

# Keeper App

종합설계과제 및 운용 23-1학기(최종발표)



지도 교수 : 최권휴 교수님

21812207 한 종 헌

21812227 황 종 원

# CONTENTS



- 01 목적 및 필요성
- 02 해결 사항
- 03 성능 목표
- 04 지적 사항 및 반영
- 05 최종 결과물
- 06 진행 상세 결과
- 07 목표 대비 진행상황
- 08 팀원 별 역할
- 09 결과물 성능 분석

### ✓ 목적

- 유독 가스의 경우, 무색 무취의 특성 -> 장시간 노출 시 위험도 증가
- 센서 이용 -> 가스 농도 파악 및 대처

### ✓ 필요성

- 기존 사례 : 농도 검출 후 경보를 울리는 경우만 존재
- 다양한 사람들이 기능의 혜택을 얻을 수 있는 제품이 필요

## 02 해결해야 하는 문제

---

1. 멀티 스레드 기반 TCP 소켓 통신을 구현하여 객실 / 119 / 어플 간 실시간 센서 정보를 받고 위급상황을 알린다.
2. 가스 농도의 기준 값과 시간에 따라 메시지 전송과 각 센서들이 동작한다.
3. 어플을 통해 객실을 제어하고 상황에 맞게 부저, LED 등이 작동한다.

### 03 성능 목표

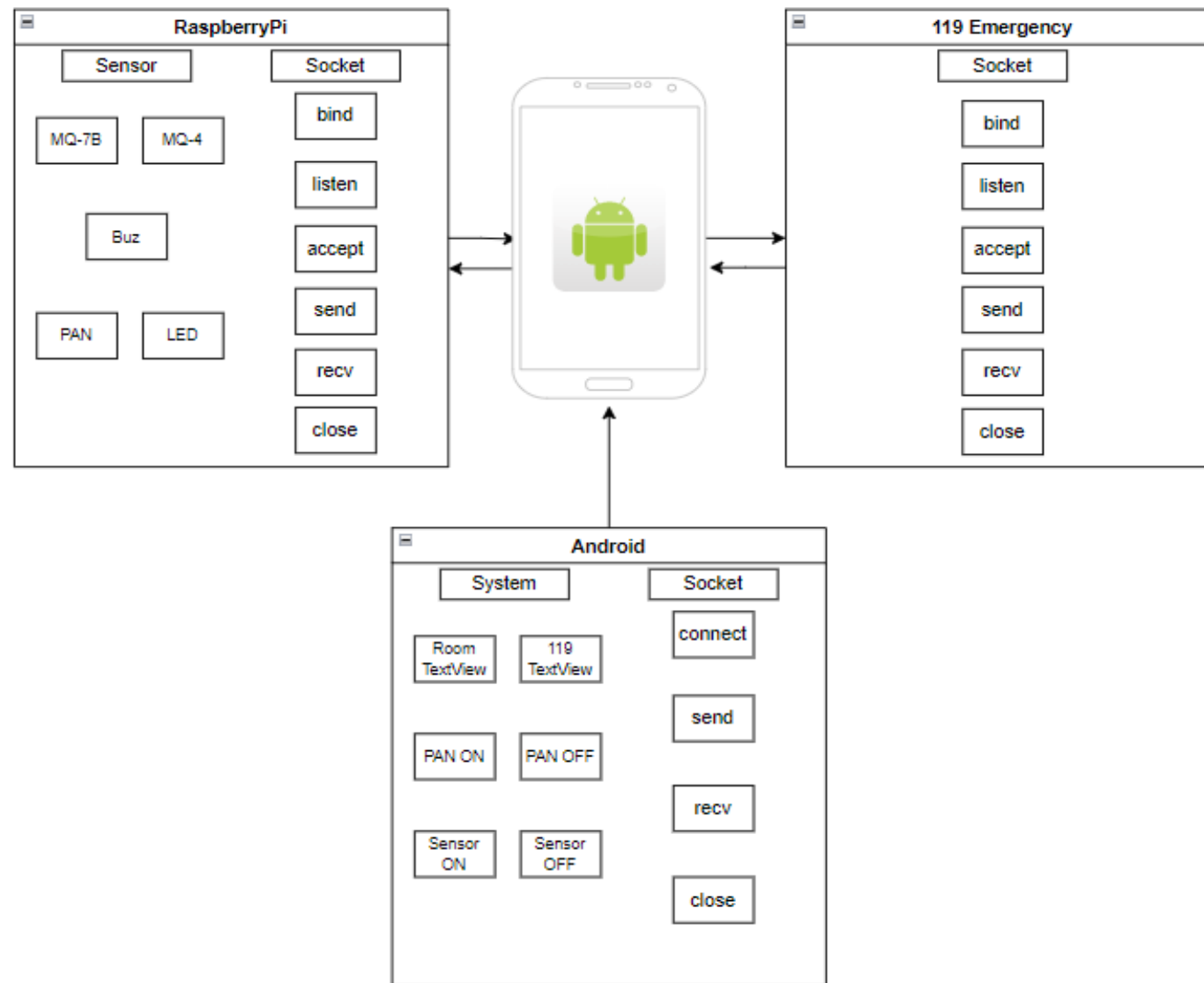
핵심 필수 기능	당초 성능 목표	고려사항
Client-Server / 양방향 Full-Duplex TCP 통신	50번 전송 시, 최소한 49번 성공 지연율 1초 이내	<ul style="list-style-type: none"><li>- MultiThread TCP 소켓 통신 이용</li><li>- Client-Server를 통해 2개의 양방향 Full-Duplex 통신으로 정보 전달</li></ul>
MQ-7B, MQ-4 센서 유독가스 검출	최소 2초 이내 감지	<ul style="list-style-type: none"><li>- 센서에서 측정된 아날로그 값을 ADC 모듈을 Digital 값으로 변환, 출력</li></ul>
LED, 부저 등 상황 전달	최소 2초 이내 작동	<ul style="list-style-type: none"><li>- 유독 가스 수치에 따라 LED를 다르게 출력</li><li>- 기준 값을 넘으면, 부저 작동</li></ul>
App -> RaspberryPi 사용자 경험(UX) 추가	최소 2초이내 작동 지연율 1초 이내	<ul style="list-style-type: none"><li>- Raspberry Pi에서 시스템 기능 전달 시, 메시지가 섞이지 않도록 MultiThread 구현</li></ul>

