

## ***TxRxApp* APP for Android Operating System**

### **1 . APP purpose and functionalities**

The *TxRxApp* APP for Android Operating System has been released by TERTIUM Technology with a “MIT License”, which is an *open source* license, with the specific purpose of simplifying, for its partners and clients, the development of APPs that interact with *TxRx Bluetooth Low Energy* (BLE) devices by TERTIUM Technology. *TxRx* devices by TERTIUM Technology expose a BLE service with two main characteristics – named respectively “Tx” e “Rx” – simulating a bidirectional ASCII *stream*. With this stream the device uses the “Rx” characteristic to receive commands from a central device, usually an Android smartphone or tablet, and uses the “Tx” characteristic to send the corresponding replies.

The APP offers functionalities for showing all the detectable BLE devices, for connecting to one of those devices (if it is a “*TxRx*” one) and a CLI; with this textual interface the users are allowed to write the commands they want to issue to the device and to read the received replies.

The core of the *TxRxApp* APP source code, named “lib”, has been written and documented in order to be reused in each and every APP interacting with *TxRx* BLE devices by TERTIUM Technology; the *activities* implementing the user interface are meant as useful examples of “lib”<sup>1</sup> module functionalities.

---

<sup>1</sup> An APP for a specific application should obviously manage the commands and text replies documented by TERTIUM Technology for the specific *TxRx* BLE device used.

## 2. Architecture of the source code of *TxRxApp* APP

The source code of *TxRxApp* APP is entirely included in the *com.tertiumtechnology.demoapp* main package: at this hierarchical level we can find all the classes implementing the APP user interface and an Android service that creates one *TxRx* BLE device manager and translates every asynchronous event generated by the device manager itself (device replies receptions, *timeout* events, errors, ...) into messages to be shown.

The main package includes the *com.tertiumtechnology.demoapp.util* package which includes utility classes for creating the user interface (connection to the BT/BLE adapter, management of the list of detected BLE devices, storage of configured preferences) and the package *com.tertiumtechnology.txrxlib*: the latter contains - divided into three distinct packages - the reusable classes for the implementation of APPs for Android operating system that interact with TERTIUM Technology *TxRx* BLE devices.

### 3. The *com.tertiumtechnology.txrxlib* package classes functionalities

The *com.tertiumtechnology.txrxlib.scan* package contains the classes and interfaces to be used for detecting BLE devices in the proximity of the smartphone or tablet<sup>2</sup>:

Class/interface	Functionality
<code>TxRxScanner</code>	The class constructor allows to register the BT/BLE adapter and an instance of the <i>TxRxScanCallback</i> interface whose methods are invoked for notification of the detected devices; the <i>startScan</i> and <i>stopScan</i> methods of the class starts and ends respectively the detection of BLE devices (the <i>setScanTimeout</i> method allows defining a time period expressed in milliseconds which determines the implicit invocation of the <i>stopScan</i> method).
<code>TxRxScanCallback</code>	This interface defines the prototype of the methods that must be implemented in a class to be instantiated as an argument of the <i>TxRxScanner</i> class constructor: the <i>onDeviceFound</i> method is invoked for each detected BLE device (the only parameter is an instance of the <i>TxRxScanResult</i> class representing the device detected, together with some parameters connected to the detection), while the <i>afterStopScan</i> method is invoked after the termination - manual or automatic - of the detection of BLE devices.
<code>TxRxScanResult</code>	This class represents a detected BLE device and exposes methods that return the system representation of the device itself - an instance of the <i>BluetoothDevice</i> class - plus the RSSI value of the radio signal strength during the detection and the bytes that constitute the recording of the BLE <i>beacon's</i> content of the detected device.

<sup>2</sup> The Android APIs for detecting BLE devices have been modified after the release of version 5 of the operating system called "Lollipop": the class code manages the new APIs if run on later versions and the original APIs otherwise; versions of Android prior to 4.3 do not allow the management of BLE devices.

The *com.tertiumtechnology.txrxlib.rw* package contains the classes and the interfaces for the connection and the communication with a *TxRx* BLE device:

Class/interface	Functionality
<code>TxRxTimeouts</code>	Class for the representation of the 4 timeout planned values for the management of a <i>TxRx</i> BLE device: <ul style="list-style-type: none"> <li>▪ connection timeout;</li> <li>▪ write timeout on the "Rx" characteristic of the device;</li> <li>▪ initial reading timeout from the "Tx" characteristic of the device;</li> <li>▪ reading timeout after the initial one of the "Tx" characteristic of the device.</li> </ul>
<code>TxRxDeviceProfile</code>	Class for profiling a category <sup>3</sup> of <i>TxRx</i> BLE devices; the constructor allows to define the following parameters: <ul style="list-style-type: none"> <li>▪ UUID of <i>TxRx</i> service;</li> <li>▪ UUID of "Tx" characteristic of the service (16/128 bit);</li> <li>▪ UUID of "Rx" characteristic of the service (16/128 bit);</li> <li>▪ possible terminator character for command strings;</li> <li>▪ possible terminator character for reply strings;</li> <li>▪ maximum length of the "Tx" characteristic of the service;</li> <li>▪ maximum length of the "Rx" characteristic of the service.</li> </ul>
<code>TxRxDeviceManager</code>	Main class for the connection and the communication with <i>TxRx</i> BLE devices; the constructor allows to register the BT/BLE adapter, an instance of the <i>TxRxDeviceCallback</i> interface whose methods are invoked for notification of events and errors and an instance of the <i>TxRxTimeouts</i> class for specifying timeout values. The class exposes, among others, the following methods: <ul style="list-style-type: none"> <li>▪ <i>connect</i>: to connect to a BLE device whose MAC address is specified;</li> <li>▪ <i>disconnect</i>: for disconnecting the connected BLE device;</li> <li>▪ <i>requestWriteData</i>: to start writing a command on the "Rx" characteristic of the connected BLE device (the reception of any reply from the BLE device is managed asynchronously and requires the invocation of a specific method of the <i>TxRxDeviceCallback</i> class instance).</li> </ul>
<code>TxRxDeviceCallback</code>	This interface defines the prototype of the methods that must be implemented in a class which instance is to be provided as an argument of the <i>TxRxManager</i> class constructor; the interface requires implementing, among others, the following methods: <ul style="list-style-type: none"> <li>▪ <i>onConnectionError</i>: invoked in case of connection error;</li> <li>▪ <i>onConnectionTimeout</i>: invoked in case of connection timeout;</li> <li>▪ <i>onDeviceConnected</i>: invoked when connecting the BLE device;</li> <li>▪ <i>onDeviceDisconnected</i>: invoked when the BLE device is disconnected;</li> <li>▪ <i>onTxRxServiceDiscovered</i>: invoked to detect the BLE <i>TxRx</i> service;</li> <li>▪ <i>onTxRxServiceNotFound</i>: invoked when no BLE <i>TxRx</i> service is</li> </ul>

<sup>3</sup> Currently two categories of *TxRx* BLE devices made by TERTIUM Technology are profiled in the APP: the devices based on Zentri's BLE ACKme hardware modules and the devices based on Texas Instruments MCU BLE CC2540.

	<p>detected;</p> <ul style="list-style-type: none"><li>▪ <i>onWriteData</i>: invoked upon confirmation of writing a command on the "Rx" characteristic;</li><li>▪ <i>onWriteError</i>: invoked in case of error in writing a command;</li><li>▪ <i>onWriteTimeout</i>: invoked in case of timeout in writing a command;</li><li>▪ <i>onNotifyData</i>: invoked upon completion of the receipt of a reply read by the "Tx" characteristic;</li><li>▪ <i>onReadNotifyTimeout</i>: invoked in case of timeout in reading a reply.</li></ul>
--	--

#### 4. Communication management algorithm with *TxRx* BLE devices

First of all you have to establish a connection by invoking the *connect* method of the *TxRxDeviceManager* class and receiving positive confirmation by the invocation of the call-back method *onTxRxServiceDiscovered*, which guarantees that the BLE device is of type *TxRx*<sup>4</sup> (in case of connection failure, the call-back methods *onConnectionError* or *onConnectionTimeout* are called, respectively).

Once the connection has been made, the communication between the smartphone or tablet and the *TxRx* BLE device is carried out following the following algorithm:

- the invocation of the *requestWriteData* method of the *TxRxManager* class - whose parameter is the string containing the text command to be sent to the BLE device - writes the "Rx" characteristic of the *TxRx* service. If the string to be written has a length greater than the declared size for the characteristic, then multiple sending of fragments of the string are made, each of length equal or inferior to the dimension of the characteristic<sup>5</sup>;
- the invocation of the call-back method *onWriteData* confirms the correct writing of the command in the "Rx" characteristic; in case of error or timeout the *onWriteError* or *onWriteTimeout* call-back methods are invoked respectively (the timeout length is a configurable parameter);
- with the sending of the last fragment of the command string (coinciding with the first in the case of a string of length equal to or less than the size of the "Rx" characteristic), a timer initialized with the timeout time for the start of the reply is started;
- the reception of the possible reply to the command sent is received by successive BLE notifications of changing of the content of the "Tx" characteristic: the string fragments notified in succession are recomposed in a single response string; if no notification is issued before the timer is reset, the *onReadNotifyTimeout* call-back method is invoked;
- after the reception of every single fragment of the reply string, a timer initialized with the timeout time for the end of the reply is started or restarted: if the timer is reset, the reading of the answer is considered finished and the string that contains is provided as a parameter of the *onNotifyData* call-back method.

---

<sup>4</sup> Calling *onDeviceConnected* notify the successful connection, but does not guarantee that the connected device is of type *TxRx*.

<sup>5</sup> The BLE protocol involves the transmission of packets with a maximum *payload* of 20 bytes: if the length of the "Rx" characteristic is greater than 20, the transmission of the value to be written in the characteristic is divided into packets each with *payload* equal to or less than 20 in a way that is transparent for the Java code that invokes the Android BLE API.

With the exception of starting the timer for the timeout, the code of a command for writing and the code of a command for reading a reply are completely asynchronous: any unsolicited variation of the "Tx" characteristic by the BLE device generates the reception of the string in the same way as receiving a reply to a command.

## 5. Documentation of "lib" module code

The Java code of the classes contained in the *com.tertiumtechnology.txrxlib* package has been commented with the style provided by the JavaDoc standard: the documentation in HTML format of the packages, classes, attributes, methods and parameters has been automatically generated accordingly.