uFCoder PC/SC Driver

v1.1

Table of contents

Reader open procedure

As of version 0.0.0.4, the uFCoder PC/SC driver will try to establish communication with the reader based on the parameters that are contained in the registry.

Path to registry keys used for reader open procedure is:
HKEY_LOCAL_MACHINE/SOFTWARE/Dlogic/PCSC

As a default, the installer will set the following key values:

- **OPEN:** Determines method and parameters used for establishing the communication. Default value of this key is "*ReaderOpen*", and as such, it will try using the ReaderOpen() method from our uFCoder API to find & establish communication with the reader. Alternatively, the user can change this value to a more advanced option.
    - "*ReaderOpenEx %d %s %d %s'* is the format of the more advanced instruction for establishing the communication. The method and the parameters provided are in the same format for the ReaderOpenEx() function, as described in our uFCoder API.
- **INTERFACE:** Specifies reader interface for connecting to a tag. Values can be:
    - **NFC:** as the name implies, it will rely on polling the tags in via NFC.
    - **SAM:** it will try to establish communication via ISO7816 protocol with the tag in SAM slot, if available.

If the value for **OPEN** key is invalid - default value *'ReaderOpen'* will be used.
If the value for **INTERFACE** is invalid - default value *'NFC'* will be used.

Instead of manual edit of these values, you can use the "**uFCoderPcSc_PortSettings**"tool provided in our SDK: https://www.d-logic.com/code/nfc-rfid-reader-sdk
And NFC Tools: https://www.d-logic.com/code/ufr_nfc_tools/

Reader open example

"OPEN" parameter should always follow the following formats:
- "ReaderOpen"
or
"ReaderOpenEx %d %s %d %s"

It conforms to the ReaderOpenEx() method as described in uFCoder Series API, and as such, is used as an advanced method for establishing communication with the uFR readers. For example, via network (uFR Online, TCP, UDP, BT Serial).
1st and 3rd parameters, reader type & port interface are in integer format. As such no other format will be recognized. Port name and additional args should always be passed as strings.

You can refer to our "Reader_Open_Examples" document that can be found here:
https://www.d-logic.com/code/nfc-rfid-reader-sdk/ufr-doc/blob/master/Reader_Open_Examples.pdf

For more details on supported communication types and examples in our API.

Supported card types

Mifare Mini®, Mifare Classic® 1K®, Mifare Classic® 4K, Type 2 tags (NTAG2XX), ISO14443-4 compatible tags.

Command list

CLA	INS	Command	Description
FF	82	LOAD_KEY	Storing the keys into either reader EEPROM or Driver memory. Additionally, it can be used to lock/unlock the reader keys.
FF	86	GENERAL_AUTHENTICATE	Authentication of the contactless tag in the field.
FF	B0	READ_BINARY	Read bytes from a contactless card
FF	CA	GET_DATA	Currently only supports getting the UID of the tag.
FF	D6	UPDATE_BINARY	Write bytes to a contactless card

LOAD_KEY (0x82)

Instruction 0x82 (LOAD_KEY) uses CRPYTO1 key(s) and depending on the P1 parameter, is capable of the following operations:

- Reader key locking/unlocking: Key provided will lock the reader's keys to prevent further changes using the 8 byte key provided.
- Writing the CRYPTO1 key into the reader: Keys stored in the reader as CRYPTO1 are always 6 bytes long.
- Writing the CRYPTO1 key into the Driver memory. Maximum 2 keys are stored as Provided Key 0 and Provided Key 1.

Since the P1 parameter is used to define which specific operation will be used, P1 can have the following values:

- 0x80 - Reader Password (used for lock/unlock of reader keys)
- 0x20 - Reader Key (will store the key into reader EEPROM)
- Other - Driver key. (will store the key into Driver memory)

Instruction format:

CLA	**INS**	**P1**	**P2**	**Lc**	**DataIn**	**Le**
FF	82	Key Type	Key Index	Length of **DataIn**	Key Value	-

Reader keys lock/unlock

CLA	**INS**	**P1**	**P2**	**Lc**	**DataIn**	**Le**
FF	82	80	00	08	FF FF FF FF FF FF FF FF	XX

This instruction will always try to unlock keys first, if they are already unlocked, then it will move on to locking them with the key (password) provided.

