BLE SDK GUIDELINE

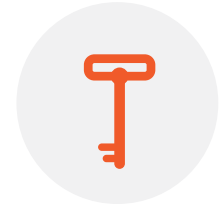
(iOS)


i-Neighbour


i-TimeTec


TimeTec Parking


TimeTec VMS


TimeTec Access

Project Owner: TimeTec Cloud Sdn Bhd

Prepared By: Nicholas Chong

Document Version Date: 23 August 2019

Disclaimer

No person should rely on the contents of this publication without first obtaining advice from a qualified professional person. The company expressly disclaims all and any liability and responsibility to any terminal or user of this book, in respect of anything, and of the consequences of anything, done by any such person in reliance, whether wholly or partially, upon the whole or any part of the contents of this book.

TimeTec Cloud Sdn Bhd

iOS TimeTec BLE Framework Guideline

This framework limits to TimeTec BLE-2 and BLE-16 with encryption features. There will be a compiler error due to the migration of Swift 4.2 to Swift 5 (Xcode incompatible issue) that caused by Apple INC. Hence, there are 2 versions of framework available which will be supporting in TimeTecBLE v4.2 and TimeTecBLE v5.0 only (We will keep it up to date with latest Swift version). However, you are encouraged to use the latest Xcode and Swift version. Anyhow, use the framework according to the Xcode version that you are using. Please take note that you will not be able to debug through Simulators, we are working hard to make this framework as Universal. Thank you.

	TimeTecBLE v4.2	TimeTecBLE v5.0
Xcode Version	v10.0 - v10.1	v10.2 onwards
Swift Version Required	v4.2 onwards	
Deployment Version	iOS 10.0 onwards	

Instructions to install TimeTecBLE Framework

1. Click on your project at Project Navigator, File -> Add Files to "..." (alt+command+A) -> select **"TimeTecBLE framework"** from your directory, ensure **"Copy items if needed"** is ticked.

2. Click on your project at Target List -> General -> Embedded Binaries -> add item and select **"TimeTecBLE framework"**

3. In your project target, Build Phases -> + Add a new build phase -> New Run Script Phase -> add in the following script

(this step is used to remove unused architectures before uploading to App Store).

```
FRAMEWORK="TimeTecBLE"
FRAMEWORK_EXECUTABLE_PATH="${BUILT_PRODUCTS_DIR}/${FRAME-
WORKS_FOLDER_PATH}/$FRAMEWORK.framework/$FRAMEWORK"
EXTRACTED_ARCHS=()
for ARCH in $ARCHS
do
lipo -extract "$ARCH" "$FRAMEWORK_EXECUTABLE_PATH" -o
"$FRAMEWORK_EXECUTABLE_PATH-$ARCH"
EXTRACTED_ARCHS+=("$FRAMEWORK_EXECUTABLE_PATH-$ARCH")
done
lipo -o "$FRAMEWORK_EXECUTABLE_PATH-merged" -create "${EXTRACTED_ARCHS[@]}"
rm "${EXTRACTED_ARCHS[@]}"
rm "$FRAMEWORK_EXECUTABLE_PATH"
mv "$FRAMEWORK_EXECUTABLE_PATH-merged" "$FRAMEWORK_EXECUTABLE_PATH"
```

4. Import the framework as **"import TimeTecBLE"** above your class.

5. You have successfully installed TimeTecBLE framework.

Types of Methods

There are 3 types of methods which are listed as below:

1) License Authentication

Declaration

```
func verifyLicense(licenseId: String, _ completion: @escaping
(Bool) -> Void)
```

Parameters

licenseId	License used to authenticate with TimeTec web server
completion	A boolean as a callback to determine whether it is success or failure

2) MAC Address Authentication

Declaration

```
func authenticateMacAddress(_ macAddress: String) -> (result:
Bool, bleType: String)
```

Parameters

macAddress	To validate MAC Address in the form of (00:00:00:00:00:00) or exactly 12 alphanumeric only. It is a validation step to authenticate the products you have.

3) BLE Data Encryption

Declaration

```
func getBLEData(bleType: BLEType, channel: Int = 1, macAddress:
String) -> Data
```

Parameters

bleType	BLE2 or BLE16
channel	Depends on the channel used in BLE with up to 16 channels (default channel given 1)
macAddress	Ensure MAC Address is passed correctly [in the form of 12 alphanumeric only or 00:00:00:00:00:00] (default MAC address given 000000000000)

Instructions to use TimeTecBLE Framework

1) License Authentication (Completion Handler)

Firstly, License Authentication is required before connecting to your TimeTec BLE products. The authentication requires a network connection to authenticate your products with the framework. Therefore, it might take a few seconds to verify your products. There's also a completion handler to get the Boolean value for identifying the license is valid. You may try to debug and look for the console messages too.

Example:

```
TimeTecBLE.shared.verifyLicense(licenseId: "String") { (Bool) in
<#code#>
}
```

Recommendation:

i) Run it on the first launch

This method can be triggered for the first time, but users will not be updated with the latest license. You can have the option to trigger this method at the first launch of app.

ii) Occasionally update users with the latest license

Not the most efficient method, but will keep users up to date if there's any additional BLE devices in the near future.

2) MAC Address Authentication (Returns Boolean, String)

Next, Mac Address Authentication is required before the data is sent to BLE devices. This method needs to be called after License Authentication or else you will get the return value as false all the time no matter the MAC Address is entered correctly. You may pass the MAC Address in the form of 00:00:00:00:00:00 or 000000000000.

Example:

```
let authentication = TimeTecBLE.shared.authenticateMacAddress("String")
if authentication.result {
print(authentication.bleType) // belongs to which type
                <#code#>
}
```

Recommendation:

Use this to authenticate the MAC Address and check is it valid among your BLE devices. If authentication.result returns TRUE, you may proceed by searching for Peripherals, Service Characteristics and write your data to Characteristics. You can know this MAC Address belongs to which type too and do your enumeration based on the result.

3) BLE Data Encryption (Returns Data)

Finally, BLE Data Encryption is the final step which will encrypt the data before writing data to Characteristics. Ensure MAC Address Authentication returns true then only call this method. BLEType is required (.BLE2 or S.BLE16), channel by default is 1. BLE-2 mostly used channel 1 by default unless there's other channel required to use.

Example:

BLE-2

```
let _ = TimeTecBLE.shared.getBLEData(bleType: .BLE2, macAddress: "String")
```

BLE-16

```
let _  = TimeTecBLE.shared.getBLEData(bleType: .BLE16, channel: Int, macAddress:
"String")
```

Recommendation:

In BLE-2, there are a total of 3 channels whereas BLE-16 contains of 16 channels. Please make sure License Authentication is called successfully in the early stage before this method. Bear in mind that you are required to determine which channel for BLE-16 but optional for BLE-2 as channel 1 is given by default.

Framework Version Updates

Version 1.0 (23 August 2019)

Supporting BLE2 and BLE16 frameworks in both Xcode 10.1 and Xcode 10.2 above.

.