논문 22-47-02-14 The Journal of Korean Institute of Communications and Information Sciences '22-02 Vol.47 No.02
https://doi.org/10.7840/kics.2022.47.2.319

비콘을 활용한 청각장애인용 화재 대피 시스템의 구현

박 지 훈◆, 이 소 연*, 정 민 우**, 김 대 영°

Fire Evacuation System Using Beacon for Hard-of-Hearning People

Jihoon Park◆, SoYeon Lee*, Minwoo Jung**, Dae-Young Kim°

요 약

청각장애인은 옥내에 화재가 발생하였을 시 범용 화재경보기에서 출력되는 청각 신호를 인지하지 못해 제시간 내에 옥외로 대피하는 데 한계가 있다. 이러한 문제를 해결하기 위해 기존연구로 개발된 청각장애인용 화재 대피 경보기가 존재한다. 해당 경보기는 벽에 부착되어 섬광이나 진동을 통해 화재경보를 출력하는 방식이다. 하지만, 이러한 형태의 경보기는 청각장애인이 관찰 가능한 공간에 위치해 있지 않다면 기존의 범용 화재경보기의 문제점과 마찬가지로 청각장애인은 화재 상황을 전달받지 못하게 될 것이다. 그리하여 본 논문에서는 앞서 설명한 문제들을 해결하고자 비콘 송·수신기를 활용하여 청각장애인이 옥내의 어느 위치에 있어도 화재경보를 전달받아 제시간 내에 대피할 수 있도록 도와주는 화재 대피 시스템을 설계 및 구현하였다.

키워드 : 화재경보기, 화재대피시스템, 비콘, 임베디드 시스템, 청각장애인
Key Words : Fire Alarm, Fire Evacuation System, Beacon, Embedded System, Hard-of-Hearning People

ABSTRACT

In the event of a fire in the building, hard-of-hearing people are not aware of the signal output from the general-purpose fire alarm, making it difficult to evacuate in time. To solve these problems, existing fire escape alarms for the hard-of-hearing people exist. This alarm is attached to a wall and outputs a fire alarm through a flash or vibration. However, this type of alarm, like the problem with conventional general-purpose fire alarms, will not be able to communicate the fire situation unless the hard-of-hearing people are located in an observable space. Therefore, this paper designed and implemented a fire evacuation system that helps hard-of-hearing people to receive fire alarms and evacuate in time no matter where they are in the room by using beacon transmitter and receivers to solve the problems described earlier.

Ⅰ. 서 론

장애인이란 신체적, 정신적 장애로 오랫동안 일상생활이나 사회생활에서 상당한 제약을 받는 자(장애인복지법 제2조)를 말한다. 장애인은 근력이 비장애인에 비해 현저히 저하되었거나 신체적 장애로 인해 위

※ 본 연구는 정부(과학기술정보통신부)의 재원으로 한국연구재단(No.2021R1C1C1013133)의 지원을 받아 수행되었습니다.
※ 본 연구는 한국연구재단 4단계 두뇌한국21사업(4단계 BK21사업)의 지원을 받아 수행되었습니다 (No.5199990914048).
◆ First Author : Soonchunhyang University, Dept. of Software Convergence, wlgns12www@sch.ac.kr, 학생회원
° Corresponding Author : Soonchunhyang University, Dept. of Computer Software Engineering, dyoung.kim@sch.ac.kr, 정회원
* Soonchunhyang University, Dept. of Software Convergence, lsy8647@sch.ac.kr, 학생회원
** Kyungpook National University, Center of Self-Organizing Software, jungminwoo80@gmail.com, 학생회원
논문번호 : 202109-235-D-RN, Received September 9, 2021; Revised December 5, 2021; Accepted December 15, 2021

급상황의 대처 능력이 부족할 뿐만 아니라 자력 이동이 어려워, 건물 내 화재 발생 시 가장 큰 피해를 입을 수 있는 재해약자라 할 수 있다.[1] 재해약자인 국내 등록 장애인 수는 2019년 12월 말 기준으로 2,618,918명으로 총인구 대비 약 5.1%를 차지하고 있으며, 장애유형별로 청각장애인은 377천 명(약 14.4%)으로 지체장애인(46.7%) 다음으로 많은 장애유형이다.[2] 2016년 국가화재통계에 따르면 화재 사상자 중 사망자 비율에서 비장애인은 13.7%인데 반해 장애인은 43.6%로 3배 이상 높은 것으로 보도되고 있다.[3] 특히 청각장애인의 경우, 소리로 알 수 있는 수많은 정보를 놓쳐 화재 상황 발생 시 경보음을 듣지 못해 인지하지 못하는 위험에 빠지게 된다. 이들을 위해 기존에 개발된 청각장애인용 화재경보기가 존재하는데 이는 화재경보기에서 발하는 화재 신호를 청각장애인에게 점멸형태의 시각경보를 전달하여 청각장애인의 대피를 유도한다. 하지만, 이 경보기는 단순히 벽에 부착되어 LED 램프나 진동을 통해 화재경보를 송출함으로 해당 경보기를 관찰 가능한 거리에 청각장애인이 위치해 있지 않을 경우에 화재경보를 전달받을 수 없는 문제가 있어 신뢰성이 다소 떨어진다.[4,5]

따라서 본 논문에서는 패킷 송출 거리가 최대 50m인 비콘 송신기를 활용하여 넓은 지역의 청각장애인들에게 화재경보를 송출하는 시스템을 제안한다. 비콘 수신기를 이용하여 밴드나 목걸이 등 웨어러블 디바이스 형태로 제공되도록 설계하였으며, 수신받은 디바이스는 액츄에이터를 통해 청각장애인들에게 즉각적인 반응을 전달해 신속한 대피를 유도한다. 이를 통해 청각장애인이 경보기를 관찰하지 못하는 위치에 있어도 화재경보를 전달받아 더욱 안전하고 편리한 삶을 사는 데 도움을 주고자 한다.

본 논문의 구성은 다음과 같다. 2절에서는 기존연구와 더불어 백그라운드를 살펴보고, 3절에서는 본 논문에서 제안하는 시스템의 구성을 살펴본다. 4, 5절에서는 시스템 설계, 구현 및 테스트를 보이고 마지막으로 6절에서는 결론과 기대효과를 제시한다.

