**2019년 1학기 인터넷 응용**

**X-Project 최종보고서**

S.M.B

(Smart Mail Box)

2019.06.13

한국외국어대학교 정보통신공학과 3팀

최동욱(201503370) 이정민(201502558) 정희원(201503164) 이경현(201702405) 최소원(201703536)

# 서론

본 프로젝트는 아두이노와 라즈베리파이를 이용하여 소켓 통신을 통해 사용자가 원격으로 우편물을 확인할 수 있게 하는 스마트 우편함이다. 아두이노에 초음파 센서와 LED를 연결하고 라즈베리파이가 센서 값 데이터를 받아 이를 처리한다. 초음파 센서가 일정 거리 안에 우편물을 감지하게 되면 아두이노 LED Actuator를 ON시키고, 라즈베리파이가 파이캠으로 우편함 내부의 사진을 찍어 사용자의 애 플리케이션에 전송한다. 사용자는 이를 애플리케이션 내에서 확인할 수 있다. 또 한 사용자가 애플리케이션 내에서 최근 기록 탭을 누르면 라즈베리파이에 연결된 카메라를 통해 가장 최근에 찍힌 우편물 사진을 볼 수 있으며, 우편물이 감지될 때마다 목록 탭에서 기록된 로그를 확인할 수 있다.

**목차**

1. **프로젝트 개요** .................................................................................................................................... **7**
2. **프로젝트 소개** .................................................................................................................................... **8**
   1. 제품 개요 ...................................................................................................................................... 8
   2. 소프트웨어 특징 ......................................................................................................................... 8
   3. 하드웨어 특징 ............................................................................................................................. 8
3. **구성도 및 설계** ................................................................................................................................. **9** 
   1. 전체 시스템 구성도 .................................................................................................................. 9
   2. 설치 장소 – 서버 구성도 ..................................................................................................... 10
   3. 서버 – 안드로이드 어플리케이션 구성도 ..................................................................... 11
   4. 시퀀스 다이어그램 .................................................................................................................. 12
4. **프로젝트 결과** ................................................................................................................................. **13**
   1. 프로그램 작동 동영상 ........................................................................................................... 13
   2. 아두이노 & 라즈베리파이 ................................................................................................... 13
   3. 안드로이드 어플리케이션 ..................................................................................................... 16
5. **프로젝트 수행 방법** ..................................................................................................................... **18**
   1. 요구 분석 .................................................................................................................................... 18
   2. 시스템 명세 ................................................................................................................................ 18
   3. 설계 ................................................................................................................................................ 19
   4. 구현 ................................................................................................................................................ 19
   5. 테스트 ........................................................................................................................................... 20
   6. 유지 보수 .................................................................................................................................... 20
6. **문제점 및 해결 방법** ................................................................................................................... **21**
7. **기대효과 및 활용분야** ................................................................................................................. **23**
   1. 기대효과 ....................................................................................................................................... 23
   2. 활용 가능 분야 ......................................................................................................................... 23
8. **프로젝트 세부 일정** ..................................................................................................................... **24**
9. **팀원 담당 업무** .............................................................................................................................. **26**
10. **부록** .................................................................................................................................................. **27**
    1. 코드 ............................................................................................................................................. 27
       1. 서버 코드 ........................................................................................................................... 27
       2. 클라이언트 코드 ............................................................................................................. 30
       3. 안드로이드 코드 ............................................................................................................. 33
       4. 아두이노 코드 .................................................................................................................. 34
    2. 주요 내용 설명 ....................................................................................................................... 36
       1. 소프트웨어 ......................................................................................................................... 36
       2. 하드웨어 ............................................................................................................................. 38
    3. 개발 환경 상세 설명 ........................................................................................................... 39
    4. 참고 문헌 .................................................................................................................................. 40
11. **느낀 점** ............................................................................................................................................ **41**

## 표 목차

[표 8-1] 프로젝트 1주차 일정 ....................................................................................................... 24

[표 8-2] 프로젝트 2주차 일정 ....................................................................................................... 24

[표 8-3] 프로젝트 3 주차 일정 ...................................................................................................... 25

[표 9-1] 팀원 개별 담당 업무 ....................................................................................................... 26

[표 10-1] 소프트웨어 주요 기능 ................................................................................................... 36

[표 10-2] 하드웨어 주요 기능 ....................................................................................................... 38

[표 10-3] 개발 환경 상세 설명 ..................................................................................................... 39

[표 10-4] 참고 문헌 ............................................................................................................................ 40

## 그림 목차

[그림 3-1] 전체 시스템 구성도 ....................................................................................................... 9

[그림 3-2] 설치 장소 – 서버 구성도 .......................................................................................... 10

[그림 3-3] 서버 – 안드로이드 어플리케이션 구성도 .......................................................... 11

[그림 3-4] 시퀀스 다이어그램 ....................................................................................................... 12

[그림 4-1] 아두이노 & 라즈베리파이 실행 장면 1 ................................................................... 13

[그림 4-2] 아두이노 & 라즈베리파이 실행 장면 2 ................................................................... 14

[그림 4-3] 아두이노 & 라즈베리파이 실행 장면 3 ................................................................... 14

[그림 4-4] 아두이노 & 라즈베리파이 실행 장면 4 ................................................................... 15

[그림 4-5] 아두이노 & 라즈베리파이 실행 장면 5 ................................................................... 15

[그림 4-6] 안드로이드 어플리케이션 실행화면 1 ................................................................ 16

[그림 4-7] 안드로이드 어플리케이션 실행화면 2 ................................................................ 16

[그림 4-8] 안드로이드 어플리케이션 실행화면 3 ................................................................ 17

[그림 4-9] 안드로이드 어플리케이션 실행화면 4 .................................................................. 17

[그림 4-10] 안드로이드 어플리케이션 실행화면 5 .............................................................. 17

[그림 10-1] 안드로이드 어플리케이션 로그인 ....................................................................... 36

[그림 10-2] 우편함 사진 찍은 기록 확인 ................................................................................ 36

[그림 10-3] 아두이노 초음파 센서 .............................................................................................. 38

[그림 10-4] 아두이노 LED Actuator ............................................................................................ 38

[그림 10-5] RPI NOIR CAMERA BOARD .................................................................................... 38

## 1.프로젝트 개요

우리나라의 현대 주거사회는 사람들이 아파트, 빌라, 오피스텔 등과 같은 다수의 사람들이 한 공간에 모여서 사는 주거형태로 주를 이루고 있다. 이 러한 환경 때문에 건물에 설치되어 있는 우편함들은 본인의 현관문 앞이 아닌 건물 외벽이나 1층 현관 내부에 설치되어 있어 집 밖에 나가 직접 우편함을 확인하지 않는 한 우편물이 도착했는지 알 수 있는 방법이 없다. 본 팀이 진행하고자 하는 이번 프로젝트는 이러한 문제점을 해결하기 위 해 사용자의 세대 우편함에 들어오는 우편물들을 스마트폰 어플리케이션 을 통해 실시간으로 확인할 수 있는 시스템을 구현하고자 한다. 본 프로젝 트를 통해 우편함에 도착한 우편물이 있는지 없는지 매번 잊지 않고 확인 을 해야 하는 문제의 해결을 목적으로 한다.

본 프로젝트는 라즈베리파이를 이용하여 소켓통신을 통해 사용자가 원격 으로 우편을 확인할 수 있게 하는 스마트 우편함이다. 라즈베리파이에 초 음파 센서와 LED Actuator를 연결하고 라즈베리파이가 센서 값의 데이터를 받아 이를 처리한다. 센서가 일정 거리 내의 우편물을 감지하게 되면 아두 이노 LED Actuator를 ON시키고 라즈베리파이에 연결된 파이캠을 이용하여 우편함 내부의 사진을 찍는다.

시중에 나와있는 제품으로 GSM 모듈을 이용하여 우편물 도착 알림과 개 수를 사용자에게 제공하는 것과 달리 본 팀이 제작한 SMART MAILBOX는 라즈베리파이를 이용하여 제작하였다. 우편함 내부의 모습을 스마트폰 어 플리케이션을 통해 실시간으로 사진을 확인할 수 있으므로 해당 우편물이 편지인지 광고 전단지인지 등을 대략적으로 파악할 수 있도록 하여 기존 에 나와있는 제품에 차별성을 두어 우편물 확인을 편리하게 할 수 있도록 하였다.

## 2.프로젝트 소개 및 특징

### 2.1. 제품 개요

SMART MAILBOX는 똑똑한 우편함을 만들겠다는 뜻으로 정한 제품명이

다. 제품을 기존의 위치처럼 1층 현관 내부에 설치하나 추가로 본 팀이 추가한 센서와 제작한 애플리케이션을 통해 스마트폰을 통해서 우편물 의 여부를 확인할 수 있게 서비스를 제공한다.

### 2.2. 소프트웨어 특징

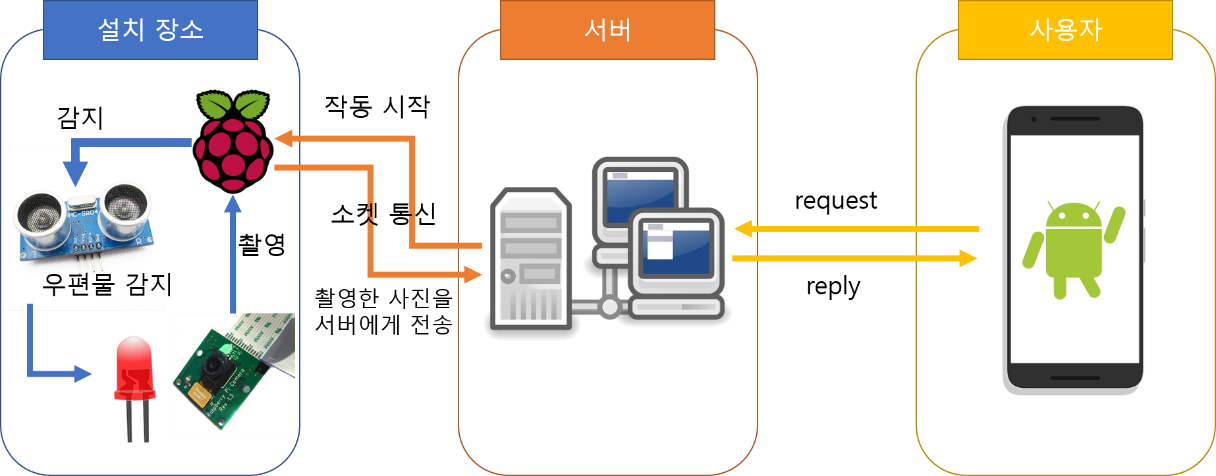
스마트폰 OS 중에 ios보다 점유율이 높은 Android OS로 선택하여 더 많은 사람들이 이용할 수 있도록 고려하였다. 또한 무선네트워크를 통 한 원격조종을 가능하게 하여 온라인으로 접속이 가능한 환경이면 우 편물이 들어온 것을 언제 어디서나 확인할 수 있다.

