

APP *TxRxApp* per sistema operativo Android

1. Scopo e funzionalità della APP

La APP per sistema operativo Android *TxRxAPP* è rilasciata da TERTIUM Technology con licenza *open source* “MIT License” allo scopo di semplificare la codifica di APP che interagiscono con i dispositivi *Bluetooth Low Energy* (BLE) *TxRx* di TERTIUM Technology da parte dei propri partner e clienti. I dispositivi *TxRx* di TERTIUM Technology espongono un servizio BLE con due caratteristiche – denominate rispettivamente “Tx” e “Rx” – che simulano uno *stream* ASCII bidirezionale con il quale il dispositivo riceve comandi da un dispositivo centrale, normalmente uno smartphone o un tablet con sistema operativo Android, utilizzando la caratteristica “Rx” ed invia le relative risposte utilizzando la caratteristica “Tx”.

La APP, oltre alle funzionalità di visualizzazione dei dispositivi BLE rilevabili e di connessione ad uno dei dispositivi rilevati se di tipo *TxRx*, realizza un’interfaccia testuale che consente all’utente di scrivere i comandi da inviare al dispositivo e di leggere le risposte ricevute e visualizzate.

Il nucleo del codice sorgente della APP *TxRxAPP*, denominato “lib”, è stato scritto e documentato per essere riutilizzato in tutte le APP che interagiscono con i dispositivi BLE *TxRx* di TERTIUM Technology; le *activity* che realizzano l’interfaccia utente sono intese come utile esemplificazione dell’uso delle funzionalità del modulo “lib”¹.

¹una APP per una specifica applicazione dovrà ovviamente gestire i comandi e le risposte testuali documentati da TERTIUM Technology per lo specifico dispositivo BLE *TxRx* utilizzato

2. Architettura del codice sorgente della APP *TxRxApp*

Il codice sorgente della APP *TxRxApp* è interamente compreso nel package principale *com.tertiumtechnology.demoapp*: a questo livello gerarchico sono presenti le classi che realizzano l'interfaccia utente della APP e un servizio Android che crea il gestore di un dispositivo BLE *TxRx* ed effettua la trasformazione in messaggi da visualizzare di tutti gli eventi asincroni generati dal gestore stesso (ricezione di risposte dal dispositivo, eventi di *timeout*, errori, ...).

Il package principale include il package *com.tertiumtechnology.demoapp.util* che comprende classi di utilità per la realizzazione dell'interfaccia utente (collegamento all'adattatore BT/BLE, gestione dell'elenco dei dispositivi BLE rilevati, memorizzazione delle preferenze configurate) e il package *com.tertiumtechnology.txrxlib*: quest'ultimo contiene – suddivise in tre package distinti – le classi riutilizzabili per la realizzazione di APP per sistema operativo Android che interagiscono con i dispositivi BLE *TxRx* di TERTIUM Technology.

3. Funzionalità delle classi del package *com.tertiumtechnology.txrxlib*

Il package *com.tertiumtechnology.txrxlib.scan* contiene le classi e le interfacce da utilizzare per la rilevazione dei dispositivi BLE presenti nelle vicinanze dello smartphone o del tablet²:

Classe/interfaccia	Funzionalità
<code>TxRxScanner</code>	Il costruttore della classe permette di registrare l'adattatore BT/BLE e un'istanza dell'interfaccia <i>TxRxScanCallback</i> i cui metodi sono invocati per la notifica dei dispositivi rilevati; i metodi <i>startScan</i> e <i>stopScan</i> della classe iniziano e terminano rispettivamente la rilevazione dei dispositivi BLE (il metodo <i>setScanTimeout</i> permette di definire un periodo temporale espresso in millisecondi che determina l'invocazione implicita del metodo <i>stopScan</i>)
<code>TxRxScanCallback</code>	Questa interfaccia definisce il prototipo dei metodi da implementare in una classe di cui fornire un'istanza come argomento del costruttore della classe <i>TxRxScanner</i> : il metodo <i>onDeviceFound</i> viene invocato per ogni dispositivo BLE rilevato (l'unico parametro è un'istanza della classe <i>TxRxScanResult</i> che rappresenta il dispositivo rilevato contestualmente ad alcuni parametri collegati alla rilevazione), mentre il metodo <i>afterStopScan</i> viene invocato dopo la terminazione – manuale o automatica – della rilevazione dei dispositivi BLE
<code>TxRxScanResult</code>	Questa classe rappresenta un dispositivo BLE rilevato ed espone metodi che restituiscono la rappresentazione di sistema del dispositivo stesso – un'istanza della classe <i>BluetoothDevice</i> – più il valore RSSI dell'intensità del segnale radio durante la rilevazione e i byte che costituiscono la registrazione del contenuto del <i>beacon</i> BLE del dispositivo rilevato

