MR Manual – MF Classic

Visit us at www.mylab.my

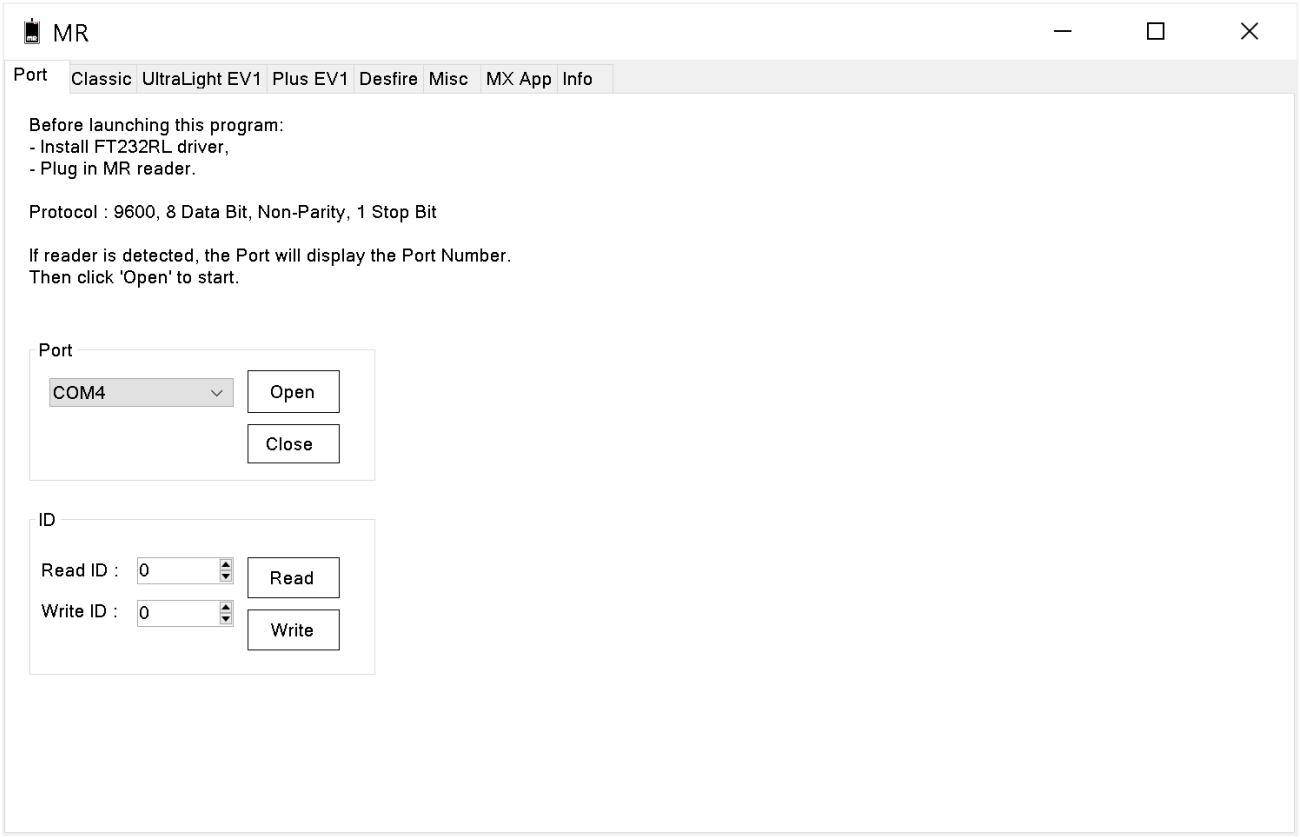
Email: sales@mylab.my

Please email to us, to get a copy of the MR software.

Contents

1. General

Plug in MR Reader then launch the MR Software. 



Main screen.

If Serial Com Port available, or USB Virtual Com available, it will be shown on screen.

Select the port where reader is connected, then click 

to open port.

Upon open port, reader will send command to get reader ID, and turn on the RF. Getting reader ID should be the first command to execute, because if the ID is not matched, reader will not response.

1.0 Protocol

To ease development, sending and receiving protocols are recorded and saved in the **C:\Mylab\MR** folder, as shown below:

Function : Open Port
6:47:51 PM, TX : 02 00 00 01 50 53 03
6:47:51 PM, RX : 02 00 00 02 00 00 00 03
6:47:51 PM, TX : 02 00 00 02 54 01 55 03

They are briefly explained as below:

Communication Protocol:

Item	Value
Baud rate	9600
Data bit	8
Parity	None
Stop bit	1

For card related command:
TX : 02 / 00 / Reader ID / No of byte / Command / PICC Command / Data / BCC / 03
RX : 02 / 00 / Reader ID / No of byte/ Message / CSN length / Card Serial Number / Card Type / PICC Message / Data / BCC / 03

For hardware setting related command:
TX : 02 / 00 / Reader ID / No of byte / Command / Data / BCC / 03
RX : 02 / 00 / Reader ID / No of byte/ Message / Data / BCC / 03

Item	Length	Meaning
STX (Start of Text)	1	02
MID	1	00
Reader ID/RID	1	00 – FF hex
No of byte/NOB	1	No of byte in TX Data or RX Data
Command/CMD	1	Refer manual or sending protocol 0 = none 10 hex = eread 11 hex = ewrite 12 hex = evalue 46 hex = load_key 48 hex = areadsnr 49 hex = areadsnrsize 50 hex = read_id 51 hex = write_id 54 hex = control_rf 6A hex = read_mode 6B hex = write_mode 6D hex = control_beep_led 6E hex = control_beep 6F hex = control_led F0 hex = read_version
PICC Command (Optional)	1	Refer manual or sending protocol
Message/MESG	1	00 – Success, others – Error
CSN length/CSNL	1	0 or 4 or 7 (when 0, there is no serial number returned, hence the Card Serial Number and Card Type column will not be available in the RX

		protocol too.)
Card Serial Number/CSN	4 or 7	4B or 7B CSN
Card Type/CT	1	Refer Mifare Card Information 08 hex = Mifare Classic 18 hex = Mifare Plus (SL1 or Mixed mode) 20 hex = Mifare Plus (SL3) 20 hex = Mifare Desfire
PICC Message/PMESG (optional)	1	00 hex - mi_ok 01 hex - mi_notagerr 02 hex - mi_err
Data	variable	Transmit or Received string
DF Name (Optional)	17	1B length, 16B DF Name (If Virtual Card is enabled)
BCC	1	Exclusive-Or from byte STX to the byte before BCC
ETX (End of Text)	1	03

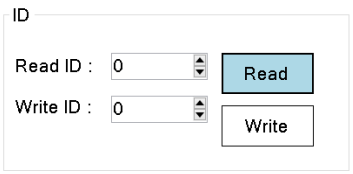
1.1 Reader ID

The default reader ID is 00, it can be changed to other value. In RS485 or multi-drop application, reader with different ID is needed to ensure proper communication.