Ⅱ. 관련연구

2.1 관련연구

2.1.1 청각장애인용 화재경보기

화재 발생 시 즉각적으로 대처하기 어려운 청각장애인을 위한 기존의 청각장애인용 화재경보기는 대부분 시각화재경보기[그림 1] 형태로 옥내의 벽에 부착

그림 1. 청각장애인 화재경보기[6]
Fig. 1. Fire alarm for hard-of-hearing people

되어 램프(섬광)에 의한 화재 발생 신호를 송출한다. 시각화재경보기는 화재를 초기에 탐지하여 소방대상물의 관계자 및 거주자에게 안전한 장소로 피난을 통보하는 자동화재탐지설비 통보장치이다. 청각장애인이 접근 가능한 공공건물 및 공중이용시설의 공용거실, 복도, 통로, 로비, 휴게실(화장실) 및 기타일반거실(회의실, 강의실, 식당) 등에 설치하며, 자동화재탐지설비와 연동하고 동작하도록 하여 청각장애인을 안전하게 피난할 수 있도록 통보한다.[7] 하지만 이는 단지 벽에 부착되어 화재경보를 송출한다는 점이 문제가 된다. 화재 발생 사실을 눈으로 확인할 수밖에 없어 화재가 발생하였을 시 청각장애인이 해당 경보기를 관찰할 수 있는 위치에 있지 않은 이상 경보기를 통한 화재 상황을 전달받기 어렵기 때문이다. 화재 상황을 즉각적으로 전달받지 못하면 화재를 파악할 때까지의 시간이 걸릴 것이고 그로 인해 제시간 내에 대피가 어려워 인명피해를 초래하게 될 것이다. 이와 같이 기존에 개발된 청각장애인용 화재경보기는 실질적으로 신뢰성이 다소 떨어진다.[8]

2.2 백그라운드

2.2.1 비콘 (Beacon)

비콘(Beacon)은 블루투스 프로토콜(Bluetooth Protocol)기반의 근거리 무선통신 장치이다.[9] 비콘의 무선통신은 블루투스를 이용하여 두 기기 간에 연결을 위해 거쳐야만 했던 페어링의 과정이 없고 저전력, 소형화, 수명연장, 수신 거리 확대 등 많은 발전으로 최근 근거리 통신기술로 각광받고 있다.[10] 또한 최대 통신 거리가 약 50m로 상대적으로 긴 편이고 실내에서는 정교한 위치 파악이 가능하다.[11] 비콘을 사용하면 비콘이 부착된 특정 사물을 UUID값으로 구분하며, 저비용으로 스마트폰을 소지한 개인에게 RSSI(Received Signal Strength Indicator)를 이용하여 근접거리 구간별로 별도의 페어링 절차 없이 사용

자에게 신호를 전달 할 수 있다. 비콘 송신기는 주기적으로 자신의 UUID와 RSSI 값을 신호로 보내 스마트폰을 가진 사람이 이 신호의 도달 거리 내로 오면 스마트폰에서 이를 인식하여 서버로 신호 정보를 보낸다.[12]

2.2.1.1 iBeacon

iBeacon은 2013년 애플이 WWDC(World Wide Development Conference)에서 iOS 7과 함께 소개한 기술로서 Bluetooth Low Energy(BLE)를 이용한 기술이다. iBeacon은 송신 단말기를 이용하여 주기적으로 BLE 신호를 송신(Advertising)하고 이를 BLE를 탑재한 스마트 기기에서 수신(Scanning)하여 필요한 기능을 제공한다. iBeacon 전체 시스템은 단방향 통신으로 다음과 같다. iBeacon 디바이스가 메시지를 일정한 간격으로 브로드캐스팅(Broadcasting)하면 BLE를 수집할 수 있는 디바이스가 iBeacon 메시지를 브로드캐스팅 하는 범위에서 메시지를 수신한다.[13]

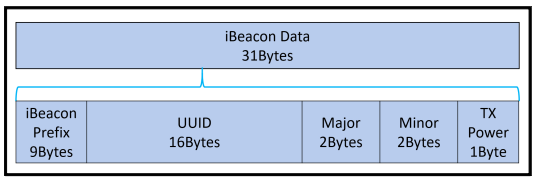


그림 2. iBeacon 패킷 규격
Fig. 2. iBeacon packet format

2.2.2 고속 푸리에 변환

푸리에 변환(Fourier Transform, FT)에서 모든 파동은 sin, cos의 합으로 나타낼 수 있으며, 이때 표현한 sin, cos의 주파수와 계수를 푸리에 변환이라고 한다. 임의의 입력 신호를 다양한 주파수를 갖는 주기함수들의 합으로 분해하여 표현하는 것이며 여기서 입력 신호는 전파, 음성, 신호 등과 같이 시간 축(time)에 대해 정의된 신호일 수도 있고 이미지(image) 등과 같이 공간 축에 대해 정의된 신호일 수도 있다. 푸리에 변환을 이용하면 소리, 전파와 같은 파동의 특성을 쉽게 파악할 수 있으며 시간 영역(Time domain)의 함수를 주파수 영역(Frequency domain) 함수로 변환할 수 있다.[15] 고속 푸리에 변환(Fast Fourier Transform, FFT)은 다양한 비 주기적인 신호의 해석에 이용이 되며 이산 푸리에 변환(Discrete Fourier Transform, DFT)은 $O(n^2)$의 시간복잡도를 가지는데 비해, FFT를 사용하면 O(nlogn)가 되므로 상대적으로 성능이 부족한 임베디드 환경에서도 빠른 속도를 가져갈 수 있게 해준다.[16] 본 논문에서는 위와 같은 기술에서 발전된 기술인 FFT를 적용하여 범용 화재경보기 알람 소리를 추출하였다.

Ⅲ. 제안하는 시스템

3.1 시스템 구조

본 논문에서는 [그림 3]과 같은 시스템을 제안한다. 먼저, [그림 3]의 오른쪽 상단에 위치한 화재 알람(Fire Alarm)과 화재 센서(Fire Sensor)는 화재와 관련된 데이터를 수집한다. 화재 알람은 청각장애인의 청각 신호를 대신해 범용 화재경보기가 작동 중인지 주기적으로 센싱 및 판단하여 알람 상태를 출력한다. 화재 센서는 가스와 온도를 주기적으로 센싱한 뒤 센싱 데이터를 출력한다. 출력된 2개의 데이터(알람 상태(alarm status), 센싱 데이터(sensing data))는 왼쪽 상단의 데이터베이스(DataBase)로 저장된다. 데이터베이스는 오로지 데이터 저장과 서버(Server)로부터 요청된 데이터를 전송해주는 역할을 한다.

다음으로 서버는 데이터베이스로부터 응답된 데이터인 알람 상태와 센싱 데이터를 얻어와 현재 옥내의 화재 위험도인 화재 위험(Fire Risk) 값을 주기적으로 판단한다. 그 후 서버는 화재 위험을 옥내에 존재하는 모든 비콘들에게 브로드캐스트로 전송한다. 비콘은 수신된 화재 위험 정보를 자신의 근방에 존재하는 클라이언트(Client)가 수신할 수 있도록 광고 패킷(Advertising Packet)에 화재 위험 정보를 부착하여 송출한다. 마지막으로, 클라이언트는 비콘이 송출한

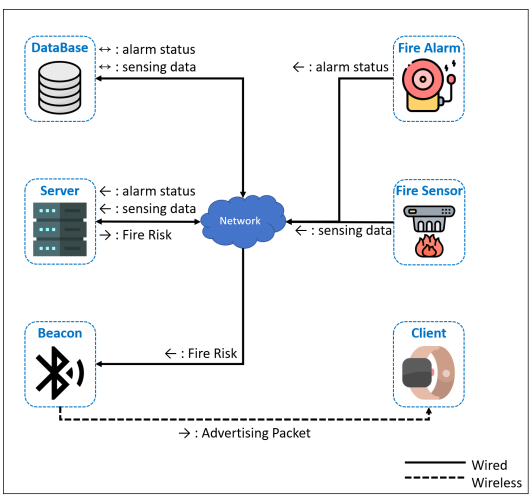


그림 3. 시스템 구조
Fig. 3. System architecture

광고 패킷을 수신하여 해당 패킷 내에 있는 화재 위험 정보를 추출 후 진동 및 LED 램프를 통해 사용자에게 현재 옥내의 화재 위험도를 알려준다.

