

μFR Mifare examples - Android

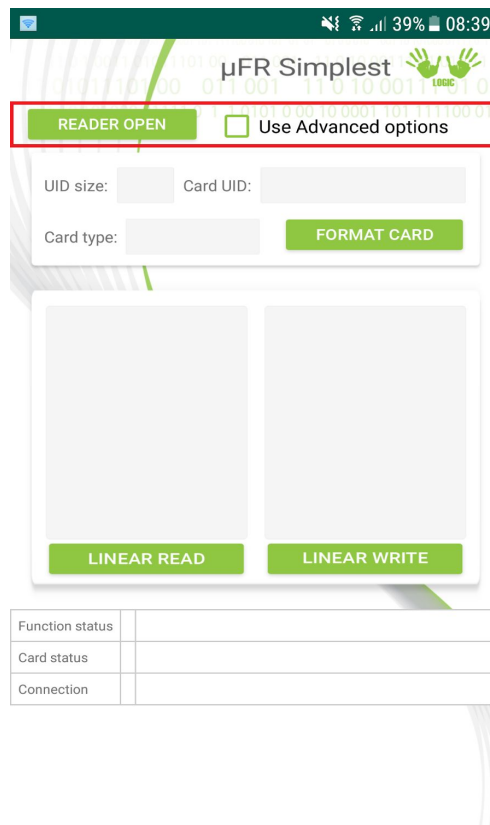
Version 1.0

Table of contents

Simplest	3
Communication with μ FR Reader	3
Reading card UID	4
Read and write operations	5
Simple	6
Communication with μ FR Reader	6
Reader options	7
Card options	8
Advanced	11
Communication with μ FR Reader	11
Reader options	11
List of options	12
Authentication	13
Block and Block in sector Read/Write	14
Value blocks	15
Revision history	20

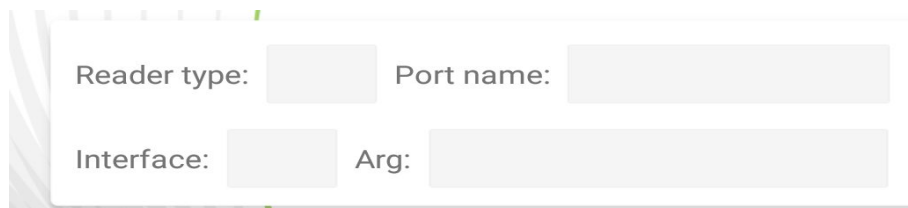
Simplest

Communication with μ FR Reader



Function status	
Card status	
Connection	

For opening communication with reader click **READER OPEN** button or select **Use Advanced options** if you are using for example μ FR Online device, fill in the blanks and then click **READER OPEN**.



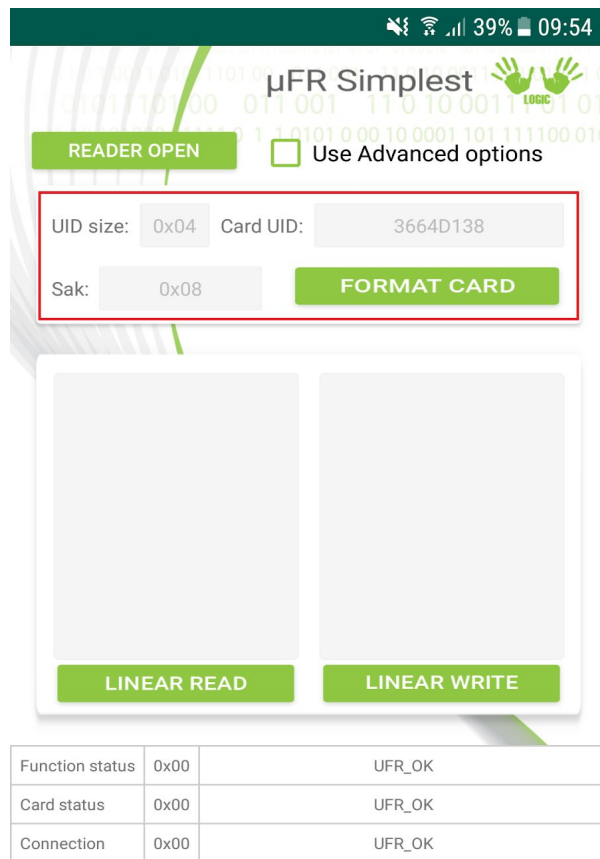
For **Reader type** fill empty field with **0**, for **Port name** fill empty field with reader **IP address**, for **Interface** fill empty field with **T** or **U** depending on your readers configuration, for **Arg** fill empty field with **0**.

If establishing connection was successful, you will see **UFR_OK** status next to **Connection**.

Connection	0x00	UFR_OK
------------	------	--------

Reading card UID

Every time you put your card on the reader you will be able to see its UID, Sak, and UID size.



