Reader Open Examples
v1.3

Digital Logic Ltd.

Address: Nemanjina 57a, 12000 Pozarevac, Serbia · Tel: +38112541022· VAT: 111385444 Reg. 21473642
e-mail: office@d-logic.com · www.d-logic.com

Table of contents

Address: Nemanjina 57a, 12000 Pozarevac, Serbia · Tel: +38112541022· VAT: 111385444 Reg. 21473642
e-mail: office@d-logic.com · www.d-logic.com

Digital Logic Ltd.

Address: Nemanjina 57a, 12000 Pozarevac, Serbia · Tel: +38112541022· VAT: 111385444 Reg. 21473642
e-mail: office@d-logic.com · www.d-logic.com

About

Starting with uFCoder library version 5.0.61 major changes to port open procedure have been introduced. In particular, ReaderOpen() method from our API has been refactored & expanded, and as such it now contains multiple steps when trying to open a port for communication between the host and the uFR Series reader(s). The more advanced method, ReaderOpenEx() has had minor bug fixes and changes, however it still works as intended - based on parameters provided.

Platforms

Windows

Communication types

Supports both FTDI communication & serial communication for cable connection, along with UDP/TCP for uFR Online series readers only.

Architectures supported

x86, x86_64.

Libraries

Directory “ufr-lib/windows” contains both static & dynamic libraries for the supported architectures.

UWP (Universal Windows Platform)

Communication types

Supports serial communication only.

Architectures supported

ARM, x86, x86_64

Libraries

In addition to “uFCoder” libraries, “uwp-serial” libraries act as dependencies so the presence of both libraries is mandatory. UWP projects require specific capability for serial communication to be enabled. Refer to “Using uFCoder library on UWP” document for more details.

UWP libraries for the supported architectures are located inside the “ufr-lib/windows/uwp” directory.

Linux

Communication types

Serial communication for “/dev/ttyS*” ports, serial communication for “/dev/ttyUSB*” ports if “ftdi_sio” module is present, FTDI communication is available only if “ftdi_sio” module is not present/blacklisted, UDP/TCP communication for uFR Online series readers.

Architectures supported

x86, x86_64, ARMel, ARMhf, ARM64 (aarch64).

Libraries

Directory “ufr-lib/linux” contains both static & dynamic libraries for the supported architectures.

MacOS

Communication types

Serial communication, FTDI communication.

Architectures supported

x86_64 only.

Libraries

Directory “ufr-lib/macos” contains both static & dynamic libraries for the supported architecture.

Android

Communication types

FTDI communication for uFR devices via OTG cable, Android internal NFC (APDU commands only), Sunmi devices (NFC, PSAM slot), Nexgo devices (contact card side slot, PSAM1, PSAM2).

Libraries

As of v5.0.61 there are three different different distributions of the uFCoder Android (.aar) library:

- Android library (supports internal Android NFC only and is located in “ufr-lib/android” directory).

- Android Nexgo (supports internal Android NFC & Nexgo devices and is located in "ufr-lib/android_nexgo" directory).
- Android Sumi (supports internal Android NFC & Sunmi devices and is located in "ufr-lib/android_sunmi" directory)

Integration & support for more devices will be in future updates.

iOS

Communication types

The iOS library currently supports µFR Nano Online reader only. It includes support for UDP, TCP, BLE communication and usage of iOS internal NFC (APDU commands only). All of these options are available by using the *ReaderOpenEx()* method with the appropriate parameters.

ReaderOpen()

About

As of uFCoder library v5.0.61, scope of *ReaderOpen()* method was expanded to encompass all of our most frequent reader setups. As such it now contains 7 steps when trying to find & establish the communication with the first available uFR Series Reader.
The purpose of these steps is the automatization of the reader open procedure. The *ReaderOpen()* method will stop at the first found & successfully connected device, and as such will return *UFR_OK (0)* status. Otherwise it will continue execution of the steps in this order, if no device was found with any of these steps - *UFR_READER_OPENING_ERROR* status is returned.