²le API Android per la rilevazione di dispositivi BLE sono state modificate successivamente al rilascio della versione 5 del sistema operativo denominata "Lollipop": il codice delle classi gestisce le nuove API se eseguito su versioni successive, altrimenti gestisce le API originali; le versioni di Android precedenti alla 4.3 non consentono la gestione dei dispositivi BLE

Il package `com.tertiumtechnology.txrxlib.rw` contiene le classi e le interfacce per la connessione e la comunicazione con un dispositivo BLE *TxRx*:

Classe/interfaccia	Funzionalità
<code>TxRxTimeouts</code>	<p>Classe per la rappresentazione dei 4 valori di timeout previsti per la gestione di un dispositivo BLE <i>TxRx</i>:</p> <ul style="list-style-type: none"> ▪ timeout di connessione ▪ timeout di scrittura sulla caratteristica “Rx” del dispositivo ▪ timeout di lettura iniziale dalla caratteristica “Tx” del dispositivo ▪ timeout di lettura successiva a quella iniziale della caratteristica “Tx” del dispositivo
<code>TxRxDeviceProfile</code>	<p>Classe per la profilazione di una categoria³ di dispositivi BLE <i>TxRx</i>; il costruttore permette di definire i seguenti parametri:</p> <ul style="list-style-type: none"> ▪ UUID del servizio <i>TxRx</i> ▪ UUID della caratteristica “Tx” del servizio (16/128 bit) ▪ UUID della caratteristica “Rx” del servizio (16/128 bit) ▪ eventuale carattere terminatore per le stringhe di comando ▪ eventuale carattere terminatore per le stringhe di risposta ▪ lunghezza massima della caratteristica “Tx” del servizio ▪ lunghezza massima della caratteristica “Rx” del servizio
<code>TxRxDeviceManager</code>	<p>Classe principale per la connessione e la comunicazione con i dispositivi BLE <i>TxRx</i>; il costruttore permette di registrare l’adattatore BT/BLE, un’istanza dell’interfaccia <i>TxRxDeviceCallback</i> i cui metodi sono invocati per la notifica degli eventi e degli errori e un’istanza della classe <i>TxRxTimeouts</i> per la specifica dei valori di timeout. La classe espone, tra gli altri, i seguenti metodi:</p> <ul style="list-style-type: none"> ▪ <i>connect</i> per la connessione ad un dispositivo BLE di cui viene specificato l’indirizzo MAC ▪ <i>disconnect</i> per la disconnessione del dispositivo BLE connesso ▪ <i>requestWriteData</i> per l’avvio della scrittura di un comando sulla caratteristica “Rx” del dispositivo BLE connesso (la ricezione dell’eventuale risposta da parte del dispositivo BLE viene effettuata in modo asincrono e comporta l’invocazione di uno specifico metodo dell’istanza della classe <i>TxRxDeviceCallback</i>)

³attualmente sono profilate nella APP due categorie di dispositivi BLE *TxRx* realizzati da TERTIUM Technology: la categoria di dispositivi basati su moduli hardware BLE ACKme di Zentri e la categoria di dispositivi basati sulla MCU BLE CC2540 di Texas Instruments