μFR Simplest

READER OPEN ☐ Use Advanced options

UID size: 0x04 Card UID: 3664D138

Sak: 0x08 **FORMAT CARD**

LINEAR READ LINEAR WRITE

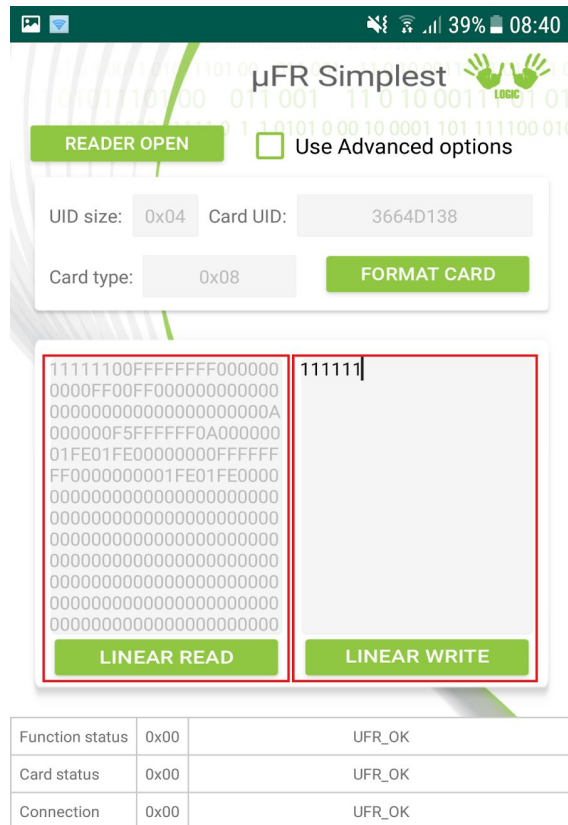
Function status	0x00	UFR_OK
Card status	0x00	UFR_OK
Connection	0x00	UFR_OK

If you click on **FORMAT CARD** button reader will write **Key A: 0xFFFFFFFFFFFF** and **Key B: 0xFFFFFFFFFFFF**

In all sectors on the card and will set **access bit 0** for all data blocks and **access bit 1** for all sector trailer blocks.

Read and write operations

For reading card, simply click on **LINEAR READ** button, and for writing, type data into text field in **hexadecimal format** and click **LINEAR WRITE** button.



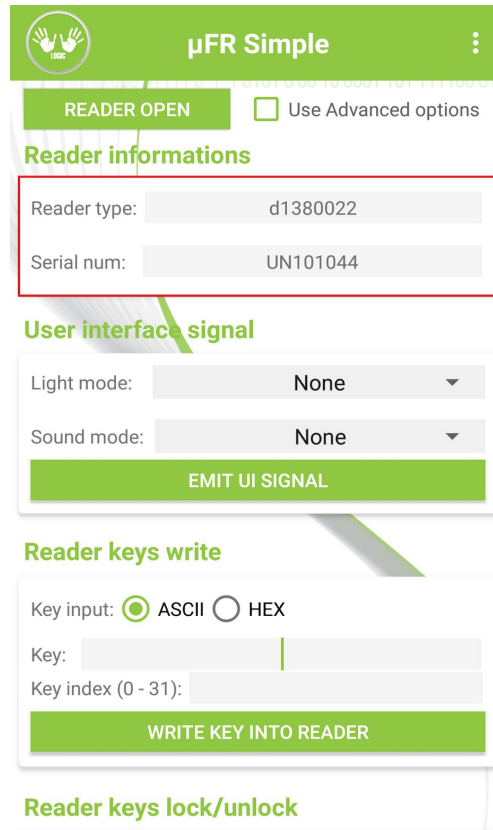
The screenshot shows the uFR Simplest application interface. At the top, there's a status bar with icons for signal, battery (39%), and time (08:40). Below the title "uFR Simplest", there's a green "READER OPEN" button and a checkbox for "Use Advanced options". The configuration section shows "UID size: 0x04", "Card UID: 3664D138", "Card type: 0x08", and a green "FORMAT CARD" button. The main area has two text fields: the left one contains a long hexadecimal string, and the right one contains "111111". Below these fields are "LINEAR READ" and "LINEAR WRITE" buttons. At the bottom, there's a table showing status information.

Function status	0x00	UFR_OK
Card status	0x00	UFR_OK
Connection	0x00	UFR_OK

Simple

Communication with μ FR Reader

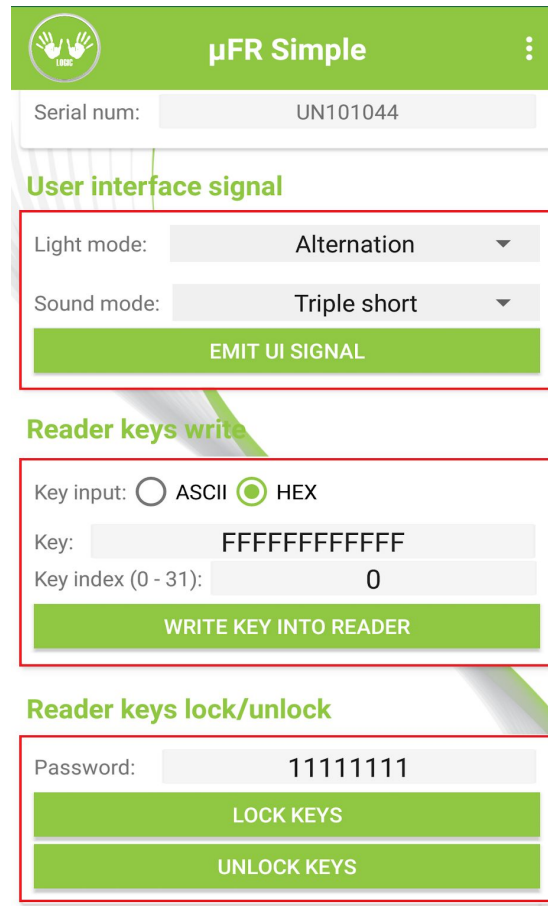
Establishing communication with μ FR Reader in Simple application is the same as in [Simplest](#) application.



The screenshot displays the **μFR Simple** application interface. At the top, there is a green header bar with the application name and a menu icon. Below the header, a green button labeled **READER OPEN** is visible, followed by a checkbox for **Use Advanced options**. The **Reader informations** section is highlighted with a red box and contains two input fields: **Reader type:** d1380022 and **Serial num:** UN101044. Below this, the **User interface signal** section includes dropdown menus for **Light mode:** (set to None) and **Sound mode:** (set to None), along with a green **EMIT UI SIGNAL** button. The **Reader keys write** section features radio buttons for **Key input:** (selected: ASCII, unselected: HEX), input fields for **Key:** and **Key index (0 - 31):**, and a green **WRITE KEY INTO READER** button. The **Reader keys lock/unlock** section is partially visible at the bottom.

