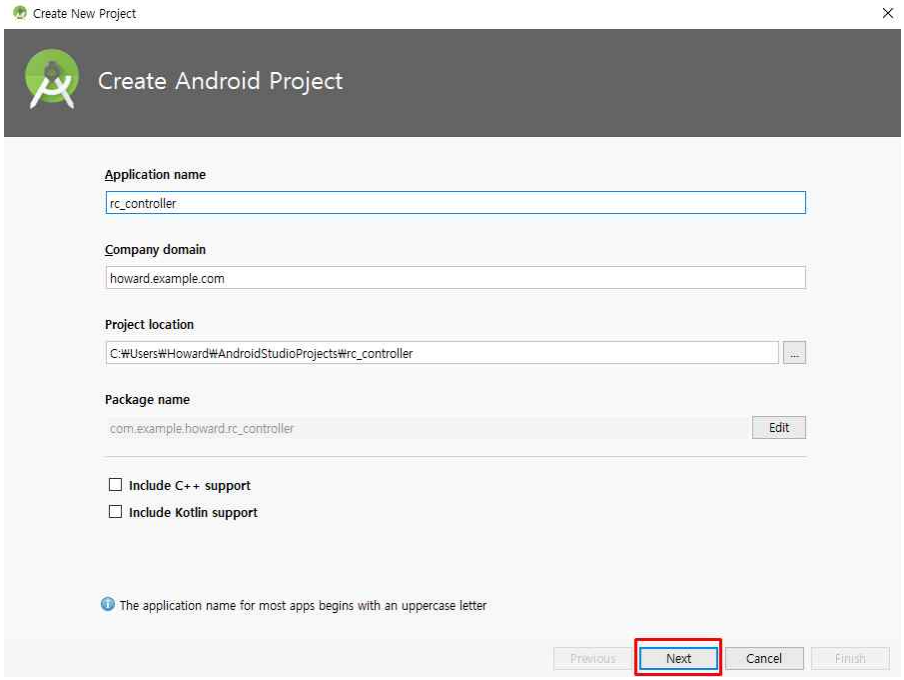
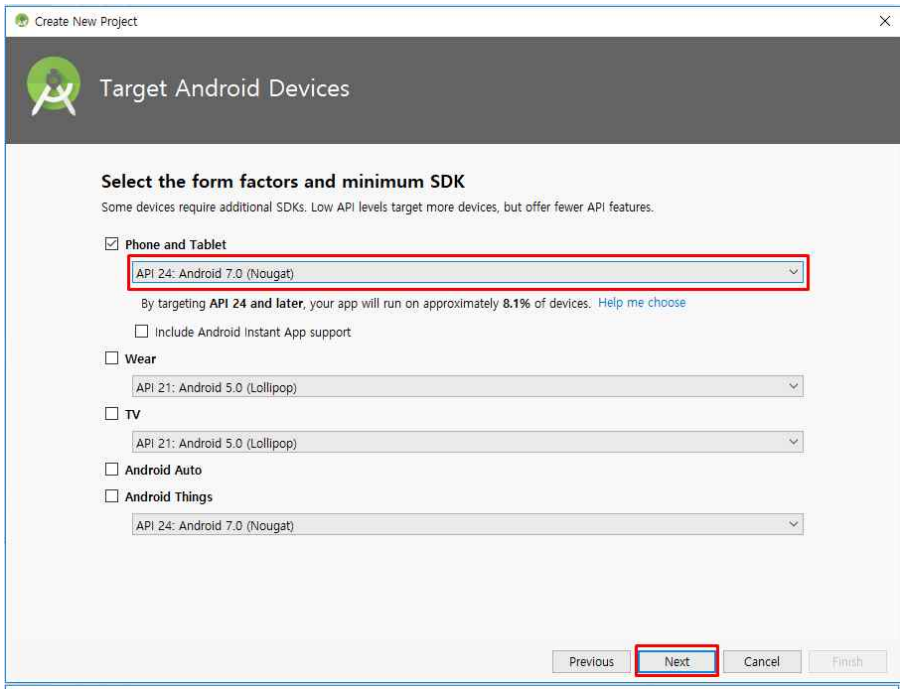
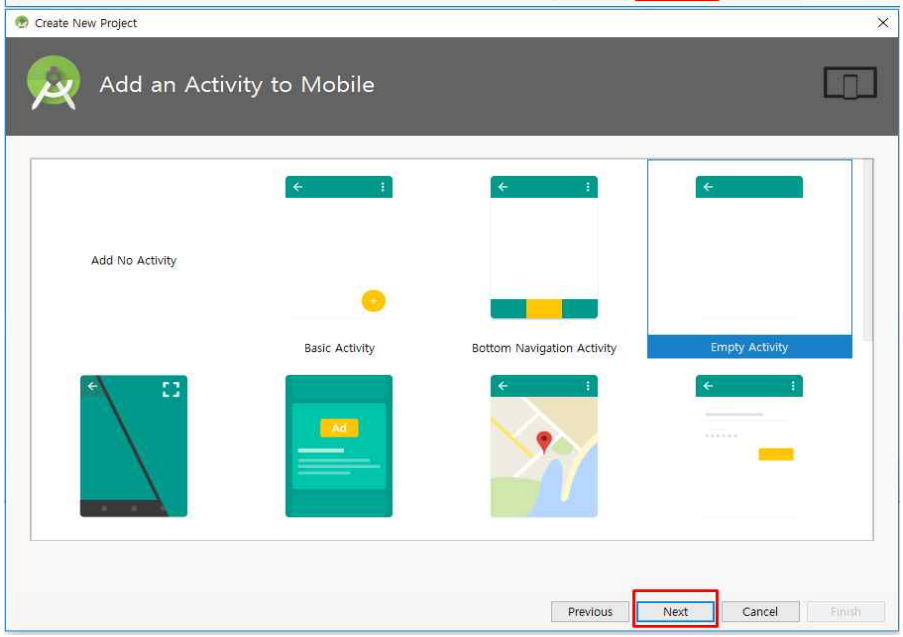
Xilinx Zynq FPGA, TI DSP, MCU 기반의 회로 설계 및 임베디드 전문가 과정

