MIFARE® EXAMPLES

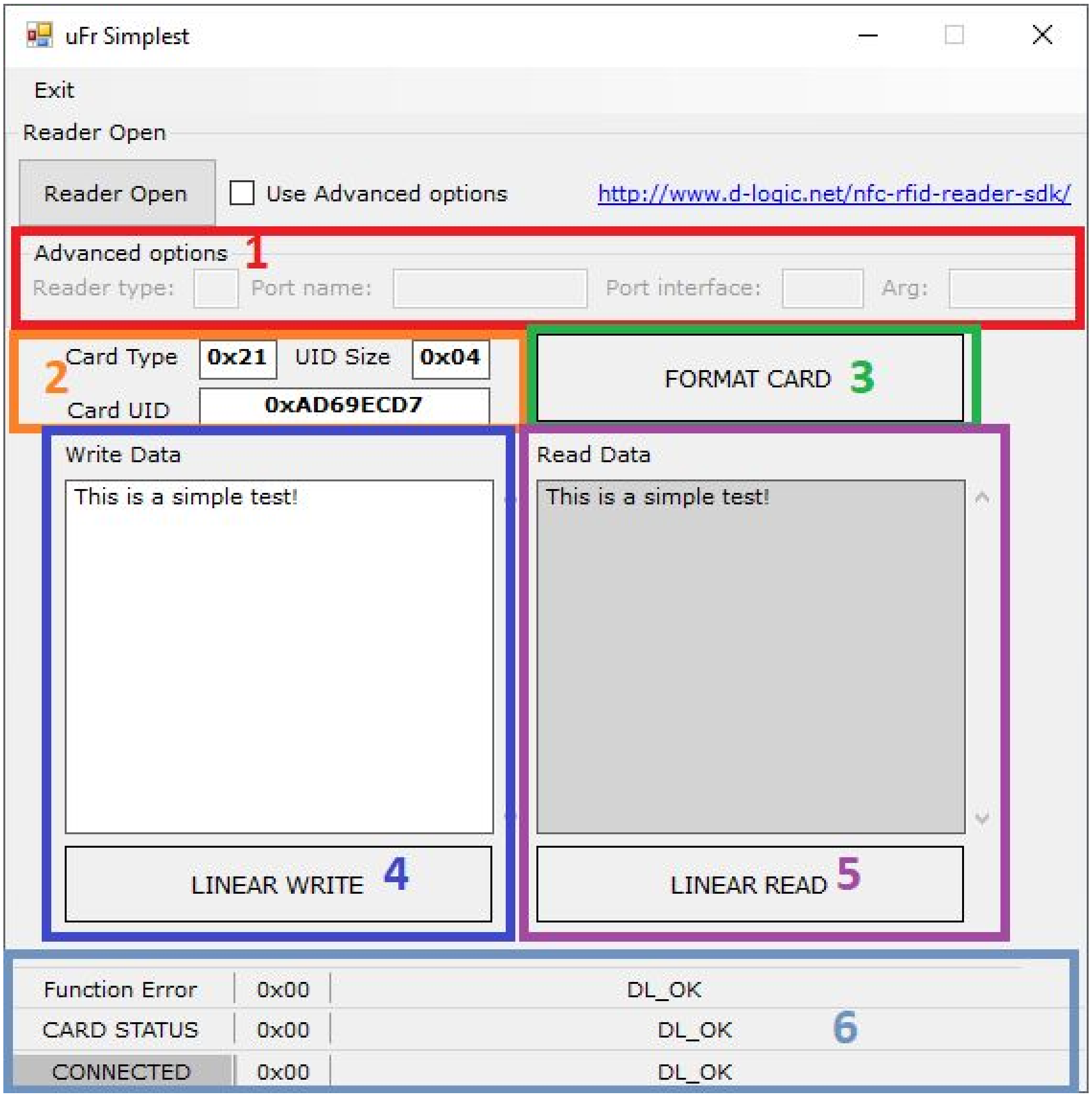
Download link: https://www.d-logic.net/code/nfc-rfid-reader-sdk/ufr-mf-examples-c_sharp.git

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



☐ Use Advanced options

Card Type 0x21 UID Size 0x04

Card UID 0xAD69ECD7

Write Data: This is a simple test!

Read Data: This is a simple test!

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”.

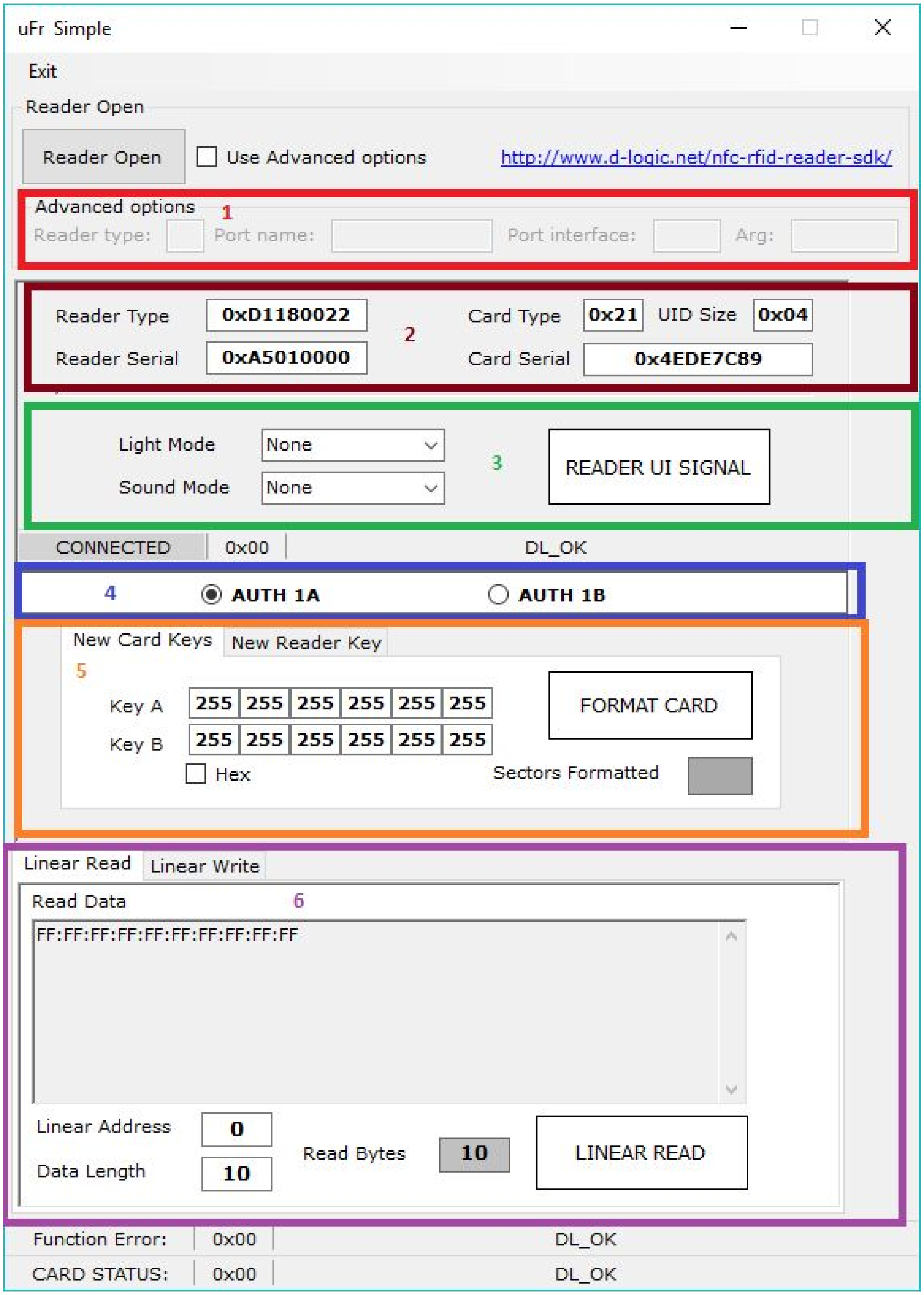
5. Linear read

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 reade.
Reader status - will display DL_OK if Reader Open function was successful.