Parameters

P1 - Value of 0x80 indicates that the procedure for reader key lock/unlock will be used.

P2 - Is not used. Set to 0x00.

LC - Indicates length of the reader key used for lock/unlock. Based on the uFCoder API, the key is always 8 bytes long, and the value in this scenario must always be in hexadecimal digits. E.g 8 bytes of 0xFF.

DataIn - Contains the key (password) that will be used to lock/unlock reader keys. 8 hex bytes long.

Write reader key into EEPROM

CLA	INS	P1	P2	Lc	DataIn	Le
FF	82	20	80	06	FF FF FF FF FF FF	XX

Parameters

P1 - With the value of 0x20 indicates that the key will be stored into the reader EEPROM.

P2 - Is used to define the key index. Value of 0x80 actually index 0. Key index is calculated as:

P2 &= 0x7F. uFR Reader key index possible values are in the range of 0-31 for CRYPTO1 keys, as such, maximum value of P2 in this scenario will be 0x9F.

LC - Indicates length of the reader key used that will be stored into the EEPROM. The CRYPTO1 key is always 6 hex bytes long.

DataIn - Contains the key that will be stored into the EEPROM.

Write Driver memory key

CLA	INS	P1	P2	Lc	DataIn	Le
FF	82	XX	YY	06	FF FF FF FF FF FF	XX

Parameters

P1 - Must differ from the previously mentioned values of 0x20 and 0x80. E.g set it to 0x00.

P2 - Indicates the index of the Driver memory key. Possible values are 0x00 and 0x01.

LC - Length of the key that will be stored into Driver Memory. The CRYPTO1 key is always 6 hex bytes long.

GENERAL_AUTHENTICATE (0x86)

Instruction format:

CLA	INS	P1	P2	Lc	DataIn	Le
FF	86	00	00	05	01 00 XX YY 00	XX

Parameters

P1 - Always set to 0x00.

P2 - Always set to 0x00.

LC - Indicates length of DataIn parameter. In this scenario the length should always be 5.

DataIn - Contents of the DataIn command can be summarised as the:

- First byte in DataIn is always 0x01 and indicates Version.
- Second and third bytes of DataIn refer to block address. Second byte is always 0x00, since the maximum value of block address for Mifare Classic® 4K is 0x00FF. As such, only the third byte actually defines the value of the block address.
- Fourth byte of DataIn indicates authentication mode. E.g Value of 0x60 is, according to the uFCoder API, MIFARE_AUTHENT1A.
- Fifth and last byte of DataIn is the key flag. Indicates which key will be used for authentication.
  The following values can be used:
  0x00 - Provided Key 0 previously stored in the Driver memory with the LOAD_KEY instruction.
  0x01 - Provided Key 1 previously stored in the Driver memory with the LOAD_KEY instruction
  0x80-0x9F - Index of a reader key stored in EEPROM. As mentioned above, key index is calculated as <value> &= 0x7F -> index result.

READ_BINARY (0xB0)

Instruction format:

CLA	INS	P1	P2	Lc	DataIn	Le
FF	B0	00	00	00	- -	XX

Parameters

Authentication key used will be based on the 5th byte previously provided in **GENERAL_AUTHENTICATE** command.

P1 - Always set to 0x00

P2 - Block address that will be read

Le - Any value just so it differs from 0x00. This instruction will always return 16 bytes of data, regardless of Le value.

GET_DATA (0xCA)

Currently, the only usage of this instruction is to get the UID of the tag.
Instruction format:

CLA	INS	P1	P2	Lc	DataIn	Le
FF	C0	00	00	00	- -	XX

UPDATE_BINARY (0xD6)

Instruction format:

CLA	INS	P1	P2	Lc	DataIn	Le
FF	C0	00	00	00	00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF	–

Parameters
Authentication key used will be based on the 5th byte previously provided in **GENERAL_AUTHENTICATE** command.
P1 - Always set to 0x00
P2 - Block address value.
LC - Always set to 0x10 (16).
LE - Omitted. No value.
DataIn - 16 hex bytes of data that will be written to the provided block address.

Revision history

Date	Version	Comment
2022-08-09	1.1	Reader open example section added
2022-08-09	1.0	Base document