3.2 비콘 송신기 데이터 포맷

[그림 3]의 시스템 구조 중 비콘이 송출하는 광고 패킷(Advertising Packet)의 규격은 [그림 4]와 같다. 해당 규격은 기존의 iBeacon규격[그림 2]의 16Bytes UUID공간을 변형한 형태이다. 기존에 16Bytes인 UUID의 Least Significant Bit(LSB)부터 왼쪽으로 8bits를 여유 공간으로 두어 본 시스템의 광고 패킷의 UUID는 [그림 4]와 같이 15Bytes의 공간을 갖는다. 나머지, 8bits 공간은 4bits씩 나누어 오른쪽 4bits는 패딩(Padding) 공간으로 '0000'으로 고정되고 나머지 왼쪽 4bits는 옥내의 화재 위험도인 화재 위험 정보를 담을 공간으로 사용된다. 즉, 본 시스템의 비콘은 서버로부터 수신된 화재 위험 정보를 자신의 광고 패킷의 8bits 여유 공간 중 왼쪽 4bits에 담게 되는 것이다.

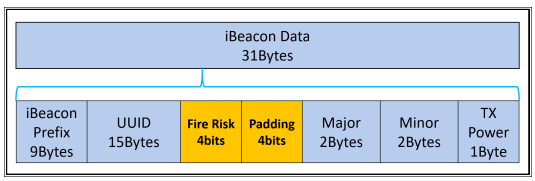


그림 4. 비콘 송신기 광고 패킷 규격
Fig. 4. Beacon Transmitter Advertising Packet format

3.3 시스템 동작 절차

본 절에서는 [그림 3]의 시스템 구조를 기반으로 전체적인 시스템 동작 절차를 설명한다. 화재 센서와 화재 알람이 주기적으로 데이터를 센싱하여 센싱된 값(알람 상태, 센싱 데이터)을 데이터베이스로 저장시킨다. 서버는 데이터베이스에 저장된 값을 바탕으로 옥내의 화재 위험도를 결정한다. 결정된 화재 위험도는 비콘으로 전달되어 자신이 주기적으로 송출시키는 광고 패킷에 부착한 뒤 주변에 존재하는 클라이언트로 전달한다. 클라이언트는 수신된 패킷 내의 화재 위험도를 추출한 뒤 안전(Safety), 위험(Danger), 대피(Evacuation) 중 각 위험도에 따라 진동이나 LED를 동작시켜 최종적으로 사용자에게 화재 위험도를 전달한다. 다음 4장에서는 본 논문에서 제안한 시스템 구조 [그림 3]의 각 모듈들을 자세히 설명한다.

Ⅳ. 시스템 설계 및 구현

4.1 데이터 수집

4.1.1 가스, 온도 데이터 수집

[그림 5]는 주기적으로 가스와 온도 값을 수집하는 화재 센서 모듈의 소프트웨어 구조이다. 가장 위의 온도 및 가스 센서에서 현재 온도와 가스 값인 센싱 데이터가 측정된다. 그 후 MySQL 클라이언트에게 해당 값이 넘겨져 쿼리문으로 변경된다. 마지막으로 쿼리문은 네트워크 모듈을 통해 데이터베이스로 전송된다.

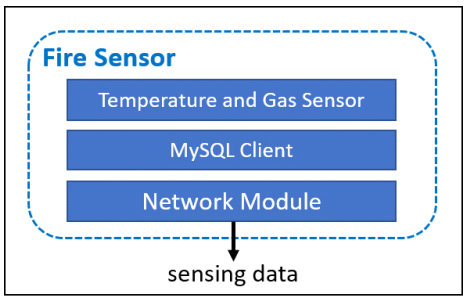


그림 5. 화재 센서 모듈 구조
Fig. 5. Fire Sensor module architecture

4.1.2 화재경보기 알람 여부

[그림 6]은 범용 화재경보기 알람 여부를 판단하는 화재 알람 모듈의 소프트웨어 구조이다. 가장 위의 사운드 센서는 현재 위치의 사운드 값을 측정한다. 측정된 사운드는 소켓(Socket)과 공유 메모리(Shared Memory)를 통해 FFT로 넘겨진다. FFT에서 시간 도메인의 사운드 값은 주파수 도메인으로 변경되어 범용 화재경보기가 울렸는지 판단된다. 판단된 값인 알

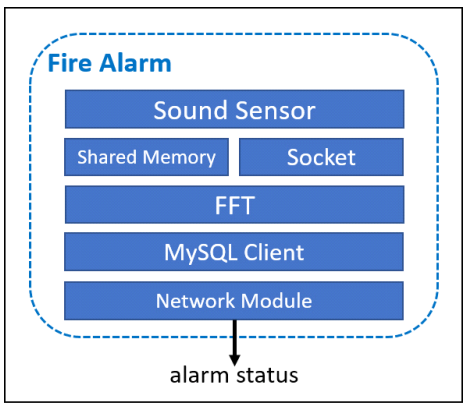


그림 6. 화재 알람 모듈 구조
Fig. 6. Fire Alarm module architecture

람 상태 정보는 MySQL 클라이언트로 넘어가 쿼리문으로 변경된다. 마지막으로 쿼리문은 네트워크 모듈을 통해 데이터베이스로 전송된다.

먼저, 화재 알람의 사운드 센서(Sound Sensor) [그림 7]는 TCP 소켓으로 localhost(127.0.0.1) 서버를 생성한 뒤 FFT의 연결요청을 기다린다. 만약 FFT와 연결이 완료되었다면 쓰레드를 하나 실행시킨다. 해당 쓰레드는 [그림 7]의 18번 라인 Sub 함수로 동작한다. 쓰레드가 처음 실행되었다면 공유 메모리를 초기화한 뒤 공유 메모리의 주소를 반환받아 shm_ptr에 저장시킨다. 그 후 다음의 동작을 반복한다. 먼저, 사운드 센서를 통해 시간 도메인 관점의 진폭 값들을 공유 메모리에 1차원 배열형태로 저장시킨다. 저장이 완료되었다면 공유 메모리의 키(key)를 TCP 세션을 통해 FFT로 전송시킨다. 그 후 쓰레드는 대기(wait) 상태가 되어 공유 메모리가 소모되기를 기다린다. FFT는 수신된 키를 활용해 공유 메모리에 접근하여 저장된 데이터들을 모두 끌어온 뒤 공유 메모리를 소모하였다는 의미로 “USED”라는 제어 메시지를 TCP 세션을 통해 사운드 센서로 전송시킨다. “USED”를 수신한 사운드 센서는 최신의 데이터를 공유 메모리에 공급하기 위하여 공유 메모리를 초기화시킨 뒤 해당 쓰레드로 신호(wake up)를 보낸다. 깨어난 스레드는 최신의 데이터를 공유 메모리에 공급하는 동작을 재수행한다.

