

# MIFARE® EXAMPLES

Download link: [https://www.d-logic.net/code/nfc-rfid-reader-sdk/ufr-mf-examples-c\\_sharp.git](https://www.d-logic.net/code/nfc-rfid-reader-sdk/ufr-mf-examples-c_sharp.git)

You will see 3 different applications:

- Simplest - It reads card UID, card type, reads/writes linear data on card. It can also format card with transport keys (FF FF FF FF FF FF).
- Simple - same as simplest with added option of reader UI signals, reader type and serial number and card authentication option.
- Advance - same as simple with more authentication option, which will be explained further in the manual.

for work with MIFARE® cards.

## Simplest

The screenshot shows the 'uFr Simplest' application window. It features a menu bar with 'Exit', a 'Reader Open' button, and a checkbox for 'Use Advanced options'. A URL is provided: <http://www.d-logic.net/nfc-rfid-reader-sdk/>. The 'Advanced options' section (1) includes fields for 'Reader type', 'Port name', 'Port interface', and 'Arg'. The 'Card Type' (2) is set to '0x21' and 'UID Size' to '0x04'. The 'Card UID' is '0xAD69ECD7'. A 'FORMAT CARD' button (3) is visible. The 'Write Data' section (4) contains a text area with 'This is a simple test!' and a 'LINEAR WRITE' button. The 'Read Data' section (5) contains a text area with 'This is a simple test!' and a 'LINEAR READ' button. At the bottom, a status table (6) shows:

Function Error	0x00	DL_OK
CARD STATUS	0x00	DL_OK
CONNECTED	0x00	DL_OK

## 1. ReaderOpen and ReaderOpenEx

This part of our application is used for opening communication with our uFR series reader. We implemented 'Advanced options' in which you input arguments necessary for our ReaderOpenEx() function. All you need to do is check the 'Use Advanced options' and provide data in specified fields.

If use of Advanced options is not checked, our application will call standard ReaderOpen() function and will try to find reader connected to your PC.

## 2. Card type and UID

After putting a card on the reader you will be able to see card type, uid, and uid length in bytes, also **CARD\_STATUS** will be changed from **NO\_CARD** to **DL\_OK**.

## 3. Format card

If you click button 'FORMAT CARD', all data on the card will be erased and all sector keys will be set to **0xFFFFFFFFFFFF** - 6 bytes. Additionally, this function will try to overwrite card data with zeros with newly set keys.

## 4. Linear write

For example, we will write '**This is a simple test!**' - simple string. Just write text into text box as shown on the picture and click "LINEAR WRITE".

## 5. Linear read

If you click button 'LINEAR READ' you will be able to see all data on the card as shown on the picture. After clicking 'FORMAT CARD' and erasing all data from card we are now able to see text that we have written in the LINEAR READ text box.

## 6. Function status, card status, and reader status

Function error - result of functions such as Format, Read or Write will be displayed here.

Card status - Displays either DL\_OK or NO\_CARD. NO\_CARD means there is no card detected by the reader.

Reader status - will display DL\_OK if Reader Open function was successful.

# Simple

uFr Simple

Exit

Reader Open

Reader Open  Use Advanced options <http://www.d-logic.net/nfc-rfid-reader-sdk/>

**Advanced options 1**

Reader type:  Port name:  Port interface:  Arg:

**Reader Type 0xD1180022 2** Card Type **0x21** UID Size **0x04**

Reader Serial **0xA5010000** Card Serial **0x4EDE7C89**

Light Mode  Sound Mode  **3**

CONNECTED | 0x00 | DL\_OK

**4**  **AUTH 1A**  **AUTH 1B**

New Card Keys | New Reader Key

**5**

Key A

Key B

Hex  Sectors Formatted

Linear Read | Linear Write

Read Data **6**

FF:FF:FF:FF:FF:FF:FF:FF:FF:FF

Linear Address  Read Bytes

Data Length

Function Error: 0x00 | DL\_OK

CARD STATUS: 0x00 | DL\_OK

## 1. ReaderOpen and ReaderOpenEx

This part of our application is same as the one in our Simplest example, it's used for opening communication with our uFR series reader.

We implemented 'Advanced options' in which you input arguments necessary for our ReaderOpenEx() function. All you need to do is check the 'Use Advanced options' and provide data in specified fields.

If use of Advanced options is not checked, our application will call standard ReaderOpen() function and will try to find reader connected to your PC.

## 2. Reader type, reader serial, card type and UID

After opening application and putting card on the reader you will see reader type and serial number, card type, card uid and card uid size represented as hexadecimal numbers as shown on the picture.

## 3. Reader signalization

You can choose reader light and sound mode from combo boxes and after clicking 'READER UI SIGNAL' signalization will be visible and you can hear sound from speaker.

## 4. Authentication mode

You can choose between authentication with key A or authentication with key B by clicking on radio boxes in section number '3'.

## 5. Format

- By clicking 'FORMAT CARD' which is on 'New Card Keys' tab, all data on the card will be erased and all sector keys will be set to **0xFFFFFFFFFFFF** - 6 bytes. Additionally it will try to write all zeros to the card with same authentication mode and settings with which the card was formatted.
- If you click on tab 'New Reader Key' you will be able to see button 'FORMAT READER KEYS' which will, after clicking, set entered reader key **0xFFFFFFFFFFFF** - 6 bytes, key under provided index parameter in EEPROM

## 6. Linear read and Write

- On 'Linear Read' tab you have to choose linear address (where to start reading) and data length (how many bytes to read) and click 'LINEAR READ' button. After clicking you will be able to see card data in text box.