### 2.3. 하드웨어 특징

우편함 내부에 설치되는 디바이스로는 라즈베리파이와 아두이노를 사 용하였다. 라즈베리파이와 아두이노 모두 비용이 저렴한 편이고 크기가 작아 쉽게 탈부착이 가능하다. 따라서 우편함 뿐만이 아니라 더 나아가 무인 택배함, 사물함과 같은 보관함에도 간단하게 적용할 수 있다. 또한, 추가로 우편함안에서 우편물이 감지되면 LED Actuator를 ON시켜 어두 운 환경에서도 우편함 내부를 환하게 하여 카메라로 확인을 하는데 조 금 더 명확하게 해주도록 하였다.

## 3.구성도 및 설계

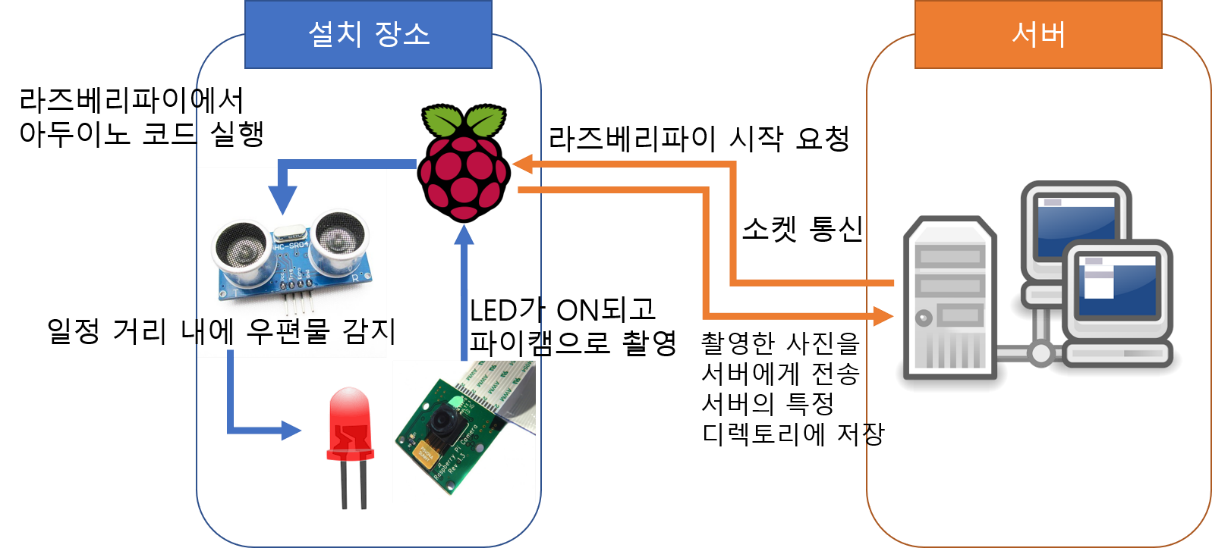
### 3.1. 전체 시스템 구성도



[그림 3-1] 전체 시스템 구성도

서버가 먼저 열려 있고, 라즈베리파이가 시작을 하면 서버에게 소켓 통 신 연결을 요청한다. 둘이 연결이 되면, 라즈베리파이는 아두이노를 통 해 우편함의 물체와의 거리 값을 측정한다. 우편물이 감지되어 일정한 범위 내의 거리 값이 측정되면 LED가 ON되고 파이캠을 통해 우편함 내부의 촬영을 한다. 라즈베리파이는 연결되어 있는 서버에게 사진을 전송하고, 서버는 이 사진 파일을 받아서 이름을 정한 후 저장을 하고, 이때 받은 날짜와 시간 또한 저장한다. 안드로이드는 이 서버가 사진을 저장해 놓은 특정 디렉토리에 접근하여 어플리케이션을 수행하도록 한 다.

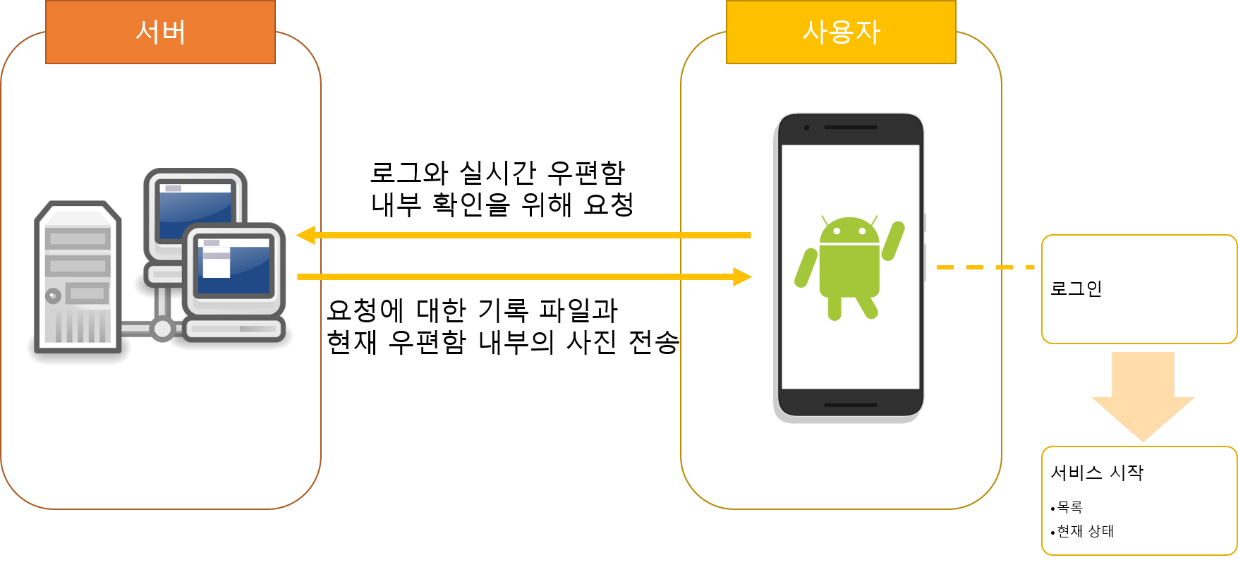
### 3.2. 설치 장소 – 서버 구성도



[그림 3-2] 설치 장소 – 서버 구성도

우선 서버는 항상 열려 있어야 한다. 위의 설명과 마찬가지로 라즈베리 파이가 시작을 하면 서버에게 소켓 통신 연결을 할 것을 요청한다. 서 버가 이 요청을 수락하게 되면, 라즈베리파이가 클라이언트가 되어 서 버와 연결된다. 라즈베리파이는 아두이노를 통해 초음파 센서를 이용하 여 우편함 내부의 물체와의 거리 값을 시리얼 통신을 통해 받아온다. 우편물이 들어와 우편물을 감지하게 되면 거리 값이 일정한 범위 내에 들어왔다는 것을 의미하고, 이는 LED Actuator의 상태를 ON시킨다. 또 한 라즈베리파이는 파이캠을 통해 우편함 내부의 사진을 찍게 된다. 찍 은 사진은 라즈베리파이 폴더에 저장되며, 이 사진 파일을 라즈베리파 이가 다시 읽고 연다. 연 파일을 바이트 단위로 서버에 전송하며, 전송 을 완료하면 서버는 지정한 위치로 파일을 쓴다.

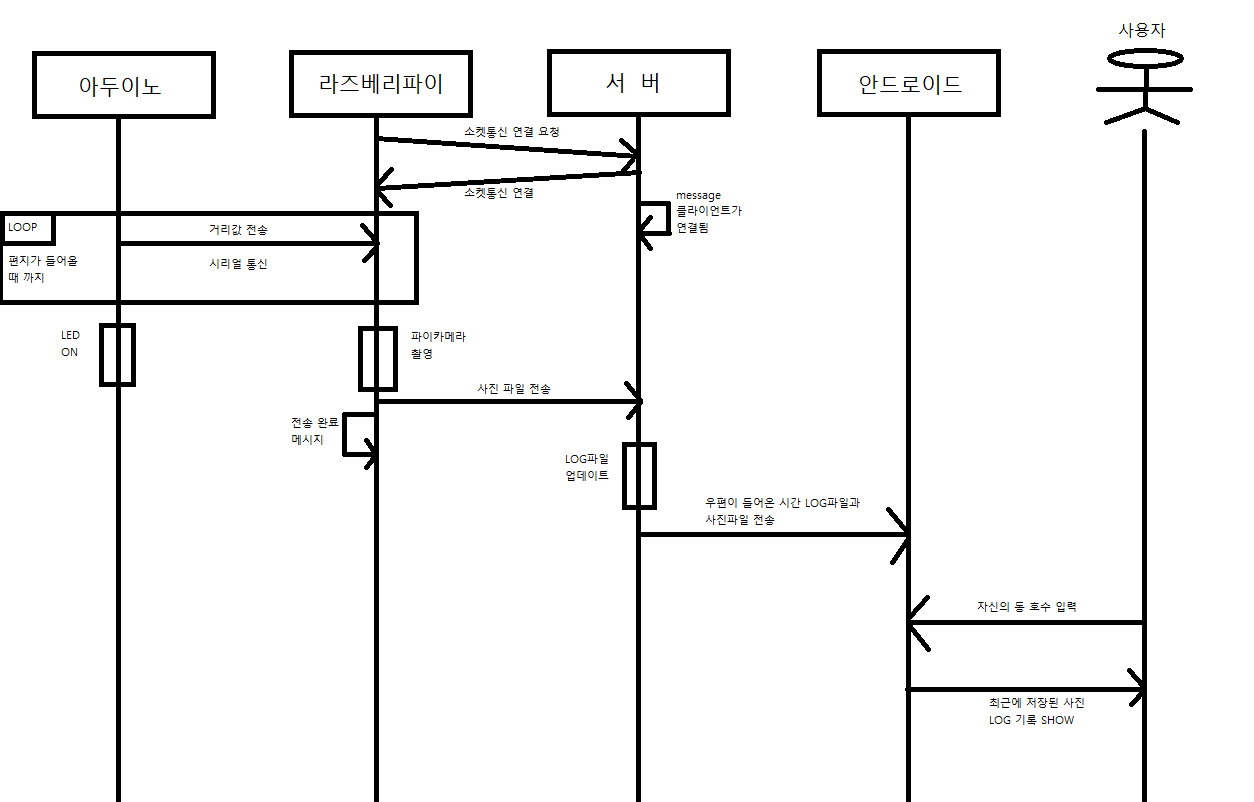
### 3.3. 서버 – 안드로이드 어플리케이션 구성도



[그림 3-3] 서버 – 안드로이드 어플리케이션 구성도

안드로이드 어플리케이션은 라즈베리파이와 서버의 소켓통신을 통해 받은 파일을 저장한 서버의 특정 디렉토리에 접근한다. 디렉토리 내부 에는 라즈베리파이가 파이캠으로 사진을 찍은 날짜, 시간이 기록되어 있고, 그와 함께 사진 파일들이 정해 놓은 형식의 이름으로 저장되어 있다. 안드로이드 어플리케이션은 이 사진파일과 기록으로 어플리케이 션을 수행하는데 사용한다. 안드로이드 어플리케이션은 본인의 거주 호 수를 선택한 후 로그인에 성공하여야 사용할 수 있게 제작하였으며, 로 그인을 성공하고 나면 그동안의 기록을 볼 수 있는 “목록”과 실시간으 로 상황을 볼 수 있는 “최근 기록” 두 가지 탭 서비스를 이용할 수 있 다.