FFT[그림 8]는 먼저, 데이터베이스와 연결을 수행하고 db_connector를 반환받는다. 그 후 사운드 센서

```
01: // Sound Sensor
02:
03: open a localhost TCP server
04: waiting for FFT connection request
05: IF connected THEN :
06:     thread_id <- thread_start(Sub)
07:     LOOP :
08:         IF received data from FFT THEN :
09:             message <- received data
10:             IF message = “USED” THEN :
11:                 initialize shared memory to null
12:                 thread_wakeup(thread_id)
13:             END IF
14:         END IF
15:     END LOOP
16: END IF
17:
18: FUNCTION Sub :
19:     shm_ptr <- initializing key and size of shared memory
20:     LOOP :
21:         sampling_sound(shm_ptr) using MAX4466
22:         shm_key <- key
23:         transmit_to_FFT(shm_key)
24:         thread_wait(thread_id)
25:     END LOOP :
26: END FUNCTION
```

그림 7. 화재 알람의 사운드 센서 의사코드
Fig. 7. Sound Sensor pseudo-code of Fire Alarm

```
01: // FFT
02:
03: db_connector <- connect to database
04: connecting to localhost TCP server
05: IF connected THEN :
06:     LOOP :
07:         waiting for a shm_key
08:         IF received shm_key from Sound Sensor THEN :
09:             buff <- get_shared_memory_data(shm_key)
10:             message <- “USED”
11:             transmit_to_Sound_Sensor(message)
12:             frequency <- fft(buff)
13:             alarm_status <- detect_alarm(frequency)
14:             send_data_to_db(db_connector, alarm_status)
15:         END IF
16:     END LOOP
17: END IF
```

그림 8. 화재 알람의 FFT 의사코드
Fig. 8. FFT pseudo-code of Fire Alarm

로 연결 요청하여 연결을 완료하고 다음의 동작을 반복한다. 사운드 센서가 공유 메모리에 데이터를 공급할 때까지 대기하다가 공유 메모리에 데이터 공급이 완료되었다면 사운드 센서는 FFT로 공유 메모리의 키를 전송할 것이다. 키를 수신한 FFT는 공유 메모리에 데이터 공급이 완료되었다고 판단하고 키를 활용해 공유 메모리에 접근한 뒤 모든 데이터를 끌어와 자신의 버퍼(buff)에 저장시킨다. 버퍼에 저장이 완료되었다면 공유 메모리를 소비하였다는 의미로 “USED”라는 메시지를 생성하여 사운드 센서로 전송해 최신의 데이터가 공유 메모리에 재공급될 수 있도록 한다. FFT는 공유 메모리에 데이터가 공급되는 동안에 버퍼에 저장된 데이터를 fft()에 입력하여 시간 도메인의 진폭 값을 주파수 도메인 진폭 값으로 변환해 주파수(frequency)에 저장시킨다. 그 후 주파수는 화재경보기 알람의 주파수 범위의 진폭 값이 임계값을 초과하였는지 판단하는 detect_alarm()에 입력되어 결과를 alarm_status에 반환시킨다. 그 후 alarm_status는 SQL문으로 구성되어 앞서 연결해둔 db_connector를 통해 데이터베이스로 전송된다.

화재 알람의 사운드 센서와 FFT는 각 각 공급 및 소비를 수행한다. 만약, FFT가 공유 메모리의 데이터를 끌어와 소비하였다면 사운드 센서는 곧바로 공유 메모리로 최신의 데이터를 재공급할 것이다. FFT는 공급이 수행되는 동안 끌어온 데이터를 활용해 화재

그림 9. 화재 알람 파이프 라인
Fig. 9. Pipeline of Fire Alarm

경보기 알람 여부를 판단할 것이다. 즉, 화재 알람의 사운드 센서와 FFT간에는 [그림 9]와 같이 2단계 파이프라인을 구성하여 동작을 수행한다.

4.1.3 데이터 저장

[그림 10]은 화재 센서와 화재 알람에서 출력되는 센싱 데이터와 알람 상태를 수신하여 저장하는 데이터베이스 모듈의 구조이다. 데이터베이스는 단순히 MySQL 서버에서 데이터 저장과 요청된 데이터를 전달해주는 역할만 수행한다.

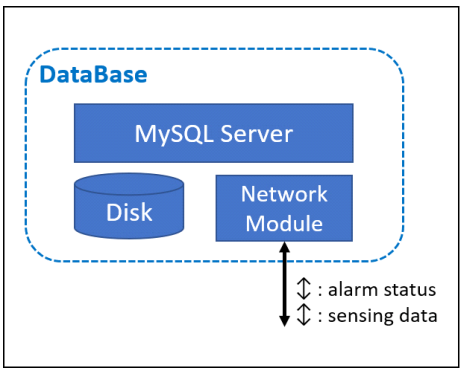


그림 10. 데이터베이스 모듈 구조
Fig. 10. DataBase module architecture

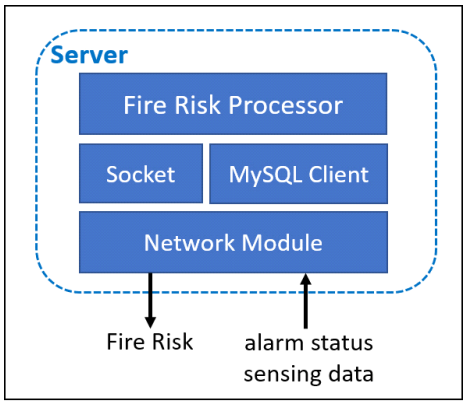


그림 11. 서버 모듈 구조
Fig. 11. Server module architecture

4.2 화재 위험도 판단 및 전송

[그림11]은 화재 위험도(Fire Risk)를 판단하고 비콘들에게 브로드캐스트하는 서버 모듈의 소프트웨어 구조이다. 서버는 네트워크 모듈과 MySQL 클라이언트를 통해서 데이터베이스에 저장된 알람 상태와 센싱 데이터를 받아온다. 수신된 두 데이터는 화재 위험 프로세서(Fire Risk Processor)로 입력되어 화재 위험도를 측정 후 화재 위험 정보를 출력한다. 출력된 정보는 브로드캐스트를 위한 UDP 소켓과 네트워크 모

```
01 : FireRisk ← Safety
02 : set up broadcast sender
03 : db_connector ← connect to database
04 : thread_start(Sub)
05 : Loop :
06 :   alarm_status_i ← get_data(fire_alarm, db_connector)
07 :   sensing_data_i ← get_data(fire_sensor, db_connector)
08 :   FireRisk ← process_FireRisk(alarm_status_i, sensing_data_i)
09 : End Loop
10 :
11 : Function Sub :
12 :   Loop :
13 :       count ← 0
14 :       IF Changed a FireRisk THEN :
15 :           Loop count < 3 :
16 :               broadcast(FireRisk)
17 :               count ← count + 1
18 :               wait(5 second)
19 :           End Loop
20 :       End IF
21 :   End Loop
22 : End Function
```

```
23 : Function process_FireRisk(alarm_status_i, sensing_data_i) :
24 :   FireRisk ← Safety
25 :   alarm_status ← 0
26 :   gas_status ← 0
27 :   fire_status ← 0
28 :
29 :   Loop i = 1 to alarm_status.length Do :
30 :       IF alarm_status_i = 1 THEN :
31 :           alarm_status ← 1
32 :       End IF
33 :   End Loop
34 :
35 :   Loop i = 1 to sensing_data.length Do :
36 :       IF sensing_data_i,0 > 3000 THEN :
37 :           gas_status ← 1
38 :       End IF
39 :       IF sensing_data_i,1 > 70.0 THEN :
40 :           fire_status ← 1
41 :       End IF
42 :   End Loop
43 :
44 :   IF (gas_status = 1 or fire_status = 1) and alarm_status = 1 THEN :
45 :       return Evacuation
46 :   ELSE IF (gas_status = 0 and fire_status = 0) and alarm_status = 1 THEN :
47 :       return Danger
48 :   ELSE IF (gas_status = 1 or fire_status = 1) and alarm_status = 0 THEN :
49 :       return Danger
50 :   ELSE IF (gas_status = 0 and fire_status = 0) and alarm_status = 0 THEN :
51 :       return Safety
52 :   End IF
53 : End Function
```

그림 12. 서버 의사코드
Fig. 12. Server pseudo-code

듈을 통해 옥내에 존재하는 모든 비콘으로 전달된다.

