

# Streaming Video Scout Vehicle

- 원격 제어를 이용한 실시간 영상 정찰 차량 -

## 프로젝트 완료 보고서

- 제정일자 : 2015년 02월 21일
- 문서버전 : Ver 2.0
- 팀명 : VISION

## 문 서 승 인 정 보

|         |                               |
|---------|-------------------------------|
| 프로젝트 명  | Streaming Video Scout Vehicle |
| TASK 명  | 원격 제어를 이용한 실시간 영상 정찰 차량       |
| 문 서 명   | 프로젝트 완료보고서                    |
| 발행 년 월일 | 2015년 02월 21일                 |

| 구 분              | 성 명   | 서 명 | 일 자        |
|------------------|-------|-----|------------|
| Team Member      | 서 강 원 |     | 2015.03.04 |
| Team Member      | 신 익 환 |     | 2015.03.04 |
| Team Member      | 이 호 정 |     | 2015.03.04 |
|                  |       |     |            |
|                  |       |     |            |
|                  |       |     |            |
| Technical Leader | 김 주 혁 |     | 2015.03.04 |
| Team Leader      | 문 선 균 |     | 2015.03.04 |
| 프로젝트관리자          | 문 상 환 |     | 2015.03.04 |

## 문 서 이 력 정 보

| Ver. | Page | 작성일자       | 변경사항 | 작성자 | 승인자 | 승인일자       |
|------|------|------------|------|-----|-----|------------|
| 1.0  | 102  | 2015.02.21 | 제 정  | 이호정 |     | 2015.02.21 |
| 2.0  | 19   | 2015.03.04 | 수 정  | 이호정 |     | 2015.03.04 |
|      |      |            |      |     |     |            |
|      |      |            |      |     |     |            |
|      |      |            |      |     |     |            |
|      |      |            |      |     |     |            |
|      |      |            |      |     |     |            |
|      |      |            |      |     |     |            |
|      |      |            |      |     |     |            |
|      |      |            |      |     |     |            |
|      |      |            |      |     |     |            |
|      |      |            |      |     |     |            |
|      |      |            |      |     |     |            |
|      |      |            |      |     |     |            |
|      |      |            |      |     |     |            |

## 목차

1. 프로젝트 배경
  - A. 솔루션 개발의 필요성
  - B. 신기술 개발 시 파급효과 및 활용방안
  - C. 프로젝트 개발 목적
2. 프로젝트 진행 보고
  - A. 프로젝트 개발 분야
  - B. 프로젝트 진행 일정
  - C. 수행업무 및 담당자
  - D. 개발 도구
  - E. 프로젝트 목표
  - F. 단계별 아키텍처
3. 프로젝트 개발 일정
  - A. 팀 개발 일정
  - B. 개인 개발 일정
4. 프로젝트 개발 내용
  - A. 프로젝트 배경 지식/기술/알고리즘
  - B. 프로젝트 상세 개발 내용
5. 사용자 매뉴얼
6. 프로젝트 마무리
  - A. 기대효과
  - B. 문제점
  - C. 개선방안
  - D. 참고문헌 및 논문
  - E. 참고사이트
  - F. 팀원 별 소감

## 1. 프로젝트 배경

### A. 솔루션 개발의 필요성

기존 CCTV 혹은 차량 블랙박스들은 영상을 바로 확인할 수 없는 불편함이 존재하고 사람이 실제로 접근할 수 없으나 정찰이 필요한 곳이 있을 수 있다. 이러한 경우, 저렴한 가격의 디바이스(Raspberry Pi)와 애플리케이션을 통해 차량을 제어하여, 실제 이동을 하지 않으면서 정찰이 가능하고, 영상을 간편하게 확인할 수 있는 기능을 제공한다.

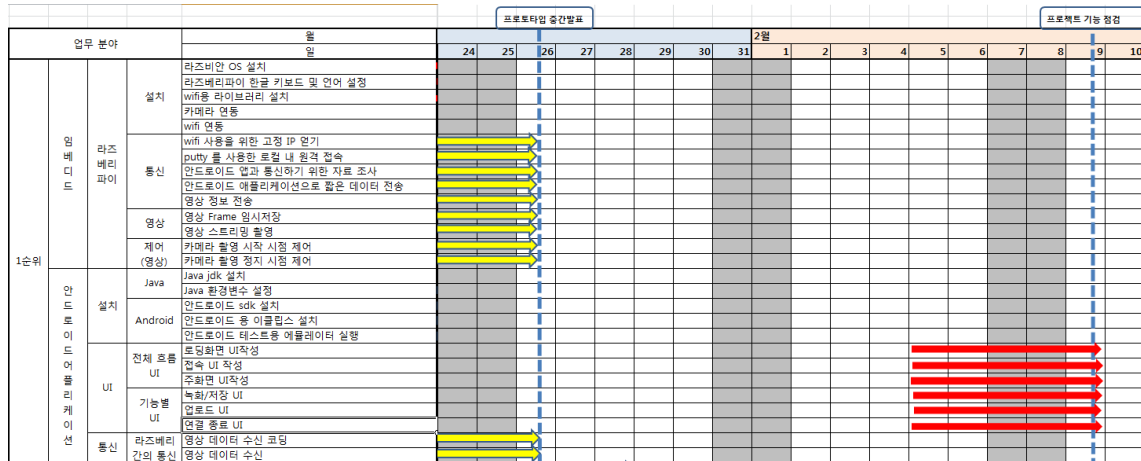
### B. 신기술 개발 시 파급효과 및 활용방안

애플리케이션을 통해서 차량을 제어하여 실제 이동하지 않으면서 정찰이 가능하며, 영상을 실시간으로 확인할 수 있다.

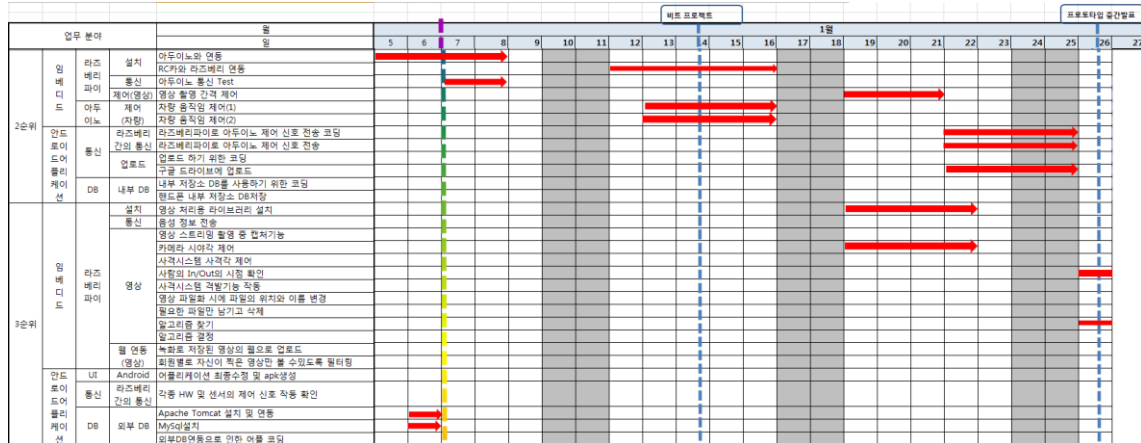
### C. 프로젝트 개발 목적

기존의 CCTV가 고정되어 있어야 한다는 인식을 벗어나 CCTV를 이동식 차량에 탑재하여 실시간으로 볼 수 있도록 하는 것이다. 또한 인간이 직접 정찰을 할 수 없는 장소 또한 무선 차량 제어를 통해 실시간 정찰이 가능하기 때문에 효율적인 정찰이 가능할 것이다.





## 2) 2순위, 3순위



|       |       |        |                             | 프로젝트 가능 점검               |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
|-------|-------|--------|-----------------------------|--------------------------|---------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|----|----|--|--|
| 업무 분야 |       |        |                             | 월                        |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
|       |       |        |                             | 일                        | 27      | 28 | 29 | 30 | 31 | 2월 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |  |  |
| 2순위   | 임베디드  | 라즈베리파이 | 설치                          | 아두이노와 연동                 |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
|       |       |        | 통신                          | RC카와 라즈베리 연동             |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
|       |       |        | 제어(영상)                      | 아두이노 통신 Test             |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
|       |       | 아두이노   | 제어                          | 영상 촬영 간격 제어              |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
|       |       |        | (자랑)                        | 자랑 움직임 제어(1)             |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
|       | 안드로이드 | 라즈베리파이 | 제어                          | 자랑 움직임 제어(2)             |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
|       |       |        | 통신                          | 라즈베리파이로 아두이노 제어 신호 전송 코딩 |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
|       |       |        | 영상                          | 라즈베리파이로 아두이노 제어 신호 전송    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
|       |       | DB     | 업로드                         | 업로드 하기 위한 코딩             |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
|       |       |        | 내부 DB                       | 구글 드라이브에 업로드             |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
| 3순위   | 임베디드  | 라즈베리파이 | 설치                          | 내부 저장소 DB를 사용하기 위한 코딩    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
|       |       |        |                             | 핸드폰 내부 저장소 DB저장          |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
|       |       |        | 통신                          | 영상 처리용 라이브러리 설치          |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
|       |       |        |                             | 영상 정보 전송                 |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
|       |       |        | 영상                          | 영상 스트리밍 촬영 중 캡처기능        |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
|       |       |        |                             | 카메라 시야각 제어               |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
|       |       |        |                             | 사격시스템 사격각 제어             |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
|       |       |        |                             | 사람의 In/Out의 시점 확인        |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
|       |       |        |                             | 사격시스템 격발기능 작동            |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
|       |       |        |                             | 영상 파일화 시에 파일의 위치와 이름 변경  |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
|       |       |        |                             | 필요한 파일만 남기고 삭제           |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
|       |       |        |                             | 알고리즘 찾기                  |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
|       |       |        |                             | 알고리즘 결정                  |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
|       | 안드로이드 | UI     | 웹 연동 (영상)                   | 녹화로 저장된 영상의 웹으로 업로드      |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
|       |       |        | 화면별로 자신이 찍은 영상만 볼 수 있도록 필터링 |                          |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
|       |       | 통신     | 아플리케이션 최종수정 및 apk생성         |                          |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
|       |       |        | 라즈베리                        | 각종 HW 및 센서의 제어 신호 작동 확인  |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
|       |       |        | 간의 통신                       | Apache Tomcat 설치 및 연동    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
| 3순위   | 임베디드  | 라즈베리파이 | DB                          | 외부 DB                    | MySQL설치 |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |
|       |       |        |                             | 외부DB연동으로 인한 어플 코딩        |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |    |    |  |  |


|       |             |           |                             |                          | 프로젝트 보고서              |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  | 프로젝트 개발완료 |  |  |
|-------|-------------|-----------|-----------------------------|--------------------------|-----------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|--|--|--|-----------|--|--|
| 업무 분야 |             |           |                             |                          | 월                     |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
|       |             |           |                             |                          | 일                     | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |  |  |  |           |  |  |
| 2순위   | 임베디드        | 라즈베리파이    | 설치                          | 아두이노와 연동                 |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
|       |             |           | 통신                          | RC카와 라즈베리 연동             |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
|       |             |           | 제어(영상)                      | 아두이노 통신 Test             |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
|       |             | 아두이노      | 제어                          | 영상 촬영 간격 제어              |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
|       |             |           | (자랑)                        | 자랑 움직임 제어(1)             |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
|       | 안드로이드어플리케이션 | 라즈베리간의 통신 | 제어                          | 자랑 움직임 제어(2)             |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
|       |             |           | 통신                          | 라즈베리파이로 아두이노 제어 신호 전송 코딩 |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
|       |             |           | 업로드                         | 라즈베리파이로 아두이노 제어 신호 전송    |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
|       |             | DB        | 업로드                         | 업로드 하기 위한 코딩             |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
|       |             |           | 내부 DB                       | 구글 드라이브에 업로드             |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
| 3순위   | 임베디드        | 라즈베리파이    | 설치                          | 내부 저장소 DB를 사용하기 위한 코딩    |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
|       |             |           |                             | 핸드폰 내부 저장소 DB저장          |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
|       |             |           | 통신                          | 영상 처리용 라이브러리 설치          |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
|       |             |           |                             | 영상 정보 전송                 |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
|       |             |           | 영상                          | 영상 스트리밍 촬영 중 캡처기능        |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
|       |             |           |                             | 카메라 시야각 제어               |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
|       |             |           |                             | 사격시스템 사격각 제어             |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
|       |             |           |                             | 사람의 In/Out의 시점 확인        |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
|       |             |           |                             | 사격시스템 격발기능 작동            |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
|       |             |           |                             | 영상 파일화 시에 파일의 위치와 이름 변경  |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
|       |             |           |                             | 필요한 파일만 남기고 삭제           |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
|       |             |           |                             | 알고리즘 찾기                  |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
|       |             |           |                             | 알고리즘 결정                  |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
|       | 안드로이드어플리케이션 | UI        | 웹 연동 (영상)                   | 녹화로 저장된 영상의 웹으로 업로드      |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
|       |             |           | 화면별로 자신이 찍은 영상만 볼 수 있도록 필터링 |                          |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
|       |             | 통신        | 아플리케이션 최종수정 및 apk생성         |                          |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
|       |             |           | 라즈베리간의 통신                   | 각종 HW 및 센서의 제어 신호 작동 확인  |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
|       |             |           | DB                          | 외부 DB                    | Apache Tomcat 설치 및 연동 |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
|       |             |           | MySQL설치                     |                          |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |
|       |             |           | 외부DB연동으로 인한 어플 코딩           |                          |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |           |  |  |



## C. 수행업무 및 담당자

| 책임     | 직무               | 사진   | 이름  | 역할 설명                                |
|--------|------------------|--|-----|--------------------------------------|
| Leader | 팀장               |    | 문선균 | 일정 관리<br>기획서 관리<br>회의 주제 및<br>회의록 정리 |
|        | Technical Leader |   | 김주혁 | 기술 서포트<br>팀장 부재시<br>팀장 보조            |
| 임베디드   | 정                |  | 문선균 | 라즈베리파이,<br>임베디드 프로<br>그래밍            |
|        | 부                |  | 이호정 | 라즈베리파이,<br>임베디드 프로<br>그래밍            |

|         |   |  |     |                  |
|---------|---|--|-----|------------------|
| 영상 및 통신 | 정 |    | 서강원 | 라즈베리와 안드로이드 간 통신 |
|         | 부 |   | 이호정 | 라즈베리와 안드로이드 간 통신 |
| 안드로이드   | 정 |  | 김주혁 | 안드로이드 프로그래밍      |
|         | 부 |  | 신익환 | 안드로이드 프로그래밍      |
| Web     | 정 |  | 신익환 | Web DB 프로그래밍     |

|        |   |  |     |            |
|--------|---|--|-----|------------|
| Design | 정 |  | 이호정 | Web Design |
|--------|---|--|-----|------------|

## D. 개발 도구

### A. 기술

- 안드로이드 프로그래밍, OpenCV, Python, 안드로이드 애플리케이션 내부 DB, 소켓 프로그래밍, H264, CSS, PHP, MySQL, APACHE,

### B. 개발 S/W

- Visual Studio 2012, MySQL, Eclipse, DEBIAN, Notepad++

### C. 장비

- 라즈베리 파이, 카메라모듈, 스마트 폰, 각종 센서, 와이파이 모듈, 외장 배터리, 모터 드라이버

## E. 프로젝트 목표

- 1단계 : 전체 프로젝트 달성을 위한 핵심적인 기능 구현.

| 업무 분야 | 기능   |
|-------|--|
| 임베디드  | ① 라즈베리파이 셋팅<br>② 모듈 제어: 디바이스에 부착된 모듈을 제어하여 데이터를 처리한다.<br>A. CAM<br>i. 영상을 촬영한다.<br>B. Wifi<br>i. 촬영된 영상을 안드로이드 애플리케이션으로 전송한다.<br>C. RC카<br>i. 안드로이드 애플리케이션에서 제어한다. |
| 안드로이드 | ① 안드로이드 셋팅<br>② 기본 UI 및 기능 구현  |

- 2단계 : 영상 전송뿐만 아니라 다양한 센서들을 이용한 추가 기능 구현 및 영상 저장 기능 구현

| 업무 분야                            | 기능  |
|----------------------------------|---|
| <b>1단계까지 개발된 기능은 기본적으로 포함된다.</b> |   |
| 임베디드                             | ① 센서제어 : 디바이스 부착된 센서들을 제어하여 데이터를 처리한다.<br>A. 온습도 센서<br>i. 온습도를 감지한다<br>B. 적외선 센서<br>i. 인체를 감지한다<br>C. 조도 센서<br>i. 빛의 밝기를 감지한다                   |
| 안드로이드                            | ① 안드로이드 NDK 설치<br>② 스트리밍 : 라즈베리파이에서 송신한 영상데이터를 수신한다.<br>③ 녹화 / 저장 : 수신한 영상 데이터를 녹화 / 저장한다.  |
| 웹                                | ① 환경설정<br>A. Mysql 설치<br>B. Apache 설치<br>C. PHP 설치<br>② UI 작성<br>A. 로그인화면, 회원가입화면, 동영상목록, 자유게시판 등 전체 흐름 UI 작성<br>B. 회원가입, 로그아웃, 글쓰기 등 기능별 UI 작성 |

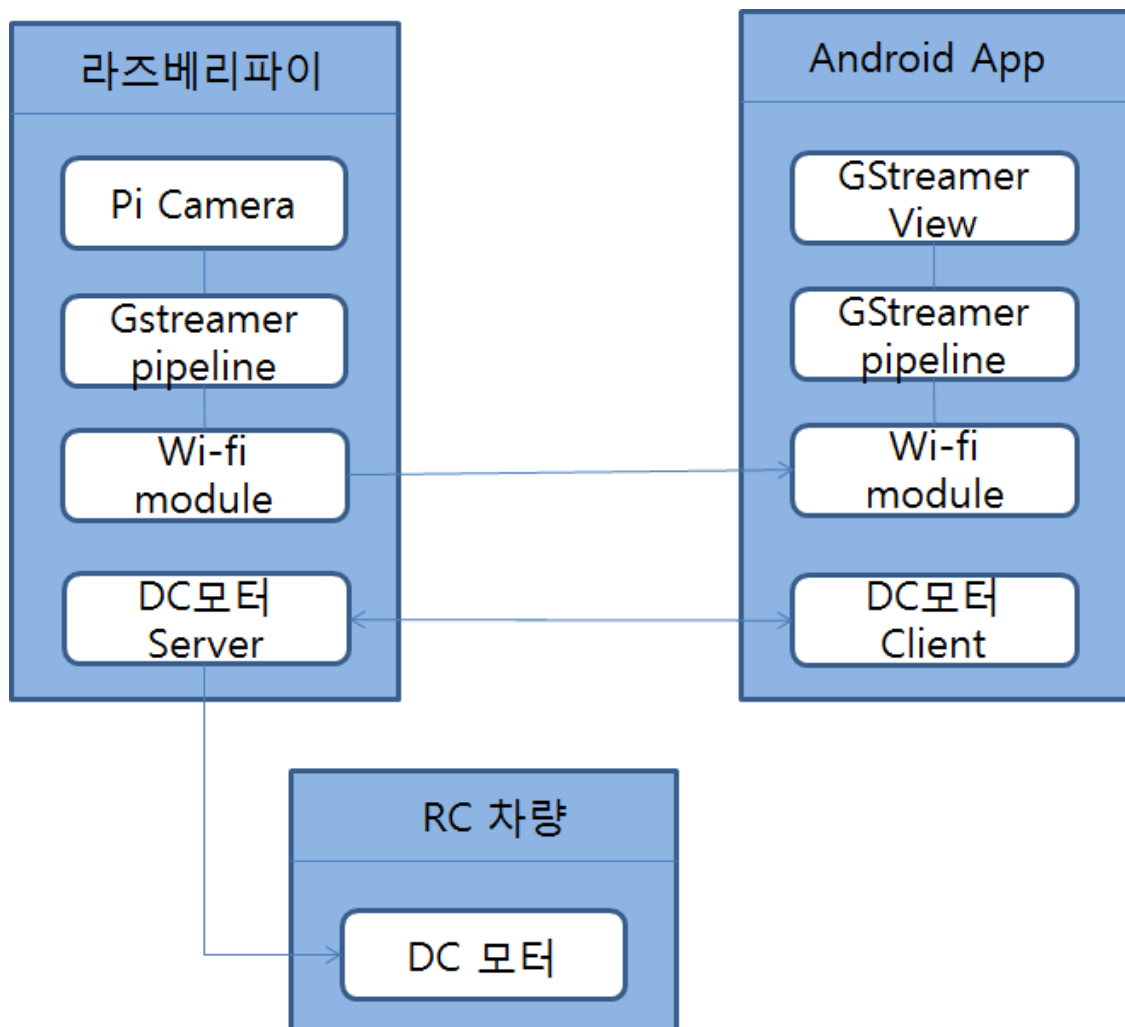
- 3단계 : 저장된 영상을 웹서버에 업로드 하는 기능 구현

| 업무 분야                            | 기능   |
|----------------------------------|--|
| <b>2단계까지 개발된 기능은 기본적으로 포함된다.</b> |  |
| 임베디드                             | ① 사격 기능 제어 : 디바이스에 사격을 위해 부착된 하드웨어를 제어하여 사격 시스템 구현 |
| 안드로이드                            | ① 업로드 : 웹서버에 업로드 및 공유 기능                           |

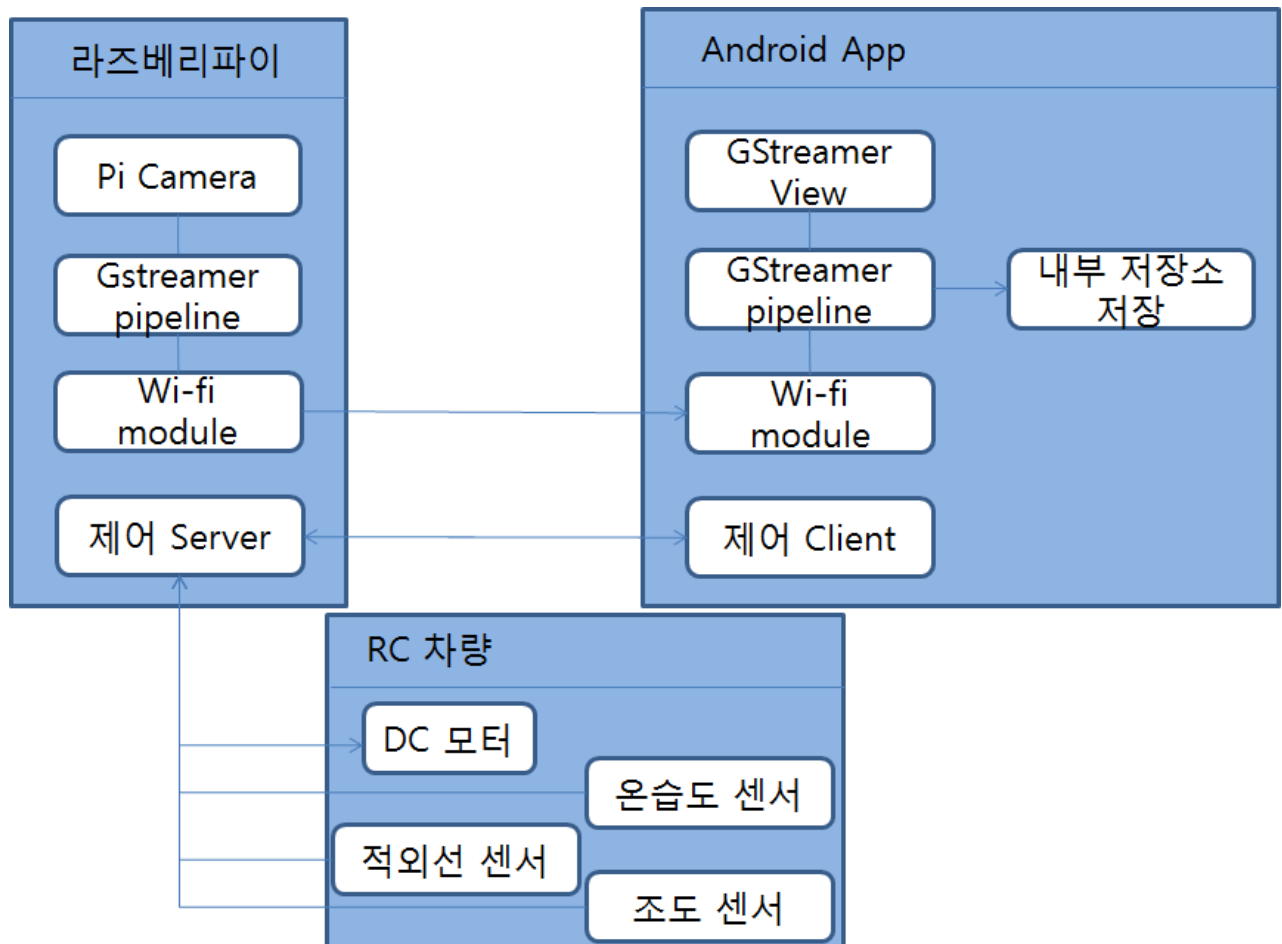
|   |   |
|---|---|
|   | ② DB와 연동<br>③ UI 작성 : 회원가입, 조도감지, 움직임 감지, 진동감지, 온습도 기능 등 기능별 UI 추가<br>④ 애플리케이션 최종 수정 및 APK 파일생성 |
| 웹 | ① 다운로드 : 다운로드 UI 작성   |