### 3.4. 시퀀스 다이어그램



[그림 3-4] 시퀀스 다이어그램

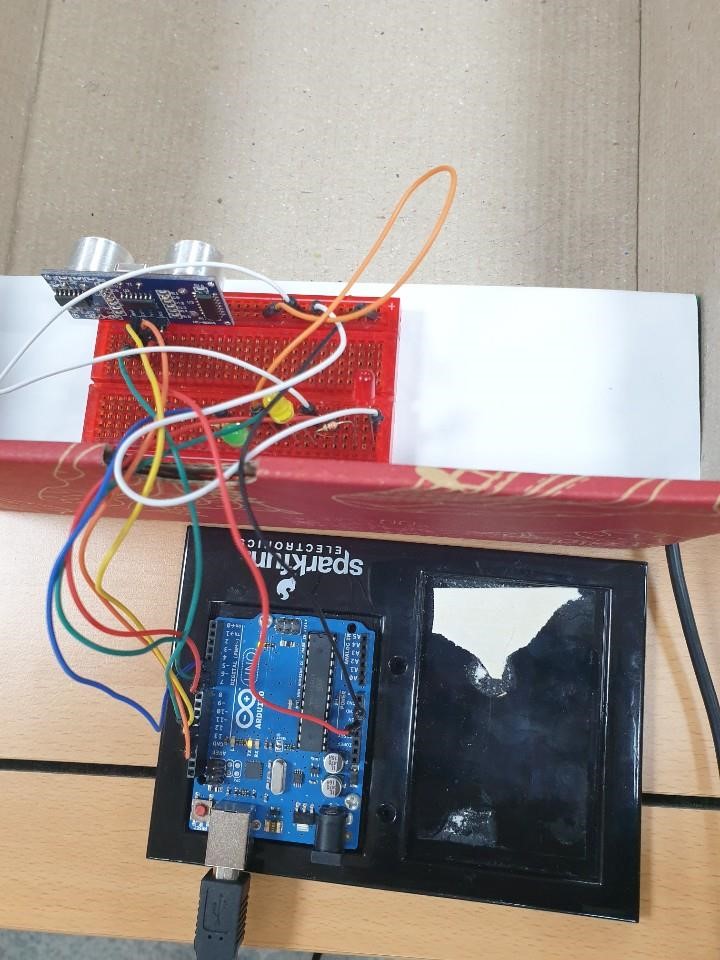
전체 흐름도를 위와 같이 시퀀스 다이어그램으로 나타낸 것이다.

## 4. 프로젝트 결과

4.1. 프로그램 작동 동영상

<https://github.com/xproject-team3/Xproject>-> readme

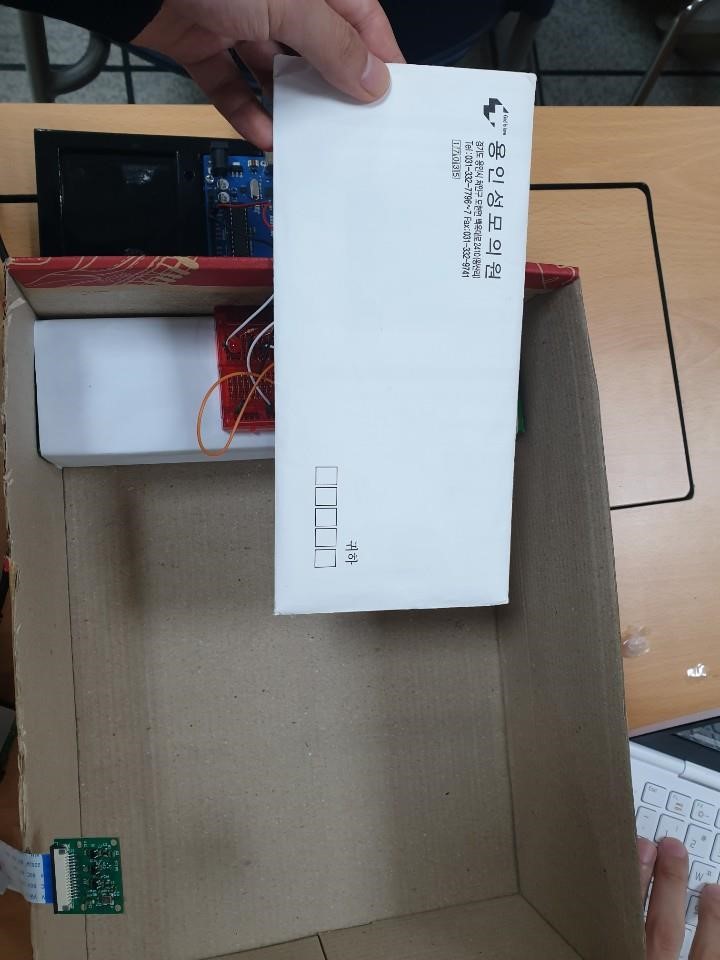
### 4.2. 아두이노 & 라즈베리파이

1. 아두이노 센서 설치 확인

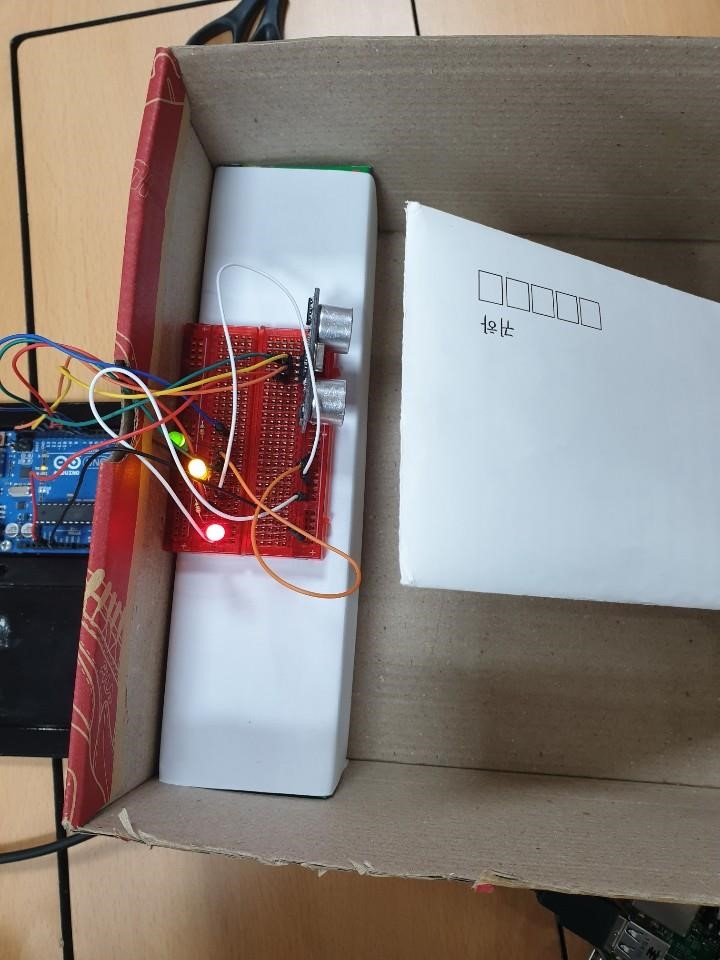
[그림 4-1] 아두이노 & 라즈베리파이 실행 장면 1

1. 라즈베리파이의 파이 캠 설치 확인

[그림 4-2] 아두이노 & 라즈베리파이 실행 장면 2

1. 우편물이 우편함 안 으로 들어오는 시뮬 레이션

[그림 4-3] 아두이노 & 라즈베리파이 실행 장면 3

1. 아두이노의 초음파 센서로 일정 거리 내 에 우편물이 있다고 감지 -> LED ON

[그림 4-4] 아두이노 & 라즈베리파이 실행 장면 4

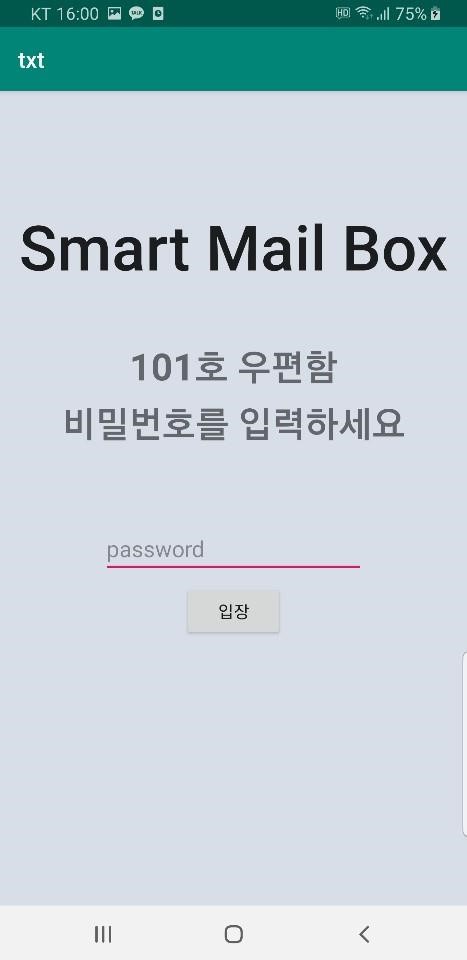
1. 초음파 센서로 측정 한 거리 값을 라즈베 리파이로 보낸 후 파 이캠으로 우편함 내 부 촬영

[그림 4-5] 아두이노 & 라즈베리파이 실행 장면 5

### 4.3. 안드로이드

1. 안드로이드 어플리케이션을 실행 하면 좌측의 사진처럼 화면이 나 온다. 이때 사용자가 거주하는 호 수를 선택하면 다음 화면으로 넘 어간다.

[그림 4-6] 안드로이드 어플리케이션 실행화면 1

1. 사용자의 호수를 선택하면 좌측의 사진처럼 비밀번호를 입력하는 화 면이 나타난다. 사용자가 비밀번호 를 맞게 입력하면 다음 창으로 넘 어간다.

[그림 4-7] 안드로이드 어플리케이션 실행화면 2

1. 좌측의 화면처럼 사용자가 비밀번호 를 틀리게 입력할 경우 “비밀번호를 다시 입력하세요”라고 메시지가 뜨고 로그인이 되지 않는다.

[그림 4-8] 안드로이드 어플리케이션 실행화면 3

1. 성공적으로 로그인을 하면 두 개의 탭 화면이 나타나는데 “목록” 탭에 들어가면 좌측 사진처럼 사진 목록 이 뜬다.

[그림 4-9] 안드로이드 어플리케이션 실행화면4

1. “최근 기록” 탭을 들어가면 가장 최 근에 들어온 우편함을 확인할 수 있

다.