## 04 지적 사항 및 반영

지적 사항	기존	반영 및 변경
앱 구현 등 다양한 기능 추가	가스 농도 값에 따른 메시지 전송 및 반응 출력	- 앱 구현 : 어플을 활용한 센서 제어 기능, 통신이 가능하도록 추가
긍정적인 주제 및 제목 설정	주제 및 제목 : 숙박시설 내 자살 사고 예방	- 제목 : Keeper App - 가스 탐지 및 신고 기능을 갖춘 건물 안전 시스템으로 변경
구현 사항 추가 및 기능 구체화	가스 농도에 따른 메시지 전송 및 반응 출력	- 상황 개선 기능 추가 : 환풍 시스템 - 라즈베리파이 코드 개선 - -> 메시지 송수신, 센서 제어를 병렬 적으로 수행 가능한 스레드 구현

## 05 최종 결과물

### 기능 블록도

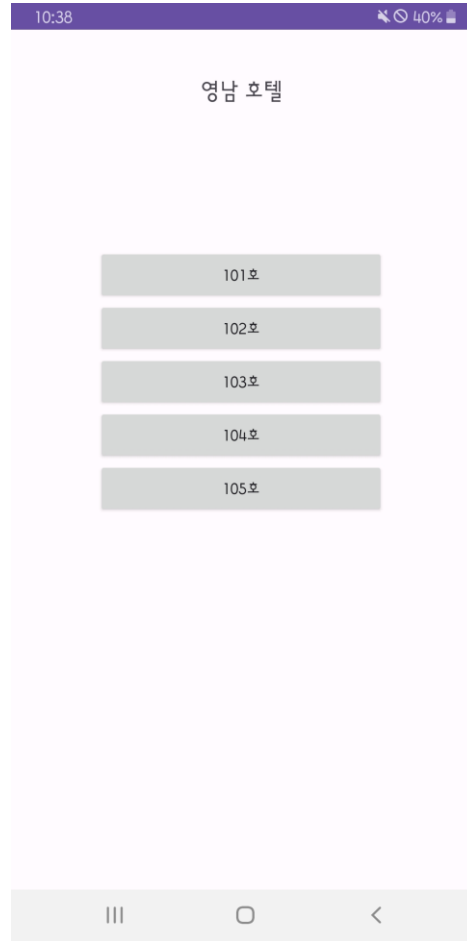


## 05 최종 결과물

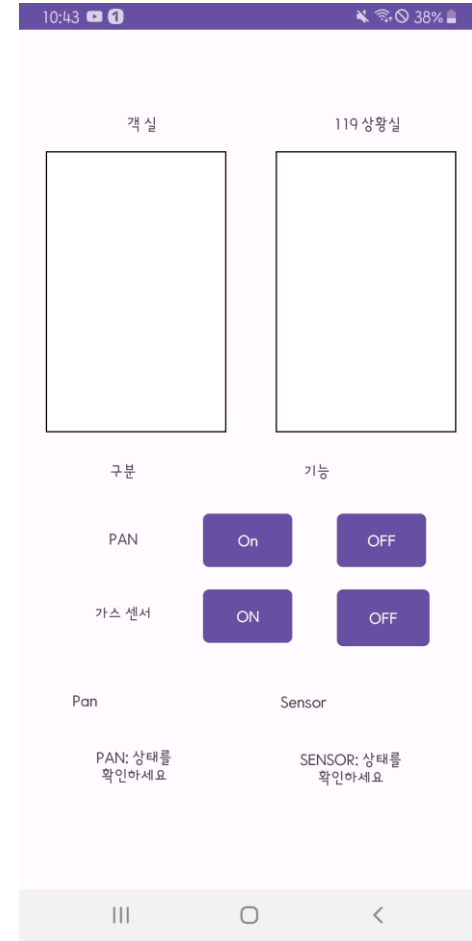
### 결과물 – 안드로이드 App



홈 화면



객실 선택 화면

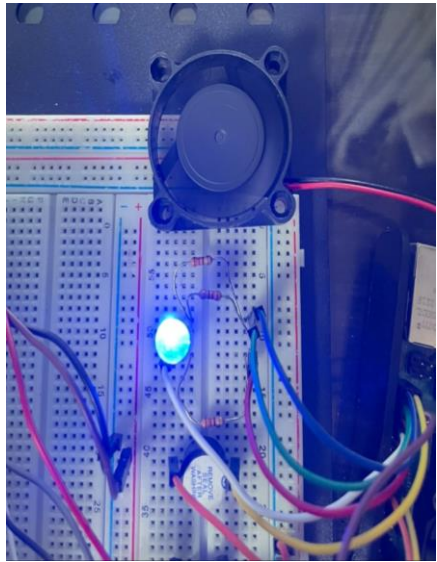


메시지 수신 및  
센서 제어 화면