- Data can be read/written based on data format selected via radio buttons above, either ASCII or Hex.
- On 'Linear Write' tab, you have to choose linear address (where to start writing) and input data into text box as shown in the picture below:

Linear read/write data format:

ASCII  Hex

Linear Read Linear Write

Write Data

FFFFFFFFFFFFFFFFFFFF

Linear Address

Data Length  Bytes Written

LINEAR WRITE

Function Error:	0x00	DL_OK
CARD STATUS:	0x00	DL_OK

Linear read/write data format:

ASCII  Hex

Linear Read Linear Write

Write Data

This is a simple test!

Linear Address

Data Length  Bytes Written

LINEAR WRITE

Function Error:	0x00	DL_OK
CARD STATUS:	0x00	DL_OK

Data length will be automatically calculated as you type based on input type (ASCII/Hex). Bytes written shows a number of bytes that are written into card after clicking 'LINEAR WRITE' button.

## Advanced

The screenshot displays the 'uFr Advanced' software interface. The window title is 'uFr Advanced'. The main menu includes 'Functions' and 'View All'. The 'Reader Open' section contains a 'Reader Open' button, a checkbox for 'Use Advanced options', and a URL: <http://www.d-logic.net/nfc-rfid-reader-sdk/>. The 'Advanced options' section is highlighted with a red border and includes fields for 'Reader type', 'Port name', 'Port interface', and 'Arg'. The 'Reader Type' section, highlighted with a dark red border, shows 'Reader Type' as 0xD1380022, 'Card Type' as 0x21, 'UID Size' as 0x4, 'Reader Serial' as 0x18AC0, and 'Card Serial' as 0xAD69ECD7. The 'Light Mode' and 'Sound Mode' sections, highlighted with a green border, show both set to 'None'. The 'Reader Key' section, highlighted with a blue border, shows 'Write User Data' selected, 'Key Index' as 0, and a 'Reader Key Write' button. The 'AUTH' section, highlighted with an orange border, shows 'AUTH 1A' selected, 'Key Index' as 0, and 'PK Key' as 255. The 'Linear read/write data format' section, highlighted with a purple border, shows 'ASCII' selected. The 'Linear Read' section shows 'LinearRead\_AKM1/AKM2' selected, and the 'Read Data' field contains the text 'This is a simple test!'. The 'Linear Address' is 0, 'Data Length' is 30, and 'Read Bytes' is 30. The 'READ' button is visible. The status bar at the bottom shows 'CONNECTED' and 'DL OK'.

uFr Advanced

Functions View All

Reader Open

Reader Open  Use Advanced options <http://www.d-logic.net/nfc-rfid-reader-sdk/>

Advanced options

Reader type:  Port name:  Port interface:  Arg:

Reader Type  Card Type  UID Size

Reader Serial  Card Serial

Light Mode  Reader UI Signal Reader Reset

Sound Mode  Soft Restart

Reader Key Write User Data

Key Index  Reader Key Write

Hex

CONNECTED 0x00 DL OK

AUTH 1A  AUTH 1B Key Index  PK Key

Linear read/write data format:

ASCII  Hex

Linear Read LinearRead\_AKM1/AKM2 LinearRead\_PK Linear Write LinearWrite\_AK

Read Data

This is a simple test!

Linear Address  Read Bytes  READ

Data Length

CARD STATUS 0x00 DL\_OK

## **1. ReaderOpen and ReaderOpenEx**

This part of our application is same as the one in our Simplest example, it's used for opening communication with our uFR series reader.

We implemented 'Advanced options' in which you input arguments necessary for our ReaderOpenEx() function. All you need to do is check the 'Use Advanced options' and provide data in specified fields.

If use of Advanced options is not checked, our application will call standard ReaderOpen() function and will try to find reader connected to your PC.

## **2. Reader type, reader serial, card type and UID**

After opening application and putting card on the reader you will see reader type and serial number, card type, card uid and card uid size represented as hexadecimal numbers as shown on the picture above.

## **3. Reader signalization and restart**

You can choose reader light and sound mode from combo boxes and after clicking 'READER UI SIGNAL' signalization will be visible and you can hear sound from speaker.

Clicking 'Reader Reset' will cause physical reset of reader communication port.

If you click on the button 'Soft Restart' it will restart the reader by software. It sets all readers parameters to default values and close RF field which resets all the cards in the field.

## **4. Reader keys and data**

- In the 'Reader Key' tab you can see button 'Key Index' combo box in which you can choose between 0 - 31 key number to write into reader by clicking 'Reader Key Write' button.
- In the 'Write User Data' tab you can see text box with caption 'New User Data' in which you can type new user data and write it into reader by clicking 'Write User Data' button.

## **5. Authentication mode**

You can choose between authentication with key A or authentication with key B by clicking on radio boxes in section number '4'.

Also, you can choose key index from combobox or enter Provided key (PK) - 6 bytes. They will be used depending on which tab you click in section '5' for linear reading and writing cards data.