[그림 4-10] 안드로이드 어플리케이션 실행화면 5

## 프로젝트 수행 방법

### 5.1. 요구 분석 (Requirement analysis)

* 라즈베리파이가 센서를 통해서 물체를 감지하고, 물체를 감지하였을 때 파이캠을 작동시켜 우편함 내부의 사진을 찍어 서버로 전송해야 한다.
* 라즈베리파이에 파이캠을 꽂을 수 있는 보드에는 본 팀이 가지고 있 는 센서를 연결할 수 없으므로 라즈베리를 아두이노와 연결하여 아 두이노의 센서를 사용하도록 한다. 본 팀이 사용할 수 있는 아두이노 센서가 제한되어 있으므로 과사무실에서 대여할 수 있는 초음파 센 서를 사용하도록 한다.
* 아두이노의 초음파 센서가 일정 거리 내에서 우편물을 감지하면 LED Actuator를 ON시키고 라즈베리파이는 파이캠으로 우편함 내부의 사 진을 찍어서 서버로 전송한다.
* 본 프로젝트를 진행하는 데에는 아두이노 초음파 센서와 LED

Actuator, 라즈베리파이, 파이캠을 필요로 한다.

### 5.2. 시스템 명세 (System Specification)

* 아두이노 센서로 우편물이 감지되면 라즈베리파이에게 알린다.
* 라즈베리파이는 파이캠을 작동한다.
* 서버는 라즈베리파이에서 파이캠으로 찍은 사진을 받는다.
* 안드로이드 애플리케이션은 서버에 접근하여 기능을 수행한다.

### 5.3. 설계 (Design)

* 아두이노는 초음파 센서를 이용하여 우편물이 일정 거리 내에 감지 되면 LED를 ON시키고, 라즈베리파이에게 알린다.
* 라즈베리파이는 파이캠을 이용하여 촬영을 하고, 사진 파일을 서버로 전송한다.
* 서버는 사진 파일을 받아서 파일의 이름을 정해주고, 미리 정해 둔 디렉토리에 저장을 하며, 날짜와 시간을 저장한다.
* 안드로이드는 서버의 특정 디렉토리에 접근하여 촬영을 했던 기록(날 짜, 시간)과 사진을 얻어 애플리케이션에서 보여지도록 수행한다.
* 위 과정을 통해 사용자가 이 안드로이드 어플리케이션을 통해 우편 함 내부의 사진을 볼 수 있게 한다.

### 5.4. 구현 (Implementation)

* 아두이노는 C언어와 C++언어를 사용하는데 그 중 C언어를 사용하여 통합개발환경(IDE)인 아두이노 소프트웨어를 통해 초음파 센서와 LED

Actuator를 구현하였다.

* 라즈베리파이는 본 팀원들이 가장 익숙한 파이썬을 사용하여 클라이 언트 프로그램을 작성하였다. 클라이언트는 사진을 서버로 전송한다.
* 서버 프로그램 또한 파이썬 언어를 사용하여 작성하였다. 서버는 클 라이언트인 라즈베리파이에서 사진을 받아 정해 둔 특정 디렉토리에 저장을 하고, 파일의 이름을 정해준다. 또한 날짜와 시간을 기록하여 저장해 둔다.
* 안드로이드 스튜디오에서는 자바와 코틀린 언어 중 자바를 사용하여 작성하였다. 코틀린은 본 팀의 팀원 중 단 한 명도 사용해 본 적이 없기 때문에 자바를 선택하여 프로그래밍을 하는데 사용하였다.
* 위와 같이 본 팀은 사용자의 요구, 프로그래머의 능력, 현재 사용중인 언어, 컴파일러의 가용성과 품질, 지원 가능한 개발도구, 언어의 호환 성, 개발 경험 등을 고려하여 각 프로그램마다 프로그래밍 언어를 선 택하여 사용하였다.

### 5.5. 테스트 (Test)

* 단위 테스트 : 아두이노, 라즈베리파이, 서버, 안드로이드 스튜디오 등 각각마다 수행이 잘 되는지 개별적으로 확인하였다.
* 통합 테스트 : 아두이노와 라즈베리파이를 연결하여 센서 값을 잘 받 아오는지 확인하였고, 라즈베리파이와 서버의 소켓 통신이 원활하게 이루어지는지 확인하였다. 또한 안드로이드 스튜디오 개발 중 서버에 접근하여 사진을 잘 불러올 수 있는지 확인하였다.
* 인수 테스트 : 아두이노 – 라즈베리파이 – 서버 – 안드로이드 어플리 케이션으로 연결하여 서버를 열고 라즈베리파이가 시작되었을 때, 아 두이노의 초음파 센서가 일정 거리 내의 우편물을 감지하고 LED를 ON시키는지, 그리고 라즈베리파이가 파이캠으로 우편함 내부의 사진 을 찍고 서버로 전송하는데 문제가 없는지 확인하였다. 또한, 그렇게 서버가 받아온 사진 파일들에 안드로이드 어플리케이션에서 제대로 접근하여 파일을 사용하는데 문제가 없는지 확인하였다.

우리는 개발자, 관리자 입장에서 알파 테스트를, 사용자 입장에서 베타 테스트 모두 마쳤다.

### 5.6. 유지 보수 (Maintenance)

- 라즈베리파이와 안드로이드 어플리케이션 간 여러 차례의 연동 테스트를 통해 오차를 줄이고 요구사항을 오류 없이 수행하도록 하였다.

## 5. 문제점 및 해결 방법

Q. “마이크로프로세서응용설계” 과목에서 사용하는 한백 라즈베리파이 에 파이캠을 연결하고자 하였으나 한백전자 회사에서 보드 2개를 연결한 후 납땜을 해 두어 파이캠을 연결하는 곳에 연결하기 어려 워졌다. 파이캠을 억지로 연결하게 되면 케이블이 꺾이는 문제도 있 는 데다가, 본 팀원들이 라즈베리파이 센서를 제어하는데 C언어만 사용했었는데 파이캠은 파이썬 언어 또는 C++언어를 지원하여 두 센서를 연동하는 것이 어려웠다.

➔ 우선 기존에 사용하던 라즈베리파이 보드가 아닌 새로운 라즈베리 파이를 구하여 새로 얻은 라즈베리파이 보드에 파이캠을 연결하도 록 하였다. 그리고 C언어를 파이썬으로 바꾸기 위해 Rpi.GPIO 모듈 을 사용하고자 했는데 센서가 입력을 제대로 받지 못하였다.

Q. 위의 문제와 별개로 라즈베리파이 보드를 바꾸고 나니 한백전자에 서 제공한 센서들을 이용하기 수 없게 되어 불을 켜주는 역할을 했 던 LED Actuator를 사용하기 어려워졌다.

➔ 저번 PA-6 IoT Client/Server 과제에서 수행했던 대로 라즈베리파이에 서 아두이노를 제어하도록 하였다. 아두이노의 센서와 Actuator를 사용하기로 하였다. 두 가지 문제를 해결할 수 있게 되었다.

Q. 본 팀이 구할 수 있는 아두이노 센서는 제한적이었다. 모션을 감지하 는 센서가 없어서 프로젝트 진행하는데 차질이 생길 뻔했다.

➔ 초음파 센서를 사용하게 되었는데 편지 봉투를 수백번씩 넣어보며 가장 적절한 거리와 적합한 센서의 위치, 가장 적합한 파이캠의 위 치를 찾을 수 있었다.

Q. 파이캠을 연결한 후 작동시켰는데 오류 메시지가 떠서 프로젝트를 진행도를 진전시키는데 어려움이 있었다.

➔ 라즈베리파이가 파이캠을 인식하지 못하는 오류였는데, 구글링을 통 해 갖은 방법을 다 동원해도 해결할 수 없어 새로운 파이캠을 구하 여 연결하였다. 기존의 파이캠의 라즈베리파이 보드에 연결하는 부 분에 결함이 생긴 것으로 추측된다.

Q. 라즈베리파이에 저장되는 사진 파일을 어떻게 서버로 전송해야 할 지 막막했다.

➔ 서버로 사진의 바이너리 데이터를 보내고, 서버에서 자체적으로 사 진 파일을 open and write하는 방식으로 구현할 수 있었다.

Q. 시리얼 통신을 하면서 데이터 전송의 지연 때문에 아두이노의 거리 값이 적절하지 못하게 나오는 문제가 생겼었다.

➔ 실행 파일에서 delay를 통해 적절한 값을 주어 조절할 수 있었다.

## 6. 기대효과 및 활용분야

### 7.1. 기대효과

* 평상시 번거로워 확인을 하지 않은 우편함을 어플리케이션을 통 해 본인 우편함의 내용 상태를 확인할 수 있다.
* 사용자가 자택이 아닌 직장 혹은 다른 장소에서 있어도 어플리케 이션을 통해 우편물이 들어오는 즉시 우편물의 출처와 보낸 사람을 확인할 수 있다.
* 우편함뿐만 아니라 다른 분야에 접목시킬 수 있는 실용적인 어플 리케이션이다.

### 7.2. 활용 가능 분야

- 우편물과 같이 유인물의 출입이 있는 곳이라면 제한 없이 사용할 수 가능하고 센서를 달리하여 어플리케이션을 구현할 시 보안 관 련 영상 및 이미지 전송 시스템을 만들 수 있다.

<예시 1> 물품 보관함 어플리케이션 사용자가 물품 보관함을 사용할 시 처음 물품 을 위치시킨 사진과 본인 물품 보관함에 이상 행동 및 물품 의 이동이 생길 시 센서가 반응하여 사용자에게 사진을 전 송하여 보관의 안정성을 높일 수 있다.

<예시 2> 택배함 자신의 택배함에 상품이 들어왔을 시 센서가 인식하여 우편 함과 마찬가지로 어플리케이션 사용자에게 사진을 보내 택 배의 출처와 상품을 확인할 수 있다. 또한 택배에 다른 움 직임이 감지될 시 사용자에게 시간과 물품 상태를 실시간으 로 확인하여 보완을 강화할 수 있다.