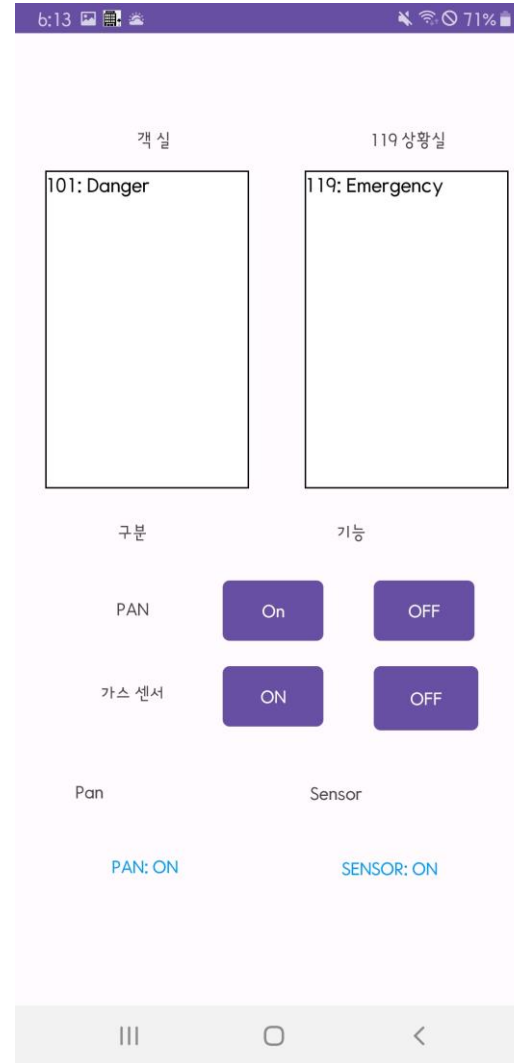
## 05 최종 결과물

### 결과물 – 팬 ON

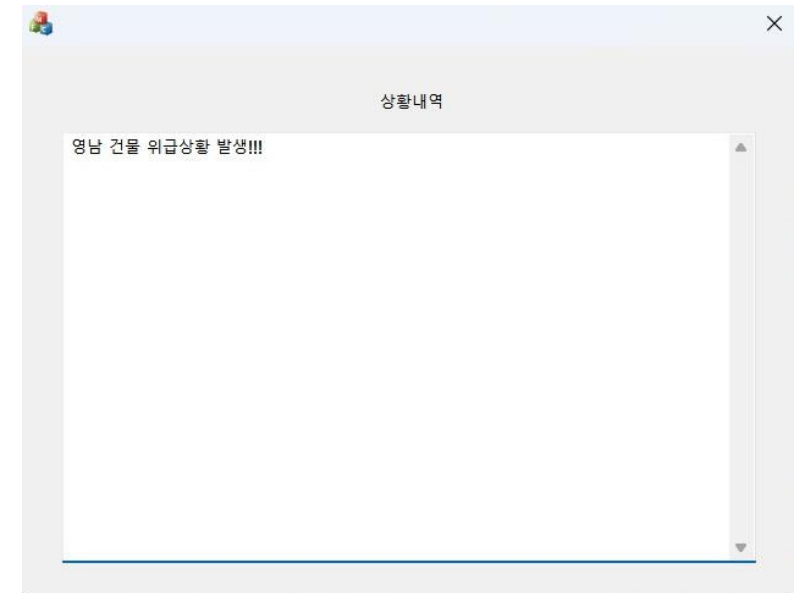


```
BLUE LED ON : Safe
BLUE LED ON : Safe
BLUE LED ON : Safe
RED LED ON : Dangerous!!
RED LED ON : Dangerous!!
BLUE LED ON : Safe
Message received: PAN_OFF
BLUE LED ON : Safe
BLUE LED ON : Safe
RED LED ON : Dangerous!!
Message received: PAN_OFF
BLUE LED ON : Safe
RED LED ON : Dangerous!!
BLUE LED ON : Safe
BLUE LED ON : Safe
Message received: PAN_ON
```

RaspberryPi – Room  
(PAN ON 메시지 수신)



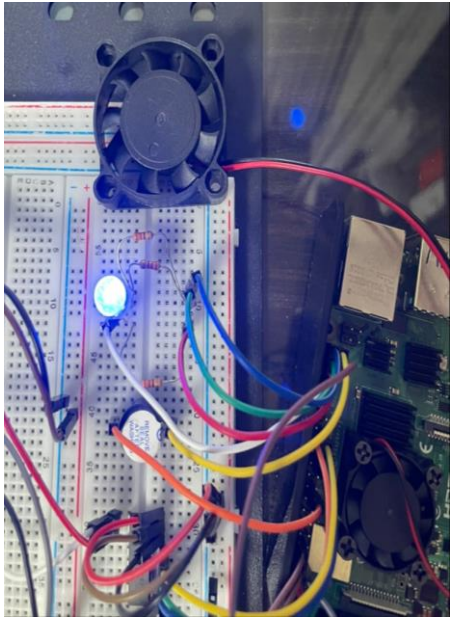
안드로이드 App  
(PAN ON 클릭)



119 상황실  
(신고 X)