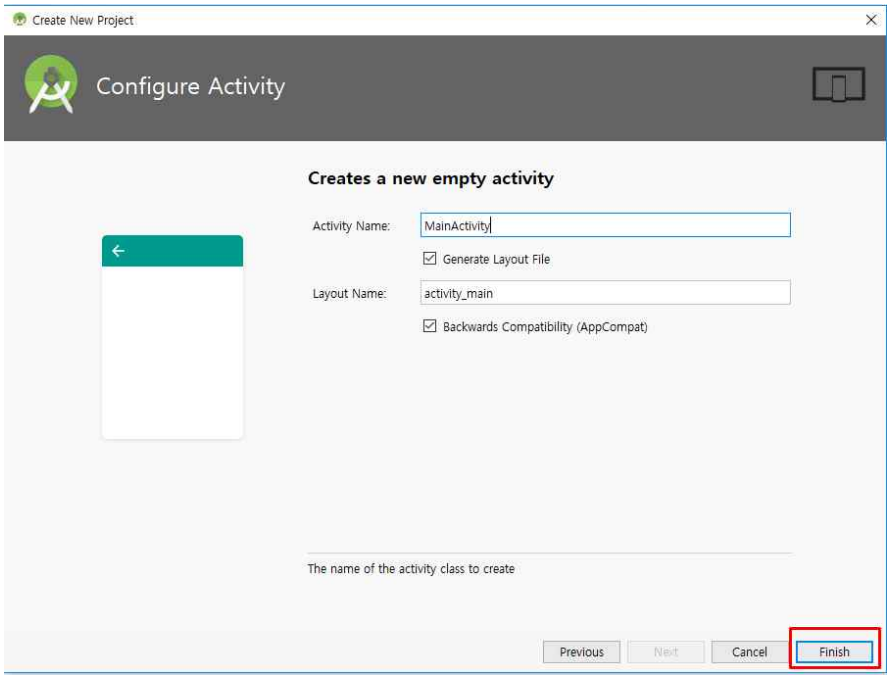
강사 - Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 - Hyungjoo Kim(김형주)
mihaelkel@naver.com

1.안드로이드 스튜디오를 실행하여, 프로젝트를 생성한다.





