

Android App

- MainTCPActivity.java

```
package ca.nbcc.rezkbouras.tcppractice;

import android.content.Context;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.AsyncTask;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.SeekBar;
import android.widget.Toast;

public class MainTCPActivity extends ActionBarActivity {

    /**
     * Declarations
     */
    // set the ip address.
    private String roverIP = "192.168.10.102";
    // set the port.
    private int roverPort = 1994;
    private Button btnConnect;
    private Button btnForward;
    private Button btnRight;
    private Button btnLeft;
    private Button btnBackward;
    private Button btnDisconnect;
```

```

private Button btnStop;
private EditText txtReplay;
private EditText txtIP;
private SeekBar speedChanger;
// set command to send.
private String commandToSend;
// Declare a tcp client.
TCPClient client;
// make a thread of maximum limit 10000 threads.
Thread[] threads = new Thread[10000];
// thread counter.
int threadsCounter = 0;

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

    btnConnect = (Button) findViewById(R.id.btnConnect);
    btnForward = (Button) findViewById(R.id.btnForward);
    btnRight = (Button) findViewById(R.id.btnRight);
    btnLeft = (Button) findViewById(R.id.btnLeft);
    btnBackward = (Button) findViewById(R.id.btnBackward);
    btnDisconnect = (Button) findViewById(R.id.btnDisconnect);
    btnStop = (Button) findViewById(R.id.btnStop);
    txtReplay = (EditText) findViewById(R.id.txtReplay);
    txtIP = (EditText) findViewById(R.id.txtIP);
    speedChanger = (SeekBar) findViewById(R.id.speedChanger);

    btnConnect.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            roverIP = txtIP.getText().toString();

            // are we connected?
            if(!client.isConnected) {
                // make a new thread for connecting.
                threads[threadsCounter] = new Thread(new ClientThread());
                // start thread we created.
                threads[threadsCounter].start();
            }
        }
    });
}

```

```
        // increase counter.
        threadsCounter++;

        /* if(client.isConnected){

            v.setBackgroundResource(R.drawable.connectivityyover);
        }*/

    }
}
});

btnDisconnect.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        try {

            Thread.enumerate(threads);
            for (Thread t : threads) {
                if (t.isAlive()) {
                    t.stop();
                }
            }
        } catch (Exception ex){

            ex.printStackTrace();
        }
    }
});

btnForward.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        commandToSend ="w";
        // new MyTask().execute();
        client.send(commandToSend);
    }
});
```

```
btnRight.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        commandToSend = "d";
        //new MyTask().execute();
        client.send(commandToSend);
    }
});
```

```
btnLeft.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        commandToSend = "a";
        new MyTask().execute();
    }
});
```

```
btnBackward.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        commandToSend = "s";
        new MyTask().execute();
    }
});
```

```
btnStop.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        commandToSend = "e";
        new MyTask().execute();
    }
});
```

```
speedChanger.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener()
{
    @Override
    public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
```

```

    }

    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {

    }

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {

        int progress = seekBar.getProgress();
        Toast.makeText(getApplicationContext(), "Speed mode is set to " + progress + "
now.", Toast.LENGTH_SHORT).show();
        commandToSend = String.valueOf(progress);
        new MyTask().execute();
    }
});

}
class MyTask extends AsyncTask<Void, Void, Void>{

    @Override
    protected Void doInBackground(Void...params) {

        // set the user command and send it to server side.
        client.send(commandToSend);

        /* if(received != null) {
            txtReplay.append(received);
        } */

        return null;
    }
}

/**
 * Runnable thread (Multi Thread) that will run for us to initiate the connection, if there
is no connection thread will crash but the app will not.
 */

```

```

class ClientThread implements Runnable {

    boolean isWorking = true;

    @Override
    public void run() {

        ConnectivityManager connectivityManager = (ConnectivityManager)
getApplicationContext().getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo networkInfo          = connectivityManager.getActiveNetworkInfo();

        if (networkInfo != null && networkInfo.isConnected() && networkInfo.isAvailable())
        {
            // Set a new instance of TCPClient class.
            client = new TCPClient(roverIP, roverPort);
            // Initiate connection session.
            client.initiateConnection();
        }
        else
        {
            final int SHORT_DELAY = 2000; // 2 seconds
            // PROMPT USER THAT NETWORK IS DISCONNECT
            Toast.makeText(MainTCPActivity.this, "There is no active network connection!",
Toast.LENGTH_SHORT).show();
            //Toast.makeText("There is no active network connection!").show();

        }

    }

}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu_main_tc, menu);
    return true;
}

```

```
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }

    return super.onOptionsItemSelected(item);
}

}
```