Algorithm

ReaderOpen() order of execution is as follows:

1. uFR reader(s) with "uFR" descriptors in their product description. FTDI port with the baud rate 1Mbps, no changes to RTS (by default uFR devices have inverted FTDI RTS)
   - e.g uFR Nano, uFR CS, uFR Nano Online (transparent mode, requires cable connection)...
      a. On UWP, the first step instead uses serial communication with the same setup, due to FTDI not being available.
2. Serial communication, at baud rate 115200bps, this step resets the device twice, first it tries setting the RTS to HIGH and tests the communication with the device, if this fails then it immediately sets

Address: Nemanjina 57a, 12000 Pozarevac, Serbia · Tel: +38112541022· VAT: 111385444 Reg. 21473642
e-mail: office@d-logic.com · www.d-logic.com

the RTS to LOW and tests again.
- e.g RS232 uFR Series readers, CDC readers…

3. FTDI communication, at baud rate 115200bps, resets the device and tests the communication.
   - e.g uFR RS232 with inverted RTS
4. FTDI devices at 115200bps **without** changes to RTS
5. Serial communication at 115200bps, **without** changes to RTS
6. FTDI communication at 250000bps **without** changes to RTS
7. Serial communication at 250000bps **without** changes to RTS

As an alternative to *ReaderOpen()* method, these steps can be instead executed via the *ReaderOpenEx()* method, with several calls & different parameters. Examples by steps:

1. ReaderOpenEx(1, "", 2, "UNIT_OPEN_RESET_DISABLE");
2. ReaderOpenEx(2, "", 1, "READER_ACTIVE_ON_RTS_HIGH");
   ReaderOpenEx(2, "", 1, "READER_ACTIVE_ON_RTS_LOW");
3. ReaderOpenEx(2, "", 2, "");
4. ReaderOpenEx(2, "", 2, "UNIT_OPEN_RESET_DISABLE");
5. ReaderOpenEx(2, "", 1, "UNIT_OPEN_RESET_DISABLE");
6. ReaderOpenEx(3, "", 2, "UNIT_OPEN_RESET_DISABLE");
7. ReaderOpenEx(3, "", 2, "UNIT_OPEN_RESET_DISABLE");

ReaderOpenEx()

About

More advanced method that requires the user to specify parameters.

Parameters

Parameters for this method are:

1. Reader type
2. Port name
3. Port interface
4. Additional argument(s)

Reader type

Mainly used to specify device baud rate based on known types of our readers.
Supported values:

Address: Nemanjina 57a, 12000 Pozarevac, Serbia · Tel: +38112541022· VAT: 111385444 Reg. 21473642
e-mail: office@d-logic.com · www.d-logic.com

0 : auto - it will iterate through different values of the baud rate (1-3). If this parameter is specified as 0 along with port interface - ReaderOpenEx() will act as if ReaderOpen() was called.

1: uFR type (1 Mbps)

2: uFR RS232 (115200 bps)

3: BASE HD (250 Kbps)

Port name

Specifies the port by it's given name. For serial communication (port interface 1), depending on the platform, will have "COM*" on Windows, "/dev/ttyS*" (or if "ftdi_sio" is loaded, "/dev/ttyUSB*" will appear for USB connections) on Linux, "/dev/tty.usb*" on Mac etc...

If the port interface is set as 2 (FTDI communication), the port name parameter should be the FTDI serial number.

If this parameter is left as an empty string, the uFCoder library will try to find & iterate through available ports on it's own.

Port interface

Specifies the type of communication interfaces (defines interface which will be used when trying to connect to the reader).

Supported values:

0: (auto) first try FTDI , then serial, if port name is not defined

1: Try serial / virtual COM port / interfaces

2: Try FTDI only communication interfaces

10: Try to open Digital Logic Shields with RS232 uFReader on Raspberry Pi (serial interfaces with GPIO reset)

84: ('T') : TCP/IP interface

85: ('U') : UDP interface

102: ('B'): BT serial interface (Android library only)

114: ('L'): BLE interface (Android library only). When uFR Online reader works in BT serial mode, port_interface must be set to 0 (Except Android)

Additional argument (arg)

Used to specify additional settings when trying to establish communication. If left as empty, the default value is assumed as "READER_ACTIVE_ON_RTS_LOW'.