## F. 단계별 아키텍처

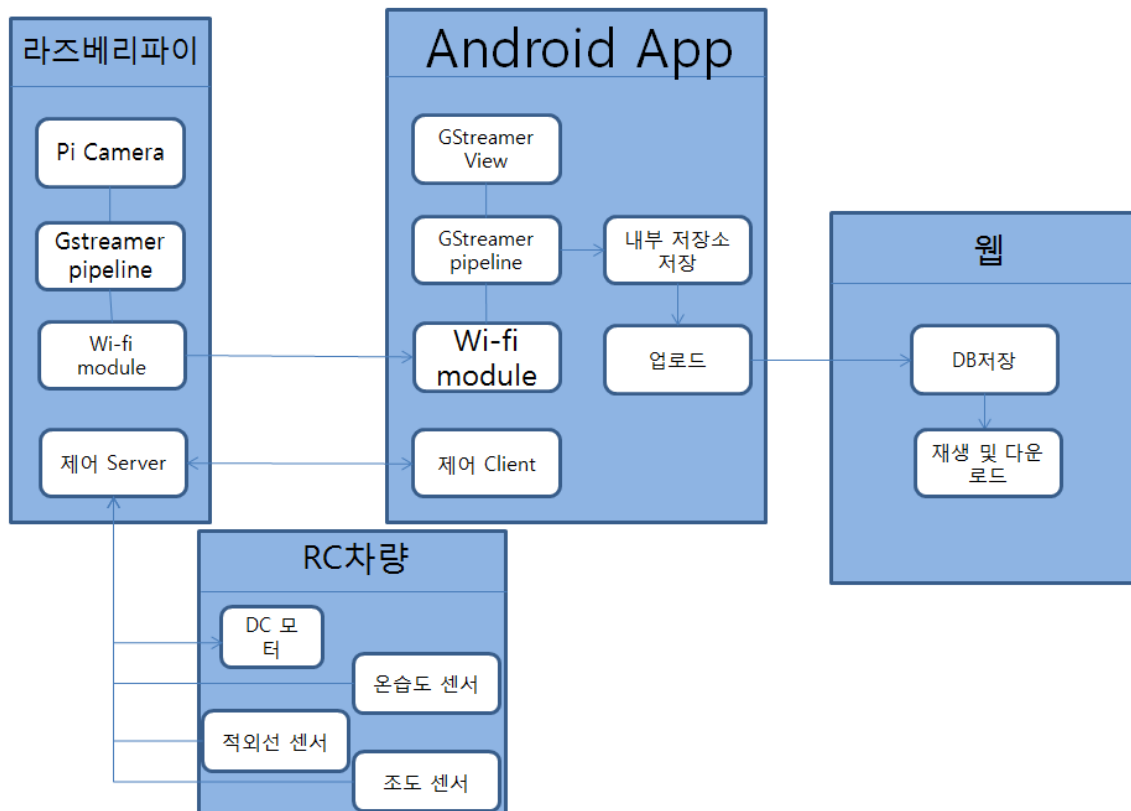
### ● 1차 아키텍처



## ● 2차 아키텍처



## ● 3차 아키텍처



### 3. 프로젝트 개발 일정

#### A. 팀 개발 일정

| 12월 다섯째 주 |  |
|-----------|--|
| 12월 29일   | 브레인스토밍, 프로젝트 주제 결정, 조명/프로젝트명 결정, 차일 논의사항 결정  |
| 12월 31일   | 역할분담및프로젝트목표재설정,프로젝트업무분담,조이름/프로젝트명재설정,개인별한일<br>체크,2일에와서할일논의   |
| 1월 첫째 주   |  |
| 1월 2일     | 장비 주문 문제에 대해 논의, 문서 작성   |
| 1월 3일     | 각자 문서 작성 후 취합하고 재검토 및 수정을 거친 후 최종 제출용 문서 제작  |
| 1월 4일     | 전체일정표,개인일정표작성.테이블정의서작성.기능정의서대분류를확실히하고기능을세<br>부적으로나눈다.  |
| 1월 둘째 주   |  |
| 1월 5일     | <b>임베디드</b>  |
|           | 아두이노기본개발환경설치및파이썬구동환경구성   |
|           | 파이썬기본문법공부및라즈베리파이를이용한프로그램설계및코드분석  |
|           |  |
| 1월 6일     | <b>안드로이드</b>   |
|           | 안드로이드개발환경설정및공부,간단한 UI 설계   |
|           | <b>안드로이드</b>   |
|           | 위젯아이콘모양변경,라즈베리파이접속 Activity 설계,모의 ID 입력시연결이되도록설계,동영<br>상출력부분에외부 URI 연결하여재생하도록구현,연결종료시라즈베리접속을해제하며접속<br>Activity 로돌아가게구현,갤러리아플리케이션으로이동하여저장된사진이나동영상을공유<br>할수있게구현.영상목록으로이동하는 Activity 전환구현 |
| 1월 7일     | <b>통신</b>  |
|           | 안드로이드기기별 wifip2p 지원여부확인,wifip2p 공부   |
|           | <b>안드로이드</b>   |
|           | 안드로이드 N 아설치및개발환경설정,아이콘이미지변경,ListView 를사용하여동영상목록을<br>타나낼수있는기초적인틀구현,연결끊기버튼구현.안드로이드어플에서 wifip2psetup 및연<br>결.   |
| 1월 8일     | <b>영상</b>  |
|           | GstOpenMax 를통해영상을 H.264 로인코딩/디코딩하는방법과 GstOpenMax 에대한전반적<br>인내용공부   |
|           | <b>서버</b>  |
|           |  |



|          |  |
|----------|--|
|          | 파일전송이아닌영상스트리밍만을위한스트리밍서버개설시도  |
| 1 월 8 일  | <b>안드로이드</b>   |
|          | 안드로이드쓰레드와서비스공부,안드로이드동영상목록구현,안드로이드에서 wifip2p 모듈을작동시키는부분까지성공.안드로이드 UI 기능추가(내부저장소모든동영상목록불러오기, 동영상이름클릭시동영상재생,공유하기버튼구현-미완성) |
|          | <b>영상전송</b>  |
|          | 바이너리영상파일을인코딩,디코딩하는다양한방법,영상의포맷을변경하는방법,GstOpenMax 를사용하여지연을최소화한영상전송예제에대하여공부   |
| 1 월 9 일  | 맥북에서개발환경구성(듀얼부트)   |
|          | <b>안드로이드</b>   |
|          | wifi 활성화및 wifi 자동연결기능구현,wifidirect 테스트결과안드로이드기기별호환의문제및지원이제대로이루어지지않아다른방식으로구현하기위한공부.안드로이드 UI 기능수정(공유기능)및오류검사             |
|          | <b>임베디드</b>  |
|          | 라즈베리파이파이카메라기본활성화설정및카메라모듈을통해사진,동영상촬영이가능한지테스트.무선랜설정및유선랜설정.(유선랜사용시 ip 확인이힘들어무선랜사용)  |
|          | RC 카본해및회로도분해작업,서보모터를이용한틸트/팬카메라제어 HW 조립   |
| 1 월 10 일 | <b>임베디드</b><br>서보모터 제어 프로그래밍, 맥에서 netcat 을 이용하여 라즈베리파이에 접근 시도  |
| 1 월 11 일 | <b>영상전송</b>  |
|          | Gstreamer 를이용한영상전송공부   |
|          | <b>통신</b>  |
|          | Wirelesslocalareanetwork 에관련된공부  |
| 1 월 셋째 주 |  |
| 1 월 12 일 | <b>임베디드</b>  |
|          | 서보모터제어성공,라즈베리파이에 gstreamer 설치실패(망가진 package 문제)  |
|          | <b>안드로이드</b>   |
|          | wifi 목록구현성공,Hotspot 공부   |
|          | <b>웹&amp;DB</b>  |
|          | MySQL,아파치,PHP 설치및연동,안드로이드 MySQLdata 송수신프로그램작성  |
| 1 월 13 일 | <b>임베디드</b>  |
|          | 아두이노키트,각종센서(온/습도,적외선,자이로,스케치로더)납땜,서보모터제어해결중(서보모터제어중라즈베리파이꺼짐현상),차량 DC 모터제어성공,gstreamer 설치완료                             |
|          | <b>웹&amp;안드로이드</b>   |

|          |  |
|----------|--|
|          | 안드로이드개발(MySQL 과연동하는로그인화면),MySQL 계정생성및회원정보테이블작성,Members 테이블에서 Id,Password 를불러오는 PHP 파일작성,안드로이드와웹서버연동(개발중)   |
|          | 영상전송   |
|          | mjpgstreamer를사용한영상전송확인,gstreamer를사용한영상전송확인(노트북에서확인완료)  |
| 1 월 14 일 | 비트발표회  |
|          | 임베디드   |
|          | 서보모터제어,영상포맷을 h264->mp4 변환하는법공부   |
|          | 안드로이드  |
|          | UI 변경(버튼추가등)   |
| 1 월 15 일 | 웹안드로이드와웹서버연동시도   |
|          | 임베디드   |
|          | 파이카메라로웹스트리밍시전송되는 uri 를알아내는방법공부,라즈베리파이에서웹상의다음팟플레이어로영상스트리밍받기,라즈베리파이에서안드로이드 vlc 프로그램으로영상스트리밍받기,라즈베리파이원격제어환경구성(XRDP,SAMBA),ArchLinux 설치시도(실패),XBMC 설치(설치는완료,실행불가),DC 모터제어코드작성,제어를위한웹페이지작성,Gstreamer 를이용한녹화및 h.264 파일을 mp4 파일로변환하는방법을공부,gstreamer 를이용해웹캠영상을안드로이드어플에서볼수있는예제를찾아실행가능한지시도(실패) |
|          | 안드로이드  |
|          | gst-RTSP 관련조사,서버프로그래밍공부,안드로이드 vlc 관련코드분석   |
|          | 웹  |
|          | 안드로이드로그인페이지오류수정,외부 PC 로교내 ip 접근이불가하여외부에웹서버및 DB 서버설치  |
|          | 임베디드   |
|          | 라즈베리파이에서 gstreamer 를이용해서영상을녹화하는방법을연구,DC 모터제어코드,제어를위한웹페이지수정으로제어성공   |
| 1 월 16 일 | 안드로이드  |
|          | 라즈베리파이파이카메라를 gstreamer 를이용해안드로이드개발앱으로영상찍우는방법연구,gstreamer 개발자관련조사,파이썬소켓조사,안드로이드 MySQL 을연동한로그인화면구현,회원가입화면구현중(테이블저장,id 중복확인)  |
|          | 안드로이드  |
| 1 월 17 일 | 안드로이드  |
|          | 안드로이드핫스팟기능구현공부,안드로이드어플내에서핫스팟이름,pw 설정및활성화구현중  |
|          | 임베디드   |
|          | Gstreamer 를통해영상을실시간전송하는것(실패)   |

|          |  |
|----------|--|
|          | <b>웹</b><br>회원가입페이지구현(ID 중복확인),오류수정(PHP 파일에서 Post 방식으로값을받을때두번째부터 NULL 값이반환되는것)   |
| 1 월 18 일 | <b>웹</b><br>안드로이드로딩,로그인,회원가입화면디자인,회원정보공백입력시메시지발생,회원가입하고 Main 으로갈때 Activity 가종료되지않는문제해결,PHP 이론공부(HTML 과의관계,변수해석,연산자,제어문)<br><b>안드로이드</b><br>안드로이드핫스팟기능구현공부,안드로이드어플내에서핫스팟이름,pw 설정및활성화구현중<br><b>임베디드</b><br>Gstreamer 를통해안드로이드어플에서영상을 streaming 으로볼수있게하는것시도   |
| 1 월 넷째 주 |  |
| 1 월 19 일 | <b>안드로이드</b><br>안드로이드핫스팟기능구현및안드로이드 sdktools/ndk 설치<br><b>임베디드</b><br>DC 모터제어코드,제어를위한웹페이지수정으로제어성공<br><b>웹</b><br>안드로이드소스코드병합및수정,Eclipse 에서 PHP 개발환경설치,웹페이지개발(db 접속,회원가입폼)   |
| 1 월 20 일 | <b>안드로이드</b><br>gststreamersdk 설치,gstreamer 튜토리얼테스트,웹뷰로스트리밍테스트<br><b>웹</b><br>db 연동한웹페이지개발중(로그인폼)<br><b>임베디드</b><br>JDK8 을라즈베리파이에설치하여 TCP/IP 기반의 server-client 를이용하는 DC 모터제어환경구축   |
| 1 월 21 일 | <b>안드로이드</b><br>안드로이드 mjpgstreamer 를온라인 firefox,핸드폰 firefox 어플로영상스트리밍성공(자체적인어플에서는실패),gstreamertutorial5 이용하여 http 방식스트리밍재생,자바소켓서버(클라이언트간간단한메시지전송프로그램작성),gstreamerrtsp 예제를찾아실행(실패),gstreamertutorial5 를이용하여안드로이드에서 http 영상재생이가능한것을확인하고 mjpg 스트리밍이가능한지확인(mjpg 는 gstreamer 와방식이달라서인지버퍼링이발생하여 gstreamer 에서사용하기에어렵다고판단)<br><b>웹</b> |

|           |  |
|-----------|--|
|           | 라즈베리파이폴더접근방법조사및설치(연동 X)(FTPserver 또는 samba 이용),웹페이지개발<br>진행                  |
| 1 월 22 일  | 중간발표준비및중간발표  |
| 1 월 23 일  | 프로토타입 발표 후 보완할 점에 대해 회의  |
| 1 월 24 일  | 문서 업데이트  |
| 1 월 다섯째 주 |  |
| 1 월 26 일  | 웹  |
|           | 웹에서경로지정을통하여디렉토리및파일목록을불러오는기능구현(파일읽기가가능한상<br>태),스마트폰핫스팟에접속한라즈베리파이와맥북이연동되는것을확인  |
|           | 임베디드   |
|           | 아두이노납땜   |
|           | 문서업데이트   |
|           | SW-UI_HW 외부구조설계서수정완료(김주혁,신익환),프로젝트기획서수정완료(이호정),기<br>능정의서수정미완료(팀전체)           |
| 1 월 27 일  | 임베디드   |
|           | 환경센서보드제작및테스트,서보모터제어모듈설치,서보모터(팬/틸트)제어테스트,제어부<br>제작중                           |
|           | 웹  |
|           | 라즈베리파이고정 IP 갖는방법조사,WINDOWS,MYSQL,APACHE,PHP 설치(맥<br>북),CYGWIN,NDK 설치(PC)     |
|           | 안드로이드  |
|           | 안드로이드 NDK 설치완료및기본예제실행  |
| 1 월 28 일  | 임베디드   |
|           | 조도센서,적외선센서,확장핀 GPIO 테스트,서보모터제어부제작중,라즈베리파이에<br>http-launch 에필요한파일설치           |
|           | 안드로이드  |
|           | gststreamerandroids 아구성성공,gststreamerplugin 개발문서공부                           |
|           | 웹  |
|           | php 이론공부및전체적인웹 UI 구상   |
| 1 월 29 일  | 안드로이드  |
|           | gststreamerpipeline 구조에대해서다양한구조가능성에대한공부및논<br>의,gstreamermp4streaming 문서조사및공부 |
|           | 임베디드   |
|           | 카메라모듈의촬영을돕기위한팬/틸트서보모터제어부완성   |
|           | 웹  |

|          |   |
|----------|---|
|          | HTML,JavaScript 공부및웹개발  |
| 1 월 30 일 | <b>안드로이드</b>  |
|          | 안드로이드애플리케이션에서.h264 포맷인영상재생성공  |
|          | <b>웹</b>  |
|          | 게시판,자료실개발   |
| 1 월 31 일 | <b>임베디드</b>   |
|          | 온/습도센서코딩에관한계산수식공부및간단한모의코드제작   |
|          | <b>안드로이드</b>  |
|          | gssocket 공부및실습  |
| 2 월 둘째 주 |   |
| 2 월 2 일  | <b>임베디드</b>   |
|          | 라즈베리파이라즈비안재설치및파이프라인 STANDARDIN,OUTDATA 교환,움직임감지적외선센서테스트(오작동),gststreamerrt 네영상전송시도(실패)                      |
|          | <b>웹</b>  |
|          | 자료실개발(파일복사하여폴더에저장및읽기,테이블에파일정보저장)  |
| 2 월 3 일  | <b>임베디드</b>   |
|          | gststreamerrtsp 를이용한뷰로띄우기부분실패(설정부분오류),PIR(적외선인체감지센서)제어코드작성,조도센서제어코드작성,온습도센서제어코드작성,라즈베리파이에 JDK8,wiringPi 재설치 |
|          | <b>영상</b>   |
|          | 이미지파일바이너리값확인,streaming-video-rtsp-rtp 조사및소스코드분석   |
|          | <b>웹</b>  |
|          | 자료실,게시판완성및메인과연동   |
| 2 월 4 일  | <b>안드로이드</b>  |
|          | gststreamer0.10 버전에서 1.0 버전으로 porting, gststreamer pipeline 을 구현하기 위한 조사 및 공부                               |
|          | <b>웹</b>  |
|          | 웹페이지(프로토타입)구현   |
| 2 월 5 일  | <b>안드로이드</b>  |
|          | gststreamerpipeline 구조파악,안드로이드 gststreamerpipeline 일부구성완료   |
|          | <b>임베디드</b>   |
|          | 센서테스트및병합작업,라즈베리파이에서 gststreamer 를사용하여스트리밍과동시에영상저장하는기능구현   |
|          | <b>웹</b>  |
|          | 아파치에서네트워크드라이브를 DocumentRoot 로 지정하는방법조사,Samba 를계정없이사용하는방법조사  |

|        |   |
|--------|---|
| 2월 6일  | 안드로이드   |
|        | gststreamerreceive 쪽파이프라인구현진행중(rtp,udp 사용)  |
|        | 임베디드  |
|        | 자이로센서제어를위해 I2C 공부,자이로센서제어를위한 I2C(커널시리얼통신)설치(실패),라즈베리파이에서 gststreamer 를사용하여스트리밍과동시에영상저장기능구현,라즈베리파이에 Apache,php,Mysql 설치후 putty 로접근하는방법조사 |
|        | 웹   |
| 2월 7일  | 아파치홈디렉토리를네트워크드라이브로하려고시도중  |
|        | 안드로이드   |
|        | gststreamerreceive 쪽 pipeline 구현진행중   |
|        | 임베디드  |
| 2월 8일  | 자이로센서활성화방법모색중,PIR 센서코드수정,I2C 포트사용법찾는중,라즈베리파이에 Apache,PHP,Mysql 설치및연동  |
|        | 임베디드  |
|        | 전반적인모든하드웨어세팅,서보모터와모터드라이버의이상작동(고장),I2C 포트의자문을 구함,라즈베리파이에 Apache,PHP,Mysql 설치및연동  |
|        | 안드로이드   |
| 2월 9일  | Gstreamerreceive 쪽파이프라인구현성공   |
|        | 2월 셋째 주   |
| 2월 9일  | 2 차 발표  |
| 2월 10일 | 안드로이드   |
|        | 한스팟 hostip 주소확인,gstreamer 동적파이프라인변경공부,gstreamer 를사용한녹화기능 구현관련조사   |
|        | 임베디드  |
|        | 자이로센서활성화성공및 3D 가상시뮬레이션모델제작중,라즈베리파이 PHP,MySQL 연동   |
| 2월 11일 | 웹   |
|        | 웹상에서원하는확장자만불러오는방법조사및코드분석  |
|        | 안드로이드   |
| 2월 12일 | 안드로이드와웹연동구현중,안드로이드에서 gststreamer 를이용한녹화기능구현중  |
|        | 웹   |
|        | 웹디자인을위한자료조사   |
| 2월 13일 | 안드로이드   |
|        | 안드로이드에서 gststreamer 를사용한녹화기능구현,안드로이드에웹서버,php,mysql 설치후 외부 pc 에서연동하고 php5.5 버전에맞게수정중   |
|        | 웹   |
| 2월 13일 | 웹디자인을위한자료조사   |
|        | 안드로이드   |

|          |   |
|----------|---|
|          | 안드로이드 UI 개선,gstreamer 를사용한녹화기능구현,안드로이드에서웹서버로파일전송에<br>제코드분석  |
|          | <b>웹</b>  |
|          | 웹디자인  |
|          | <b>임베디드</b>   |
|          | 센서부분안드로이드와연동  |
| 2 월 14 일 | <b>안드로이드</b><br>안드로이드 UI 개선,gstreamer 를사용한녹화기능구현,안드로이드에서웹서버로파일전송에<br>제코드분석및샘플코드작성   |
|          | <b>웹</b>  |
|          | 웹디자인에필요한샘플페이지분석   |
| 2 월 15 일 | <b>안드로이드</b><br>안드로이드 UI 개선,gstreamer 를사용한녹화기능구현,H264 영상파일재생  |
|          | <b>웹</b>  |
|          | 웹디자인에필요한샘플페이지분석   |
| 2 월 넷째 주 |   |
| 2 월 16 일 | <b>안드로이드</b><br>안드로이드동영상목록,재생재수정,안드로이드이미지최종수정,gstreamerandroidplayer 시<br>도,안드로이드에서웹서버로단일파일전송(텍스트 O,이미지/영상 X),안드로이드회원별영상<br>목록불러오는방법구상중 |
|          | <b>웹</b>  |
|          | 웹디자인을위한샘플페이지작성  |
|          | <b>임베디드</b><br>PIR(적외선인체감지센서)파이썬코드를자바로재코딩,조도센서및자이로센서의파이썬코드<br>를자바로변환하는작업진행중,사격시스템에필요한하드웨어제작중(서보모터하단부와기동<br>제작)                          |
| 2 월 17 일 | <b>안드로이드</b><br>gstreamer 를이용한녹화및재생기능완성,안드로이드에서웹서버로단일파일전송,로그인,회<br>원가입 UI 병합  |
|          | <b>임베디드</b><br>사격시스템완성(상,하,좌,우제어성공,단일사격성공),PIR 센서연동및작동테스트,조도센서코<br>드완성및연동테스트와다른센서들의핀호환여부및작동테스트를기준으로자바코드제작여<br>부결정                       |
|          | <b>웹</b>  |
|          | 웹디자인샘플페이지작성및기존웹페이지와병합   |
| 2 월 18 일 | <b>안드로이드</b><br>안드로이드동영상목록,재생 2 차재구현,안드로이드배경,로그 UI 통일화,리스트뷰에서통클<br>릭시기능추가(공유또는웹서버전송선택후실행)   |

|          |  |
|----------|--|
|          | <b>임베디드</b><br>사격시스템하드웨어부프로토타입테스트,전력분배문제해결중,모터드라이버동시제어문제해결중,센서부코딩문제해결중,GPIO 핀확장문제해결중(PI4J 라이브러리파일수정중)  |
| 2 월 19 일 | <b>안드로이드</b><br>안드로이드업로드기능합치기(앱간공유,웹서버로전송),안드로이드 Activity 간의이동중발생하는어플종료현상해결,gstreamer 를이용한녹화및재생기능완성<br><b>임베디드</b><br>사격시스템프로토타입테스트진행중(격발부분문제),전력분배문제해결,다중모터드라이버문제해결,배선을위한약간의차량몸체개조,핀확장문제해결(PI4J 의라이브러리를수정)<br><b>웹</b><br>웹디자인적용중 |
| 2 월 20 일 | 1 차 리허설 준비   |

