논문 21-46-04-15 The Journal of Korean Institute of Communications and Information Sciences '21-04 Vol.46 No.04
https://doi.org/10.7840/kics.2021.46.4.696

사물인터넷 기반 사무실 출입 제어 시스템 구현

안 젤 라•, 헤 나 르*, 아 딥*, 이 제 이*, 임 완 수°

Design and Implementation of an IoT-Based Door Signage Control System

Angela C. Caliwag•, Henar M. O. Canilang*,
M. Adib Kamali*, EJ M. F. C. Caliwag*, Wansu Lim°

요 약

IoT를 응용한 기술은 가정용품부터 스마트공장까지 넓은 영역에서 활용되고 있다. 본 논문은 사무실에서 활용할 수 있는 IoT 응용 기술을 제안한다. 일반적으로 사무실 출입을 위해서는 사무실에 출입하려는 사람은 노크하고, 사무실 안에 있는 사람은 노크를 인지하고 사무실 상황에 따라 출입을 허용한다. 그러나 소리에 의한 의사전달로 인해 사무실 안과 밖에 있는 사람의 의사전달이 불명확할 뿐 아니라 사무실 안에서 중요한 회의가 있다면 회의를 방해할 수도 있다. 이러한 문제점을 해결하기 위해 본 논문은 사무실 안과 밖의 원활하고 효율적인 의사소통을 위해 IoT기술 기반 사무실 출입 제어 시스템을 제안한다. 제안하는 사무실 출입 제어 시스템은 사무실 안에 있는 사람의 의사를 전달하는 제어부와 의사 표현이 표시되는 표시부로 구성한다. 제어부와 표시부는 각각의 기능에 따른 알고리즘을 구현하고 사업화 가능성을 확인하기 위해 PCB로 설계하여 시작품 제작하였다.

키워드 : 사물인터넷, 스마트 기기, 임베디드 시스템, 신호처리, 시작품 구현
Key Words : IoT, smart devices, embedded systems, signal processing, product implementation

ABSTRACT

Door signage system is used to indicate the status of the person inside a room. This is helpful for both the room owner and the outsider. Conventional door signages needs both time and effort to accurately indicate the status of the person inside the room. This paper describes the design and implementation of an IoT-based door signage. The IoT-based door signage reduces the time to change the status of the room, increases the accuracy of the door signage and eliminates the need of physical contact with the signage.

Ⅰ. Introduction

Nowadays, several IoT systems and devices have been developed[1,2]. Such systems and device were also proven to change and improve people's lives. Despite that fact, researchers do not stop and continue the development, with the main goal of reducing the time and effort needed to perform any

※ 본 연구는 2020년 국립대학 육성사업비로 지원되었음.
• First Author : Kumoh National Institute of Technology, Department of Aeronautics, Mechanical and Electronic Convergence Engineering, a_caliwag@kumoh.ac.kr, 학생회원
° Corresponding Author : Kumoh National Institute of Technology, Department of Aeronautics, Mechanical and Electronic Convergence Engineering, wansu.lim@kumoh.ac.kr, 정회원
* Kumoh National Institute of Technology, Department of Aeronautics, Mechanical and Electronic Convergence Engineering
논문번호 : 202012-302-D-RN, Received December 7, 2020; Revised January 19, 2021; Accepted January 19, 2021

task[3,4]. The IoT systems and devices nowadays are able to replace humans in performing some process necessary in accomplishing a task. To do this, the systems and devices must be developed or configured to function in a similar way a specific part of human body functions.