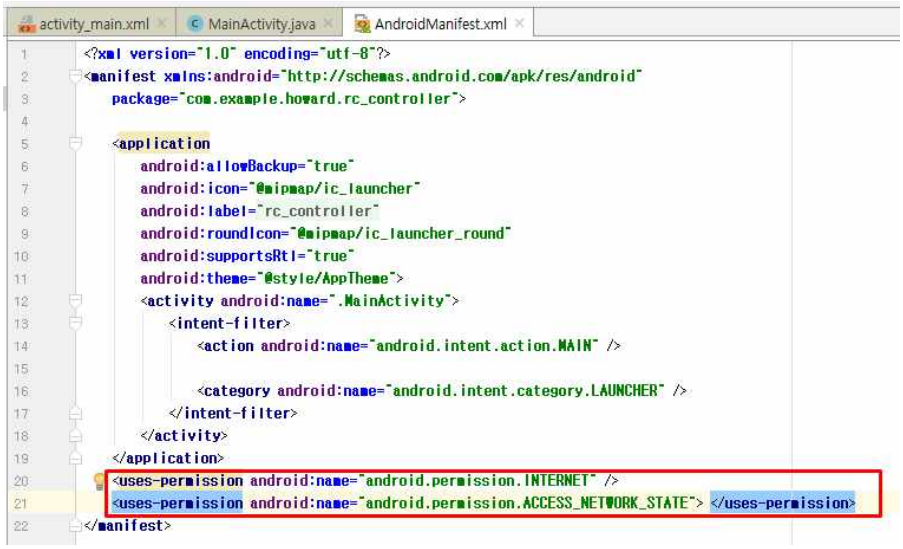




2.app -> manifests ->AndroidManifest.xml

AndroidManifest.xml 파일에 아래 두줄을 추가한다. 이 두 줄을 추가해야 소켓 접속이 가능해진다.

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"> </uses-permission>
```



3.Gradle Scripts -> build.gradle (Module:app) 에 들어가서 SDK 버전을 확인한다.
아래와 같이 28버전이라면, 27버전으로 수정해야 한다.(28버전은 UI 코드가 안먹음)

activity_main.xml | rc_controller | app | MainActivity.java | AndroidManifest.xml

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 28
    defaultConfig {
        applicationId "com.example.howard.rc_controller"
        minSdkVersion 24
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:28.0.0-beta01'
    implementation 'com.android.support.constraint:constraint-layout:1.1.2'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
}
```