## B. 개인 개발 일정

|     |  |
|-----|--|
|     | 01 월 05 일  |
| 문선균 | <p>아두이노 기본 개발환경을 설치. (아두이노 스케치) 서보 모터와 DC 모터에 대해 조사하고 각 모터의 구동 방법 및 명칭을 조사했으며 이들의 구동원리도 확인하였다.</p> <p>또한 파이썬 구동 환경을 구성하였으며 간단한 예제 몇 가지를 실행해 보았고 아두이노 스케치의 모터구동 예제들의 소스도 살펴보았다.</p>   |
| 서강원 | <p>파이썬 기본 문법 및 언어적 특성을 어느 정도 파악하고 내부 함수 및 유용한 모듈 몇가지를 살펴보았음</p> <p>안드로이드 개발 환경설정, 공부, 간단한 UI 설계</p> <ul style="list-style-type: none"> <li>- 리소스, 액티비티, 매니페스트, 인텐트에 대한 공부</li> <li>- 동영상 출력부분, 조작부분구현</li> <li>- 녹화 / 저장, 공유, 연결끊기 버튼구현</li> </ul> |
| 이호정 | <p>라즈베리 파이를 이용한 간단한 프로그램들을 통해 구동원리를 공부하였다. C 언어 기반으로된 프로그램을 위주로 찾아보았으나, 이후 파이썬으로 작성된 프로그램을 찾아서 공부를 할 예정이다.</p>   |



|     |   |
|-----|---|
| 김주혁 | 안드로이드 어플 UI 설계에 필요한 지식 책으로 공부<br>안드로이드 어플 기본 로딩 UI, 주화면 UI, 연결 UI 설계  |
| 신익환 | 안드로이드 어플리케이션 개발을 위해서 먼저 개발환경을 설정하고 간단한 프로젝트를 생성했다. 그리고 장치드라이버를 통해 휴대폰과 연동을 확인했다. 도서관에 가서 안드로이드 개발과 관련된 기초적인 책을 대여하여 어플리케이션 구성요소에 대한 공부와 프로젝트 생성 파일을 분석해 보았다. 어플리케이션 구성요소로서 리소스, 액티비티, 매니페스트, 인텐트에 대해 공부하고 간단한 액티비티 화면을 구성해 보았다.<br>먼저 LinearLayout 을 이용해서 동영상 출력부분, 조작부분으로 나누었고 동영상 출력부분은 일단 비워두었다. 그리고 조작부분은 녹화 / 저장, 공유, 연결끊기 버튼을 만들었고 RC 카 조작을 위한 방향키버튼도 만들었다. 그리고 어플리케이션을 실행하면 2 초동안 로딩화면이 뜰수 있게하였고 그리고 사용자 인터페이스에서 토스트를 이용한 알림 메시지, 프래그먼트 액션바에 대해 자세히 공부해 보았다. |
|     | 01 월 06 일   |
| 문선균 | 팀에서 사용할 서버를 구축 (FTP 서버)<br>개발 환경을 위한 파이썬 설치 및 이클립스와의 연동 확인<br>비품이 도착하지 않아 비품 세팅은 내일 하기로 함   |
| 서강원 | Hello android 수행<br>몇몇 회사 및 기기 별 wifi p2p 지원 여부 확인<br>wifi p2p 공부<br>우리의 프로젝트와 비슷한 프로젝트 관련 자료 조사  |
| 이호정 | 파이썬 언어에 대한 공부와 라즈베리 파이에서 파이썬을<br>사용해 개발한 프로그램 예제들을 찾아 공부하였다.  |
| 김주혁 | 안드로이드 지식공부 및 UI 기능 추가작업<br>- 위젯 아이콘 모양 변경 (VISION 팀의 상징 마크로 변경)<br>- 라즈베리 접속 Activity 설계<br>- 모의적으로 정해진 ID 를 입력시 연결이 되도록 만들어놓음.<br>- 동영상 출력부분에다가 임의적으로 외부 URI 를 연결하여 재생되도록 만들어놓음.<br>- 연결 종료시 라즈베리 접속을 해제하며 라즈베리 접속 Activity 로 다시 돌아감.  |

|     |  |
|-----|--|
|     | <ul style="list-style-type: none"> <li>- 현재 Application Flow = 인트로화면 &gt; 라즈베리 접속 화면 &gt; 주화면<br/>(연결종료 클릭시 주화면 &gt; 라즈베리 접속화면)</li> <li>- 애초에 계획했던 동영상 화면 부분을 녹화 할려는 계획에 차질이 생겼다.</li> </ul>  |
| 신익환 | 안드로이드 어플리케이션 구성요소 공부 및 UI 기능추가<br>Layout(종류,속성), Toast, ToggleButton, Activity 전환 에 대한 공부를 하고 조합하여서 UI 를 구현 하였다. 또 버 튼을 눌렀을 때 갤러리 어플리케이션으로 이동하여 저장된 사진이나 동영상을 공유할 수 있게 구현하였다. 그리고 다 른 방법으로 영상목록으로 이동하는 Activity 전환도 구현 하였다.  |
|     | 01 월 07 일  |
| 문선균 | 파일전송 서버 말고도 영상 스트리밍만을 위한 스트리밍 서버 개설 시도<br>(그러나, 학교의 아이피와 포트가 내부망이여서 외부에서의 접속이 막힘. 추가로 시도할 예정)<br>맥 지원을 받아 개발환경 구성중(내일중으로 윈도우 올릴 예정)<br>'산딸기마을' 의 '공개프로젝트 하나' 인 라즈베리파이와 멀티파이 및 아두이노шил드를 이용한 DC 모터제어 및 RC 카 제작 영상을 시청 (약 3 시간)<br>안드로이드 어플의 스트리밍 영상 저장 한계를 느껴 다른 아이디어를 제안 |
| 서강원 | 안드로이드 개발 환경 세팅, 안드로이드 어플에서 wifi p2p 를 set up 및 연결  |
| 이호정 | GstOpenMax 를 통해 영상을 H.264 로 인코딩 / 디코딩 하는 방법과 GstOpenMax 에 대한 전반적인 내용 공부, 맥북 환경 설정 및 안드로이드 개발 환경 설정   |
| 김주혁 | 네이티브 안드로이드 애플리케이션 개발을 위한 안드로이드 NDK 아 설치 및 개발 환경 설정   |
| 신익환 | 아이콘 이미지 변경, ListView 를 사용하여 동영상 목록을 나타낼 수 있는 기초적인 틀 구현, 연결 끊기 버튼 구현  |
|     | 01 월 08 일  |
| 문선균 | 안드로이드 조작부 UI 수정방안제시<br>기존 화살표 방식에서 조이스틱 방식으로 자유도를 높임 (각도방식으로써, 일정 각도로 조이스틱 패드를 움직이면 그 방향으로 나아가도록 함)<br>MATLAB 을 이용한 DC 모터 컨트롤러 구현  |

|           |   |
|-----------|---|
|           | <ul style="list-style-type: none"> <li>- pulse type 설정 후, amplitude, period, pulse width, (phase) delay 입력함</li> <li>- 위의 입력값대로 position, current, velocity 를 파형으로 표현</li> </ul>                  |
| 서강원       | 안드로이드에서 wifi p2p 모듈을 작동시키는 부분까지 성공함. 다만 안드로이드 기기는 이 모듈이 작동하는지 아직 확인하지 못함. 따라서 내일 이 부분에 대하여 처리할 계획   |
| 이호정       | 바이너리 영상 파일을 인코딩, 디코딩하는 다양한 방법, 영상의 포맷을 변경하는 방법, GstOpenMax 를 사용하여 지연을 최소화한 영상 전송 예제에 대하여 공부   |
| 김주혁       | 안드로이드 NDK 구축은 완료했으나 빌드 부분에서 문제 발생하여 보류<br>-안드로이드 쓰레드와 서비스 공부<br>-안드로이드 동영상 목록 구현  |
| 신익환       | 안드로이드 UI 기능추가<br>- 내부저장소 모든 동영상목록 불러오기<br>(파일이름,형식,크기,저장경로)<br>- 동영상이름 클릭시 동영상재생<br>(일시정지,빨리감기,되감기,재생시간)<br>- 공유하기 버튼 구현<br>(미완성:파일선택 X)  |
| 01 월 09 일 |   |
| 문선균       | RC 카 분해 및 회로도 분해작업<br>라즈베리파이 세팅 및 카메라 세팅<br>서보모터를 이용한 틸트/팬 카메라 제어 HW 조립<br>파이카메라의 구동 확인<br>맥 개발환경 구성(듀얼부트)  |
| 서강원       | wifi p2p 테스트로 안드로이드 기기에서 지원이 부족함을 확인함 따라서 다른 방식을 조사 및 공부  |
| 이호정       | 라즈베리 파이에 라즈비안 설치 및 카메라 모듈 설치. 카메라 모듈을 통해 사진 촬영, 동영상 촬영이 가능한지 테스트. 무선랜 설정 및 유선랜 설정. 하지만 유선랜을 사용했을때 ip 를 확인할 수 없어서 우선 무선랜을 통해 작업을 하기로 함. 맥북에서 netcat 을 통해 라즈베리 파이에 접속하려 하였으나 netcat 에서 문제가 생겨 이 부분에 |

|           |   |
|-----------|---|
|           | 대해서는 내일 추가적으로 고민하기로 함.  |
| 김주혁       | 안드로이드 앱<br>-wifi 활성화 및 wifi 자동연결 기능 구현<br>- wifi direct 테스트 결과 안드로이드 기기 별 호환의 문제 및 지원이 제대로 이루어지지않아 다른 방식을 조사<br>- 라즈베리파이 파이카메라 기본활성화 설정 및 라즈베리파이 공부   |
| 신익환       | 안드로이드 UI 기능수정(공유기능) 및 오류검사<br>1.공유기능 완성<br>- AdapterView 를 사용하여 리스트의 Column 들이 각각의 이벤트를 갖게 함<br>- OnItemLongClickListener 를 사용하여 Column 을 길게 눌렀을 때 공유할 수 있는 앱이 나타나게 함<br>2.오류검사<br>- 리스트뷰에서 getView 가 중복호출되는 문제 해결<br>- 어플 실행 시 로딩화면이 두번 호출되는 문제 해결 |
| 01 월 10 일 |   |
| 문선균       | 서보모터 제어 구현 완료(그러나 점퍼케이블 및 브레드보드가 없어 실제 제어 확인은 못함. 제어 UI 및 제어 기능은 구현완료<br>영상 스트리밍의 가장 기본적인 확인을 위해 gstreaming 을 이용해 스트리밍을 확인하고자 하였으나 문제발생   |
| 서강원       | 라즈베리파이 설정을 조사하고 시도함   |
| 이호정       | 라즈베리파이 설정을 조사하고 시도함   |
| 김주혁       | 휴일  |
| 신익환       | 휴일  |
| 01 월 11 일 |   |
| 문선균       | 접속원인이 학교망이라고 잠정적 결론   |
| 서강원       | 접속원인이 학교망이라고 잠정적 결론   |
| 이호정       | 접속원인이 학교망이라고 잠정적 결론   |
| 김주혁       | 여러 방법의 원격시도. 그러나 모두 실패하고 사설망만 성공  |

|     |  |
|-----|--|
| 신익환 | 안드로이드 UI 지속적인 보수   |
|     | 01 월 12 일  |
| 문선균 | 서보모터 제어 구현 완료. 테스트 및 코드 완성<br>시험도 성공적. 다소의 소음과 진동이 발생하였으나 제대로 돌아감을 확인하였음.<br>아두이노 가세팅 완료. (내일 납땜작업 예정)<br>스트리밍을 위한 Gstreaming 은 오늘도 성공하지 못함          |
| 서강원 | 1. servo 모터 컨트롤 코딩(python)<br>2. 통신 방식 찾기<br>3. Gstreamer 설치 시도 in raspbian  |
| 이호정 | Gstreamer 를 이용하여 실시간 영상 전송을 구현하기 위해 Gstreamer 설치 시도.<br>그러나 Gstreamer 설치에 실패   |
| 김주혁 | wifi 수동연결을 위한 wifi scan 목록 구현 완료<br>wifi hotspot 구현을 위한 개념 및 소스 분석   |
| 신익환 | MySQL,아파치서버,PHP 설치 및 연동<br>안드로이드 MySQL 데이터 주고받는<br>프로그램  |
|     | 01 월 13 일  |
| 문선균 | 1. 아두이노 키트 납땜 / 각종 센서 납땜 (온/습도, 적외선, 자이로, 스케치로더)<br>2. 서보모터 제어 해결중 (조원 서강원이 전력 부족 문제를 해결하려 노력중)<br>3. 차량 DC 모터 제어 성공 (기본제어)<br>p.s. 서보모터는 절대 포기할수 없음 |
| 서강원 | 1. 서보 모터 관련 라즈베리 파이 전원 종료 원인 찾기<br>2. 서버 관련 정보 조사  |
| 이호정 | gstreamer 를 통한 영상 전송 확인 및 mjpg streaming 을 통한 영상 전송 확인.<br>gstreamer 명령어를 간단하게 수행할 수 있는 스크립트를 작성.  |

|           |   |
|-----------|---|
| 김주혁       | 1. mjpg streamer 설치 및 원격 영상 스트리밍 성공<br>2. gstreamer 설치 및 원격 영상 스트리밍 성공<br>3. 아두이노 부품 및 스케치 로더 및 각종 센서 납땜  |
| 신익환       | 안드로이드 개발 (MySQL 과 연동하는 로그인 화면)<br>MySQL 계정생성 및 회원정보 테이블 작성<br>Members 테이블에서 Id,Password 를 불러오는 PHP 파일 작성<br>안드로이드와 웹서버 연동(개발중)                    |
| 01 월 14 일 |   |
| 문선균       | 비트 프로젝트 발표회 견학  |
| 서강원       | 비트 프로젝트 발표회 견학  |
| 이호정       | 비트 프로젝트 발표회 견학  |
| 김주혁       | 비트 프로젝트 발표회 견학  |
| 신익환       | 비트 프로젝트 발표회 견학  |
| 01 월 15 일 |   |
| 문선균       | 1.라즈베리파이 원격 제어환경 2 가지 구성(XRDP, SAMBA)<br>2.Arch Linux 설치 시도 (실패), XBMC 설치 (설치는 완료, 실행 불가)<br>3. DC 모터 제어 코드 작성<br>4. 제어를 위한 웹페이지 작성               |
| 서강원       | 1. gst-RTSP 관련 조사<br>2. 서버 프로그래밍 공부<br>3. 안드로이드 vlc 관련 코드 분석  |
| 이호정       | Gstreamer 를 이용한 녹화 및 h.264 파일을 mp4 파일로 변환하는 방법을 찾아서 공부하였으며, gstreamer 를 이용해 웹캠 영상을 안드로이드 어플에서 볼 수 있는 예제를 찾아 실행 가능한지 시도해보았으나 실패하였음. 물품 견적서 초안을 작성. |
| 김주혁       | 파이카메라로 웹스트리밍 할때 전송되는 uri 를 알아내는 방법을 공부<br>라즈베리파이에서 웹 상의 다음팟 플레이어로 영상 스트리밍 받기<br>라즈베리파이에서 안드로이드 vlc 프로그램으로 영상 스트리밍 받기                              |
| 신익환       | 안드로이드 로그인 페이지 오류수정<br>외부 PC 로 교내 ip 접근이 불가하여 외부에 웹서버 및 DB 서버 설치   |

|     |   |
|-----|---|
|     | 01 월 16 일   |
| 문선균 | DC 모터 원격제어 시도. 라즈베리파이와 안드로이드와의 통신 자체는 성공하였지만 제어가 안됨. 과전류로 인한 보드 손상을 초래할 뻔함.               |
| 서강원 | gstreamer 개발자 관련 조사<br>파이선 소켓 조사  |
| 이호정 | 라즈베리파이에서 gstreamer 를 이용하여 영상을 녹화하는 방법을 연구   |
| 김주혁 | 라즈베리파이 파이카메라를 g streamer 를 이용해 안드로이드 개발 앱으로 영상찍우기<br>방법 연구                                |
| 신익환 | 안드로이드 개발<br>안드로이드 MySQL 을 연동한 로그인화면 구현<br>회원가입화면 구현중(테이블저장,id 중복확인)                       |
|     | 01 월 17 일   |
| 문선균 | 주말 휴일   |
| 서강원 | 주말 휴일   |
| 이호정 | Gstreamer 설치 및 연동과 세팅   |
| 김주혁 | 안드로이드 핫스팟 기능 구현 공부<br>안드로이드 어플내에서 핫스팟 이름, pw 설정 및 활성화 구현 (10 프로 구현)                       |
| 신익환 | 회원가입 페이지 구현(ID 중복확인)<br>오류 수정(PHP 파일에서 Post 방식으로 값을 받을 때 두번째 부터 NULL 값이 반환되는 것)           |
|     | 01 월 18 일   |
| 문선균 | 주말 휴일   |
| 서강원 | 주말 휴일   |
| 이호정 | Gstreamer 설치 및 연동과 세팅   |
| 김주혁 | 안드로이드 핫스팟 기능 구현 공부<br>안드로이드 어플내에서 핫스팟 이름, pw 설정 및 활성화 구현 (10 프로 구현)                       |
| 신익환 | 안드로이드 로딩,로그인,회원가입 화면 디자인<br>회원정보 공백 입력시 메시지 발생, 회원가입하고 Main 으로 갈때 Activity가 종료되지 않는 문제 해결 |

|     |   |
|-----|---|
|     | 01 월 19 일   |
| 문선균 | DC 모터 제어부 제작중   |
| 서강원 | 안드로이드 ndk 및 gstreamer sdk 설치  |
| 이호정 | 안드로이드 환경에서 gstreamer 를 이용한 영상 전송이 가능한지 확인하기 위해서 이클립스 및 NDK 등 환경 설정. 윈도우에서 gstreamer 를 통해 영상을 볼 수 있는지에 대한 공부. 안드로이드에서 gstreamer 를 어떻게 사용하는지에 대한 공부   |
| 김주혁 | 안드로이드 핫스팟 기능 구현 및 안드로이드 sdktools /ndk 설치  |
| 신익환 | 안드로이드 소스코드 병합 및 수정<br>Eclipse 에서 PHP 개발환경 설치<br>웹페이지 개발(db 접속, 회원가입폼)   |
|     | 01 월 20 일   |
| 문선균 | JDK8 을 라즈베리파이에 설치하여 TCP/IP 기반의 Server-Client 를 이용하는 DC 모터 제어 환경 구축  |
| 서강원 | 1. gstreamer sdk 튜토리얼 테스트<br>2. gstreamer tutorial 공부<br>3. PiCamera test   |
| 이호정 | 이클립스에서 gstreamer 사용이 가능한 환경 구축 및 안드로이드에서 gstreamer tutorial 5 가지 성공   |
| 김주혁 | 1.gstreamer sdk 설치<br>2.gstreamer 튜토리얼 테스트<br>3.웹뷰로 스트리밍테스트   |
| 신익환 | 안드로이드 SDK tools/NDK 설치<br>NDK 기초적인 이론공부<br>DB 연동한 웹페이지 개발중(로그인폼)  |
|     | 01 월 21 일   |
| 문선균 | 스트리밍 파트의 완성이 아직 안된 점을 미루어 조작부와의 연동을 어떻게 할지 찾아봄. 화면이 나옴과 동시에 조작부의 작동도 실행되도록 하며 조작부의 통신이 스트리밍되는 영상 통신에 영향을 주지 않도록하며, 마찬가지로 스트리밍되는 영상 통신이 조작부의 통신에 영향을 주지 않도록 한다.<br>당장 구동 및 연동은 힘들지만 내일 안에 어떻게든 끝내야 함을 인지. 생각처럼 |



|     |  |
|-----|--|
|     | 작업이 진행되지 않음  |
| 서강원 | gstreamer tutorial 5 이용 http 방식 스트리밍 재생<br>자바 소켓 서버, 클라이언트간 간단한 메시지 전송 프로그램 작성   |
| 이호정 | Gstreamer tutorial 5 를 이용하여 안드로이드에서 http 영상 재생이 가능한 것을 확인,<br>mjpg 스트리밍이 가능한지 확인, gstreamer rtsp 예제를 찾아 실행시키려 하였으나<br>실패 |
| 김주혁 | 안드로이드 mjpg streamer 를 온라인 firefox, 핸드폰 firefox 어플로 영상스트리밍<br>성공(자체적인 어플에서는 실패)   |
| 신익환 | 라즈베리파이 폴더접근방법 조사 및 설치(연동 X)<br>(FTP server 또는 samba 이용)<br>웹페이지 개발진행   |
|     | 01 월 22 일  |
| 문선균 | 앞, 뒤의 제어 모터 제어부를 완전한 형태로 코딩 (자바와 파이썬의 합 형태로 연동.<br>쓰레드 방식을 사용함) (정)<br>안드로이드 UI 와 연동 (부)<br>Gstreamer 의 연동 (부)           |
| 서강원 | gstreamer 안드로이드 어플 수정<br>php 관련 문제 수정  |
| 이호정 | 중간발표 피피티 작성, 안드로이드 어플 병합   |
| 김주혁 | 안드로이드 어플 기능 합치기<br>DC 모터 제어 기능 구현<br>프로토타입 기능 촬영   |
| 신익환 | 프로토타입 PPT 부분제작(안드로이드)<br>웹에서 경로지정을 통하여 디렉토리 와 파일목록 불러오는 기능구현<br>(현재폴더 이외의 경로는 실행 X)                                      |
|     | 01 월 23 일  |
| 문선균 | 프로토 타입 발표 및 보완   |
| 서강원 | 프로토 타입 발표 및 보완   |

|     |   |
|-----|---|
| 이호정 | 프로토 타입 발표 및 보완  |
| 김주혁 | 프로토 타입 발표 및 보완  |
| 신익환 | 프로토 타입 발표 및 보완  |
|     | 01 월 24 일   |
| 문선균 | 역할 재분담 완료 및 향후 계획 토의, 필요한 부품 다시 체크, 기획서 등의 문서작업 시작                                      |
| 서강원 | 기능정의서 수정  |
| 이호정 | 프로젝트 기획서 수정   |
| 김주혁 | 향후 대책 회의 2<br>SW-UI_HW 외부구조 설계서 수정(문서화작업)   |
| 신익환 | 테이블 정의서 수정  |
|     | 01 월 25 일   |
| 문선균 | 자체 휴일   |
| 서강원 | 자체 휴일   |
| 이호정 | 자체 휴일   |
| 김주혁 | 자체 휴일   |
| 신익환 | 자체 휴일   |
|     | 01 월 26 일   |
| 문선균 | 카메라 팬/틸트 제어를 위한 서보모터 제어부 제작 시작<br>환경센서보드 제작을 위한 아두이노작업 (문제발생)<br>환경센서보드 제작을 위한 추가 물품 구입 |
| 서강원 | Gstreamer 연구 및 공부   |
| 이호정 | 문서 작성   |
| 김주혁 | SW-UI_HW 외부구조 설계서 수정(문서화작업-완료)<br>아두이노 납땜   |
| 신익환 | 1. 웹에서 경로지정을 통하여 디렉토리나 파일목록 불러오는 기능구현(현재폴더만 접근가능한 문제해결, 파일읽기 가능)                        |

|     |   |
|-----|---|
|     | 2. 스마트폰 핫스팟에 접속한 라즈베리파이와 맥북이 연동되는 것을 확인<br>3. SW-UI-HW 외부구조설계서 초안작성, 기능정의서(DB,웹페이지)작성 |
|     | 01 월 27 일   |
| 문선균 | 환경센서보드 제작<br>환경센서보드 테스트<br>서보모터제어모듈 설치<br>서보모터 (팬/틸트) 제어 테스트<br>서보모터 (팬/틸트) 제어부 제작중   |
| 서강원 | 1. gstreamer 공부<br>2. gstreamer ndk 시도  |
| 이호정 | 문서 작업   |
| 김주혁 | 환경센서 보드 제작을 위한 아두이노 납땜<br>안드로이드 NDK 설치완료 및 기본예제 실행<br>Gstreamer NDK 시도                |
| 신익환 | 라즈베리파이 고정 ip 갖는방법 조사<br>Windows, MySQL, Apache, PHP 설치(맥북), Cygwin, NDK 설치(PC) C143   |
|     | 01 월 28 일   |
| 문선균 | 조도센서 테스트<br>적외선 센서 테스트<br>서보모터 제어부 제작중  |
| 서강원 | 1. gstreamer android sdk 구성 성공<br>2. gstreamer plugin 개발 문서 공부 시작                     |
| 이호정 | Gstreamer http streaming 조사 및 라즈베리파이에 http-launch 에 필요한 파일 설치                         |
| 김주혁 | 1. gstreamer android sdk 구성 성공<br>2. gstreamer plugin 개발 문서 공부 시작                     |
| 신익환 | PHP 이론공부 및 전체적인 웹 UI 구상   |
|     | 01 월 29 일   |
| 문선균 | 카메라 모듈의 촬영을 돕기위한 팬/틸트 서보모터 제어부 완성   |

|           |  |
|-----------|--|
| 서강원       | 지스트리머 공부 및 테스트   |
| 이호정       | gststreamer mp4 streaming 공부   |
| 김주혁       | 지스트리머 공부 및 테스트   |
| 신익환       | HTML,JavaScript 공부 및 웹개발   |
| 01 월 30 일 |  |
| 문선균       | 쥬스트리머 안드로이드 연동방법 검색 (논문 정독)  |
| 서강원       | 회의 안드로이드 .h264 재생 보조   |
| 이호정       | gststreamer 설치 및 http-launch 설치(실패)  |
| 김주혁       | 교수님과 면담을 위한 준비 회의 ( 질문 준비 및 시각화 자료)<br>안드로이드 어플리케이션에 .h264 포맷인 영상 재생 성공      |
| 신익환       | 웹개발(게시판,자료실)   |
| 01 월 31 일 |  |
| 문선균       | 월요일 교수님면담 관련 피피티 제작<br>온도/습도 센서 코딩에 관한 계산 수식 공부 및 간단한 모의코드제작                 |
| 서강원       | gsocket 공부 및 실습  |
| 이호정       | gst-launch 공부  |
| 김주혁       | 정기 휴일  |
| 신익환       | 정기 휴일  |
| 02 월 01 일 |  |
| 문선균       | 정기 휴일  |
| 서강원       | 정기 휴일  |
| 이호정       | 정기 휴일  |
| 김주혁       | 정기 휴일  |
| 신익환       | 정기 휴일  |
| 02 월 02 일 |  |
| 문선균       | 서광덕 교수님께 자문구하러 미팅<br>움직임 감지 적외선 센서 테스트 (오작동)<br>라즈베리파이 이상으로 공장초기화 및 시스템 다시올림 |

|     |  |
|-----|--|
| 서강원 | 교수님과 면담<br>팀회의<br>유닉스 파이프라인 확인   |
| 이호정 | 조퇴   |
| 김주혁 | 교수님과의 상담<br>조별 회의<br>라즈베리파이 재설치 및 파이프 라인 standard in, out 데이터 교환   |
| 신익환 | 자료실개발(파일복사하여 폴더에 저장 및 읽기,테이블에 파일정보저장)  |
|     | 02 월 03 일  |
| 문선균 | PIR (적외선 인체감지센서) 제어 코드 작성 (using python)<br>조도센서 제어 코드 작성 (using python)<br>온습도센서 제어 코드 작성 (using C)<br>라즈베리파이에 JDK8 재설치<br>라즈베리파이에 wiringPi 재설치 |
| 서강원 | 이미지 파일 바이너리 값 확인   |
| 이호정 | gststreamer rtp streaming 조사, streaming-video-rtsp-rtp 조사 및 소스코드 분석  |
| 김주혁 | 라즈베리파이 gststreamer 재설치<br>gststreamer rtsp 를 이용한 뷰로 띄우기 부분 실패(설정부분 오류)   |
| 신익환 | 웹(자료실,게시판완성 및 메인과 연동)  |
|     | 02 월 04 일  |
| 문선균 | GStreamer 윈도우 버전을 안드로이드로 변환하기 위한 자료조사 및 코드분석   |
| 서강원 | gststreamer 0.10 버전에서 1.0 버전으로 변경하는 방법에 대하여 공부 및 시도  |
| 이호정 | Gstreamer 에 대한 자료조사  |
| 김주혁 | 안드로이드 gststreamer 0.10 버전에서 1.0 버전으로 porting (변경 성공)   |
| 신익환 | 웹페이지(프로토타입)구현  |

