ORA SDK Library
V1.2.1 – Connectivity

Summary

1 Introduction

The ORA-1 is a best in class hands free wireless see-through wearable display platform in the form of digital eyewear allowing hands-free mobile computing applications. It is based on Optinvent's patented Clear-Vu display engine. ORA-1 will enable many yet unimagined "always-on" hands free mobile applications including geo-localization (GPS), sports, messaging, situation awareness, and more. It can be connected via a standard wi-fi connection to a smartphone or tablet and will act as a hands-free wearable computer.

Optinvent's ORA-1 is the most viable product to achieve widespread industry and consumer acceptance for AR glasses based on its bright, see-through, large field of view display with true see-through capability. Furthermore, the display has a patented "Flip-Vu" feature which can position the virtual image in full augmented reality mode (directly in the user's line of sight), and a dashboard mode (below the user's line of sight). Along with the patented display engine, the ORA-1 includes a WiFi and Bluetooth connectivity, front facing camera, 9 axis motion sensor, ambient light sensor, microphone, loudspeaker, and a high capacity rechargeable battery.

Optinvent's ORA-1 is a standalone Android based device.

The present documentation will provide you a software approach that would allow you to use ORA SDK.

The described SDK features rely on connectivity stack: bluetooth and GPS.

Other features will be taken into account in future SDK versions, such as Camera, Sensors, ... But can already be reached through Android native SDK.

1.1 Target Audience

This documentation is mainly dedicated to Android application developers, and will help them to develop applications for ORA devices.

Required Acknowledgement :

– Java

– Android System 4.2.2 (minimum)

– Eclipse IDE v4.2.1 (minimum) + ADT + NDK

Optinvent France
Bâtiment Le Gallium, 80 avenue des Buttes de Coësmes
35700 Rennes, France

Optinvent Inc. USA
16133 Hillvale Ave
Monte Sereno, CA 95030-4112

e-mail: support@optinvent.com

2 ORA SDK V1.2.1

ORA Connectivity SDK gives you the opportunity to connect whatever bluetooth device to your ORA Optinvent glasses.

The SDK offers you the suitable tools to send/receive multiple data type such as event notifications through bluetooth.

Sample code illustrate the way to use both Android SDK and ORA SDK, on Client side (smartphone) and server side (ORA glasses), by providing remote touchscreen and keyboard.

Optinvent Connectivity SDK enable :

- – Bluetooth devices Discovery
- – Bluetooth device Configuration as Bluetooth Server
- – Bluetooth device Configuration as Bluetooth Client
- – Transferring datas between devices (Send / Receive bluetooth messages).
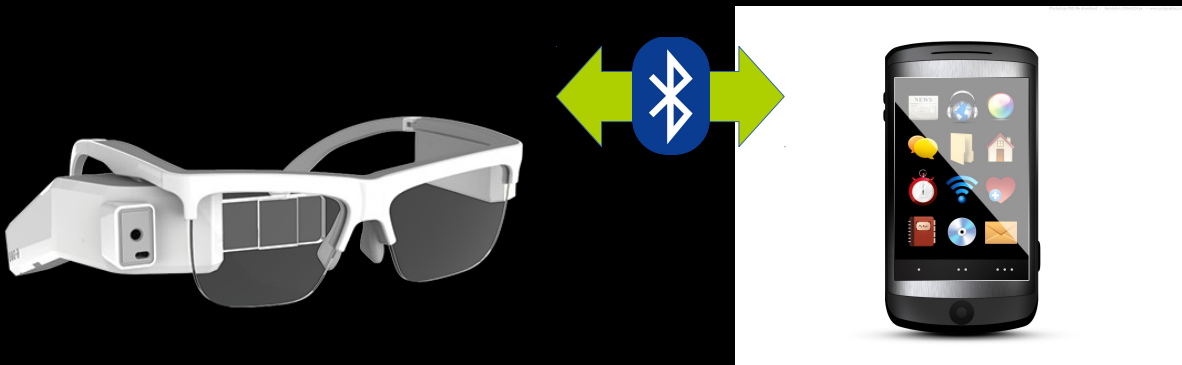
2.1 Getting started

The sdk contains four eclipse projects in three archive files (.zip).

Import them in your eclipse workspace.

The optinvent-sdk project is the actual sdk. The current version only contains bluetooth facilities.

The ora-gps project is a tool to allow ORA Glasses to get GPS data from another Device (See 2.4 Note about the GPS).

The two other projects are sample code, demonstrating how to turn a usual android device into a bluetooth remote controller for the ORA glasses.

2.2 Run the sample code

Before using the sample, you must pair a bluetooth android device with the ORA glasses.

Then, install and run sample-server on the ORA glasses and sample-client and the other device.

Select the ORA glasses on the device. Now the device can be used as a mouse and a keyboard for the glasses.

2.3 Create your own bluetooth Client/Server application using the sdk

First create two android application projects in eclipse, one for the client and the other for the server.

In the android project properties, add the project library optinvent-sdk-dist (in both projects). See the eclipse documentation if needed : http://help.eclipse.org/luna/index.jsp .

Choose a bluetooth UUID service for your application (see http://www.itu.int/ITU-T/asn1/uuid.html).

