

설계 프로젝트

Raspberry Pi Music Player

Linux Server와 Raspberry Pi의

TCP Socket 통신을 통한 음악 플레이어

2009104027	2013104110	2012104030	2010104092
오세인	장윤경	정준영	하재권

1. 프로젝트 개요

본 프로젝트는 Linux서버와 Raspberry Pi의 Socket 통신을 통해 음악 파일을 전달 받고, IR Remote Controller를 사용하여 음악의 재생이 가능하도록 한 "라즈베리파이 뮤직 플레이어"이다. Raspberry Pi에는 LCD 모듈과 IR 센서가 연결되어 있으며 IR Remote Controller의 Play 버튼을 누르게 되면, LCD 모듈의 글씨가 변하면서 노래가 재생된다.

프로젝트의 작업 환경과 역할은 다음과 같다.

Server

OS	Ubuntu 12.04.5 LTS
Program Language	C

Server는 음악 파일(example.wav)를 가지고 있으며, 이를 TCP Socket 통신을 통해 Client로 전송한다.

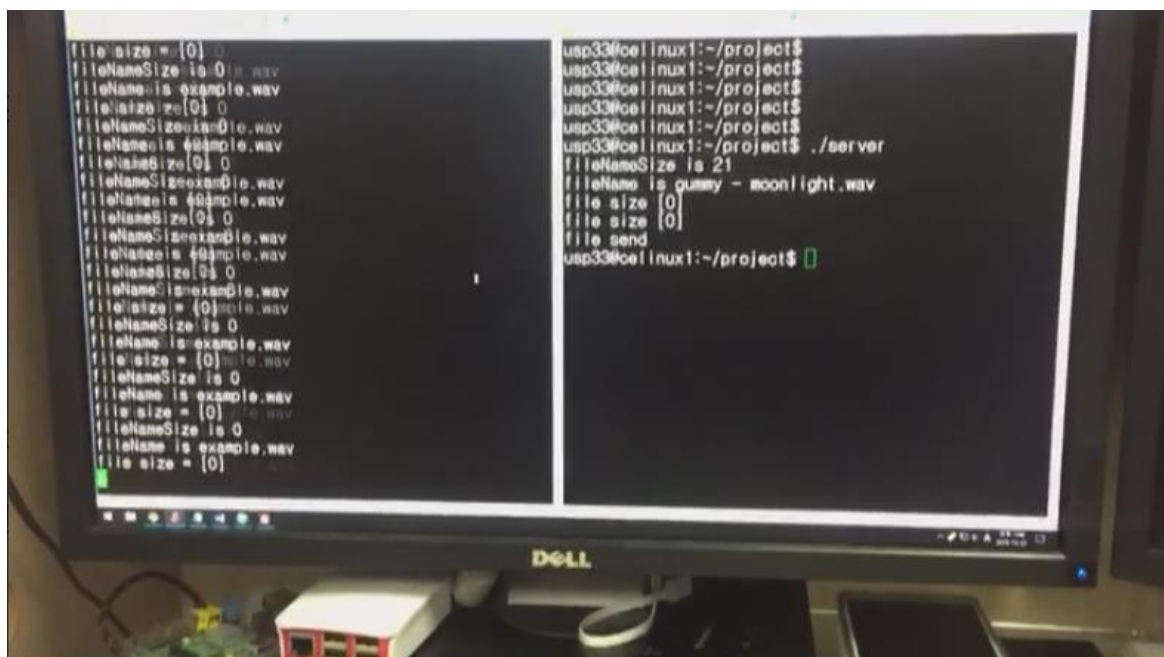
Client

Device	Raspberry Pi 3
OS	Raspbian GNU/Linux 8
LCD Module	QAPASS 1602A
IR Sensor	CHQ 1838
Speaker	Hamburger mini speaker
Program Language	C

Client에 전원이 인가되면 "Hello USP"라는 문구를 LCD Module에 출력한다. 이 후, Server로부터 TCP Socket 통신을 통해 음악 파일(example.wav)를 받게 되면, IR Remote Controller의 신호를 기다린다.



< 그림 1. 대기 상태의 Raspberry Pi : "Hello USP"를 LCD에 출력 >



< 그림 2. TCP Socket 통신을 통해 Server(우)에서 Client(좌)로 음악파일의 전송 >

IR Remote Controller의 신호를 받게 되면, 음악 파일(example.wav)를 재생시키고 LCD Module에 "Play"라는 문구를 출력한다.



< 그림 3. IR Remote Controller를 통한 음악 재생 : LCD에 "Play"가 출력되고 노래 재생 >

2. 사전 조사

프로그램의 구현을 위해 단계별 과정을 거치는 가운데, 실제 구현을 위하여 필요한 내용을 정리하고, 시행 착오 등을 방지하고 수정하기 위하여 사전에 조사된 내용은 다음과 같다.