- Linear Read - using Key Index
- LinearRead\_AKM1/AKM2 - using auth mode (AUTH 1A or AUTH 1B)
- LinearRead\_PK - using Provided key
- Linear Write - using Key Index
- LinearWrite\_AKM1/AKM2 - using auth mode (AUTH1A or AUTH1B)
- LinearWrite\_PK - using Provided key

## 6. Functions:

If you click on “Functions” at the top of the application, you will see dropdown list with more options for work.



### 6.1 Linear read / Linear write

### 6.2 Block read / Block write

### 6.3 Block in sector read / Block in sector write

### 6.4 Value block read / Value block write

### 6.5 Value block increment / Value block decrement

### 6.6 Value block in sector read / Value block in sector write

### 6.7 Value block in sector increment / Value block in sector decrement

### 6.8 Sector trailer write

### 6.9 Linear format card

## 6.1 Linear read / Linear write

### Reading:

- On 'Linear Read' tab you have to choose key index and authentication mode (AUTH 1A or AUTH 1B) for reading which is in section '4', linear address (where to start reading) and data length (how many bytes to read) and click 'READ' button. After clicking you will be able to see card data in text box.
- On 'LinearRead\_AKM1/AKM2' tab you have to choose authentication mode (AUTH 1A or AUTH 1B) for reading which is in section '4', linear address (where to start reading) and data length (how many bytes to read) and click 'READ' button. After clicking you will be able to see card data in text box.



- On 'LinearRead\_PK' tab you have to enter Provided key (6 bytes - 0xFFFFFFFFFFFF is default hex or 255 255 255 255 255 255 decimal) for reading which is in section '4', linear address (where to start reading) and data length (how many bytes to read) and click 'READ' button. After clicking you will be able to see card data in text box.

Writing:

- On 'Linear Write' tab, you have to choose key index and authentication mode (AUTH 1A or AUTH 1B) for writing which is in section '4', linear address (where to start writing) and input data into text box as shown in the picture below:

The screenshot shows a software window titled 'Linear read/write data format:'. At the top, there are radio buttons for 'ASCII' (selected) and 'Hex'. Below this is a tabbed interface with tabs for 'Linear Read', 'LinearRead\_AKM1/AKM2', 'LinearRead\_PK', 'Linear Write' (selected), and 'LinearWrite\_AK'. The 'Linear Write' tab contains a 'Write Data' section with a text box containing 'This is a simple test!'. Below the text box are input fields for 'Linear Address' (0) and 'Data Length' (22). To the right of these is a 'Bytes Written' field showing 22 and a 'WRITE' button. At the bottom of the window, there is a 'Function Error' field showing 0x00 and a 'DL\_OK' button.

Data length will be automatically calculated. Bytes written shows a number of bytes that are written into card after clicking 'WRITE' button.

- On 'LinearWrite\_AKM1/AKM2' tab, you have to choose authentication mode (AUTH 1A or AUTH 1B) for writing which is in section '4', linear address (where to start writing) and input data into text box. Data length will be automatically calculated. Bytes written shows a number of bytes that are written into card after clicking 'WRITE' button.
- On 'LinearWrite\_PK' tab, you have to enter Provided key (6 bytes - 0xFFFFFFFFFFFF is default hex or 255 255 255 255 255 255 decimal) for writing which is in section '4', linear

address (where to start writing) and input data into text box. Data length will be automatically calculated based on input type (ASCII/Hex). Bytes written shows a number of bytes that are written into card after clicking 'WRITE' button.

## 6.2 Block read / Block write

Now, we will explain Block Read/Write option.