Add the following permissions in the manifests files :

```xml
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

The server

In the server project, extend the BTServerService class to create the server in the server project :

```java
import android.bluetooth.BluetoothDevice;

import fr.eurogiciel.ora.bluetooth.BTMessage;
import fr.eurogiciel.ora.bluetooth.BTServerService;

public class MyService extends BTServerService {
    @Override
    public BTMessage onMsgReceived(BluetoothDevice sender, BTMessage msg) {
        /*The server received a request.
         *Do some job and return a response or null if no response is needed
         */
    }

    @Override
    public boolean isAuthorized(BluetoothDevice device) {
        /* return true if you want to authorize the device to talk with the
         * server
         */
    }

    @Override
    public void onDisconnected(BluetoothDevice device) {
        /* This method is called when a client disconnected. */
    }

    @Override
    public void onServerStopped() {
        /* This method is called when the server stopped for any reason. */
    }
}
```

Do not forget to declare the service in the manifest.

Start the service somewhere in your code :

```
Intent intent = new Intent(this, MyService.class);
intent.putExtra(BTServerService.MSG_SERVER_NAME, serverName);
intent.putExtra(BTServerService.MSG_BT_UUID, APP_UUID);
startService(intent);

APP_UUID is your application UUID.
```

Please note that the APP_UUID must be a String and only accepts capital letters.

The client

In the client project, declare your client activity :

```
import java.io.IOException;
import java.util.UUID;

import fr.eurogiciel.ora.bluetooth.BTClient;
import fr.eurogiciel.ora.bluetooth.BTSocketListener;
import fr.eurogiciel.ora.bluetooth.BTMessage;

import android.app.Activity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;


public class MyClientActivity extends Activity implements BTSocketListener.Callback {

    /*Your application UUID */
    private static final UUID APP_UUID = UUID.fromString(MY_APP_UUID);
    private BTClient client;

    @Override
    public void onStart() {
        super.onStart();

        /* Get your device using android bluetooth API, see
http://developer.android.com/guide/topics/connectivity/bluetooth.html */
        BluetoothDevice myServerDevice = ;

        //Connection thread
        new Thread("Client creation") {
            public void run() {
                try {

                    client = new BTClient(MyClientActivity.this,
MyClientActivity.this, myServerDevice, APP_UUID);

                    /* The device has been connected to the server.
                     * You can start sending messages and receive responses in
                     * onMsgReceived().
                     */
                    client.sendMessage(myMessage);
```

```
                        } catch (IOException e) {
                                finish();
                        }
                };
        }.start();
    }

    @Override
    public BTMessage onMsgReceived(BluetoothDevice sender, BTMessage msg) {
            return null; /* The server sent a message. */
    }

    @Override
    public boolean isAuthorized(BluetoothDevice device) {
            return true; /* Is the device authorized to talk to the client. */
    }

    @Override
    public void onDisconnected(BluetoothDevice device) {
            /* The server connection is lost or closed. */
    }
}
```

Do not forget to declare the activity in the manifest.

Now pair the ORA glasses and the other device with the android bluetooth settings menu. Then run the server application on the glasses and the client application on the other device.

You may take a look at the provided sample code to see a complete example.

2.4 Note about the GPS

As the current ORA glasses do not have yet a GPS feature, the sdk provides the ora-gps sample application. It is compatible with the sample code of the server and it allows another android device to send GPS coordinates to the ORA glasses.

To do so, pair the ORA glasses with an android device, install and run the sample-server project on the glasses, then the ora-gps project on the android device.

The future version of ORA will have a GPS so it is temporary solution.

2.5 API References

See the accompanying javadoc directory.

3 Property Rights & Licenses

The ORA Software Development Kit v1.2.1 is provided by Eurogiciel, Copyright (c) EUROGICIEL, 2014. All rights reserved.

This SDK is licensed to Optinvent for use with the product ORA-1.

Using this SDK is subject to agree to all the terms of the ORA SDK License Agreement, See document "ORA SDK EULA".

You may obtain a copy of the License at : *http://optinvent.com/pdf/ora-sdk.pdf*

Unless required by applicable law or agreed to in writing, software and documentation under the License are distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

4 Revision History

Authors:

Xavier Roche – Eurogiciel

Christophe Moreau – Eurogiciel

Thierry Gayet – Eurogiciel

Fabien Seznec - Eurogiciel

Revision History				
Cover Date	**Created by**	**Description**	**Version**
March 26th 2014	Xavier Roche - Eurogiciel	Creation of the document	V1.0
July 15th 2014	Fabien Seznec - Eurogiciel	Updated to version 1.2	V1.2
July 29th 2014	Fabien Seznec - Eurogiciel	Updated to version 1.2.1 Compliance with Android 4.4 + bug fixes	V1.2.1

5 Document identification

Name : ORA_SDK-V1.2.1

Reference : ORA00741907010201

Version : V1.2.1

Optinvent France
Bâtiment Le Gallium, 80 avenue des Buttes de Coësmes
35700 Rennes, France

Optinvent Inc. USA
16133 Hillvale Ave
Monte Sereno, CA 95030-4112

e-mail: support@optinvent.com