1. 리눅스 시스템에서의 TCP / IP 통신방법

TCP / IP 통신의 근본적인 원리는 같지만, 윈도우와 리눅스 시스템은 함수의 사용 방법이나 통신 방법이 다르기 때문에, 라즈베리파이의 OS인 리눅스에 맞는 TCP/IP 프로그램을 베이스로 하여 TCP / IP 통신을 구현해야 할 필요가 있다.

2. 리눅스에서의 준비된 음원 재생

윈도우에서는 라이브러리에서 winmm을 추가하여, PlaySound 함수를 통해 wav파일을 재생 하면 쉽게 문제를 해결 할 수 있지만, 라즈베리 기반인 리눅스에서는 해당 함수를 사용하기에 어려움이 있기 때문에 libao 함수 등을 다른 방법을 통해 구현 할 필요가 있다.

3. IR REMOTE의 제어와 사용

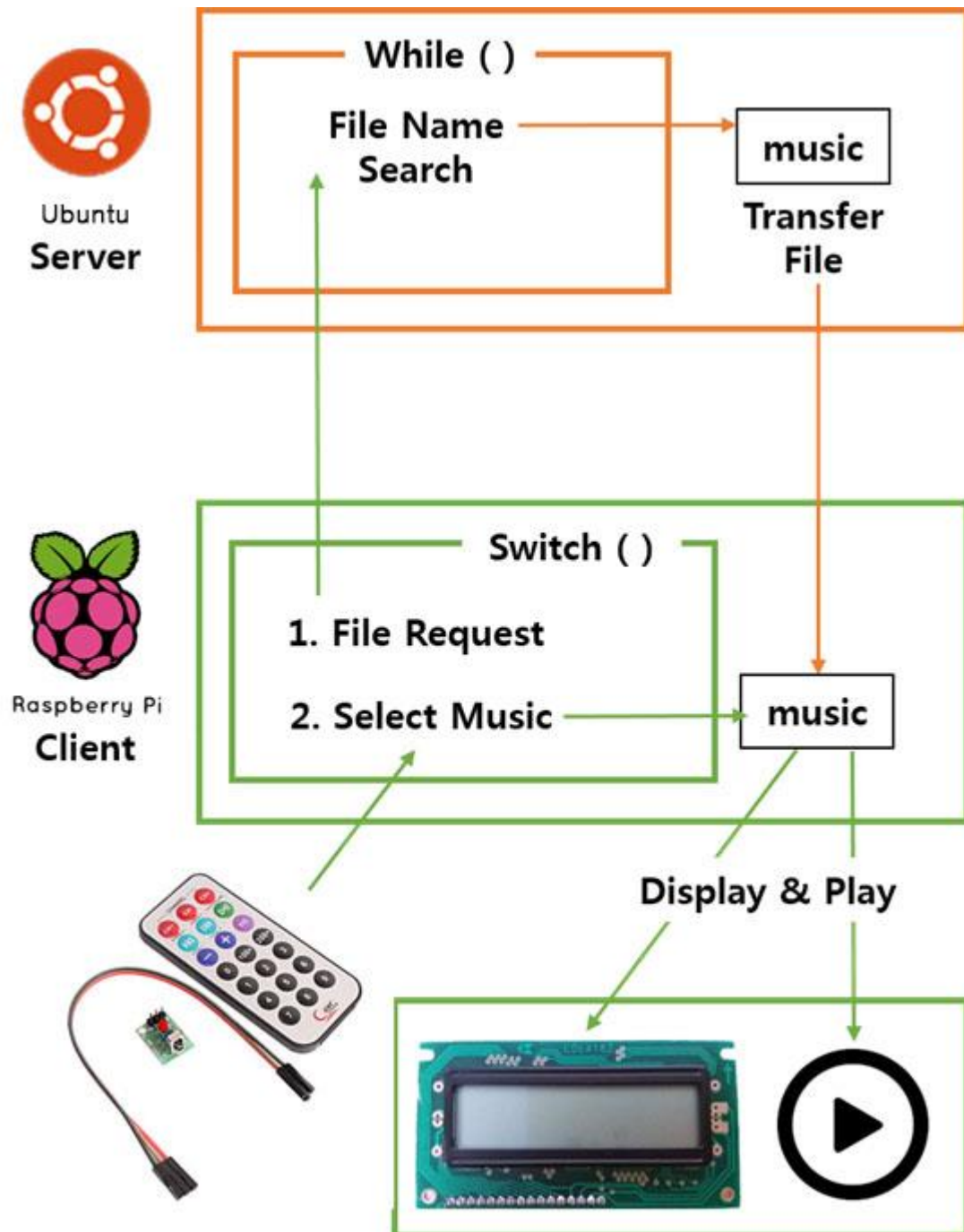
Raspberry Pi에서 IR Sensor를 사용하기 위한, 필요 라이브러리 설치 및 함수 사용과 관련된 설계/구현 방법을 익혀야 할 필요가 있다.