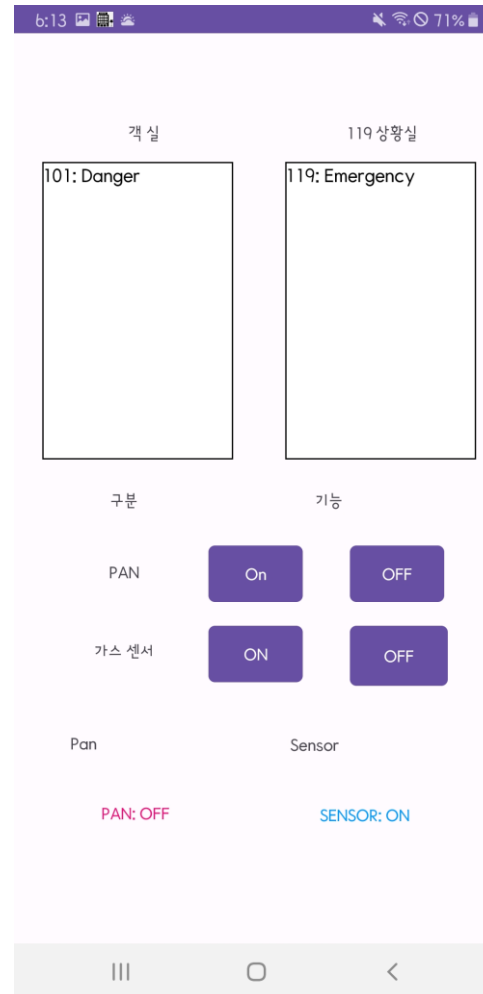
## 05 최종 결과물

### 결과물 – 팬 OFF

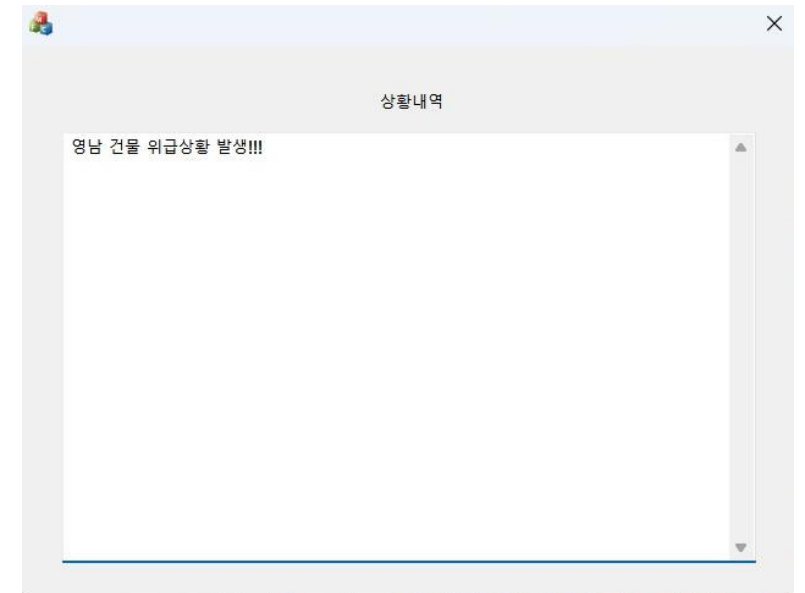


```
RED LED ON : Dangerous!!  
RED LED ON : Dangerous!!  
Message received: SENSOR_OFF  
Message received: SENSOR_ON  
BLUE LED ON : Safe  
BLUE LED ON : Safe  
BLUE LED ON : Safe  
BLUE LED ON : Safe  
BLUE LED ON : Safe  
RED LED ON : Dangerous!!  
RED LED ON : Dangerous!!  
BLUE LED ON : Safe  
Message received: PAN_OFF  
BLUE LED ON : Safe  
BLUE LED ON : Safe  
RED LED ON : Dangerous!!  
Message received: PAN_OFF  
BLUE LED ON : Safe
```

RaspberryPi – Room  
(PAN OFF 메시지 수신)



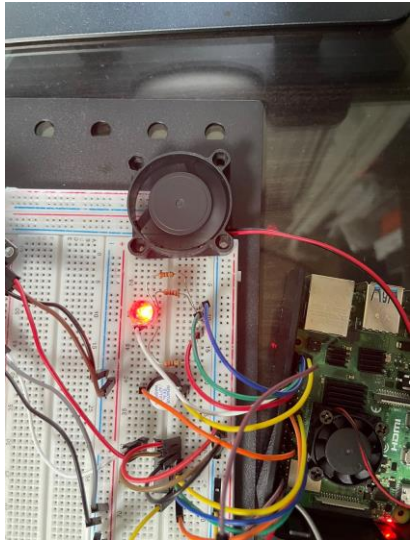
안드로이드 App  
(PAN OFF 클릭)



119 상황실  
(신고 X)