If establishing communication was successful, you will be able to see readers type and serial number as shown in the picture above.

Reader options



µFR Simple

Serial num: UN101044

User interface signal

Light mode: Alternation

Sound mode: Triple short

EMIT UI SIGNAL

Reader keys write

Key input: ☐ ASCII ☒ HEX

Key: FFFFFFFFFF

Key index (0 - 31): 0

WRITE KEY INTO READER

Reader keys lock/unlock

Password: 11111111

LOCK KEYS

UNLOCK KEYS

- User interface signal

For emitting user interface signal, choose wanted **light and sound** mode from dropdown menu and click **EMIT UI SIGNAL** and you will be able to hear beeping and to see readers light signal.

- Reader key write

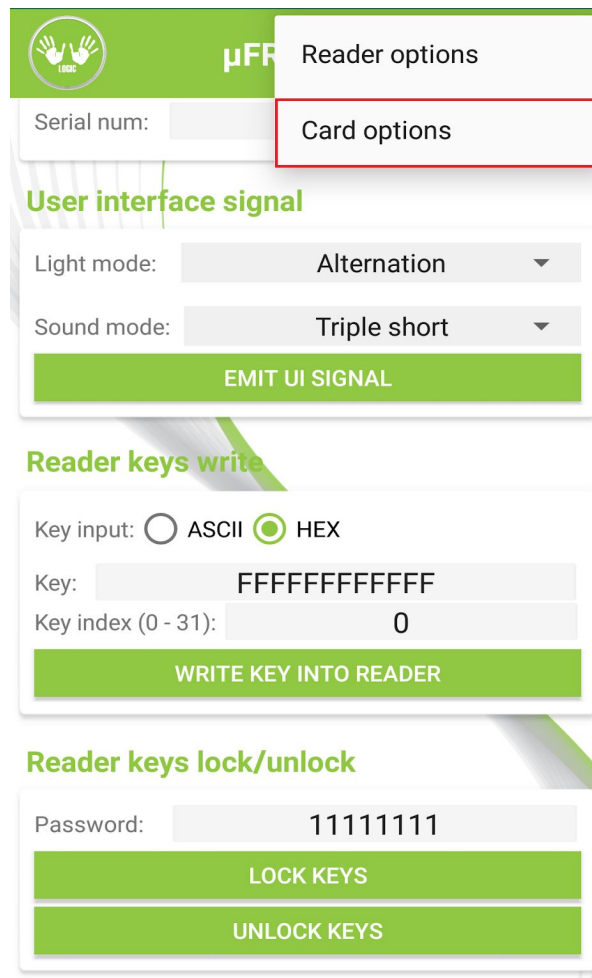
For writing key into reader, first, select **key input mode** (ASCII or HEX) then type in **key** you want to store into empty text field and type in **key index** for storing into reader, at the end, click **WRITE KEY INTO READER**.

- Reader key lock/unlock

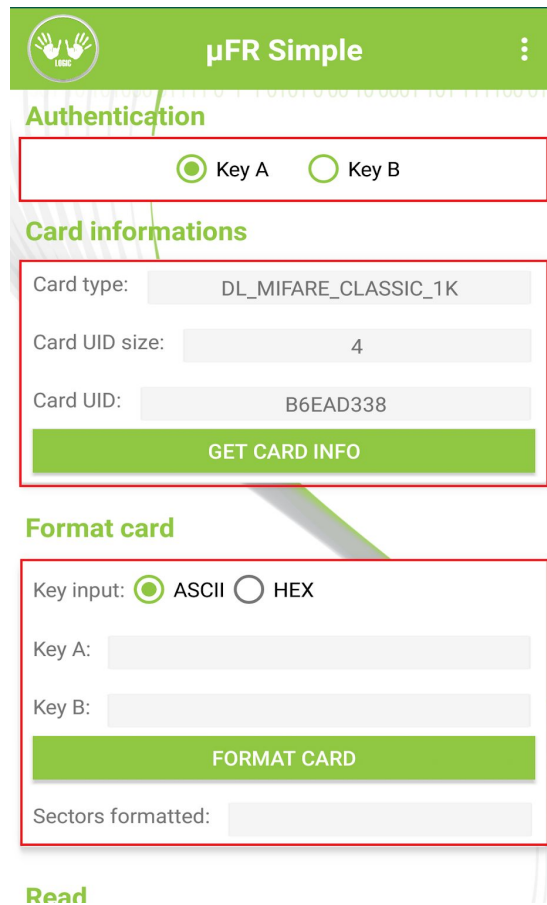
This option allows you to lock or unlock reader for writing keys into it. Type in **8 characters long password** and click **LOCK KEYS** or **UNLOCK KEYS** button.

Card options

For opening Card options, click on three white dots at the top right corner and select Card options



The screenshot shows the Digital Logic web interface. At the top, there is a green header with a logo and the text 'μFR'. Below the header, there is a 'Reader options' dropdown menu. The 'Card options' option is highlighted with a red box. Below the dropdown, there is a 'User interface signal' section with 'Light mode' set to 'Alternation' and 'Sound mode' set to 'Triple short'. There is a green button labeled 'EMIT UI SIGNAL'. Below that is a 'Reader keys write' section with 'Key input' set to 'HEX', a 'Key' field containing 'FFFFFFFFFFFF', and a 'Key index (0 - 31)' field containing '0'. There is a green button labeled 'WRITE KEY INTO READER'. At the bottom is a 'Reader keys lock/unlock' section with a 'Password' field containing '11111111'. There are two green buttons labeled 'LOCK KEYS' and 'UNLOCK KEYS'.