4. LCD 사용

Raspberry Pi에서 LCD를 사용하여 원하는 정보를 출력하기 위한, 라이브러리 설치 및 함수 사용과 관련된 설계/구현 방법을 익혀야 할 필요가 있다.

3. 프로젝트 설계

본 프로젝트의 구성도는 다음과 같다.



< 그림 4. 프로젝트 구성도 >

4. 결론

Unix System Programming 강의에서 우리는 Unix/Linux System상에서 코딩을 하고, 컴파일 및 링킹 하는 법을 배웠다. 또한 C언어를 사용하여 File I/O, Scket Programming 등을 구현하였다. 본 프로젝트에서 사용한, Unix System Programming 강의 내용은 다음과 같다.

<p>TCP 통신을 통한 File I/O</p>	<pre>// while loop as much as the number of file while(1) { // file name transfer char* fileName = "example.wav"; int fileNameSize = strlen(fileName); printf("fileNameSize is %d\n", strlen(fileName)); printf("fileName is %s\n", fileName); send(client_sockfd, &fileNameSize, sizeof(fileNameSize), 0); send(client_sockfd, fileName, strlen(fileName), 0); // file transfer file = fopen("example.wav", "rb"); fseek(file, 0, SEEK_END); fsize = ftell(file); fseek(file, 0, SEEK_SET); printf("file size [%d]\n", fsize); fsize2 = htonl(fsize); printf("file size [%d]\n", fsize2); send(client_sockfd, &fsize, sizeof(fsize), 0); while(nsize != fsize) { fsize = fread(buf, 1, 256, file); nsize = nsize+fsize; send(client_sockfd, buf, fsize, 0); } printf("file send \n"); fclose(file); // break because the number of file is 1 break; }</pre> <p>< SERVER: server.c ></p>
<p>System 함수를 사용한 음악 재생</p>	<pre>else if(strstr(code, "KEY_PLAY")) { downloadFromServer(); lcdPuts(lcd, "PLAY"); system("omxplayer -o local example.wav"); buttonTimer = millis(); }</pre> <p>< CLIENT: client.c ></p>

<p>IR Sensor를 통한 Socket 통신</p> <p>및</p> <p>LCD Module를 통한 File I/O</p>	<pre> if(lirc_readconfig(NULL, &config, NULL) == 0) { while(lirc_nextcode(&code) == 0) { if(code == NULL) continue; { if(millis() - buttonTimer > 400) { if(strstr(code, "KEY_PREVIOUS")) { printf("PREVIOUS"); lcdPuts(lcd, "PREVIOUS"); buttonTimer = millis(); } else if(strstr(code, "KEY_NEXT")) { printf("NEXT"); lcdPuts(lcd, "NEXT"); buttonTimer = millis(); } else if(strstr(code, "KEY_PLAY")) { downloadFromServer(); lcdPuts(lcd, "PLAY"); system("omxplayer -o local example.wav"); buttonTimer = millis(); } else { lcdPuts(lcd, "ERROR"); buttonTimer = millis(); } } } } } </pre> <p>< CLIENT : client.c ></p>
--	--

이 외에도 Client 구현을 위해 사용한 wiringPi, lirc 라이브러리는 “Processes and Threads”와 “Signal” 강의 시간에 배운 *fork*, *interrupt* 등이 사용되었다.

본 프로젝트의 설계와 구현과정을 통해 Unix System Programming 강의에서 배운 내용을 복습하고 응용하는 법을 공부할 수 있었으며, 프로젝트의 완성도는 훌륭하진 않지만 기초적인 내용은 모두 포함하였다 생각한다.

5. 참고 자료

[1] Linux의 TCP / IP 통신

http://www.linuxhowtos.org/C_C++/socket.htm

[2] Linux TCP / IP 프로그래밍 개요

http://forum.falinux.com/zbxe/index.php?document_srl=406054&mid=network_programming

[3] PLAY MP3 with Libao

<http://hzqtc.github.io/2012/05/play-mp3-with-libmpg123-and-libao.html>

[4] HOW TO CONTROL GPIO IR REMOTE

<http://ozzmaker.com/how-to-control-the-gpio-on-a-raspberry-pi-with-an-ir-remote/>

[5] LCD Library

<https://projects.drogon.net/raspberry-pi/wiringpi/lcd-library/>