To set ID value, enter a value in the Write ID numeric box, and then create 'Write'. After you changed the ID, remember to click 

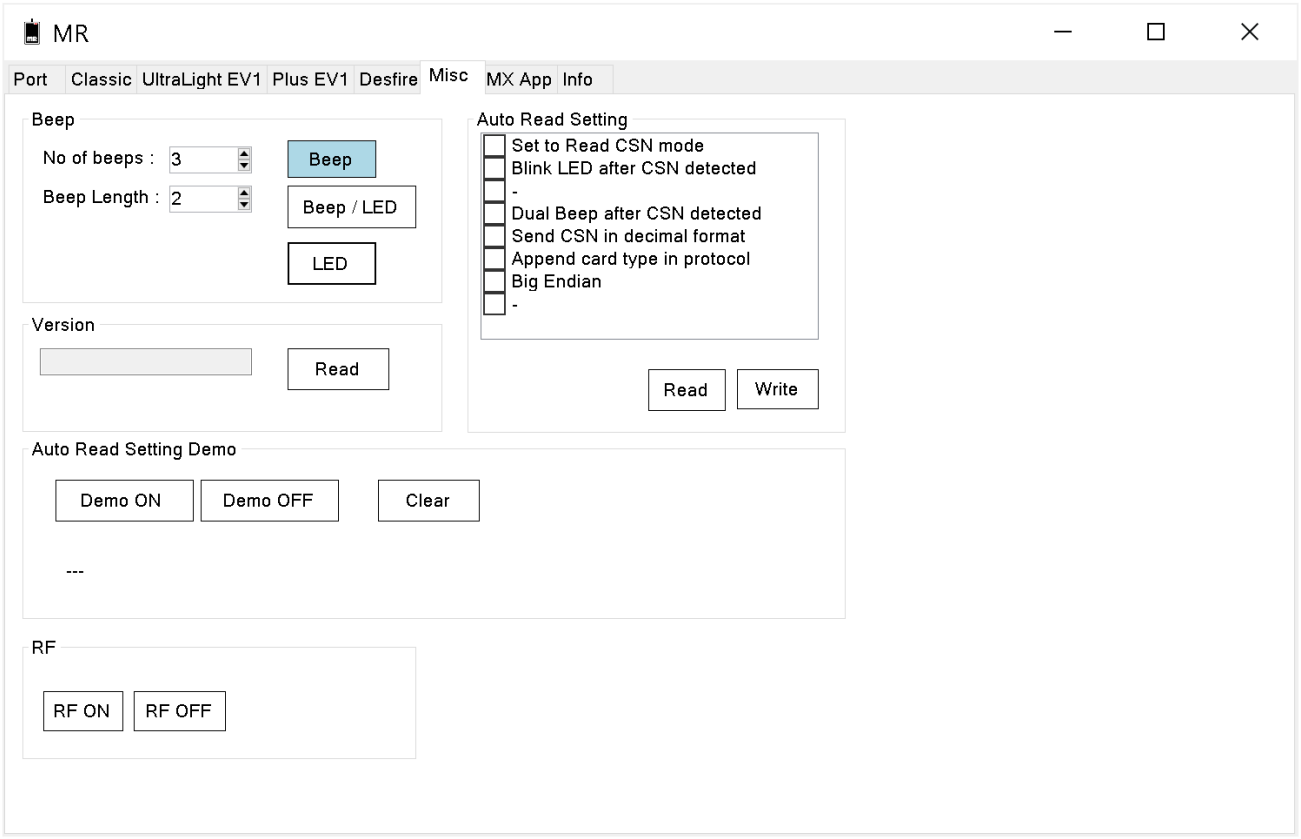
to get the new reader ID, or manually update the read ID numeric box. If not, communication may be fail because the ID in the program is different from the ID in the reader. Therefore, always get the reader ID first before proceeding further.

The value will be saved in the non volatile memory.



1.2 Beep

There are two beep mode available; beep only and beep with LED. Beep with LED will generate beep sound and trigger the on board LED. Multiple beeps can be generated by setting the 'No of beeps' and 'Beep length'.



Click Beep will generate 3 beeps.

1.3 RF

No communication between card and reader is possible when the RF is turned off. Therefore, it is important to ensure the RF is turned on when accessing the card. By default, the RF is on.

Sometimes it may be necessary to control the RF to reset the card in the field.

To reset RF in the reader field, click RF ON, and then click RF OFF.

1.4 Reader Internal EEPROM

The hardware settings, and the authentication keys are saved into the internal EEPROM by using instruction. This is a non-volatile memory, the content will be kept even though power is removed.

Due to security reason, the authentication keys are not readable.

- Please bear in mind that EEPROM is intended to provide nonvolatile storage for configuration data and settings that do not need to change frequently. If an application program were to write to an EEPROM cell frequently it would quickly wear it out, shorten the lifetime of the product.

