

# 화재 예방 시스템

노대연\*, 홍지현, 김주아, 조우리

\*선문대학교 컴퓨터공학과

e-mail:nararol@naver.com

## System of Fire Prevention

Dae-Yeon Noh, Ji-Heon Hong, Joo-Ah Kim, Woo-Ri Jo

\*Dept of Computer Science and Engineering, Sun Moon University

### 요 약

화재 정보 시스템은 안드로이드, MEAN 스택을 이용한 서버와 데이터베이스, 아두이노를 이용한 장치를 통해 사용자에게 온도, 가스농도 정보와 화재 발생 유무를 알려준다. 화재 시 빠르게 대처, 대피할 수 있고, 또한 경보를 울리도록 만들었다. 또 애플리케이션을 통해 데이터베이스 서버에 적재된 최신의 데이터 값들을 받아와 표와 그래프로 보여준다.

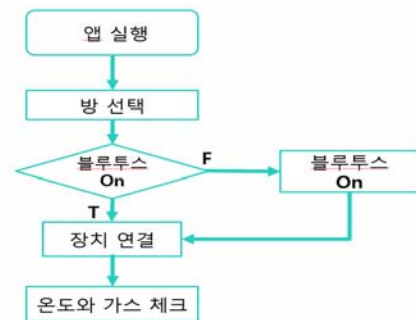
### 1. 서론

소방안전 분야 대부분의 연구는 ‘예방’ 보다는 재난이 일어난 후의 ‘대비’에 관련된 분야이다. 따라서, 화재의 초기 발생 시각과 위치를 알기 힘들다. 따라서 본 논문에서는 모든 지점에 화재 예방 장치가 설치되어 있다고 가정, 화재의 초기 발생 위치와 시각을 알 수 있도록 하고자 한다. 또한 앱을 설치한 사람들 누구든 주변에 있는 지점의 온도와 가스 농도를 실시간으로 확인하고 위험한 경우 알람을 울리게 하여 대피하고 소방서에 연락할 수 있도록 하고자 한다.

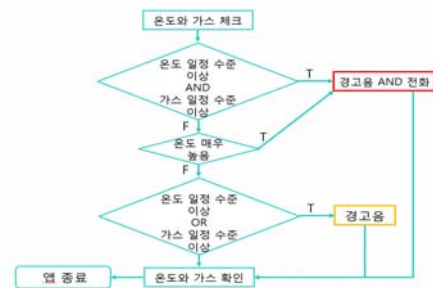
### 2. 화재 예방을 위한 알고리즘과 해결방법

본 논문은 앞서 언급했던 문제점들을 해결하기 위하여 안드로이드 애플리케이션, MEAN 스택을 이용한 서버와 데이터베이스, 아두이노를 이용한 하드웨어 장치를 설계했다. 화재 예방을 위한 시스템은 다음과 같이 구성되어 있다: 아두이노 장치는 블루투스 센서, 가스 센서, 온도 센서, 소리 센서로 이루어져 있다. 가스 센서, 온도 센서로 데이터를 측정하고, 블루투스 센서를 이용하여 통신하고, 소리 센서로 위험을 알린다. 서버측은 두 부분으로 나눌 수 있는데 Node.js를 이용한 서버와 MongoDB를 이용한 데이터베이스 서버이다. 애플리케이션이 Node.js 서버에 연결되면 MongoDB 데이터베이스 서버로 데이터를 보내거나 받아와 현재 데이터를 실시간으로 보거나 저장할 수 있도록 설계했다.

화재 예방용 안드로이드 애플리케이션의 동작방식은 다음과 같다. 화재 예방을 위한 알고리즘은 그림 5, 그림 6과 같이 블루투스 연결부와 애플리케이션 실행부로 되어



(그림 1) 화재 예방을 위한 알고리즘 연결부



(그림 2) 화재 예방을 위한 알고리즘 실행부

있다. 블루투스 연결부에서는 원하는 방을 선택한 후 그 방에 설치된 화재 예방 장치와 연결하는 과정을 거친다. 그 이후에 애플리케이션 실행부로 넘어가는데 애플리케이션 실행부에서는 온도와 가스 농도를 실시간으로 체크하여 온도와 가스농도가 둘 다 일정 수준 이상일 경우에는 위험으로 판단하여 경고음과 소방서로 전화를 걸게끔 한다. 또한 온도와 가스농도가 둘 다 일정 수준 이상이 아니더라도 온도가 매우 높을 경우에는 위험으로 판단한다. 온도와 가스농도 둘 중 하나의 수치가 일정 수준 이상일 경

우 이를 경고(위험의 가능성이 있음)로 판단하고 경고음을 울린다. 만약 온도와 가스농도의 수치가 전부 일정 수준 이상이 아닐 경우 이를 안전으로 판단한다. 그림 4 처럼 위험의 경우 애플리케이션 자체에서 빨간색 그림으로 표시되며, 아두이노 장치에서도 경고음을 울려서 장치가 있는 방에 있는 사람들에게도 위험을 알린다. 경고의 경우 자료를 보고 있는 사람들에게만 위험의 가능성이 있음을 알려도 괜찮겠다고 판단하여 애플리케이션에서만 경고음이 울리고 노란색 그림으로 표시된다. 그림 5처럼 안전의 경우 아무런 경고음이 울리지 않고 초록색 그림으로 표시된다. 이는 신호등의 빨간색, 노란색, 초록색과 같은 색의 개념이어서 처음 보는 사람들도 쉽게 판단할 수 있을 것이라 생각된다.