|     |   |
|-----|---|
|     | 02 월 05 일   |
| 문선균 | PIR (적외선 인체감지센서) 제어 코드 재코딩 (python -> java)<br>제어코드 기반 서버코드에 융합 성공 (컴파일 성공)<br>자이로센서 제어 준비중 (리눅스 커널 문제) |
| 서강원 | 안드로이드 gstreamer 어플리케이션  |
| 이호정 | Gstreamer 에 대한 자료조사   |
| 김주혁 | 안드로이드 gstreamer pipeline 구조 파악<br>안드로이드 gstreamer pipeline 일부 구성완료                                      |
| 신익환 | 아파치에서 네트워크 드라이브를 DocumentRoot 로 지정하는 방법조사<br>Samba 를 계정없이 사용하는 방법조사                                     |
|     | 02 월 06 일   |
| 문선균 | 자이로 센서 제어를 위해 I2C 공부<br>자이로 센서 제어를 위한 I2C (커널 시리얼 통신) 설치 (실패)   |
| 서강원 | gstreamer client 구현중  |
| 이호정 | 라즈베리파이에서 gstreamer 를 이용하여 스트리밍과 동시에 영상 저장 구현 중  |
| 김주혁 | gstreamer receive 쪽 파이프라인 완벽 구현 진행중(rtp, udp 사용)  |
| 신익환 | 아파치 홈디렉토리를 네트워크 드라이브로 하려고 시도 했으나 진전 없음<br>따라서 파이에(Apache,php,Mysql)설치 후 putty 로 접근하는 방법 조사              |
|     | 02 월 07 일   |
| 문선균 | 자이로센서 활성화 방법 모색중<br>Gstreamer 안드로이드 연동을 위한 자료조사<br>PIR 센서 코드 수정<br>I2C 포트 사용방법 찾는중                      |
| 서강원 | gstreamer receive 쪽 파이프라인 완벽 구현 진행중(80%완성)  |
| 이호정 | 정기 휴무   |
| 김주혁 | gstreamer receive 쪽 파이프라인 완벽 구현 진행중(80%완성)  |

|     |  |
|-----|--|
| 신익환 | 라즈베리파이에 Apache,PHP,MySQL 설치 및 연동   |
|     | 02 월 08 일  |
| 문선균 | Gstreamer 개발 보좌 (부)<br>전반적인 모든 하드웨어 세팅<br>서보모터와 모터드라이버의 이상작동 (고장)<br>I2C 포트의 자문을 구함 (답변을 얻음) |
| 서강원 | gstreamer receive 쪽 파이프라인 완벽 구현 성공   |
| 이호정 | 정기 휴무  |
| 김주혁 | gstreamer receive 쪽 파이프라인 완벽 구현 성공   |
| 신익환 | Apache,PHP,MySQL 연동 성공   |
|     | 02 월 09 일  |
| 문선균 | 발표   |
| 서강원 | 발표   |
| 이호정 | 발표   |
| 김주혁 | 발표   |
| 신익환 | 발표   |
|     | 02 월 10 일  |
| 문선균 | 온습도 센서 활성화 성공<br>자이로 센서 활성화 성공<br>3d 가상 시뮬레이션 모델 제작중   |
| 서강원 | 녹화 관련 공부 및 조사  |
| 이호정 | 녹화 기능 구현을 위한 공부  |
| 김주혁 | 핫스팟 Host Ip 주소 확인<br>gstreamer 동적 파이프라인 변경 공부  |
| 신익환 | 라즈베리파이 php mysql 연동<br>웹상에서 원하는 확장자만 불러오는 방법 조사 및 코드 분석                                      |
|     | 02 월 11 일  |
| 문선균 | 펜/틸트 수리 완료   |

|     |  |
|-----|--|
|     | <p>자이로 센서 3D 시뮬레이션 제작중</p> <p>새로운 계획안 수립</p> <p>새로운 계획에 대한 부품 및 구성도 제작중</p>  |
| 서강원 | <p>tee 공부</p> <p>requeset pad 공부</p> <p>ndk native methon 생성</p> <p>클라이언트부 파이프라인 구조 변경</p>   |
| 이호정 | <p>녹화 기능, 웹디자인을 위한 자료조사</p>  |
| 김주혁 | <p>2 차 발표 이후 회의</p> <p>서보모터 재구매 이후 서보모터 하단부 연결 부분 제작</p> <p>동영상을 저장할 수 있고 안드로이드와 연동되는 db 선택</p>  |
| 신익환 | <p>웹에서 한 디렉토리 내 원하는 확장자(ex: mp4)만 호출</p> <p>녹화파일 저장시 파일이름에 관리자가 지정해준 문자열(일반적으로 사용자이름 이니셜, ex: sih)을 포함할 시 개인별 영상 불러오기 가능, 다른 회원파일에는 접근불가 (단 관리자는 회원들이 가입할 때마다 회원별 코드를 추가해줘야 함)</p> |
|     | 02 월 12 일  |
| 문선균 | <p>새로운 기능 추가를 위한 부품 수급을 위해 서울 외근</p> <p>새로운 기능 추가를 위한 부품 분해 및 재조립</p> <p>추가되는 서보모터의 전원공급제어 및 테스트</p> <p>새로운 기능을 위한 포맥스 기반의 하우징 제작중 (1)</p>   |
| 서강원 | <p>녹화기능 구현 80% 완성</p>  |
| 이호정 | <p>웹디자인</p>  |
| 김주혁 | <p>새로운 기능 추가를 위한 부품 수급을 위해 서울 외근</p> <p>새로운 기능 추가를 위한 부품 분해 및 재조립</p> <p>추가되는 서보모터의 전원공급제어 및 테스트</p> <p>새로운 기능을 위한 포맥스 기반의 하우징 제작중 (1)</p>   |



|     |   |
|-----|---|
| 신익환 | 안드로이드에 웹서버,PHP,MYSQL 설치후 외부 pc 에서 연동하고 php5.5 버전에 맞게 수정중  |
|     | 02 월 13 일   |
| 문선균 | 파이썬을 이용한 3D 자이로센서 시뮬레이션 제작<br>터렛의 좌 우 360 도 제어를 위한 아래 서보모터 개조<br>터렛 본체 제작 도움 (부)<br>서보모터 제어부분 코드 개선 |
| 서강원 | 녹화기능 구현 완성 및 스트리밍 기능을 적절히 호환되게 개선중  |
| 이호정 | 웹 디자인 및 어플 디자인을 위한 자료조사   |
| 김주혁 | 안드로이드 어플리케이션 UI 개선<br>터렛 본체 제작 (정)  |
| 신익환 | 핫스팟 범위내에서의 웹 접속을 위한 UI 및 기능 개선  |
|     | 02 월 14 일   |
| 문선균 | 주말 휴일   |
| 서강원 | 녹화기능 구현   |
| 이호정 | 웹 디자인 및 어플 디자인을 위한 자료조사   |
| 김주혁 | 안드로이드 동영상 목록, 재생  |
| 신익환 | 안드로이드에서 웹서버로 파일전송 예제 코드 분석  |
|     | 02 월 15 일   |
| 문선균 | 주말 휴일   |
| 서강원 | 어플리케이션 상에서의 재생 기능 구현을 위한 공부   |
| 이호정 | 웹 디자인 및 어플 디자인을 위한 자료조사   |
| 김주혁 | 안드로이드 UI 전몇 교체 및 보완과 안정화  |
| 신익환 | 휴일  |
|     | 02 월 16 일   |
| 문선균 | PIR(적외선인체감지센서) 파이썬 코드를 자바로 재코딩, 조도 센서 및 자이로 센서의<br>파이썬 코드를 자바로 변환하는 작업 진행중                          |

|     |   |
|-----|---|
|     | 사격시스템에 필요한 하드웨어 제작중   |
| 서강원 | gstreamer android player 시도   |
| 이호정 | 웹디자인 샘플페이지 작성   |
| 김주혁 | 안드로이드 동영상 목록 , 재생 재수정<br>안드로이드 이미지 최종 수정<br>모터 총 서보모터 하단부와 기동 제작  |
| 신익환 | 안드로이드 웹서버로 단일파일전송(텍스트 O, 이미지,영상은 X), 안드로이드 회원별<br>영상목록 불러오는 방법 구상중  |
|     | 02 월 17 일   |
| 문선균 | PIR 센서 연동 실패<br>조도센서 제어프로그램 코딩중<br>사격시스템 하드웨어부 프로토타입 완성 (상/하, 좌/우 구동 확인)<br>차량의 전반적인 배선 보완                          |
| 서강원 | 안드로이드에서 스트리밍 영상을 일정구간 녹화  |
| 이호정 | 웹디자인  |
| 김주혁 | 사격 시스템 완성 (상,하,좌,우 제어 성공, 단일 사격 성공)   |
| 신익환 | 안드로이드에서 웹서버로 단일파일전송,로그인,회원가입 UI 와 주혁이형 UI 병합  |
|     | 02 월 18 일   |
| 문선균 | 사격시스템 하드웨어부 프로토타입 테스트<br>전력분배 문제 해결중<br>모터드라이버 동시제어 문제 해결중<br>센서부 코딩 문제 해결중<br>GPIO 핀 확장 문제 해결중 (PI4J 라이브러리 파일 수정중) |
| 서강원 | 안드로이드에서 스트리밍 영상을 일정구간 녹화  |

|     |  |
|-----|--|
| 이호정 | 휴일   |
| 김주혁 | 안드로이드 동영상 목록, 재생 2 차 재구현   |
| 신익환 | 리스트뷰에서 롱클릭할 때 기능 추가(팝업메뉴나오고 공유 또는 웹서버로 전송 선택후 실행)  |
|     | 02 월 19 일  |
| 문선균 | 사격시스템 프로토타입 테스트 진행중 (격발부분 문제)<br>전력분배 문제 해결<br>다중 모터드라이버 문제 해결<br>배선을 위한 약간의 차량 몸체 개조<br>핀 확장 문제 해결 (PI4J 의 라이브러리를 수정) |
| 서강원 | 안드로이드에서 영상파일 재생  |
| 이호정 | 웹 디자인 추가 및 변경  |
| 김주혁 | 안드로이드 업로드 기능 합치기(앱간의 공유, 웹서버로 전송)<br>안드로이드 Activity 간의 이동중 발생하는 어플 종료 현상 해결<br>RC 차량에 모터 드라이버 1 개 추가(현재 드라이버 2 개로 늘어남) |
| 신익환 | PPT 1 페이지 제작, 웹디자인   |
|     | 02 월 20 일  |
| 문선균 | 발표 준비 및 프로필 촬영, 문서작업   |
| 서강원 | 발표 준비 및 프로필 촬영, 문서작업   |
| 이호정 | 발표 준비 및 프로필 촬영, 문서작업   |
| 김주혁 | 발표 준비 및 프로필 촬영, 문서작업   |
| 신익환 | 발표 준비 및 프로필 촬영, 문서작업   |

## 4. 프로젝트 개발 내용

### A. 프로젝트 배경 지식/기술/알고리즘

#### I. Database

##### 1. Mysql

표준 데이터베이스 질의 언어인 SQL(Structured Query Language)을 사용하는 개방 소스의 관계형 데이터베이스 관리 시스템(RDBMS). 매우 빠르고, 유연하며, 사용하기 쉬운 특징이 있다. 다중 사용자, 다중 스레드를 지원하고, C, C++, Eiffel, 자바, 펄, PHP, Python 스크립트 등을 위한 응용 프로그램 인터페이스(API)를 제공한다. 유닉스나 리눅스, 윈도 운영 체제 등에서 사용할 수 있다. LAMP, 즉 리눅스 운영 체제와 Apache 서버 프로그램, MySQL, PHP 스크립트 언어 구성은 상호 연동이 잘되면서도 오픈 소스로 개발되는 무료 프로그램이어서 홈 페이지나 쇼핑몰 등 일반적인 웹 개발에 널리 이용되고 있다. mySQL은 오픈 소스 라이선스를 따르기는 하나 상업적으로 사용할 때에는 상업용 라이선스를 구입해야 한다.

#### II. Hardware

##### 1. Raspberry PI

라즈베리 파이(영어: Raspberry Pi)는 영국의 라즈베리 파이 재단이 학교에서 기초 컴퓨터 과학 교육을 증진시키기 위해 만든 싱글 보드 컴퓨터이다. 라즈베리 파이는 그래픽 성능이 뛰어나면서도 저렴한 가격(세금을 포함하지 않은 모델 A의 경우 25달러, 모델 B, 모델 B+, 2 모델 B의 경우 35달러)라는 특징을 갖고 있다.

라즈베리 파이 2를 제외한 라즈베리 파이 모델들은 브로드컴의 BCM2835 단일 칩 시스템을 사용하며, 이 칩에는 ARM1176JZF-S 700 MHz 싱글코어프로세서(일반 데스크탑은 보통 2500 MHz~3500 MHz), 비디오코어 IV VGA와 512 메가바이트 RAM이 들어 있다. 그리고 라즈베리 파이의 프로세서는 오버클럭시 최대 1000 MHz까지의 성능을 발휘 할 수 있으며, 하드 디스크 드라이브나 솔리드 스테이트 드라이브를 내장하고 있지 않으며, SD 카드(B+, 2 B+ 모델은 Micro SD Card를 사용)를 외부 기억장치로 사용한다. 새로출시한 2 모델 B는 ARM Cortex-A7 0.9GHz프로세서와 램용량이 1GB로 성능이 업그레이드되어 출시 되었다. 라즈베리 재단은 컴퓨터 교육 증진을 위해 2가지 모델을 내놔으며, 각각 25달러와 35달러로 책정되었다. 2012년 2월 29일 재단에서 35달러짜리 모델의 주문을 받기 시작하였다.

또한 라즈베리 파이 재단측에서는 라즈베리 파이에 포팅한 데비안과 아치 리눅스, QtonPi등의 리눅스 배포판을 제공하고 있다. 그리고 재단측에서는 라즈베리 파이의

주 프로그래밍 언어인 파이썬에 BBC 베이직, C, 펄등을 지원할 수 있는 툴을 계획 중이다.

이번 프로젝트에서는 라즈베리파이 B+ 보드를 사용하여 개발을 진행하였다.



# Raspberry Pi



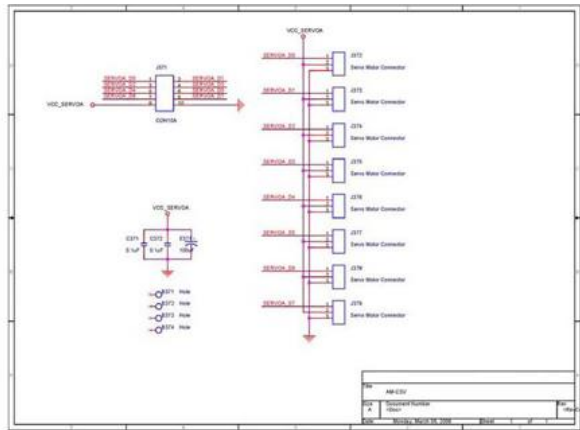
## MODEL B+

| Product Name        | Raspberry Pi Model B+  |
|---------------------|--|
| Product Description | The Raspberry Pi Model B+ incorporates a number of enhancements and new features. Improved power consumption, increased connectivity and greater IO are among the improvements to this powerful, small and lightweight ARM based computer. |

|                          |   |
|--------------------------|---|
| <b>Specifications</b>    |   |
| <b>Chip</b>              | Broadcom BCM2835 SoC  |
| <b>Core architecture</b> | ARM11   |
| <b>CPU</b>               | 700 MHz Low Power ARM1176JZFS Applications Processor  |
| <b>GPU</b>               | Dual Core VideoCore IV® Multimedia Co-Processor<br>Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode<br>Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure |
| <b>Memory</b>            | 512MB SDRAM   |
| <b>Operating System</b>  | Boots from Micro SD card, running a version of the Linux operating system   |
| <b>Dimensions</b>        | 85 x 56 x 17mm  |
| <b>Power</b>             | Micro USB socket 5V, 2A   |
| <b>Connectors:</b>       |   |
| <b>Ethernet</b>          | 10/100 BaseT Ethernet socket  |
| <b>Video Output</b>      | HDMI (rev 1.3 & 1.4)<br>Composite RCA (PAL and NTSC)  |
| <b>Audio Output</b>      | 3.5mm jack, HDMI  |
| <b>USB</b>               | 4 x USB 2.0 Connector   |
| <b>GPIO Connector</b>    | 40-pin 2.54 mm (100 mil) expansion header: 2x20 strip<br>Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines  |
| <b>Camera Connector</b>  | 15-pin MIPI Camera Serial Interface (CSI-2)   |
| <b>JTAG</b>              | Not populated   |
| <b>Display Connector</b> | Display Serial Interface (DSI) 15 way flat flex cable connector with two data lanes and a clock lane  |
| <b>Memory Card Slot</b>  | SDIO  |

## 2. [AM-CSV] 서보모터 제어모듈

8개의 서보모터를 제어 할 수 있는 모듈이다. 포트의 출력을 사용하여 서보모터를 제어 할 수 있다. 해당 포트 출력을 pwm으로 on/off 주기를 설정하여주면, 서보모터가 원하는 각도로 움직인다. 전원은 외부 메인보드의 전원을 사용해야 한다. (자체 전원 지원 안함)



J371 커넥터 핀 번호

| 1 번 | 3 번 | 5 번 | 7 번 | 9 번     |
|-----|-----|-----|-----|---------|
| D0  | D2  | D4  | D6  | GND     |
| 2 번 | 4 번 | 6 번 | 8 번 | 10 번    |
| D1  | D3  | D5  | D7  | VCC(5V) |



## 3. MG-995 (2개)

토크 8.5kg, 스피드 0.2sec @ 4.8V/60도, 무게 42g, 사이즈 40.6\*19.8\*42.9mm, 코어리스 모터, 듀얼볼베어링 및 풀메탈기어, FUTABA 서보혼 호환. 2개를 사용하여 사격시스템의 상/하, 좌/우 제어를 담당하게 함.

## Basic Information

|                   |  |
|-------------------|--|
| Modulation:       | Analog   |
| Torque:           | <b>4.8V:</b> 138.9 oz-in ( 8.5 kg-cm)  |
| Speed:            | <b>4.8V:</b> 0.20 sec/60°  |
| Weight:           | 1.94 oz (55.0 g)   |
| Dimensions:       | Length: 1.60 in (40.6 mm)<br>Width: 0.78 in (19.8 mm)<br>Height: 1.69 in (42.9 mm) |
| Motor Type:       | Coreless   |
| Gear Type:        | Metal  |
| Rotation/Support: | Dual Bearings  |

## Additional Specifications

|                   |              |
|-------------------|--------------|
| Rotational Range: | 180°         |
| Pulse Cycle:      | 20 ms        |
| Pulse Width:      | 1000-2000 µs |
| Connector Type:   | JR           |





## 4. SG-90 미니서보(2개)

토크 1.8kg, 스피드 0.10sec @ 4.8V, 무게 9g, 사이즈 21.8\*11.4\*22.6mm, 아날로그서보 모터, 플라스틱기어, 카메라 모듈의 시야각 제어를 위한 팬/틸트의 상/하, 좌/우 제어를 위해 2개를 사용

**Specifications:**

Dimensions (L x W x H) : 21.8 x 11.4 x 22.6 mm

Weight = 9 grams

Weight with wire and connector : 10.6 grams

Stall Torque at 4.8 volts = 1.8 kg/cm

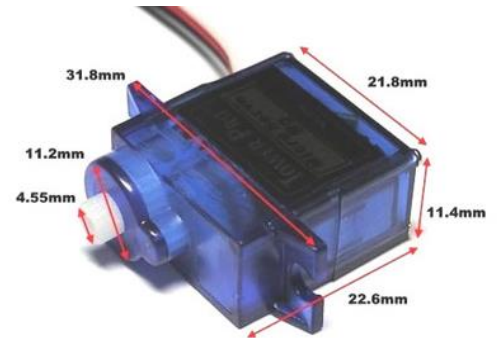
Operating Voltage = 4.0 to 7.2 volts

Operating Speed at 4.8 volts (no load) = 0.10 sec/ 60 degrees

Connector Wire Length = 248 mm

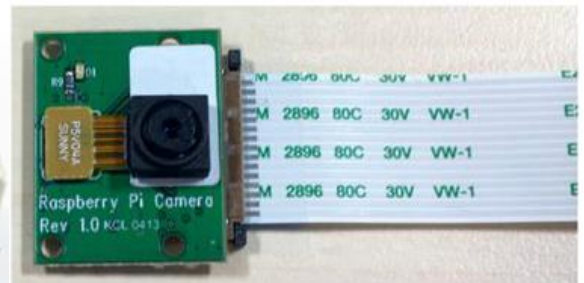
Universal "S" type connector fits most receivers

Servo-Horn include Single, Double, Cross 3 Type



## 5. 라즈베리파이 카메라 모듈

보드의 상단 표면에 두 개의 작은 소켓 중 F-PCB 컨택터를 통하여 라즈베리 파이에 연결. 전용 CSI 인터페이스를 사용하며 CSI 버스는 매우 높은 데이터 속도를 자랑하며, 독점적으로 픽셀 데이터를 전달. 보드 자체는 약 25mm X 20mm X 9mm이며, 무게는 3g. 카메라가 CSI 버스, 프로세서 카메라에서 픽셀 데이터를 전송하는 높은 대역폭 링크를 통해 파이의 BCM2835 프로세서에 연결. 센서 자체는 5 메가 픽셀의 해상도를 가지고 있으며, 고정 초점 렌즈 내장. 스틸 이미지의 측면에서, 카메라는 2592 X 1944 픽셀의 정적인 이미지에서 할 수 있으며, 1080p30의,으로 720p60 및 640x480p60/90 비디오를 지원.

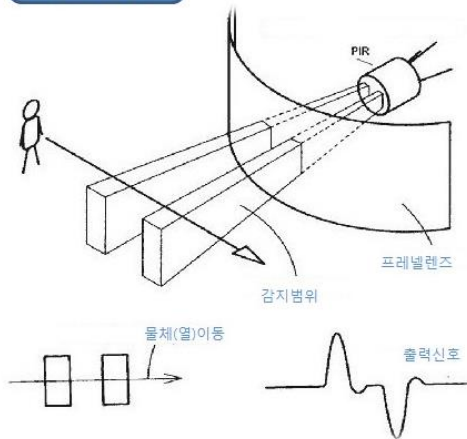




## 6. 디지털 적외선 모션센서(PIR)

공급전원 3~5V, 작동온도 -10 ~ 70°C, 출력신호 TTL, 감도조절 볼륨 내장, 감지후 출력 지연(Delay) 조정용 볼륨 내장, 감지각도 110도, 감지거리 최대 5미터이내, 크기 24 x 32 x 25mm (WHD), 중량 25g

PIR 센서 작동원리



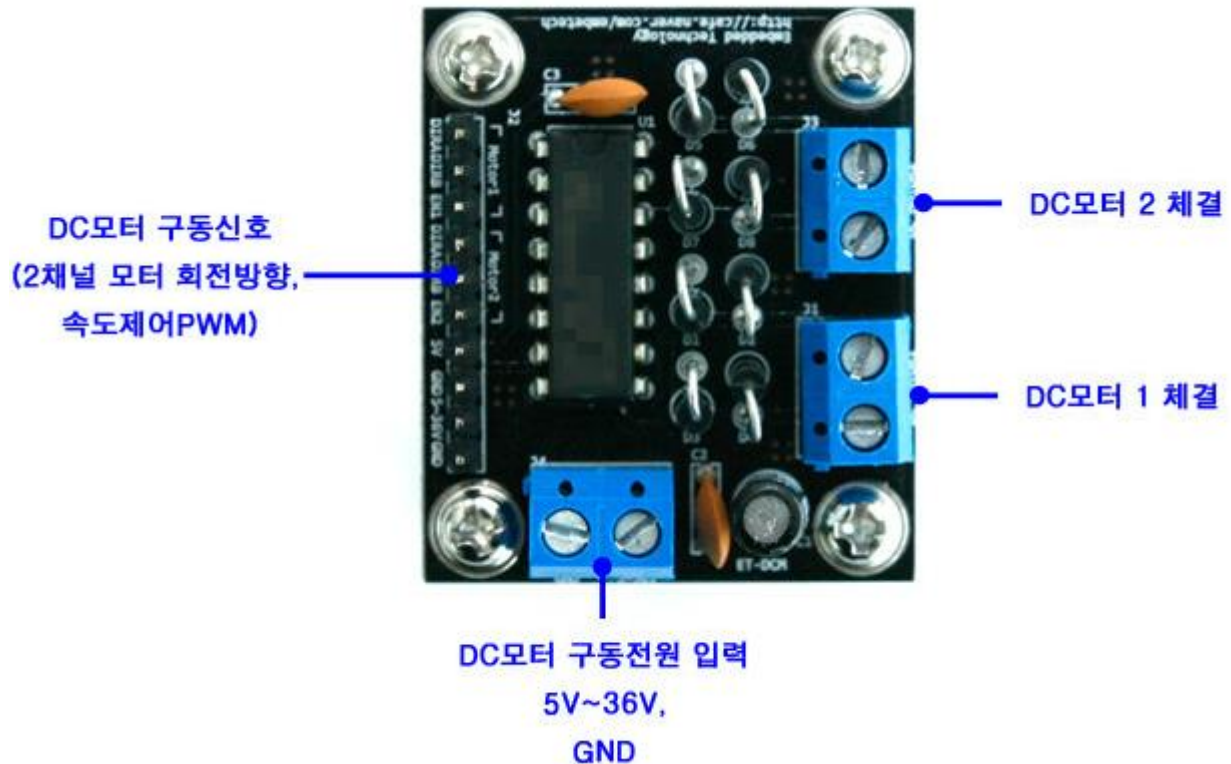
## 7. Si7021 Temperature and Humidity Sensor(온습도 센서)

온/습도 측정 가능, 작동 전압 1.9~3.6V, 대기전력 60nA, 온도범위 -40~85°C, 오차범위  $\pm 0.4^{\circ}\text{C}$ , 습도 범위 0~100%RH, 오차범위  $\pm 3\% \text{RH}$ , 온도 측정시간 7ms, 습도 측정시간 17ms, 인터페이스 I2C, 사이즈 32\*32mm



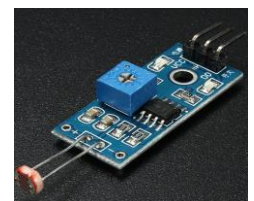
## 8. DC모터 드라이버 모듈(2개)

구동 전압 5~36V, 전류 각 채널당 최대 출력전류 2A, 채널당 출력전류 1A, 1x10 핀헤더 및 터미널단자 커넥터, MCU에서 발생하는 HIGH/LOW 신호를 DIRA, DIRB, EN1, DIRA, DIRB, EN2 핀에 입력하여 DC모터 정회전, 역회전 속도 제어 가능



#### 9. 디지털 조도 센서 모듈

단일 출력, 출력 신호 low, 감도 조절 가능, GPIO 포트기반, 메인 칩 LM393, 구동 전압 3~5V, 사이즈 32\*14mm



### III. Language

#### 1. JAVA

미국의 선 마이크로시스템즈사가 개발한 객체 지향 프로그래밍 언어. 1995년 5월에 발표, 1996년 1월부터 정식 버전을 배포하였다. C++를 바탕으로 언어 규격을 규정하였다. 버그의 원인이 되기 쉬운 기능, 예를 들면 지시자(pointer) 연산을 생략하였다. 또 C++에서는 사용이 끝난 객체(object)를 명시적으로 폐기하도록 프로그램에 기술하였으나 그 대신 자동 폐영역 회수(garbage collection) 기능을 추가하였다. 이와 같이 기

역 장치 관리의 오류에서 기인하는 버그를 방지한다. 자바는 객체 클래스의 계승 (inheri-tance) 관계를 실행할 때에 확정하는 동적 모델을 채용하였다. C++에서는 컴파일 시에 확정하기 때문에 상위 클래스에 새로운 기능을 추가하면 그 클래스와 계승 관계가 있는 모든 클래스를 재컴파일해야 한다. 자바에서는 클래스의 계승 관계를 실행할 때 확정하기 때문에, 하위의 클래스를 재컴파일할 필요가 없다. 자바로 작성한 프로그램의 원시 코드는 자바 컴파일러로 컴파일한다. 자바 컴파일러는 바이트 코드라는 중간 코드를 생성한다. 이 중간 코드를 자바 가상 머신이라는 소프트웨어로 해석해서 실행한다. 이 소프트웨어를 실장한 컴퓨터라면 컴퓨터의 명령 집합 아키텍처나 운영 체제(OS)에 관계없이 같은 바이트 코드를 변경하지 않고 실행할 수 있다. 자바의 원시 코드를 고쳐 쓰거나 재컴파일할 필요가 없다. 그렇기 때문에 자바는 기종이나 운영 체제와 무관한 응용 프로그램의 개발 도구로 각광받고 있다. 자바 가상 머신의 실현 방법으로는 현재 3종류가 있다. 즉, 바이트 코드를 해석하여 축차적으로 실행하는 자바 해석기, 바이트 코드를 일괄적으로 2진 코드로 변환하고 실행하는 JIT 컴파일러, 바이트 코드를 고속으로 해석하여 실행하기 위한 자바 칩(마이크로프로세서)이다.