In [5], a system that detects a forgotten child inside an enclosed car is developed. In this paper, IoT system does not only replace the human but also exceeds its capability in performing the task. Unlike human, the IoT device can monitor the car 24 hours in a day. In [6], a system that monitors the environment of fish, and controls the feeding time brightness of the LED lamps is developed. The IoT system is used to automate the monitoring and control tasks. As a result, the tasks are performed with higher accuracy. In [7], a system that controls the pedestrian crossing areas is designed and developed. In this study, the IoT system is designed to automatically detect the presence of pedestrians and controls the crosswalk traffic lights. In [8], an IoT-based embedded system that logs and monitors the weather is designed and developed. The study integrates electronic devices, sensors and wireless technology to sense, store, and visualize the climate parameters such as temperature, humidity, and some gases. In [9], a mobility recognition system capable of assisting visually challenged individuals is developed. The system is designed to recognize manholes and stairs using ultrasonic sensors. In addition to that, the system also uses camera from mobile phones. The input from the camera is sent to the system wirelessly via Bluetooth. Once a manhole or staircase is detected, the system notifies the user using sound or vibration. In [10], an IoT based smart shopping cart is developed. The study utilized radio frequency identification (RFID) sensors, Arduino microcontroller (MCU), Bluetooth module and a mobile application. The RFID sensors and Bluetooth modules are directly connected to the Arduino MCU. The RFID sensors detect the products; and the Arduino sends the product information to the mobile application or to the supermarket wirelessly via Bluetooth for user management or billing, respectively.

In this paper, we aim to improve the conventional door signage by developing an IoT-based door signage control system. Door signages indicates the state of the person in this room. The current door signage systems require the person inside the room to manually and physically change the indication on the door signage. The manual operation of conventional signage systems requires both time and effort for the person inside to accurately indicate the status of that person.

This paper focuses on the design and implementation of an IoT-based door signage control system. Instead of manually changing the status of the person by going to the door and physically switching the indicator, we integrate a simple wireless network to change the status. The IoT network enables the person to change the status without having physical contact with the door signage. The IoT-based door signage is expected to improve the conventional door signage by 1) reducing the time needed to change the status of the door signage, 2) increasing the accuracy and reliability of door signage by adding more status, and 3) eliminating the need of physical contact with the door signage to change its status.

Ⅱ. Circuit design

The IoT-based door signage control system developed in this study consists of two main components: 1) signage and 2) controller. The circuit design of signage and controller are shown in Fig. 1. The main objective in circuit design is to effectively connect all the components together such that the system requirements are met. The system requirements are defined as: 1) signage and controller must communicate wirelessly, 2) the status must be indicated with LEDs, 3) the signage and controller must alert the user whenever an inquiry or a command is sent using buzzers, and 4) both signage and LEDs must be powered with batteries.

First, to enable the wireless communication between the signage and the controller, two components are required: 1) MCU and 2) wireless

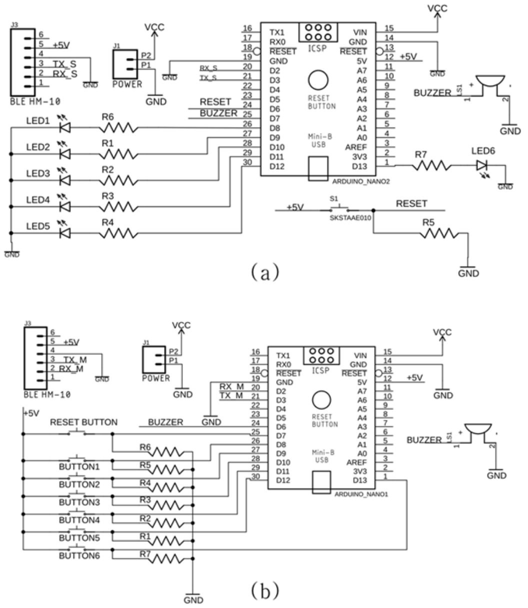


Fig. 1. Circuit design of Arduino (a) signage, and (b) controller

modules. In this paper we used Arduino nano as the MCU and low power HM-10 Bluetooth modules as the wireless module. HM-10 Bluetooth modules enable two MCUs to send and receive data using the Bluetooth protocol without consuming much power from the battery. The HM-10 module has four pins: 1) VCC, 2) RX, 3) TX, and 4) GND. VCC and GND pins are used to power the Bluetooth module. In our design, we used the +5V output from the on-board regulator of the Arduino nano to supply power to the Bluetooth module. Hence, the VCC pin of the HM-10 module is connect to pin 12 of the Arduino nano. On the other hand, the RX and TX pins are used to send and receive data from one MCU to another. The Arduino nano has digital input/output pins that can be configured as serial RX and TX pins. In this paper we configured D2 (pin 20) and D3 (pin 21) of the Arduino board as RX and TX pins respectively. It is important to note that the RX and TX pins of the Arduino nano must be connected to the TX and RX pins of the HM-10 Bluetooth module respectively. The connection of Arduino nano and HM-10 Bluetooth module is same for both signage and controller.