## 7. 프로젝트 세부 일정

### <1주차>

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 구분 | 추진내용 |  |  |  | 수행기간 |  |  |  |
| 5/24 | 5/25 | 5/26 | 5/27 | 5/28 | 5/29 | 5/30 |
| 기획 | 프로젝트 계획 회의 |  |  |  |  |  |  |  |
| 시나리오 구상 |  |  |  |  |  |  |  |
| 개발 | 관련 지식 학습 |  |  |  |  |  |  |  |
| 기획 | 주제 선정 재회의 |  |  |  |  |  |  |  |
| 프로젝트 계획 수립 |  |  |  |  |  |  |  |
| 시나리오 구상 |  |  |  |  |  |  |  |
| 개발 | 관련 지식 학습 |  |  |  |  |  |  |  |

[표 8-1] 프로젝트 1주차 일정

### <2주차>

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 구분 | 추진내용 |  |  |  | 수행기간 |  |  |  |
| 5/31 | 6/1 | 6/2 | 6/3 | 6/4 | 6/5 | 6/6 |
| 개발 | 아두이노 개발 |  |  |  |  |  |  |  |
| 라즈베리파이와 아두이노 연동 |  |  |  |  |  |  |  |
| 라즈베리파이 카메라 연동 |  |  |  |  |  |  |  |
| 서버 구축 |  |  |  |  |  |  |  |
| 점검 | 1차 보고서 작성 |  |  |  |  |  |  |  |
| 구상도 재확인 |  |  |  |  |  |  |  |

[표 8-2] 프로젝트 2주차 일정

### <3주차>

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 구분 | 추진내용 |  |  |  | 수행기간 | |  |  |
| 6/7 | 6/8 | 6/9 | 6/10 | 6/11 | 6/12 | 6/13 |
| 기획 | 최종 제품 구상 |  |  |  |  |  |  |  |
| 개발 | 라즈베리파이와 아두이노 연동 |  |  |  |  |  |  |  |
| 라즈베리파이 카 메라 연동 |  |  |  |  |  |  |  |
| 안드로이드 앱  개발 |  |  |  |  |  |  |  |
| 최종 | 디버깅 |  |  |  |  |  |  |  |
| 최종보고서 작성 |  |  |  |  |  |  |  |
| 발표 |  |  |  |  |  |  |  |

[표 8-3] 프로젝트 3주차 일정

## 8. 팀원 담당 업무

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **순번** | **이름** | **학번** | **학년** | **담당역할** | |
| **정** | **부** |
| 1 | 최동욱 | 201503370 | 3 | 총괄 | 서버(소켓통신) |
| 2 | 이정민 | 201502558 | 3 | 안드로이드(APP) | 라즈베리파이 |
| 3 | 정희원 | 201503164 | 3 | 서버 | 장치 |
| 4 | 이경현 | 201702405 | 3 | 아두이노 | 라즈베리파이 |
| 5 | 최소원 | 201703536 | 3 | 안드로이드(UI) | 안드로이드(APP) |

[표 9-1] 팀원 개별 담당 업무

## 9. 부록

### 10.1. 코드

10.1.1. 서버 코드

|  |
| --- |
| final-server.py |
| from socket import socket, AF\_INET, SOCK\_STREAM # 소켓 import import time    #이미지 파일 저장 위치  src =  "C:/Users/cheal/AndroidStudioProjects/xproject/app/src/main/res/ drawable/"    # a를 넣고 부르면 이미지파일 저장 b를 넣고 부르면 text 파일에 저장 def fileName(x) :  dte = time.localtime() Year = dte.tm\_year  Mon = dte.tm\_mon  Day = dte.tm\_mday  Hour = dte.tm\_hour  Min = dte.tm\_min  Sec = dte.tm\_sec    textFile = str(Year) + '\_' + str(Mon) + '\_' + str(Day) + '\_' + str(Hour) + '\_' + str(Min) imgFileName = src + textFile + '.jpg'    if x == 'a': |

|  |
| --- |
| return imgFileName elif x == 'b' :  return textFile    def echo\_server(my\_port) :  sock = socket(AF\_INET, SOCK\_STREAM) # 소켓 객체를 생성  sock.bind(('192.168.0.63', my\_port)) # 소켓객체에 주소  값을 바인딩 시킴. 호스트와 포트로 된 튜플 값을 인자로 받음  sock.listen(5) # 리스닝 수 = 5 print('server started')  while True: # 프로세스가 죽을 때까지 conn, client\_addr = sock.accept() # 서버 소켓에 클라이언트가 연결되면 클라이언트 소켓, 주소를 반환  print('connected by', client\_addr) # 어떤 주소에서  연결되었는지 프린트 try :  f = open(fileName('a'), 'wb') #시간정보를 받아와서 파일을 write binary로 open while True:  data = conn.recv(4096) #데이터를  512Byte씩 받아와서  if not data: # 데이터가 오지 않을  때까지  break else:  f.write(data) # 파일을 작성  f.close()  file = open(src + 'filename.txt', 'a') #  텍스트파일(LOG)에  file.write(fileName('b') + '\n') # 현재 시간을 |
| 저장 file.close()    #''' # wb모드로 open한 이유  # 바이너리 파일에는 파일의 형식에 관한 내용이 포함되기 때문에  # 바이너리 파일을 전송하면 파일 형식에 대한 언급 없이도 자동으로 JPG, txt 파일을 만들 수 있다.  # 또한 소켓은 bytes를 전달하기 때문에 별다른 디코딩 과정없이 파일에 작성해도 된다.'''#    # Exception Handling except OSError as e : print('socket error: ', e) except Exception as e :  print('Exception at listening:'.format(e)) else:  print('client closed', client\_addr) finally :  conn.close()    if \_\_name\_\_ == '\_\_main\_\_' :  echo\_server(50011) # 포트번호 |

10.1.2. 클라이언트 코드

|  |
| --- |
| picam-client.py |
| import socket import serial import time import picamera    port = "/dev/ttyACM0" #아두이노의 포트  ser = serial.Serial(port, 9600) ser.flushInput()    src = "/home/pi/tt/" # 저장되는 디렉터리    dte = time.localtime() Year = dte.tm\_year  Mon = dte.tm\_mon  Day = dte.tm\_mday  Hour = dte.tm\_hour  Min = dte.tm\_min  Sec = dte.tm\_sec    time = str(Year) + "-" + str(Mon) + "-" + str(Day) + " " + str(Hour) + ":" + str(Min) # 메시지에 보낼 시간 정보  time2 = str(Year) + "\_" + str(Mon) + "\_" + str(Day) + "\_" + str(Hour) + "\_" + str(Min) #파일명에 쓰일 시간  def echo\_client(server\_addr) :  sock = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) sock.connect(server\_addr) |

|  |
| --- |
| print('connected: ', sock.getpeername()) while True :    ser.write("in".encode('utf-8')) distance = float(ser.readline()) print(distance) if distance < 20 and distance >4 :  print('cam') # 사진 촬영 시 cam이라고 print with picamera.PiCamera() as camera :  #context관리를 자동으로 해주는 with 사용 camera.start\_preview(fullscreen = False, window = (100, 20, 640, 480))  camera.capture('/home/pi/tt/%s.jpg' % time2) # 라즈베리파이의 tt폴더에 저장 camera.stop\_preview() capture\_file\_name = src + str(time2) +  ".jpg" # 어떤 파일 형태로    file = open(capture\_file\_name, "rb") # 전달할 사진을 바이너리 형식으로 읽음 data = file.read(8192) #1KB를 읽음 while (data) :  sock.send(data) # 데이터 전송 data = file.read(8192) #1KB씩 읽음 print("send finished") # 서버에 전송 되었을 때 send finished 라고 전송        sock.close() |

if \_\_name\_\_ == '\_\_main\_\_':

echo\_client(('192.168.0.63', 50011))

10.1.3. 안드로이드 코드

|  |
| --- |
| AndroidManifest.xml |
| < ? xml version = "1.0" encoding = "utf-8" ? >  <manifest xmlns : android =  "http://schemas.android.com/apk/res/android" package = "com.example.txt"> <uses - permission android : name =  "android.permission.INTERNET" / >  <uses - permission android : name =  "android.permission.ACCESS\_NETWORK\_STATE" / >  <uses - permission android : name =  "android.permission.ACCESS\_WIFI\_STATE" / >    <application  android : allowBackup = "true" android : icon = "@mipmap/ic\_launcher" android : label = "@string/app\_name" android : roundIcon = "@mipmap/ic\_launcher\_round" android : networkSecurityConfig =  "@xml/network\_security\_config" android : supportsRtl = "true" android : theme = "@style/AppTheme">  <activity android : name = ".ResentActivity">< / activity>  <activity android : name = ".LoginActivity" / >  <activity android : name = ".Main2Activity" / >  <activity android : name = ".StateActivity" / >  <activity android : name = ".MainActivity">  <intent - filter>  <action android : name = "android.intent.action.MAIN" / > |

|  |
| --- |
| <category android : name =  "android.intent.category.LAUNCHER" / >  < / intent - filter>  < / activity>  < / application>    < / manifest> |
| LoginActivity.java |
| //선택한 호수의 우편함에 접근하기 위해 비밀번호를 입력하는 화면이다.  package com.example.txt;    import androidx.appcompat.app.AppCompatActivity;    import android.content.Intent; import android.os.Bundle; import android.view.View; import android.widget.Button; import android.widget.EditText; import android.widget.TextView;    import org.w3c.dom.Text;    public class LoginActivity extends AppCompatActivity {    @Override protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState); setContentView(R.layout.activity\_login); |

|  |
| --- |
| //비밀번호를 입력하는 editText 호출  final EditText editText =  (EditText)findViewById(R.id.edittext);  //입력된 string과 비밀번호를 비교하는 login버튼 호출  Button button = (Button)findViewById(R.id.button); final TextView textView =  (TextView)findViewById(R.id.textview);    //버튼이 클릭되었을 때 입력된 string과 비밀번호를 비교  button.setOnClickListener(new  View.OnClickListener() {  @Override  public void onClick(View view) { final String userPassword = editText.getText().toString();  //비밀번호가 일치하면 아래  텍스트로 일치함을 보여주며 우편함 기록  화면(Main2Activity)으로 이동 if (userPassword.equals("1234")) { textView.setText("일치");  Intent intent = new  Intent(getApplicationContext(), Main2Activity.class);  startActivity(intent);  }  //비밀번호가 일치하지 않으면 아래  다시 입력을 요청하는 텍스트 띄움  else {  textView.setText("비밀번호를 |

|  |
| --- |
| 다시 입력하세요");  }  }  });    }  } |
| Main2Activity.java |
| //우편함 목록 확인 화면이다.  package com.example.txt;    import androidx.appcompat.app.AppCompatActivity;    import android.app.Activity; import android.content.Intent; import android.os.Bundle; import android.view.View; import android.widget.AdapterView; import android.widget.ArrayAdapter; import android.widget.Button; import android.widget.ImageButton; import android.widget.ImageView; import android.widget.ListView;    import java.io.ByteArrayOutputStream; import java.io.IOException; import java.io.InputStream; import java.util.ArrayList; import java.io.\*;    public class Main2Activity extends AppCompatActivity { |

|  |
| --- |
| //기록을 배열에 넣기위해 선언  ArrayList<String> mDatas= new ArrayList<String>();  ListView listview;    @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main2);    String str = readTxt();  String[] array = str.split("\n");  for (int i=0;i<array.length;i++) { array[i] = array[i].replaceFirst("\_", "년 "); array[i] = array[i].replaceFirst("\_", "월 "); array[i] = array[i].replaceFirst("\_", "일 "); array[i] = array[i].replaceFirst("\_", "시 "); array[i] = array[i].replaceFirst("\_", "분 ");    mDatas.add(array[i]);  }    //기록을 차례대로 넣은 리스트를 보여주기위한 뷰 호출  ArrayAdapter adapter= new ArrayAdapter(this, android.R.layout.simple\_expandable\_list\_item\_1, mDatas); listview= (ListView)findViewById(R.id.listview); listview.setAdapter(adapter); listview.setOnItemClickListener(listener);  Button statebutton = (Button) |

findViewById(R.id.statebutton);

statebutton.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View view) {

Intent intent = new

Intent(getApplicationContext(), ResentActivity.class); startActivity(intent);

}

});

}

AdapterView.OnItemClickListener listener= new

AdapterView.OnItemClickListener() {

@Override public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

Intent intent = new Intent(getApplicationContext(), StateActivity.class); intent.putExtra("imagename",mDatas.get(position)); startActivity(intent);

