**동계 특강**

**프로젝트**

|  |  |
| --- | --- |
| 프로젝트 명 | *나만의 스마트 룸 만들기* |
| 팀 명 | *김한결, 박준석* |
| 문서 제목 | 결과보고서 |

|  |  |
| --- | --- |
| **Version** | 1.0 |
| **Date** | 2015-MaR-04 |

|  |  |
| --- | --- |
| **팀원** | 김 한결 |
| 박 준석 |
|  |
|  |
|  |
| **지도교수** | 임 성수 교수 |

|  |
| --- |
| **CONFIDENTIALITY/SECURITY WARNING**  이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 학생 중 프로젝트 “동계 훈련 프로젝트”를 수행하는 팀 “김한결, 박준석”의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 “김한결,박준석”의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다. |

**문서 정보 / 수정 내역**

|  |  |
| --- | --- |
| **Filename** | 결과보고서-동계프로젝트.doc |
| **원안작성자** | 김한결, 박준석 |
| **수정작업자** | 김한결, 박준석 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 수정날짜 | 대표수정자 | Revision | 추가/수정 항목 | 내 용 |
| 2014-11-15 | 김한결 | 1.0 | 최초 작성 | 구현 내용 작성 |
| 2015-03-06 | 박준석 | 1.0 | 추가 작성 | 구현 내용 작성 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**목 차**

[**1** **프로젝트 목표** 4](#_Toc230855220)

[**2** **수행 내용 및 중간결과** 5](#_Toc230855221)

[**2.1** **계획서 상의 연구내용** 5](#_Toc230855222)

[**2.2** **수행내용** 5](#_Toc230855223)

[**3** **최종보고서 본문** 7](#_Toc230855224)

[**4** **자기평가** 8](#_Toc230855225)

# **프로젝트 목표**

집 외부 또는 내부에서 집 안의 각종 시설물들을 원격으로 제어하여 보다 쾌적한 삶을 누리고 나아가 IoT에 대해 어떤 것을 접목 할 수 있을지 깊이 생각하는 것이 목표입니다.

이 과정에서 외부 서버가 반드시 필요하게 됨으로 적절한 서버를 구현하는 것이 첫번째 목표 입니다.

두번째로 제어할 사물은 컴퓨터로써 인터넷을 기본적으로 사용가능하기 때문에 제어하기가 쉬울 것으로 예상됩니다. 단순히 같은 네트워크에서 컴퓨터의 전원을 제어하는데서 그치지 않고 다른 장소에서 컴퓨터를 켜고 끌 수 있게 만들며 원격데스크톱을 이용해 컴퓨터를 사용할 수 있도록 합니다. 데스크탑에서 서버로의 접속은 MFC 소켓프로그래밍을 통해서 구현합니다. 컴퓨터를 제어하는데는 WOL(Wake on Lan) 기술을 사용하며 홈 네트워크에 설치된 라즈베리파이가 매직패킷을 전송하는 구조로 만들 것 입니다.

두번째로 제어할 사물은 형광등으로써 실질적으로 인터넷이 안되는 사물에 인터넷을 연결시키는 것으로 본 프로젝트의 목표에 좀더 부합하는 대상이라고 할 수 있습니다. 실질적으로 전등을 밖에서 제어할 일이 많지는 않으나 생활하면서 여러가지 쓸모가 많을 것이라고 예상됩니다. 사물을 인터넷에 연결하기 위해서 라즈베리파이와 릴레이 스위치 등이 필요합니다.

# **수행 내용 및 중간결과**

## **계획서 상의 연구내용**

1. 서버의 구현
2. 사용 언어 : C 언어
3. 다중 접속 방식 : 멀티플렉싱(select 함수 이용)
4. AWS 서비스 사용
5. Socket 사용
6. 클라이언트들이 전송한 메시지를 다른 클라이언트에게 전송
7. 클라이언트들의 연결과 연결종료를 관리
8. 컴퓨터의 제어
9. 사용 언어 : MFC
10. 서버 접속시 MFC 소켓 프로그래밍 이용
11. 서버가 전송해주는 메시지를 해석하고 컴퓨터의 전원을 제어
12. 컴퓨터 부팅시 자동실행
13. 백그라운드 실행
14. 라즈베리파이
15. 라즈베리파이 GPIO 제어 언어 : wiringPi라이브러리
16. 형광등의 전원을 제어할 릴레이 스위치 구성
17. 서버와 통신하기 위한 소켓프로그램 작성 : C 언어
18. 서버에서 신호를 받고 WOL 매직패킷을 컴퓨터로 전송
19. 실제 스위치를 라즈베리파이로 대체
20. 안드로이드 스튜디오
21. 위의 모든 기능을 안드로이드에서 사용이 가능하도록 합니다.
22. 서버에 접속 하기 위해서 역시 소켓프로그램을 사용합니다.
23. 사용언어: 자바 와 XML

# 최종보고서 본문

## 구현 결과

1. 서버의 구현

AWS의 t1.micro 서버를 론치 해서 서버로 사용하고 있습니다. 서버와 클라이언트는 간단한 문자열을 주고받기만 하는 간단한 통신이 이루어질 예정이므로 소켓프로그래밍을 하기로 하였습니다.