## 2. Python

1989년 암스테르담의 귀도 반 로섬(Guido van Rossum)이 개발한 인터프리터 언어. 루비(Ruby), 펄(Perl)과 함께 대표적인 3세대 객체 지향형 스크립트 언어의 하나로서 문법 구조가 간단 명확하고, 언어 코아가 상대적으로 작은 대신 많은 시스템 호출과 풍부한 라이브러리, 윈도 시스템(X11, Motif, Tk, Mac, MFC)에 대한 인터페이스 등을 제공하여 개발·생산·재사용성이 높다. 또한 시스템 언어와 스크립트 언어의 중간 위치 성격과 C, C++를 사용해 새로운 모듈을 쉽게 만들 수 있는 등 다른 언어와 잘 융합되는 특징, 그리고 자유 소프트웨어 운동의 기수로서 다양한 분야에서의 활용이 기대되고 있다.

## 3. Shell Script

Shell에서 사용되는 명령어들의 조합으로 구성되고 운영 체제(OS)의 명령 해석기에 의해 실행되는 스크립트. 유닉스 계열의 본 셸, C 셸, 콘 셸에 사용되는 스크립트들이 대표적이다. OS에 따라서는 명령어들을 파이프(), 리다이렉션(>), 필터 등으로 연결하여 좀 더 복잡한 결과를 수행시킬 수 있는데, 이런 방식으로 묶여진 명령어 조합이 반복적으로 사용된다면 이를 셸 스크립트화하여 단일 명령처럼 사용하거나, 문자열을 명령어에 부가해 두고 셸 스크립트를 독립 변수(argument)로서 넘겨줄 수도 있다.

## IV. Socket

### 1. Socket

어떤 통신망의 특정 노드상의 특정 서비스를 식별하는 식별자. 소켓은 노드 주소와 서비스를 식별하는 포트 번호로 구성된다. 예를 들면, 인터넷 노드상의 포트 80은 월드 와이드 웹(WWW) 서버를 표시한다.

## V. Gstreamer

### 1. Gstreamer

GStreamer는 GNOME의 GObject 기반 자료형 체계와 더불어 C 프로그래밍 언어로 작성한 파이프라인 기반 멀티미디어 프레임워크이다.

GStreamer는 단순 오디오 재생, 오디오 및 비디오 재생, 녹음, 스트리밍 및 편집을 비롯하여 프로그래머가 다양한 미디어 관리 구성 요소를 만들 수 있게 해 준다. 파이프라인 디자인은 영상 편집기, 스트리밍 미디어 브로드캐스터, 미디어 플레이어와 같은 수많은 종류의 멀티미디어 응용 프로그램을 만드는 역할을 한다.

## VI. Android

### 1. Android

안드로이드는 모바일 운영 체제로 자바를 기반으로 소프트웨어를 작성할 수 있도록 되어있으며, 컴파일 된 바이트 코드를 구동할 수 있는 런타임 라이브러리를 제공한다. 그리고 안드로이드는 리눅스 커널 위에서 동작하며, 다양한 안드로이드 시스템 구성 요소에서 사용되는 C/C++ 라이브러리들을 포함하고 있다. 안드로이드는 기존의 자바 가상 머신과는 다른 가상 머신인 달빅 가상 머신을 통해 자바로 작성된 응용 프로그램을 별도의 프로세스에서 실행하는 구조로 되어 있다.

- SDK

Software development kit

- NDK

Native development kit

## VII. WEBPAGE

### 1. Apache

1995년 처음 발표된 월드와이드웹(WWW:World Wide Web) 서버용 소프트웨어이다. NCSA(National Center for Supercomputing Applications:미국국립수퍼컴퓨터활용센터) 소속 개발자들이 개발한 NCSA httpd 1.3 웹서버를 자신들이 개량한 것으로 소스코드까지 공개되고 있다. NCSA httpd 1.3 서버에 패치(patch)파일을 제공했던 개발자들이 'A PAtCH server'라는 용어에서 아파치라는 이름을 따왔다. 1995년 3월 18일 공개된 아파치0.2가 NCSA httpd 1.3에 패치파일을 제공하였다. 패치파일을 꾸준히 개선해 제공하고 있으며, 최고 수준의 성능을 발휘하기 때문에 월드와이드웹 서버용 소프트웨어로 가장 많이 사용되고 있다. 오픈소스(open source) 라이선스에 따라 무료로 배포되어 원하는 사람들이 자유롭게 사용할 수 있다. 유닉스·윈도 등을 비롯해 거의 모든 운영체제와 시스템에서 운용이 가능하다. 아파치소프트웨어재단(Apache Software Foundation)이라는 단체가 오픈소스 프로젝트의 아파치 커뮤니티를 지원하고 있다. 이 재단은 아파치 인큐베이터 프로젝트·아파치 엔트 프로젝트 등의 아파치 프로젝트를 진행 중이다.

### 2. HTML

HTML은 Hyper Text Markup Language의 약자로서 HTML은 하나의 마크업(Markup) 언어이다. 마크업 언어라는 것은 한 문서에 대해서 그 문서의 특정 부분을 표시하고 그 부분에 대해 특별한 의미를 부여하는 것을 말한다. 또한 HTML은하이퍼텍스트를 만들기 위한 언어인데, 여기서 하이퍼 텍스트란 어떤 형태를 가지고 거기에 부가적인 기능을 할 수 있는 문서를 말한다. 이러한 부가적인 기능들로는 문서의 일정 부분이 문서의 다른 부분으로 연결이 되도록 하거나 문서의 일부분에서 다른 부분으로 쉽게 이동할 수 있게 하는 기능 등이 있다.

HTML 문서 작성은 일반 텍스트 편집기부터 웹 전용 편집기까지 다양한 편집기로 작성하고 편집할 수 있다. 이번 프로젝트에서는 Notepad++로 HTML 파일을 작성했다.

### 3. JavaScript

Javascript는 서버와 클라이언트 응용프로그램을 위한 플랫폼에 영향을 받지 않는 객체기반 스크립트 언어이다. Javascript의 특징으로는 사용하기 간편한 작고 쉬운 스크립트 언어이고, 프로그램 제작시 컴파일 과정이 필요 없는 인터프리터형 언어이다. 또

한 이벤트 핸들링 기능을 제공하여 사용자의 다양한 반응에 대응하는 동적 HTML 문서 작성이 가능하며, 프로그램 코드는 HTML 문서에 내장되어 있으나 코드가 HTML 문서에서 분리되어 처리될 수 있다. 이번 프로젝트에서는 javascript 코드는 따로 분리를 하여 처리하였다.

#### 4. PHP

하이퍼텍스트 생성 언어(HTML)에 포함되어 동작하는 스크립팅 언어. 별도의 실행 파일을 만들 필요 없이 HTML 문서 안에 직접 포함시켜 사용하며, C, 자바, 펄 언어 등에서 많은 문장 형식을 준용하고 있어 동적인 웹 문서를 빠르고 쉽게 작성할 수 있다. ASP(Active Server Pages)와 같이 스크립트에 따라 내용이 다양해서 동적 HTML 처리 속도가 빠르며, PHP 스크립트가 포함된 HTML 페이지에는 .php, .php3, .phtml이 붙는 파일 이름이 부여된다. 처음에는 'Personal Home Page Tools'이라 불렸으며, 공개된 무료 소스이다.

#### 5. CSS

CSS란 Cascading Style Sheet를 줄여서 사용하는 것으로 HTML 문서에서 자주 사용되는 서체, 색상, 정렬 등의 유형을 가리킨다. 이런 유형들을 미리 정의해 두었다가 필요할 때마다 불러서 사용할 수 있도록 별도의 파일로 모아 놓은 것을 말한다. CSS는 소스의 관리와 수정이 쉽고, 페이지 로딩 속도가 빨라지며 HTML 요소 기능의 확장이 가능한 등의 특징을 가지고 있다. 이번 프로젝트에서 웹디자인을 하면서 CSS를 사용하여 대부분의 틀을 작성하였다.

## B. 프로젝트 상세 개발 내용

### I. 기능정의 리스트

#### ● 임베디드

| 대분류  | 중분류        | 소분류 | 상세항목                               | 항목번호         | 설명  | 담당자              | 순위 |
|------|------------|-----|------------------------------------|--------------|---|------------------|----|
| 임베디드 | 라즈베리<br>파이 | 설치  | 라즈비안<br>OS 설치                      | Em-Rp-St-01  | SD 카드로 라즈비안 OS 를 복사<br>후 라즈베리파이에서 설치한다.                             | 문선균,서강<br>원, 이호정 | 1  |
|      |            |     | 라즈베리파이<br>한글 키보드<br>및 언어 설정        | Em-Rp-St-02  | 라즈베리파이에서 한글을<br>사용하기 위한 설정을 해준다.                                    | 문선균,서강<br>원, 이호정 | 1  |
|      |            |     | RC 카와<br>라즈베리 연동                   | Em-Rp- St-03 | RC 카와 라즈베리파이를 연동  | 문선균,서강<br>원, 이호정 | 1  |
|      |            |     | 카메라 연동                             | Em-Rp-St-04  | 라즈베리파이에 카메라 모듈을<br>설치하고 연동시킨다.                                      | 문선균,서강<br>원, 이호정 | 1  |
|      |            |     | Wifi 용<br>라이브러리<br>설치              | Em-Rp-St-05  | 라즈베리로 통신을 하기 위한<br>wifi 라이브러리를 설치한다.                                | 문선균,서강<br>원, 이호정 | 1  |
|      |            |     | wifi 연동                            | Em-Rp-St-06  | 라즈베리파이에 wifi 모듈을<br>설치하고 연동시킨다.                                     | 문선균,서강<br>원, 이호정 | 1  |
|      |            | 통신  | wifi 사용을<br>위한<br>고정 IP 얻기         | Em-Rp-Co-01  | wifi 통신을 하기 위해<br>라즈베리파이가 잡을 고정 IP 를<br>설정한다                        | 문선균,서강<br>원, 이호정 | 1  |
|      |            |     | putty 를<br>사용한 로컬내<br>원격 접속        | Em-Rp-Co-02  | 라즈베리파이가 로컬 내 통신이<br>가능한지 확인하기 위해<br>putty 를 통한 원격 접속과<br>삼바서버에 접속한다 | 문선균,서강<br>원, 이호정 | 1  |
|      |            |     | 안드로이드<br>애플리케이션<br>으로 짧은<br>데이터 전송 | Em-Rp-Co-03  | 안드로이드 애플리케이션과<br>통신이 가능한지 여부를 알기<br>위해서 샘플 데이터를 전송                  | 문선균,서강<br>원, 이호정 | 2  |
|      |            |     | 영상 정보<br>전송                        | Em-Rp-Co-04  | 영상 정보를 안드로이드<br>애플리케이션으로 전송   | 문선균,서강<br>원, 이호정 | 2  |
|      |            |     | 삼바 설치                              | Em-Rp-Co-05  | FTP 통신을 위해 설치.  | 문선균,서강<br>원, 이호정 | 2  |
|      |            |     | FTP 통신 확인                          | Em-Rp-Co-06  | 삼바를 통해 FTP 통신이<br>가능한지 테스트  | 문선균,서강<br>원, 이호정 | 2  |

|  |      |        |                |             |                                       |              |   |
|--|------|--------|----------------|-------------|---------------------------------------|--------------|---|
|  |      | 영상     | 영상 스트리밍 촬영     | Em-Rp-Im-01 | 영상을 찍기 시작한다.                          | 문선균,서강원, 이호정 | 2 |
|  |      | 제어(영상) | 카메라 촬영시작 시점 제어 | Em-Rp-Ct-01 | 안드로이드 애플리케이션과 통신 연결이 되면 촬영 시작         | 문선균,서강원, 이호정 | 2 |
|  |      |        | 영상 촬영 간격 제어    | Em-Rp-Ct-02 | 영상 전송을 FPS = 24 인 간격으로 전송 (GStreamer) | 문선균,서강원, 이호정 | 2 |
|  | RC 카 | 제어(차량) | 차량 움직임 제어(1)   | Em-Ca-Ct-01 | 앞, 뒤 움직임을 DC 모터의 제어로 구현한다.            | 문선균, 김주혁     | 1 |
|  |      |        | 차량 움직임 제어(2)   | Em-Ca-Ct-02 | 좌, 우 움직임을 DC 모터의 제어로 구현한다.            | 문선균, 김주혁     | 1 |
|  |      | 제어(사격) | 총구 움직임 제어      | Em-Ca-Gu-01 | 위, 아래 움직임을 서보모터의 제어로 구현한다.            | 문선균, 김주혁     | 3 |
|  |      |        | 사격 속도 제어       | Em-Ca-Gu-02 | DC모터 제어로 사격 속도 제어                     | 문선균, 김주혁     | 3 |
|  |      | 센서     | 환경탐사센서 보드 설치   | Em-Ca-Ss-01 | 조도 센서를 통한 환경 탐사 센서 보드를 제작             | 문선균          | 2 |

● 안드로이드 애플리케이션

| 대분류          | 중분류 | 소분류     | 상세항목                | 항목번호        | 설명                                   | 담당자      | 순위 |
|--------------|-----|---------|---------------------|-------------|--------------------------------------|----------|----|
| 안드로이드 애플리케이션 | 설치  | JAVA    | Java jdk 설치         | Aa-St-Jv-01 | 안드로이드 개발 환경 구축을 위해 java jdk 설치       | 김주혁, 신익환 | 1  |
|              |     |         | Java 환경 변수 설정       | Aa-St-Jv-02 | Java jdk 설치 이후 시스템 설정의 환경변수 설정       | 김주혁, 신익환 | 1  |
|              |     | Android | 안드로이드 sdk 설치        | Aa-St-Ad-01 | 안드로이드 개발 환경 구축을 위해 안드로이드 sdk 설치      | 김주혁, 신익환 | 1  |
|              |     |         | 안드로이드용 이클립스 설치      | Aa-St-Ad-02 | 안드로이드 개발환경을 위해 이클립스 설치               | 김주혁, 신익환 | 1  |
|              |     |         | 안드로이드 테스트용 에뮬레이터 실행 | Aa-St-Ad-03 | 안드로이드 설치가 제대로 완료됐는지 확인하기 위해 에뮬레이터 실행 | 김주혁, 신익환 | 1  |
|              |     |         | 애플리케이션 최종수정 및       | Aa-St-Ad-04 | 어플리케이션 최종수정을 하고 어플 실행 파일인 Apk를 생성 후  | 김주혁, 신익환 | 3  |



|  |    |                            |                           |             |   |                     |   |
|--|----|----------------------------|---------------------------|-------------|---|---------------------|---|
|  |    |                            | Apk파일 생성                  |             | play스토어에 등록한다                                   |                     |   |
|  |    |                            | 안드로이드<br>ndk 설치           | Aa-St-Ad-05 | 안드로이드 gstreamer 개발 환경을<br>위하여 ndk 설치            | 김주혁,<br>서강원,<br>이호정 | 2 |
|  | UI | 전체<br>흐름<br>UI             | 로딩화면 UI<br>작성             | Aa-Ui-Au-01 | 로딩화면 UI를 제작                                     | 김주혁,<br>신익환         | 1 |
|  |    |                            | 접속 UI 작성                  | Aa-Ui-Au-02 | 라즈베리파이로 접속하는 UI를 제작                             | 김주혁,<br>신익환         | 1 |
|  |    |                            | 주화면 UI<br>작성              | Aa-Ui-Au-03 | 주화면을 사용할 UI를 제작                                 | 김주혁,<br>신익환         | 1 |
|  |    | 기능<br>별<br>UI              | 회원가입 UI<br>작성 및 코딩        | Aa-Ui-Fu-01 | 회원가입을 위한 UI를 제작 및 코딩                            | 김주혁,<br>신익환         | 3 |
|  |    |                            | 모터제어 부분<br>UI 작성 및<br>코딩  | Aa-Ui-Fu-02 | 모터제어 부분을 위한 UI를 제작 및<br>코딩                      | 김주혁,<br>신익환         | 1 |
|  |    |                            | 녹화/저장 UI<br>작성 및 코딩       | Aa-Ui-Fu-03 | 녹화/저장을 위한 UI를 제작 및 코딩                           | 김주혁,<br>신익환         | 2 |
|  |    |                            | 조도 감지 UI<br>작성 및 코딩       | Aa-Ui-Fu-04 | 조도 감지를 위한 UI를 제작 및 코딩                           | 김주혁,<br>신익환         | 3 |
|  |    |                            | 동영상 목록<br>UI 작성 및<br>코딩   | Aa-Ui-Fu-07 | 동영상 목록을 위한 UI를 제작 및<br>코딩                       | 김주혁,<br>신익환         | 1 |
|  |    |                            | 사격 기능 UI<br>작성 및 코딩       | Aa-Ui-Fu-09 | 사격 기능을 위한 UI를 제작 및 코딩                           | 김주혁,<br>신익환         | 3 |
|  | 통신 | 라즈<br>베리<br>파이<br>와의<br>통신 | Gstreamer<br>영상 수신<br>코딩  | Aa-Co-Rp-01 | 라즈베리파이에서 촬영된<br>영상데이터를 수신받기 위해<br>안드로이드 코딩을 한다. | 김주혁,<br>신익환         | 2 |
|  |    |                            | Gstreamer<br>영상 데이터<br>수신 | Aa-Co-Rp-02 | 라즈베리파이에서 촬영된<br>영상데이터를 수신                       | 김주혁,<br>신익환         | 2 |
|  |    |                            | 라즈베리파이<br>로 제어신호<br>전송 코딩 | Aa-Co-Rp-03 | 제어버튼을 클릭했을 때<br>라즈베리파이를 제어하기 위한<br>코드를 작성한다.    | 김주혁,<br>신익환         | 1 |
|  |    |                            | 라즈베리파이<br>로 제어신호          | Aa-Co-Rp-04 | 라즈베리파이로 제어 신호를 전송                               | 김주혁,<br>신익환         | 1 |

|  |      |         |                       |             |  |          |   |
|--|------|---------|-----------------------|-------------|--|----------|---|
|  |      |         | 전송                    |             |  |          |   |
|  |      | 공유      | 업로드 하기 위한 코딩          | Aa-Co-Up-01 | 안드로이드에서 다른 매체로 업로드 하기 위한 코딩                    | 김주혁, 신익환 | 2 |
|  |      |         | 서버에 업로드               | Aa-Co-Up-02 | 안드로이드에서 다른 매체로 업로드 하기 위한 코딩                    | 김주혁, 신익환 | 2 |
|  | DB   | 설치 및 연동 | Apache Tomcat 설치 및 연동 | Aa-Db-Ed-01 | Apache Tomcat을 설치하고 Mysql과 연동한다.               | 김주혁, 신익환 | 1 |
|  |      |         | Mysql 설치              | Aa-Db-Ed-02 | Mysql을 설치하고 PHP와 연동                            | 김주혁, 신익환 | 1 |
|  |      |         | PHP 설치 및 연동           | Aa-Db-Ed-03 | PHP를 설치하고 Mysql과 연동한다.                         | 김주혁, 신익환 | 1 |
|  |      | 코딩      | Mysql과 연동하는 애플리케이션 코딩 | Aa-Db-Cd-01 | Mysql과 연동할 수 있게 애플리케이션에서 코딩을 구현한다.             | 김주혁, 신익환 | 2 |
|  | 웹페이지 | 설치      | Samba 설치 및 연동         | Aa-Wb-Ed-01 | 라즈베리파이에 Samba를 설치하고 Windows와 연동한다.             | 신익환      | 2 |
|  |      |         | FTP 서버 설치             | Aa-Wb-Ed-02 | 라즈베리파이에 FTP서버를 설치한다.                           | 신익환      | 2 |
|  |      |         | FileZilla 설치 및 연동     | Aa-Wb-Ed-03 | Windows환경에 FileZilla를 설치하고 Raspberry pi와 연동한다. | 신익환      | 2 |
|  |      | 전체 흐름UI | 로그인화면 UI              | Aa-Wb-Au-01 | 로그인화면 UI를 제작                                   | 신익환      | 2 |
|  |      |         | 회원가입화면 UI             | Aa-Wb-Au-02 | 회원가입 UI를 제작                                    | 신익환      | 2 |
|  |      |         | 동영상 목록 UI             | Aa-Wb-Au-03 | 동영상 목록 UI를 제작                                  | 신익환      | 2 |
|  |      |         | 자유게시판 UI              | Aa-Wb-Au-04 | 자유게시판 UI를 제작                                   | 신익환      | 2 |
|  |      |         | 프로젝트 소개 UI            | Aa-Wb-Au-05 | 프로젝트 소개 페이지를 제작                                | 신익환      | 2 |
|  |      | 기능 별 UI | 회원가입 UI               | Aa-Wb-Fu-01 | 정상적인 회원가입 완료 시 알림창 설계                          | 신익환      | 2 |
|  |      |         | 로그아웃 UI               | Aa-Wb-Fu-02 | 로그아웃 시 알림창 설계                                  | 신익환      | 2 |

|  |  |  |         |             |                  |     |   |
|--|--|--|---------|-------------|------------------|-----|---|
|  |  |  | 다운로드 UI | Aa-Wb-Fu-03 | 다운로드를 할 시 알림창 설계 | 신익환 | 3 |
|  |  |  | 글쓰기 UI  | Aa-Wb-Fu-04 | 글쓰기 완료시 알림창 설계   | 신익환 | 2 |

## II. 주요 소스코드

### ● RCcar 서버 코드

```

import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;
import java.net.SocketAddress;
import java.util.logging.Logger;
public class Racar {
    private static final int sizeBuf = 50;
    public static void main(String[] args) throws IOException {
        int port = 8888;
        ServerSocket serverSock = new ServerSocket(port);
        Logger logger = Logger.getLogger("Racar");
        while (true) {
            Socket clientSock = serverSock.accept();
            SocketAddress clientAddress =
clientSock.getRemoteSocketAddress();
            // 스레드 방식으로써, RcvThread2 를 통해 소켓 기반의
Server-Client 통신을 시작하게 된다.
            Thread thread = new Thread(new RcvThread2(clientSock,
clientAddress, logger));
            thread.start();
            if(clientSock.isConnected())
                System.out.println("Connected! Client IP : " +
clientAddress);
        }
    }
}