**Block Read/Write (AKM1,AKM2,PK)**

AUTH 1A  AUTH 1B Key Index 0 PK Key 255 255 255 255 255 255

Block Read Block Write

**Block Read**

Block Address 0 READ

Read Data  Hex  
44CA0362EF080400015E31AA7

**BlockRead\_AKM1**

Block Address 0 READ

Read Data  Hex  
44CA0362EF080400015E31AA7

**BlockRead\_AKM2**

Block Address 0 READ

Read Data  Hex  
44CA0362EF080400015E31AA7

**BlockRead\_PK**

Block Address 0 READ

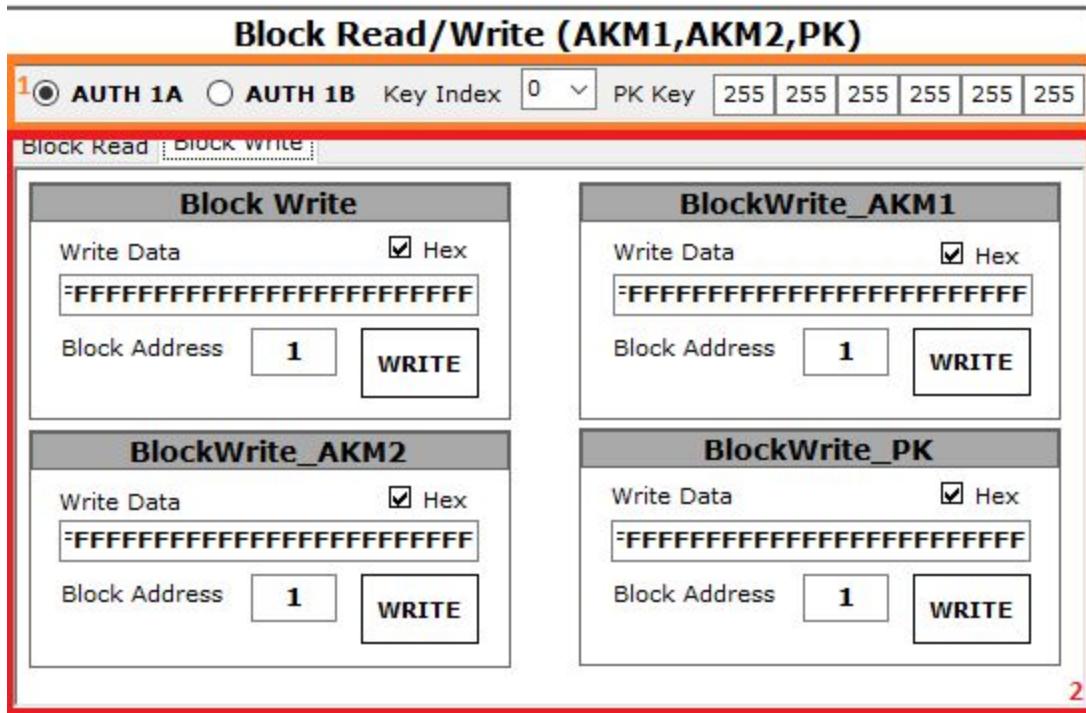
Read Data  Hex  
44CA0362EF080400015E31AA7

### Block read:

- In "Block Read" window you can choose block address and ASCII or hex data representation by checking 'Hex' checkbox. For successful reading, you have to choose appropriate key index and authentication mode (AUTH 1A or AUTH 1B) from section '1'. Block 0 data is shown at the picture above.
- In "BlockRead\_AKM1" window you can choose block address and ASCII or hex data representation by checking 'Hex' checkbox. For successful reading, you have to choose authentication mode (AUTH 1A or AUTH 1B) from section '1'. Block 0 data is shown at the picture above.
- In "BlockRead\_AKM2" window you can choose block address and ASCII or hex data representation by checking 'Hex' checkbox. For successful reading, you have to choose

authentication mode (AUTH 1A or AUTH 1B) from section '1'. Block 0 data is shown at the picture above.

- In "BlockRead\_PK" window you can choose block address and ASCII or hex data representation by checking 'Hex' checkbox. For successful reading, you have to enter Provided key (6 bytes - 0xFFFFFFFFFFFF is default hex or 255 255 255 255 255 255 decimal) which is in section '1'. Block 0 data is shown at the picture above.



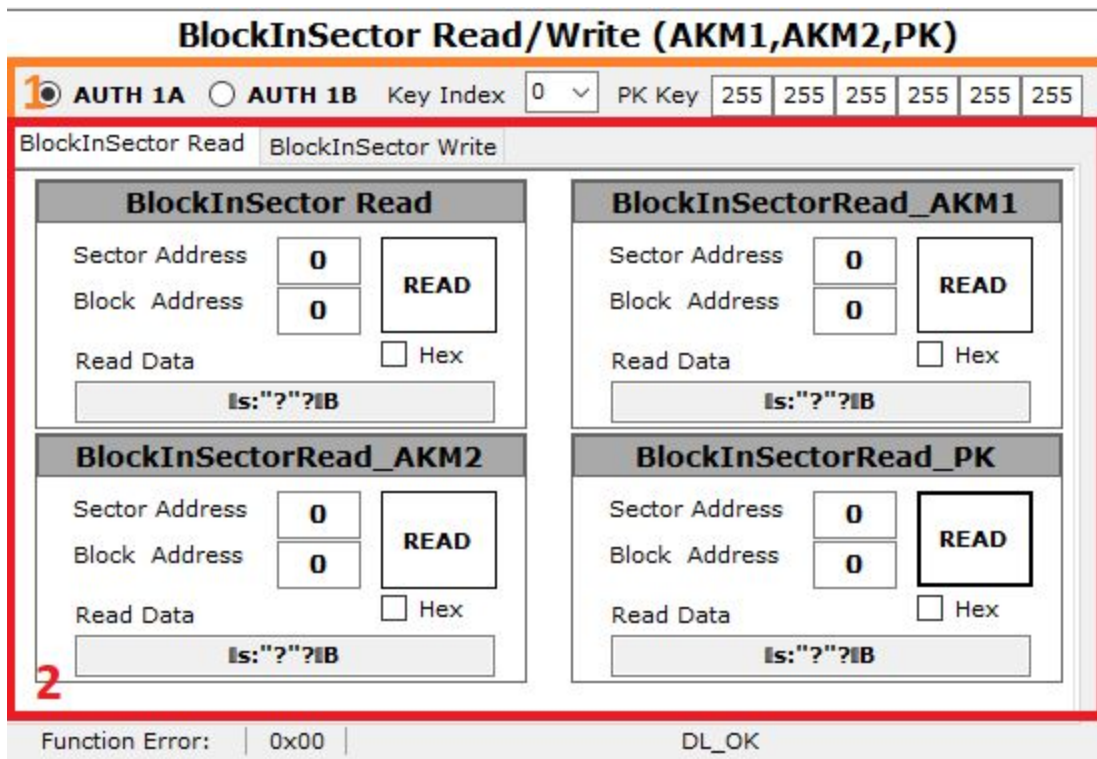
#### Block write:

- In "Block Write" window you can choose block address and ASCII or hex data input by checking 'Hex' checkbox. For successful writing, you have to enter 16 bytes of data in textbox, choose appropriate key index and authentication mode (AUTH 1A or AUTH 1B) from section '1' and then click "WRITE" button. Block 1 data writing is shown at the picture above.
- In "BlockWrite\_AKM1" window you can choose block address and ASCII or hex data input by checking 'Hex' checkbox. For successful writing, you have to enter 16 bytes of data in textbox, choose authentication mode (AUTH 1A or AUTH 1B) from section '1' and then click "WRITE" button. Block 1 data writing is shown at the picture above.
- In "BlockWrite\_AKM2" window you can choose block address and ASCII or hex data input by checking 'Hex' checkbox. For successful writing, you have to enter 16 bytes of

data in textbox, choose authentication mode (AUTH 1A or AUTH 1B) from section '1' and then click "WRITE" button. Block 1 data writing is shown at the picture above.

- In "BlockWrite\_PK" window you can choose block address and ASCII or hex data input by checking 'Hex' checkbox. For successful writing, you have to enter 16 bytes of data in textbox, Provided key (6 bytes - 0xFFFFFFFFFFFF is default hex or 255 255 255 255 255 255 decimal) which is in section '1' and then click "WRITE" button. Block 1 data writing is shown at the picture above.

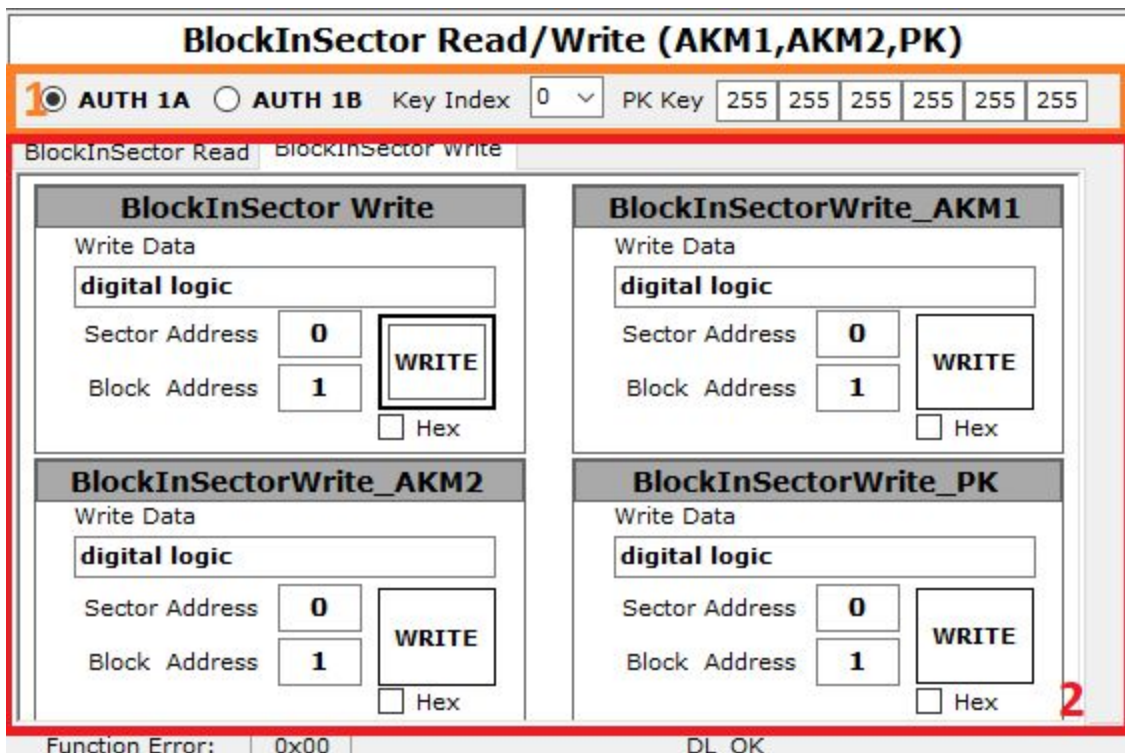
### 6.3 Block in sector read / Block in sector write



#### Block in sector read:

- In "BlockInSector Read" window you can choose sector address and block address. For successful reading, you have to choose appropriate key index and authentication mode (AUTH 1A or AUTH 1B) from section '1'. Block 0 in sector 0 data is shown at the picture above.
- In "BlockInSectorRead\_AKM1" window you can choose sector address and block address. For successful reading, you have to choose authentication mode (AUTH 1A or AUTH 1B) from section '1'. Block 0 in sector 0 data is shown at the picture above.

- In “BlockInSectorRead\_AKM2” ’window you can choose sector address and block address. For successful reading,you have to choose authentication mode (AUTH 1A or AUTH 1B) from section ‘1’. Block 0 in sector 0 data is shown at the picture above.
- In “BlockInSectorRead\_PK” ’window you can choose sector address and block address. For successful reading,you have to enter Provided key (6 bytes - 0xFFFFFFFFFFFF is default hex or 255 255 255 255 255 255 decimal) which is in section ‘1’. Block 0 in sector 0 data is shown at the picture above.