(그림 3) 위험

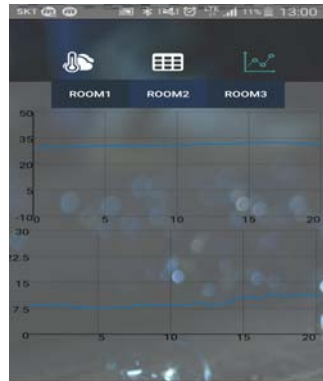


(그림 4) 안전

방 안의 온도나 가스농도의 수치 중 과거의 데이터(지금까지의 데이터)를 보고 싶을 경우나 통계를 보고 싶을 경우를 대비하여 표와 그래프를 통해 볼 수 있도록 만들었다. 표와 그래프는 서버에게 요청할 경우 데이터베이스 서버를 통해 데이터가 사용자에게 넘어오게 된다. 그림 5 처럼 표를 클릭할 경우 날짜, 시간, 온도, 가스농도 순으로 데이터가 보이게 된다. 100개의 데이터가 보이며 최신의 데이터일 수록 위쪽에 보이게 만들었다. 그림 6처럼 그래프의 경우도 마찬가지로 최신의 데이터 100개를 보이게 만들었다. 이 그래프는 드래그가 가능하고, 데이터를 한 눈에 보기 편하게 X축 이동이 가능하게 설계되었다.

날짜	시간	온도	가스
160615	125933	29.34	8.21
160615	125932	29.80	8.41
160615	125931	29.62	8.21
160615	125930	29.80	8.21
160615	125929	30.35	7.53
160615	125928	30.54	7.72
160615	125927	30.17	7.62
160615	125926	30.26	7.92
160615	125925	30.45	8.41
160615	125924	30.63	8.41
160615	125923	30.82	8.31
160615	125922	31.00	8.60

(그림 5) 최신 데이터 표



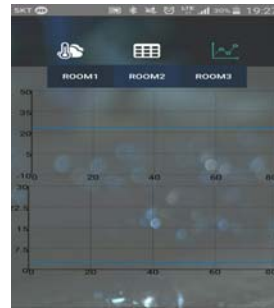
(그림 6) 최신 데이터 그래프

### 3. 실험 및 분석

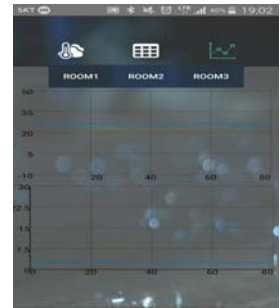
본 논문에서 구현한 화재 예방 시스템에 대한 실험 및

분석을 해 보기 위하여, 실내·외의 온도와 가스 농도 수치를 측정해 보았다. 실험 상황은 동일하게 공중에서 100초 동안 측정해 보았다. 공중에서 실험한 이유는 바닥이나 책상 위에는 가스가 모여 있거나 바닥 자체의 온도 등으로 인해 온도와 가스 농도 수치가 정확하게 측정되지 않기 때문이다. 그래서 공기 중의 온도와 가스 농도만을 정확하게 측정하기 위하여 공중의 데이터를 측정했다.

실험 장소로는 총 다섯 지점으로 에어컨이 가동된 연구실과 그렇지 않은 연구실, 실내 계단, 실내 중앙 복도 그리고 실외에서 각각 측정해 보았다.



(그림 7) 연구실(에어컨)



(그림 8) 중앙 복도

그림 7과 8의 결과처럼 측정 결과 실내외 전부 그래프 모양(데이터 값)에 큰 변화가 없었음을 알 수 있다. 온도는 실내외 전부 비슷한 값을 가졌는데(에어컨 틀은 경우 제외) 가스 농도 수치의 경우 같은 건물 안이어도 위치에 따라서 조금씩 다른 모습을 확인할 수 있었다. 하지만 두 데이터 모두 일정 수치 이상을 넘지는 않았다. 따라서 일정 수치 이상일 경우를 화재로 가정하고 테스트를 진행했다.

#### 1. 애플리케이션 설치한 경우 (다른 지점)

애플리케이션을 설치한 경우 현재 A 지점에 있다고 가정하면, B 지점의 온도와 가스농도 수치를 실시간으로 받아와 위험에 대비할 수 있었다. 위험의 경우 그림 3 처럼 위험 모양의 그림이 나타나고 일정 시간 이후 119로 전화를 걸 수 있게 해 주었다.

#### 2. 애플리케이션 설치한 경우 (같은 지점)

애플리케이션을 설치한 경우 측정자가 현재 A 지점에 있을 때, A 지점의 온도와 가스농도 수치를 실시간으로 받아온다고 하자. 이 경우에는 실험 1과 동일했다.

#### 3. 애플리케이션 설치하지 않은 경우 (같은 지점)

애플리케이션을 설치하지 않은 경우라도 위험일 경우에는 화재에 대한 예방을 할 수 있었다. 위험(온도와 가스 농도 수치가 일정 수준 이상)의 경우에는 화재 예방 장치(아두이노 장치)에서도 큰 소리를 내어 위험을 알려주기 때문에 위험을 알려줄 수 있었다.

### 참고문헌

[1] 통계청 지표체계 - 지표 상세 - 화재발생현황 2014  
 [2] M. Choi, "A experimental study on responsiveness characteristics of fire detectors and sprinkler heads", 2006