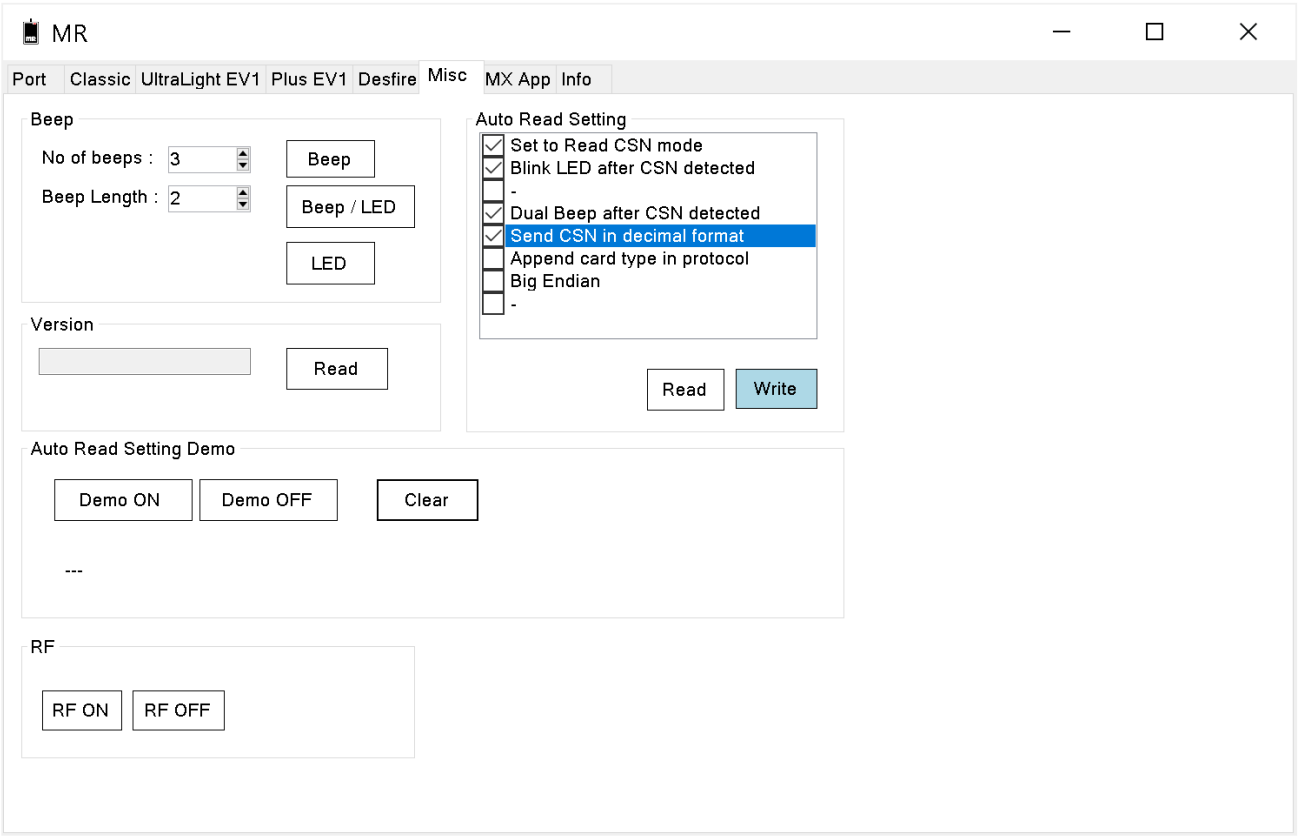
1.5 Remark

After a Mifare card‘s sector has been authenticated successfully, the sector will be opened, and read or write is permitted. The card will remain in the ‘Open’ state until the RF field is reset, or the card is removed from the RF area, then a new authentication is needed.

In the examples given below, the testing result may be different if the RF reset is not performed after changing Key or changing authentication method. Therefore, if the testing result is different from the given answer, kindly reset the RF by calling command ‘RF Off’ and ‘RF On’, or merely remove the card from reader, and then put back the card on the reader. The author has omitted this, to avoid duplicated words and for easy reading. It is also recommended to perform RF reset prior to create new application or new file.

2. How to set to Read Serial Number mode

By default the reader is in Read/Write mode. User can switch the reader to work on Auto Read Serial Number mode by checking the following box. Both 4B and 7B serial number lengths are supported.

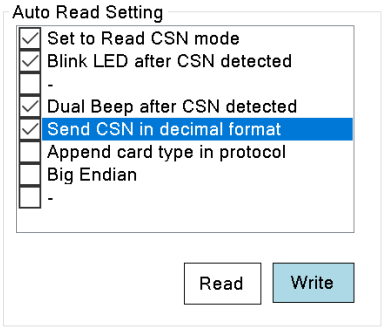


The above setting will switch the reader to work on Read Serial Number mode. When a valid card is detected, it will blink LED, sound two beeps, and send the serial number to Host in decimal and Little Endian format.

Available functions:

a. Set to Read Card SNR mode – Enable Read Serial Number mode, default is Read / Write mode.
b. Blink LED after SNR detected – Blink LED when a valid card is detected.
c. Dual Beep after SNR detected – Sound dual beep when a valid card is detected.
d. Send SNR in Decimal format – Send Serial Number in decimal format, default is sending Serial Number in Hex format.
e. Append Card Type in protocol – Append card type in the protocol. Only if Hex format is enabled, or 'Send SNR in Decimal format' is unchecked.
f. Big Endian – Send Serial Number in Big Endian format, default is sending Serial Number in Little Endian format.

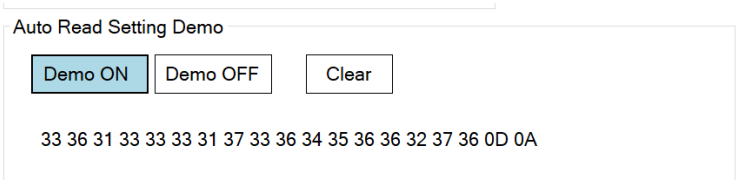
We will try it now by setting the check box as shown below:

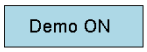


Click 

to send the check box setting to reader.

We will see how it works now by turning on the Demo mode.

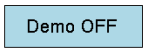


Click 

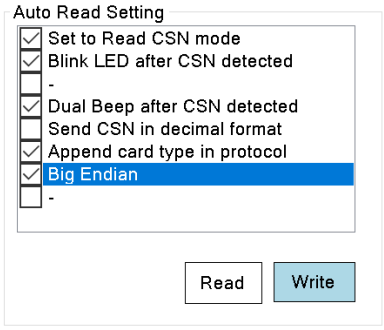
to detect any valid card in the RF field. If available, display the Serial Number in decimal format, blink LED, and sound dual beep.

You may refer Activity text file in C:\Mylab\MR, for detailed protocol received string.

Function : Write Card Setting
12:08:47 PM, TX : 02 00 00 02 6B 1B 70 03
12:08:47 PM, RX : 02 00 00 01 00 03 03
Function : Demo On
12:08:55 PM, RX : 32 34 31 38 31 34 32 31 39 30 0D 0A
12:08:57 PM, RX : 33 36 31 32 35 32 32 36 33 32 33 38 30 31 38 36 30 0D 0A
12:09:07 PM, RX : 33 36 31 33 33 33 31 37 33 36 35 35 34 39 33 31 36 0D 0A

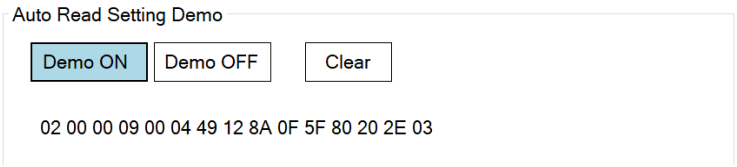
Turn off the demo now by clicking 

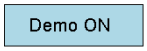
, and try another setting as shown below:

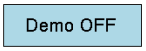


Click 

to send the check box setting to reader. This setting will include the card type in protocol.