```

```
// Server(라즈베리파이) 와 Client(안드로이드) 간 통신을 위한 라즈베리파이 전용 라이브러리
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.Socket;
import java.net.SocketAddress;
import java.util.logging.Level;
import java.util.logging.Logger;
import java.lang.Thread;
// 라즈베리파이에서 JDK8을 통해 각종 HW를 제어하기 위한 라이브러리들 (주 용도는 GPIO핀 제어)
import com.pi4j.io.gpio.impl.PinImpl; // 핀 확장
(GPIO 27~40번)
import com.pi4j.io.gpio.GpioController;
import com.pi4j.io.gpio.GpioFactory;
import com.pi4j.io.gpio.GpioPinDigitalInput;
import com.pi4j.io.gpio.GpioPinDigitalOutput;
import com.pi4j.io.gpio.PinDirection;
import com.pi4j.io.gpio.PinMode;
import com.pi4j.io.gpio.PinPullResistance; // 모든 핀 레지스턴스
import com.pi4j.io.gpio.PinState;
import com.pi4j.io.gpio.RaspiPin;
import com.pi4j.io.gpio.trigger.GpioCallbackTrigger;
import com.pi4j.io.gpio.trigger.GpioPulseStateTrigger;
import com.pi4j.io.gpio.trigger.GpioSetStateTrigger;
import com.pi4j.io.gpio.trigger.GpioSyncStateTrigger;
import com.pi4j.io.gpio.event.GpioPinListener; //특정 센서를 위한 리스너 (항상 동작하는 센서등)
import com.pi4j.io.gpio.event.GpioPinDigitalStateChangeEvent;
import com.pi4j.io.gpio.event.GpioPinEvent;
import com.pi4j.io.gpio.event.GpioPinListenerDigital;
import com.pi4j.io.gpio.event.PinEventType;
```

```

import com.pi4j.io.gpio.event.*;
public class RcvThread2 implements Runnable // 소켓통신 기
반의 클래스 RcvThread2
{
    private static final int sizeBuf = 50;
    private Socket clientSock;
    private Logger logger;
    private SocketAddress clientAddress;
    // 서보모터 제어를 위한 변수들
    int counter = 0;
    int period = 10;
    int repeatLoop = 3;
    double clockwise = 2.25;
    double counterclockwise = 0.75;
    final GpioController gpio = GpioFactory.getInstance();
    // 센서(PIR 센서) 제어를 위한 GPIO 제어부 변수선언 및 GPIO 핀
번호 할당
    final GpioController gpioSensor = GpioFactory.getInstance();
    final GpioPinDigitalInput PIR =
gpioSensor.provisionDigitalInputPin(RaspiPin.GPIO_08,
PinPullResistance.PULL_DOWN);
    // RC 차량에서 좌/우 제어를 담당하는 DC모터 GPIO 제어부 변수선언 및
GPIO 핀 번호 할당
    final GpioPinDigitalOutput pinfpower =
gpio.provisionDigitalOutputPin(RaspiPin.GPIO_01, "lpwm", PinState.LOW);
    final GpioPinDigitalOutput pinfleft =
gpio.provisionDigitalOutputPin(RaspiPin.GPIO_04, "lfwd", PinState.LOW);
    final GpioPinDigitalOutput pinfright =
gpio.provisionDigitalOutputPin(RaspiPin.GPIO_06, "lrev", PinState.LOW);
    // RC 차량에서 앞/뒤 제어를 담당하는 DC모터 GPIO 제어부 변수선언 및
GPIO 핀 번호 할당
    final GpioPinDigitalOutput pinlpower =
gpio.provisionDigitalOutputPin(RaspiPin.GPIO_00, "rpwm", PinState.LOW);
    final GpioPinDigitalOutput pinlback =

```

```

gpio.provisionDigitalOutputPin(RaspiPin.GPIO_02, "rfwd", PinState.LOW);
    final          GpioPinDigitalOutput          pinlgo          =
gpio.provisionDigitalOutputPin(RaspiPin.GPIO_03, "rrev", PinState.LOW);
    // RC 차량에서 사격시스템의 격발 제어를 담당하는 DC모터 GPIO 제어부
변수선언 및 GPIO 핀 번호 할당
    final          GpioPinDigitalOutput          pinshcont        =
gpio.provisionDigitalOutputPin(RaspiPin.GPIO_27, "scon", PinState.LOW);
    final          GpioPinDigitalOutput          pinshleft        =
gpio.provisionDigitalOutputPin(RaspiPin.GPIO_28, "slef", PinState.LOW);
    final          GpioPinDigitalOutput          pinshright       =
gpio.provisionDigitalOutputPin(RaspiPin.GPIO_29, "srig", PinState.LOW);
    // 카메라 모듈을 펜/틸트를 이용한 시야각 제어에 사용되는 서보모터의
GPIO 제어부 변수선언 및 GPIO 핀 번호 할당
    final          GpioPinDigitalOutput          pinservlr        =
gpio.provisionDigitalOutputPin(RaspiPin.GPIO_22, "servlr", PinState.LOW);
    final          GpioPinDigitalOutput          pinservud        =
gpio.provisionDigitalOutputPin(RaspiPin.GPIO_23, "servud", PinState.LOW);
    // 사격시스템의 사격각 제어에 사용되는 포신제어 서보모터의 GPIO 제어부
변수선언 및 GPIO 핀 번호 할당
    final          GpioPinDigitalOutput          pinshotlr        =
gpio.provisionDigitalOutputPin(RaspiPin.GPIO_24, "shotlr", PinState.LOW);
    final          GpioPinDigitalOutput          pinshotud        =
gpio.provisionDigitalOutputPin(RaspiPin.GPIO_25, "shotud", PinState.LOW);
    // Client와의 통신을 위한 Socket 설정 부분
    public RcvThread2(Socket clntSock, SocketAddress clientAddress, Logger
logger) {
        this.clientSock = clntSock;
        this.logger = logger;
        this.clientAddress = clientAddress;
    }
    // Client에서 Server로 해당 명령을 보낼 시, 수신하여 라즈베리파이에서 처
리하는 부분
    public void run() {

```

```
try {
    InputStream ins = clientSock.getInputStream();
    OutputStream outs = clientSock.getOutputStream();
    int rcvBufSize;
    byte[] rcvBuf = new byte[sizeBuf];
    while ((rcvBufSize = ins.read(rcvBuf)) != -1)
    {
        String rcvData = new String(rcvBuf, 0, rcvBufSize,
"UTF-8");

        // RC차량의 '전진' 기능부를 담당
        if (rcvData.compareTo("Go") == 0)
        {
            pinfpower.low();
            pinfleft.low();
            pinfright.low();
            pinlpower.high();
            pinlback.low();
            pinlgo.high();
            try {
                Thread.sleep(200);
            }
            catch (InterruptedException e){}
            pinfpower.low();
            pinlpower.low();
            System.out.println("Go!");
        }
        // RC차량의 '좌측' 기능부를 담당
        if (rcvData.compareTo("Left") == 0)
        {
            pinfpower.high();
            pinfleft.high();
            pinfright.low();
            pinlpower.low();
        }
    }
}
```

```
        pinlback.low();
        pinlgo.low();
        try {
            Thread.sleep(200);
        }
        catch (InterruptedException e) {}
        pinfpower.low();
        pinlpower.low();
        System.out.println("Left!");
    }
    // RC차량의 '우측' 기능부를 담당
    if (rcvData.compareTo("Right") == 0)
    {
        pinfpower.high();
        pinleft.low();
        pinfright.high();
        pinlpower.low();
        pinlback.low();
        pinlgo.low();
        try {
            Thread.sleep(200);
        }
        catch (InterruptedException e) {}
        pinfpower.low();
        pinlpower.low();
        System.out.println("Right!");
    }
    // RC차량의 '후진' 기능부를 담당
    if (rcvData.compareTo("Back") == 0)
    {
        pinfpower.low();
        pinleft.low();
        pinfright.low();
```



```
        pinlpower.high();
        pinlback.high();
        pinlgo.low();
        try {
            Thread.sleep(200);
        }
        catch (InterruptedException e) {}
        pinfpower.low();
        pinlpower.low();
        System.out.println(" Back!");
    }
    // RC차량의 '좌측으로 전진' 기능부를 담당
    if (rcvData.compareTo("LeftGo") == 0)
    {
        pinfpower.high();
        pinleft.high();
        pinfright.low();
        pinlpower.high();
        pinlback.low();
        pinlgo.high();
        try {
            Thread.sleep(200);
        }
        catch (InterruptedException e) {}
        pinfpower.low();
        pinlpower.low();
        System.out.println(" LeftGo!");
    }
    // RC차량의 '우측으로 전진' 기능부를 담당
    if (rcvData.compareTo("RightGo") == 0)
    {
        pinfpower.high();
        pinleft.low();
```

```
        pinfright.high();
        pinlpower.high();
        pinlback.low();
        pinlgo.high();
        try {
            Thread.sleep(200);
        }
        catch (InterruptedException e) {}
        pinfpower.low();
        pinlpower.low();
        System.out.println("RightGo!");
    }
    // RC차량의 '좌측으로 후진' 기능부를 담당
    if (rcvData.compareTo("LeftBack") == 0)
    {
        pinfpower.high();
        pinleft.high();
        pinfright.low();
        pinlpower.high();
        pinlback.high();
        pinlgo.low();
        try {
            Thread.sleep(200);
        }
        catch (InterruptedException e) {}
        pinfpower.low();
        pinlpower.low();
        System.out.println("LeftBack!");
    }
    // RC차량의 '우측으로 후진' 기능부를 담당
    if (rcvData.compareTo("RightBack") == 0)
    {
        pinfpower.high();
```

능부를 담당

```

        pinleft.low();
        pinfright.high();
        pinlpower.high();
        pinlback.high();
        pinlgo.low();
        try {
            Thread.sleep(200);
        }
        catch (InterruptedException e) {}
        pinfpower.low();
        pinlpower.low();
        System.out.println("RightBack!");
    }
    // 사격시스템의 '격발' 기능부를 담당
    if (rcvData.compareTo("ServoShoot") == 0)
    {
        pinshcont.high();
        pinshleft.low();
        pinshright.high();

        System.out.println("engage. Fire!");
    }
    // 사격시스템 및 주행 등 모든 기능의 '정지' 기
    if (rcvData.compareTo("Stop") == 0)
    {
        pinshcont.low();
        pinfpower.low();
        pinlpower.low();

        System.out.println("Stop!");
    }
    // 카메라 시야각 제어 및 사격시스템 사격각 제

```

어의 '좌측' 기능부를 담당

```
        if (rcvData.compareTo("ServoLeft") == 0)
        {
            for (counter = 0; counter < repeatLoop;
counter++)
            {
                pinservlr.high();

                try {
                    java.lang.Thread.sleep(2,
0);

                }
                catch (InterruptedException e) {}
                pinservlr.low();
                try {

                    java.lang.Thread.sleep(period);

                }
                catch (InterruptedException e) {}
                pinshotlr.high();

                try {
                    java.lang.Thread.sleep(2,
0);

                }
                catch (InterruptedException e) {}
                pinshotlr.low();

                try
                {

                    java.lang.Thread.sleep(period);

                }
            }
        }
```

어의 '우측' 기능부를 담당

counter++)

0);

java.lang.Thread.sleep(period);

```

        catch (InterruptedException e) {}
    }
    try {
        Thread.sleep(100);
    }
    catch (InterruptedException e) {}
    pinshotlr.low();
    pinservlr.low();
    System.out.println("Servo Left!");
}
// 카메라 시야각 제어 및 사격시스템 사격각 제
if (rcvData.compareTo("ServoRight") == 0)
{
    for (counter = 0; counter < repeatLoop;
        counter++)
    {
        pinservlr.high();
        try {
            java.lang.Thread.sleep(0,
                0);

        }
        catch (InterruptedException e) {}
        pinservlr.low();
        try {

        }
        catch (InterruptedException e) {}
        pinshotlr.high();

        try {
            java.lang.Thread.sleep(0,

```

```

0);

                                }
                                catch (InterruptedException e) {}
                                pinshotlr.low();

                                try {

java.lang.Thread.sleep(period);

                                }
                                catch (InterruptedException e) {}
                                }
                                try{
                                    Thread.sleep(100);
                                }
                                catch (InterruptedException e) {}
                                pinshotlr.low();
                                pinservlr.low();
                                System.out.println("Servo Right!");
                                }
                                // 카메라 시야각 제어 및 사격시스템 사격각 제
어의 '상' 기능부를 담당

                                if (rcvData.compareTo("ServoUp") == 0)
                                {
                                    for (counter = 0; counter < repeatLoop;
counter++)
                                    {
                                        pinservud.high();
                                        try {
                                            java.lang.Thread.sleep(0,
0);

                                        }
                                        catch (InterruptedException e) {}
                                        pinservud.low();

```

```

                                try {
                                java.lang.Thread.sleep(period);
                                }
                                catch (InterruptedException e){
                                pinshotud.high();
                                try {
                                java.lang.Thread.sleep(0,
0);
                                }
                                catch (InterruptedException e) {}
                                pinshotud.low();
                                try {
                                java.lang.Thread.sleep(period);
                                }
                                catch (InterruptedException e) {}
                                }
                                try {
                                Thread.sleep(100);
                                }
                                catch (InterruptedException e) {}
                                pinshotud.low();
                                pinservud.low();
                                System.out.println("Servo Up!");
                                }
                                // 카메라 시야각 제어 및 사격시스템 사격각 제
                                어의 '하' 기능부를 담당
                                if (rcvData.compareTo("ServoDown") == 0)
                                {
                                for (counter = 0; counter < repeatLoop;
                                counter++)
                                {

```

```
0);

    pinservud.high();
    try {
        java.lang.Thread.sleep(2,

    }
    catch (InterruptedException e) {}
    pinservud.low();

    try {

    }
    catch (InterruptedException e) {}
    pinshotud.high();

    try {
        java.lang.Thread.sleep(2,

    }
    catch (InterruptedException e) {}
    pinshotud.low();

    try {

    }
    catch (InterruptedException e) {}

    }
    try {
        Thread.sleep(100);
    }
    catch (InterruptedException e) {}
    pinshotud.low();
```



```

        pinservud.low();
        System.out.println("Servo Down!");
    }
    // 수신된 데이터가 어떤 데이터인지를 출력해주
    logger.info("Received data : " + rcvData + " (" +
clientAddress
        + ")");
    outs.write(rcvBuf, 0, rcvBufSize);
    }
    logger.info(clientSock.getRemoteSocketAddress() +
"Closed");
    //Server가 Client와 연결을 종료하는 부분
    } catch (IOException ex)
    {
        logger.log(Level.WARNING, "Exception in RcvThread", ex);
    }
    finally {
        try {
            clientSock.close();
            System.out.println("Disconnected! Client IP : " +
clientAddress);
        }
        catch (IOException e) {}
    }
}
}

```

## ● Android

### 1. 로딩소스

```

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;

```

```
import android.os.Handler;
import android.os.Message;
import android.view.Window;
public class Loading extends Activity {
    public void onCreate(Bundle savedInstanceState) { // 하면 띄우는 부분
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.loading); // 정의된 XML를 부르는 부분
        Loading();
    }
    private void Loading() { // 핸들러를 통해서 일정 시간 이후 로그인 화면으로 전환
        Handler hd = new Handler() {
            public void handleMessage(Message msg) {
                Intent intent = new Intent(getApplicationContext(),
                    LoginActivity.class);
                startActivity(intent); // login 액티비티 실행
                finish(); // 현재 액티비티 종료
            }
        };
        hd.sendMessageDelayed(0, 2000); // 지연로딩시간을 설정 할 수 있음 현재2초로 설정
    }
}
```

Loading.java

## 2. 모터제어 Client측 소스

```
import java.io.IOException;
import java.net.Socket;
public class RcvThread implements Runnable {
    private Logger logger;
    private final int sizeBuf = 50;
    private int flag;
    private Socket socket;
```

```

private String rcvData = "Error";
private byte[] rcvBuf = new byte[sizeBuf];
private int rcvBufSize;
// 생성자 정의 부분
public RcvThread(Logger logger, Socket socket){
    this.logger = logger;
    flag = 1;
    this.socket = socket;
}
public void setFlag(int setflag) {
    flag = setflag;
}
public void run() {
    while(flag == 1){
        try{
            //Server 측에서 보내온 메시지를 받아서 logger class를
            //이용해서 함수 호출 후 받은 메시지를 띄운다.
            rcvBufSize = socket.getInputStream().read(rcvBuf);
            rcvData = new String(rcvBuf, 0, rcvBufSize, "UTF-8");
            if (rcvData.compareTo("[close]") == 0){
                flag = 0;
            }
            logger.log("Recive Data : " + rcvData);
        } catch (IOException e){
            e.printStackTrace();
        }
    }
    logger.log("Exit loop");
}
}

```

RcvThread.java

import android.os.Handler;

```
import android.widget.TextView;
//메시지를 받아서 TextView를 수정해주는 부분
class Ps implements Runnable{
    TextView t;
    String s;
    public Ps(TextView t, String s){
        this.t = t;
        this.s = s;
    }
    public void run(){
        t.setText(s); // 최종적으로 이부분에서 text내용을 변경시켜준다.
    }
}

//핸들러를 받아서 메시지를 상태 Textview에 전달하는 부분
public class Logger {
    Handler h;
    TextView t;
    public Logger(TextView t){
        this.t = t;
        h = new Handler();
    }
    public void log(String s){
        h.post(new Ps(t, s)); // textview를 변경해주는 Ps함수를 인자를 넣어서 호출
    }
}
```

Logger.java

```
import android.os.Handler;
import android.widget.TextView;
//메시지리르 받아서 TextView를 수정해주는 부분
class Ps implements Runnable{
    TextView t;
    String s;
```

```
public Ps(Textview t, String s){
    this.t = t;
    this.s = s;
}

public void run(){
    t.setText(s); // 최종적으로 이부분에서 text내용을 변경시켜준다.
}
}

//핸들러를 받아서 메시지를 상태 Textview에 전달하는 부분
public class Logger {
    Handler h;
    Textview t;
    public Logger(Textview t){
        this.t = t;
        h = new Handler();
    }
    public void log(String s){
        h.post(new Ps(t, s)); // textview를 변경해주는 Ps함수를 인자를 넣어서 호출
    }
}
```

MainActivity.java (연결, 연결종료 부분)

```
private ImageButton buttonGo;
    private ImageButton buttonLeft;
    private ImageButton buttonLeftGo;
    private ImageButton buttonLeftBack;
    private ImageButton buttonRight;
    private ImageButton buttonRightGo;
    private ImageButton buttonRightBack;
    private ImageButton buttonServoUp;
    private ImageButton buttonServoDown;
    private ImageButton buttonServoLeft;
    private ImageButton buttonServoRight;
```

```
private ImageButton buttonServoshoot;  
private ImageButton buttonBack;  
private ImageButton buttonStop;
```

Main에서 방향 제어 버튼 선언 부분

```
buttonGo = (ImageButton) this.findViewById(R.id.b_up);  
    buttonLeft = (ImageButton) this.findViewById(R.id.b_left);  
    buttonLeftGo = (ImageButton) this.findViewById(R.id.b_leftgo);  
    buttonLeftBack = (ImageButton) this.findViewById(R.id.b_leftback);  
    buttonRight = (ImageButton) this.findViewById(R.id.b_right);  
    buttonRightGo = (ImageButton) this.findViewById(R.id.b_rightgo);  
    buttonRightBack = (ImageButton) this.findViewById(R.id.b_rightback);  
    buttonBack = (ImageButton) this.findViewById(R.id.b_down);  
    buttonStop = (ImageButton) this.findViewById(R.id.b_stop);  
    buttonServoUp = (ImageButton) this.findViewById(R.id.b_survoupp);  
    buttonServoDown = (ImageButton) this.findViewById(R.id.b_survodown);  
    buttonServoLeft = (ImageButton) this.findViewById(R.id.b_survoleft);  
    buttonServoRight = (ImageButton) this.findViewById(R.id.b_survorange);  
    buttonServoshoot = (ImageButton) this.findViewById(R.id.b_servoshoot);  
    buttonGo.setOnClickListener(this);  
    buttonLeft.setOnClickListener(this);  
    buttonLeftGo.setOnClickListener(this);  
    buttonLeftBack.setOnClickListener(this);  
    buttonRight.setOnClickListener(this);  
    buttonRightGo.setOnClickListener(this);  
    buttonRightBack.setOnClickListener(this);  
    buttonBack.setOnClickListener(this);  
    buttonStop.setOnClickListener(this);  
    buttonServoUp.setOnClickListener(this);  
    buttonServoDown.setOnClickListener(this);  
    buttonServoLeft.setOnClickListener(this);  
    buttonServoRight.setOnClickListener(this);  
    buttonServoshoot.setOnClickListener(this);
```

Main에서 방향 제어 버튼 클릭 리스너등록 부분

```
@Override
    public void onClick(View arg0) {
        if (arg0 == buttonGo || arg0 == buttonLeft || arg0 == buttonLeftGo
            || arg0 == buttonLeftBack || arg0 == buttonRight
            || arg0 == buttonRightGo || arg0 == buttonRightBack
            || arg0 == buttonBack || arg0 == buttonStop
            || arg0 == buttonServoUp || arg0 == buttonServoDown
            || arg0 == buttonServoLeft || arg0 == buttonServoRight || arg0
            == buttonServoshoot) {
            String sndOpkey = "CMD";
            if (socket != null) {
                if (arg0 == buttonGo)
                    sndOpkey = "Go";
                if (arg0 == buttonLeft)
                    sndOpkey = "Left";
                if (arg0 == buttonLeftGo)
                    sndOpkey = "LeftGo";
                if (arg0 == buttonLeftBack)
                    sndOpkey = "LeftBack";
                if (arg0 == buttonRight)
                    sndOpkey = "Right";
                if (arg0 == buttonRightGo)
                    sndOpkey = "RightGo";
                if (arg0 == buttonRightBack)
                    sndOpkey = "RightBack";
                if (arg0 == buttonBack)
                    sndOpkey = "Back";
                if (arg0 == buttonStop)
                    sndOpkey = "Stop";
                if (arg0 == buttonServoUp)
                    sndOpkey = "ServoUp";
                if (arg0 == buttonServoDown)
                    sndOpkey = "ServoDown";
                if (arg0 == buttonServoLeft)
                    sndOpkey = "ServoLeft";
```

```

        if (arg0 == buttonServoRight)
            sndOpkey = "ServoRight";
        if (arg0 == buttonServoshoot)
            sndOpkey = "ServoShoot";
        try {
            outs.write(sndOpkey.getBytes("UTF-8"));
            outs.flush();
        } catch (IOException e) {
            logger.log("Fail to send");
            e.printStackTrace();
        }
    } else {
    }
}

```

MainActivity OnClickListener에서 방향제어 버튼 onClick 부분

### 3. 메인 화면 옆에 나오는 Navigation Drawer

```

private String[] navItems = {"RasPI 연결", "동영상 목록", "빛 센서", "진동 센서", "움직임 센서", "로그 아웃"};
private ListView lvNavList;
private FrameLayout flContainer;

```

변수 선언

```

lvNavList = (ListView) findViewById(R.id.lv_activity_main_nav_list);
flContainer = (FrameLayout) findViewById(R.id.fl_activity_main_container);
lvNavList.setAdapter(new ArrayAdapter<String>(this,
        R.layout.simpleitem, navItems));
lvNavList.setOnItemClickListener(new DrawerItemClickListener());

```

Drawer 객체 생성 등록

```

private class DrawerItemClickListener implements
        ListView.OnItemClickListener {
    @Override
    public void onItemClick(AdapterView<?> adapter, View view,
        int position, long id) {
    }
}

```



```

switch (position) {
case 0:
    showDialog(DIALOG_YES_NO_MESSAGE1);
    break;
case 1:
    Intent intentVideoList = new Intent(MainActivity.this,
                                           videolist2.class);
    // pause streaming and stop the recording
    nativePause();
    startActivity(intentVideoList);
    break;
case 2:
    lightsensor_switch();
    break;
case 3:
    vibrationsensor_switch();
    break;
case 4:
    movedetectsensor_switch();
    break;
case 5:
    showDialog(DIALOG_YES_NO_MESSAGE2);
    break;
}
}
}

```

Drawer를 클릭했을 때 리스너 등록 부분

#### 4. 원하는 디렉터리 영상을 나오게 하는 video list

```

public class videolist2 extends Activity {
    String mRoot = "";
    String mPath = "";
    String videoname;

```

```
        ListView mListFile;
        ArrayList<String> mArFile;
        String filePath;
        private static final int DIALOG_YES_NO_MESSAGE = 1;
    @Override
        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.videolist2);
            // SD 카드가 장착되어 있지 않다면 앱 종료
            if (isSdCard() == false)
                finish();
            // SD 카드 루트 폴더의 경로를 구한다
            mRoot = Environment.getExternalStorageDirectory().getAbsolutePath()
                    + "/Movies/SVSVcar";
            // mTextMsg.setText(mRoot);
            String[] fileList = getFileList(mRoot);
            for (int i = 0; i < fileList.length; i++)
                Log.d("tag", fileList[i]);
            // ListView 초기화
            initListView();
            fileList2Array(fileList);
        }
    // SD 카드 장착 여부를 반환
    public boolean isSdCard() {
        String ext = Environment.getExternalStorageState();
        if (ext.equals(Environment.MEDIA_MOUNTED) == false) {
            Toast.makeText(this, "SD Card does not exist", Toast.LENGTH_SHORT)
                    .show();

            return false;
        }
        return true;
    }
    // 특정 폴더의 파일 목록을 구해서 반환
    public String[] getFileList(String strPath) {
        // 폴더 경로를 지정해서 File 객체 생성
        File fileRoot = new File(strPath);
```

```
// 해당 경로가 폴더가 아니라면 함수 탈출
if (fileRoot.isDirectory() == false)
    return null;
mPath = strPath;
// mTextMsg.setText(mPath);
// 파일 목록을 구한다
String[] fileList = fileRoot.list();
return fileList;
}
// 파일 목록을 ListView 에 표시
public void fileList2Array(String[] fileList) {
    if (fileList == null)
        return;
    mArFile.clear();
    for (int i = 0; i < fileList.length; i++) {
        Log.d("tag", fileList[i]);
        mArFile.add(fileList[i]);
    }
    ArrayAdapter adapter = (ArrayAdapter) mListFile.getAdapter();
    adapter.notifyDataSetChanged();
}
```