Simple



Reader Type 0xD1180022

Reader Serial 0xA5010000

Card Type 0x21 UID Size 0x04

Card Serial 0x4EDE7C89

Light Mode None

Sound Mode None

CONNECTED 0x00 DL_OK

● AUTH 1A ○ AUTH 1B

Key A 255 255 255 255 255 255

Key B 255 255 255 255 255 255

□ Hex

Sectors Formatted

Read Data

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

Linear Address 0

Data Length 10

Read Bytes 10

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.

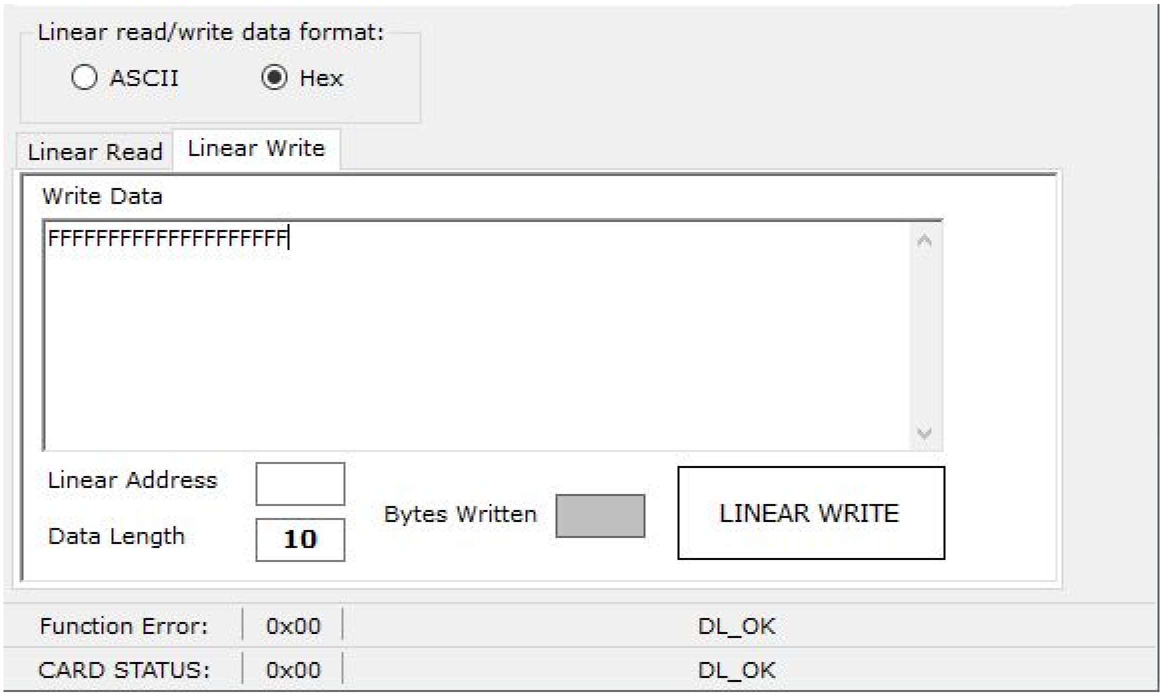
- If you click on tab 'New Reader Key' you will be able to see button 'FORMAT READER KEYS' which will, after clicking, set entered reader key **0xFFFFFFFFFFFF -** 6 bytes, key under provided index parameter in EEPROM

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

Write Data: FFFFFFFFFFFFFFFFFFFF

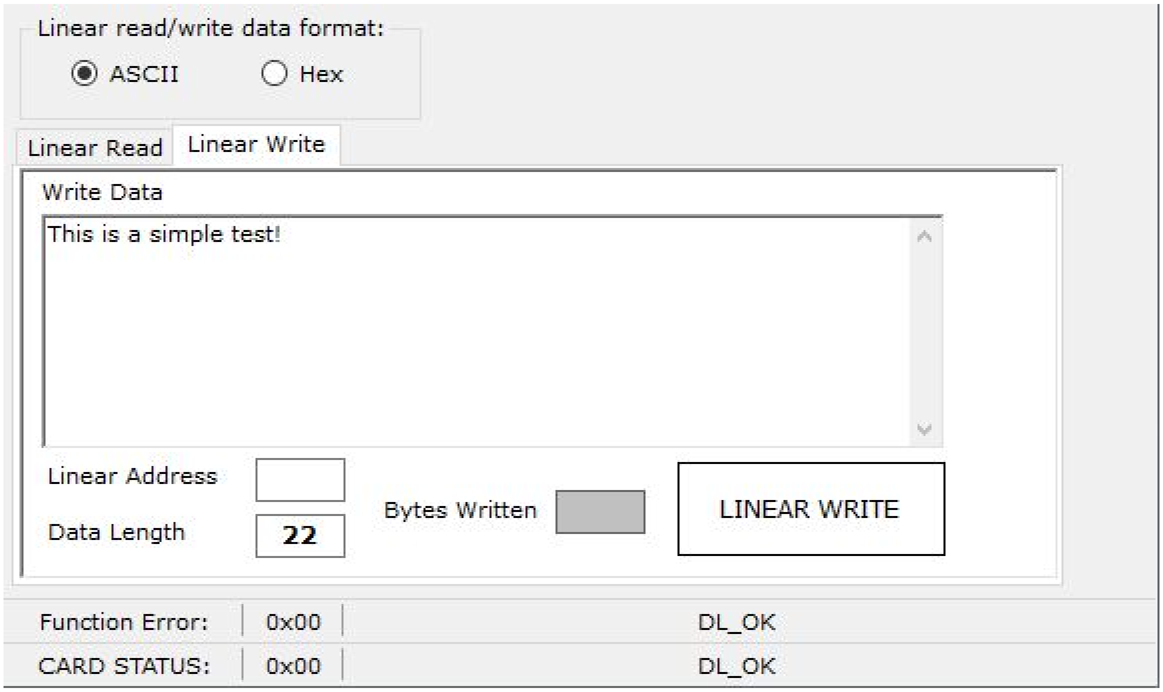
Linear Address:

Data Length: 10

Bytes Written:

Function Error: 0x00 DL_OK

CARD STATUS: 0x00 DL_OK



Linear read/write data format: ● ASCII ○ Hex

Write Data: This is a simple test!

Linear Address:

Data Length: 22

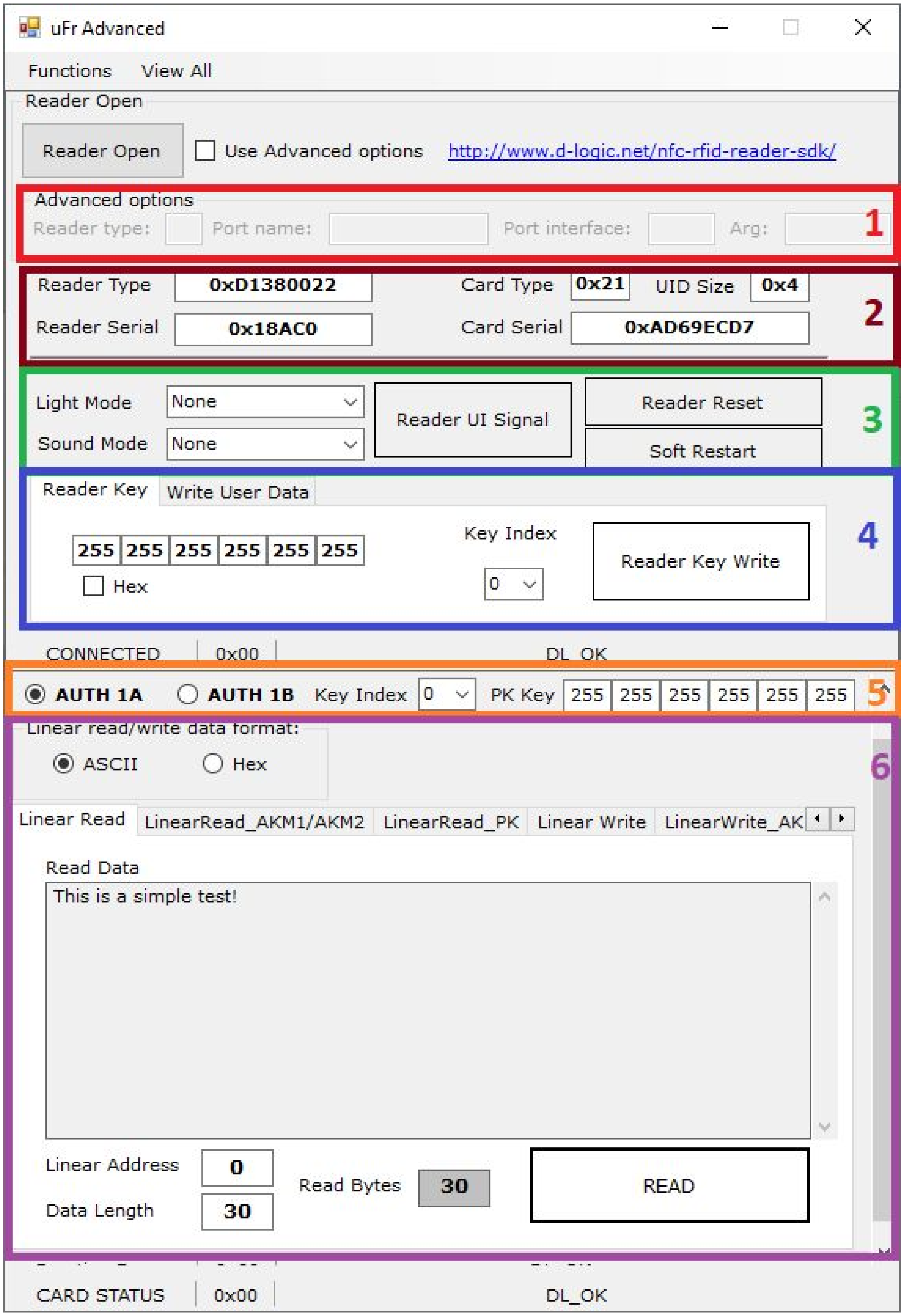
Bytes Written:

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



□ Use Advanced options

Reader Type 0xD1380022
Card Type 0x21
UID Size 0x4
Reader Serial 0x18AC0
Card Serial 0xAD69ECD7

Light Mode None
Sound Mode None

255 255 255 255 255 255
□ Hex
Key Index 0

CONNECTED 0x00 DL_OK

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

Linear read/write data format:
● ASCII ○ Hex

Read Data
This is a simple test!

Linear Address 0
Data Length 30
Read Bytes 30

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

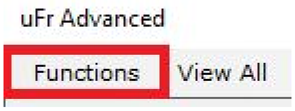
You can choose between authentication with key A or authentication with key B by clicking on radio boxes in section number '4'.

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:

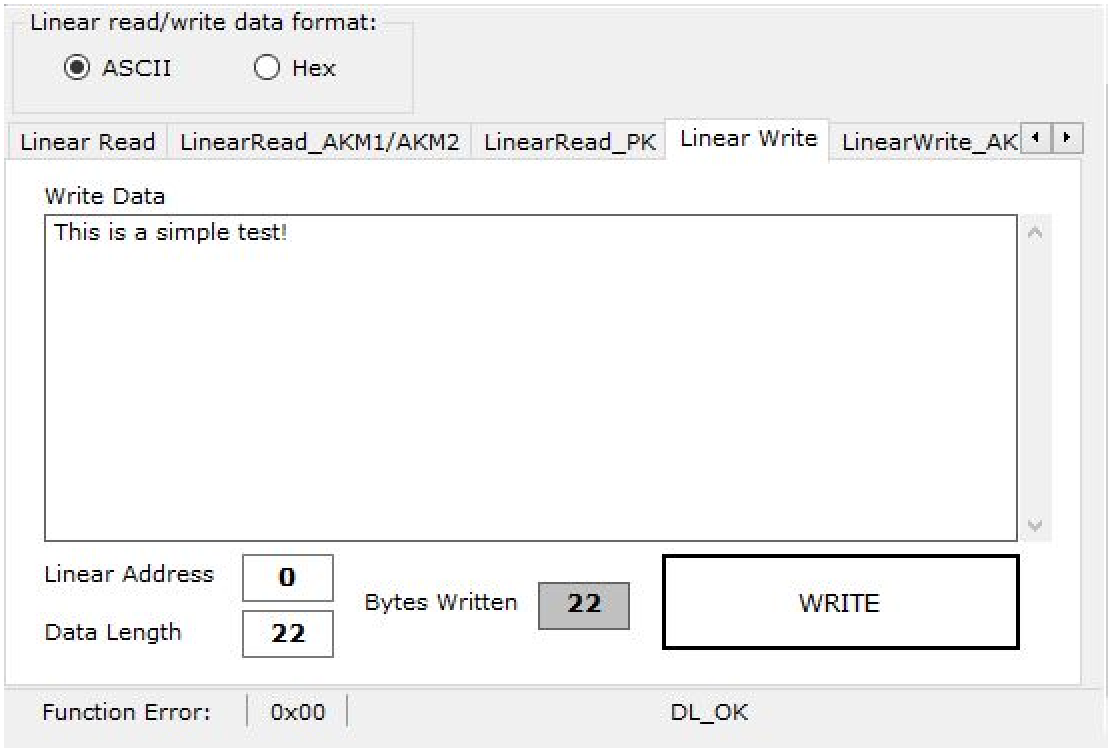
- On ‘Linear Read’ tab you have to choose key index and authentication mode (AUTH 1A or AUTH 1B) for reading which is in section ‘4’, linear address (where to start reading) and data length (how many bytes to read) and click ‘READ’ button. After clicking you will be able to see card data in text box.

- 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:



Linear read/write data format: ● ASCII ○ Hex

Write Data: This is a simple test!

Linear Address: 0

Data Length: 22

Bytes Written: 22

Function Error: 0x00 DL_OK

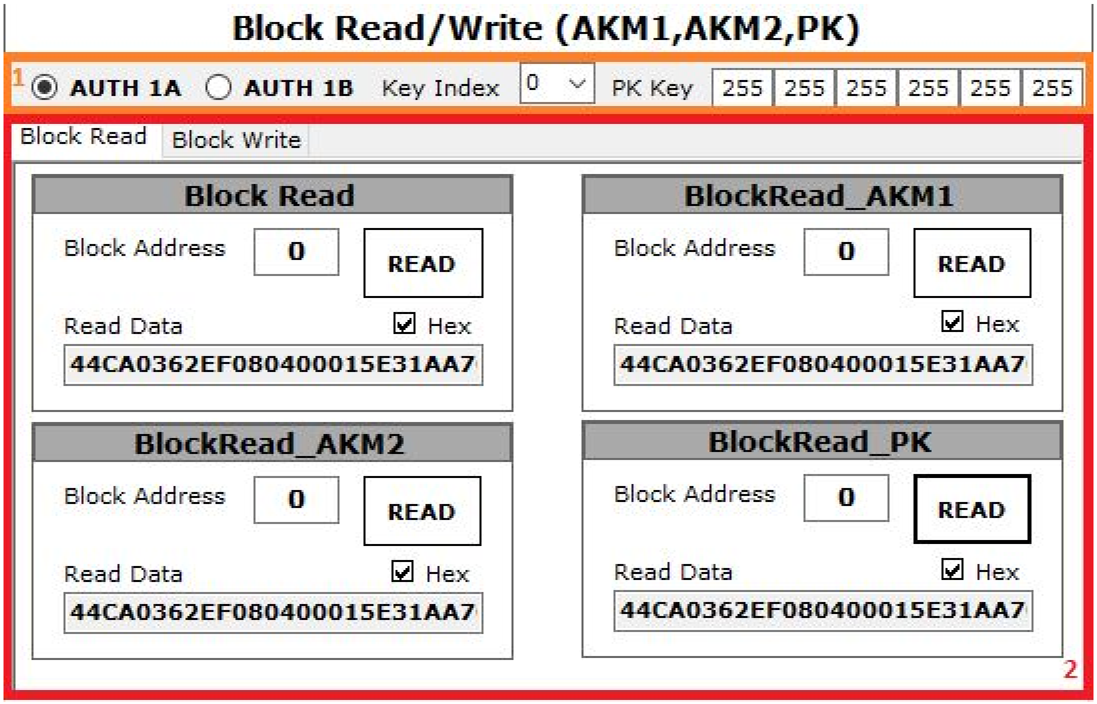
Data length will be automatically calculated. Bytes written shows a number of bytes that are written into card after clicking ‘WRITE’ button.