[그림12]에서 서버는 먼저, UDP 브로드캐스트를 위한 송신자(Sender)를 초기화하고 데이터베이스와 연결 후 db_connector를 반환받는다. 그 다음 쓰레드(sub thread)를 하나 실행시킨다. 해당 쓰레드는 단순히 화재 위험도가 변경될 때마다 5초 간격으로 3번씩 변경된 화재 위험도를 브로드캐스트하는 역할을 한다.

서버의 화재 위험도 판단은 process_FireRisk() 함수에서 실행된다. 해당 함수는 데이터베이스에서 끌어온 모든 데이터들을 입력으로 사용하며, 모든 데이터들을 하나씩 조회하는 방식으로 동작한다. 알람 데이터는 울렸는지 울리지 않았는지를 판단하고, 가스와 온도 데이터는 임계값을 초과하였는지를 판단한다. 가스와 온도 값이 정상범위이고 화재 알람이 울리지 않았다면 안전(Safety) 신호를 반환한다. 만약, 가스나 온도가 임계값을 초과하였고 화재 알람도 함께 울렸다면 대피(Evacuation) 신호를 반환한다. 또한, 가스나 온도의 임계값 초과나 화재 알람 중 단 하나에만 이상이 존재한다면 위험(Danger) 신호를 반환한다.

4.3 화재 위험도 수신 및 송출

[그림 13]은 서버에서 수신된 화재 위험도를 최종적으로 클라이언트에게 전달해주는 비콘 모듈의 소프트웨어 구조이다. 비콘 모듈은 서버로부터 전달된 화재 위험 정보를 네트워크 모듈과 소켓을 통해 화재 위험 수신자(Fire Risk Receiver)로 수신한다. 수신된 화재 위험 정보는 곧바로 패킷 변환기(Packet Changer)에게 전달되어 [그림 4]의 규격을 가진 광고 패킷의 예약된 공간(Fire Risk 4bit 공간)을 [그림 14]와 같이 화재 위험도에 맞는 bit로 변환시킨다. 안전은 '0001'bit, 위험은 '0010'bit, 그리고 대피는 '0100'bit로 변환된다. 그 후 패킷을 클라이언트가 수신할 수

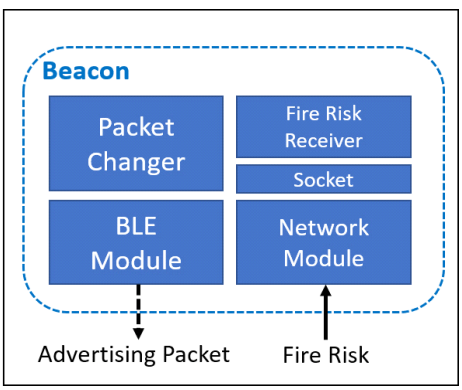


그림 13. 비콘 모듈 구조
Fig. 13. Beacon module architecture

bit	Fire Risk
0001	Safety
0010	Danger
0100	Evacuation

그림 14. 옥내 화재 위험도 테이블
Fig. 14. Fire Risk mapping table

있게끔 BLE 모듈을 통해 주기적으로 송출 반경 범위로 송출한다.

4.4 클라이언트

[그림 15]는 비콘으로부터 수신된 광고 패킷 내의 화재 위험 정보에 따라 액츄에이터를 동작시키는 클라이언트 모듈의 소프트웨어 구조이다. 클라이언트는 BLE 모듈을 통해 수신반경 내에 존재하는 광고 패킷을 끌어와 화재 위험 정보를 추출한다. 추출된 화재 위험 정보의 값에 따라 액츄에이터를 동작시켜 화재 위험도를 클라이언트 사용자에게 전달한다.

[그림 16]에서 클라이언트는 가장 먼저, 광고 패킷의 수신을 위한 수신기(receiver)를 초기화한다. 그 후

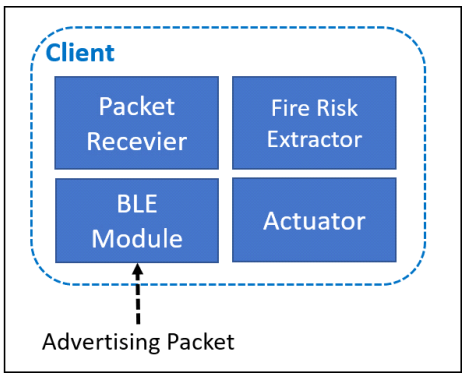


그림 15. 클라이언트 모듈 구조
Fig. 15. Client module architecture

```
01 : set up advertising_packet receiver
02 : Loop :
03 :   IF received advertising_packet from the beacon THEN :
04 :       fire_risk ← extract_fire_risk(advertising_packet)
05 :       IF fire_risk = safety THEN :
06 :           action(none, green light)
07 :       ELSE IF fire_risk = danger THEN :
08 :           action(weak vibration, orange light)
09 :       ELSE IF fire_risk = evacuation THEN :
10 :           action(strong vibration, red light)
11 :       End IF
12 :   End IF
13 : End Loop
```

그림 16. 클라이언트 의사코드
Fig. 16. Client pseudo-code

다음의 동작을 반복한다. 만약, 비콘으로부터 광고 패킷을 전달받았다면 패킷내에 존재하는 화재 위험 정보를 추출한다. 추출된 화재 위험 정보의 값(안전, 위험, 대피)에 따라 상이한 동작을 액츄에이터로 명령한다.

V. 테스트

5.1 테스트 환경

테스트 환경은 [그림17]과 같다. 화재 센서와 화재 알람은 BeagleBoneBlack(BBB) SoC를 사용하였으며 가스, 온도, 사운드 센서는 각각 MQ-9[17], MLX90614[18], MAX4466[19]를 사용하였다. 데이터베이스는 데스크탑 PC에 우분투(Ubuntu) 리눅스 서버를 올렸으며 RDBMS로는 MySQL을 사용하였다. 서버와 비콘은 RaspberryPi 3 B+ SoC를 사용하였으며 비콘의 BLE 역할은 RaspberryPi 3 B+에 기본적으로 탑재되어있는 BLE 모듈을 사용하였다. 클라이언트는 웨어러블 디바이스로 제작하기 위해 아두이노 나노(Arduino Nano) 초소형 MCU를 사용하였으며 비콘이 송출하는 광고 패킷을 수신하기 위해 HM10[20]을 사용하였다. 그리고 액츄에이터로는 진동과 RGB 모듈을 사용하였다.