원하는 디렉토리의 영상을 리스트로 불러오는 부분

```
public void initListView() {
    mArFile = new ArrayList<String>();
    ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
        android.R.layout.simple_list_item_1, mArFile);
    mListFile = (ListView) findViewById(R.id.listFiles);
    mListFile.setAdapter(adapter);
    mListFile.setOnItemClickListener(new OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView parent, View view,
            int position, long id) {
            String strItem = mArFile.get(position);
            Toast.makeText(getApplicationContext(),
                ((TextView) view).getText(),
```

```

Toast.LENGTH_SHORT).show();

                                videoname = ((TextView) view).getText().toString();
                                Intent intent = new Intent(videolist2.this,
videoplay2.class);

                                intent.putExtra("filename", videoname);
                                startActivity(intent);

                                }
                            });
                            mListFile.setOnItemLongClickListener(new OnItemLongClickListener() {
                                @Override
                                public boolean onItemLongClick(AdapterView<?> parent, View
view,

                                    int position, long id) {
                                        videoname = ((TextView) view).getText().toString();
                                        showDialog(DIALOG_YES_NO_MESSAGE);
                                        return false;

                                    });
                            }

```

list를 클릭 혹은 롱 클릭했을 때의 리스너 등록 부분

#### 5. 공유 혹은 웹서버로 업로드

```

@Override
    protected Dialog onCreateDialog(int id) {
        switch (id) {
            case DIALOG_YES_NO_MESSAGE:
                final CharSequence[] items = {"다른 앱과 공유", "웹 서버로 전송", "취소"};

                return new AlertDialog.Builder(this)
                    .setTitle("원하는 작업을 선택하세요.")
                    .setItems(items,

                        new DialogInterface.OnClickListener() {
                            public void onClick(DialogInterface dialog,
                                int item) {

```

```

if( item == 0){
    File dirName = new
File(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_MOVIES) + "/SVSVcar");
//디렉토리를 지정합니다.

    String filename =
videoname; //공유할 이미지 파일 명

    File file = new
File(dirName, filename); //image 파일의 경로를 설정

    Uri mSaveImageUri =
Uri.fromFile(file); //file의 경로를 uri로 변경

    Log.i(file.toString(), "" +
file.toString());

    Intent intent = new
Intent(Intent.ACTION_SEND); //전송 메소드를 호출

    intent.setType("video/*"); //jpg 이미지를 공유 하기 위해 Type을 정의
    intent.putExtra(Intent.EXTRA_STREAM, mSaveImageUri); //사진의 Uri를 가지고 올
    startActivity(Intent.createChooser(intent, "Choose")); //Activity를 이용하여 호출
}
else if(item == 1){
    int serverResponseCode
= 0;

    String sdRootPath =
Environment.getExternalStorageDirectory().getAbsolutePath() + "/Movies/SVSVcar";
//String sdRootPath =
"/mnt/sdcard/Movies/SVSVcar";

    filePath = sdRootPath +
"/" + videoname.toString();

    String uploadServerUri
= "http://115.86.166.18/UploadToServer.php";

    ProgressDialog dialog2
= ProgressDialog.show(videolist2.this, "", "Uploading file...", true);

    String fileName =

    HttpURLConnection conn = null;

    DataOutputStream dos
= null;

```

```

String lineEnd =
"WrWn";

String twoHyphens =
"--",

String boundary =
"*****",

int bytesRead,

bytesAvailable, bufferSize;

byte[] buffer;
int maxBufferSize = 1 *
1024 * 1024;

File sourceFile = new
File(filePath);

if (!sourceFile.isFile()) {

    dialog2.dismiss();
    Log.e("uploadFile", "Source File not exist ." + fileName);

} else {
    try {
        // url를 연다

        FileInputStream fileInputStream = new FileInputStream(sourceFile);
        URL url = new URL(uploadServerUri);
        // http 연결
        conn = (URLConnection) url.openConnection();

        conn.setDoInput(true);
        conn.setDoOutput(true);
        conn.setUseCaches(false);
        conn.setRequestMethod("POST");
        conn.setRequestProperty("Connection", "Keep-Alive");
        conn.setRequestProperty("ENCTYPE", "multipart/form-data");
        conn.setRequestProperty("Content-Type",
            "multipart/form-data;boundary=" + boundary);
        conn.setRequestProperty("userfile", fileName);

        dos = new DataOutputStream(conn.getOutputStream());
        dos.writeBytes("--" + "*****" + "WrWn");
        dos.writeBytes("Content-Disposition: form-data; name=userfile;filename="+ fileName + " +
WrWn");

```

```

dos.writeBytes("WrWn");

// create a buffer of maximum size
bytesAvailable = fileInputStream.available();
bufferSize = Math.min(bytesAvailable, maxBufferSize);
buffer = new byte[bufferSize];

// read file and write it into form...
bytesRead = fileInputStream.read(buffer, 0, bufferSize);
while (bytesRead > 0) {
dos.write(buffer, 0, bufferSize);
bytesAvailable = fileInputStream.available();
bufferSize = Math.min(bytesAvailable, maxBufferSize);
bytesRead = fileInputStream.read(buffer, 0, bufferSize);}

// send multipart form data necessary after file data...
dos.writeBytes(lineEnd);
dos.writeBytes(twoHyphens + boundary + twoHyphens + lineEnd);

// Responses from the server (code and message)
serverResponseCode = conn.getResponseCode();
String serverResponseMessage = conn.getResponseMessage();
Log.i("uploadFile", "HTTP Response is : "+ serverResponseMessage + ": " +
serverResponseCode);

if (serverResponseCode == 200) {
    Toast.makeText(videolist2.this,
        "업로드 성공!", Toast.LENGTH_SHORT)
        .show();
}

// close the streams //
fileInputStream.close();
dos.flush();
dos.close();

} catch
(MalformedURLException ex) {
    dialog2.dismiss();
    ex.printStackTrace();
    runOnUiThread(new Runnable() {
        public void run() {
            Toast.makeText(videolist2.this,

```

```

                                "MalformedURLException", Toast.LENGTH_SHORT)
                                .show();

                                }

                                });

                                Log.e("Upload file to server", "error: " + ex.getMessage(),ex);

                                } catch

(Exception e) {
                                dialog2.dismiss();
                                e.printStackTrace();
                                runOnUiThread(new Runnable() {
                                public void run() {

                                Toast.makeText(videolist2.this,
                                "Got Exception : see logcat ",
                                Toast.LENGTH_SHORT).show();

                                }

                                });

                                Log.e("Upload file to server Exception",
                                "Exception : " + e.getMessage(), e);

                                }

                                dialog2.dismiss();

                                } // End else block

                                }

                                else if(item == 2){
                                dialog.cancel();

                                }

                                }

                                }).create();

                                }

                                return null;

                                }

```

공유와 웹서버로 전송할때의 두가지 코드를 기술한 부분

## 6. 동영상 재생

@Override

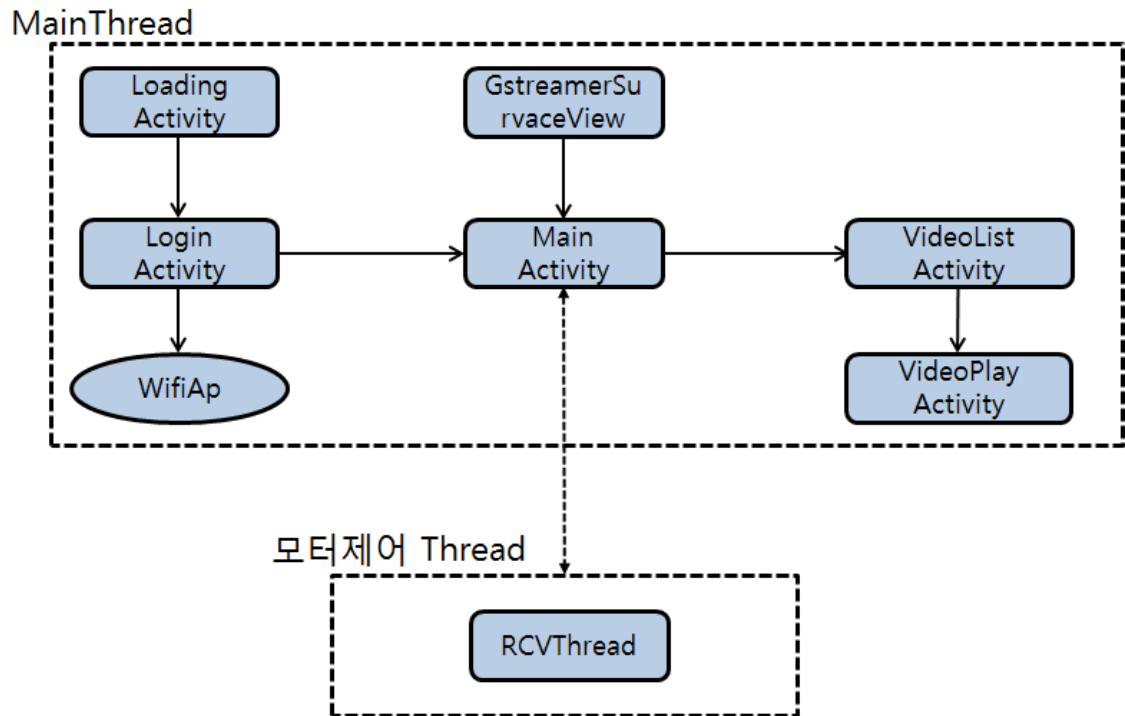
public void onItemClick(AdapterView parent, View view,



|  |  |
|--|--|
| <pre>                                 int position, long id) {                                 String strItem = mArFile.get(position);                                 Toast.makeText(getApplicationContext(),                                     ((TextView) view).getText(),                                 Toast.LENGTH_SHORT).show();                                 videoname = ((TextView)                                 view).getText().toString();                                 Intent intent = new Intent(videolist2.this,                                 videoplay2.class);                                 intent.putExtra("filename", videoname);                                 startActivity(intent);                                 } </pre> |  |
| 동영상을 눌렀을 때 intent로 화면전환과 intent값 전달 부분 (videolist2.java)   |  |
| 동영상 재생은 main에서 gstreamer 영상 스트리밍 부분과 동일  |  |

### III. 시스템 아키텍처

- 안드로이드



## 5. 사용자 매뉴얼

### A. 개발환경설치

#### I. 임베디드

라즈베리파이에 OS로 라즈비안을 설치한다.

>sudo apt-get install gstreamer1.0 명령어를 통해 gstreamer package 파일을 설치한다.

#### II. 안드로이드

컴퓨터에 안드로이드 SDK, NDK를 설치하고 Gstreamer SDK를 설치한다.

#### III. 웹서버

컴퓨터에 Apache 웹서버를 설치하고 웹브라우저로 테스트하여 확인한다.

#### IV. PHP

컴퓨터에 php5.3버전을 설치하고 Apache와 연동하고 확인한다.

#### V. DBMS

컴퓨터에 MySQL을 설치하고 php와 연동되는지 확인한다.

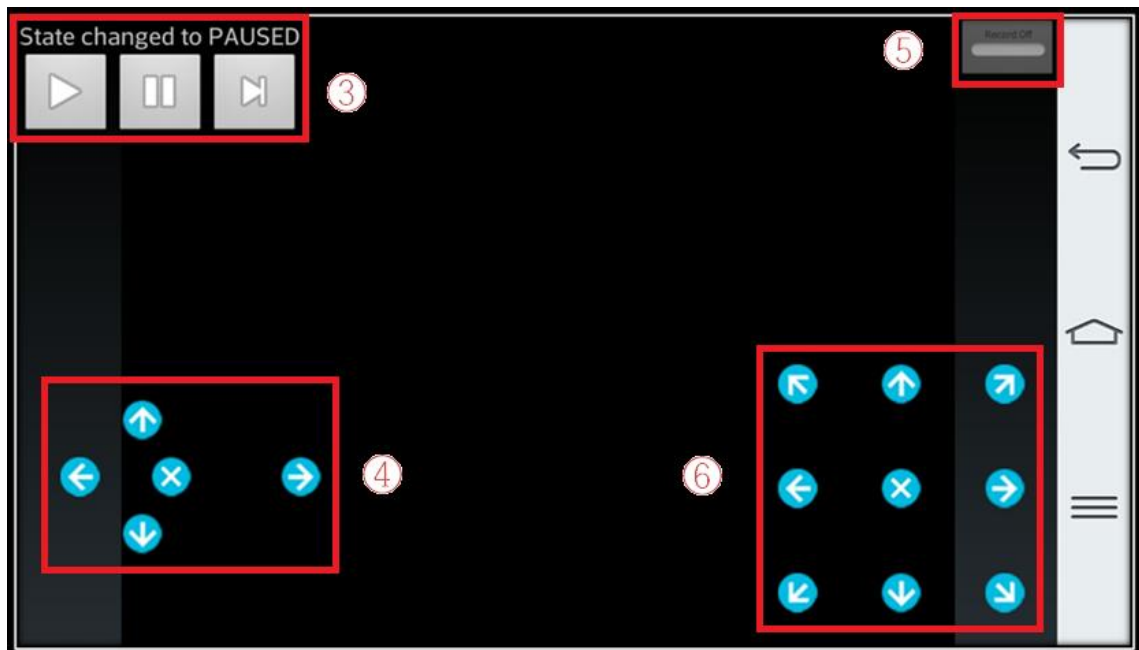
### B. 안드로이드

#### I. 로그인 페이지



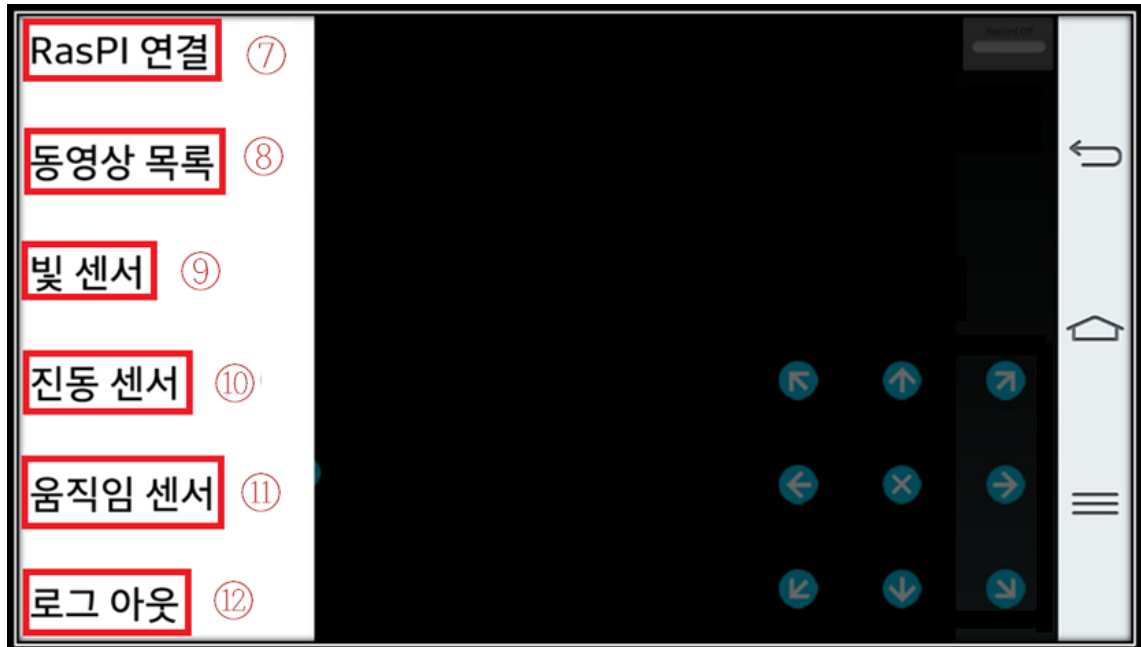
- ① 어 어플을 사용하기 위해서는 로그인을 해야한다. 로그인 ID 없을 경우 회원가입을 해야 한다. 여기서 로그인 한 아이디로 웹으로 접속 할 수 있다.
- ② 어 어플을 사용하기 위해서는 핫스팟을 켜주어야 한다.

## II. 스트리밍 및 조작화면



- ③ 영상 스트리밍을 시작하기 위한 시작 버튼, 일시 정지 버튼, 파일 디렉토리를 볼 수 있는 기능이다
- ④ 카메라 서보모터 제어 및 모터 총 제어 컨트롤 버튼이다.(상, 하, 좌, 우, 총 장전)
- ⑤ 녹화를 시작/중지할 수 있는 토글 버튼이다.
- ⑥ RC차량을 제어할 수 있는 버튼(8방향 버튼, 정지 버튼)

## III. 메뉴 목록 화면



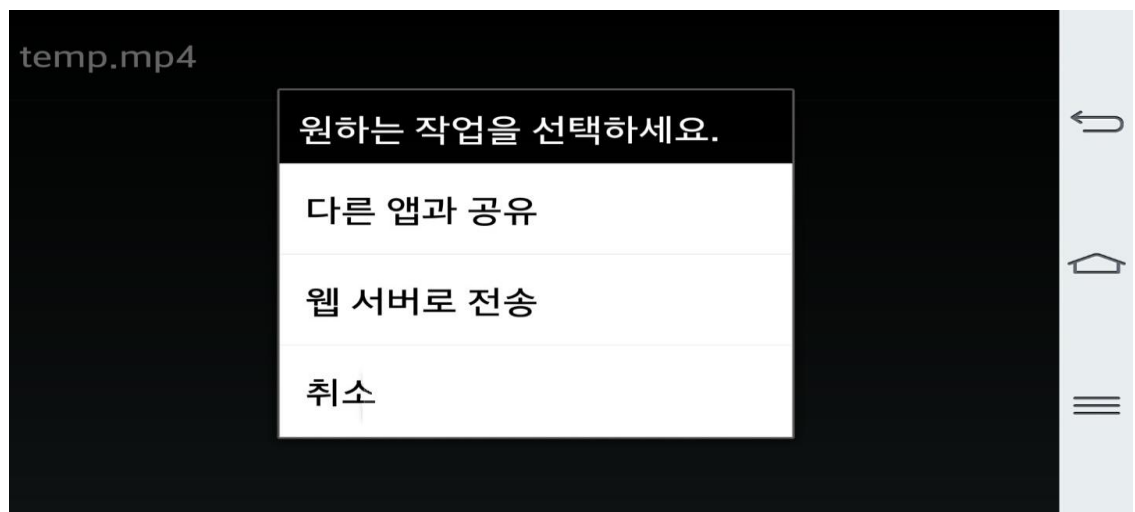
- ⑦ R C 차량을 제어 하기 위해서 RASPI 연결을 하기 위한 기능 (클릭시 라즈베리파이 IP입력 후 Connect버튼을 누른다)
- ⑧ 녹화한 영상 목록을 보여주는 기능
- ⑨ 빛 센서를 활성화 시킬 수 있는 기능. 빛 감지를 할 경우 알림메시지를 띄운다.
- ⑩ 진동 센서를 활성화시킬 수 있는 기능. 진동을 감지 할 경우 알림메시지를 띄운다.
- ⑪ 움직임 센서를 활성화시킬 수 있는 기능. 움직임을 감지 할 경우 알림메시지를 띄운다.
- ⑫ 메인화면을 나가는 동시에 로그인한 아이디를 로그아웃 할 수 있는 기능(클릭시 로그인화면으로 돌아간다)

#### IV. 메뉴 목록 세부 화면

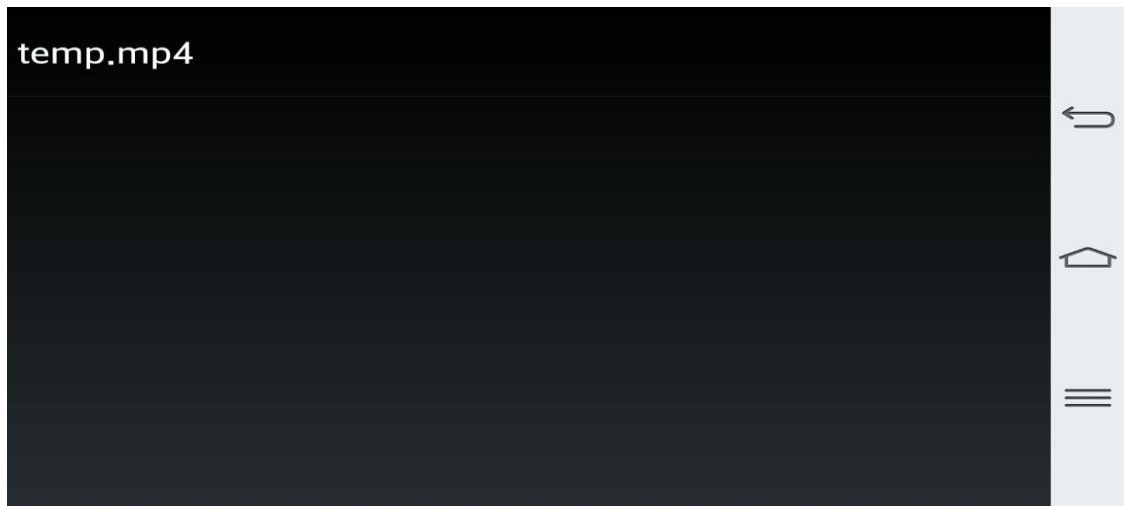
## ⑬ Raspberry PI 연결 클릭 화면



## ⑭ 동영상 목록 클릭 화면



## ⑮ 동영상 목록 길게 눌렀을 때 화면



## C. 웹 페이지

### I. 메인화면



- ① 카테고리를 나타내는 부분
- ② ID / PW 를 받아 로그인 하고 회원가입 및 ID/PW찾기 기능을 하는 부분
- ③ 메인이미지로서 네가지 이미지가 차례로 넘어가는 부분



## II. Information



**①**

### Our Project



- 명칭 : Streaming Video Scout Vehicle (DSV)
- 주제 : 원격 제어를 이용한 실시간 영상 전달 차량
- 목적 : 기존의 CCTV가 고정되어 있어야 한다는 인식을 벗어나 CCTV를 이동식 차량에 탑재하여 실시간으로 볼 수 있도록 하는 것이다.
- 이점 : 애플리케이션을 통해서 차량을 제어하여 실제 이동하지 않으면서도 관찰이 가능하다.
- 필요성 : 기존 CCTV 혹은 차량 블랙 박스들은 영상을 바로 확인할 수 없는 불편함이 존재하지만, 애플리케이션을 이용하여 어디서나 간편하게 영상을 확인할 수 있어 접근성이 높다.