Second, to enable the signage to display the status using the LEDs, two components are required: 1) LEDs and 2) current-limiting resistors. LEDs must be turned on or off depending on the command from the controller. To be able to control the state of the LED (on or off), we connect its positive terminal to the digital pins of the Arduino nano and its negative terminal to the ground. Hence, a closed circuit is formed with the digital pins of the Arduino nano as the variable supply (+5V or 0V). The digital pins of the Arduino nano can be programmed to output +5V or 0V by writing a value of 1 or 0 respectively. In addition to that, a current-limiting resistor is placed in series with the LEDs for protection. Since we used 6 different status, we connected 6 LEDs and resistors to pins 25 (D7) to 30 (D12) of Arduino nano.

Third, to alert the user whenever an inquiry or a command is sent, two components are required: 1) buzzer and 2) current-limiting resistors. Similar to LEDs, the buzzer is also turned on or off. To be able to control the state of the buzzer (turned on or off), we connect its positive terminal to the digital pin of Arduino nano and its negative terminal to the ground. The signage and controller have one buzzer each. In our design, we connect the buzzer to pin 24 (D6) of Arduino nano.

Finally, to enable the signage and controller to be powered with batteries, two components are required: 1) battery connection header and 2) battery itself. The battery connection header establishes a reliable connection between the battery and the signage/controller. The battery is used to supply power to the system. The battery used is 4 pieces of AA battery cells connected in series. One AA battery cell has a nominal voltage of 1.5V. Hence, 4 AA battery cells connected in series has a total nominal voltage of 6V. The battery only supplies power to the Arduino nano which has a maximum power consumption of 0.2W. In our design, we connect the one terminal of the battery connection header to the VIN pin (supply voltage pin) of the Arduino nano and the other terminal to the GND pin (supply ground pin).

Ⅲ. Software design

In this section we discuss in detail the design of the software part of the IoT-based door signage. First, the design of algorithm is explained using a flow chart of the overall process. Then, the wireless communication between the signage and controller is described in detail. Finally, the overall process and flow of the data from the user of the controller to the user of the signage, and vice versa is discussed.

3.1 Algorithm design

The block diagram of the IoT-based door signage that we developed is shown in Fig. 2. To enable the wireless control of door signage, a controller and a signage capable of sending, receiving and responding to the command is developed. As shown in Fig. 2, conventional signage is modified such that it can receive status and clear command from the controller. The status command is a command sent by the controller which asks the signage to display the corresponding status using the LED indicators. Once the status command is sent, the signage activates that status using the LED indicators for a certain amount of time. The clear command is a command sent by the controller which asks the signage to stop displaying any status that is currently activated by a previous status command. Once the clear command is sent, the activated status command is interrupted. The status inquiry is an inquiry sent by the signage to the controller which indicates that no status is currently activated and that the status is unknown.

A flowchart of the IoT-based door signage algorithm is shown in Fig. 3. As shown in the figure, the signage has two modes: 1) standby mode and 2) display mode. In standby mode, the signage