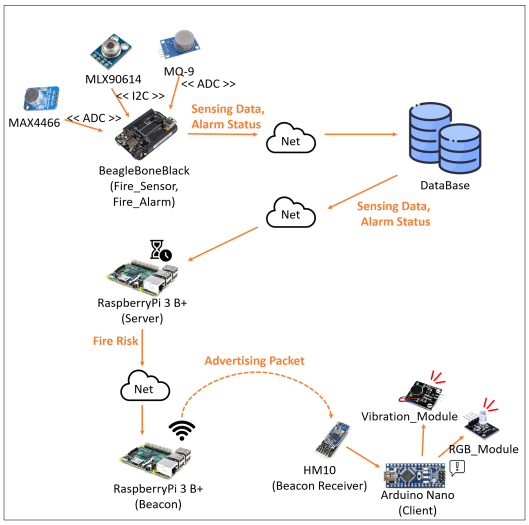


그림 17. 테스트 시스템 하드웨어 구조
Fig. 17. System hardware architecture for testing

5.2 시스템 테스트

5.2.1 Fire Sensor, Fire Alarm 데이터 수집

[그림18]은 BBB에 화재 센서와 화재 알람의 구현에 필요한 센서들을 연결한 핀맵 테이블이다. 화재 센서에 필요한 온도 센서는 I2C 통신으로 데이터를 전달함으로 BBB의 SCL, SDA 핀에 연결하였고 가스 센서는 ADC 핀을 통해 데이터를 전달함으로 A1 핀에 연결하였다. 화재 알람에 사용되는 마이크 센서 또한, ADC 핀을 통해 데이터를 전달함으로 A5 핀에 연결하였다.

BeagleBoneBlack PinMap	Fire_Sensor		Fire_Alarm
	MLX90614 (온도센서)	MQ9 (가스센서)	MAX4466 (사운드센서)
GND (P9_01)	GND	GND	GND
+3.3V (P9_04)			VCC
+5V (P9_07)	VCC	VCC	
I2C_SCL (P9_19)	SCL		
I2C_SDA (P9_20)	SDA		
A5 (P9_36)			OUT
A1 (P9_40)		AO	

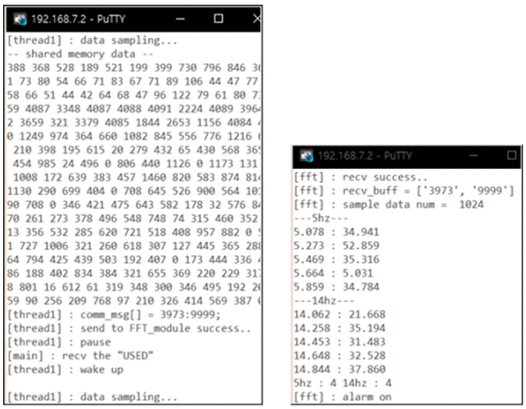
그림 18. 화재 센서, 화재 알람 핀맵 테이블[21]
Fig. 18. Pin Map table of Fire Sensor and Fire Alarm

화재 센서는 주기적으로 가스와 온도 값을 센싱한다. 센싱을 완료하면 [그림19]와 같이 센싱된 값을 출력 후에 데이터베이스로 전송시킨다. [그림19]에서 두 번째 센싱된 결과로 가스값(Gas Value)이 높게 상승한 것을 확인할 수 있는데 이는 테스트를 위해 직접 휴대용 라이터에서 발생 되는 가스를 주입하였기 때문이다.

사운드 센서[그림20]는 먼저, 1024개의 데이터를 수집하고 수집된 데이터를 FFT로 전달한다. FFT는 전달받은 데이터를 시간 도메인에서 주파수 도메인으로 변환 후에 범용 화재경보기의 소리에 해당하는 주파수 값들을 조회하여 주파수가 울렸는지를 판단한다. 그리고 해당 결과는 데이터베이스로 전송된다. 테스트를 위해 사용된 범용 화재경보기 사운드는 Google Play에서 제공하는 알람 및 사이렌 사운드 앱의 화재 알람 사운드를 활용하였다. 해당 앱에서 발생된 알람 사운드는 5hz와 14hz의 주파수 대역에서 높은 피크(Peak)가 발생되었다. 그래서 FFT 결과를 통해 5hz와 14hz 두 영역에서 높은 피크가 동시에 존재한다면 알람 설정(alarm on)이 출력되도록 [그림21]과 같이 구현하였다. 화재 센서와 화재 알람에서 생성된 데이터

```
debian@beaglebone:~/project/fire_alarm_system/module/fire_sensor$ ./main
[fire_sensor] : Sensing complete
Object  Temperature :   25.17°C
Ambient Temperature :   25.07°C
Gas     Value       :    1535
[fire_sensor] : Sensing complete
Object  Temperature :   30.03°C
Ambient Temperature :   25.07°C
Gas     Value       :    3860
```

그림 19. 화재 센서 실행 터미널
Fig. 19. Fire Sensor execution terminal



왼, 그림 20. 화재 알람의 사운드 센서 실행 터미널
오, 그림 21. 화재 알람의 FFT 실행 터미널
Left, Fig. 20. Sound Sensor exec terminal of Fire Alarm
Right, Fig. 21. FFT exec terminal of Fire Alarm

들은 모두 데이터베이스에 저장된다.

5.2.2 화재 위험도 판단 및 전송 테스트

서버는 주기적으로 데이터베이스의 모든 데이터들을 끌어와 화재 위험도를 판단한다. [그림22]에서 화재 센서의 데이터가 화재 알람보다 시간상 먼저 데이터베이스에 저장되었으므로 서버는 먼저 옥내의 가스나 온도에 이상이 있다 판단하여 화재 위험 정보를 위험(Danger)으로 설정한 뒤 브로드캐스트를 수행한다. 그 후 데이터베이스의 화재 알람 테이블이 갱신되면 서버는 옥내의 가스나 온도에 이상 그리고 화재경보기가 작동 중임을 인지하고 즉각적으로 화재 위험 정보를 대피(Evacuation)로 설정한 뒤 브로드캐스트를 수행하는 것을 [그림23]에서 확인할 수 있다.

[그림24]에서 비콘은 먼저, 광고 패킷의 화재 위험 정보 공간을 안전(Safety)으로 초기화한 뒤 UDP 브로드캐스트 패킷 수신을 위한 소켓을 바인딩한다. 그 후 서버로부터 패킷이 수신되기를 기다린다.