The screenshot shows the μFR Simple application interface. It has a green header with the Digital Logic logo and the title "μFR Simple". Below the header, there are three main sections: "Authentication", "Card informations", and "Format card". The "Authentication" section has two radio buttons for "Key A" (selected) and "Key B". The "Card informations" section displays "Card type: DL_MIFARE_CLASSIC_1K", "Card UID size: 4", and "Card UID: B6EAD338", with a green "GET CARD INFO" button below. The "Format card" section has "Key input: ASCII" (selected) and "HEX" radio buttons, input fields for "Key A:" and "Key B:", a green "FORMAT CARD" button, and a "Sectors formatted:" field. The "Read" section is partially visible at the bottom.

- Authentication

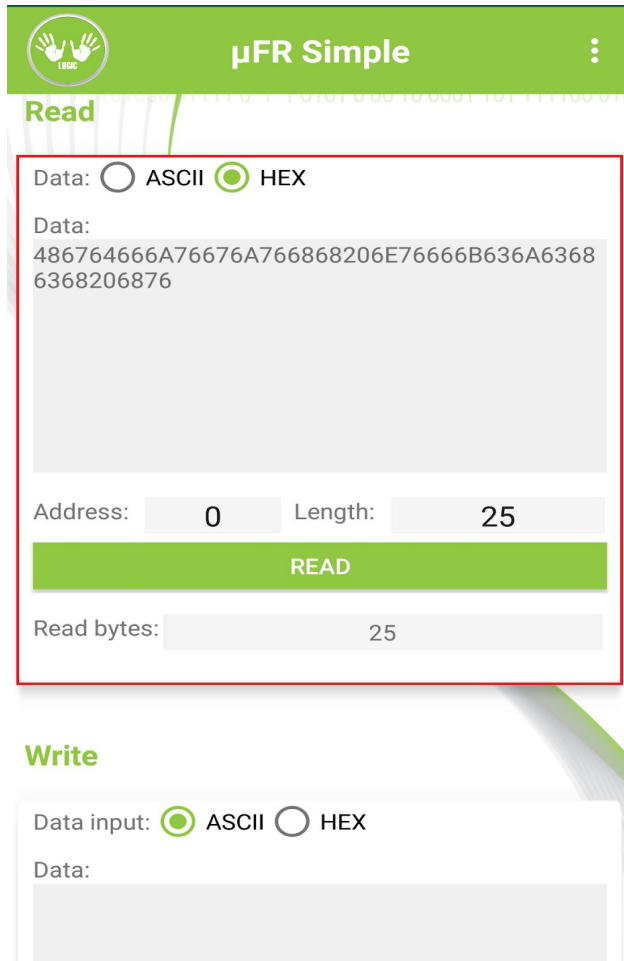
In this section you can switch authentication from **key A** to **key B** and vice versa.

- Card informations

Put card on the reader and simply click on **GET CARD INFO** button to see **cards type**, **UID** and **UID size**.

- Format card

You can use format card option to store **new key A** and **new key B** into all sectors in card.



μFR Simple

Read

Data: ☐ ASCII ☒ HEX

Data:
486764666A76676A766868206E76666B636A63686368206876

Address: Length:

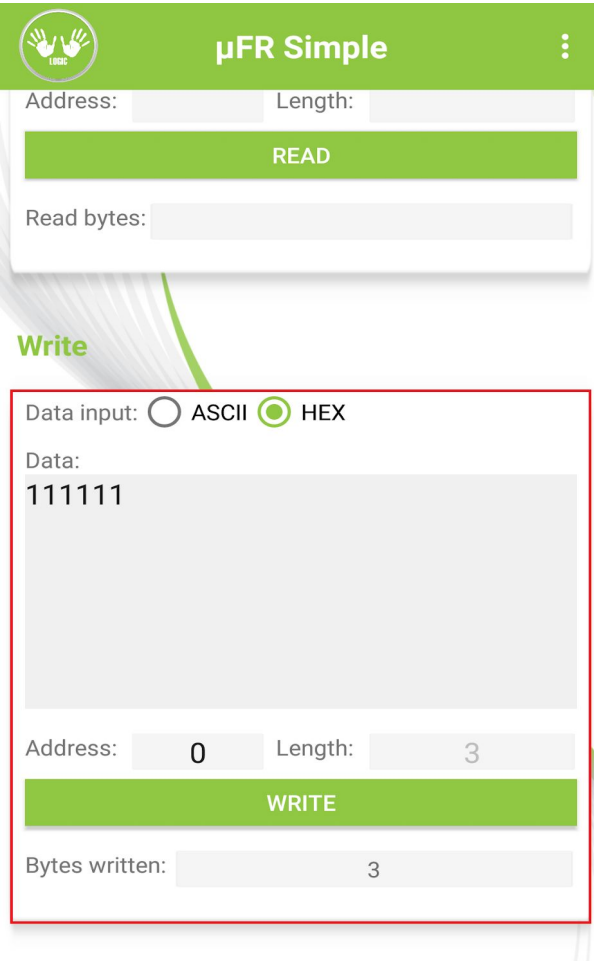
READ

Read bytes:

Write

Data input: ☒ ASCII ☐ HEX

Data:



μFR Simple

Address: Length:

READ

Read bytes:

Write

Data input: ☐ ASCII ☒ HEX

Data:
111111

Address: Length:

WRITE

Bytes written:

- Read

For reading, type in **address** (where to start reading) and how many bytes to read (into empty **length** field) and simply click **READ** button to see card data. Also, you can choose data representation switching between ASCII and HEX radio buttons.

- Write

For writing, select what kind of data you want to store (**ASCII** string or **HEX** string), type in data into empty text field, type in **address** (where to start writing) and simply click **WRITE** button, length will be automatically calculated.