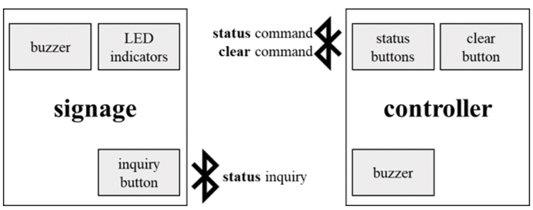


Fig. 2. Diagram of door signage control system

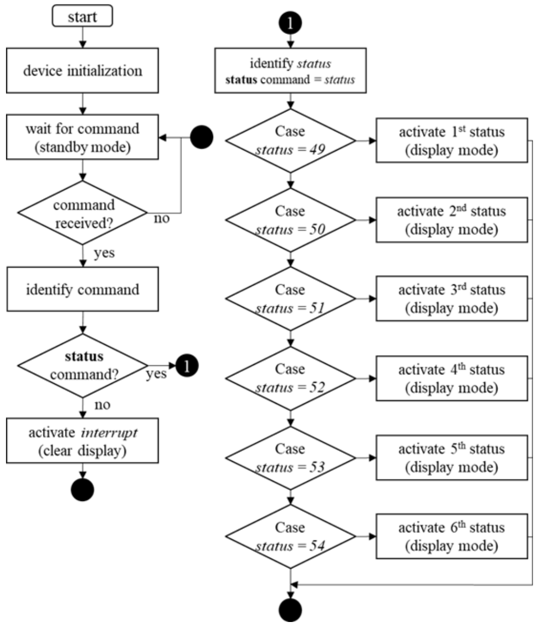


Fig. 3. Flow chart of IoT-based door signage algorithm

is simply waiting for the command from the controller. Once a command from the controller is received, the signage identifies the type of command: status command or clear command. If a status command is detected, the type of status is then identified. In this paper we set six status: 1) "please come in", 2) "please come back later", 3) "in a meeting", 4) "in a call", 5) "not available", 6) left as spare. Depending on the status number contained in the status command received, the corresponding status is activated. If a clear command is detected, the activated status is cleared and the system goes back to standby mode. To further illustrate this, a state diagram is shown in Fig. 4.

As shown in Fig. 4, the IoT-based door signage has two modes or states: 1) standby mode and 2) display mode. In display mode all LED indicators

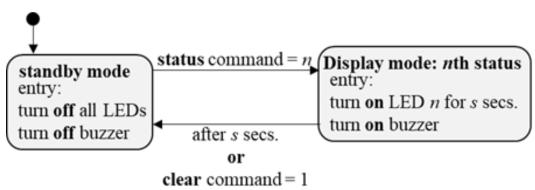


Fig. 4. IoT-based door signage state diagram.

and the buzzers are turned off. In display mode, the LED corresponding to the activated states and the buzzer are turned on for a certain amount of time. In Fig. 4, LED n is turned on for s seconds after the status n is activated. To exit standby mode, the controller simply needs to send a status command to the signage. After status command is activated, the mode is shifted to the display of the corresponding status activated. To exit the display mode, two methods can be performed: 1) wait for the time set to execute the display and 2) the controller simply needs to send a clear command to the signage. In Fig. 4, LED n is set to turn on for seconds after entering the display mode. After seconds, the mode is automatically set to standby mode. However, when a clear command is sent by the controller, the mode is set to standby mode even though the time requirement (s seconds) is not yet satisfied.

3.2 Wireless communication

To enable the wireless control of door signage, a controller capable of sending/receiving commands and a signage capable of sending, receiving and responding to the commands is developed. The wireless communication between the controller and signage is established using Bluetooth as the communication interface. The Bluetooth module we used is HC-05, which is a Bluetooth SPP (Serial port protocol) module. It uses a SPP which means it communicates with the Arduino using Serial communication. In a SPP, the data is transferred one bit at a time. SPP has three transmission modes: 1) simplex, 2) half duplex and 3) full duplex. Simplex method is a one-way communication technique in which a module can only either send or receive data. In half duplex method, a module can both send and receive data, but not at the same time. In full duplex method, a module can send and receive data at the same time. In developing the IoT-based door signage we utilized Bluetooth SPP module with full duplex mode.

In developing the IoT-based door signage, we used a SPP Bluetooth module for the wireless signage control. In wireless signage control, first, the command is defined by the user. The command is assigned based on the button the user presses. Then, the command is identified by the signage using the flowchart in Fig. 3. In identifying the command, a corresponding decimal value is assigned to the *command* register. After the command is identified, the value of *command* register is sent to the signage. Finally, the command is processed by the signage.

The value of the *command* register is sent from the signage to the controller via SPP. Before sending the value via SPP, it is necessary to be converted its equivalent 8-bit binary value. This 8-bit binary value is then sent to the signage one bit at a time. The Bluetooth frame format in sending the value of the *command* register is shown in Fig. 5.