- 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

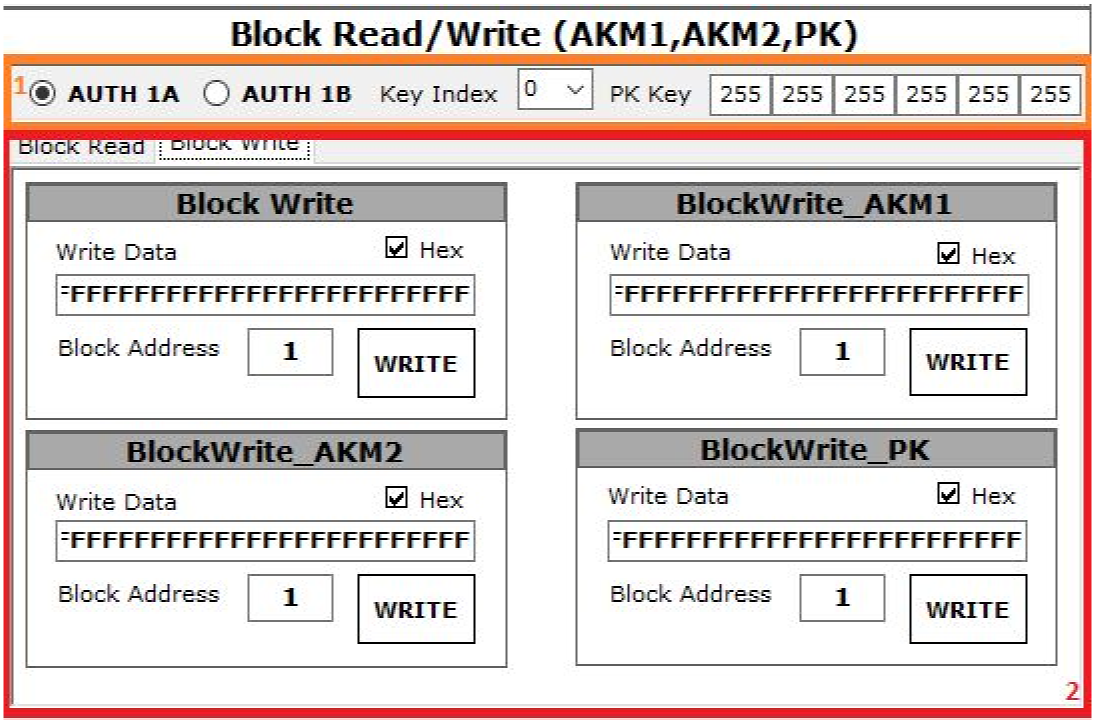
<u>Block read:</u>

- 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 Read/Write (AKM1,AKM2,PK)

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

Block Write: Write Data ☑ Hex FFFFFFFFFFFFFFFFFFFFFFFF Block Address 1

BlockWrite_AKM1: Write Data ☑ Hex FFFFFFFFFFFFFFFFFFFFFFFF Block Address 1

BlockWrite_AKM2: Write Data ☑ Hex FFFFFFFFFFFFFFFFFFFFFFFF Block Address 1

BlockWrite_PK: Write Data ☑ Hex FFFFFFFFFFFFFFFFFFFFFFFF Block Address 1

Block write:

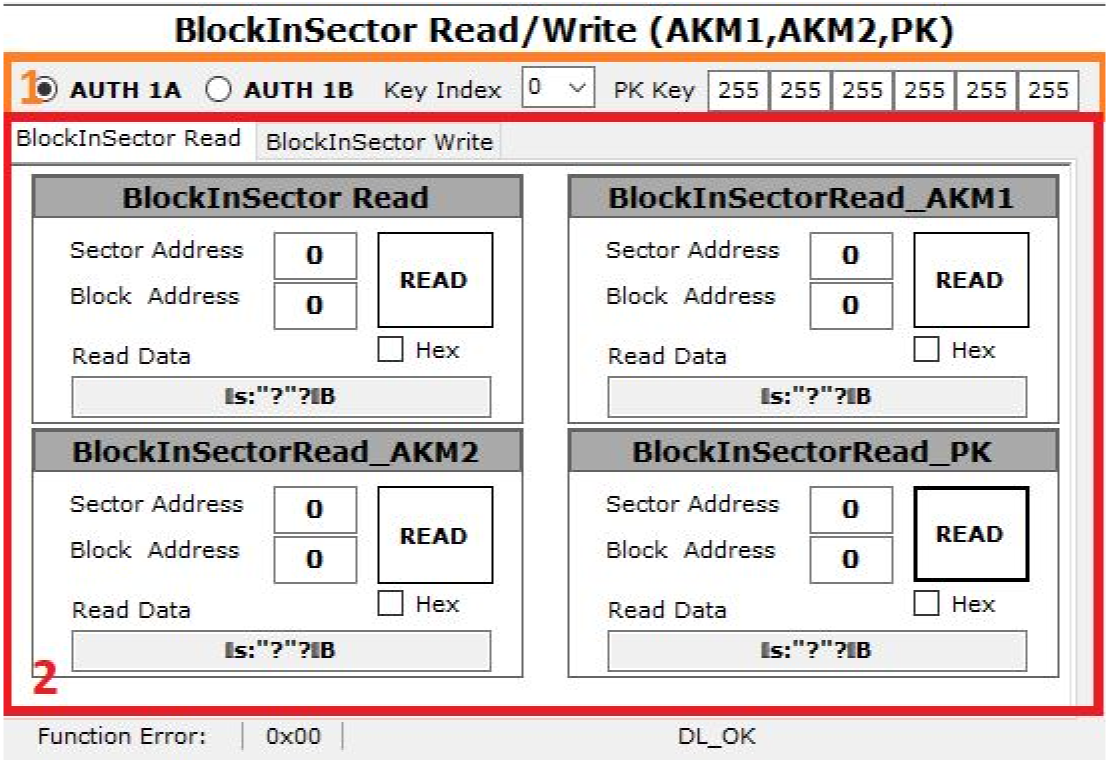
- In ‘’Block Write” window you can choose block address and ASCII or hex data input by checking ‘Hex’ checkbox. For successful writing, you have to enter 16 bytes of data in textbox, choose appropriate key index and authentication mode (AUTH 1A or AUTH 1B) from section ‘1’ and then click ‘’WRITE’’ button. 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



BlockInSector Read/Write (AKM1,AKM2,PK)

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

BlockInSector Read: Sector Address 0, Block Address 0, □ Hex, Read Data Is:"?"?IB

BlockInSectorRead_AKM1: Sector Address 0, Block Address 0, □ Hex, Read Data Is:"?"?IB

BlockInSectorRead_AKM2: Sector Address 0, Block Address 0, □ Hex, Read Data Is:"?"?IB

BlockInSectorRead_PK: Sector Address 0, Block Address 0, □ Hex, Read Data Is:"?"?IB