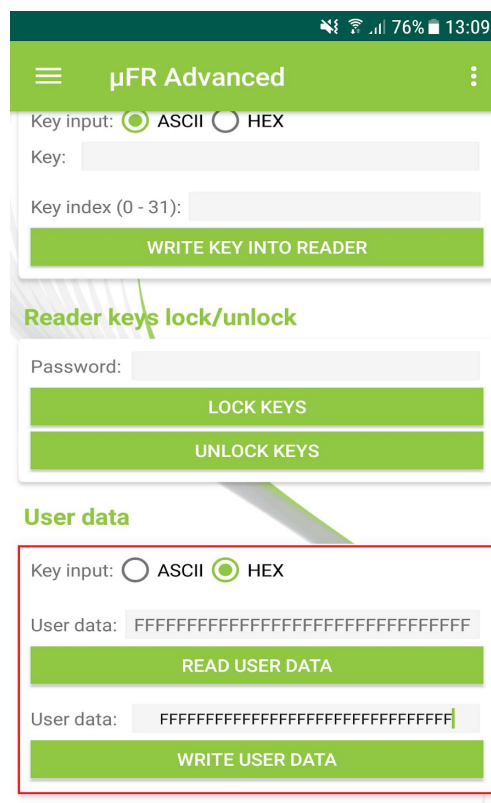
Advanced

Communication with μ FR Reader

Establishing communication with μ FR Reader in Advanced application is the same as in [Simplest](#) application.

Reader options

All reader options are mostly the same as in the [Simple](#) application except one extra option named **User data**. In User data option you will be able to read and write 16 bytes from reader EEPROM if you click on button **WRITE USER DATA**. See picture below:

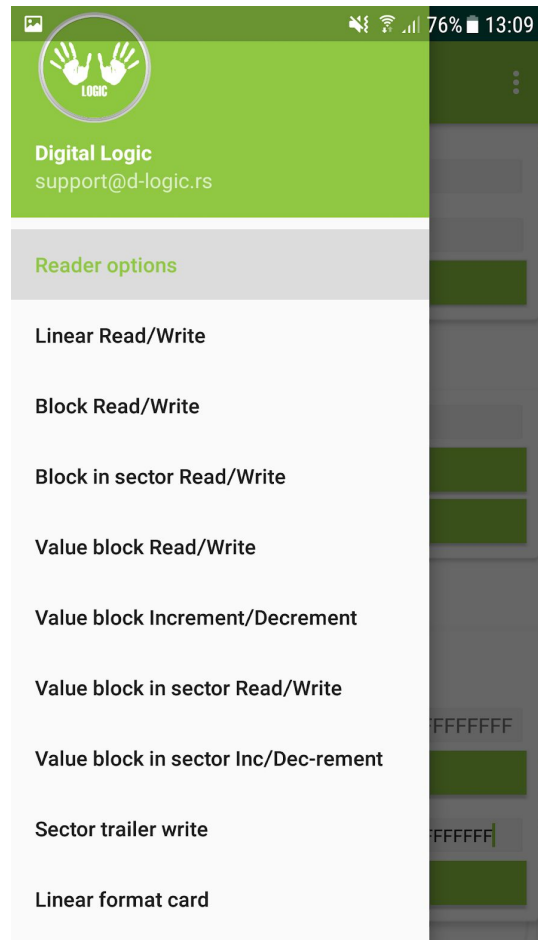


The screenshot shows the **μFR Advanced** application interface. It has a green header with a menu icon, the title **μFR Advanced**, and a settings icon. The interface is divided into three main sections:

- Key input:** Contains radio buttons for **ASCII** (selected) and **HEX**. Below are input fields for **Key:** and **Key index (0 - 31):**, followed by a green button labeled **WRITE KEY INTO READER**.
- Reader keys lock/unlock:** Contains a **Password:** input field, followed by green buttons for **LOCK KEYS** and **UNLOCK KEYS**.
- User data:** This section is highlighted with a red box. It contains radio buttons for **ASCII** and **HEX** (selected). Below are two **User data:** input fields, each followed by a green button labeled **READ USER DATA** and **WRITE USER DATA** respectively.

List of options

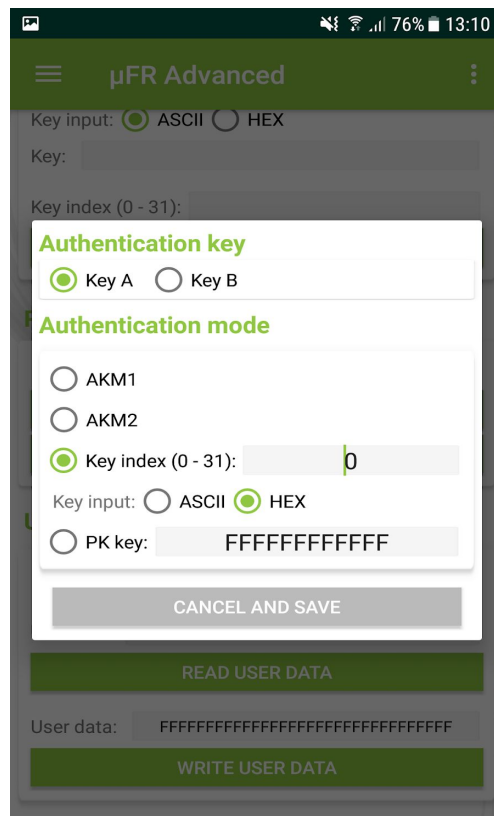
In Advanced application, there are multiple ways to read and write data from card, such as:



- Linear Read/Write - works the same as in the [Simple](#) application.
- Block and block in sector Read/Write
- Value block and value block in sector Read/Write/Increment/Decrement
- Sector trailer write
- Linear format card