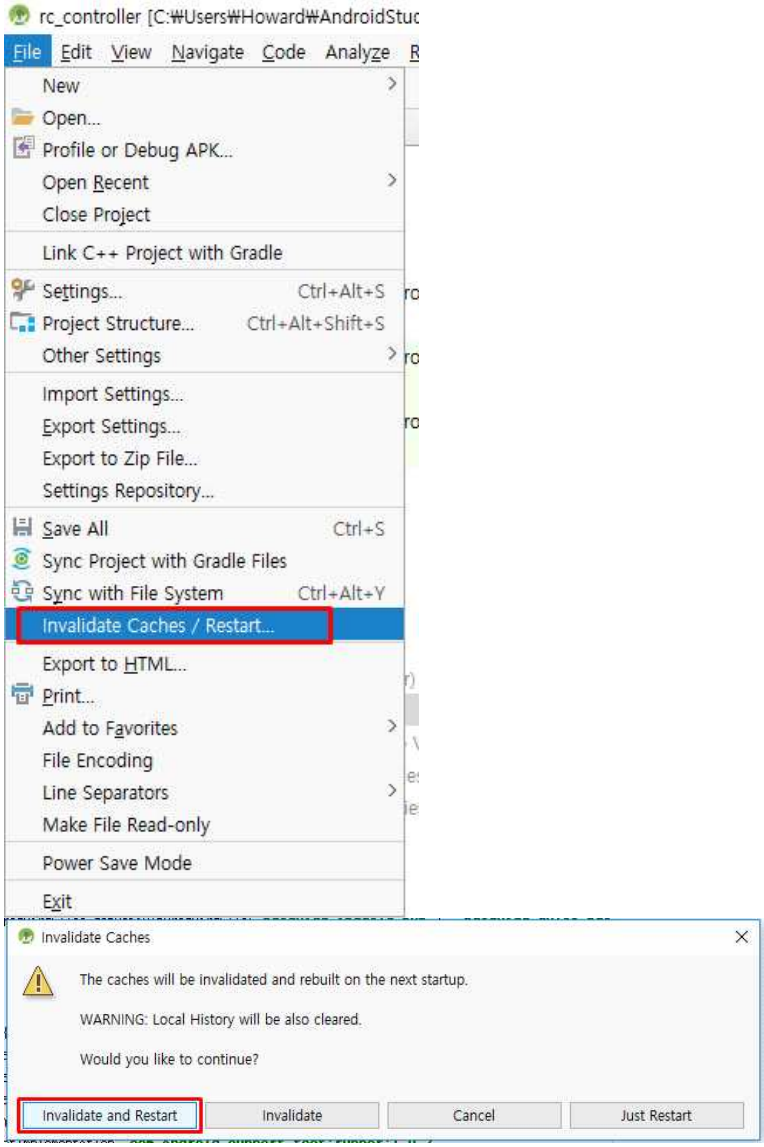
3-1.28.0.0을 27.0.2 버전으로 바꾸고, beta01을 지운다.

activity_main.xml | rc_controller | app | MainActivity.java | AndroidManifest.xml

Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly.

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 27
    defaultConfig {
        applicationId "com.example.howard.rc_controller"
        minSdkVersion 24
        targetSdkVersion 27
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:27.0.2'
    implementation 'com.android.support.constraint:constraint-layout:1.1.2'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
}
```

4.File -> Invalidate Caches / Restart...를 클릭한다.



5.다시 실행되는 데 약 2분정도 걸리는 것 같다. 완료되면 ui를 수정한다.

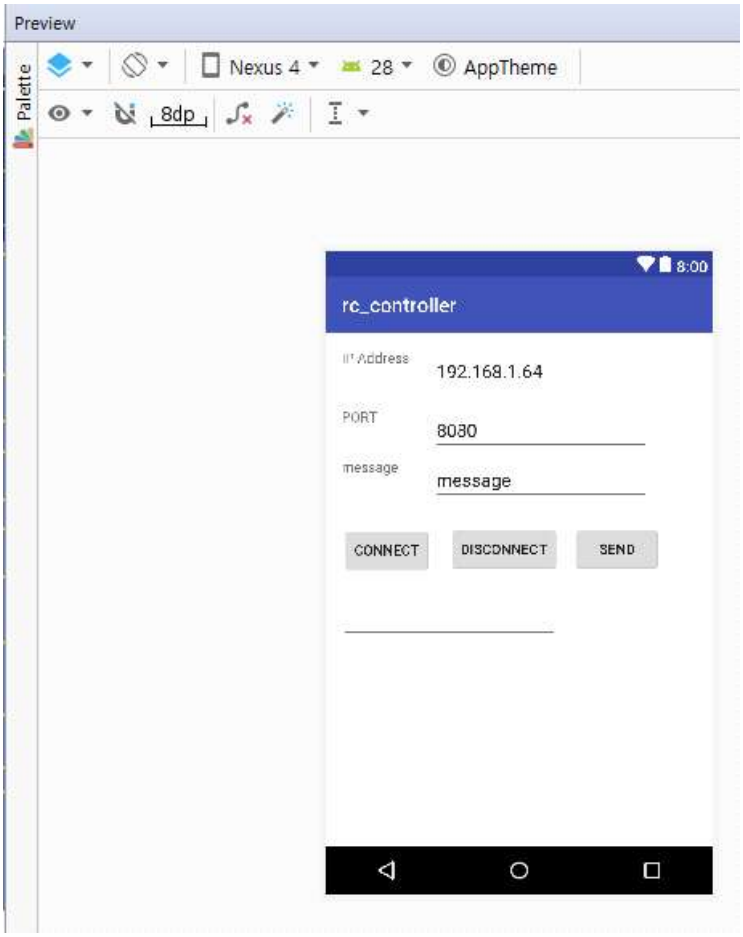
apps -> res -> layout -> activity_main.xml