Function Error: 0x00 DL_OK

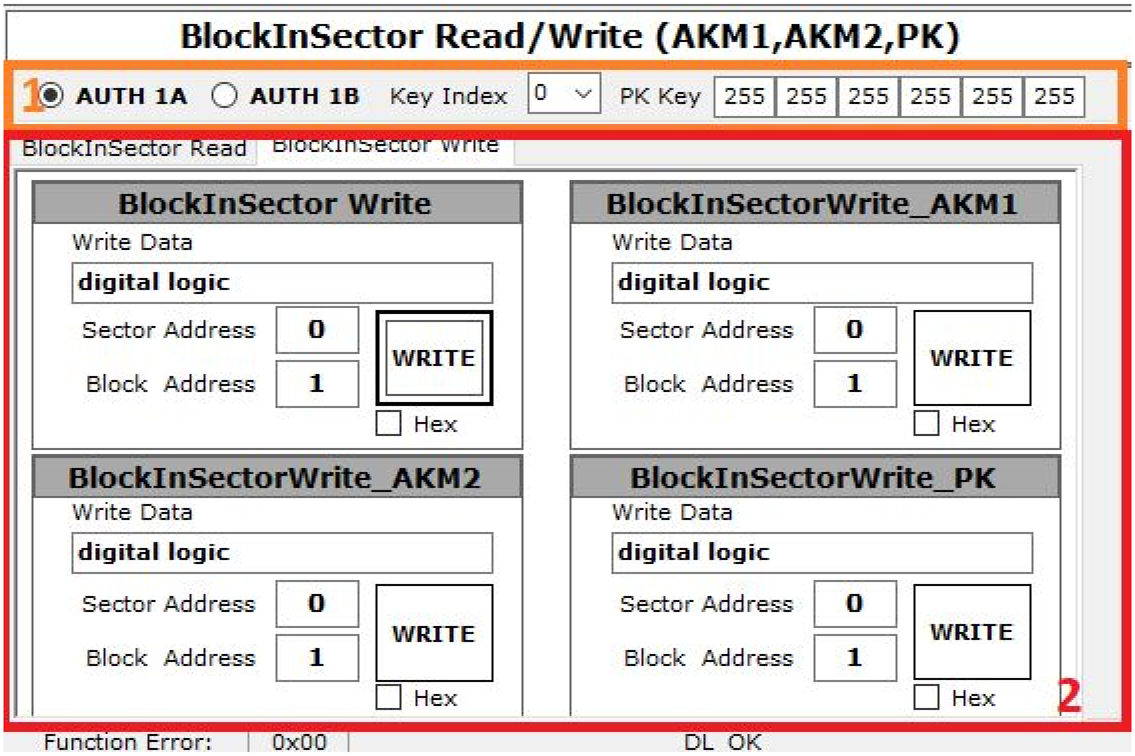
<u>Block in sector read:</u>

- 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.



BlockInSector Read/Write (AKM1,AKM2,PK)

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

BlockInSector Read | BlockInSector Write

BlockInSector Write
Write Data: digital logic
Sector Address: 0
Block Address: 1
WRITE
□ Hex

BlockInSectorWrite_AKM1
Write Data: digital logic
Sector Address: 0
Block Address: 1
WRITE
□ Hex

BlockInSectorWrite_AKM2
Write Data: digital logic
Sector Address: 0
Block Address: 1
WRITE
□ Hex

BlockInSectorWrite_PK
Write Data: digital logic
Sector Address: 0
Block Address: 1
WRITE
□ Hex

2

Function Error: 0x00 DL OK

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.

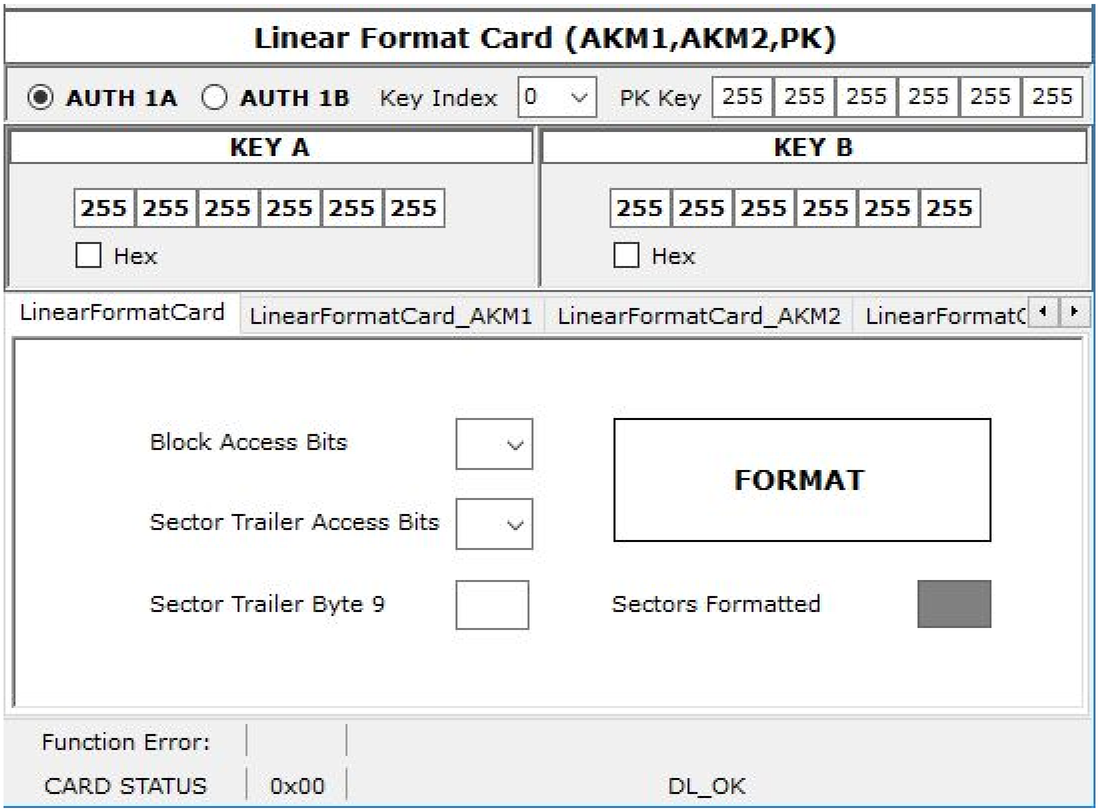
- In ‘’BlockInSectorWrite_AKM1” window you can choose sector address and block address and enter ASCII data. For successful writing,you have to choose authentication mode (AUTH 1A or AUTH 1B) from section ‘1’. Block 1 in sector 0 data writing is shown at the picture above.

- 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 - 0xFFFFFFFFFFFF 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.



Linear Format Card (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

Block Access Bits

Sector Trailer Access Bits

Sector Trailer Byte 9

Sectors Formatted

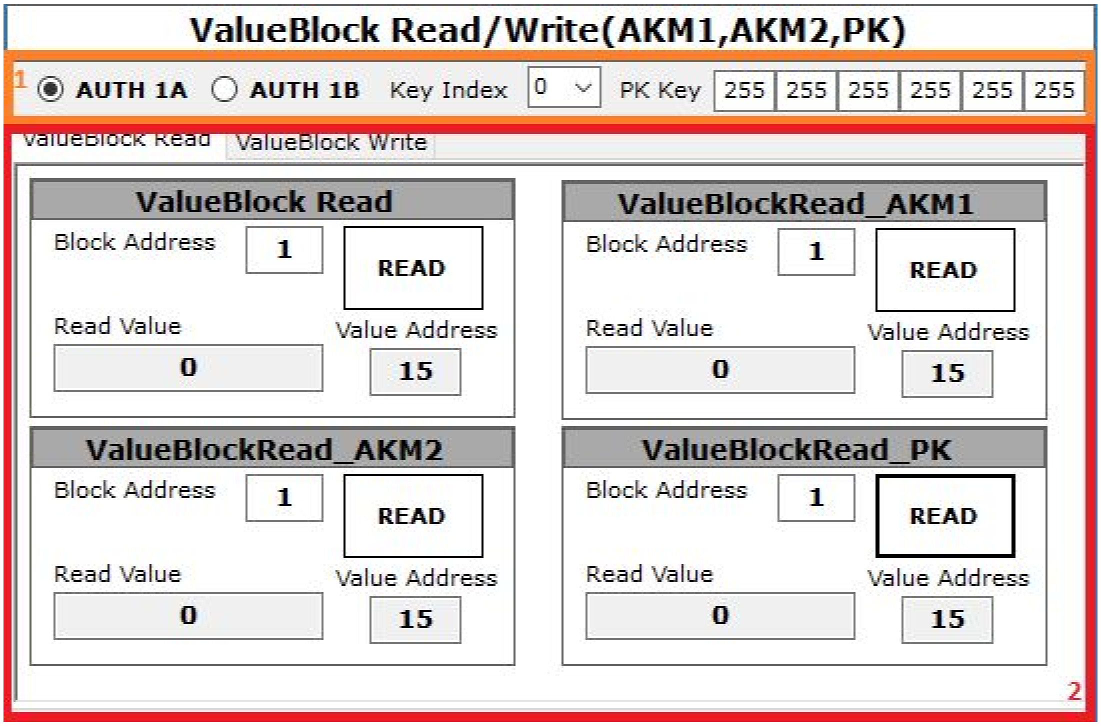
Function Error:

CARD STATUS 0x00 DL_OK

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



ValueBlock Read/Write(AKM1,AKM2,PK)

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

ValueBlock Read | ValueBlock Write

ValueBlock Read: Block Address 1, Read Value 0, Value Address 15

ValueBlockRead_AKM1: Block Address 1, Read Value 0, Value Address 15

ValueBlockRead_AKM2: Block Address 1, Read Value 0, Value Address 15

ValueBlockRead_PK: Block Address 1, Read Value 0, Value Address 15

Value block read:

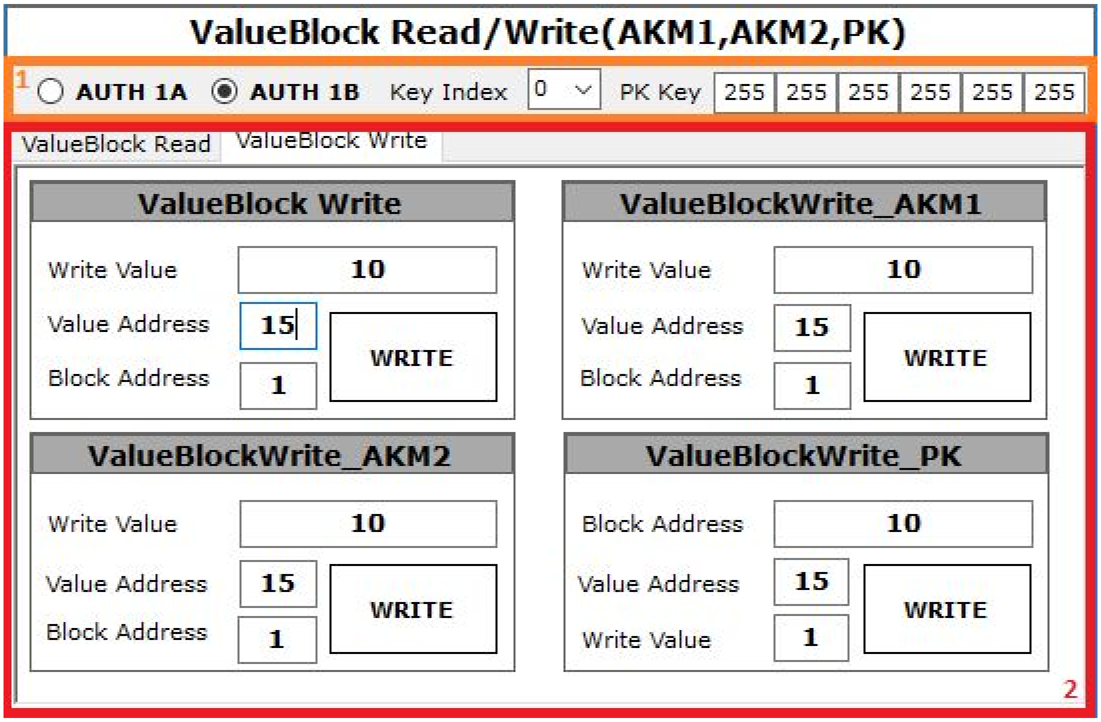
- In “ValueBlock Read” window you have to choose block address, appropriate key index and authentication mode (AUTH 1A or AUTH 1B) which are in section ‘1’, then click ‘’READ’’ button. Block 1 value reading is shown above

- 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.


ValueBlock Read/Write(AKM1,AKM2,PK)

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

ValueBlock Read | ValueBlock Write

ValueBlock Write
Write Value 10
Value Address 15
Block Address 1

ValueBlockWrite_AKM1
Write Value 10
Value Address 15
Block Address 1

ValueBlockWrite_AKM2
Write Value 10
Value Address 15
Block Address 1

ValueBlockWrite_PK
Block Address 10
Value Address 15
Write Value 1

<u>Value block write:</u>

- 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.

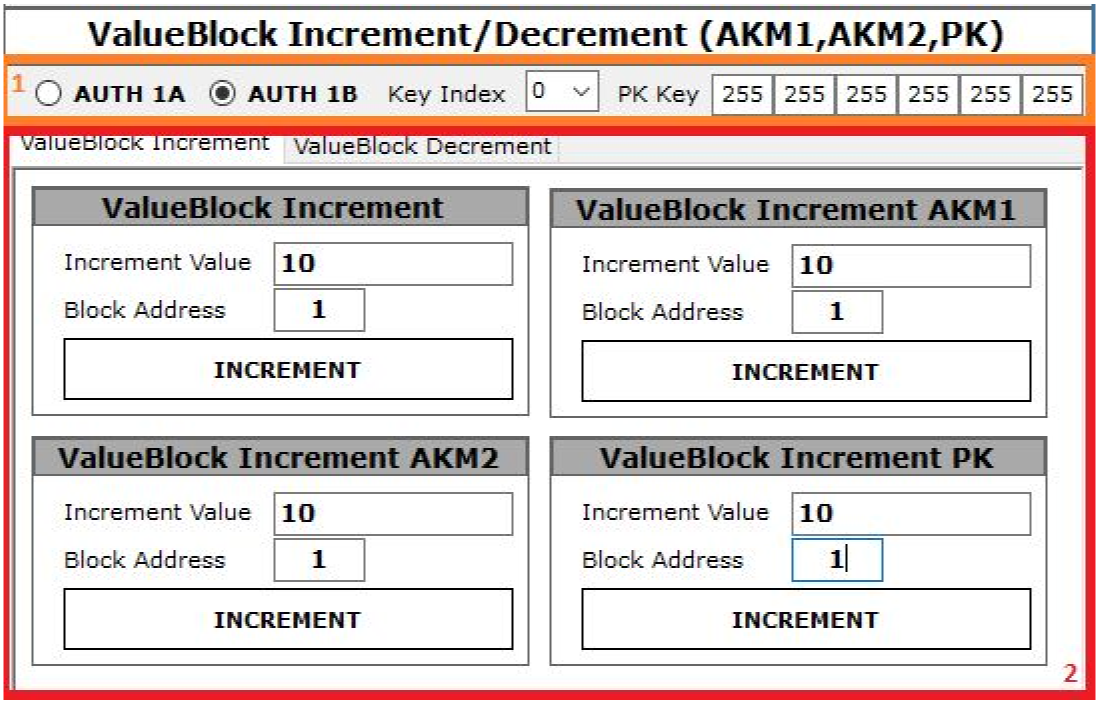
- In “ValueBlockWrite_AKM2” you have to enter value, value address, block address and choose authentication mode (AUTH 1A or AUTH 1B) which is in section ‘1’, then click ‘’WRITE’’ button. Block 1 value writing is shown above.