As shown in Fig. 5, the communication starts as soon as the transition from logic 1 to logic 0 is detected. Hence, the start bit is set to logic 0. After the logic 0 is detected, the 8-bit (Word) data is transmitted one bit at a time starting at the least significant bit (LSB). After successfully sending the 8-bit data, the logic transitions back to 1.

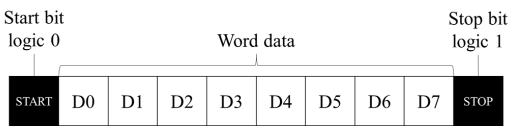


Fig. 5. Bluetooth frame structure

3.3 Overall process

The overall process governing the IoT-based door signage is shown in Fig. 6. The process starts when a user sends a command. IoT-based door signage consists of a controller and a signage capable of both sending and receiving commands. The commands can either be a 1) status command or clear command or 2) status inquiry. The status and clear commands are sent from the controller to the signage while the status inquiry is sent from the signage to the controller. The status command asks the signage to activate a status by turning on the corresponding status indicators. The clear command clears the previously displayed status on the signage. The status inquiry turns on the indicator on the controller which notifies and requests the user of the controller to activate a status on the signage. Both

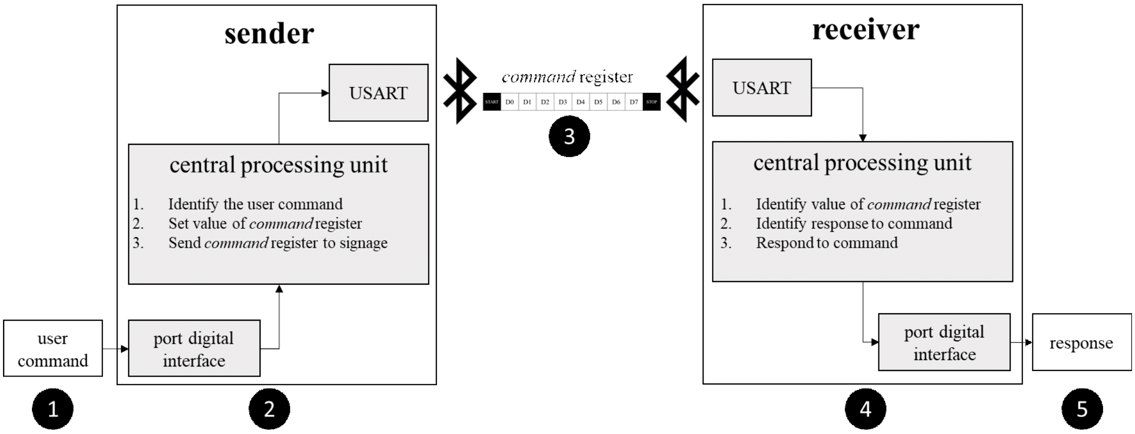


Fig. 6. Overall process for IoT-based door signage

controller and signage are capable of sending and receiving commands at the same time due to the use of the Bluetooth SPP module with full duplex transmission mode. Hence, in operating the IoT-based door signage, both controller and signage can be THE sender and/or receiver. Specifically, the user sends a command by pressing a button that corresponds to the desired command. As show in the Fig. 6. the sender receives the command from the user through the port digital interface of the device. Each button has a distinct decimal value which is received by the sender through the port digital interface. The values are listed in Table 1.

Second, the command is processed by the central processing unit (CPU). In processing, the value received by the sender through the port digital interface is identified. Then, the value is assigned to the *command* register. Finally, the value in the *command* register is sent to the signage serially through the Bluetooth SPP module. The Bluetooth SPP module is interfaced with the CPU through the universal synchronous/asynchronous receiver/transmitter (USART). The USART is configure to transmit and receive 8-bits of data or Word. To send the value in the *command* register is converted to decimal value first, then to 8-bit binary value.

Third, the 8-bit binary value is sent from the sender to the receiver at a specified baud rate one bit at a time. The baud rate is defined as the rate at which the data is transmitted. In the IoT-based door signage, the baud rate is set to 9600 bits per second. Since each command consist of 8-bits, a baud rate of 9600 bits per second enables 1,200 commands to be sent consecutively.

