

APP *TertiumBleScan* e *TertiumBleScanSwift* (TxRx) per sistema operativo iOS

1. Scopo e funzionalità della APP

Le APP per sistema operativo iOS *TertiumBleScan* e *TertiumBleScanSwift* sono rilasciate 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. Le 2 APP hanno le stesse funzionalità e presentano la stessa interfaccia utente, si distinguono in base al linguaggio di programmazione utilizzato per la realizzazione, rispettivamente: Objective-C e Swift.

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 iPhone o un tablet iPad con sistema operativo iOS, utilizzando la caratteristica “Rx” ed invia le relative risposte utilizzando la caratteristica “Tx”.

Le 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 delle APP *TertiumBleScan* e *TertiumBleScanSwift* (denominato “TxRxLibrary”) è stato scritto e documentato in forma di libreria per essere riutilizzato in tutte le APP che interagiscono con i dispositivi BLE *TxRx* di TERTIUM Technology; la GUI è intesa come utile esemplificazione dell’uso delle funzionalità di “TxRxLibrary”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 delle APP *TertiumBleScan* e *TertiumBleScanSwift*

Il codice sorgente della libreria delle APP *TertiumBleScan* e *TertiumBleScanSwift* è interamente compreso nel progetto libreria “TxRxLibrary” sia per la versione Objective-C che per la versione Swift. La classe *Core*, contenuta nella directory di progetto “Core”, realizza un *proxy* di tipo *singleton* che interfaccia la libreria con il resto del codice che implementa le interfacce utente delle 2 APP e un oggetto delegato che effettua la trasformazione in messaggi da visualizzare di tutte le notifiche asincrone generate dal codice della libreria e riflesse dalla classe *Core* (ricezione di risposte dal dispositivo, eventi di *timeout*, errori, ...). L’impiego della classe *Core* non è tuttavia necessario per l’uso della libreria: lo sviluppatore può scegliere un diverso *pattern* di gestione dei delegati di “TxRxLibrary”.

3. Funzionalità della classe *Core* delle APP *TertiumBleScan* e *TertiumBleScanSwift*

Per la APP Objective-C *TertiumBleScan* la classe *Core* comprende i seguenti metodi:

Metodo	Funzionalità
<code>getCore</code>	restituisce l’istanza <i>singleton</i> della classe <i>Core</i>
<code>isScanning</code>	restituisce <i>true</i> se è attiva la fase di ricerca dei dispositivi BLE, <i>false</i> altrimenti
<code>startScan</code>	attiva la fase di ricerca dei dispositivi BLE: ogni singolo dispositivo trovato genera una notifica, anche nel caso non sia un dispositivo <i>TxRx</i>
<code>stopScan</code>	disattiva la fase di ricerca dei dispositivi BLE
<code>getScannedDevices</code>	restituisce l’elenco dei dispositivi BLE trovati nella fase di ricerca (comprende anche i dispositivi non <i>TxRx</i>)
<code>connectDevice</code>	connette il dispositivo specificato tra quelli trovati nella fase di ricerca: la connessione avviene esclusivamente con dispositivi <i>TxRx</i>
<code>sendData</code>	invia una stringa di dati al dispositivo connesso: la ricezione della risposta genera una notifica
<code>disconnectDevice</code>	disconnette il dispositivo specificato

Inoltre la classe *Core* Objective-C riflette le seguenti notifiche:

Notifica	Descrizione
<code>TxRxScanBegan</code>	inizio della fase di ricerca
<code>TxRxScanError</code>	errore nella fase di ricerca
<code>TxRxScanEnded</code>	fine della fase di ricerca
<code>TxRxDeviceError</code>	errore del dispositivo