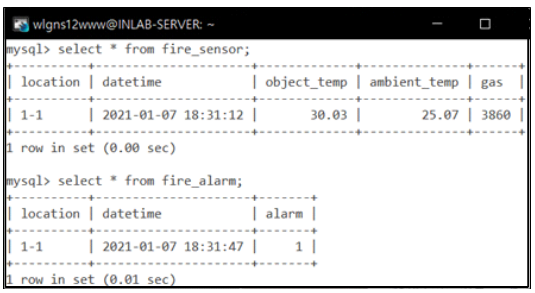


그림 22. 데이터베이스에 저장된 데이터
Fig. 22. Data stored in DataBase

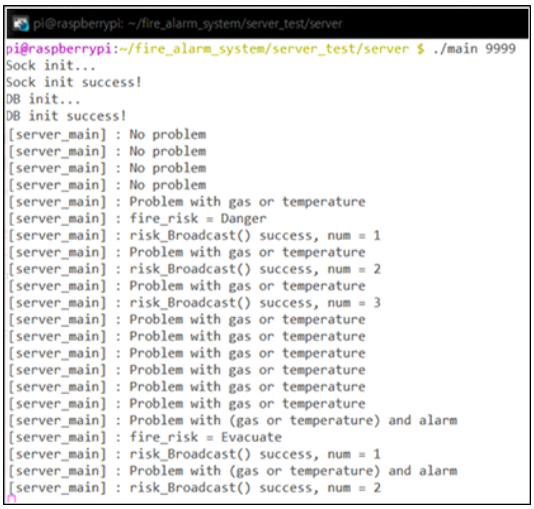


그림 23. 서버 실행 터미널
Fig. 23. Server execution terminal

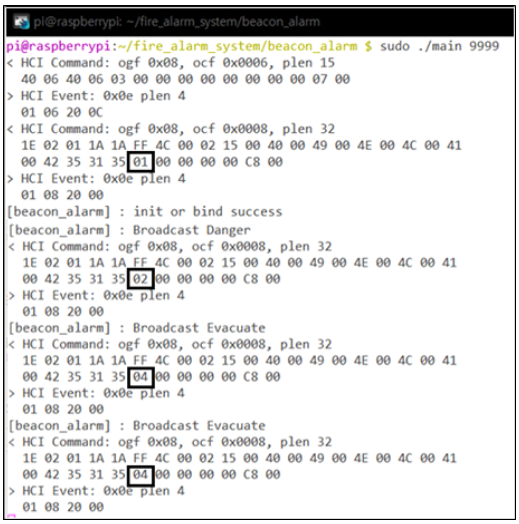


그림 24. 비콘 실행 터미널
Fig. 24. Beacon execution terminal

[그림23]에서 서버는 위험을 브로드캐스트한 후에 대피를 브로드캐스트 한다. 비콘 또한, 서버로부터 먼저 수신된 위험을 광고 패킷에 부착하고 그 이후 수신된 대피로 광고 패킷을 변경하는 것을 [그림24]에서 확인할 수 있다. [그림24]에서 네모박스 부분이 화재 위험 정보가 결합된 공간이다. 이렇게 화재 위험 정보가 결합된 광고 패킷은 주기적으로 클라이언트에게 송출된다.

5.2.3 클라이언트 테스트

클라이언트는 수신반경에 존재하는 광고 패킷을 수

신하여 해당 패킷의 UUID 공간의 마지막 1Byte를 추출해낸다. 앞의 [그림24]에서 비콘이 가장 먼저 송출한 과고 패킷의 화재 위험 정보 값은 '0001'bit인 안전이다. 클라이언트 또한, 안전을 먼저 읽는 것을 [그림25]에서 확인할 수 있다. 뒤이어 광고 패킷에 '0010'bit가 담겨와 클라이언트는 위험 상황임을 인지하였고 그다음 패킷에 '0100'bit가 담겨와 클라이언트는 대피 상황임을 인지하였다. 이렇게 인지된 화재 위험 정보에 따라 클라이언트의 액츄에이터가 동작할 것이고 각 동작하는 사진은 다음 [그림26]에서 확인할 수 있다. [그림26]의 왼쪽부터 안전 위험, 대피 신호에

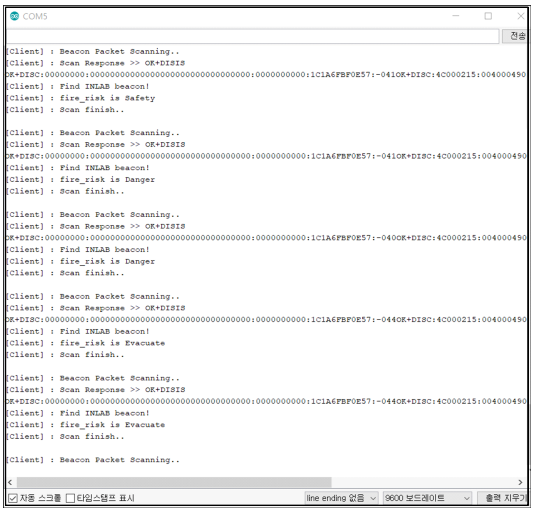


그림 25. 클라이언트 실행 터미널
Fig. 25. Client execution terminal

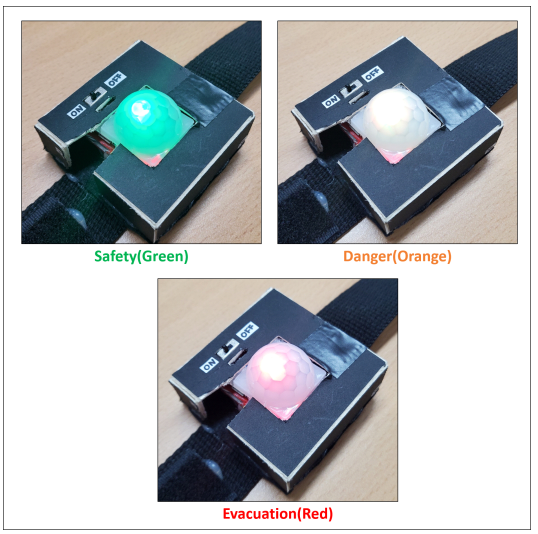


그림 26. 클라이언트 디바이스 구현 사진
Fig. 26. Implemented Client device

따른 클라이언트 디바이스의 액츄에이터 동작 결과 사진을 나타낸다. 클라이언트 디바이스는 사용자의 신체에 가장 가깝게 위치해야 하므로 손목시계 모양의 웨어러블 디바이스 형태로 제작하였다.

Ⅵ. 결 론

청각장애인은 범용 화재경보기에서 송출되는 청각 신호를 인지하지 못해 화재 상황을 비장애인보다 비교적 늦게 전달받는 현상이 발생할 수 있다. 이 문제에 대한 기존 연구로 청각장애인용 화재경보기인 시각화재경보기가 존재하는데 해당 경보기는 단지 벽에 부착되어 빛이나 진동을 통해 주변 인원들에게 송출하는 형태이다. 그러나 이러한 경보기는 직접 관찰할 수 있는 위치에 있지 않은 이상 화재 상황을 올바르게 전달받기 어려워 신뢰성이 다소 떨어진다는 단점이 존재한다.

그리하여 본 논문에서는 청각장애인이 화재경보기를 관찰하지 못하는 위치에 있어도 화재 상황을 즉각적으로 전달받아 제시간 내에 대피할 수 있도록 도와주는 "비콘을 활용한 청각장애인 화재 경보시스템"을 설계 및 구현하였다. 본 시스템은 화재 알람, 화재 센서, 데이터베이스, 서버, 클라이언트로 구성된다. 화재 센서와 화재 알람은 가스 · 온도 · 사운드 센서를 통해 주기적으로 화재와 관련된 데이터를 센싱하고 센싱된 값을 데이터베이스에 저장한다. 서버는 데이터베이스에 저장된 값을 바탕으로 옥내의 화재 위험도(Fire Risk)를 결정하고 비콘으로 전달한다. 비콘은 전달받은 화재 위험도를 자신이 송출하는 패킷에 부착시켜 주변에 존재하는 클라이언트로 송출한다. 클라이언트는 수신된 패킷 내의 화재 위험도를 추출한 뒤 3가지의 위험도 중 각 위험도에 따라 액츄에이터의 진동 세기나 LED를 변경하여 최종적으로 사용자에게 화재 위험도를 전달한다. 또한, 클라이언트는 신속한 정보 전달을 위해 User의 신체 가까이 위치해야 하므로 아두이노 나도를 사용하여 손목시계 모양의 웨어러블 디바이스 형태로 제작하였다. 해당 시스템을 통해 기존 화재경보기보다 신뢰성 있는 시스템을 제공하여 화재 상황 시 옥내에 존재하는 청각장애인들의 신속한 대피를 유도하여 사회적 약자의 안전을 한 층 더 높일 수 있을 것이라 기대한다.

