­­

과목명 : IOT시스템응용

담당교수 : 봉진숙

학번 : 20170660

학과 : 컴퓨터공학과

이름 : 김동현

**REPORT**

**1. led.client**

코드)

package com.dong.led\_client;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.os.Handler;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.io.OutputStream;

import java.io.PrintWriter;

import java.net.Socket;

public class MainActivity extends AppCompatActivity {

Socket socket;

OutputStream os;

InputStream is;

BufferedReader in;

PrintWriter out;

ConnectThread thread;

Button btnConnect, btnOn, btnOff, btnDisconnect;

EditText edServerIP;

TextView tvMessage;

Handler handler = new Handler();

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

tvMessage = (TextView) findViewById(R.id.tvMessage);

edServerIP = (EditText) findViewById(R.id.edServerIP);

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

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

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

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

btnConnect.setEnabled(true);

btnOn.setEnabled(false);

btnOff.setEnabled(false);

btnDisconnect.setEnabled(false);

btnConnect.setOnClickListener(new View.OnClickListener() {

public void onClick(View v) {

String addr = edServerIP.getText().toString().trim();

thread = new ConnectThread(addr);

thread.start();

btnConnect.setEnabled(false);

btnDisconnect.setEnabled(true);

btnOn.setEnabled(true);

btnOff.setEnabled(true);

}

});

btnDisconnect.setOnClickListener(new View.OnClickListener() {

public void onClick(View v) {

thread.setStop();

tvMessage.setText("LED 상태 : ");

btnConnect.setEnabled(true);

btnDisconnect.setEnabled(false);

btnOn.setEnabled(false);

btnOff.setEnabled(false);

}

});

btnOn.setOnClickListener(new View.OnClickListener() {

public void onClick(View v) {

new Thread() {

public void run() {

out.println("on");

out.flush();

thread.readServer();

}

}.start();

}

});

btnOff.setOnClickListener(new View.OnClickListener() {

public void onClick(View v) {

new Thread() {

public void run() {

out.println("off");

out.flush();

thread.readServer();

}

}.start();

}

});

}

// 소켓 연결할 스레드 정의

class ConnectThread extends Thread {

String hostname;

public ConnectThread(String addr) {

hostname = addr;

}

public void run() {

try {

int port = 9000;

socket = new Socket(hostname, port);

os = socket.getOutputStream();

is = socket.getInputStream();

in = new BufferedReader(new InputStreamReader(is));

out = new PrintWriter(os);

} catch (IOException e) {

e.printStackTrace();

try {

socket.close();

} catch (IOException ex) {

ex.printStackTrace();

}

}

}

public void readServer() {

try {

final String msg1 = in.readLine();

handler.post(new Runnable() {

@Override

public void run() {

tvMessage.setText("LED 상태 : " + msg1);

}

});

} catch (IOException e) {

e.printStackTrace();

}

}

public void setStop() {

if(socket.isConnected()) {

try {

socket.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

}

}

코드) activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

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

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

tools:context=".MainActivity">

<TextView

android:id="@+id/TextView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_weight="0"

android:text='버튼을 누르면 소켓 연결됩니다. "' />

<EditText

android:id="@+id/edServerIP"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_marginTop="20dp"

android:ems="10"

android:gravity="center"

android:hint="연결할 소켓 서버 IP"

android:inputType="textPersonName"

android:text="192.168.219.109"

android:textSize="16sp"

android:textStyle="bold" />

<Button

android:id="@+id/btnConnect"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_marginTop="20dp"

android:text="서버 연결하기"

android:textSize="20sp"

android:textStyle="bold" />

<Button

android:id="@+id/btnOn"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_marginTop="20dp"

android:text="LED 켜기"

android:textSize="20sp"

android:textStyle="bold" />

<Button

android:id="@+id/btnOff"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_marginTop="20dp"

android:text="LED 끄기"

android:textSize="20sp"

android:textStyle="bold" />

<TextView

android:id="@+id/tvMessage"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_marginTop="20dp"

android:hint="LED 상태: "

android:textSize="16sp"

android:textStyle="bold" />

<Button

android:id="@+id/btnDisconnect"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_marginTop="20dp"