TxRxDeviceFound	trovato un dispositivo BLE
TxRxDeviceConnectError	errore di connessione con il dispositivo ² (comprende l'errore di timeout)
TxRxDeviceConnected	effettuata connessione al dispositivo
TxRxDeviceReady	il riconoscimento del dispositivo di tipo <i>TxRx</i> è avvenuto correttamente: il dispositivo è pronto a ricevere comandi
TxRxDeviceDisconnected	effettuata disconnessione dal dispositivo
TxRxDeviceDataSent	dati trasmessi correttamente al dispositivo
TxRxDeviceDataSendError	errore di trasmissione dei dati al dispositivo
TxRxDeviceDataSendTimeout	timeout di trasmissione dei dati al dispositivo
TxRxDeviceDataReceived	ricezione di dati dal dispositivo
TxRxDeviceDataReceiveError	errore di ricezione dei dati dal dispositivo (comprende l'errore di timeout)
TxRxDeviceInternalError	errore di gestione del dispositivo

La classe *Core Objective-C* non intercetta i seguenti metodi di utilità esposti dalla classe *TxRxManager*:

Metodo	Funzionalità
deviceWithIndexedName	restituisce un riferimento a un dispositivo trovato mediante <i>indexed-name</i> (nome del dispositivo seguito dal simbolo “_” e dalla posizione dello stesso nella lista dei dispositivi trovati in fase di ricerca) ³
getDeviceIndexedName	restituisce l' <i>indexed-name</i> del dispositivo specificato
setTimeoutDefaults	imposta i tempi di timeout di default
getTimeoutValue	restituisce il tempo di timeout specificato (connessione, ricezione primo frammento dei dati di risposta, ricezione successivi frammenti dei dati di risposta, trasmissione dati di un comando)
setTimeoutValue	imposta il tempo di timeout specificato (connessione, ricezione primo frammento dei dati di risposta, ricezione successivi frammenti dei dati di risposta, trasmissione dati di un comando)
deviceFromDeviceName	restituisce un riferimento a un dispositivo trovato mediante il nome del dispositivo (nome del dispositivo nella lista dei dispositivi trovati in fase di ricerca)
getDeviceName	restituisce il nome del dispositivo

²errore generato anche nel caso che il dispositivo non sia di tipo *TxRx*

³non rendendo possibile le API del sistema operativo iOS il recupero del MAC *address* del dispositivo BLE il ricorso all'*indexed-name* permette di ottenere un identificatore univoco nel caso che il nome BLE di due dispositivi rilevati coincida

Per la APP Swift *TertiumBleScanSwift* la classe *Core* comprende i seguenti metodi:

Metodo	Funzionalità
<code>getCore</code>	restituisce l'istanza <i>singleton</i> della classe <i>Core</i>
<code>isScanning</code>	restituisce <i>true</i> se è attiva la fase di ricerca dei dispositivi BLE, <i>false</i> altrimenti
<code>startScan</code>	attiva la fase di ricerca dei dispositivi BLE: ogni singolo dispositivo trovato genera una notifica
<code>stopScan</code>	disattiva la fase di ricerca dei dispositivi BLE
<code>getScannedDevices</code>	restituisce l'elenco dei dispositivi BLE trovati nella fase di ricerca (comprende anche i dispositivi non TxRx)
<code>connectDevice</code>	connette il dispositivo specificato tra quelli trovati nella fase di ricerca
<code>sendData</code>	invia una stringa di dati al dispositivo connesso: la ricezione della risposta genera una notifica
<code>disconnectDevice</code>	disconnette il dispositivo specificato

Inoltre la classe *Core* Swift riflette le seguenti notifiche:

Notifica	Descrizione
<code>TxRxScanBegan</code>	inizio della fase di ricerca
<code>TxRxScanError</code>	errore nella fase di ricerca
<code>TxRxScanEnded</code>	fine della fase di ricerca
<code>TxRxDeviceError</code>	errore del dispositivo
<code>TxRxDeviceFound</code>	trovato un dispositivo BLE
<code>TxRxDeviceConnectError</code>	errore di connessione con il dispositivo ⁴ (comprende l'errore di timeout)
<code>TxRxDeviceConnected</code>	effettuata connessione al dispositivo di tipo <i>TxRx</i>
<code>TxRxDeviceReady</code>	il riconoscimento del dispositivo di tipo <i>TxRx</i> è avvenuto correttamente: il dispositivo è pronto a ricevere comandi
<code>TxRxDeviceDisconnected</code>	effettuata disconnessione dal dispositivo
<code>TxRxDeviceDataSent</code>	dati trasmessi correttamente al dispositivo
<code>TxRxDeviceDataSendError</code>	errore di trasmissione dei dati al dispositivo
<code>TxRxDeviceDataSendTimeout</code>	timeout di trasmissione dei dati al dispositivo
<code>TxRxDeviceDataReceived</code>	ricezione di dati dal dispositivo
<code>TxRxDeviceDataReceiveError</code>	errore di ricezione dei dati dal dispositivo (comprende l'errore di timeout)
<code>TxRxDeviceInternalError</code>	errore del dispositivo

⁴errore generato anche nel caso che il dispositivo non sia di tipo *TxRx*

La classe *Core* Swift non intercetta i seguenti metodi di utilità esposti dalla classe *TxRxManager*:

Metodo	Funzionalità
<code>deviceWithIndexedName</code>	restituisce un riferimento a un dispositivo trovato mediante <i>indexed-name</i> (nome del dispositivo seguito dal simbolo “_” e dalla posizione dello stesso nella lista dei dispositivi trovati in fase di ricerca) ⁵
<code>getDeviceIndexedName</code>	restituisce l' <i>indexed-name</i> del dispositivo specificato
<code>setTimeoutDefaults</code>	imposta i tempi di timeout di default
<code>getTimeoutValue</code>	restituisce il tempo di timeout specificato (connessione, ricezione primo frammento dei dati di risposta, ricezione successivi frammenti dei dati di risposta, trasmissione dati di un comando)
<code>setTimeoutValue</code>	imposta il tempo di timeout specificato (connessione, ricezione primo frammento dei dati di risposta, ricezione successivi frammenti dei dati di risposta, trasmissione dati di un comando)
<code>deviceFromDeviceName</code>	restituisce un riferimento a un dispositivo trovato mediante il nome del dispositivo (nome del dispositivo nella lista dei dispositivi trovati in fase di ricerca)
<code>getDeviceName</code>	restituisce il nome del dispositivo

⁵non rendendo possibile le API del sistema operativo iOS il recupero del MAC *address* del dispositivo BLE il ricorso all'*indexed-name* permette di ottenere un identificatore univoco nel caso che il nome BLE di due dispositivi rilevati coincida

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

Una volta effettuata la connessione mediante invocazione del metodo *connectDevice* della classe *Core* e ricevuta positiva conferma dalla generazione delle notifiche *TxRxDeviceConnected* e *TxRxDeviceReady* che garantisce che il dispositivo BLE è di tipo TxRx (in caso di fallimento della connessione viene generata la notifica *TxRxDeviceConnectError*), la comunicazione tra lo smartphone iPhone o il tablet iPad e il dispositivo BLE TxRx avviene seguendo il seguente algoritmo:

- l'invocazione del metodo *sendData* della classe *Core* – 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 caratteristica
- la generazione della notifica *TxRxDeviceDataSent* conferma la corretta scrittura del comando nella caratteristica “Rx”; in caso di errore o timeout sono invocati rispettivamente i metodi di call-back *TxRxDeviceSendError* o *TxRxDeviceSendTimeout* (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 generata la notifica *TxRxDeviceReceiveError*
- 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 e la stringa che la contiene viene fornita come parametro della notifica *TxRxDeviceDataReceived*

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 della libreria delle APP *TertiumBleScan* e *TertiumBleScanSwift*

Il codice Objective-C e Swift delle classi contenute nella libreria delle APP *TertiumBleScan* e *TertiumBleScanSwift* è stato commentato con lo stile previsto dall'ambiente di sviluppo Apple per APP per sistema operativo iOS (*Quick-help*).