Block in sector write:

- In “BlockInSector Write’ window you can choose sector address and block address and enter ASCII data For successful writing, you have to choose appropriate key index and authentication mode (AUTH 1A or AUTH 1B) from section ‘1’. Block 1 in sector 0 data writing is shown at the picture above.
- In “BlockInSectorWrite\_AKM1” window you can choose sector address and block address and enter ASCII data. For successful writing,you have to choose authentication mode (AUTH 1A or AUTH 1B) from section ‘1’. Block 1 in sector 0 data writing is shown at the picture above.

- In “BlockInSectorWrite\_AKM2” window you can choose sector address and block address and enter ASCII data. For successful writing,you have to choose authentication mode (AUTH 1A or AUTH 1B) from section ‘1’. Block 1 in sector 0 data writing is shown at the picture above.
- In “BlockInSectorWrite\_PK” window you can choose sector address and block address and enter ASCII data. For successful writing,you have to enter Provided key (6 bytes - 0xFFFFFFFF is default hex or 255 255 255 255 255 255 decimal) which is in section ‘1’. Block 1 in sector 0 data writing is shown at the picture above.

### VALUE BLOCKS

If you want to configure blocks for value, you have to change blocks access bits. Click on “Functions” dropdown list at the top of the application and then select “Linear Format Card” option.

For configuring blocks as value blocks please refer to:  
<https://www.d-logic.net/code/nfc-rfid-reader-sdk/ufr-doc/blob/master/uFR%20Series%20NFC%20Reader%20API.pdf>

and look for “block access bits” and “sector trailer access bits”.

#### 6.4 Value block read / Value block write

The screenshot displays the 'ValueBlock Read/Write(AKM1,AKM2,PK)' interface. At the top, there are radio buttons for 'AUTH 1A' (selected) and 'AUTH 1B', a 'Key Index' dropdown set to '0', and a 'PK Key' field containing six '255' values. Below this, there are two tabs: 'valueBlock Read' (selected) and 'ValueBlock Write'. The main area contains four sub-windows, each with a 'Block Address' field set to '1' and a 'READ' button. The 'Read Value' field in each window displays '0', and the 'Value Address' field displays '15'. The sub-windows are titled 'ValueBlock Read', 'ValueBlockRead\_AKM1', 'ValueBlockRead\_AKM2', and 'ValueBlockRead\_PK'. A red box highlights the entire interface, with a '1' in the top-left corner and a '2' in the bottom-right corner.

Value block read:

- In “ValueBlock Read” window you have to choose block address, appropriate key index and authentication mode (AUTH 1A or AUTH 1B) which are in section ‘1’, then click “READ” button. Block 1 value reading is shown above
- In “ValueBlockRead\_AKM1” window you have to choose block address and authentication mode (AUTH 1A or AUTH 1B) which is in section ‘1’, then click “READ” button. Block 1 value reading is shown above
- In “ValueBlockRead\_AKM2” window you have to choose block address and authentication mode (AUTH 1A or AUTH 1B) which is in section ‘1’, then click “READ” button. Block 1 value reading is shown above.
- In “ValueBlockRead\_PK” window you have to choose block address and you have to enter Provided key (6 bytes - 0xFFFFFFFFFFFF is default hex or 255 255 255 255 255

255 decimal) which is in section '1', then click "READ" button. Block 1 value reading is shown above.

The screenshot shows a software interface titled "ValueBlock Read/Write(AKM1,AKM2,PK)". At the top, there are radio buttons for "AUTH 1A" and "AUTH 1B" (selected), a "Key Index" dropdown set to "0", and "PK Key" fields with values "255", "255", "255", "255", "255", "255". Below this are two tabs: "ValueBlock Read" and "ValueBlock Write" (selected). The main area contains four panels:

- ValueBlock Write:** Write Value: 10, Value Address: 15, Block Address: 1, WRITE button.
- ValueBlockWrite\_AKM1:** Write Value: 10, Value Address: 15, Block Address: 1, WRITE button.
- ValueBlockWrite\_AKM2:** Write Value: 10, Value Address: 15, Block Address: 1, WRITE button.
- ValueBlockWrite\_PK:** Block Address: 10, Value Address: 15, Write Value: 1, WRITE button.

Value block write:

- In "ValueBlock Write" you have to enter value, value address, block address and choose appropriate key index and authentication mode (AUTH 1A or AUTH 1B) which are in section '1', then click "WRITE" button. Block 1 value writing is shown above.
- In "ValueBlockWrite\_AKM1" you have to enter value, value address, block address and choose authentication mode (AUTH 1A or AUTH 1B) which is in section '1', then click "WRITE" button. Block 1 value writing is shown above.
- In "ValueBlockWrite\_AKM2" you have to enter value, value address, block address and choose authentication mode (AUTH 1A or AUTH 1B) which is in section '1', then click "WRITE" button. Block 1 value writing is shown above.
- In "ValueBlockWrite\_PK" you have to enter value, value address, block address and you have to enter Provided key (6 bytes - 0xFFFFFFFFFFFF is default hex or 255 255 255



255 255 255 decimal) which is in section '1', then click "WRITE" button. Block 1 value writing is shown above.