Supported values:

"UNIT_OPEN_RESET_DISABLE": do not reset the reader when opening

"UNIT_OPEN_RESET_FORCE" : force reset the reader when opening

"READER_ACTIVE_ON_RTS_LOW" : (default) Reset the reader when RTS is high - the reader works when RTS is low.

"READER_ACTIVE_ON_RTS_HIGH" : Reset the reader when RTS is low - the reader works when RTS is high.

You can specify more than one setting with space as the delimiter of the string.
If the port interface is set as 2 (FTDI communication), RTS is checked internally and will be set correctly to reset the device, without the need to specify it through additional argument(s).

Examples

Serial port

Windows: ReaderOpenEx(1, "COMXX", 1, "");
UWP: ReaderOpenEx(1, "WINIOT:X", 1, "");
Linux:
ReaderOpenEx(1, "/dev/ttyUSBX", 1, ""); (if module "ftdi_sio" is loaded)
ReaderOpenEx(1, "/dev/ttySX", 1, "");
MacOS: ReaderOpenEx(1, "/dev/tty.usbserial-xxxxxxx", 1, "");

FTDI

ReaderOpenEx(1, "A67M24SC", 2, "");
UWP does not support this type of communication.

UDP

ReaderOpenEx(0, "ip_address", 85, "");
or
ReaderOpenEx(0, "ip_address", 'U', "");

TCP

ReaderOpenEx(0, "ip_address", 84, "");
or
ReaderOpenEx(0, "ip_address", 'T', "");

Android internal NFC

ReaderOpenEx(5, "", 0, "");

Digital Logic Ltd.
Address: Nemanjina 57a, 12000 Pozarevac, Serbia · Tel: +38112541022· VAT: 111385444 Reg. 21473642
e-mail: office@d-logic.com · www.d-logic.com

BLE

Android

"port_name" parameter can contain either MAC address with ":" delimiter, or uFR Online series reader serial number (ONXXXXXX)
e.g ReaderOpenEx(0, "ONXXXXXX", 'L', "");
or
ReaderOpenEx(0, "aa:bb:cc:dd:ee:ff", 'L', "");

iOS

"port_name" parameter should contain uFR Online series reader serial number (ONXXXXXX):
e.g ReaderOpenEx(0, "ONXXXXXX", 'L', "");

BT Serial

Android

"port_name" parameter can contain either MAC address with ":" delimiter, or uFR Online series reader serial number (ONXXXXXX)
e.g ReaderOpenEx(0, "ONXXXXXX", 'B', "");
or
ReaderOpenEx(0, "aa:bb:cc:dd:ee:ff", 'B', "");

Additional notes (BT & BLE)

Up to uFCoder version 5.0.64: uFR Online Series reader needs pairing with the Android beforehand.
As of uFCoder version 5.0.65: uFR Online Series reader no longer requires pairing with the Android device. The user can simply provide the MAC address/serial number as '*port_name*' parameter of the *ReaderOpenEx()* method, and the uFCoder library will try to find the device & connect to it. Depending on the settings of the uFR Online reader, user will be then prompted to pair it after a successful connection.

Address: Nemanjina 57a, 12000 Pozarevac, Serbia · Tel: +38112541022· VAT: 111385444 Reg. 21473642
e-mail: office@d-logic.com · www.d-logic.com

Revision history

Date	Version	Comment
2022-03-14	1.3	Section Additional notes (BT & BLE) added.
2022-03-10	1.2	Examples updated: Android & iOS BLE. Android BT.
2022-03-08	1.1	Revision history updated. Title of the document updated. Section Algorithm updated. Android & iOS communication details updated.
2022-03-07	1.0	Base document

Digital Logic Ltd.

Address: Nemanjina 57a, 12000 Pozarevac, Serbia · Tel: +38112541022· VAT: 111385444 Reg. 21473642
e-mail: office@d-logic.com · www.d-logic.com