```
<?xml version="1.0" encoding="utf-8"
<android.support.constraint.ConstraintLayout 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"
tools:context=".MainActivity">

<TextView
android:id="@+id/ip_viewer"
android:layout_width="0dp"
android:layout_height="0dp"
android:layout_marginBottom="197dp"
android:layout_marginEnd="10dp"
android:layout_marginStart="16dp"
android:layout_marginTop="16dp"
android:text="IP Address"
app:layout_constraintBottom_toTopOf="@+id/recv_msg"
app:layout_constraintEnd_toStartOf="@+id/input_ip"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />

<TextView
android:id="@+id/port_viewer"
android:layout_width="0dp"
android:layout_height="42dp"
android:layout_marginBottom="57dp"
android:layout_marginEnd="13dp"
android:layout_marginStart="16dp"
android:layout_marginTop="59dp"
android:text="PORT"
```

```
app:layout_constraintBottom_toBottomOf="@+id/msg_viewer"
app:layout_constraintEnd_toStartOf="@+id/input_port"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="@+id/input_ip" />

<TextView
android:id="@+id/msg_viewer"
android:layout_width="0dp"
android:layout_height="49dp"
android:layout_marginBottom="67dp"
android:layout_marginEnd="13dp"
android:layout_marginStart="16dp"
android:layout_marginTop="63dp"
android:text="message "
app:layout_constraintBottom_toBottomOf="@+id/connect_button"
app:layout_constraintEnd_toStartOf="@+id/input_msg"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/input_ip" />

<EditText
android:id="@+id/input_ip"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginEnd="63dp"
android:layout_marginTop="16dp"
android:ems="10"
android:inputType="textPersonName"
android:text="192.168.1.64"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toEndOf="@+id/ip_viewer"
app:layout_constraintTop_toTopOf="parent" />

<EditText
android:id="@+id/input_port"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginEnd="63dp"
android:layout_marginTop="13dp"
android:ems="10"
android:inputType="textPersonName"
android:text="8080"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toEndOf="@+id/port_viewer"
app:layout_constraintTop_toBottomOf="@+id/input_ip" />

<EditText
android:id="@+id/input_msg"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginEnd="63dp"
android:layout_marginTop="4dp"
android:ems="10"
android:inputType="textPersonName"
android:text="message"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toEndOf="@+id/msg_viewer"
app:layout_constraintTop_toBottomOf="@+id/input_port" />

<Button
android:id="@+id/connect_button"
android:layout_width="89dp"
android:layout_height="wrap_content"
android:layout_marginBottom="21dp"
android:layout_marginStart="16dp"
android:text="Connect"
app:layout_constraintBottom_toTopOf="@+id/recv_msg"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_chainStyle="packed" />

<Button
android:id="@+id/disconnect_button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginEnd="13dp"
android:layout_marginTop="22dp"
android:text="Disconnect"
app:layout_constraintEnd_toStartOf="@+id/send_button"
app:layout_constraintTop_toBottomOf="@+id/input_msg" />