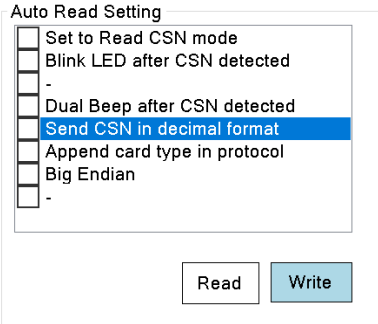
Click  to detect any valid card in the RF field. If card available, it will return the Serial Number in Hex format, append the card type in the protocol, blink LED, and sound beep. As shown in the protocol 02 00 00 09 00 04 49 12 8A 0F 5F 80 20 2E 03, the card type is 20 hex. Refer Chapter 1-0 Protocol, we know that the card is Mifare Desfire or Mifare Plus (SL3).

Always remember to turn off the demo by clicking  .

The setting will be stored as non volatile in the reader's memory.

3. How to set to Read / Write mode

By default the reader is in Read/Write mode. If the reader has been set to Auto Read Serial Number mode, we can switch it back to Read/Write mode by un-checking all the items in the following box, then click Write.



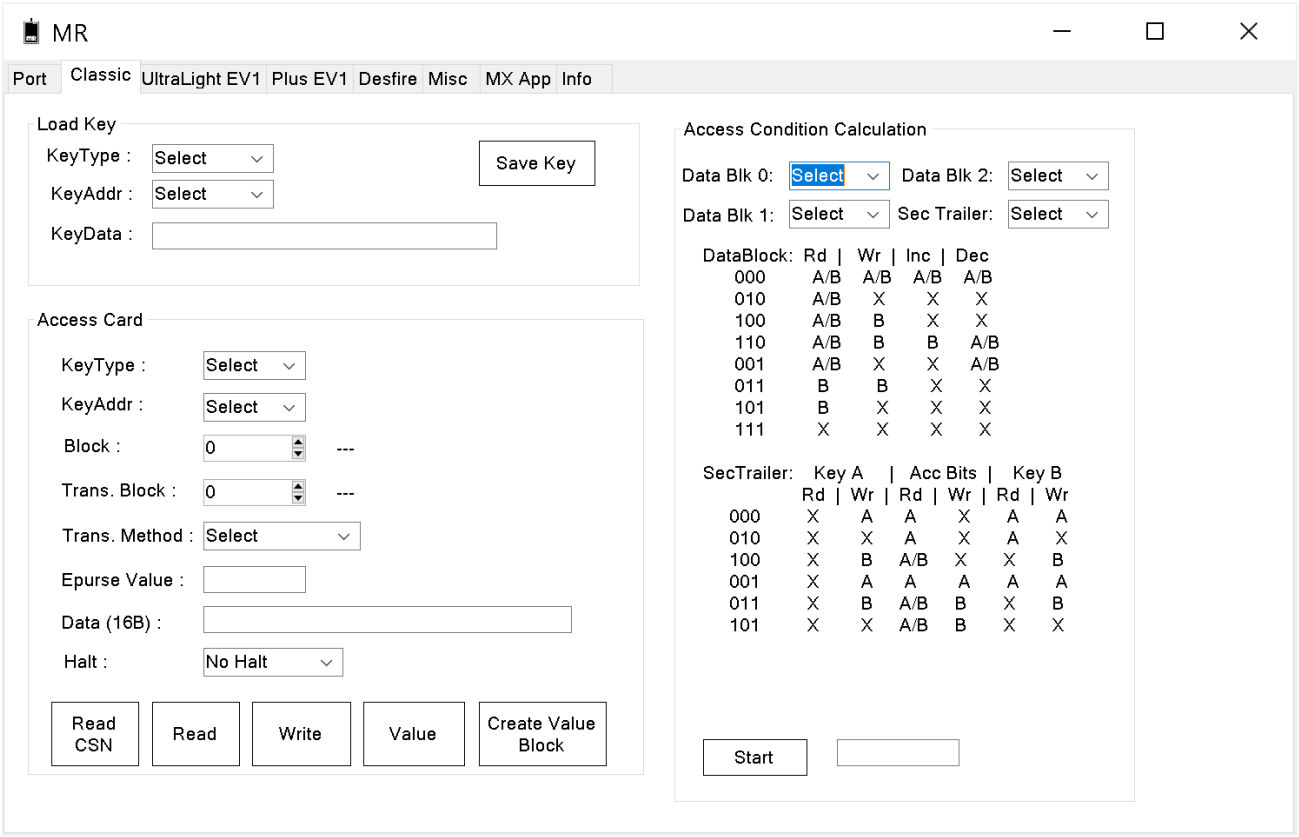
The reader is now in Read/Write mode.

The reader must be in the Read/Write mode in order to follow the examples in the chapters that follow.

The setting will be stored as non volatile in the reader's memory.

4. Mifare Classic

Click the 'Classic' tab now to enter the page. This program will show you how to access MF Classic card.



Classic main screen.

4.0 Access Method

To access the MF Classic card content, a successful authentication is necessary. The card is protected by 6B Key A by default, and the Key is FFFFFFFFFFFF hex. If the Key does not match, an error code will be returned. Before authentication can be successful, a 'Load Key' command to load the authentication key to the reader is required. The key will be stored in non volatile memory.

4.1 Card Structure

Depends on the card size, their memory location may be different. User is advised to refer to the card manufacturer's documentation for details.

The 1K Byte Mifare ® Standard memory is organized in 16 sectors with 4 blocks each, whereas 4K Byte Mifare ® Standard memory is organized in 32 sectors with 4 blocks and in 8 sectors with 16 blocks. One block consists of 16 bytes.

The last block of every sector is the sector trailer. Each sector trailer holds secret keys A and B, and the access condition for all blocks of that sector.

The memory organization for 4K Mifare ® Standard is shown below, the 1K Mifare ® Standard has the same structure but occupies sector 0 to 15 only.

Sector	Block	Byte number within a block 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Description
0	0																	Manufacturer Data
	1																	Data
	2																	Data
	3	Key A						Access condition						Key B				Sector Trailer
1	4																	Data
	5																	Data
	6																	Data
	7	Key A						Access condition						Key B				Sector Trailer
2	8																	Data
	--																	--
	--																	--
	--																	--
31	7C																	Data
	7D																	Data
	7E																	Data
	7F	Key A						Access condition						Key B				Sector Trailer
32	80																	Data
	81																	Data
	--																	--
	--																	--
	--																	--
	8E																	Data
	8F	Key A						Access condition						Key B				Sector Trailer
	--																	--
	--																	--
	--																	--
39	F0																	Data
	--																	--
	--																	--
	FC																	Data
	FD																	Data
	FE																	Data
	FF	Key A						Access condition						Key B				Sector Trailer

Mifare Classic Memory Structure

Block 0 of sector 0 is reserved for manufacturer's data, and it is read-only.