Fourth, the 8-bit data is received by the receiver through the Bluetooth SPP module. The Bluetooth SPP module is interfaced with the CPU through the USART. The 8-bit data is processed by the CPU. In processing, the value in the *command* register is identified by converting the 8-bit data to decimal. After identifying the value of the *command* register, the corresponding response of the command is identified using switch case algorithm. The response corresponding to the commands are listed in Table 1.

Fifth, the response is activated using the port digital interface that is physically connected to the hardware peripherals of the controller and the receiver.

For further understanding, an example is provided as follows: First, the command is sent by the user by pressing button 1 on the controller. Then, the distinct decimal value corresponding to that button is identified, which is given in Table 1 as “49”. That value is stored to the *command* register. After that, the value in the *command* register is converted to 8-bit binary value. The corresponding 8-bit binary value is “00110001”. Then, each bit of the 8-bit binary value is sent to the signage starting from the LSB. In this case, bit “1”. Then “0”, “0”, “0”, “1”, until all bits are sent. Similarly, the signage also

Table 1. List of commands and its corresponding buttons, decimal value and response.

Command	Button	Decimal	Response
"please come in"	Controller button 1	49	Turn on signage LED 1 and buzzer
"please come back later"	Controller button 2	50	Turn on signage LED 2 and buzzer
"in a meeting"	Controller button 3	51	Turn on signage LED 3 and buzzer
"in a call"	Controller button 4	52	Turn on signage LED 4 and buzzer
"not available"	Controller button 5	53	Turn on signage LED 5 and buzzer
left as spare	Controller button 6	54	Turn on signage LED 6 and buzzer
CLEAR	Controller button 7	55	Turn off all LED and buzzer
STATUS inquiry	Signage button	56	Turn on controller buzzer

received the 8-bit binary value expecting the LSB first. The 8-bit data is sent and received wirelessly through the Bluetooth SPP module interfaced with the USART. After the signage receives all 8-bit value, it is converted back to its decimal value. Then, the decimal value is passed thru a switch case algorithm to determine the response corresponding to the command. In this case, the response corresponding to the command when the button with a value of "49" is pressed, as listed in Table 1, as "Turn on signage LED 1 and buzzer." Finally, the LED 1 and buzzer are turned on through the port digital interface on the signage.

Ⅳ. Actual device development

The actual device is built based on the designed circuit and software. After the circuit a software is designed, a prototype is built. A prototype is built to test both the circuit and software design. In building the prototype, a bread board is used first to enable the flexibility in changing the circuit. After the circuit and the software is tested and verified, the components can be transferred from the bread board to a universal printed circuit board (PCB) for further testing and sampling. We soldered all components on a universal PCB. This is helpful to test the robustness of the hardware. Using a bread board only is not practical since the components are not held in place. Hence, by soldering the components on a universal PCB, further test can be performed to verify the reliability and efficiency of both circuit and software design.

After verifying the circuit and software design, we built the actual device. The following steps are performed to build the actual device: 1) printing the circuit on a board (printed circuit board), 2) assembling the components on the boards, 3) uploading the software on the MCU, and 4) creating a device housing to protect its components. First, to print the circuit on the board, the board has to be designed. The PCB design is similar to the circuit design except that in circuit design, we don't mind the placement of components. After the PCB design is completed, the board can then be fabricated. An example of the result of PCB fabrication is shown in Fig. 7.

After the PCB is fabricated, the components can then be assembled on the board. An example of PCB assembly is shown in Fig. 8 (a). Finally, after the PCB is assembled, the device housing can be placed to protect the components. An example of the

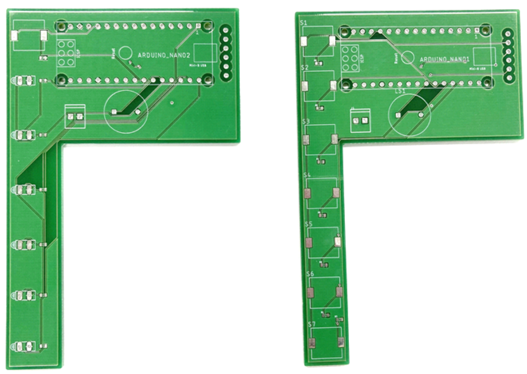
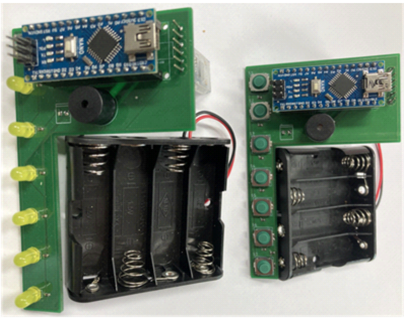


