

Using BLE on iOS

v1.2

Table of contents

About	3
Device Discovery	3
Connecting the device	4
BLE	4
Device Read/Write methods	4
SDK Example	4
Recommended practices	5
Error handling	5
Reconnecting the device	5
References	5
Revision history	7

About

uFCoder library supports communication & read/write operations via BLE.

As such, the support is limited to **uFR Online series** readers only.

Update uFCoder libraries to **5.0.83** and uFR Online reader to firmware version **2.9.1** for best results.

Latest uFCoder libraries can be found here:

- <https://www.d-logic.com/code/nfc-rfid-reader-sdk/ufr-lib>

Usage of uFCoder library in XCode can be found here:

- https://www.d-logic.com/code/nfc-rfid-reader-sdk/ufr-doc/blob/master/Using_uFR_library_in_Xcode.pdf

uFR SDK documentation can be found here:

- <https://www.d-logic.com/code/nfc-rfid-reader-sdk/ufr-doc.git>

Device Discovery

Due to the compatibility and necessity of platform-specific operations for the uFR device discovery, uFCoder library currently has **no** methods for the querying devices either on network (for UDP/TCP) and/or nearby devices (BLE) on any of the platforms.

For this use case, several examples have been created, most notably our "Finder" example that is available for iOS:

- iOS: https://www.d-logic.com/code/nfc-rfid-reader-sdk/ufr_online-examples-ios

And a simple iOS example for BLE only:

- iOS: https://www.d-logic.com/code/nfc-rfid-reader-sdk/ufr_online-examples-ios-ble.git

However, other than device discovery, uFCoder library supports connecting to the uFR series readers via ReaderOpenEx() method in our API, and read/write operations provided by the methods in our SDK (e.g GetCardIdEx(), GetDlogicCardType(), and more...).

Connecting the device

Users can utilize the **ReaderOpenEx()** method provided in our API for connecting to the device.

BLE

For example, to connect to the uFR Online via BLE, following parameters are necessary:

- Port name: Should contain uFR Online serial number that starts with "ON" prefix
- Port interface: the value 'L' implies the desired interface for connection (BLE)

Other parameters should be set to 0 or "" as shown below:

```
ReaderOpenEx(0, "ONXXXXXX", 'L', "");
```

or

```
ReaderOpenEx(0, "ONXXXXXX", 76, "");
```

More details and examples about supported communication types for different platforms can be found in the following document:

- https://www.d-logic.com/code/nfc-rfid-reader-sdk/ufr-doc/blob/master/Reader_Open_Examples.pdf

Device Read/Write methods

By simply calling methods from our API (e.g GetCardIdEx() method to get card UID) the uFCoder library will do necessary read/write operations based on connection type, and return values as specified in our API.

SDK Example

Example using native methods for discovering BLE devices and uFCoder library for communication with the uFR Online device can be found here:

<https://www.d-logic.com/code/nfc-rfid-reader-sdk/ufr-examples-ios-ble-get-uid>

Recommended practices

Error handling

Always check UFR_STATUS codes returned from the methods in uFCoder API, and proceed accordingly. Most common are UFR_COMMUNICATION_ERROR or UFR_COMMUNICATION_BREAK status codes. In such cases, try implementing a “retry” which will resend the command a few times, and then if all of the retries fail - try reconnecting to the device.

Reconnecting the device

The most straightforward way of checking if the device is connected and communication still established, is using the **ReaderStillConnected()** method from our API. If the method returns UFR_OK, parameter value should be 1 if the device is still available, and 0 in case of error(s) or a disconnect.

If the user concludes that reconnecting to the device is necessary, using **ReaderOpenEx()** to open communication with the reader again is mandatory...

References

For more details, refer to the following documents in our [ufr-doc](#) git repository:

- uFR Series NFC Reader API
- Reader Open Examples
- Reader opening - possible problems
- uFCoder API - Recommended practices
- Using uFR library in XCode

Revision history

Date	Version	Comment
2023-01-24	1.2	Recommended practices section added. References section added. About section updated.
2023-01-18	1.1	SDK Example section added.
2023-01-11	1.0	Base document