<Button
android:id="@+id/send_button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginEnd="51dp"
android:text="SEND"
```

```
app:layout_constraintBaseline_toBaselineOf="@+id/disconnect_button"
app:layout_constraintEnd_toEndOf="parent" />

<EditText
android:id="@+id/recv_msg"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginBottom="12dp"
android:ems="10"
android:inputType="textMultiLine"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toStartOf="@+id/disconnect_button"
app:layout_constraintHorizontal_bias="0.558"
app:layout_constraintStart_toStartOf="@+id/disconnect_button"
app:layout_constraintTop_toBottomOf="@+id/connect_button" />


</android.support.constraint.ConstraintLayout>
```

위의 코드를 작성하면 아래와 같이 ui가 생성된다.



6.app -> java -> com.example.name.rc_controller -> MainActivity로 가서 메인 코드를 작성한다.
복제시 맨 윗줄의 package 부분을 주의한다. (package com.example.howard.rc_controller;)

```
package  com.example.howard.rc_controller;

import  android.annotation.SuppressLint;
import  android.os.Handler;
import  android.os.Message;
import  android.os.StrictMode;
import  android.support.v7.app.AppCompatActivity;
import  android.os.Bundle;
import  android.util.Log;
import  android.view.View;
import  android.widget.Button;
```

```
import android.widget.EditText;
import android.widget.Toast;

import java.io.DataOutputStream;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.InetSocketAddress;
import java.net.Socket;

public class MainActivity extends AppCompatActivity {
    public EditText ip, port, tx_msg, rx_msg;
    public Button connect_btn, disconnect_btn, send_btn;
    public String msg, recv;
    public Socket socket;

    private Handler mHandler = new Handler();
    private DataOutputStream writeSocket;
    private InputStream input;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        StrictMode.ThreadPolicy policy=new StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);


        //ip, port, tx_msg, rx_msg를 각각의 EditText와 연결
        ip = (EditText)findViewById(R.id.input_ip);
        port = (EditText)findViewById(R.id.input_port);
        tx_msg = (EditText)findViewById(R.id.input_msg);
        rx_msg = (EditText)findViewById(R.id.recv_msg);

        //connect_btn, disconnect_btn, send_btn을 각각의 Button과 연결
        connect_btn = (Button)findViewById(R.id.connect_button);
        disconnect_btn = (Button)findViewById(R.id.disconnect_button);
        send_btn = (Button)findViewById(R.id.send_button);

        //connect_btn 버튼 클릭 이벤트 설정
        connect_btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                (new Connect()).start();
            }
        });
        disconnect_btn.setOnClickListener(new View.OnClickListener() {
            @Override
```

```
            public void onClick(View view) {
                (new Disconnect()).start();
            }
        });

        /*
         * 단순히 SEND 버튼 클릭시, input_msg에 들어있는 글자를 송신하는 예제이다.
         * 리모콘 제작시, 버튼을 여러개(방향키, 등)을 추가하여
         * 각각의 버튼마다 어떤 메세지를 보낼 것인지 프로토콜을 정하면 된다.
         * 예를 들어, 위쪽버튼, 아래쪽버튼, 왼쪽버튼, 오른쪽 버튼이 있다고 할 때,
         * 위쪽버튼 클릭시 -> u 전송
         * 아래쪽버튼 클릭시 -> d 전송
         * 왼쪽버튼 클릭시 -> l 전송
         * 오른쪽버튼 클릭시 -> r 전송
         * 이라는 프로토콜을 정하여, MCU에서도 u를 입력 받았을 때 직진,
         * d를 입력받았을 때 후진,
         * l을 입력받았을 때 좌회전
         * r을 입력받았을 때 우회전
         * 등 프로토콜을 맞춰주면 된다.
         * */
        send_btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view){
                (new sendMessage()).start();
            }
        });
    }
    //소켓 연결 클래스
    class Connect extends Thread {
        public void run() {