<p>TxRxDeviceCallback</p>	<p>Questa interfaccia definisce il prototipo dei metodi da implementare in una classe di cui fornire un'istanza come argomento del costruttore della classe <i>TxRxManager</i>; l'interfaccia impone di implementare tra gli altri i seguenti metodi:</p> <ul style="list-style-type: none"> ▪ <i>onConnectionError</i> invocato in caso di errore di connessione ▪ <i>onConnectionTimeout</i> invocato in caso di timeout di connessione ▪ <i>onDeviceConnected</i> invocato al momento della connessione del dispositivo BLE ▪ <i>onDeviceDisconnected</i> invocato al momento della disconnessione del dispositivo BLE ▪ <i>onTxRxServiceDiscovered</i> invocato alla rilevazione del servizio BLE <i>TxRx</i> ▪ <i>onTxRxServiceNotFound</i> invocato nel caso nessun servizio BLE <i>TxRx</i> sia rilevato ▪ <i>onWriteData</i> invocato al momento della conferma di scrittura di un comando sulla caratteristica "Rx" ▪ <i>onWriteError</i> invocato in caso di errore nella scrittura di un comando ▪ <i>onWriteTimeout</i> invocato in caso di timeout nella scrittura di un comando ▪ <i>onNotifyData</i> invocato al completamento della ricezione di una risposta letta dalla caratteristica "Tx" ▪ <i>onReadNotifyTimeout</i> invocato in caso di timeout nella lettura di una risposta
---------------------------	---

4. Algoritmo di gestione della comunicazione con i dispositivi BLE TxRx

Una volta effettuata la connessione mediante invocazione del metodo *connect* della classe *TxRxDeviceManager* e ricevuta positiva conferma dall'invocazione del metodo di call-back *onTxRxServiceDiscovered* che garantisce che il dispositivo BLE è di tipo *TxRx*⁴ (in caso di fallimento della connessione sono alternativamente invocati i metodi di call-back *onConnectionError* o *onConnectionTimeout*), la comunicazione tra lo smartphone o il tablet e il dispositivo BLE *TxRx* avviene seguendo il seguente algoritmo:

- l'invocazione del metodo *requestWriteData* della classe *TxRxManager* – il cui parametro è la stringa che contiene il comando testuale da inviare al dispositivo BLE – scrive la caratteristica “Rx” del servizio *TxRx*: nel caso che la stringa da scrivere abbia lunghezza maggiore della dimensione dichiarata per la caratteristica sono effettuati invii multipli di frammenti della stringa ciascuno di lunghezza uguale o inferiore alla dimensione della caratteristica⁵
- l'invocazione del metodo di call-back *onWriteData* conferma la corretta scrittura del comando nella caratteristica “Rx”; in caso di errore o timeout sono invocati rispettivamente i metodi di call-back *onWriteError* o *onWriteTimeout* (il tempo di timeout è un parametro configurabile)
- con l'invio dell'ultimo frammento della stringa di comando (coincidente con il primo nel caso di stringa di lunghezza uguale o inferiore alla dimensione della caratteristica “Rx”) viene avviato un timer inizializzato con il tempo di timeout per l'inizio della risposta
- la ricezione dell'eventuale risposta al comando inviato avviene mediante successive notifiche BLE di variazione del contenuto della caratteristica “Tx”: i frammenti di stringa notificati in successione sono ricomposti in un'unica stringa di risposta; nel caso che nessuna notifica sia generata prima che il timer si azzeri viene invocato il metodo di call-back *onReadNotifyTimeout*
- successivamente alla ricezione di ogni singolo frammento della stringa di risposta viene avviato o riavviato un timer inizializzato con il tempo di timeout per la fine della risposta: nel caso che il timer si azzeri la lettura della risposta è considerata terminata

⁴l'invocazione del metodo *onDeviceConnected* notifica l'avvenuta connessione, ma non garantisce che il dispositivo connesso sia di tipo *TxRx*

⁵il protocollo BLE prevede la trasmissione di pacchetti con un *payload* massimo di 20 byte: nel caso che la lunghezza della caratteristica “Rx” sia superiore a 20, la trasmissione del valore da scrivere nella caratteristica viene suddivisa in pacchetti ciascuno con *payload* uguale o inferiore a 20 in modo trasparente per il codice Java che invoca le API BLE di Android

e la stringa che la contiene viene fornita come parametro del metodo di call-back *onNotifyData*

Ad eccezione dell'avvio del timer per il timeout, il codice di scrittura di un comando e quello di lettura di una risposta sono completamente asincroni: l'eventuale variazione non sollecitata della caratteristica "Tx" da parte del dispositivo BLE genera la ricezione della stringa con le stesse modalità della ricezione della risposta ad un comando.

5. Documentazione del codice del modulo "lib"

Il codice Java delle classi contenute nel package *com.tertiumtechnology.txrxlib* è stato commentato con lo stile previsto dallo standard JavaDoc: la documentazione in formato HTML dei package, delle classi, degli attributi, dei metodi e dei parametri è di conseguenza stata generata automaticamente.