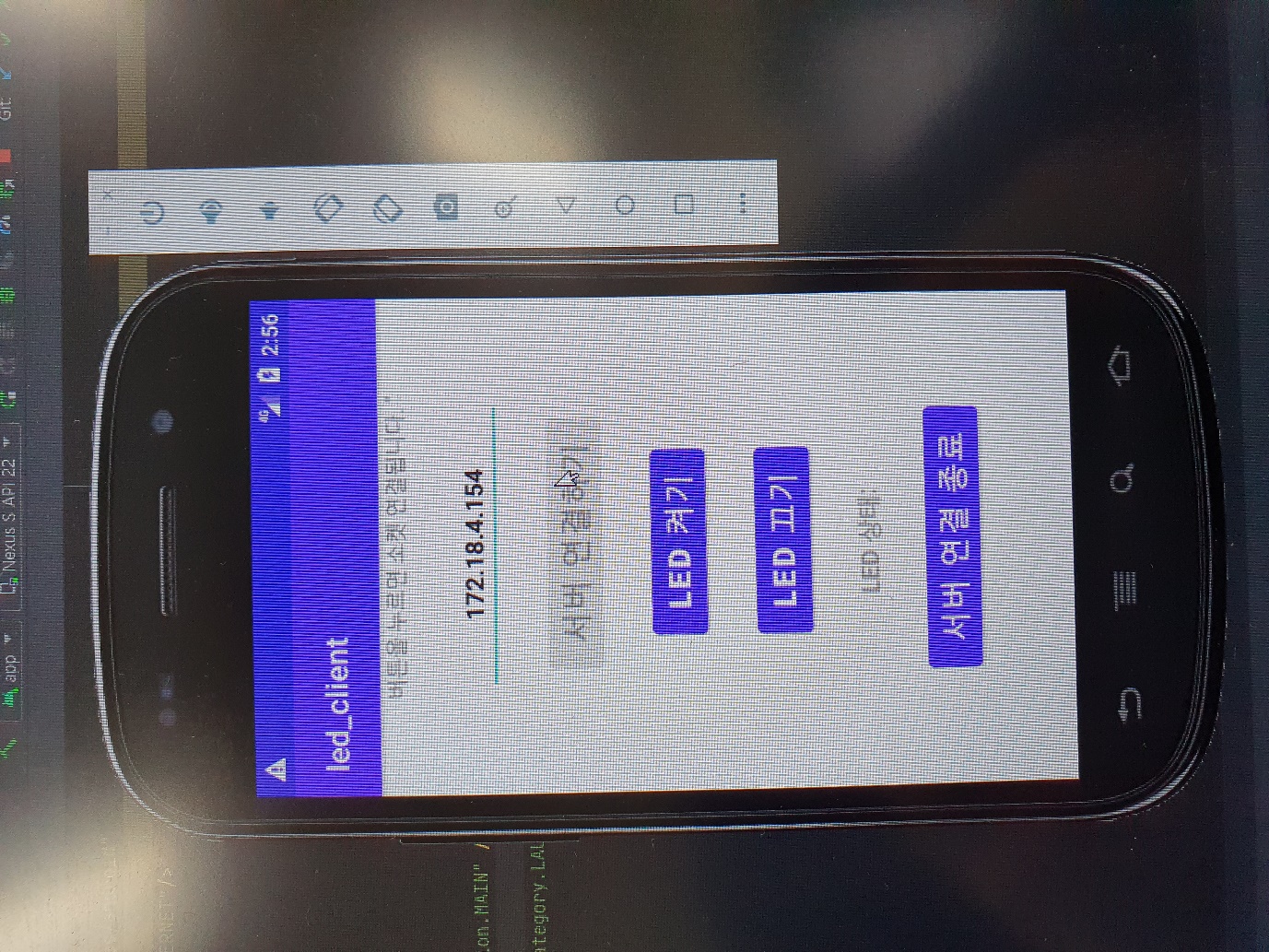
android:text="서버 연결 종료"

android:textSize="20sp"

android:textStyle="bold" />

</LinearLayout>

실행 결과)



**2. led\_server**

코드)

#include <stdio.h>

#include <string.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#include <wiringPi.h>

#include <unistd.h>

#define PORT 9000

#define LED 4

int main(void){

    if(wiringPiSetupGpio() == -1)

        return -1;

    pinMode(LED, OUTPUT);

    digitalWrite(LED, LOW);

    int s\_socket, c\_socket;

    struct sockaddr\_in s\_addr, c\_addr;

    int n;

    int len;

    char rcvBuffer[BUFSIZ];

    s\_socket = socket(PF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

    memset(&s\_addr, 0, sizeof(s\_addr));

    s\_addr.sin\_addr.s\_addr = htonl(INADDR\_ANY);

    s\_addr.sin\_family = AF\_INET;

    s\_addr.sin\_port = htons(PORT);

    if(bind(s\_socket, (struct sockaddr\*)&s\_addr, sizeof(s\_addr)) == -1){

        printf("Can not Bind!!!\n");

        return -1;

    }

    if(listen(s\_socket, 5) == -1){

        printf("Listen Fail!!!\n");

        return -1;

    }

    printf("LED Server started...\n");

    while(1){

        len = sizeof(c\_addr);

        c\_socket = accept(s\_socket, (struct sockaddr\*)&c\_addr, &len);

        printf("Connected IP : %s\n", inet\_ntoa(c\_addr.sin\_addr));

        while((n = read(c\_socket, rcvBuffer, sizeof(rcvBuffer))) > 0){

            rcvBuffer[n] = '\0';

            if(strncmp(rcvBuffer, "on", 2) == 0){

                printf("%s", rcvBuffer);

                digitalWrite(LED, LOW);

                delay(500);

            }

            write(c\_socket, rcvBuffer, n);

        }

    }

    return 0;

}

실행 결과)