Authentication

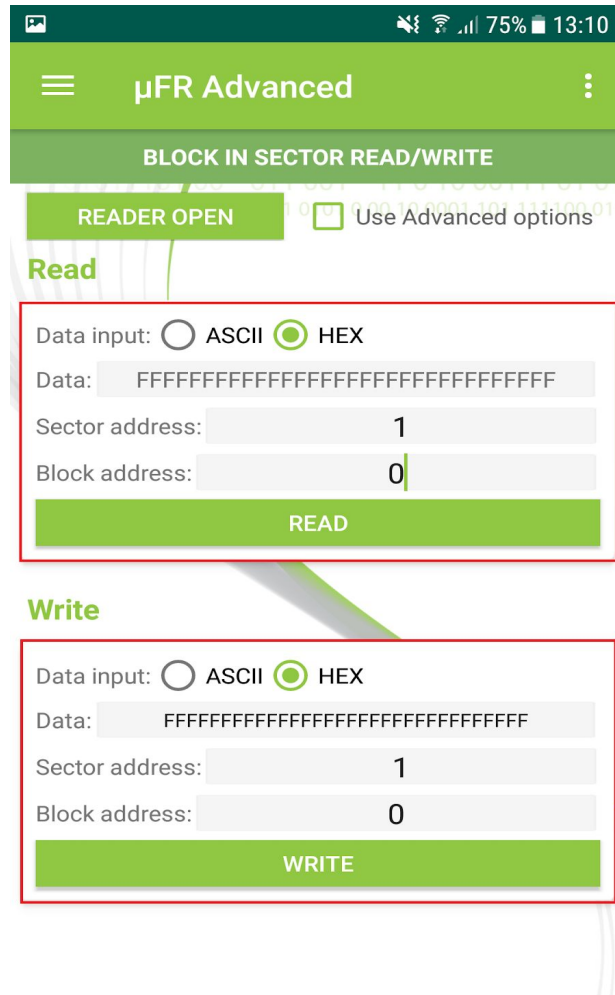
For authentication settings, click on three white dots in top right corner and select Authentication settings.



In this pop-up window you will be able to switch authentication key from A to B and vice versa. Also you can switch between authentication modes: Automatic key mode 1, Automatic key mode 2, Key index and Provided key. For explanations about these authentication refer to [μFR Series NFC Reader API](#).

Block and Block in sector Read/Write

These two options are very similar, the only difference is in Block in sector read and write options you have to provide both, sector and block address and in block read and write options, you have to provide only block address for read and write operations. Here is some example of block in sector reading and writing data:



μFR Advanced

BLOCK IN SECTOR READ/WRITE

READER OPEN ☐ Use Advanced options

Read

Data input: ☐ ASCII ☒ HEX

Data: FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

Sector address: 1

Block address: 0

READ

Write

Data input: ☐ ASCII ☒ HEX

Data: FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

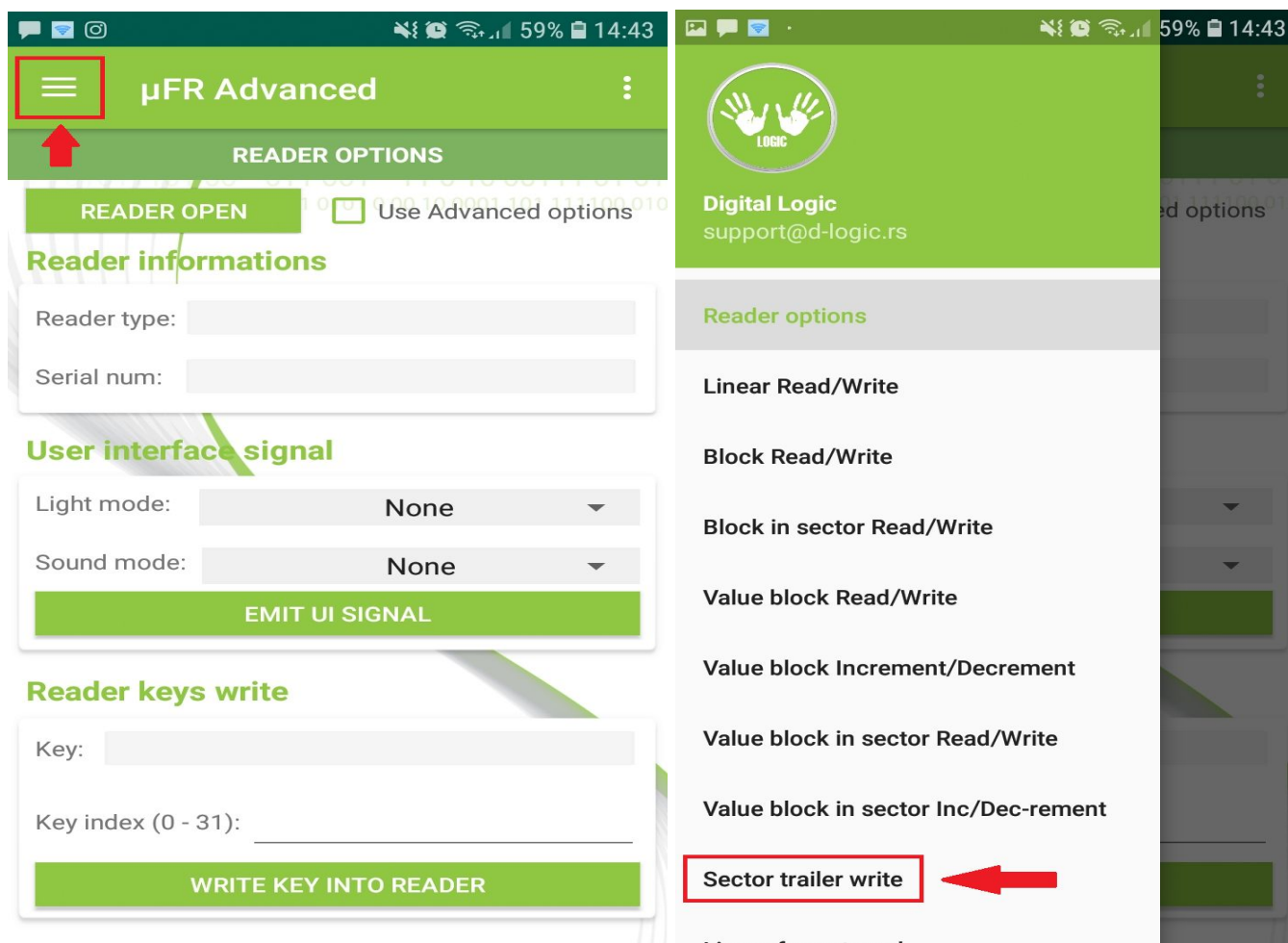
Sector address: 1

Block address: 0

WRITE

Value blocks

If you want to configure blocks as value blocks, you have to change blocks access bits. Click on side bar menu and then select 'Sector trailer write' option.



Sector trailer access bits:

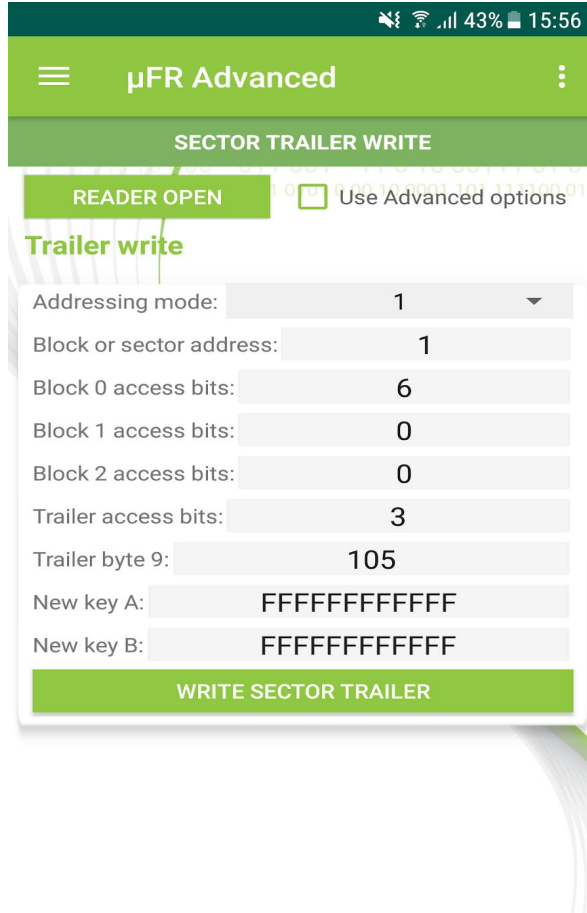
Access value arg.	Access bits			Access condition for						Remark
				KEYA		Access bits		KEYB		
	C1 ₃	C2 ₃	C3 ₃	read	write	read	write	read	write	
0	0	0	0	never	key A	key A	never	key A	key A	Key B may be read ^[1]
2	0	1	0	never	never	key A	never	key A	never	Key B may be read ^[1]
4	1	0	0	never	key B	key A B	never	never	key B	
6	1	1	0	never	never	key A B	never	never	never	
1	0	0	1	never	key A	key A	key A	key A	key A	Key B may be read, transport configuration ^[1]
3	0	1	1	never	key B	key A B	key B	never	key B	
5	1	0	1	never	never	key A B	key B	never	never	
7	1	1	1	never	never	key A B	never	never	never	

Block access bits:

Access value (to the function)	Access bits			Access condition for				Application
	C1	C2	C3	read	write	increment	decrement, transfer, restore	
0	0	0	0	key A B ¹	key A B ¹	key A B ¹	key A B ¹	transport configuration
2	0	1	0	key A B ¹	never	never	never	read/write block
4	1	0	0	key A B ¹	key B ¹	never	never	read/write block
6	1	1	0	key A B ¹	key B ¹	key B ¹	key A B ¹	value block
1	0	0	1	key A B ¹	never	never	key A B ¹	value block
3	0	1	1	key B ¹	key B ¹	never	never	read/write block
5	1	0	1	key B ¹	never	never	never	read/write block
7	1	1	1	never	never	never	never	read/write block

For configuring data blocks as value blocks, please refer to [μFR Series NFC Reader API](#)

For example, we will configure first block in first sector as 'VALUE' block.



Addressing mode:

0 - absolute sector trailer address

1 - relative sector trailer address

Block or sector address:

It depends on your addressing mode choice

Block 0 access bit:

We will set '6' for value block

Block 1 access bit:

We will set '0' and block 1 will stay 'data block'

Block 2 access bit:

We will set '0' and block 2 will stay 'data block'

Sector trailer access bit:

We will set '3'

