uFCoder API – Recommended practices
v1.1

Table of contents

Digital Logic Ltd.

About

This document covers most oftenly addressed issues & suggestions for overcoming problems with µFR API integration.

Reader Open

When using a cable connection with µFR series devices, the simplest way to establish communication would be using the ReaderOpen() method. The algorithm behind the ReaderOpen() method ensures establishing communication with almost all of the uFR Series readers.
In cases where some settings of the device were altered, e.g baudrate or RTS, it is recommended to use more advanced version of this method and provide the necessary parameters: ReaderOpenEx()

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

Testing the connection

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.

Error handling

Unstable connection with the device, such as malfunctioning cable/network connection (depending on previously discussed methods of establishing communication) or interference with reader's NFC field, may lead to errors.

Most frequent error codes returned via our API:

- UFR_NO_CARD - No card was detected in readers' NFC field. Indicates that either simply there was no card present in the readers' field at the time, or that the scanned card is not supported by the device.
- UFR_COMMUNICATION_ERROR - Indicates that data was transmitted, but the response was invalid. E.g data was corrupted in the process.
- UFR_COMMUNICATION_BREAK - Indicates that insufficient data was transmitted, e.g an unexpected connection loss occurred. It is recommended to make a small pause and then check with ReaderStillConnected() if the reader is still present and confirm if it was a temporary data loss.
- UFR_FT_STATUS_ERROR_# - Indicates error occurred within FTDI driver. Since FTDI driver/library is a dependency of the uFCoder library, it is uncommon to experience these errors. Most frequent error would be UFR_STATUS_ERROR_2. This error code most often implies a non-existent connection with the reader. Usually, ReaderOpen/Ex() was not called or it failed beforehand.

For the UFR_COMMUNICATION_* errors, other than checking with ReaderStillConnected() method, we suggest implementing a retry loop with a delay of e.g 20-30 ms between method calls. If the retries fail, try re-establishing communication with the reader via ReaderOpen()/ReaderOpenEx() methods.

Card polling

It is recommended to loop the GetCardIdEx() with a timeout of 100-150ms.
Depending on the communication type (cable connection, UDP/TCP, BLE, etc...) you can modify this value to suit your needs.
It is also recommended to use GetDlogicCardType() method, and check the card type, to ensure the correct tag type is in the field before you move on to using other methods from our API.

Revision history

Date	Version	Comment
2022-11-15	1.1	Section Card polling added.
2022-09-30	1.0	Base document