}

};

private String readTxt() {

String data = null;

InputStream inputStream =

getResources().openRawResource(R.raw.filename);

ByteArrayOutputStream byteArrayOutputStream = new

ByteArrayOutputStream();

|  |
| --- |
| int i; try { i = inputStream.read(); while (i != -1) { byteArrayOutputStream.write(i); i = inputStream.read();  }    data = new  String(byteArrayOutputStream.toByteArray(), "MS949"); inputStream.close(); } catch (IOException e) { e.printStackTrace(); } return data;  }  } |
| MainActivity.java |
| //Smart Mail Box라는 타이틀을 보여주며 호수를 선택하는 첫 화면이다.  package com.example.txt;    import androidx.appcompat.app.AppCompatActivity;    import android.app.Activity; import android.content.Context; import android.content.Intent; import android.os.Bundle; import android.view.View; import android.widget.Button; |

|  |
| --- |
| import android.widget.TextView;    import java.io.ByteArrayOutputStream; import java.io.IOException; import java.io.InputStream;    public class MainActivity extends AppCompatActivity {    @Override protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);    //호수를 선택하면 해당 호수 우편함에 접근하기 위한 로그인 액티비티로 넘어가는 버튼호출  Button loginbutton =  (Button)findViewById(R.id.loginbutton);    loginbutton.setOnClickListener(new  View.OnClickListener() {  @Override  public void onClick(View view) {  Intent intent = new  Intent(getApplicationContext(), LoginActivity.class);  startActivity(intent);  }  });  }  } |
| ResentActivity.java |

package com.example.txt;

import androidx.appcompat.app.AppCompatActivity;

import android.app.Activity; import android.content.Intent; import android.graphics.Bitmap; import android.graphics.BitmapFactory; import android.os.AsyncTask; import android.os.Bundle; import android.util.Log; import android.view.View; import android.widget.Button; import android.widget.ImageView; import android.widget.ListView; import android.widget.TextView;

import java.io.ByteArrayOutputStream; import java.io.IOException; import java.io.InputStream; import java.net.HttpURLConnection; import java.net.URL; import java.util.ArrayList;

public class ResentActivity extends AppCompatActivity {

ArrayList<String> mDatas= new ArrayList<String>();

ListView listview;

String str = readTxt();

String[] value = str.split("\n");

int valueLength =value.length;

String maxValue = value[0];

MyMethod1 mymethod1 = new MyMethod1();

String filename = mymethod1.findMaxValue(value, valueLength, maxValue);

@Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_resent);

String str = readTxt();

String[] array = str.split("\n");

new DownloadImageTask((ImageView)

findViewById(R.id.imageView5))

.execute("http://10.0.2.2/"+filename+".jpg");

Button listbutton = (Button) findViewById(R.id.listbutton);

listbutton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent intent = new

Intent(getApplicationContext(),Main2Activity.class); startActivity(intent);

}

});

}

private class DownloadImageTask extends AsyncTask<String,

Void, Bitmap> {

ImageView bmImage;

public DownloadImageTask(ImageView bmImage) { this.bmImage = bmImage;

}

protected Bitmap doInBackground(String... urls) {

String urldisplay = urls[0]; Bitmap mIcon11 = null; try {

InputStream in = new

java.net.URL(urldisplay).openStream(); mIcon11 = BitmapFactory.decodeStream(in);

} catch (Exception e) {

Log.e("Error", e.getMessage()); e.printStackTrace(); } return mIcon11;

}

protected void onPostExecute(Bitmap result) { bmImage.setImageBitmap(result);

}

}

private String readTxt() {

String data = null;

InputStream inputStream =

getResources().openRawResource(R.raw.filename); ByteArrayOutputStream byteArrayOutputStream = new

ByteArrayOutputStream();

int i; try { i = inputStream.read(); while (i != -1) { byteArrayOutputStream.write(i); i = inputStream.read();

}

data = new

String(byteArrayOutputStream.toByteArray(), "MS949"); inputStream.close(); } catch (IOException e) { e.printStackTrace(); } return data;

}

}

class MyMethod1{

// 최대값 구하기

String findMaxValue(String[] value,int valueLength, String maxValue){ for(int i=0; i<valueLength; i++){ if(value[i].compareTo(maxValue)==1){ maxValue = value[i];

}

}

|  |
| --- |
| return maxValue;  }  } |
| StateActivity.java |
| package com.example.txt;    import androidx.appcompat.app.AppCompatActivity;    import android.content.Intent; import android.content.res.AssetManager; import android.graphics.Bitmap; import android.graphics.BitmapFactory; import android.os.AsyncTask; import android.os.Bundle; import android.util.Log; import android.view.View; import android.widget.Button; import android.widget.ImageView; import android.widget.TextView;    import java.io.ByteArrayOutputStream; import java.io.IOException; import java.io.InputStream; import java.net.HttpURLConnection; import java.net.URL;    public class StateActivity extends AppCompatActivity {    @Override protected void onCreate(Bundle savedInstanceState) { |

super.onCreate(savedInstanceState); setContentView(R.layout.activity\_state);

Intent intent = getIntent();

String image =

intent.getExtras().getString("imagename");

TextView textView =

(TextView)findViewById(R.id.imageN);

image = image.replace("년 ", "\_"); image = image.replace("월 ", "\_"); image = image.replace("일 ", "\_"); image = image.replace("시 ", "\_"); image = image.replace("분 ", "\_"); image = image.replace(" ", "");

new

DownloadImageTask((ImageView)findViewById(R.id.imageView5))

.execute("http://10.0.2.2/d" + image +

".jpg");

// String file = "@drawable/textimage";

// String packName =

this.getPackageName();

// int resID =

getResources().getIdentifier(file, "drawable",packName);

//

// ImageView imageView =

(ImageView)findViewById(R.id.imageView5);

// imageView.setImageResource(resID); Button listbutton =

(Button)findViewById(R.id.listbutton);

Button statebutton =

(Button)findViewById(R.id.statebutton);

textView.setText(image);

listbutton.setOnClickListener(new

View.OnClickListener() {

@Override

public void onClick(View view) {

Intent intent = new

Intent(getApplicationContext(), Main2Activity.class);

startActivity(intent);

}

});

statebutton.setOnClickListener(new

View.OnClickListener() {

@Override

public void onClick(View view) {

Intent intent = new

Intent(getApplicationContext(), ResentActivity.class);

startActivity(intent);

}

});

}

private class DownloadImageTask extends

AsyncTask<String, Void, Bitmap> {

ImageView bmImage;

|  |
| --- |
| public DownloadImageTask(ImageView bmImage) {  this.bmImage = bmImage;  }    protected Bitmap doInBackground(String... urls) {  String urldisplay = urls[0]; Bitmap mIcon11 = null;  try {  InputStream in = new  java.net.URL(urldisplay).openStream();  mIcon11 =  BitmapFactory.decodeStream(in);  }  catch (Exception e) {  Log.e("Error", e.getMessage()); e.printStackTrace();  }  return mIcon11;  }    protected void onPostExecute(Bitmap result) { bmImage.setImageBitmap(result);  }  }    } |
| activity\_login.xml |
| < ? xml version = "1.0" encoding = "utf-8" ? >  <androidx.constraintlayout.widget.ConstraintLayout xmlns : android = "http://schemas.android.com/apk/res/android" |

xmlns : app = "http://schemas.android.com/apk/res-auto" xmlns : tools = "http://schemas.android.com/tools"

android : layout\_width = "match\_parent" android : layout\_height = "match\_parent" tools : context = ".LoginActivity">

<View

android : id = "@+id/view4" android : layout\_width = "422dp" android : layout\_height = "739dp" android : layout\_marginTop = "10dp" android : background = "#D8DEE8" app : layout\_constraintStart\_toStartOf = "parent" app : layout\_constraintTop\_toBottomOf =

"@android:id/background" / >

<TextView

android : id = "@+id/textView" android : layout\_width = "wrap\_content" android : layout\_height = "wrap\_content" android : layout\_marginStart = "8dp" android : layout\_marginTop = "100dp" android : layout\_marginEnd = "8dp" android : text = "Smart Mail Box" android : textAppearance =

"@style/Base.TextAppearance.AppCompat.Title" android : textSize = "50sp" app : layout\_constraintEnd\_toEndOf = "parent" app : layout\_constraintStart\_toStartOf = "parent" app : layout\_constraintTop\_toTopOf = "parent" / >

|  |
| --- |
| <EditText android : id = "@+id/edittext"  android : layout\_width = "wrap\_content" android : layout\_height = "wrap\_content" android : layout\_marginStart = "8dp" android : layout\_marginTop = "380dp" android : layout\_marginEnd = "8dp" android : ems = "10" android : hint = "password" android : inputType = "textPersonName" app : layout\_constraintEnd\_toEndOf = "parent" app : layout\_constraintStart\_toStartOf = "parent" app : layout\_constraintTop\_toTopOf = "parent" / >    <TextView  android : id = "@+id/textView3" android : layout\_width = "wrap\_content" android : layout\_height = "wrap\_content" android : layout\_marginStart = "8dp" android : layout\_marginTop = "270dp" android : layout\_marginEnd = "8dp" android : text = "비밀번호를 입력하세요"  android : textSize = "30sp" android : textStyle = "bold" app : layout\_constraintEnd\_toEndOf = "parent" app : layout\_constraintStart\_toStartOf = "parent" app : layout\_constraintTop\_toTopOf = "parent" / >    <TextView  android : id = "@+id/textView4" android : layout\_width = "wrap\_content" |

android : layout\_height = "wrap\_content" android : layout\_marginStart = "8dp"

android : layout\_marginTop = "220dp" android : layout\_marginEnd = "8dp" android : text = "101호 우편함" android : textSize = "30sp" android : textStyle = "bold" app : layout\_constraintEnd\_toEndOf = "parent" app : layout\_constraintStart\_toStartOf = "parent" app : layout\_constraintTop\_toTopOf = "parent" / >

<Button

android : id = "@+id/button" android : layout\_width = "wrap\_content" android : layout\_height = "wrap\_content" android : layout\_marginStart = "8dp" android : layout\_marginTop = "120dp" android : layout\_marginEnd = "8dp" android : text = "입장" app : layout\_constraintEnd\_toEndOf = "parent" app : layout\_constraintStart\_toStartOf = "parent" app : layout\_constraintTop\_toBottomOf =