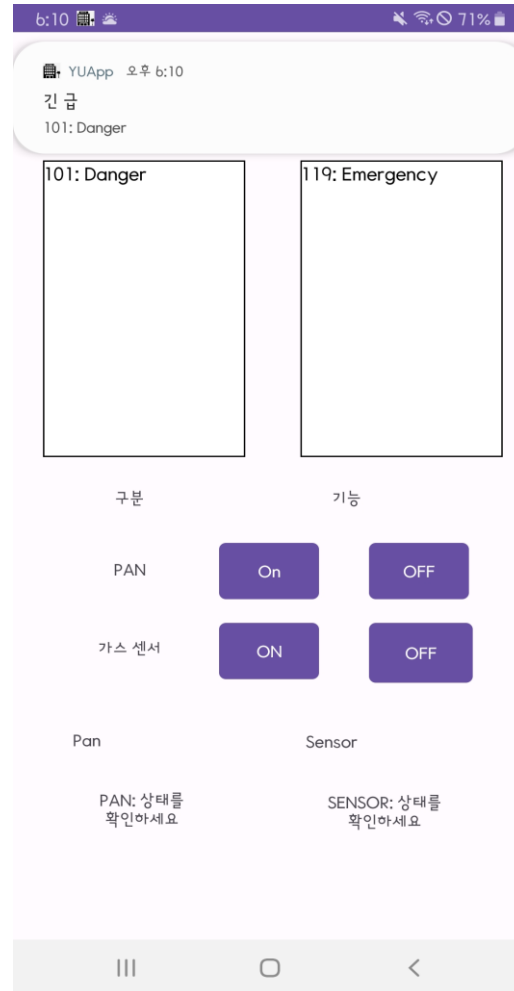
## 05 최종 결과물

### 결과물 – 메시지 전송

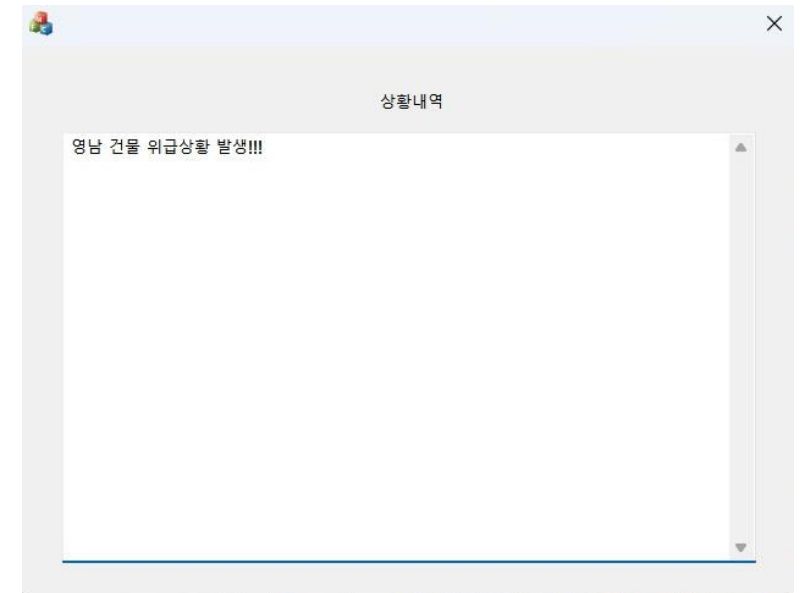


```
pi@raspberrypi:~/WiringPi/exa
TCP network
..
...
wiringPiSPISetup return = 5
RED LED ON : Dangerous!!
RED LED ON : Dangerous!!
RED LED ON : Dangerous!!
RED LED ON : Dangerous!!
GREEN LED ON : Warning!!
RED LED ON : Dangerous!!
RED LED ON : Dangerous!!
RED LED ON : Dangerous!!
RED LED ON : Dangerous!!
RED LED ON : Dangerous!!
RED LED ON : Dangerous!!
RED LED ON : Dangerous!!
RED LED ON : Dangerous!!
```

RaspberryPi – Room  
기준치 초과 (30초 지속)



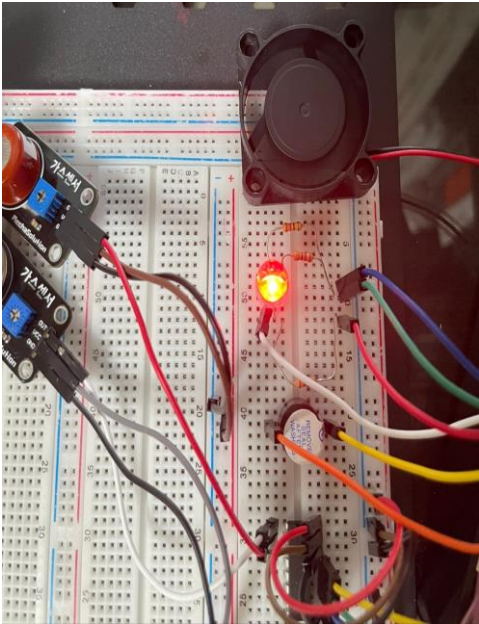
안드로이드 App  
(메시지 수신)



119 상황실  
(신고 접수)