Every sector trailer consists of:

- Key A (6B)
- Access Condition (4B)
- Key B (6B)

By default, the values are:

- Key A – FFFFFFFFFFFF

- Key B – FFFFFFFFFFFF
- Access Condition – FF 07 80 xx (Key A is not readable and is used to read or write, Key B as data)

4.2 Before we start

As mentioned, authentication is required before you can perform read and write access. If you want to read block 0 of sector 1, then first you need to load sector 1's key to reader. If both the reader's key and the card's key are matched, authentication will be successful. Else an error will be returned.

Some E-purse applications require Key A and B, for better security. In these applications, usually Key A is used to perform read, and decrease value, while Key B is for write, and increase value purposes. We will demo an example to illustrate this later.

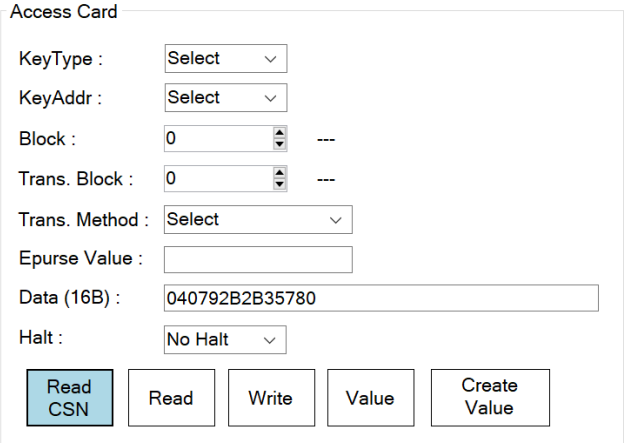
We will use sector 2, 3, and 4 in our examples. From the card memory structure, we know that:

- Sector 2 consists of block 8, 9, 10 and 11.
- Sector 3 consists of block 12, 13, 14, and 15.
- Sector 4 consists of block 16, 17, 18, and 19.

Block 11, 15, and 19 are the sector trailers, not for user memory. So we will perform basic read and write to blocks 8, 9, and 10. Then in the 'Appliction 1 (Key A)' example, we will use sector 3 to demo how to change Key. And in the 'Application 2 (Key A and B)', we will use sector 4 to demo how to change Access Condition, and enable Key B.

5. Read Card Serial Number

Place a Card on the USB reader, then click 'Read CSN' to read the unique serial number. This version supports both 4B and 7B serial number. There are displayed in Hex, and is displayed from LSB to MSB.



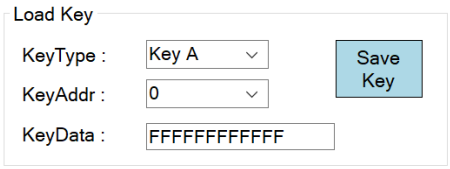
Click 

to read the card CSN.

Function : Classic Read CSN
7:02:07 PM, TX : 02 00 00 01 48 4B 03
7:02:07 PM, RX : 02 00 00 09 00 07 04 07 92 B2 B3 57 80 4B 03

6. Load Key

We will use sector 2 in our following examples. Before authentication can be successful, we need to load the authentication key to the reader. The key will be stored in non volatile memory.



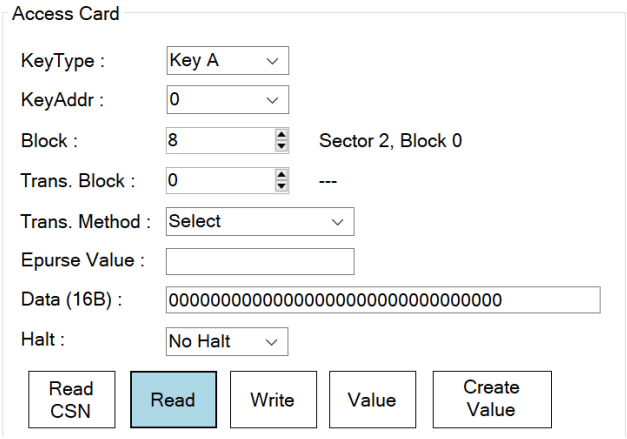
Click 

to load FFFFFFFFFFFF hex to Eeprom address 0, as Key A. Since Key B is disabled, we don't need to load for Key B now.

Function : Classic Save Key
7:03:21 PM, TX : 02 00 00 09 46 00 00 FF FF FF FF FF FF 4D 03
7:03:22 PM, RX : 02 00 00 01 00 03 03

7. Read

Since we have loaded the authentication Key, we can now read block 8 (Sector 2, block 0), using Key A from eeprom address 0. Bear in mind that, for new MF Classic card, Key A is FFFFFFFFFFFF hex, and Key B is disabled.



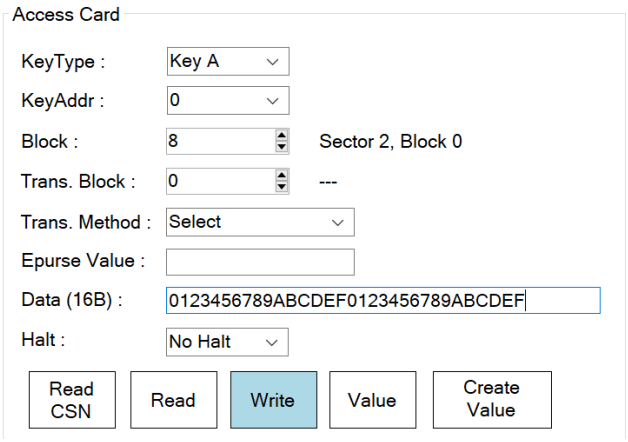
Click 

to read memory block 8 of the card. Authenticate the card using Key A of Eeprom address 0. The returned protocol shows the card serial number is 04 07 92 B2 B3 57 80, the card type is 08, and the card content of block 8 is 00000000000000000000000000000000 hex.

Function : Classic Read
7:03:59 PM, TX : 02 00 00 04 10 08 00 00 1E 03
7:04:00 PM, RX : 02 00 00 1A 00 07 04 07 92 B2 B3 57 80 08 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 50 03

8. Write

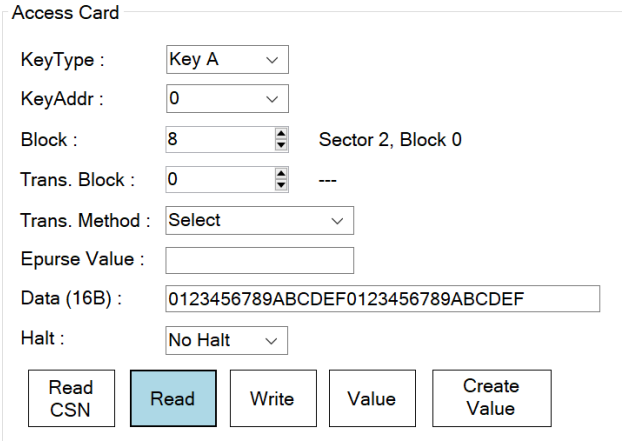
We will write 16B data to block 8 (Sector 2, block 0), and then read it out to verify.



