {
 …………………….. //Serial configuration settting
WiFi.begin(ssid, password)
while (WiFi.status() != WL_CONNECTED)  Display “connect Error”
Serial.println(“WiFi connected”)
 Serial.println(“IP address: “)
      Serial.println(WiFi.localIP())
 client.setServer(mqtt_server, mqtt_port)
       client.setCallback(callback)
}
void loop() {
3.   if (!client.connected()) {
      Serial.print(“Attempting MQTT connection, ...,”)
if (client.connect(“Servo00001")) {
       Serial.println(“connected”)
            client.subscribe(“Servo01”)
      } else    {   ……………………..//  Display “not connected”
  return;   }
}
    client.loop()
}
4. void callback(char* topic, byte* payload, unsigned int length) {
String msg = “”
 int i = 0
 while (i<length) msg += (char)payload[i++]
  digitalWrite(LED_PIN, (msg == “Relay ON” ? HIGH : LOW))
 Serial.println(msg)
}
```

CONCLUSION

In South Korea, elderly population over the age of 65 is increasing quickly while working age population engaged in economic activities is decreasing rapidly. It is expected that productivity will aggravate mainly in intensive labor industries such as industry and agriculture or fishing. To respond to productivity aggravation, automation through IoT technology among information communication technologies, can be a suitable alternative. Using open source software and open hardware on the internet will also be able to reduce cost and enable automation to be built easily in various fields that need this technology.

RECOMMENDATIONS

This study showed an example of home automation development that can measure temperature and humidity as well as control a motor that can open and close a window based on IoT technology. Home automation demonstrated that it can reduce cost through its development using open source hardware and software and that productivity can be improved through automation. Future studies will be conducted on productivity improvement using database accumulated by IoT.

REFERENCES

Anonymous, 2017. European foundation for the improvement of living and working conditions, employment initiatives for an ageing workforce. European Union, Brussels, Belgium. https://europa.eu/european-union/about-eu/agencies/eurofound_en.

Boockmann, B., J. Fries and G. Christian, 2011. Specific measures for older employees and late career employement. Zentrum für Europäische Wirtschaftsforschung, Mannheim, Germany.

Chung, S.D., H.J. Park and B.K. Kim, 2011. Public perception and countermeasures on an aging society a content analysis of newspaper articles. Korean J. Soc. Welfare, 63: 203-224.

Doukas, C., 2012. Building Internet of Things with the ARDUINO. CreateSpace, USA., ISBN:9781470023430, Pages: 340.

Feller, J. and B. Fitzgerald, 2002. Understanding Open Source Software Development. Addison-Wesley, Boston, Massachusetts, USA., ISBN:9780201734966, Pages: 211.

Gaikwad, P.P., J.P. Gabhane and S.S. Golait, 2015. A survey based on smart homes system using internet-of-things. Proceedings of the 2015 International Conference on Computation of Power, Energy Information and Commuincation (ICCPEIC), April 22-23, 2015, IEEE, Chennai, India, ISBN:978-1-4673-6525-3, pp: 0330-0335.

Gigli, M. and S. Koo, 2011. Internet of things: Services and applications categorization. Adv. Internet Things, 1: 27-31.

Jonathan, L., 2013. Open source hardware. Master Thesis, Department of Technology Management and Economics, Chalmers University of Technology, Gothenburg, Sweden.

Min-Sin, S. and J. Yoo-Ra, 2016. Aging society against U-health care service industry promotion plan. Korean J. Sports Sci., 25: 337-347.

Mun, H.J., G.H. Choi and Y. Hwang, 2016. Countermeasure to underlying security threats in IoT communication. Convergence Soc. SMB., 6: 37-44.

OSHWA., 2017. Open hardware summit 2017 hotel rooms and post-summit events. Open Source Hardware Association, New York, USA. http://www.oshwa.org/.

Ray, B., D. Posnett, V. Filkov and P. Devanbu, 2014. A large scale study of programming languages and code quality in github. Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering, November 16-21, 2014, ACM, Hong Kong, China, ISBN:978-1-4503-3056-5, pp: 155-165.

Ryu, D.H., 2015. Development of BLE sensor module based on open source for IoT applications. J. Korea Inst. Electron. Commun. Sci., 10: 419-424.

Wang, M., G. Zhang, C. Zhang, J. Zhang and C. Li, 2013. An IoT-based appliance control system for smart homes. Proceedings of the 2013 14th International Conference on Intelligent Control and Information Processing (ICICIP), June 9-11, 2013, IEEE, Beijing, China, ISBN:978-1-4673-6248-1, pp: 744-747.

Yun-Young, K. and B. Min-Seok, 2015. Social investment for older people. Korean Assoc. Comp. Government, 19: 23-48.