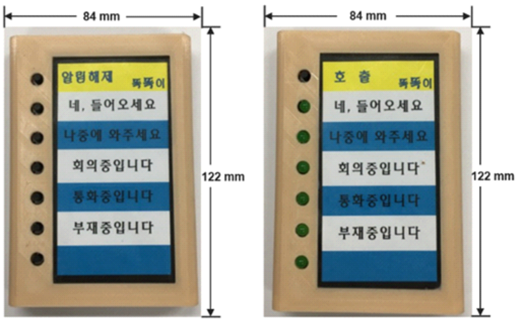
Fig. 7. Actual board of signage and controller

device housing is shown in Fig. 8 (b).

Overall, the development consists of four steps: 1) circuit design, 2) software design, 3) prototype, and 4) actual device. Steps 1 and 2 are performed at the same time. That is, the software is considered in designing the circuit and vice versa. After every completion of circuit and software design, the prototype is developed to test and find the possible errors in the current design. If error is found, then the circuit or the software has to be re-designed. The re-designed version is then put to test using the prototype. The design-test cycle is performed until all the requirements of the system are satisfied. After successfully building the prototype, the actual device is then built by printing the circuit on a board (PCB), assembling the components on the PCB, uploading the software on the MCU, and creating a device housing to protect its components.



(a)



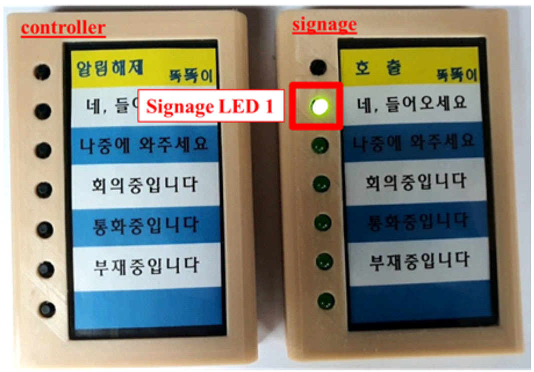
(b)

Fig. 8. IoT-based door signage (a) PCB assembly and (b) final device

V. Results and discussion

After the developing the device, we performed tests to verify the reliability and efficiency of both circuit and software design. The test conducted is described as follows. First, the controller is tested by sending different commands (user status and clear command) to the signage. Each command is sent for 50 trials each. An example testing scenario is shown in Fig. 9. Fig. 9(a) shows that the button 1 on the controller is pressed. Button 1 on the controller corresponds to the status "please come in status" which is pressed to activate LED 1 on the signage. Fig. 9(b) shows the response of the signage when button 1 on the controller is pressed. The response is shown by activating LED 1. LED 1 corresponds to the status "please come in. In addition to that, the buzzer inside the device housing turns on with the LED simultaneously. Similar procedures were performed with the remaining buttons on both the controller and signage. Each procedure performed corresponds to one trial and a total of 50 trials were

(a)



(b)

Fig. 9. Example process of sending command from controller to signage

Table 2. Result of testing IoT-based door signage device for 50 trials

BUTTON PRESSED	RESPONSE	TRIALS	ERRORS
Controller button 1	Turn on signage LED 1 and buzzer	50	0
Controller button 2	Turn on signage LED 2 and buzzer	50	0
Controller button 3	Turn on signage LED 3 and buzzer	50	0
Controller button 4	Turn on signage LED 4 and buzzer	50	0
Controller button 5	Turn on signage LED 5 and buzzer	50	0
Controller button 6	Turn on signage LED 6 and buzzer	50	0
Controller button 7	Turn off all LED and buzzer	50	0
Signage button	Turn on controller buzzer	50	0

performed. The result is listed in Table 2.