Click 

to write 0123456789ABCDEF0123456789ABCDEF hex to block 8.

We will read out the block 8 to verify.



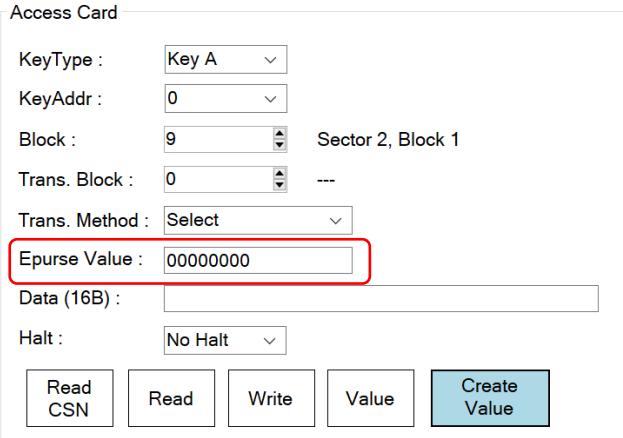
Click 

to read the content of block 8. The returned data shows the previous 'Write' was successful.

Function : Classic Write
8:19:28 PM, TX : 02 00 00 14 11 08 00 00 01 23 45 67 89 AB CD EF 01 23 45 67 89 AB CD EF 0F 03
8:19:28 PM, RX : 02 00 00 0A 00 07 04 07 92 B2 B3 57 80 08 40 03
Function : Classic Read
8:19:29 PM, TX : 02 00 00 04 10 08 00 00 1E 03
8:19:30 PM, RX : 02 00 00 1A 00 07 04 07 92 B2 B3 57 80 08 01 23 45 67 89 AB CD EF 01 23 45 67 89 AB CD EF 50 03

9. Create Value Block

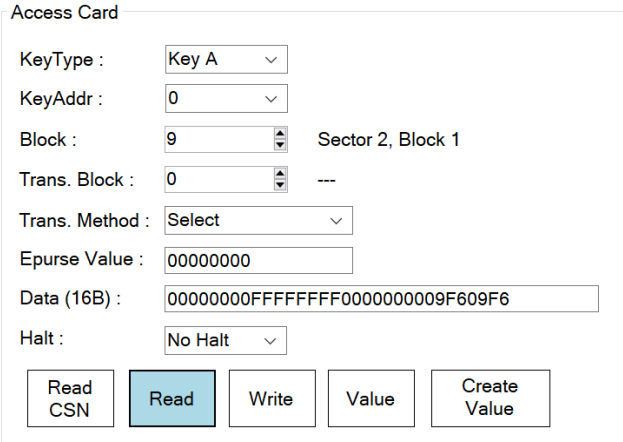
For E-Purse application, it is necessary to create value block. We will now convert block 9 (Sector 2, block 1) to value block. Since this block is still belongs to Sector 2, same authentication key, Key A from Eeprom address 0, will be used.



Assign a 4B init Epurse Value, and then click 

. In this example we set the init value as 00000000 hex.

Now, we will read block 9 to get the value.



Click 

to read the content of block 9. The returned data shows the previous 'Create Value Block' was successful. In the 16B returned data, the first 4B is the significant E-purse value, which is 00000000 hex.

Function : Classic Create Value Block
8:21:43 PM, TX : 02 00 00 14 11 09 00 00 00 00 00 00 FF FF FF FF 00 00 00 00 09 F6 09 F6 0E 03
8:21:43 PM, RX : 02 00 00 0A 00 07 04 07 92 B2 B3 57 80 08 40 03
Function : Classic Read
8:21:50 PM, TX : 02 00 00 04 10 09 00 00 1F 03
8:21:50 PM, RX : 02 00 00 1A 00 07 04 07 92 B2 B3 57 80 08 00 00 00 00 FF FF FF FF 00 00 00 00 09 F6 09 F6 50 03

10. Value - Increase

Since value block is created, we can now perform increment and decrement commands.

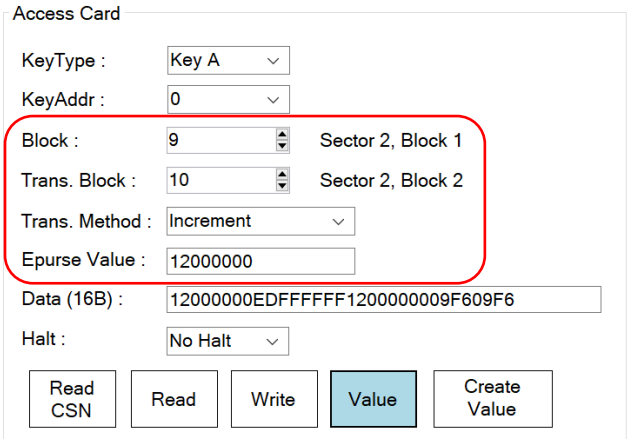
A typical E-Purse program flow for Increment or Decrement command is:

- Increase or decrease an amount from an original value block, and transfer the remaining value to a temp value block. Must be from the same sector
- Restore the remaining value from the temp value block to the original value block.
- When performing these steps, the card should remain in the RF field.

Block 9 is the original value block and 10 is the temp value block.

In the previous examples, we have programmed default value 00000000 hex to original block (Block 9). We will now do the first part; top up 12000000 hex, (LSB to MSB), to the value, and transfer the added value to temp value block (Block 10).

Item	Value
Block	9
Transfer Block	10 (0A hex)
Transfer Method	Increment
Epurse Value (4B)	12000000 hex



Click 

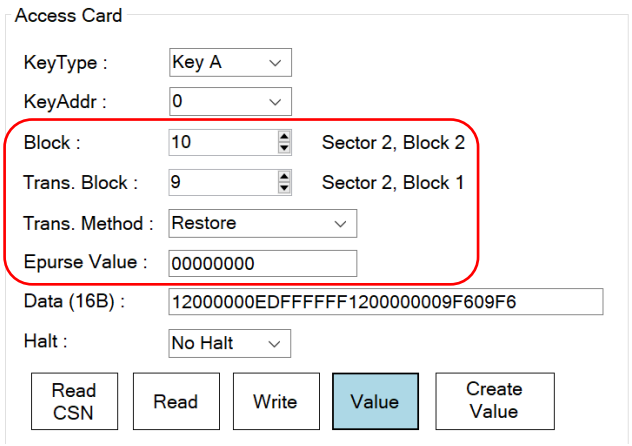
to increase 12000000 hex to the block 9 ‘s value, and transfer the result which is 12000000 hex to block 10. The returned data shows the process is successful.