References

[1] J. S. Lee, H. S. Kwon, and E. S. Kim, "A

study on the fire drill behavior characteristics in rehabilitation center for visually impaired persons," *J. Korea Academia-Ind. Cooperation Soc.*, vol. 16, no. 8, pp. 5646-5653, 2015, https://doi.org/10.5762/KAIS.2015.16.8.5646

[2] H. J. Kim, *2020 Disability Statistical Yearbook*, pp. 27-28, Korea Disabled people's Development Institue, 2020, https://www.koddi.or.kr/

[3] J. H. Kim, *2019 The Disabled Wite Book*, pp. 454-455, Korea Disabled people's Development Institute, 2019, https://www.koddi.or.kr/

[4] J. H. Kim, *2017 The Disabled Wite Book*, pp. 506-507, Korea Disabled people's Development Institute, 2017, https://www.koddi.or.kr/

[5] S.-U. Jang, "A study on establishment standar improvement of strobe light," Unpublished M.S. Thesis, Yeungnam Univ., Gyeongsan, 2011.

[6] Y.-H. Lee, "*Daejeon City Corpration, installation of fire alarms for the hearing impaired in Nuribodm Apartment*," Newstnt, Jun. 29, 2019, Retrieved from http://www.newstnt.com/news/articleView.html?idxno=33729

[7] S. T. Park, B. Y. Lee, C. H. Park, W. J. Yang, and S. H. Hong, "A study on the visible fire alarm appliance of low power consumption," *Fire Insures Laboratories of Korea*, vol. 2005, pp. 106-111, 2005.

[8] J. S. Son and E. S. Yi, "A study on warning pictogram system for the hearing-impaired," *Archives of Design Res.*, vol. 31, no. 3, pp. 151-163, 2018, https://doi.org/10.15187/adr.2018.08.31.3.151

[9] S. B. Son, S. H. Jeon, and Y. J. Choi, "Smart tolling system using beacon," in *Proc. KIISE Conf.*, vol. 2018, no. 12, pp. 2288-2290, 2018.

[10] D. Y. Kim, U. J. Hyeon, and S. J. Yoon, "Customized marketing design strategies using beacon," in *Proc. The Korean Soc. Comput. and Inf. Conf.*, vol. 27, no. 2, pp. 149-150, 2019.

[11] H. N. Go, S. B. Lee, S. W. Lee, J. H. Kwon, and E. J. Kim, "School bus getting on/off recognition system using BLE beacon," in *Proc. KIISE Conf.*, vol. 2019, no. 12, pp. 1332-1333, 2019.

[12] K. M. Kim, "Design of school commuting system using beacon," *J. KIICE*, vol. 20, no. 10, pp. 1941-1948, 2016, https://doi.org/10.6109/jkiice.2016.20.10.1941

[13] J. Y. Lee, C. S. Nam, and D. R. SHin, "Comparing distance measurements on iOS and Android using iBeacon to measure istance from pedestrian to vehicle," in *Proc. KIISE Conf.*, vol. 2017, no. 12, pp. 493-495, 2017.

[14] A. Mackey, P. Spachos, L. Song, and K. N. Plataniotis, "Improving BLE beacon proximity estimation accuracy through bayesian filtering," *IEEE Internet of Things J.*, vol. 7, no. 4, pp. 3160-3169, 2020, http://dx.doi.org/10.1109/JIOT.2020.2965583

[15] Y. S. Ahn, Y. J. Lee, E. J. Oh, and B. S. Kim, "Building energy demand prediction model using fourier transform based schedule analysis algorithm," *Korean J. Air-Conditioning and Refrigeration Eng.*, vol. 32, no. 8, pp. 386-397, 2020, http://dx.doi.org/10.6110/KJACR.2020.32.8.386

[16] B. J. Park and H. Jin, "A study on the method for analyzing wind power outputs through fast fourier transform," in Proc. *KIEE Conf.*, vol. 2015, no. 7, pp. 87-88, 2015.

[17] A. S. Falohun, A. O. Oke, B. M. Abolaji, and O. E. Oladejo, "Dangerous gas detection using an integrated circuit and MQ-9," *Int. J. Comput. Appl.*, vol. 135, no. 7, pp. 30-34, 2016, https://doi.org/10.5120/ijca2016908473

[18] T. U. Urbach and W. Wildian, "Rancang bangun sistem monitoring dan kontrol temperatur pemanasan zat cair menggunakan sensor inframerah MLX90614," *J. Fisika Unand*, vol. 8, no. 3, pp. 273-280, 2019, https://doi.org/10.25077/jfu.8.3.273-280.2019

[19] L. A. S. Lapono and R. K. Pingak, "Design of sound level meter using sound sensor based on arduino uno," *J. ILMU DASAR*, vol. 19, no. 2, pp. 111-116, 2018, https://doi.org/10.19184/jid.v19i2.7268

[20] I. Naghi S. Akbar, and B. Prasetio,

"Implementasi sistem pervasive pada smart home berbasis bluetooth versi 4.0 menggunakan modul BLE HM-10 dan sensor," *J. Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 1, no. 9, pp. 940-949, 2017, http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/285

[21] *BeagleBone Black Pin Map*, 2021, from MathWorks, https://www.mathworks.com/help/supportpkg/beagleboneio/ug/beaglebone-black-pin-map.html

박 지 훈 (Jihoon Park)

2021년 2월 : 대구가톨릭대학교 컴퓨터공학전공 학사

2021년 3월~현재 : 순천향대학교 일반대학원 소프트웨어융합학과 석사과정

<관심분야> MPTCP, 네트워크 시스템

이 소 연 (SoYeon Lee)

2021년 2월 : 대구가톨릭대학교 모바일소프트웨어전공 학사

2021년 3월~현재 : 순천향대학교 일반대학원 소프트웨어융합학과 석사과정

<관심분야> 딥러닝, 네트워크 시스템

정 민 우 (Minwoo Jung)

2015년 2월 : 경북대학교 대학원 전자공학 박사

2015년~2018년 : 경북IT융합산업기술원 선임연구원

2018년~2021년 : (주)카네비컴 책임연구원

2021년~현재 : 경북대학교 자율군집소프트웨어센터 연구교수

<관심분야> 임베디드 소프트웨어, 사물인터넷, 차량용 센서, LiDAR 시스템

김 대 영 (Dae-Young Kim)

2010년 8월 : 경희대학교 대학원 컴퓨터공학과 박사

2010년~2013년 : LIG넥스원 통신연구센터 선임연구원

2013년~2015년 : ㈜에어플러그 선임연구원

2015년~2017년 8월 : 창신대학교 컴퓨터소프트웨어공학과 조교수

2017년 9월~2021년 2월 : 대구가톨릭대학교 컴퓨터소프트웨어학부 조교수

2021년 3월~현재 : 순천향대학교 컴퓨터소프트웨어공학과 조교수

<관심분야> 모바일 네트워킹 및 컴퓨팅, 머신러닝, 네트워크 시스템

[ORCID:0000-0003-4901-3075]