- 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

<u>Value block increment:</u>



ValueBlock Increment/Decrement (AKM1,AKM2,PK)

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

ValueBlock Increment

Increment Value 10

Block Address 1

ValueBlock Increment AKM1

Increment Value 10

Block Address 1

ValueBlock Increment AKM2

Increment Value 10

Block Address 1

ValueBlock Increment PK

Increment Value 10

Block Address 1

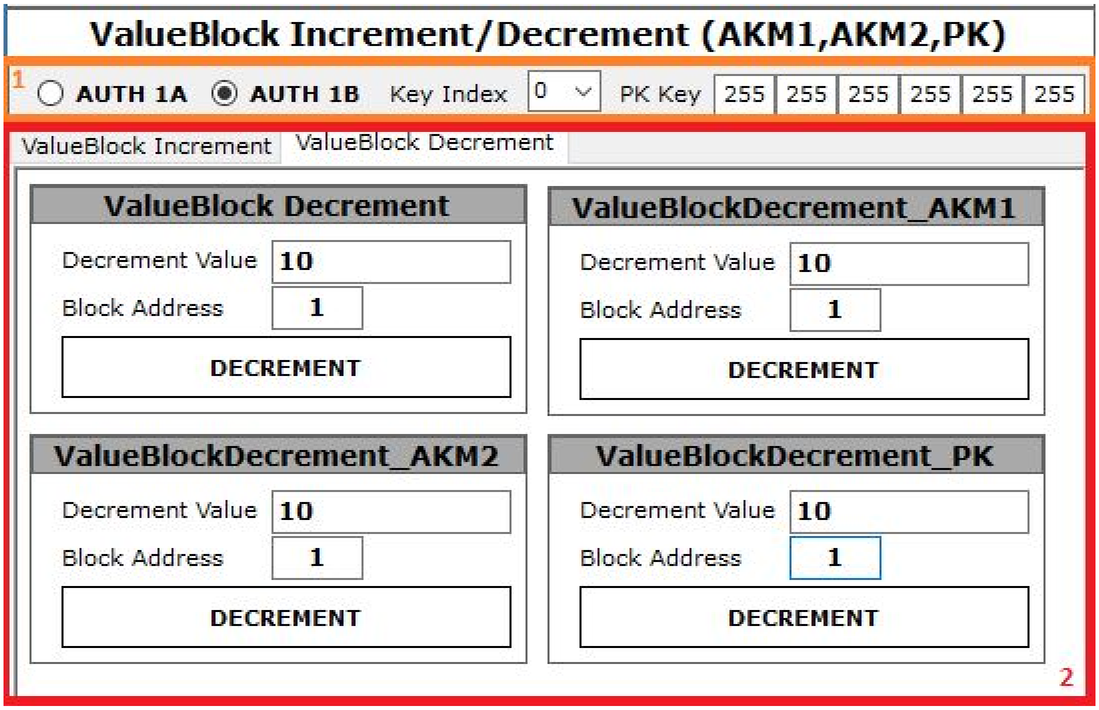
- In “ValueBlock Increment” window you have to enter increment value, block address, and choose appropriate key index and authentication mode (AUTH 1A or AUTH 1B) which are in section ‘1’, then click ‘’INCREMENT’’ button. Block 1 value incrementing is shown above.

- 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 255 255 255 255 255 255 decimal) which is in section ‘1’, then click ‘’INCREMENT’’ button. Block 1 value incrementing is shown above.

<u>Value block decrement:</u>



ValueBlock Increment/Decrement (AKM1,AKM2,PK)

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

ValueBlock Decrement: Decrement Value 10, Block Address 1

ValueBlockDecrement_AKM1: Decrement Value 10, Block Address 1

ValueBlockDecrement_AKM2: Decrement Value 10, Block Address 1

ValueBlockDecrement_PK: Decrement Value 10, Block Address 1

- 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.

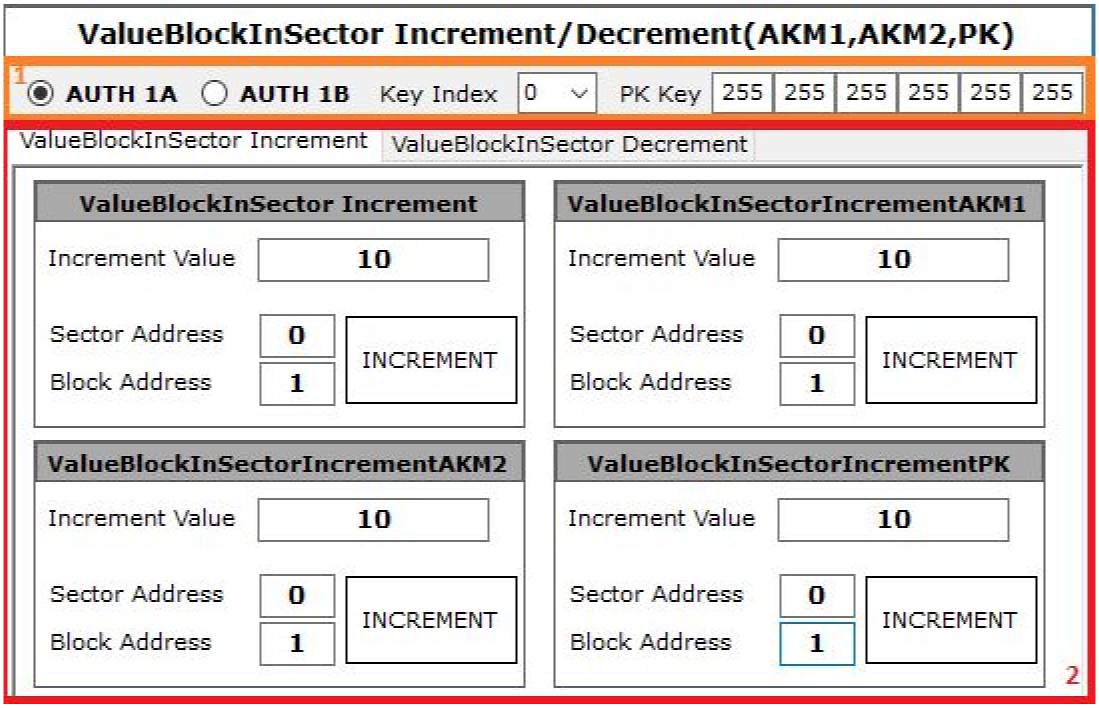
- In “ValueBlock Decrement AKM1” window you have to enter decrement value, block address and choose authentication mode (AUTH 1A or AUTH 1B) which is in section ‘1’, then click ‘’DECREMENT’’ button. Block 1 value decrementing is shown above.

- 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)

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

ValueBlockInSector Increment | ValueBlockInSector Decrement

ValueBlockInSector Increment

Increment Value 10

Sector Address 0

Block Address 1

ValueBlockInSectorIncrementAKM1

Increment Value 10

Sector Address 0

Block Address 1

ValueBlockInSectorIncrementAKM2

Increment Value 10

Sector Address 0

Block Address 1

ValueBlockInSectorIncrementPK

Increment Value 10

Sector Address 0

Block Address 1

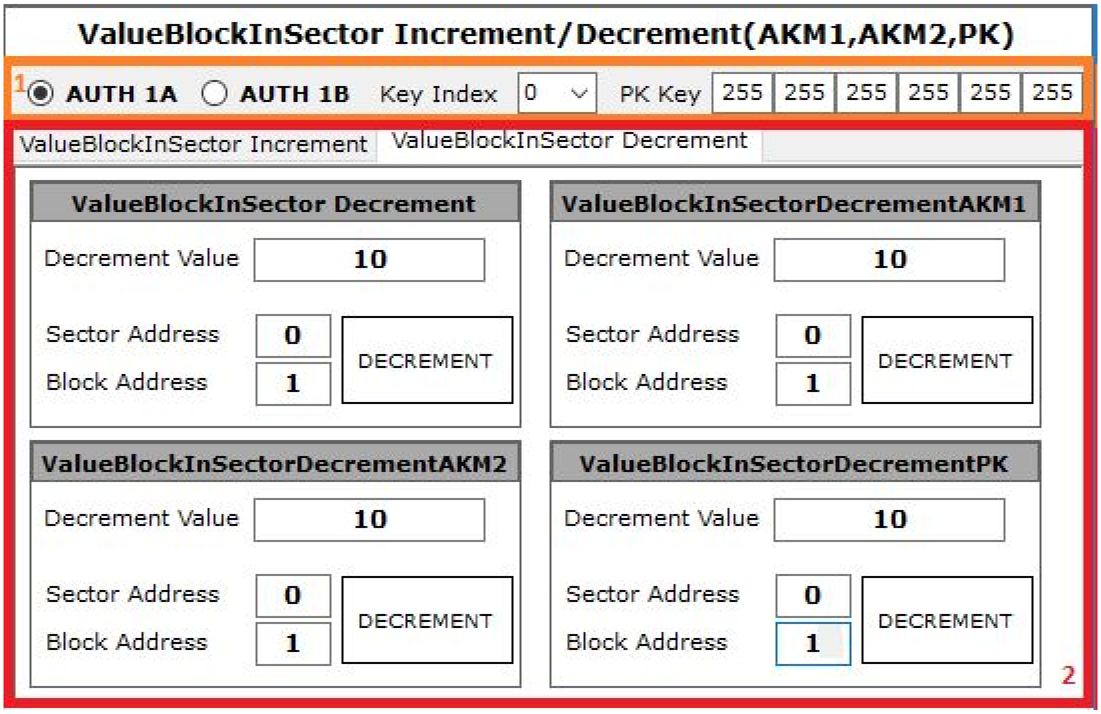
- In “ValueBlockInSector Increment” window you have to enter increment value, sector address, block address and choose appropriate key index and authentication mode (AUTH 1A or AUTH 1B) which are in section ‘1’, then click ‘’INCREMENT’’ button. Block 1 in sector 0 value incrementing is shown above.