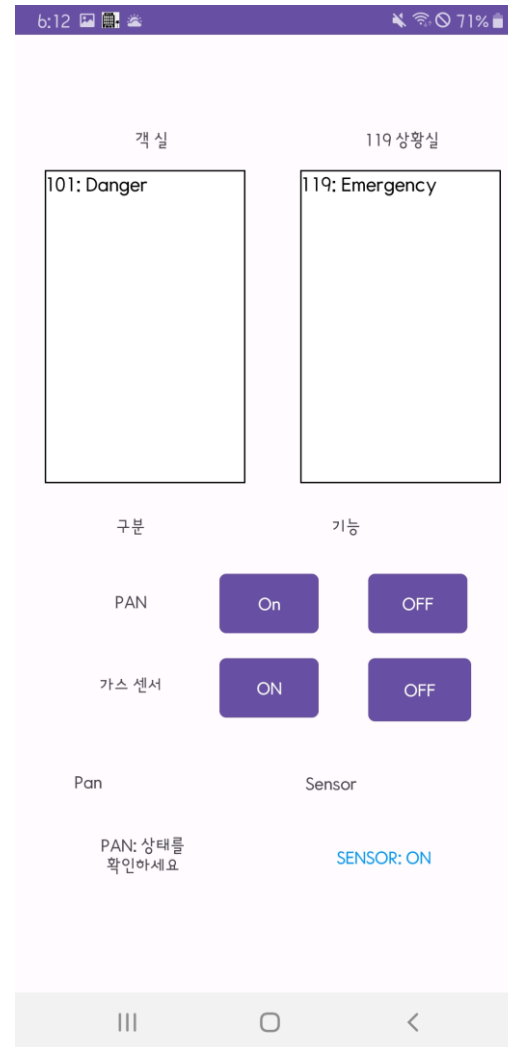
## 05 최종 결과물

### 결과물 – 센서 ON

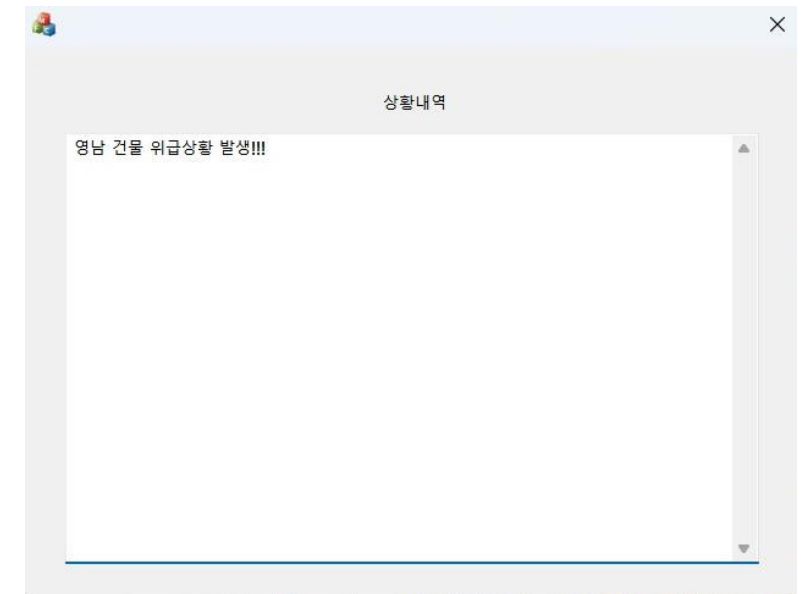


```
RED LED ON : Dangerous!!  
Message received: PAN_OFF  
RED LED ON : Dangerous!!  
RED LED ON : Dangerous!!  
BLUE LED ON : Safe  
RED LED ON : Dangerous!!  
Message received: SENSOR_OFF  
Message received: SENSOR_ON  
RED LED ON : Dangerous!!  
RED LED ON : Dangerous!!  
BLUE LED ON : Safe  
BLUE LED ON : Safe  
Message received: SENSOR_ON  
이미 센서가 작동 중입니다 .BLUE LED ON : Safe
```

RaspberryPi – Room  
(SENSOR ON 메시지 수신)



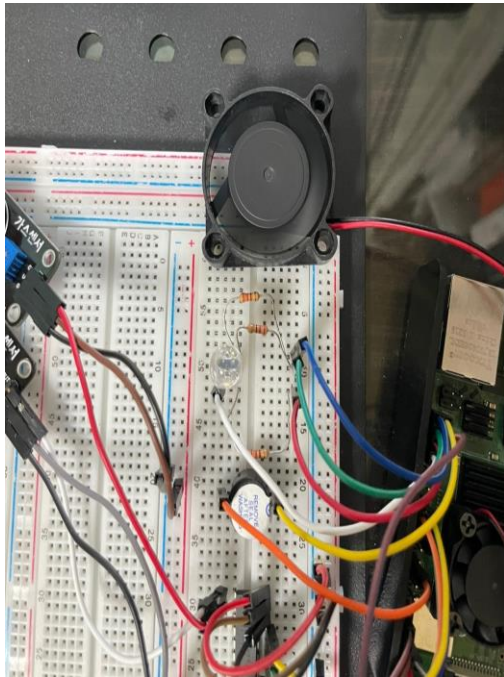
안드로이드 App  
(센서 ON 클릭)



119 상황실  
(신고 X)