At this stage, the original block still holds the original value, which is 00000000 hex, and not yet updated. The actual value is stored at temp value block.

Now, we will complete the second part; restore the value from the temp value block (Block 10) to original block (Block 9).

Item	Value
Block	10 (0A hex)
Transfer Block	9

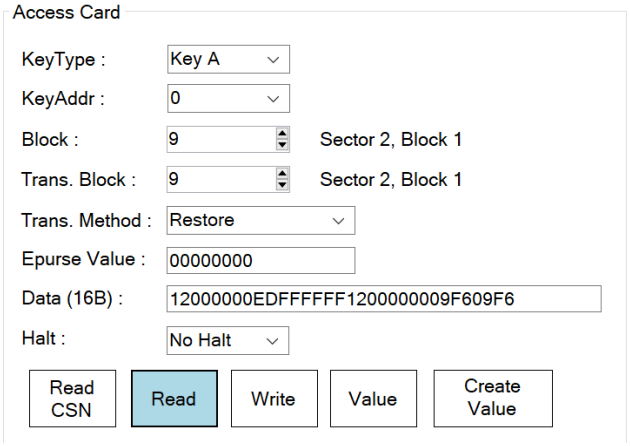
Transfer Method	Restore
Epurse Value (4B)	00000000 hex



Reset the Epurse Value to 00000000 hex, select 'Restore' in the option, and click 

.

Now we will read block 9 to verify the result.



Click 

to read block 9. The first 4B of the returned data is 12000000 hex, which is correct. Now, block 9 has been updated with the actual value.

Function : Classic Value
8:26:56 PM, TX : 02 00 00 0A 12 09 00 00 0A C1 12 00 00 00 CA 03
8:26:57 PM, RX : 02 00 00 1A 00 07 04 07 92 B2 B3 57 80 08 12 00 00 00 ED FF FF FF 12 00 00 00 09 F6 09 F6 42 03
Function : Classic Value
8:27:32 PM, TX : 02 00 00 0A 12 0A 00 00 09 C2 00 00 00 00 DB 03
8:27:32 PM, RX : 02 00 00 1A 00 07 04 07 92 B2 B3 57 80 08 12 00 00 00 ED FF FF FF 12 00 00 00 09 F6 09 F6 42 03
Function : Classic Read
8:27:54 PM, TX : 02 00 00 04 10 09 00 00 1F 03
8:27:54 PM, RX : 02 00 00 1A 00 07 04 07 92 B2 B3 57 80 08 12 00 00 00 ED FF FF FF 12 00 00 00 09 F6 09 F6 42 03

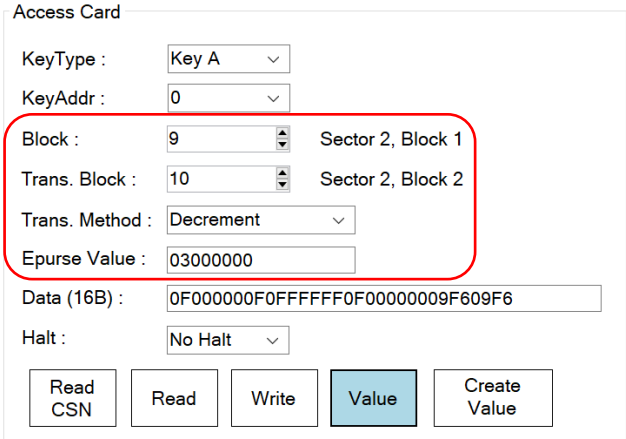
11. Value - Decrease

The process is same as Increase.

From the previous example, we know that block 9 has a value of 12000000 hex. Now, we will deduct 03000000 hex from this block. We will still use block 9 as original block, and block 10 as temp value block.

We will do the first part; deduct 03000000 hex, (LSB to MSB), to the value, and transfer the result to temp value block (Block 10).

Item	Value
Block	9
Transfer Block	10 (0A hex)
Transfer Method	Decrement
Epurse Value (4B)	03000000 hex

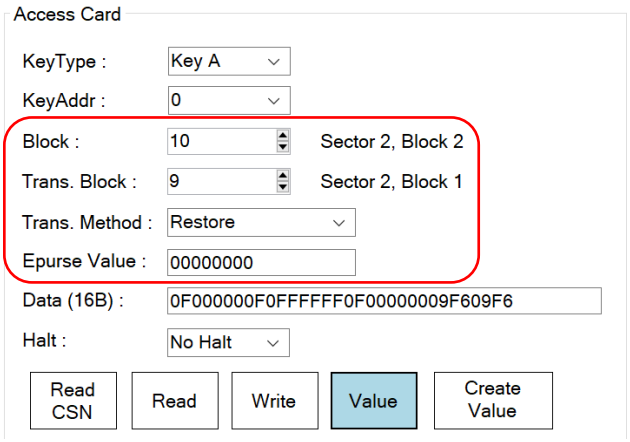


Click 

to decrease 03000000 hex to block 9's value, and transfer the result which is 0F000000 hex to block 10. The returned data shows the process is successful.

At this stage, the original block still holds the original value, which is 12000000 hex, and not yet updated. The actual value is stored at temp value block.

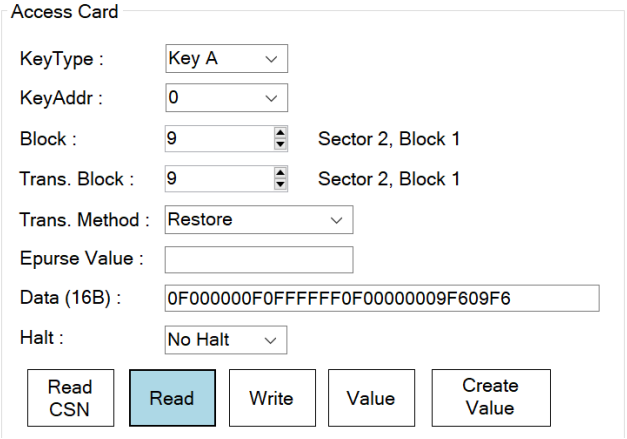
Now, we will complete the second part; restore the value from the temp value block (Block 10) to original block (Block 9).