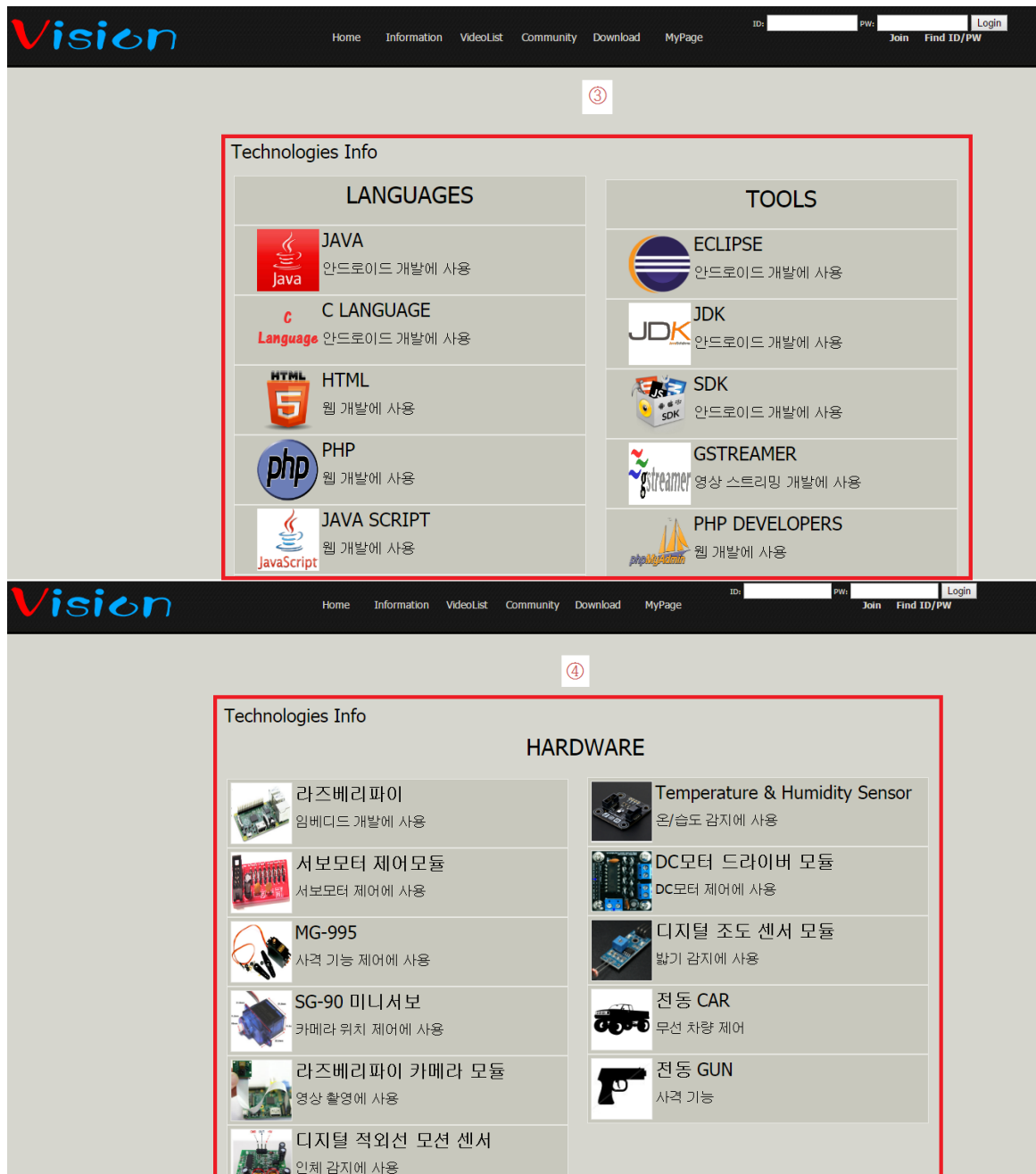
Yonsei University & @itcamp  
Team VISION



**②**

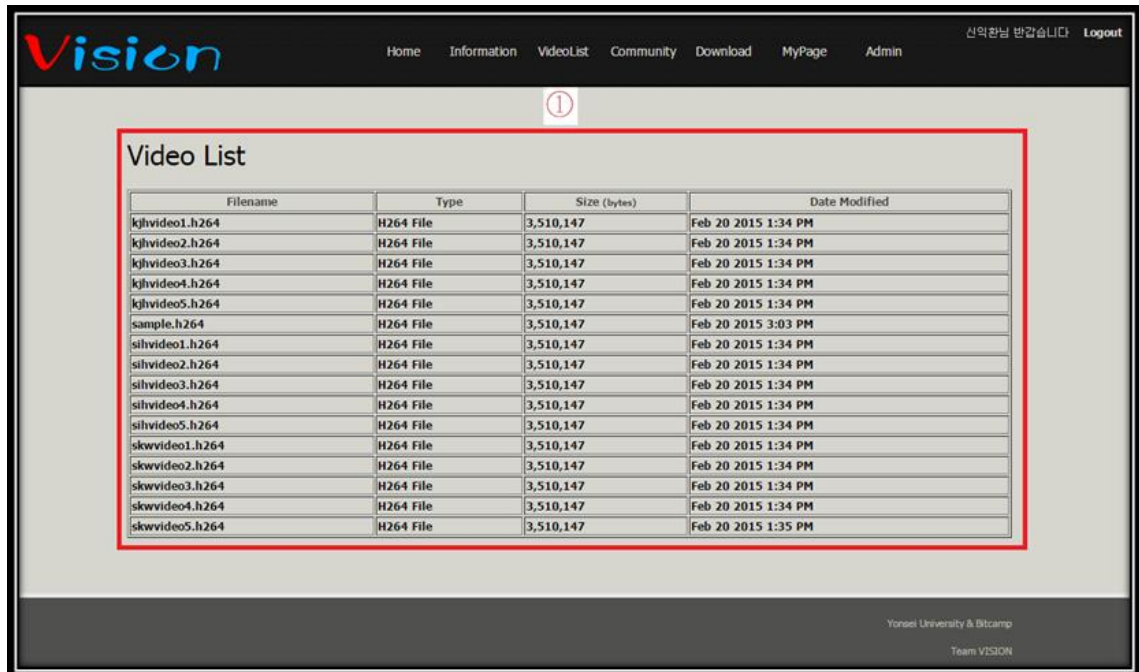
### Member Info

|   |  |   |
|---|--|---|
| <br><b>김주혁</b><br>Technical Leader, 안드로이드 담당 | <br><b>문선준</b><br>Team Leader, 임베디드 담당 | <br><b>서광원</b><br>영상 및 통신 담당 |
| <br><b>신역한</b><br>웹서버 담당                     | <br><b>이호정</b><br>서기 디자인 담당            |   |



- ① 프로젝트의 명칭, 주제, 목적, 이점, 필요성 등을 알려주는 페이지
- ② 프로젝트에 참여한 조원들의 사진, 이름, 담당업무를 나타내는 페이지
- ③ 프로젝트를 진행하면서 사용된 기술 개발환경을 나타내는 페이지
- ④ 프로젝트를 진행하면서 사용된 하드웨어들을 나타내는 페이지

### III. VideoList



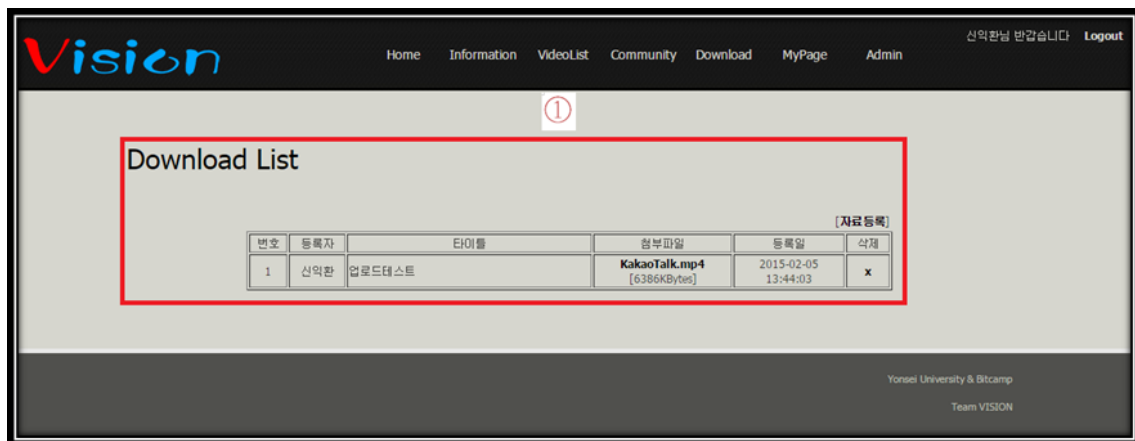
① 안드로이드에서 전송한 녹화파일을 볼 수 있는 페이지

### IV. Community



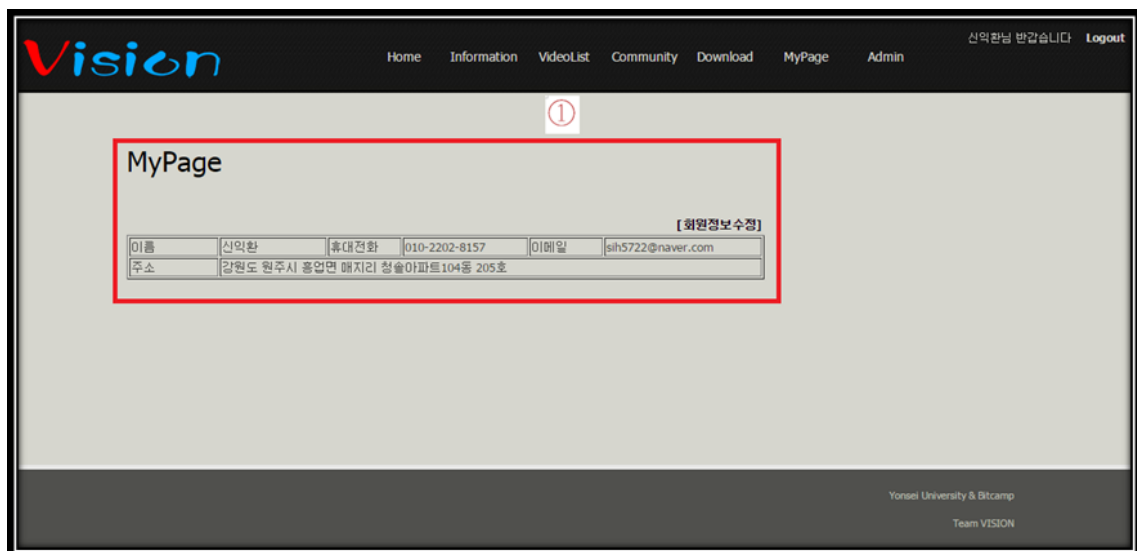
① 회원들이 자유롭게 의견을 나눌 수 있는 게시판 (수정, 삭제, 답변, 검색 기능)

## V. Download



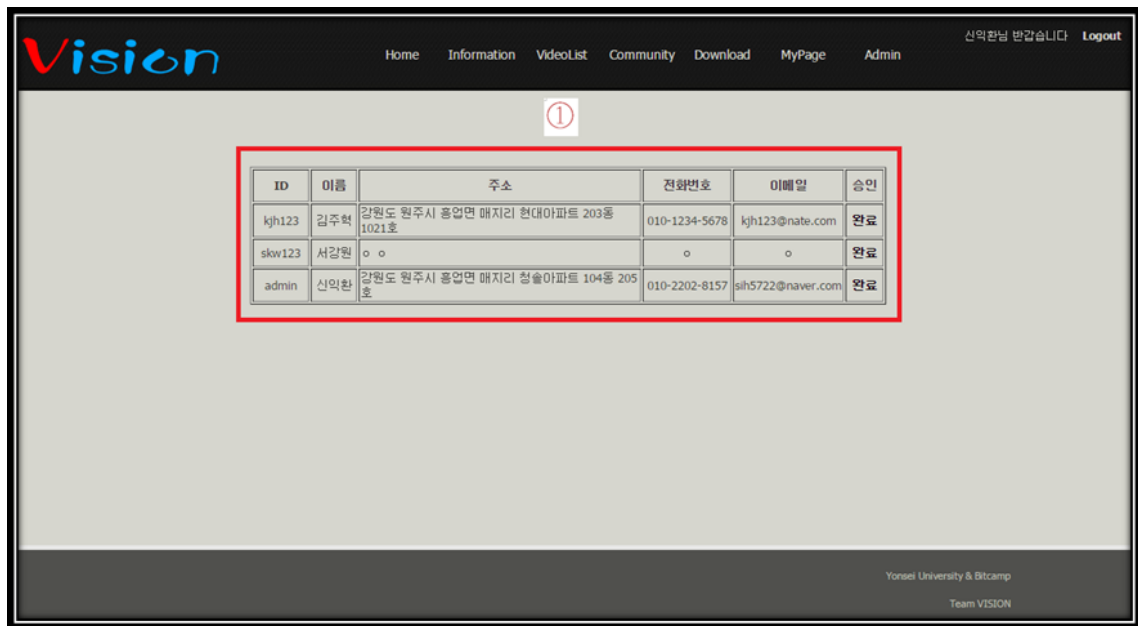
① 회원들이 개인의 영상 혹은 첨부파일을 올릴 수 있는 자료실

## VI. Mypage



① 회원정보를 확인하고 수정할 수 있는 페이지

## VII. Admin



① 관리자만 확인할 수 있는 페이지로 회원가입을 승인 및 대기상태로 바꿀 수 있는 페이지

## 6. 프로젝트 마무리

### A. 필수 S/W, H/W, Program

하드웨어

냉장고에서 영상 촬영과 촬영 영상 전송을 위해 필요한 하드웨어로 (1) 라즈베리파이와 (2) wifi Dongle, (3) 웹캠, (4) 충전식 배터리가 있다.

촬영 영상을 수신하고 영상 처리 프로그램으로 영상을 처리한 후, 데이터를 DB 서버에 저장하기 위해 서버로 사용할 (4) 컴퓨터가 필요하다.

사용자에게 영상 처리 결과로 얻은 데이터를 웹으로 보여주고 사용자로부터 데이터를 입력 받기 위해 (5) 컴퓨터나 핸드폰이 필요하다.

소프트웨어 및 프로그램

라즈베리파이가 냉장고에서 움직임을 인식하고 사람의 손을 인식하여 영상을 촬영하기 위해 (1) openCV 라이브러리를 이용하여 (2) IDLE에 파이썬으로 코딩했다.

영상의 송수신을 위한 소켓과 영상 처리 프로그램을 구동하는 데에는 (3) Visual studio 2010에 C#언어를 사용하고, 데이터를 저장하고 관리하기 위해 (4) MSSQL, (5) SQL Server Management Studio를 사용했다.

영상에서 글자나 식자재를 인식하고 데이터를 추출하기 위한 영상 처리 프로그램은 Visual studio 2010에서 C++로 코딩했고, openCV 라이브러리를 이용했다.

웹 페이지를 만들기 위해서는 (6) EditPlus와 (7) SQL Server Management Studio, (8) Apache, (9) 웹 브라우저가 이용되었다.

### B. 개발환경 설치

#### i. 임베디드

라즈베리파이에 OS로 라즈비안을 설치하고, openCV를 설치한다.

컴퓨터에 파이썬 인터프리터인 IDLE를 설치한다.

#### ii. 서버/DB

영상 처리 프로그램을 구동하고 FTP 서버와 시리얼 통신을 하기 위해서 Visual studio 2010을 설치한다.

데이터를 저장하고 관리하기 위해 DBMS로 MSSQL을 설치하고, SQL Server Management Studio를 설치한다.

## iii. 영상 처리

컴퓨터에 영상 처리 프로그래밍을 할 Visual studio 2010 을 설치하고, openCV 를 설치한다.

## iv. 웹

웹 개발은 메모장에도 해도 되지만, 좀 더 편한 코딩을 위해 컴퓨터에 EditPlus와 같은 편집기를 설치한다. php와 apache를 설치하고 이용하기 쉽게 APMSETUP을 설치한다. 웹 서버로 사용할 포트를 방화벽에서 외부 접속을 허용한다. 공유기로 ip가 할당된 경우, 포트 포워딩을 해준다.

**C. 실행**

냉장고에 프로그래밍을 마친 라즈베리파이를 설치하고 전원을 연결한다.

식자재나 이름을 적은 반찬통을 냉장고에 넣으면 라즈베리파이가 움직임을 감지하고 손을 인식하여 촬영을 하고 영상을 서버로 보낸다.

서버에서는 영상을 받아 영상 처리를 한 후, 데이터를 DB 에 저장한다.

사용자는 컴퓨터나 모바일 웹에서 냉장고 식자재와 관련한 데이터를 확인할 수 있다.

**7. 프로젝트 마무리****A. 기대효과**

- i. 원격 제어를 통해 차량을 제어하기 때문에 사람이 접근할 수 없는 장소를 정찰할 수 있는 점을 활용하여 효과적인 정찰이 가능하다. 또한 정찰을 하면서 녹화된 영상을 서버로 전송함으로써 외부에서도 영상을 확인할 수 있다는 장점이 있다. 서버에 저장된 영상은 사용자가 웹페이지에 공개파일로 올려둬으로써 다양한 사용자가 영상을 확인하는 것이 가능해진다.

**B. 문제점**

- i. 안드로이드 기기의 핫스팟을 이용하여 영상 스트리밍을 하기 때문에 안드로이드 기기와 떨어질 수 있는 거리에 한계점이 존재한다.
- ii. 안드로이드 기기의 핫스팟을 이용하여 서버에 영상을 전송하기 때문에 안드로이드 기기의 데이터 사용량이 증가한다는 문제점이 존재한다.

- iii. 차량 배터리의 한계로 인해 장시간 정찰이 불가능하다.
- iv. 녹화된 파일이 .h264파일이기 때문에 영상파일의 포맷을 변경하는 과정을 거쳐야하는 한계점이 존재한다.

### C. 개선방안

- i. 핫스팟이 아닌 wifi-direct를 사용하는 방안을 연구하여 안드로이드 기기의 데이터 사용량을 줄일 수 있다.
- ii. 대용량 배터리를 사용함으로써 정찰 시간을 증가시킬 수 있다.
- iii. 파일을 녹화할 때 .mp4 파일로 저장하는 방안을 연구하여 포맷을 변경하는 과정을 생략할 수 있다.

### D. 참고문헌 및 논문

- i. 임베디드
  - 사이먼 몽크 저, 박경욱 외 2명 역 / 라즈베리 파이 쿡북 / 한빛미디어 / 2015.01.05
  - 이재상, 표윤석 / 라즈베리 파이 활용 백서 / 비제이퍼블릭 / 2013.10.25
  - 사이먼 몽크 저, 배장열 역 / 라즈베리 파이 프로그래밍(파이썬으로 시작하는) / 제이펍 / 2013.09.
- ii. 안드로이드
  - 데무라 나리키즈 저, 이혜란 역 / 안드로이드 ndk 네이티브 프로그래밍 / 위키북스 / 2012
  - 실뱅 라타부이 저, 허윤규 역 / 안드로이드 n아 프로그래밍:JNI와 C/C++ 라이브러리를 활용한 네이티브 안드로이드 애플리케이션 /에이콘출판 / 2012
  - 고현철, 전호철 지음 / 안드로이드의 모든 것 NDK : C/C++라이브러리를 이용한 안드로이드 앱 개발 방법 / 한빛미디어 / 2012
- iii. 웹
  - 황민태 / PHP와 MySQL을 이용한 웹프로그래밍 / 도서출판 한산 / 2012.08.20
  - 홍세진 / PHP 웹 솔루션 실무 / 세진북스 / 2009.07.20
  - 이두진 / 고수로가는 PHP 이렇게 시작하세요 / 구민미디어 / 2008.08.05



- 신윤호 / Web Energy PHP 실무프로젝트 / IT미디어 / 2009.04.20
- 천인국 / 그림으로 쉽게 설명하는 안드로이드 프로그래밍 / 생능출판사 / 2012.08.20.
- 최규석, 박종지 / 쉽게 설명한 html+javascript+CSS / 도서출판 글로벌 / 2011.08.24.
- 김은기, 이진훈, 김은영 / HTML+JAVASCRIPT&CSS / 성안당 / 2009.07.14.
- 최미선 / 디자인의 폭을 넓혀주는 웹스타일북 / 안그라픽스 / 2009.02.25

## E. 참고사이트

### i. 임베디드

- 라즈베리파이  
[www.raspberrypi.org](http://www.raspberrypi.org)
- 오픈소스 로봇기술 공유카페 오로카  
[cafe.naver.com/openrt](http://cafe.naver.com/openrt)
- 산딸기마을  
[www.rasplay.org](http://www.rasplay.org)

### ii. 안드로이드



- GSTREAMER OPEN SOURCE MULTIMEDIA FRAMEWORK  
<http://gststreamer.freedesktop.org/>

### iii. 웹



- 서버로 파일 업로드 예제  
[http://androidexample.com/Upload\\_File\\_To\\_Server\\_-\\_Android\\_Example/index.php?view=article\\_discription&aid=83&aaid=106](http://androidexample.com/Upload_File_To_Server_-_Android_Example/index.php?view=article_discription&aid=83&aaid=106)
- 파일 공유 기능  
<http://jhrun.tistory.com/131>
- 각종 PHP 자료  
<http://www.phpschool.com/>

## F. 팀원 별 소감

| 팀원 | 소감 |
|----|----|
|----|----|

|   |  |
|---|--|
| <p style="text-align: center;"><b>김주혁</b></p>    | <p>5명 조원으로 이루어진 팀으로 2달 동안 프로젝트를 진행하면서 여러가지 겪은 일도 많고 배운점도 많았습니다. 일단 프로젝트의 주제를 정할때도 서로의 의견을 일치해 가는 일도 쉬운일이 아니였고, 프로젝트 보고서를 작성할 때 협업의 중요성을 많이 느꼈습니다. 또한, 효율적으로 업무를 나누는 기술도 프로젝트의 경험과 노하우가 있어야 하는것 같습니다. 또한, 중간에 프로젝트 개발 과정 중 난관에 봉착했을 때 위기 대처 능력도 중요하다고 생각합니다. 이번 프로젝트를 통해 여러가지 위기에 봉착했을 때 침착하게 다른 대안을 생각해내는 방법을 배웠습니다. 그리고 시도를 하기 전에 겁먹지 않고 일단 시도를 해보는 마인드를 길렀습니다. 이 프로젝트를 통해 좀 더 실무 프로젝트 방식에 익숙해지고 협업 능력을 배웠습니다.</p>  |
| <p style="text-align: center;"><b>문선균</b></p>  | <p>우선 이런 팀 단위의 프로젝트를 처음 접해보아서 생소하기도 하고 초반부에는 적응하기도 힘들었습니다. 거기다 프로젝트 총괄의 전반적인 부책임자 겸 팀장을 맡게 되니 알게 모르게 부담감도 컸습니다. 또한 학기중에 수업으로써 진행하는 프로젝트와 달리, 이 프로젝트는 말 그대로 실무를 위주로 한 프로젝트라서 그 중간 중간의 과정보다는 마지막의 '결과'가 중시되는 프로젝트임을 느꼈습니다.</p> <p>저는 평소에 공부나 개발이나 학습하는 방법에 있어서 맨 처음부터 차례차례 알아가며 하나씩 익혀가는 습관이 학창시절부터 몸에 배어있어 처음에는 적응하기가 꽤나 힘들었습니다. 실무 중심의 프로젝트는 처음부터 하나씩 알아가는게 아닌, 하나의 '목표'를 위해 필요한 것들만 수집하고 모으는 일이란게 제게는 너무 적응하기 힘든 방식이었습니다. 왜 이러한 부분에서 이러한 방식을 사용해야 하는가 하는 의문점도 생기고 이러한 의문점이 생기면 쉽사리 넘어가지 못하는 성격 때문에 학기중에도 어떠한 공부에 있어서 따라가지 못하는 부분이 생기면 그 부분이 이해가 될 때까지 학습하고 넘어가는 성격이기에 더욱더 이번 프로젝트는 힘이 들었습니다.</p> <p>그렇지만 이번 프로젝트는 저 혼자만의 개인 프로젝트가 아닌 팀 단위로 협업하여 진행해야하는 프로젝트였기에 저 혼자만의 문제를 다른 팀원들에게 까지 안기게 하고 갈 수가 없었습니다. 공부하는 방법에서는 이게 올바른 방법일지 몰라도, 하나의 목표를 가지고 그 목표를 이루기 위해 가는 길에는 결코 이 방법 하나만이 옳다고 할 수 없기에 저는 결국</p> |

|   |  |
|---|--|
|   | <p>제 자신을 바꾸기로 마음먹었습니다.</p> <p>원리가 이해가 되지 않고, 궁금증이 쌓이게 되더라도 나중에 해결하자 생각하고 우선 프로젝트에 필요한 자료 및 기능들을 수집하고 설계하였습니다. 하나의 목표라는 것에 모든 것을 집중하고 진행하다 보니 자연스럽게 적응이 되었고 결국 어떤 목표를 위해서라면 순서에 관계없이 우선적으로 필요한 것들만 선별하게 되는 능력을 키우게 되었습니다.</p> <p>이러한 능력이 점점 길러지게 되니 처음 접하는 분야도 금방 익숙해지고 제 것이 되가는 것을 느꼈습니다. 제가 이번 프로젝트에서 맡은 파트는 프로젝트에서 사용되는 전반적인 모든 HW 및 센서, 임베디드 시스템입니다. 저는 학기중에 임베디드에 관련된 수업을 한번도 들은 적이 없고, HW에 대한 것은 더욱더 다뤄본 적이 없습니다. 하지만 하나의 목표를 이뤄야한다는 목적이 생기니 이런 분야들을 조사하고 제 것으로 만드는 것이 결코 어려운 일이 아니었습니다.</p> <p>프로젝트 막바지에 다다른 지금, 저는 현재 저희 팀에 세팅되어있는 모든 하드웨어의 원리와 그 하드웨어를 이용한 Server-Client 기반의 데이터 통신에 대한 원리를 대부분 파악하고 있으며 이 파트에 대한 Server 부분의 모든 프로그램을 구축하였습니다. 학교를 4년간 다니면서 코딩 자체가 두려워 피하면서 자바, 파이썬 등을 하나도 다룰 줄 몰랐던 제가 하나의 목표를 가지게 되니 이렇게 다루게 되었습니다. 물론 모든 이론 및 원리를 정독하며 익힌건 아니지만 이번 프로젝트를 기초로 앞으로 더 공부하게 된다면 각 언어들에 대해서는 큰 어려움이 없을거라 생각 됩니다. 이번 프로젝트를 진행하는 2달의 기간이 제 인생의 터닝포인트가 된 것 같아 의미 있는 시간들이었다고 생각합니다.</p> |
| <p style="text-align: center;"><b>서강원</b></p>  | <p>저는 비트에 지원할 때, 이전처럼 2,3인 규모보다 좀 더 큰 팀 프로젝트를 경험하자는 것이 목적이었고 결국 5인 팀으로써 약 7주간 이전에 해봤던 것 보다 규모가 큰 프로젝트를 진행하였습니다. 프로젝트를 진행하면서 5명이 모두 다른 의견을 가지고 있을 때 의견을 모으는 것이 가장 어렵다는 것을 알게 되었습니다. 그리고 프로그래밍을 할 때, 정말 사소한 부분인데, 경험이 부족하여 어떤 것을 추가해야하는 지 몰라서 또는 주어진 정보가 너무 적어서 고생을 한 경우가 많았는데, 이런 것들을 해결할 수 있는 능력을 갖게 된 것이 가장 큰 성과인 것 같습니다. 마지막으로 다음에 이러한 프로젝트를 진행하게 될 때에는 좀 더 나은 모습을 보일 수 있을</p>   |

|  |  |
|--|--|
|  | 것 같습니다.  |
| <b>신익환</b><br>  | <p>프로젝트를 시작하기 전에는 의욕이 넘치고 열รน하고 싶었는데 막상 시작하고 보니 스스로가 많이 부족하다는 것을 깨달았다. 처음 접해보는 언어 및 개발환경이 대부분이었고 누군가가 가르쳐주는 것이 아닌 스스로 알아가야 했다. 그런 과정들이 당시에는 매우 힘들었지만 혼자서는 절대 하지 못할 일들을 조원들끼리 다 같이 해나가면서 단순히 지식만 얻은 것이 아니라 문제를 헤쳐 나가는 인내심과 자신감을 얻었다.</p>  |
| <b>이호정</b><br> | <p>이런 실무 위주 방식의 프로젝트는 처음인지라 적응하기가 쉽지 않았고, 많은 어려운 점들이 있었습니다. 서로 다른 생각을 가진 5명이 모여서 프로젝트 주제를 결정하고, 개발하는 과정은 처음 생각했던 것보다 쉽지 않았습니다. 하지만 지금 생각해 보면 그 힘들었던 과정들이 모두다 의미 있는 시간이었다고 생각이 됩니다. 하나의 의견으로 모으기 위해서 하던 회의, 하루에 10시간이 넘도록 같은 공간에서 하나의 프로젝트를 개발하던 팀원들, 생각했던 일이 제대로 진행되지 않아서 힘들었던 일들까지 모두 이번 프로젝트가 아니면 경험해보지 못했을 것입니다. 무엇보다 쉽게 넘어가지 못하고 고생하면서 하나의 결과를 낸 만큼 기억에 많이 남고, 함께 프로젝트를 진행하는 팀원들이 무엇보다 가장 중요하다는 것을 깨달은 의미 깊은 프로젝트였습니다.</p> |