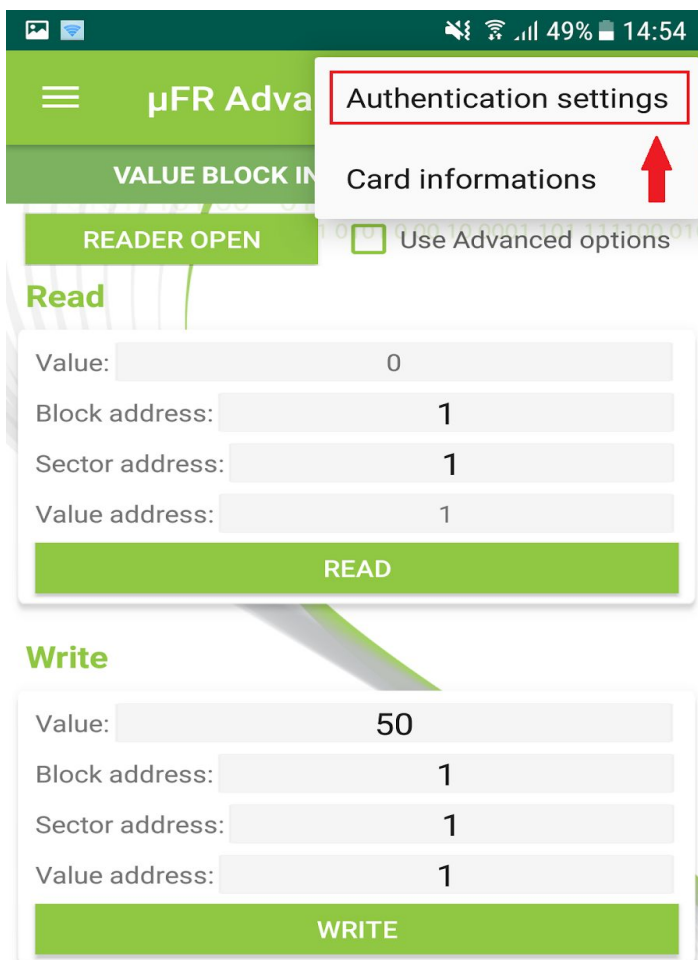
Sector trailer byte 9:

We will set default value 105 (0x69 hexadecimal)

We will use default key A and B (0xFFFFFFFFFFFF)

With this access bits configuration, you can read and decrement value blocks with both keys (A or B), but for writing and increment value you have to use key B. For switching authentication key to key B, follow these steps:

1. Click on three white dots at the top right corner
2. Choose 'Authentication settings option
3. Select 'Key B' as authentication key



μFR Advanced

Authentication settings

Card informations

READER OPEN

Use Advanced options

Read

Value: 0

Block address: 1

Sector address: 1

Value address: 1

READ

Write

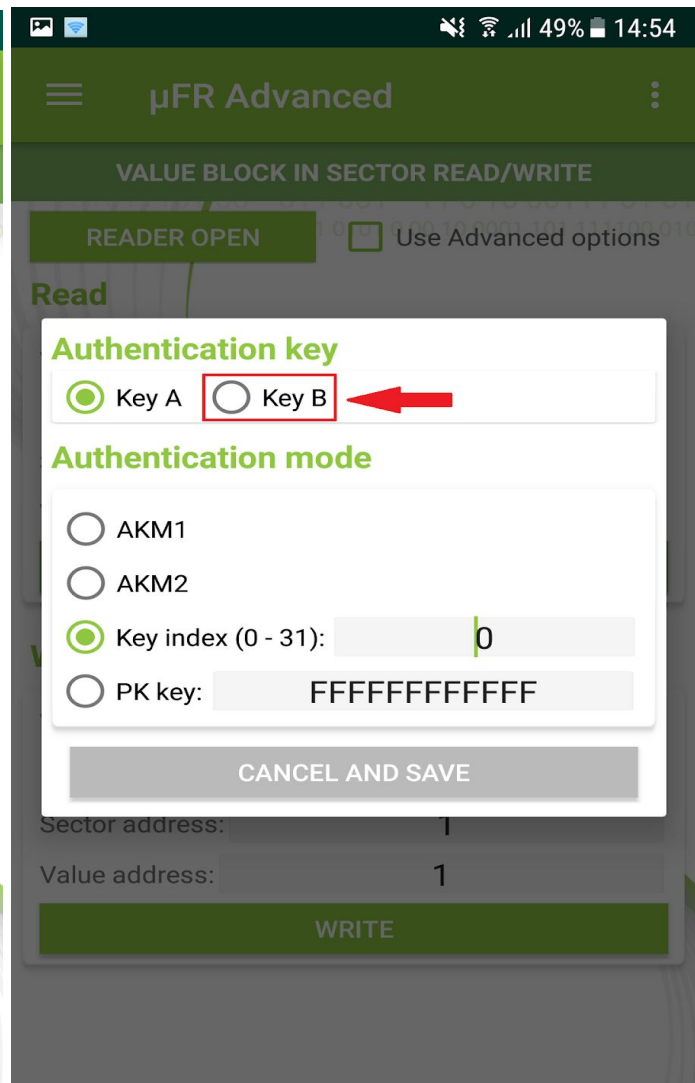
Value: 50

Block address: 1

Sector address: 1

Value address: 1

WRITE



μFR Advanced

Authentication key

☒ Key A ☐ Key B

Authentication mode

☐ AKM1

☐ AKM2

☒ Key index (0 - 31): 0

☐ PK key: FFFFFFFFFFFFFFFF


CANCEL AND SAVE

Sector address: 1

Value address: 1

WRITE

After successful configuration of value blocks you can, for example, choose 'Value block in sector Read/Write' or 'Value block in sector Increment/Decrement' option in sidebar menu:



μFR Advanced

VALUE BLOCK IN SECTOR READ/WRITE

READER OPEN ☐ Use Advanced options

Read

Value: 50

Block address: 1

Sector address: 1

Value address: 1

READ

Write


Value: 50

Block address: 1

Sector address: 1

Value address: 1

WRITE



μFR Advanced

VALUE BLOCK IN SECTOR INCREMENT/DECREMENT

READER OPEN ☐ Use Advanced options

Increment

Increment value: 20

Block address: 1

Sector address: 1

INCREMENT

Decrement

Decrement value: 70

Block address: 1

Sector address: 1

DECREMENT

Revision history

Date	Version	Comment
2019-20-09	1.0	Base document