The result listed in Table 2 shows that out of 50 trials in sending commands from the controller to the signage and vice versa, 0 failure was encountered. That is, all status commands sent from the controller is received and displayed by the signage. Similarly, all clear commands sent by the controller deactivates all the status being displayed by the signage. Furthermore, all status inquiry sent from the signage is received by the controller.

Ⅵ. Conclusion

In this paper, we described in detail the process of developing the IoT-based door signage control system. The process includes circuit design, software design, prototype building and actual device building. Using MCUs and wireless modules, we are able to improve conventional door signage by enabling wireless operation and control. In addition to that, we also presented the advantage of the IoT-based door signage over the conventional ones. The device developed is expected to lead to further developments of objects into devices that provides more convenience to our daily lives.

References

[1] C. H. Yoon and J. I. Kim, "A systematic review on AI platform for public sector," *J. KICS*, vol. 45, no. 11, pp. 1994-2003, 2020.

[2] J. W. Son and S. M. Hdng, "Provision of responsive web user interface service for wireless router network setting in iot home hubs and devices," *J. KICS*, vol. 45, no. 2, pp. 368-374, 2020.

[3] S. M. Yang, W. J. Song, I. S. Choi, and S. J. Yoo, "Deep learning-based motion classification system for iot device control in ultrasonic sound," *J. KICS*, vol. 42, no. 9, pp. 1796-1805, 2017.

[4] Y. Y. Sung, H. J. Woo, and M. J. Lee, "Organizing the smart devices' set for control of periodic sensing data in internet of things," *J. KICS*, vol. 42, no. 4, pp. 758-767, 2017.

[5] P. Visconti, R. D. Fazio, P. Costantini, S. Miccoli, and D. Cafagna, "Innovative complete solution for health safety of children unintentionally forgotten in a car: A smart arduino-based system with user app for remote control," *IET Sci., Measurement & Technol.*, pp. 665-675, Jul. 2020.

[6] S. J. Hsiao and W. T. Sung, "Building a fish-vegetable coexistence system based on a wireless sensor network," *IEEE Access*, pp. 192119-192131, Oct. 2020.

[7] W. K. Saad, Y. Hashim, and W. A. Jabbar, "Design and implementation of portable smart wireless pedestrian crossing control system," *IEEE Access*, pp. 106109-106120, Jun. 2020.

[8] Y. W. Lin, Y. B. Lin, M. T. Yang, and J. H. Lin, "ArduTalk: An arduino network application development platform based on IoTtalk," *IEEE Syst. J.*, vol. 13, no. 1, pp.

468-476, 2019.

[9] S. Ponnada, S. Yarramalle, and R. T. V. Madhusudhana, “A hybrid approach for identification of manhole and staircase to assist visually challenged,” *IEEE Access*, vol. 6, pp. 41013-41022, 2018.

[10] M. Shahroz, M. F. Mushtaq, M. Ahmad, S. Ullah, A. Mehmood, and G. S. Choi, “IoT-based smart shopping cart using radio frequency identification,” *IEEE Access*, vol. 8, pp. 68426-68438, Apr. 2018

안 젤 라 (Angela C. Caliwag)
2019년 10월~현재 : 금오공대, 항공기계융합전공
<관심분야> 임베디드 시스템, 지능형 제어

헤 나 르 (Henar M. O. Canilang)
2020년 3월~현재 : 금오공대, 항공기계융합전공
<관심분야> 임베디드 시스템, 지능형 제어

아 딥 (M. Adib Kamali)
2020년 3월~현재 : 금오공대, 항공기계융합전공
<관심분야> 임베디드 시스템, 지능형 제어

이 제 이 (EJ M. F. C. Caliwag)
2020년 3월~현재 : 금오공대, 항공기계융합전공
<관심분야> 임베디드 시스템, 지능형 제어

임 완 수 (Wansu Lim)
2014년 9월~현재 : 금오공대 전자공학부 교수
<관심분야> 지능형 제어, 임베디드 시스템
[ORCID:0000-0003-2533-3496]