Bluetooth Source Code

- ✓ This Application work with command Mode only.
- ✓ It will support all Essae Bluetooth weighing machine.

Code:

```
package com.example.bluetoothcmd;


import android.app.ActivityManager;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.android.material.snackbar.Snackbar;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import android.os.Handler;
import android.os.Looper;
import android.os.Message;
import android.os.StrictMode;
import android.text.method.ScrollingMovementMethod;
import android.util.Log;
import android.view.KeyEvent;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.UnsupportedEncodingException;
import java.lang.reflect.Method;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import java.util.Set;
import java.util.UUID;
import static java.lang.System.exit;
import static java.lang.System.out;
import static java.lang.Thread.currentThread;
import static java.lang.Thread.sleep;
public class MainActivity extends AppCompatActivity {

    private final static int REQUEST_ENABLE_BT = 1; // used to identify adding bluetooth names
    public final static int MESSAGE_READ = 2; // used in bluetooth handler to identify message update
    private final static int CONNECTING_STATUS = 3; // used in bluetooth handler to identify message status
    private static final UUID BT_MODULE_UUID = UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");
    private final String TAG = MainActivity.class.getSimpleName();
ImageButton Conbutton;
String[] DevicenameArray=new String[30];
    private static UUID BTMODULEUUID = null;

    private BluetoothAdapter btAdapter = null;
    private BluetoothDevice mmDevice=null;
    private BluetoothSocket mmSocket=null;
    public Handler handler;
private String SelectedMAC=null;
    private Button BTConnect,BTDisconnect,BtStreamMode;
    BluetoothSocket BTSocket = null;
    BluetoothDevice device12;
    TextView ttresult;

    private Handler mHandler; // Our main handler that will receive callback notifications
    private ConnectedThread mConnectedThread; // bluetooth background worker thread to send and receive data
    private BluetoothSocket mBTSocket = null; // bi-directional client-to-client data path
    String selectedDevice = "";
    public MainActivity()
    {
        UUID BTMODULEUUID12 = UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");
        BTMODULEUUID=BTMODULEUUID12;
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // Toolbar toolbar = findViewById(R.id.toolbar);
        // setSupportActionBar(toolbar);
        int DeviceCount = 0;
        final List<String> BDAddresslist = new ArrayList();
        List<String> BDlist = new ArrayList();

        final BluetoothAdapter mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
        BtStreamMode = (Button) findViewById(R.id.btnStream);
        BtStreamMode.setEnabled(false);
        BtStreamMode.setVisibility(View.GONE);
        if (mBluetoothAdapter.isEnabled()) {
            // Bluetooth is enabled

            if (mBluetoothAdapter.getState() == BluetoothAdapter.STATE_ON) {
                Set<BluetoothDevice> pairedDevices = mBluetoothAdapter.getBondedDevices();
            if (pairedDevices.size() > 0) {
                for (BluetoothDevice device : pairedDevices) {
                    String devicename = device.getName();
                    String DeviceAddress = device.getAddress();
                    BDlist.add(devicename);
                    BDAddresslist.add(DeviceAddress);
                    DeviceCount = DeviceCount + 1;
                }

                DevicenameArray = new String[BDlist.size()];
                DevicenameArray = BDlist.toArray(DevicenameArray);
            }
        }
        ttresult = (TextView) findViewById(R.id.txtresult);

        mHandler = new Handler(Looper.getMainLooper()) {
            @Override
            public void handleMessage(Message msg) {
                if (msg.what == MESSAGE_READ) {
                    String readMessage = null;
                    try {
                        readMessage = new String((byte[]) msg.obj, "UTF-8");
                    } catch (UnsupportedEncodingException e) {
                        e.printStackTrace();
                    }
                    // mReadBuffer.setText(readMessage);
                }

                if (msg.what == CONNECTING_STATUS) {
                    // if(msg.arg1 == 1)
                    // mBluetoothStatus.setText("Connected to Device: " + msg.obj);
                    // else
                // mBluetoothStatus.setText("Connection Failed");
            }
        }
    };
    Conbutton = (ImageButton) findViewById(R.id.BluetoothimageButton);
    Conbutton.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(final View view) {
            AlertDialog.Builder alertdialogbuilder = new AlertDialog.Builder(MainActivity.this);
            alertdialogbuilder.setTitle("Select A Device ");
            alertdialogbuilder.setItems(DevicenameArray, new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialogInterface, int which) {
                    selectedDevice = Arrays.asList(DevicenameArray).get(which);
                    SelectedMAC = BluetoothDeviceTOMacfind(selectedDevice);

                    if (SelectedMAC != null) {
                        btAdapter = BluetoothAdapter.getDefaultAdapter();
                        device12 = btAdapter.getRemoteDevice(SelectedMAC);
                        mmDevice = device12;
                        new Thread() {
                            @Override
                            public void run() {
                                boolean fail = false;
                                BluetoothDevice device = btAdapter.getRemoteDevice(SelectedMAC);
                                try {
                                    mBTSocket = createBluetoothSocket(device);
                                } catch (IOException e) {
                                    fail = true;
                                    Toast.makeText(getBaseContext(), "Socket creation failed", Toast.LENGTH_SHORT).show();
                                }
                                // Establish the Bluetooth socket connection.
                                try {
                                    mBTSocket.connect();
                                } catch (IOException e) {
                                    try {
                                        fail = true;
                                        mBTSocket.close();
                                        mHandler.obtainMessage(CONNECTING_STATUS, -1, -1)
                                                .sendToTarget();
                                    } catch (IOException e2) {
                                        //insert code to deal with this
                                        Toast.makeText(getBaseContext(), "Socket creation failed", Toast.LENGTH_SHORT).show();
                                    }
                                    if (!fail) {
                                        mConnectedThread = new ConnectedThread(mBTSocket, mHandler);
                                        mConnectedThread.start();
                                        mHandler.obtainMessage(CONNECTING_STATUS, 1, -1, selectedDevice)
                                                .sendToTarget();
                                    }
                                }
                            }
                        }.start();
                    }
                }
            });
            AlertDialog dialog = alertdialogbuilder.create();
            dialog.show();
        }
    });

    BTConnect = (Button) findViewById(R.id.btnCommand);
    BTConnect.setOnClickListener(new View.OnClickListener() {
        OutputStream out;
        char[] ScaleCommand = new char[]{'\u0005'};
```

