MultiReader API

v1.1

Table of contents

Digital Logic Ltd.

About

uFcoder library supports handling more than one compatible uFR reader connected to the host. The MultiReader API relies on both FTDI & COM ports for enumerating the available readers.
The methods used for this in the uFCoder library are:

- ReaderList_UpdateAndGetCount(),
- ReaderList_Add(),
- ReaderList_GetInformation(),
- ReaderList_OpenBySerial(),
- ReaderList_Destroy(),
- All of the functions provided by our uFCoder API that end with ***_M()** suffix

For more details on these methods, refer to "uFR Series NFC Reader API".
https://www.d-logic.com/code/nfc-rfid-reader-sdk/ufr-doc

Note: As of uFCoder library version 5.0.76, MultiReader API supports **UWP** (Universal Windows Platform).

Usage

ReaderList_UpdateAndGetCount()

This function should always be the first to be called in your software since it creates an internal list of available readers and enables interacting with them by the UFR handles later on provided by the ReaderList_GetInformation() function. The devices connected to the host will be queried by this function, opened and tested if they are valid uFR devices. Number of valid, available readers is returned through the parameter of this method.

As of uFCoder library version 5.0.73, this function has been refactored and expanded so that it supports readers connected via both FTDI & COM ports. On platforms where FTDI is unavailable (such as Linux systems with "ftdi_sio" module enabled) it will strictly query serial ports and try to open available uFR readers.

ReaderList_Add()

Used for manual addition of an uFR reader and getting their handle for usage with MultiReader API. Best used for cases where ReaderList_UpdateAndGetCount() did not manage to find the reader, or for example

adding uFR Online series reader via UDP/TCP or other interface.

Parameters used are the same as when using the ReaderOpenEx() function, except for the first parameter, which will store the handle of the reader on success.

After adding the reader to the list successfully via this method, the details of the reader can be retrieved normally via ReaderList_GetInformation().

ReaderList_GetInformation()

This function should be called as many times as there are readers. Since the method has an internal counter that indexes readers and that is reset by the ReaderList_UpdateAndGetCount(), it should always be called for every reader found by the ReaderList_UpdateAndGetCount().

This function simply returns the valid handle of the reader along with the details of the reader as the other parameters.

As of uFCoder version 5.0.76 this method has been refactored so that it takes a device index as a parameter, and can be called multiple times for that index to return the details about the device. Device index is determined based on the number of devices found by ReaderList_UpdateAndGetCount() and starts at zero. E.g if ReaderList_UpdateAndGetCount() has found 4 devices, the device index is in range 0-3.

ReaderList_OpenBySerial()

Used for opening the readers from the list by their serial number, that were, for example, in the meantime closed by the ReaderCloseM() method. The parameters are the handle and the serial number of the device in question.

ReaderList_Destroy()

Used for removing a reader from the list of devices. Simply provide the handle, and this method will try to properly close the communication with the reader, delete it from the list, and finally the handle will be freed and set to null.

*_M() suffix

Methods from uFCoder API have their counterparts for MultiReader API with the _M() suffix, and the handle as the first parameter. The first parameter, handle, is used to specify the reader for the selected operation.

For example, GetCardIdExM() will get the reader specified by the handle to try & read the card in the field.

Multireader query

ReaderList_UpdateAndGetCount() tests following connections in order:
1. 1Mbps baudrate, no changes to RTS
   - On platforms with FTDI support (Windows, Linux, OSX etc), the first step will check if the device is FTDI supported and the descriptors. If the descriptors contains "uFR" or "nFR " keywords, assumed baudrate will be 1Mbps and there will be no changes to RTS.
2. 1Mbps baudrate, no changes to RTS
3. 1Mbps, sets RTS to LOW
4. 1Mpbs, sets RTS to HIGH
5. 115200kps, no changes to RTS
6. 115200kbps, sets RTS to LOW
7. 115200kbps, sets RTS to HIGH

The exceptions to this procedure are UWP (Universal Windows Platform) and Linux OS with "ftdi_sio" module enabled.
Due to FTDI drivers not being enabled on them, the procedure slightly differs:
1. 1Mbps baudrate, no changes to RTS
2. 1Mbps baudrate, sets RTS to LOW
3. 1Mpbs baudrate, sets RTS to HIGH
4. 115200kps baudrate, no changes to RTS
5. 115200kbps baudrate, sets RTS to LOW
6. 115200kbps baudrate, sets RTS to HIGH

For an optimal performance when using ReaderList_UpdateAndGetCount() method & searching for the devices, it'd be best if your device(s) is/are configured as the more prioritized configuration described above.

On UWP, it is recommended to use non-inverted RTS for uFR devices (Nano Plus, CS, etc...) so that they are active while RTS is set to HIGH, due to platform specific conditions when using serial communication.
If the reader has inverted RTS, the initial state of the reader will be as he is in the state of reset, until correct configuration is applied from the procedure above to open the reader.

Examples

Examples for the MultiReader API are provided in our git repository:
https://www.d-logic.com/code/nfc-rfid-reader-sdk
- C example
- C# Example
- Lazarus example

Troubleshooting

Known issues

~~ReaderList_GetInformation()~~

~~As of uFCoder version 5.0.73 and below this method is known to cause problems when used without previously calling ReaderList_UpdateAndGetCount().~~

~~To avoid issues while trying to get information about the readers, please keep in mind the following:~~
- ~~Should only be executed after everything related to creating the device list and adding readers to the list. So keep in mind the following order of execution:~~
    - ~~ReaderList_UpdateAndGetCount()~~
    - ~~ReaderList_Add()~~
    - ~~ReaderList_GetInformation()~~
- ~~Loop the function based on the number of devices connected. No more than that. For example, if ReaderList_UpdateAndGetCount() finds two readers, and you add another one with ReaderList_Add() - ReaderList_GetInformation() should run three times.~~
- ~~This method contains an internal counter that indexes the readers and returns their data. This may change in the future. This counter is only reset when ReaderList_UpdateAndGetCount() is called.~~

As of ufCoder version 5.0.76 there are no known issues with the MultiReader API.

uFCoder library is periodically updated, as such, the related documentation and software examples will be updated to match the latest library version.
Latest libraries can be found here:
https://www.d-logic.com/code/nfc-rfid-reader-sdk/ufr-lib.git

Revision history

Date	Version	Comment
2022-11-02	1.1	Section Multireader query added. UWP supported procedure added. Known issues section updated.
2022-10-26	1.0	Base document