## 6.5 Value block increment / Value block decrement

Value block increment:

The screenshot shows a software interface titled "ValueBlock Increment/Decrement (AKM1,AKM2,PK)". At the top, there are radio buttons for "AUTH 1A" and "AUTH 1B" (selected), a "Key Index" dropdown set to "0", and a "PK Key" field with six "255" entries. Below this, there are two tabs: "ValueBlock Increment" (selected) and "ValueBlock Decrement". The main area contains four panels, each with a title, two input fields, and an "INCREMENT" button:

- ValueBlock Increment:** Increment Value: 10, Block Address: 1, INCREMENT button.
- ValueBlock Increment AKM1:** Increment Value: 10, Block Address: 1, INCREMENT button.
- ValueBlock Increment AKM2:** Increment Value: 10, Block Address: 1, INCREMENT button.
- ValueBlock Increment PK:** Increment Value: 10, Block Address: 1, INCREMENT button.

- In "ValueBlock Increment" window you have to enter increment value, block address, and choose appropriate key index and authentication mode (AUTH 1A or AUTH 1B) which are in section '1', then click "INCREMENT" button. Block 1 value incrementing is shown above.
- In "ValueBlock Increment AKM1" window you have to enter increment value, block address and choose authentication mode (AUTH 1A or AUTH 1B) which is in section '1', then click "INCREMENT" button. Block 1 value incrementing is shown above.
- In "ValueBlock Increment AKM2" window you have to enter increment value, block address and choose authentication mode (AUTH 1A or AUTH 1B) which is in section '1', then click "INCREMENT" button. Block 1 value incrementing is shown above.

- In “ValueBlock Increment PK” window you have to enter increment value, block address and you have to enter Provided key (6 bytes - 0xFFFFFFFFFFFF is default hex or 255 255 255 255 255 decimal) which is in section ‘1’, then click “INCREMENT” button. Block 1 value incrementing is shown above.

Value block decrement:

- In “ValueBlock Decrement” window you have to enter decrement value, block address, and choose appropriate key index and authentication mode (AUTH 1A or AUTH 1B) which are in section ‘1’, then click “DECREMENT” button. Block 1 value decrementing is shown above.
- In “ValueBlock Decrement AKM1” window you have to enter decrement value, block address and choose authentication mode (AUTH 1A or AUTH 1B) which is in section ‘1’, then click “DECREMENT” button. Block 1 value decrementing is shown above.
- In “ValueBlock Decrement AKM2” window you have to enter decrement value, block address and choose authentication mode (AUTH 1A or AUTH 1B) which is in section ‘1’, then click “DECREMENT” button. Block 1 value decrementing is shown above.
- In “ValueBlock Decrement PK” window you have to enter decrement value, block address and you have to enter Provided key (6 bytes - 0xFFFFFFFFFFFF is default hex

or 255 255 255 255 255 255 decimal) which is in section '1', then click "INCREMENT" button. Block 1 value decrementing is shown above.

## 6.7 Value block in sector increment / Value block in sector decrement

Value block in sector increment:

ValueBlockInSector Increment/Decrement(AKM1,AKM2,PK)							
<input checked="" type="radio"/> AUTH 1A <input type="radio"/> AUTH 1B              Key Index 0 ▾              PK Key 255 255 255 255 255 255							
ValueBlockInSector Increment				ValueBlockInSector Decrement			
<b>ValueBlockInSector Increment</b> Increment Value <input type="text" value="10"/> Sector Address <input type="text" value="0"/> <input type="text" value="INCREMENT"/> Block Address <input type="text" value="1"/>				<b>ValueBlockInSectorIncrementAKM1</b> Increment Value <input type="text" value="10"/> Sector Address <input type="text" value="0"/> <input type="text" value="INCREMENT"/> Block Address <input type="text" value="1"/>			
<b>ValueBlockInSectorIncrementAKM2</b> Increment Value <input type="text" value="10"/> Sector Address <input type="text" value="0"/> <input type="text" value="INCREMENT"/> Block Address <input type="text" value="1"/>				<b>ValueBlockInSectorIncrementPK</b> Increment Value <input type="text" value="10"/> Sector Address <input type="text" value="0"/> <input type="text" value="INCREMENT"/> Block Address <input type="text" value="1"/>			

- In "ValueBlockInSector Increment" window you have to enter increment value, sector address, block address and choose appropriate key index and authentication mode (AUTH 1A or AUTH 1B) which are in section '1', then click "INCREMENT" button. Block 1 in sector 0 value incrementing is shown above.
- In "ValueBlockInSector Increment AKM1" window you have to enter increment value, sector address, block address and choose authentication mode (AUTH 1A or AUTH 1B) which is in section '1', then click "INCREMENT" button. Block 1 in sector 0 value incrementing is shown above.
- In "ValueBlockInSector Increment AKM2" window you have to enter increment value, sector address, block address and choose authentication mode (AUTH 1A or AUTH 1B) which is in section '1', then click "INCREMENT" button. Block 1 in sector 0 value incrementing is shown above.

- In “ValueBlockInSector Increment PK” window you have to enter increment value, sector address, block address you have to enter Provided key (6 bytes - 0xFFFFFFFF is default hex or 255 255 255 255 255 255 decimal) which is in section ‘1’, then click “INCREMENT” button. Block 1 in sector 0 value incrementing is shown above.