그리고 클라이언트가 언제 접속 할지 모르므로 프로그램이 계속 작동하고 있어야 하며 접속과 접속종료시 발생할 수 있는 에러와 예기치못한 종료등을 처리해야 했습니다. 다중 접속의 구현방식은 멀티프로세싱, 멀티쓰레드, 멀티플렉싱이 있으나 멀티프로세싱은 전달해야 하는 문자열을 공유하지 못하고 또한 시스템리소스가 많이 필요한 작업이 아니므로 하나의 프로세스에서 모든걸 처리하는 멀티플렉싱이 가장 적절한 것으로 생각되어 선택하였습니다. 서버의 기능은 단순합니다. 클라이언트가 문자열을 서버로 전송하면 접속한 모든 클라이언트들에게 문자열을 전달하게 됩니다. 그리고 각 문자열에 해당하는 기능을 수행해야 하는 클라이언트는 특정기능을 실행하게 되는 구조 입니다.

프로젝트에서 사용하는 문자열은 4종류로 p0,p1, l0,l1 이 있습니다. P0, p1 은 컴퓨터의 on/off , l0, l1 은 형광등을 제어하는 역할입니다.

소스: <https://github.com/winterproject/server/blob/master/select.c>

1. 컴퓨터 제어

컴퓨터에서 수행해야 하는 기능은 서버로부터 특정 문자열(p0,p1)를 받게 되면 컴퓨터를 종료 또는 부팅하는 것 입니다. 먼저 종료하는 것은 대화상자 기반으로 프로그램을 만들어서 프로그램이 생성될 때 서버로 연결하여 문자열을 받는 것을 무한 반복 합니다. 그러다가 p0를 받게 되면 systemshutdown() 함수를 실행합니다. 이 함수는 윈도우를 종료 시킬 때 필요한 권한을 획득하여 윈도우를 종료해주는 함수 입니다. 백그라운드로 실행되어 사용자는 프로그램이 작동되는지 알 수 없습니다.

컴퓨터의 부팅에는 WoL(Wake on Lan)기술을 이용하여 랜카드를 전원이 종료된 후에도 신호를 대기하는 상태로 만듭니다. 이를 위해서 약간의 랜카드 설정이 필요합니다. 그 후에 그 컴퓨터에 매직패킷을 보내면 전원이 켜집니다. 매직 패킷은 FF FF FF FF FF FF 이후에 컴퓨터의 MAC어드레스를 16번 일렬로 중단없이 이은 102byte의 패킷입니다. 기본적으로 이 기술은 같은 네트워크에 있있을 때 작동하게 됩니다. 만약 외부 네트워크에서 이 기술을 사용하고 싶다면 켜고 싶은 컴퓨터와 같은 네트워크에 항상 켜져있는 컴퓨터가 필요합니다. 따라서 저희는 라즈베리파이를 사용하기로 하였습니다. 서버에서 p1이라는 문자열이 라즈베리파이의 프로그램에 전송되면 라즈베리파이가 wakeonlan이라는 이미 존재하는 프로그램으로 제 컴퓨터에 매직패킷을 보내서 전원이 켜지는 구조 입니다.

MFC 소스

<https://github.com/winterproject/mfcClient>

1. 라즈베리파이

라즈베리파이에서 수행해야 하는 기능은 p1 문자열에 wakeonlan 프로그램 실행과 형광등과 연결되어있는 릴레이를 l1문자열에 맞추어서 작동시키는 것입니다.

먼저 wakeonlan은 라즈베리파이용 os인 라즈비안에서 쉽게 설치할 수 있습니다. 그리고 나서 popen함수를 이용하여서 서버와 통신하는 소켓프로그램 내에서 wakeonlan을 실행시킬 수 있습니다.

다음으로 릴레이를 작동시켜야 하는데 릴레이는 라즈베리파이의 GPIO에 연결되어있고 또 이 GPIO핀의 입출력을 조작하는 것으로 작동시킬 수 있습니다. GPIO는 General perpose input/output 으로 말그대로 일반적인 입출력을 라즈베리파이가 받아들이는 곳입니다. 그리고 릴레이란 특정한 전압이 걸릴때에만 작동하는 스위치로써 라즈베리파이로 전기기구들을 제어하는데 용이한 스위치 입니다. GPIO를 작동시키는데 python을 사용할 수도 있고 C언어를 사용할 수도 있습니다만 서버가 C언어로 구성되어 있어서 자료형의 차이에서 오는 혼란을 막고 소스의 재사용이 용이하므로 C언어로 GPIO를 제어하기로 했습니다. 이미 C언어로 GPIO핀을 제어할 수 있는 라이브러리가 존재했는데 그것이 wiringPi입니다. 무료로 배포되는 것이고 사용하는 방법도 간편했습니다. 코딩할때 wiringPi.h 헤더파일을 include해주면 핀의 입출력을 digitalWrite(int, int)함수로 제어할 수 있습니다. 핀을 사용하기 전에 초기화 해주어야 하며 사용하고 난 뒤에 일괄적으로 초기화 해주는 함수가 없으므로 주의 해야 합니다. 저희 프로젝트에서는 light함수에서 GPIO핀을 제어하고 있으며 릴레이 하나만을 사용하므로 GPIO 7번핀만을 사용하고 있습니다.