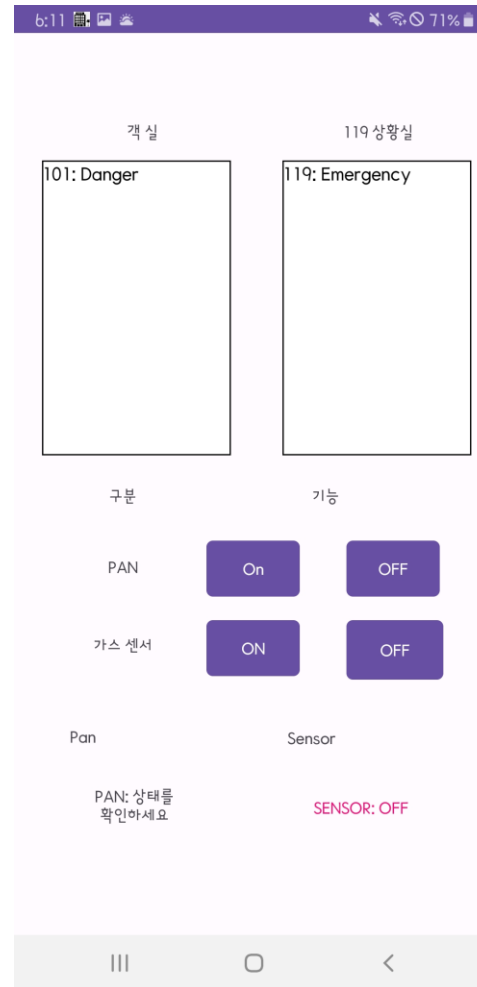
## 05 최종 결과물

결과물 – 센서 OFF

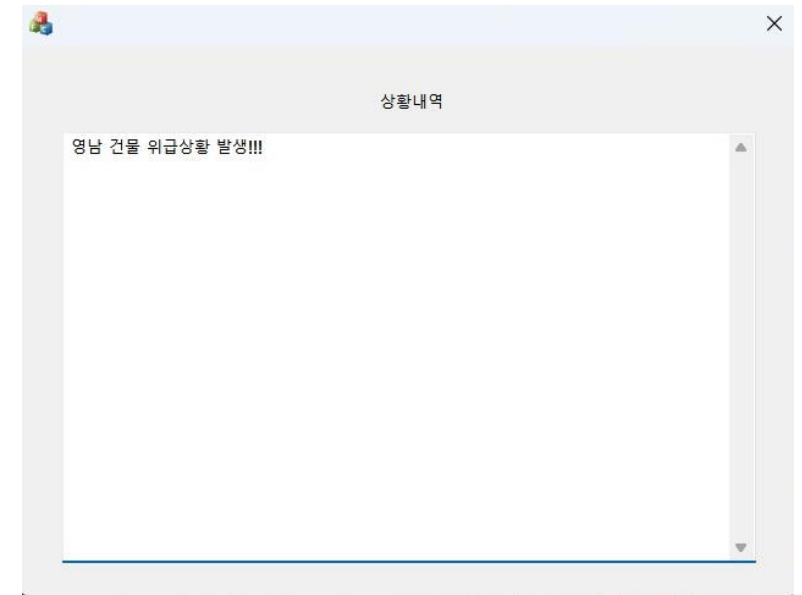


```
RED LED ON : Dangerous!!  
RED LED ON : Dangerous!!  
RED LED ON : Dangerous!!  
RED LED ON : Dangerous!!  
RED LED ON : Dangerous!!  
RED LED ON : Dangerous!!  
GREEN LED ON : Warning!!  
BLUE LED ON : Safe  
BLUE LED ON : Safe  
RED LED ON : Dangerous!!  
RED LED ON : Dangerous!!  
Message received: SENSOR_OFF
```

RaspberryPi – Room  
(SENSOR OFF 메시지 수신)



안드로이드 App  
(센서 OFF 클릭)

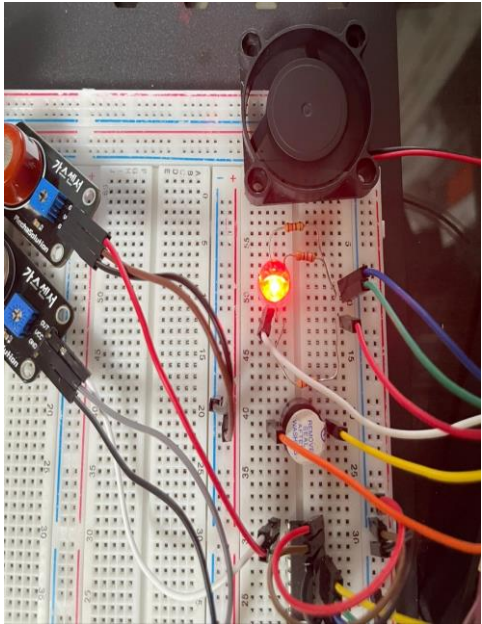


119 상황실  
(신고 X)



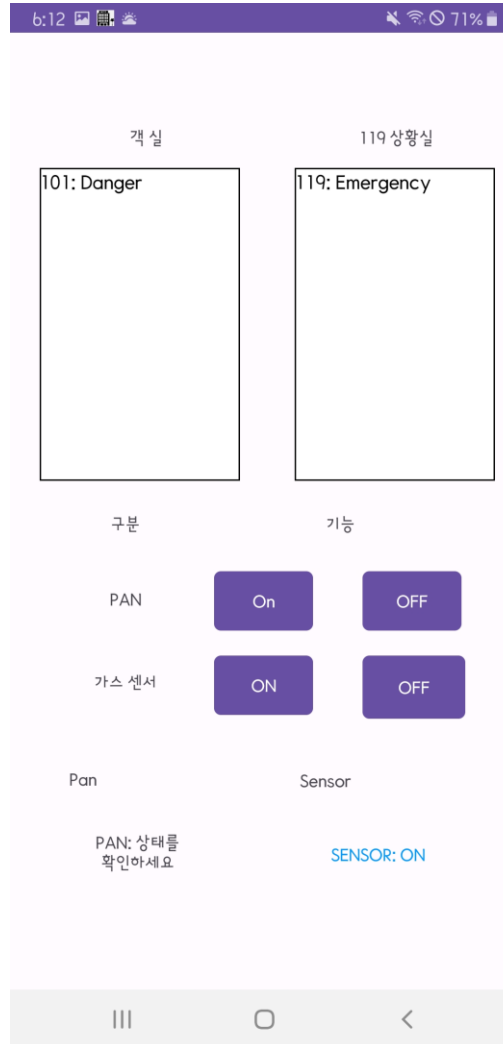
## 05 최종 결과물

결과물 – 센서 ON (처음 상태로)