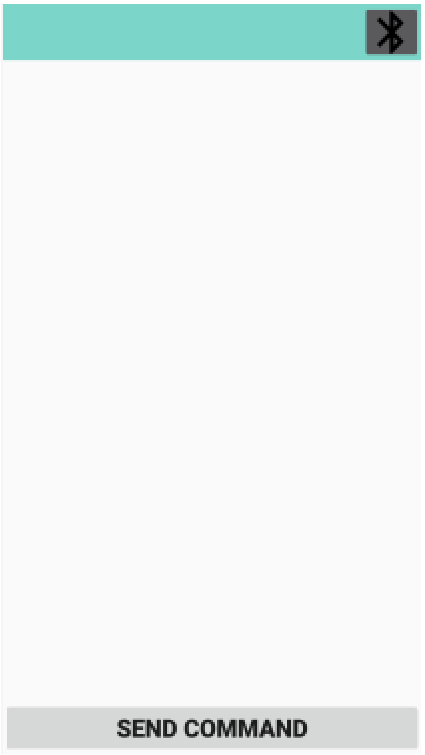
```
                //byte[] buffer = new byte[10];
                int red = 0;
                String redDataText = null;
                String Result = "No Data \n";
                @Override
                public void onClick(final View view) {
                    if(mConnectedThread != null) //First check to make sure thread created
                        mConnectedThread.write("1");
                    ttresult.append(Result);
                    ttresult.setMovementMethod(new ScrollingMovementMethod());
                }
            });
        }
        else
        {
            Toast.makeText(MainActivity.this, "Please Enable Bluetooth",
                    Toast.LENGTH_LONG).show();
        }
    }

    private BluetoothSocket createBluetoothSocket(BluetoothDevice device) throws IOException {
        try {
            final Method m = device.getClass().getMethod("createInsecureRfcommSocketToServiceRecord", UUID.class);
            return (BluetoothSocket) m.invoke(device, BT_MODULE_UUID);
        } catch (Exception e) {
            Log.e(TAG, "Could not create Insecure RFComm Connection",e);
        }
        return  device.createRfcommSocketToServiceRecord(BT_MODULE_UUID);
    }
    @Override
    public void onBackPressed(){
/*          Intent intent = new Intent(Intent.ACTION_MAIN);
        intent.addCategory(Intent.CATEGORY_HOME);
    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    startActivity(intent);
    finish();*/
    Intent intent = new Intent(Intent.ACTION_MAIN);
    intent.addCategory(Intent.CATEGORY_HOME);
    intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);//***Change Here***
    startActivity(intent);
    finish();
    System.exit(0);
}

@Override
protected void onPause() {
    super.onPause();
    ActivityManager activityManager = (ActivityManager) getApplicationContext()
            .getSystemService(Context.ACTIVITY_SERVICE);
    activityManager.moveTaskToFront(getTaskId(), 0);
}
private String BluetoothDeviceTOMacfind(String selectedDevice) {
    String DeviceAddress=null;
    BluetoothAdapter mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
    if (mBluetoothAdapter == null) {
        // Device does not support Bluetooth
        Toast.makeText(getApplicationContext(),"Bluetooth Connection Failed", Toast.LENGTH_LONG).show();
        Log.e("Bluetooth ","not found");
    }
    if (mBluetoothAdapter.isEnabled()) {
        Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
        startActivity(enableBtIntent);
        Set<BluetoothDevice> pairedDevices = mBluetoothAdapter.getBondedDevices();
        if (pairedDevices.size() > 0) {
            for (BluetoothDevice device : pairedDevices) {
                String devicename = device.getName();
```

```
                    if ( devicename.equals(selectedDevice) )
                    {
                        DeviceAddress = device.getAddress();
                    }
                    //BluetoothDevice device = mBluetoothAdapter.getRemoteDevice(address);
                }
            }
        }
        return DeviceAddress;
    }
    @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, 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);
    }
}
```

Sample Screen:


SEND COMMAND