라즈베리파이소스

<https://github.com/winterproject/clientTest/blob/master/testClient.c>

1. 안드로이드 앱

안드로이드 앱에서 수행해야 하는 기능은 어플의 전반적인 디자인을 구현하고 각 기능을 수행하여 프로그램이 동작할 수 있도록 작동시키는 것입니다. 우선 안드로이드 스튜디오에서 컴퓨터를 켜고 끄는 버튼과 전등을 켜고 끄는 버튼 등 총 4개의 버튼을 만들어 줍니다. 컴퓨터가 켜지고 꺼질 때에는 서버에 p0, p1의 신호를 보내주고 전등이 켜지고 꺼질 때에는 서버에 l0, l1의 신호를 보내줍니다. 버튼은 안드로이드 스튜디오에서 제공하는 아이콘을 드래그 앤 드롭을 하여 원하는 위치에 가져다 놓은 후 activity\_main.xml에서 각 버튼의 이름과 어플에서 보여지는 버튼의 이름을 설정해 준 뒤 onClik의 기능을 설정해 주고 난 뒤 MainActivity.java 에서 각 버튼이 클릭되었을 때 수행해야 할 내용들을 넣어줍니다. 각 버튼이 클릭 되었을 때 socket을 이용하여 서버에 접속한 뒤 컴퓨터 켜고 끌 때 또는 전등을 켜고 끌 때에 해당하는 신호를 각각 보내줍니다.

안드로이드 소스

https://github.com/winterproject/androidClient

# 자기평가

**김한결 :**  처음 프로젝트를 구상했을 때는 굉장히 구현하기가 어렵고 만든 사람이 없을 것 같았다. 하지만 막상 구현을 시작하니 이미 같은 생각을 가지고 시도하여 성공한 사람들이 많다는 것을 알게되었습니다. 그래서 어떠한 독창적인 서비스를 만드는 것이 굉장히 어렵다는것을 느끼게 되었습니다. 그리고 생소한 소켓프로그래밍에 도전하여 컴퓨터와 컴퓨터를 통신으로 연결하여 간단하게나마 데이터를 주고 받아보았는데 이 과정을 정확히 이해하지는 못하였으니 굉장히 신기하였고 재미있었습니다. 다시금 코딩하면서 호기심이 생기고 에너지가 생겼던 것 같습니다. 라즈베리파이도 굉장히 흥미로운 물건이었습니다. 이러한 물건을 가지고 무언가를 해보는 것이 처음인데 생각보다 굉장히 다양한 것이 가능하고 조작하는 방식도 쉽게 되어 있었습니다. 무엇보다 간단하게 주변에 여러 장치들을 (센서 등) 제어하기도 쉽고 작은데도 성능은 꽤 좋아서 왠만한 서버로도 이용이 가능 할 듯 싶습니다. 다음엔 이 서버로써의 기능을 사용하여 인터넷에서 자료를 찾는 프로그램을 만들어 보고 싶은 생각이 들었습니다. 다양한 새로운 것들을 접하고 다시금 흥미가 생겼다는 것이 가장 의미가 있었다고 생각합니다.

**박준석 :** 한결이와 처음 이야기를 하면서 목표가 나만의 스마트 룸을 만들자고 하였습니다. 그래서 처음에는 컴퓨터를 스스로 켜고 끄는 것을 구현하기로 하였습니다. 둘 다 이런 개발 쪽으로 코딩을 해보지 않아서 처음에는 우여곡절이 굉장히 많았습니다. 하루 종일 구글을 검색하면서 어떤 방향으로 해야 하는지 살펴보고 구현해보고 실패하고 다시 구현하고의 연속이었습니다. 저는 안드로이드 부분을 맡으면서 처음에는 이클립스로 집에서 버튼을 구현하고 각 버튼을 클릭하였을 때 신호를 보내도록 구현을 하고 한결이와 같이 동작을 시키려는데 리눅스에서 SDK와 ADT를 잘못 깔아서 인지 구현이 되지 않았다. 그래서 이번에는 이클립스가 아닌 안드로이드 스튜디오를 깔아서 사용법을 다시 익힌 후 작업을 하였습니다. 그런데 서버를 C언어로 짜고 안드로이드 스튜디오는 자바여서 통신에 이상이 생겨 난항을 겪었습니다. 이렇게 작업을 수행하면서 문제점을 발견하고 같이 작업하는 경험을 하여 뜻 깊은 프로젝트였습니다. 앞으로 이런 작업을 많이 하면서 실력을 향상시키면 좋을 것 같습니다.