- 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 - 0xFFFFFFFFFFFF 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.

<u>Value block in sector decrement:</u>



ValueBlockInSector Increment/Decrement(AKM1,AKM2,PK)

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

ValueBlockInSector Increment | ValueBlockInSector Decrement

ValueBlockInSector Decrement
Decrement Value 10
Sector Address 0
Block Address 1

ValueBlockInSectorDecrementAKM1
Decrement Value 10
Sector Address 0
Block Address 1

ValueBlockInSectorDecrementAKM2
Decrement Value 10
Sector Address 0
Block Address 1

ValueBlockInSectorDecrementPK
Decrement Value 10
Sector Address 0
Block Address 1

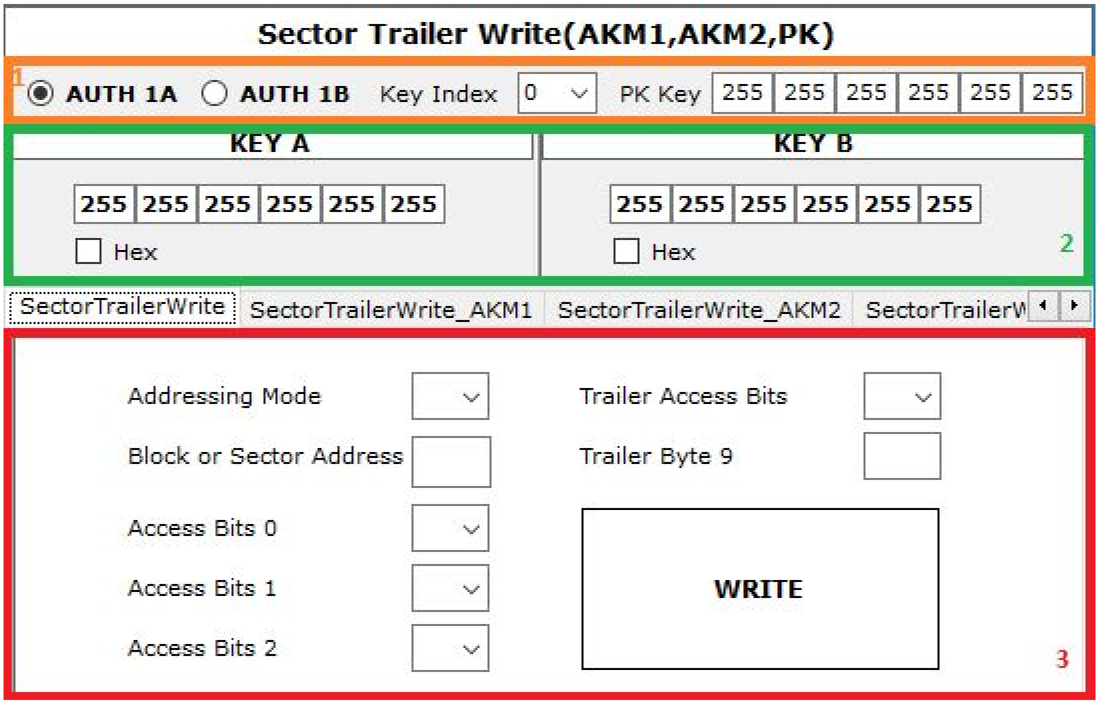
- In “ValueBlockInSector Decrement” window you have to enter decrement value, sector address, block address and choose appropriate key index and authentication mode (AUTH 1A or AUTH 1B) which are in section ‘1’, then click ‘’DECREMENT’’ button. Block 1 in sector 0 value decrementing is shown above.

- 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 - 0xFFFFFFFFFFFF 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 | SectorTrailerW

Addressing Mode

Block or Sector Address

Access Bits 0

Access Bits 1

Access Bits 2

Trailer Access Bits

Trailer Byte 9

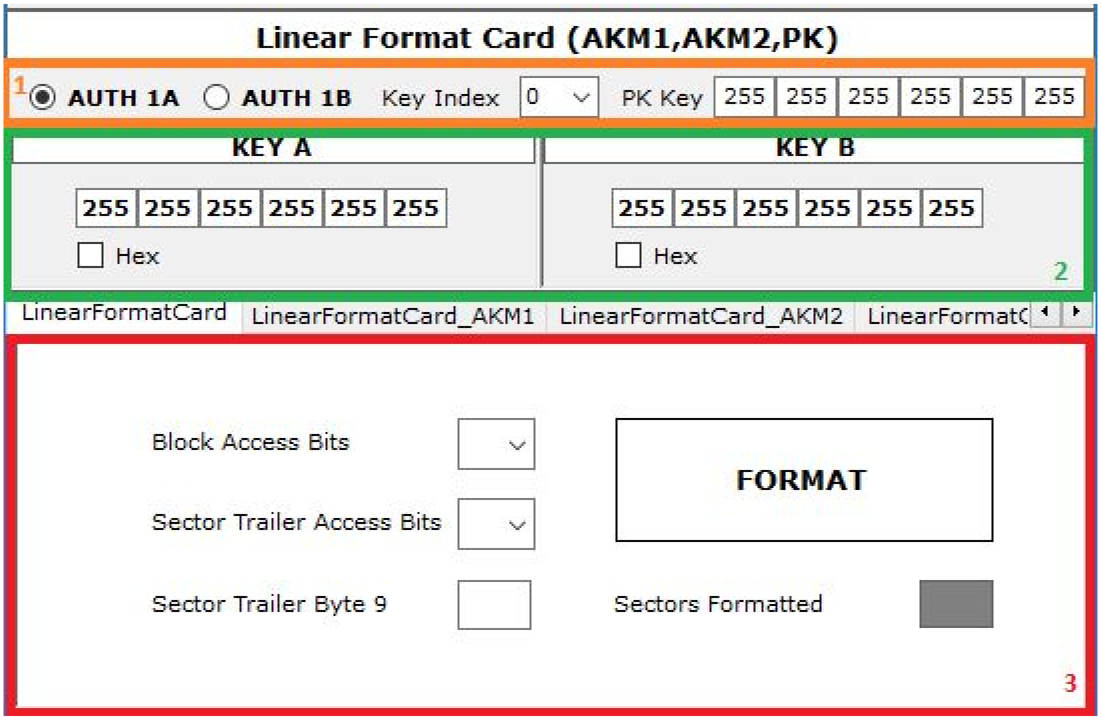
- In ‘’SectorTrailerWrite’’ tab you have to enter new key A and new key B which are in section ‘2’, choose addressing mode (0 - absolute or 1 - relative), sector address, block 0 access bits, block 1 access bits, block 2 access bits, sector trailer access bits, sector trailer byte 9 which are in section ‘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



Linear Format Card (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

Block Access Bits

Sector Trailer Access Bits

Sector Trailer Byte 9

Sectors Formatted

- 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