Value block in sector decrement:

**ValueBlockInSector Increment/Decrement(AKM1,AKM2,PK)**

AUTH 1A    AUTH 1B   Key Index: 0   PK Key: 255 255 255 255 255 255

ValueBlockInSector Increment   ValueBlockInSector Decrement

ValueBlockInSector Decrement		ValueBlockInSectorDecrementAKM1	
Decrement Value	10	Decrement Value	10
Sector Address	0	Sector Address	0
Block Address	1	Block Address	1
DECREMENT		DECREMENT	

ValueBlockInSectorDecrementAKM2		ValueBlockInSectorDecrementPK	
Decrement Value	10	Decrement Value	10
Sector Address	0	Sector Address	0
Block Address	1	Block Address	1
DECREMENT		DECREMENT	

- In “ValueBlockInSector Decrement” window you have to enter decrement value, sector address, block address and choose appropriate key index and authentication mode (AUTH 1A or AUTH 1B) which are in section ‘1’, then click “DECREMENT” button. Block 1 in sector 0 value decrementing is shown above.
- In “ValueBlockInSector Decrement AKM1” window you have to enter decrement value, sector address, block address and choose authentication mode (AUTH 1A or AUTH 1B) which is in section ‘1’, then click “DECREMENT” button. Block 1 in sector 0 value decrementing is shown above.
- In “ValueBlockInSector Decrement AKM2” window you have to enter decrement value, sector address, block address and choose authentication mode (AUTH 1A or AUTH 1B) which is in section ‘1’, then click “DECREMENT” button. Block 1 in sector 0 value decrementing is shown above.

- In “ValueBlockInSector Decrement PK” window you have to enter decrement value, sector address, block address you have to enter Provided key (6 bytes - 0xFFFFFFFF is default hex or 255 255 255 255 255 255 decimal) which is in section ‘1’, then click “DECREMENT” button. Block 1 in sector 0 value decrementing is shown above.

## 6.8 Sector trailer write

**Sector Trailer Write(AKM1,AKM2,PK)**

AUTH 1A  AUTH 1B Key Index 0 PK Key 255 255 255 255 255 255

**KEY A** 255 255 255 255 255 255  Hex

**KEY B** 255 255 255 255 255 255  Hex

SectorTrailerWrite | SectorTrailerWrite\_AKM1 | SectorTrailerWrite\_AKM2 | SectorTrailerV

Addressing Mode [v] Trailer Access Bits [v]

Block or Sector Address [ ] Trailer Byte 9 [ ]

Access Bits 0 [v]

Access Bits 1 [v]

Access Bits 2 [v]

**WRITE**

- In “SectorTrailerWrite” tab you have to enter new key A and new key B which are in section ‘2’, choose addressing mode (0 - absolute or 1 - relative), sector address, block 0 access bits, block 1 access bits, block 2 access bits, sector trailer access bits, sector trailer byte 9 which are in section ‘3’ and choose appropriate key index and authentication mode (AUTH 1A or AUTH 1B) which is in section ‘1’, then click “WRITE” button.
- In “SectorTrailerWrite\_AKM1” tab you have to enter new key A and new key B which are in section ‘2’, choose addressing mode (0 - absolute or 1 - relative), sector address, block 0 access bits, block 1 access bits, block 2 access bits, sector trailer access bits, sector trailer byte 9 which are in section ‘3’ and choose authentication mode (AUTH 1A or AUTH 1B) which is in section ‘1’, then click “WRITE” button.
- In “SectorTrailerWrite\_AKM2” tab you have to enter new key A and new key B which are in section ‘2’, choose addressing mode (0 - absolute or 1 - relative),

sector address, block 0 access bits, block 1 access bits, block 2 access bits, sector trailer access bits, sector trailer byte 9 which are in section '3' and choose authentication mode (AUTH 1A or AUTH 1B) which is in section '1', then click "WRITE" button.

- In "SectorTrailerWrite\_PK" tab you have to enter new key A and new key B which are in section '2', choose addressing mode (0 - absolute or 1 - relative), sector address, block 0 access bits, block 1 access bits, block 2 access bits, sector trailer access bits, sector trailer byte 9 which are in section '3' and you have to enter Provided key (6 bytes - 0xFFFFFFFFFFFF is default hex or 255 255 255 255 255 255 decimal) which is in section '1', then click "WRITE" button.

## 6.9 Linear format card

- In "LinearFormatCard" tab you have to enter new key A and new key B which are in section '2', block access bits, sector trailer access bits and sector trailer byte 9 which are in section '3' and choose appropriate key index and authentication mode (AUTH 1A or AUTH 1B) which is in section '1', then click "FORMAT" button.
- In "LinearFormatCard\_AKM1" tab you have to enter new key A and new key B which are in section '2', block access bits, sector trailer access bits and sector trailer byte 9 which are in section '3' and choose authentication mode (AUTH 1A or AUTH 1B) which is in section '1', then click "FORMAT" button.

- In “LinearFormatCard\_AKM2” tab you have to enter new key A and new key B which are in section ‘2’, block access bits, sector trailer access bits and sector trailer byte 9 which are in section ‘3’ and choose authentication mode (AUTH 1A or AUTH 1B) which is in section ‘1’, then click “FORMAT” button.
- In “LinearFormatCard\_PK” tab you have to enter new key A and new key B which are in section ‘2’, block access bits, sector trailer access bits and sector trailer byte 9 which are in section ‘3’ and you have to enter Provided key (6 bytes - 0xFFFFFFFFFFFF is default hex or 255 255 255 255 255 255 decimal) which is in section ‘1’, then click “FORMAT” button