            String ip_addr = null;
            int port_num = 0;
            //ip와 port값이 형식에 맞게 입력됐는지 확인
            try {
                ip_addr = ip.getText().toString();
                port_num = Integer.parseInt(port.getText().toString());
            } catch (Exception e) {
                final String recvInput = "정확히 입력하세요!";
                mHandler.post(new Runnable() {
                    @Override
                    public void run() {
                        // TODO Auto-generated method stub
                        setToast(recvInput);
                    }
                });
            }
            //해당 ip와 port값에 socket 연결
```

```
            try {
                socket = new Socket(ip_addr, port_num);
                writeSocket = new DataOutputStream(socket.getOutputStream());
                input=socket.getInputStream();
                mHandler.post(new Runnable() {
                    @Override
                    public void run() {
                        // TODO Auto-generated method stub
                        setToast("연결에 성공하였습니다.");
                        mCheckRecv.start();
                    }
                });
            } catch (Exception e) {
                final String recvInput = "연결에 실패하였습니다.";
                Log.d("Connect", e.getMessage());
                mHandler.post(new Runnable() {
                    @Override
                    public void run() {
                        // TODO Auto-generated method stub
                        setToast(recvInput);
                    }
                });
            }
        }
    }

    //소켓 연결 해제 클래스
    class Disconnect extends Thread {
        public void run() {
            try {
                if (socket.isConnected()) {
                    socket.close();
                    mHandler.post(new Runnable() {
                        @Override
                        public void run() {
                            // TODO Auto-generated method stub
                            setToast("연결이 종료되었습니다.");
                        }
                    });
                }
            } catch (Exception e) {
                final String recvInput = "연결을 끊는데 실패했습니다.";
                mHandler.post(new Runnable() {
                    @Override
                    public void run() {
                        // TODO Auto-generated method stub
                        setToast(recvInput);
                    }
```

```
                });
            }
        }
    }

    //송신 쓰레드
    class sendMessage extends Thread {
        public void run() {
            try {
                byte[] msg = new byte[10240];
                msg = tx_msg.getText().toString().getBytes();
                writeSocket.write(msg,0, msg.length);
                mHandler.post(new Runnable() {
                    @Override
                    public void run() {
                        // TODO Auto-generated method stub
                        setToast("메세지 전송 성공");
                    }
                });
            } catch (Exception e) {
                final String recvInput = "메시지 전송에 실패하였습니다.";
                Log.d("Message", e.getMessage());
                mHandler.post(new Runnable() {
                    @Override
                    public void run() {
                        // TODO Auto-generated method stub
                        setToast(recvInput);
                    }
                });
            }
        }
    }

    //수신 핸들러
    @SuppressLint("HandlerLeak")
    private Handler mReceiver = new Handler() {
        public void handleMessage(Message msg) {
            //이 어플이 데이터를 수신했을 시, 어떤 동작을 할 지 작성하면 됨
            //이 예제에서는 단순히 받은 데이터를 표시하게 되어있음
            setToast("메세지 수신 성공");
            rx_msg.setText(recv);
        }
    };
    //수신된 메세지가 있는지 확인하고, 있다면 수신 핸들러를 동작시킴.
    private Thread mCheckRecv = new Thread(){
        public void run(){
            int size;
            byte[] words=new byte[10240];
```

```
            try{
                while(true){
                    size=input.read(words);
                    if(size<=0)
                        continue;
                    recv=new String(words,0,size,"utf-8");
                    mReceiver.sendEmptyMessage(0);
                }
            }catch(Exception e){
                Log.d("tag", "Receive error");
            }
        }
    };

    //알림 메세지 띄우기 함수
    void setToast(String msg) {
        Toast.makeText(this, msg, Toast.LENGTH_SHORT).show();
    }
}
```

7.작성 후 빌드하면, 잘 동작하는 것을 볼 수 있다.
물론 IP 주소와 PORT 번호는 각자 서버용 디바이스에 맞게 세팅해준다.