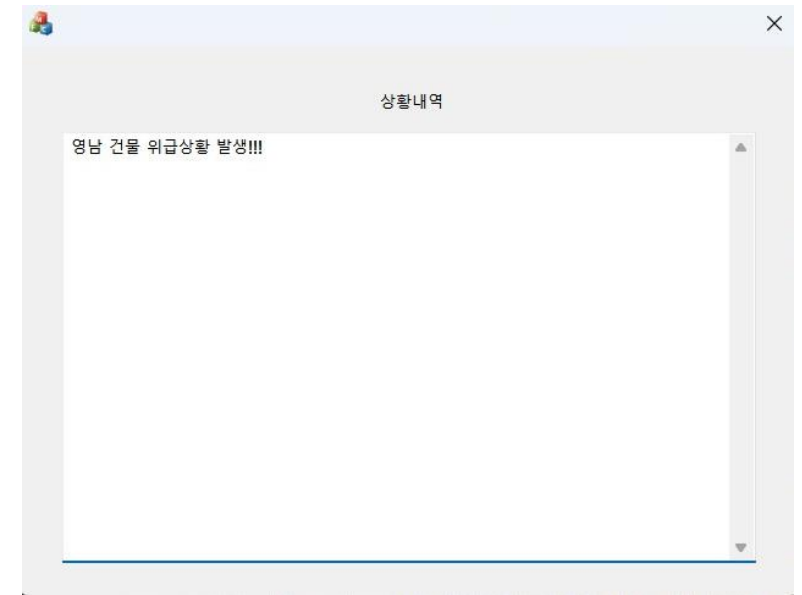


```
RED LED ON : Dangerous!!  
Message received: PAN_OFF  
RED LED ON : Dangerous!!  
RED LED ON : Dangerous!!  
BLUE LED ON : Safe  
RED LED ON : Dangerous!!  
Message received: SENSOR_OFF  
Message received: SENSOR_ON  
RED LED ON : Dangerous!!  
RED LED ON : Dangerous!!  
BLUE LED ON : Safe  
BLUE LED ON : Safe  
Message received: SENSOR_ON  
이미 센서가 작동중입니다.BLUE LED ON : Safe
```

RaspberryPi – Room  
(SENSOR OFF 메시지 수신)




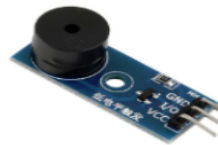



안드로이드 App  
(센서 ON 클릭)



119 상황실  
(신고 X)

## 06 진행 상세 결과

핵심 모듈		모듈 설명	완성도
양방향 Full-Duplex TCP 통신		<ul style="list-style-type: none"> <li>- MultiThread TCP 소켓 통신</li> <li>- Client-Server : 2개의 양방향 Full-Duplex 통신으로 정보 전달</li> <li>- Appplication 및 MFC 이용</li> </ul>	100%
		<ul style="list-style-type: none"> <li>- 센서에서 측정된 아날로그 값을 ADC 모듈을 Digital 값으로 변환, 출력</li> </ul>	100%
MQ-4	MQ-7B		
		<ul style="list-style-type: none"> <li>- 유독 가스 수치에 따라 LED를 다르게 출력</li> <li>- 기준 값을 넘으면, 부저 작동</li> </ul>	100%
LED	부저		
		<ul style="list-style-type: none"> <li>- FAN 제어 버튼을 클릭하여 상황에 따라 제어</li> <li>- 센서에서 측정된 값에 따른 작동 조건 설정</li> </ul>	100%
팬 (환기용)			

## 07 목표 대비 진행사항

	1주차	2주차	3주차	4주차	5주차	6주차	7주차	8주차	9주차	10주차	11주차	12주차	13주차	14주차	15주차	강원 강원 강원	진행 과목
아이디어 선정																강원 강원 강원	완료
재료 구입 및 주제 구체화																강원 강원 강원	완료
프로젝트 제안 발표 준비																강원 강원 강원	완료
TCP 통신 및 Application 구현																강원 강원 강원	완료
HW 설계																강원 강원 강원	완료
중간발표																강원 강원 강원	완료
중간 점검																강원 강원 강원	완료
프로토 타입																강원 강원 강원	완료
기말 발표																강원 강원 강원	완료



## 08 팀원 별 역할

구성원	이름	담당 역할	비고 (참고 사항)
팀장	한 종 헌	App 구현, HW 설계 Web Socket 구현	-
팀원 1	황 종 원	App 구현, HW 설계 Web Socket 구현	-

## 09 결과물 성능 분석

핵심 필수 기능	당초 성능 목표	구현 성능	달성 여부 분석
Client-Server / 양방향 Full-Duplex TCP 통신	50번 전송 시, 최소한 49번 성공 지연율 1초 이내	<ul style="list-style-type: none"> <li>- MultiThread TCP 소켓 통신 이용</li> <li>- Client-Server를 통해 2개의 양방향 Full-Duplex 통신으로 정보 전달</li> </ul>	다수의 MultiThread 이용 양방향 Socket 통신 구현 완료
MQ-4, MQ-7B 센서 유독가스 검출	최소 2초 이내 감지	<ul style="list-style-type: none"> <li>- 센서에서 측정된 아날로그 값 -&gt; ADC 모듈을 Digital 값으로 변환, 출력</li> </ul>	구현 완료
LED, 부저, 팬 상황 전달 및 개선	최소 2초 이내 작동	<ul style="list-style-type: none"> <li>- 가스 수치 -&gt; RGB LED 변화</li> <li>- 기준 값 초과 -&gt; 부저 작동</li> <li>- 상황 개선을 위한 팬 작동</li> </ul>	구현 완료
App -> RaspberryPi 사용자 경험(UX) 추가	최소 2초 이내 작동 지연율 1초 이내	<ul style="list-style-type: none"> <li>- Raspberry Pi에서 시스템 기능 전달 시, 메시지가 섞이지 않도록 MultiThread 구현</li> </ul>	APP 이용 RaspberryPi 제어 완료



Thank you