안드로이드로 LED 켜고 끄기

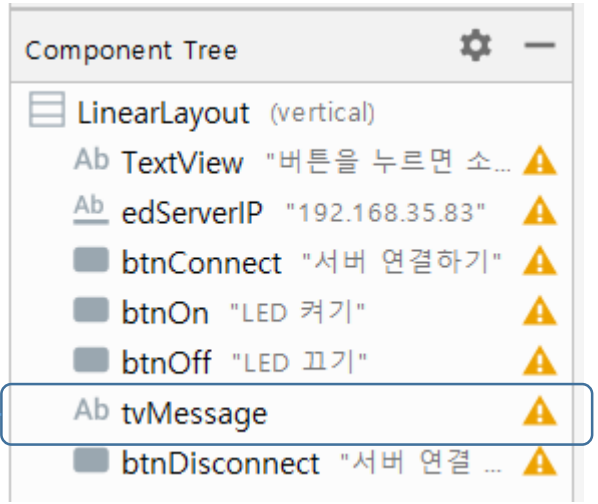
라즈베리파이_LED제어

```
1.  #include <stdio.h>
2.  #include <wiringPi.h>

3.  #define LED 4
4.  int main(void) {
5.      if(wiringPiSetupGpio()==-1)
6.          return 1;

7.      pinMode(LED, OUTPUT);
8.      digitalWrite(LED, LOW);

9.      digitalWrite(LED, HIGH);
10.     delay(500);
11.     digitalWrite(LED, LOW);
12.     delay(500);
13.     return 0;
14. }
```

LED 소켓 통신_서버 (led_server.c)

```
1.  #include <stdio.h>
2.  #include <string.h>
3.  #include <sys/socket.h>
4.  #include <arpa/inet.h>
5.  #include <wiringPi.h>
6.  #include <unistd.h>

7.  #define PORT 9000
8.  #define LED 4

9.  int main(void){
10.
11.     if(wiringPiSetupGpio() == -1)
12.         return -1;
13.     pinMode(LED, OUTPUT);
14.     digitalWrite(LED, LOW);

15.     int  s_socket, c_socket;
16.     struct sockaddr_in s_addr, c_addr;

17.     int n;
18.     int len;
19.     char rcvBuffer[BUFSIZ];

20.     s_socket = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP);

21.     memset(&s_addr, 0, sizeof(s_addr));
22.     s_addr.sin_addr.s_addr = htonl(INADDR_ANY);
23.     s_addr.sin_family = AF_INET;
24.     s_addr.sin_port = htons(PORT);
25.     if(bind(s_socket, (struct sockaddr*)&s_addr, sizeof(s_addr)) == -1){
26.         printf("Can not Bind!!!₩n");
27.         return -1;
28.     }

29.     if(listen(s_socket, 5) == -1){
30.         printf("Listen Fail!!!₩n");
31.         return -1;
32.     }

33.     printf("LED Server started...₩n");
34.     while(1){
35.         len = sizeof(c_addr);
36.         c_socket = accept(s_socket, (struct sockaddr*)&c_addr, &len);
37.         printf("Connected IP : %s₩n", inet_ntoa(c_addr.sin_addr));

38.         while((n = read(c_socket, rcvBuffer, sizeof(rcvBuffer))) > 0){
39.             rcvBuffer[n] = '₩0';
40.             if(strncmp(rcvBuffer, "on", 2) == 0){
41.                 printf("%s", rcvBuffer);
42.                 digitalWrite(LED, HIGH);
43.                 delay(500);
44.             }else if(strncmp(rcvBuffer, "off", 3) == 0){
45.                 printf("%s", rcvBuffer);
46.                 digitalWrite(LED, LOW);
47.                 delay(500);
48.             }
49.             write(c_socket, rcvBuffer, n);
50.
51.         }
52.     }
53.     return 0;
54. }
```

LED 소켓 통신_클라이언트 (led_client)




```
<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="16dp"
    android:textStyle="bold" />
```

안드로이드 소켓 통신 예제_클라이언트

```
package com.bong.led_client;

import …

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);
    }
});
```

안드로이드 소켓 통신 예제_클라이언트

```
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 (Exception ex) {
            ex.printStackTrace();
            try {
                socket.close();
            }catch (Exception e){
                e.printStackTrace();
            }

        }
    }
```

안드로이드 소켓 통신 예제_클라이언트

```
    public void readServer(){
        try {
            String msg1 = in.readLine();

            handler.post(new Runnable() {
                @Override
                public void run() {
                    tvMessage.setText("LED 상태 : "+msg1);
                }
            });
        }catch (Exception e){
            e.printStackTrace();
        }
    }
```

```
    public void setStop() {
        if(socket.isConnected()) {
            try {
                socket.close();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
```

```
}
}
```

AndroidManifest

activity_main.xml | MainActivity.java | AndroidManifest.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.bong.led_client">
    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="LED_client"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.LED_client">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
```