"@+id/textView3" / >

<TextView

android : id = "@+id/textview" android : layout\_width = "wrap\_content" android : layout\_height = "wrap\_content" android : layout\_marginStart = "8dp" android : layout\_marginTop = "80dp" android : layout\_marginEnd = "8dp"

|  |
| --- |
| app : layout\_constraintEnd\_toEndOf = "parent" app : layout\_constraintHorizontal\_bias = "0.498"  app : layout\_constraintStart\_toStartOf = "parent" app : layout\_constraintTop\_toBottomOf = "@id/edittext" / >  < / androidx.constraintlayout.widget.ConstraintLayout> |
| activity\_main.xml |
| < ? xml version = "1.0" encoding = "utf-8" ? >  <androidx.constraintlayout.widget.ConstraintLayout xmlns : android = "http://schemas.android.com/apk/res/android" xmlns : app = "http://schemas.android.com/apk/res-auto" xmlns : tools = "http://schemas.android.com/tools" android : layout\_width = "match\_parent" android : layout\_height = "match\_parent" tools : context = ".MainActivity">    <View  android : id = "@+id/view" android : layout\_width = "422dp" android : layout\_height = "739dp" android : layout\_marginTop = "10dp" android : background = "#D8DEE8" app : layout\_constraintStart\_toStartOf = "parent" app : layout\_constraintTop\_toBottomOf =  "@android:id/background" / >    <Button  android : id = "@+id/testbutton" android : layout\_width = "200dp" android : layout\_height = "60dp" android : layout\_marginStart = "105dp" |

android : layout\_marginTop = "440dp" android : text = "103호"

android : textColor = "#706C6C" android : textSize = "26sp" app : layout\_constraintStart\_toStartOf = "parent" app : layout\_constraintTop\_toTopOf = "parent" / >

<Button

android : id = "@+id/loginbutton" android : layout\_width = "200dp" android : layout\_height = "60dp" android : layout\_marginStart = "105dp" android : layout\_marginTop = "280dp" android : text = "101호" android : textColor = "#706C6C" android : textSize = "26sp" app : layout\_constraintStart\_toStartOf = "parent" app : layout\_constraintTop\_toTopOf = "parent" / >

<Button

android : id = "@+id/listbutton5" android : layout\_width = "200dp" android : layout\_height = "60dp" android : layout\_marginStart = "105dp" android : layout\_marginTop = "360dp" android : text = "102호" android : textColor = "#706C6C" android : textSize = "26sp" app : layout\_constraintStart\_toStartOf = "parent" app : layout\_constraintTop\_toTopOf = "parent" / >

|  |
| --- |
| <TextView android : id = "@+id/textView2"  android : layout\_width = "wrap\_content" android : layout\_height = "wrap\_content" android : layout\_marginStart = "85dp" android : layout\_marginTop = "200dp" android : text = "호수를 선택하세요" android : textSize = "30sp" android : textStyle = "bold" app : layout\_constraintStart\_toStartOf = "parent" app : layout\_constraintTop\_toTopOf = "parent" / >    <TextView  android : id = "@+id/textView6" android : layout\_width = "wrap\_content" android : layout\_height = "wrap\_content" android : layout\_marginStart = "8dp" android : layout\_marginTop = "100dp" android : layout\_marginEnd = "8dp" android : text = "Smart Mail Box" android : textAppearance =  "@style/Base.TextAppearance.AppCompat.Title" android : textSize = "50sp" app : layout\_constraintEnd\_toEndOf = "parent" app : layout\_constraintStart\_toStartOf = "parent" app : layout\_constraintTop\_toTopOf = "parent" / >    < / androidx.constraintlayout.widget.ConstraintLayout> |
| activity\_main2.xml |
| <?xml version="1.0" encoding="utf-8"?>  <androidx.constraintlayout.widget.ConstraintLayout |

xmlns:android="http://schemas.android.com/apk/res/android" xmlns:app="http://schemas.android.com/apk/res-auto" xmlns:tools="http://schemas.android.com/tools" android:layout\_width="match\_parent" android:layout\_height="match\_parent" tools:context=".Main2Activity">

<Button android:id="@+id/listbutton" android:layout\_width="184dp" android:layout\_height="60dp" android:layout\_marginStart="10dp" android:layout\_marginTop="10dp" android:text="목록" android:textColor="#706C6C" android:textSize="26sp" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toTopOf="parent" />

<Button android:id="@+id/statebutton" android:layout\_width="184dp" android:layout\_height="60dp" android:layout\_marginStart="210dp" android:layout\_marginTop="10dp" android:text="최근 기록" android:textColor="#878484" android:textSize="26sp" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toTopOf="parent" />

|  |
| --- |
| <View android:id="@+id/view2" android:layout\_width="417dp" android:layout\_height="700dp" android:layout\_marginTop="10dp" android:background="#D8DEE8" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/listbutton" />    <Button android:id="@+id/button4" android:layout\_width="129dp" android:layout\_height="40dp" android:layout\_marginStart="40dp" android:layout\_marginTop="17dp" android:background="@drawable/rectangle" android:text="Date" android:textColor="#F5F2F2" android:textSize="18sp" app:layout\_constraintStart\_toStartOf="@+id/listbutton" app:layout\_constraintTop\_toBottomOf="@+id/listbutton" />    <Button android:id="@+id/button5" android:layout\_width="106dp" android:layout\_height="40dp" android:layout\_marginStart="205dp" android:layout\_marginTop="17dp" android:background="@drawable/rectangle" |

|  |
| --- |
| android:text="Time" android:textColor="#F5F2F2" android:textSize="18sp" app:layout\_constraintStart\_toStartOf="@+id/listbutton" app:layout\_constraintTop\_toBottomOf="@+id/listbutton" />    <ImageView android:id="@+id/imageView" android:layout\_width="174dp" android:layout\_height="4dp" app:layout\_constraintEnd\_toEndOf="@+id/listbutton" app:layout\_constraintStart\_toStartOf="@+id/listbutton" app:layout\_constraintTop\_toBottomOf="@+id/listbutton" app:srcCompat="@android:color/holo\_orange\_dark" />    <ListView android:id="@+id/listview" android:layout\_width="368dp" android:layout\_height="576dp" android:layout\_marginTop="5dp" android:layout\_marginEnd="8dp" app:layout\_constraintEnd\_toEndOf="@+id/view2" app:layout\_constraintStart\_toStartOf="@+id/view2" app:layout\_constraintTop\_toBottomOf="@+id/button5" />      </androidx.constraintlayout.widget.ConstraintLayout> |
| activity\_resent.xml |
| <?xml version="1.0" encoding="utf-8"?> |

|  |
| --- |
| <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android" xmlns:app="http://schemas.android.com/apk/res-auto" xmlns:tools="http://schemas.android.com/tools" android:layout\_width="match\_parent" android:layout\_height="match\_parent" tools:context=".ResentActivity">    <Button android:id="@+id/listbutton" android:layout\_width="184dp" android:layout\_height="60dp" android:layout\_marginStart="10dp" android:layout\_marginTop="10dp" android:text="목록" android:textColor="#706C6C" android:textSize="26sp" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toTopOf="parent" />    <Button android:id="@+id/statebutton" android:layout\_width="184dp" android:layout\_height="60dp" android:layout\_marginStart="210dp" android:layout\_marginTop="10dp" android:text="최근 기록" android:textColor="#878484" android:textSize="26sp" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toTopOf="parent" /> |

|  |
| --- |
| <View android:id="@+id/view2" android:layout\_width="417dp" android:layout\_height="631dp" android:layout\_marginTop="10dp" android:background="#D8DEE8" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toBottomOf="@+id/listbutton" />    <ImageView android:id="@+id/imageView" android:layout\_width="174dp" android:layout\_height="4dp" app:layout\_constraintEnd\_toEndOf="@+id/statebutton" app:layout\_constraintStart\_toStartOf="@+id/statebutton"  app:layout\_constraintTop\_toBottomOf="@+id/statebutton" app:srcCompat="@android:color/holo\_orange\_dark" />    <ImageView android:id="@+id/imageView5" android:layout\_width="367dp" android:layout\_height="294dp" android:layout\_marginTop="15dp" android:layout\_marginEnd="8dp" android:scaleType="fitXY" app:layout\_constraintEnd\_toEndOf="@+id/view2" app:layout\_constraintStart\_toStartOf="@+id/view2" app:layout\_constraintTop\_toTopOf="@+id/view2" |

|  |
| --- |
| />    <Button android:id="@+id/button1" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginTop="8dp" android:layout\_marginEnd="8dp" android:layout\_marginBottom="8dp" android:text="Button" app:layout\_constraintBottom\_toBottomOf="@+id/view2" app:layout\_constraintEnd\_toEndOf="@+id/view2" app:layout\_constraintStart\_toStartOf="@+id/view2"  app:layout\_constraintTop\_toBottomOf="@+id/imageView5" />    </androidx.constraintlayout.widget.ConstraintLayout> |
| activity\_state.xml |
| < ? xml version = "1.0" encoding = "utf-8" ? >  <androidx.constraintlayout.widget.ConstraintLayout xmlns : android = "http://schemas.android.com/apk/res/android" xmlns : app = "http://schemas.android.com/apk/res-auto" xmlns : tools = "http://schemas.android.com/tools" android : layout\_width = "match\_parent" android : layout\_height = "match\_parent" tools : context = ".StateActivity">    <Button  android : id = "@+id/listbutton" android : layout\_width = "184dp" android : layout\_height = "60dp" |

|  |
| --- |
| android : layout\_marginStart = "10dp" android : layout\_marginTop = "10dp" android : text = "목록" android : textColor = "#706C6C" android : textSize = "26sp" app : layout\_constraintStart\_toStartOf = "parent" app : layout\_constraintTop\_toTopOf = "parent" / >    <Button  android : id = "@+id/statebutton" android : layout\_width = "184dp" android : layout\_height = "60dp" android : layout\_marginStart = "210dp" android : layout\_marginTop = "10dp" android : text = "최근 기록" android : textColor = "#878484" android : textSize = "26sp" app : layout\_constraintStart\_toStartOf = "parent" app : layout\_constraintTop\_toTopOf = "parent" / >    <View  android : id = "@+id/view2" android : layout\_width = "417dp" android : layout\_height = "631dp" android : layout\_marginTop = "10dp" android : background = "#D8DEE8" app : layout\_constraintStart\_toStartOf = "parent" app : layout\_constraintTop\_toBottomOf =  "@+id/listbutton" / >    <ImageView |

|  |
| --- |
| android : id = "@+id/imageView5" android : layout\_width = "367dp" android : layout\_height = "294dp" android : layout\_marginTop = "15dp" android : layout\_marginEnd = "8dp" android : scaleType = "fitXY" app : layout\_constraintEnd\_toEndOf = "@+id/view2" app : layout\_constraintStart\_toStartOf = "@+id/view2" app : layout\_constraintTop\_toTopOf = "@+id/view2"  / >    <TextView  android : id = "@+id/imageN" android : layout\_width = "wrap\_content" android : layout\_height = "wrap\_content" android : layout\_marginStart = "8dp" android : layout\_marginTop = "8dp" android : layout\_marginBottom = "8dp" app : layout\_constraintBottom\_toBottomOf =  "@+id/view2" app : layout\_constraintEnd\_toEndOf = "@+id/view2" app : layout\_constraintStart\_toStartOf = "@+id/view2" app : layout\_constraintTop\_toBottomOf =  "@+id/imageView5" / >    < / androidx.constraintlayout.widget.ConstraintLayout> |
| network\_security\_config.xml |
| < ? xml version = "1.0" encoding = "utf-8" ? >  <network - security - config>  <!--Set application - wide security config using base - config tag.--> |