● TCPClient.java

```
package ca.nbcc.rezkbouras.tcppractice;

import android.content.Intent;
import android.util.Log;
import android.widget.Toast;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.io.PrintWriter;
import java.net.DatagramSocket;
```

```
import java.net.InetAddress;
import java.net.InetSocketAddress;
import java.net.MulticastSocket;
import java.net.Socket;
import java.net.SocketAddress;
import java.net.UnknownHostException;

/**
 * Coded by Rezk Bouras on 19/03/2015.
 */
public class TCPClient {

    private static String IP = "";
    private static int PORT = 0;
    //private MulticastSocket connectivity_socket;
    private Socket connectivity_socket;
    private boolean stayWithOpenConnection = true;
    public static boolean isConnected = false;
    private PrintWriter out = null;

    public TCPClient(String IP, int port) {

        // Set the IP Address
        this.setIP(IP);
        // Set the Port
        this.setPORT(port);
    }

    public boolean initiateConnection() {

        try {
            // SocketAddress socket_addr;
            SocketAddress socket_addr = new InetSocketAddress(IP, PORT);
            connectivity_socket = new Socket();
            // Set keep alive feature.
            connectivity_socket.setKeepAlive(stayWithOpenConnection);
            // open the socket to open a connection to the rover socket
            connectivity_socket.connect(socket_addr, 3000);
            isConnected = true;
        }
    }
}
```



```

    } catch (UnknownHostException e) {
        // System.err.println("Don't know about host " + IP);
        System.exit(1);
    } catch (IOException e) {
        // System.err.println("Couldn't get connection I/O for " + IP);
        e.getMessage();
        System.exit(1);
    }

    return connectivity_socket.isConnected();
}

/**
 * Function that will send command to the server side.
 * @param command
 */
public void send(String command) {

    // are we connected?
    if (connectivity_socket.isConnected()) {

        try {
            // Declaring a new command sender that will send the command for us read the
            user inputs, convert it to bytes, and send it.
            out = new PrintWriter(new BufferedWriter(new
            OutputStreamWriter(connectivity_socket.getOutputStream()),true);
            // Send the command given by the user.
            out.println(command);

        } catch (IOException ex) {
            // show debug message
            Log.d("TAG", ex.getMessage());
        }
    }

}
}

```

```
public static String getIP() {
    return IP;
}

public static void setIP(String IP) {
    TCPClient.IP = IP;
}

public static int getPORT() {
    return PORT;
}

public static void setPORT(int PORT) {
    TCPClient.PORT = PORT;
}
}
```

- **Activity_main_tcp.xml (Layout code)**

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context=".MainTCPActivity">

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="textMultiLine"
        android:ems="10"
```

```
android:id="@+id/txtReplay"  
android:layout_alignParentRight="true"  
android:layout_alignParentEnd="true"  
android:enabled="false"  
android:textColor="#ff72004c"  
android:layout_alignParentLeft="true"  
android:layout_alignParentStart="true" />
```

```
<Button  
  android:layout_width="wrap_content"  
  android:layout_height="wrap_content"  
  android:text="Connect"  
  android:id="@+id/btnConnect"  
  android:layout_alignParentBottom="true"  
  android:layout_alignParentLeft="true"  
  android:layout_alignParentStart="true" />
```

```
<Button  
  android:layout_width="wrap_content"  
  android:layout_height="wrap_content"  
  android:text="Disconnect"  
  android:id="@+id/btnDisconnect"  
  android:layout_alignParentBottom="true"  
  android:layout_toRightOf="@+id/btnConnect"  
  android:layout_toEndOf="@+id/btnConnect" />
```

```
<Button  
  android:layout_width="wrap_content"  
  android:layout_height="wrap_content"  
  android:text="Advance"  
  android:id="@+id/btnForward"  
  android:layout_below="@+id/txtReplay"  
  android:layout_centerHorizontal="true" />
```

```
<Button  
  android:layout_width="wrap_content"  
  android:layout_height="wrap_content"  
  android:text="Right"  
  android:id="@+id/btnRight"  
  android:layout_below="@+id/btnForward"  
  android:layout_alignRight="@+id/txtReplay"
```

```
android:layout_alignEnd="@+id/txtReplay"  
android:layout_marginTop="53dp" />
```

```
<Button  
  android:layout_width="wrap_content"  
  android:layout_height="wrap_content"  
  android:text="Left"  
  android:id="@+id/btnLeft"  
  android:layout_alignTop="@+id/btnRight"  
  android:layout_toLeftOf="@+id/btnDisconnect"  
  android:layout_toStartOf="@+id/btnDisconnect" />
```

```
<Button  
  style="?android:attr/buttonStyleSmall"  
  android:layout_width="wrap_content"  
  android:layout_height="wrap_content"  
  android:text="Backward"  
  android:id="@+id/btnBackward"  
  android:layout_centerVertical="true"  
  android:layout_alignRight="@+id/btnStop"  
  android:layout_alignEnd="@+id/btnStop" />
```

```
<EditText  
  android:layout_width="wrap_content"  
  android:layout_height="wrap_content"  
  android:id="@+id/txtIP"  
  android:layout_above="@+id/btnConnect"  
  android:layout_alignParentLeft="true"  
  android:layout_alignParentStart="true"  
  android:layout_alignRight="@+id/btnDisconnect"  
  android:layout_alignEnd="@+id/btnDisconnect"  
  android:text="192.168.10.100" />
```

```
<SeekBar  
  android:layout_width="wrap_content"  
  android:layout_height="wrap_content"  
  android:id="@+id/speedChanger"  
  android:layout_below="@+id/btnBackward"  
  android:layout_marginTop="59dp"  
  android:layout_alignRight="@+id/btnRight"  
  android:layout_alignEnd="@+id/btnRight"
```

```
android:max="9"
android:indeterminate="false"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true" />
```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Stop"
    android:id="@+id/btnStop"
    android:layout_alignTop="@+id/btnLeft"
    android:layout_alignRight="@+id/btnForward"
    android:layout_alignEnd="@+id/btnForward" />
```

```
</RelativeLayout>
```

- **AndroidManifest.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="ca.nbcc.rezkbouras.tcppractice"
    android:installLocation="preferExternal">
```

```
/*
*****
*****\
//
// What you need to add
//
*****
*****\
```

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

```
/*
*****
*****\
//
// What you need to add
//
*****
*****\
```

```
<application
  android:allowBackup="true"
  android:icon="@drawable/ic_launcher"
  android:label="@string/app_name"
  android:theme="@style/AppTheme" >
  <activity
    android:name=".MainTCPActivity"
    android:label="@string/app_name" >
    <intent-filter>
      <action android:name="android.intent.action.MAIN" />

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