<base - config cleartextTrafficPermitted = "true" / >

< / network - security - config>

10.1.4. 아두이노 코드

|  |
| --- |
| UsLED.ino |
| int Trig = 13; int Echo = 12; int led\_R = 7; int led\_G = 6; int led\_B = 5;    void setup() {  Serial.begin(9600); pinMode(Trig, OUTPUT); pinMode(Echo, INPUT); pinMode(led\_R, OUTPUT); pinMode(led\_G, OUTPUT); pinMode(led\_B, OUTPUT);  } void loop() {  digitalWrite(Trig, LOW); digitalWrite(Echo, LOW); delayMicroseconds(2); digitalWrite(Trig, HIGH); delayMicroseconds(10); digitalWrite(Trig, LOW);    unsigned long duration = pulseIn(Echo, HIGH); float distance = ((float)(340 \* duration) / 10000) / 2;    Serial.println(int(distance)); |

|  |  |  |  |
| --- | --- | --- | --- |
|  | } | if (distance < 20) {  digitalWrite(led\_R, HIGH); digitalWrite(led\_G, HIGH); digitalWrite(led\_B, HIGH);  } else {  digitalWrite(led\_R, LOW); digitalWrite(led\_G, LOW); digitalWrite(led\_B, LOW);  }  delay(1000); | |
|  | |  |  |

10.2. 주요 내용 설명 10.2.1. 소프트웨어

|  |  |  |
| --- | --- | --- |
| 기능 | 설명 | 사진 (스크린 캡쳐) |
| LOG IN | 안드로이드 어플리케이션 을 통해 사용자의 세대원 이 아닌 다른 사용자가 확 인할 수 있는 문제를 해결 하고자 로그인 액티비티를 적용하였다. 각 세대 별로 고유 ID가 부여되며, 본인 의 건물(아파트, 빌라, 오 피스텔 등)을 선택 후 동/ 호수를 선택하여 로그인을 할 수 있도록 하였다. | [그림 10-1] 안드로  이드 어플리케이션 로그인 |
| 기록 확인 | 우편함에 우편물이 들어올 때 초음파 센서가 감지하 여, LED가 ON되고 파이캠 으로 촬영을 한다. 서버는 소켓통신을 통해 라즈베리 파이(클라이언트)에서 사진 을 전송받는데, 이때 날짜 와 시간이 기록되고 사용 자는 안드로이드 어플리케 이션에서 확인할 수 있도 록 하였다. | [그림 10-2] 우편함 사진 찍은 기록 확  인 |
| 최근 기록 | 우편함에 들어오는 모든 우편물 중 가장 최근에 들 어온 우편물 사진을 안드 로이드 어플리케이션을 통 해 확인할 수 있게 하였  다. | [그림 10-3] 우편함 사진 찍은 최근 기 록 확인 |

[표 10-1] 소프트웨어 주요 기능

10.2.2. 하드웨어

|  |  |  |
| --- | --- | --- |
| 부품 | 설명 | 사진 |
| 초음파 센 서 | 우편함 안쪽 끝 면에 부착 하여 우편물이 들어왔을 때 일정 거리 내 에 들어왔는지 감지한다. | [그림 10-3] 아두이노 초음파 센서 |
| LED Actuator | 우편함은 닫혀 있어 촬영을 할 때 다소 어 두워 잘 보이 지 않을 수 있 으므로 주위를 환하게 해주기 위해 부착하였  다. | [그림 10-4] 아두이노 LED Actuator |
| RPI NOIR  CAMERA  BOARD | 우편함 내부 전체의 모습을 볼 수 있도록 우편함 내부에 부착하였다. | [그림 10-5] RPI NOIR CAMERA  BOARD |

[표 10-2] 하드웨어 주요 기능

### 10.3. 개발 환경 상세 설명

|  |  |  |
| --- | --- | --- |
| 구분 | 항목 | 적용 내역 |
| OS | Android(oreo 8.1) | 안드로이드 기반 스마트폰을 위한 어플리케이션 개발 |
| Raspbian | 라즈베리파이에 포팅 공식 홈페이지에서 다운받아 포팅 |
| Window 10 | 서버 개발 |
| 개발환  경(IDE) | Android Studio | 안드로이드 어플리케이션 개발을 위한 통합 개발 환경 구축 공식 홈페이지에서 다운받아 설치 |
| Pycharm | 서버 개발을 위한 환경 구축 |
| Arduino Sketch | 아두이노 센서 및 Actuator 제어를 위한 개발 환경 구축 공식 홈페이지에서 다운받아 설치 |
| 개발도 구 | Android Studio | 안드로이드 어플리케이션 소스 코 드 컴파일 및 디버깅. Android  Studio는 Android 앱 개발을 위한 공식 통합 개발 환경(IDE)이며,  IntelliJ IDEA 를 기반으로 한다.  IntelliJ의 강력한 코드 편집기와 개 발자 도구 외에도, Android Studio 는 Android 앱을 빌드할 때 생산  성을 높여주는 다양한 기능을 제공  한다. |
| 개발언 어 | C | Arduino Sketch(IDE)를 조작 |
| Python | 서버 및 클라이언트 소스 코드 컴  파일 및 디버깅. 객체 지향형 스크  립트 언어의 하나로서 문법 구조가 간단 명확하고, 언어 코어가 상대 적으로 작은 대신 많은 시스템 호 |
|  |  | 출과 풍부한 라이브러리 모듈을 제 공한다. 또한 시스템 언어와 스크  립트 언어의 중간 위치 성격을 띄 어서 C, C++을 사용하여 새로운  모듈을 쉽게 만들 수 있는 등 다른 언어와 잘 융합되는 특징이 있다. |
| Java | 안드로이드 어플리케이션 소스 코  드 |

[표 10-3] 개발 환경 상세 설명

### 10.4. 참고 문헌

|  |  |
| --- | --- |
| 구분 | URL 혹은 문서 제목 |
| 파이캠 | <https://blog.naver.com/seharhc/221316225015> |
| <http://www.dreamy.pe.kr/zbxe/CodeClip/3769345> |
| 안드로이드  스튜디오 | [https://m.blog.naver.com/PostView.nhn?blogId=beaq on&logNo=221076390196&proxyReferer=https%3A](https://m.blog.naver.com/PostView.nhn?blogId=beaqon&logNo=221076390196&proxyReferer=https%3A%2F%2Fwww.google.com%2F)  [%2F%2Fwww.google.com%2F](https://m.blog.naver.com/PostView.nhn?blogId=beaqon&logNo=221076390196&proxyReferer=https%3A%2F%2Fwww.google.com%2F) |

[표 10-4] 참고 문헌

## 10. 느낀 점

|  |  |
| --- | --- |
| 팀원 | 느낀 점 |
| 최동욱 | 처음 제안서 발표 때 별로 좋은 평가를 받지 못해서 주제 를 바꾸어야 했는데, 마침 교수님께서 IOT 과제를 내주셔 서 방향성도 잡고 실제로 프로젝트 할 때도 많은 도움이 되었던 것 같다. 힘들었던 점은, 개인적으로는 자료들에 나 와있는 코드나 방법들을 내 상황에 맞게 변형하여 사용하 는 것이 힘들었다. 또한 팀 프로젝트이다 보니 모두가 짰던 코드를 병합하는 과정에서도 많은 에러가 나서 에러를 잡 는 데에만 오랜 시간이 걸렸다. 하지만 이런 힘든 과정을 해결해 가는 과정에서 많은 노하우를 얻어서 좋은 경험이 되었다고 생각한다. |
| 이정민 | 팀프로젝트를 시작할 때 소프웨어적인 문제가 많을 것이라 예상했다. 물론 소프트웨어에서도 어려움은 있었으나 예상 과 달리 접촉 불량과 전원부족 등 하드웨어적인 문제도 많 이 발생했다. 이를 통해 소프트웨어와 하드웨어 양쪽 측면 모두 고려해서 문제를 해결해야 된다는 것을 배웠다. 또한 팀프로젝트를 하는 동안 팀원들과 문제 발생시 해결 하는 방법을 배우는 좋은 기회였다고 생각한다. |
| 정희원 | 캡스톤 과목을 수강함에 앞서 걱정과 기대가 많았었다. 최 종 프로젝트를 진행하면서 팀원들과 아이디어 구상부터 제 작 진행까지 여러가지 시행착오를 거치며 많은 점들을 느 낄 수 있었다. 우선, 아이디어 상 구성과 실제 프로젝트 진 행에 많은 차이가 있음을 알게 되었고, 아이디어를 그대로 구현하기 위해 많은 노력이 필요함을 알 수 있었다. 또한, 프로젝트를 진행함에 있어 강의 때 이론적으로 배운 소켓 통신을 비롯한 여러 내용들을 실제 프로젝트에 적용함에 따라 수업 때 잘못 알고 넘어간 부분들을 다시 되짚어 볼 수 있는 좋은 계기가 되었다고 생각한다. |
| 이경현 | 처음 제안서 발표를 했던 주제가 생각보다 구현하기 어려 |
|  | 울 것 같고 고려해야 할 사항이 많았기 때문에 팀원들끼리 회의를 통해서 주제를 바꾸기로 하였다. 아이디어 회의가 생각보다 길어지고 바뀐 주제에 알맞은 하드웨어를 구하는 데에 시간을 다 쏟았던 점이 다소 아쉽기도 하다. 그렇지만 남은 기간 동안 팀원들끼리 서로 독려하며 프로젝트를 완 성하는데 애썼기 때문에 결과물을 완성시킬 수 있었다. 작 년 2학년 2학기 “통신 시스템 프로그래밍 및 실습” 과목에 서 팀프로젝트를 처음 진행하였으나 간단하게 C언어를 사 용하여 하나의 프로그램을 만드는 정도였는데, 이번 “인터 넷 응용”을 수강하며 하드웨어 파트와 소프트웨어 파트 모 두 다 생각하며 프로젝트를 진행해야 하니 버겁기도 했지 만 좋은 경험이 된 것 같다. |
| 최소원 | 프로젝트 초기에 라즈베리파이를 구성하면서 파이캠을 설 치하는데 어려움을 겪었다. 라즈베리파이 내부소프트웨어 문제였는데 파이캠이 인식되지 않는다는 오류였기에 꽤 오 래 동안 당연히 하드웨어 상 문제일 거라 판단하고 문제 해결에 힘썼지만 소프트웨어 문제인 걸 알고나서 하드웨어 와 소프트웨어 둘 다 신경 써야 한다는 것을 알게 되었다. 이후엔 안드로이드를 다루게 되었는데 한번도 해본적이 없 어서 새롭게 공부하게 되었다. 급하게 만들어야 했기에 안 드로이드 스튜디오를 설치하고 구글링으로 알아내서 초기 설정하였고 당장 만들어야 하는 기능만 구축해보려 했는데 기초지식이 없었던 탓인지 오류가 많이 발생하게 되었다. 액티비티와 layout파일 간의 관계에 대해 제대로 알지 못 했었고 자바 언어는 공부한적이 있지만 안드로이드에서 쓰 이는 모듈은 꽤나 달라 이를 이해하는데 시간이 많이 소요 되었지만 프로젝트를 마무리하며 이에 대한 이해를 높여 성취감을 느꼈다. |