Reset the Epurse Value to 00000000 hex, select 'Restore' in the option, and click 

.

Now we will read block 9 to verify the result.



Click 

to read block 9. The first 4B of the returned data is 0F000000 hex, which is correct. Now, block 9 has been updated with the resulted value.

Function : Classic Value
8:29:39 PM, TX : 02 00 00 0A 12 09 00 00 0A C0 03 00 00 00 DA 03
8:29:39 PM, RX : 02 00 00 1A 00 07 04 07 92 B2 B3 57 80 08 0F 00 00 00 F0 FF FF FF 0F 00 00 00 09 F6 09 F6 5F 03
Function : Classic Value
8:30:25 PM, TX : 02 00 00 0A 12 0A 00 00 09 C2 00 00 00 00 DB 03
8:30:25 PM, RX : 02 00 00 1A 00 07 04 07 92 B2 B3 57 80 08 0F 00 00 00 F0 FF FF FF 0F 00 00 00 09 F6 09 F6 5F 03
Function : Classic Read
8:30:34 PM, TX : 02 00 00 04 10 0A 00 00 1C 03
8:30:34 PM, RX : 02 00 00 1A 00 07 04 07 92 B2 B3 57 80 08 0F 00 00 00 F0 FF FF FF 0F 00 00 00 09 F6 09 F6 5F 03

12. Access Condition

The access bit control the rights of memory access using the keys A and B. We have developed a calculator to generate the 4B access condition value, based on selection.

There are 4 blocks in a sector in MF Classic card; 3 data blocks, and 1 sector trailer.

First we look at the access option for data blocks.

DataBlock:	Rd	Wr	Inc	Dec
000	A/B	A/B	A/B	A/B
010	A/B	X	X	X
100	A/B	B	X	X
110	A/B	B	B	A/B
001	A/B	X	X	A/B
011	B	B	X	X
101	B	X	X	X
111	X	X	X	X

Access option for Data Block.

We can define how the data block can be accessed, or using which Key to access it, and what function can the key perform.

If we choose 000 for a data block, it means Key A or Key B can be used to perform Read, Write, Increment and Decrement. If we choose 100 for a data block, it means Key A or B can be used to perform Read, Key B can perform Write, but Increment and Decrement are prohibited.

By default, 000 is chosen for all data blocks in a new MF Classic card.

Now we look at the access option for sector trailer.

SecTrailer:	Key A		Acc Bits		Key B	
	Rd	Wr	Rd	Wr	Rd	Wr
000	X	A	A	X	A	A
010	X	X	A	X	A	X
100	X	B	A/B	X	X	B
001	X	A	A	A	A	A
011	X	B	A/B	B	X	B
101	X	X	A/B	B	X	X

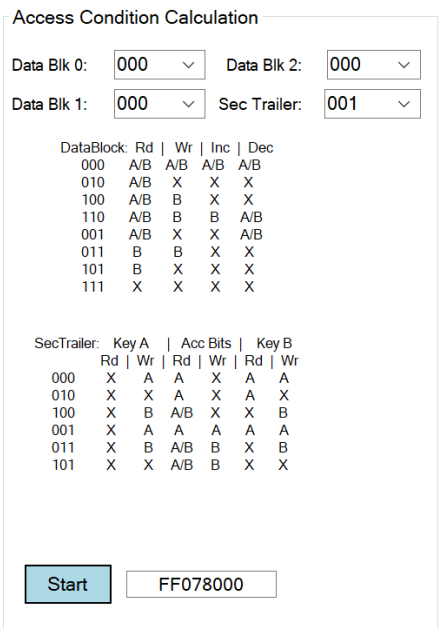
Access option for Sector Trailer.

This access bit control whether Key A, Key B and the Access Condition can be read or changed.

If we choose 000 for a sector, it means Key A can be used to change Key A, read Access Bits, to read and write Key B. if we choose 011 for a sector, it means Key A can be used to read Access Bits only, but Key B can be used to write Key A, read and write Access Bits, and write Key B.

By default, 001 is chosen for all sectors in a new MF Classic card.

We will now use the calculator to compile the value



Click  to generate the access condition for the sector.

The value is FF078000, which is same as FF0780xx or FF078069. The last two digits are insignificant. This access condition means

- Key A is not readable,
- Key A is used to read or write,
- Key B is disabled.

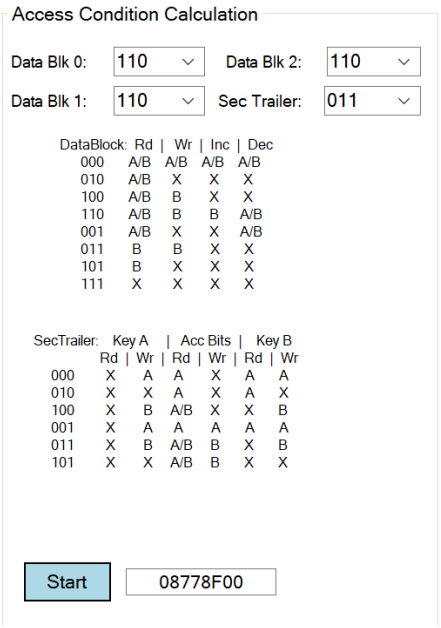
This is the default access condition for a new MF Classic card.

Now we will generate another access condition, which is common, and used in our 'Application 2 (Key A and B)' example. This new access condition will

- enable Key A and B,
- Key A is used for Read / Decrement,
- Key B for All functions.

To fulfill this requirement, we choose 110 for all data blocks, and 011 for sector trailer.

We use calculator to compile the value.



Click  to generate the access condition for the sector.

The value is 08778F00. We will use this access condition in the ‘Application 2 (Key A and Key B)’ example later.

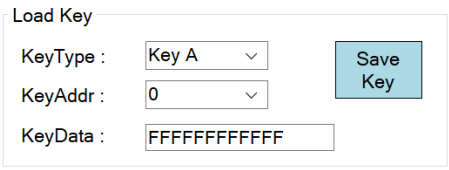
13. Application 1 (Key A) – Change Key

Most application will not use the default Key FFFFFFFFFFFF hex. Thus, card personalization is needed before project deployment. Here, we will show you how to change the card default key A to a new Key A 010203040506 hex, in sector 3.

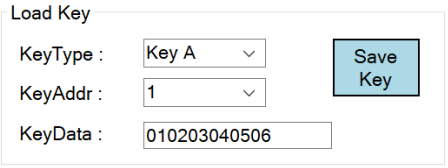
Step 1:

Load the following key to reader:

- Default key A FFFFFFFFFFFF to KeyAddr 0,
- New Key A 010203040506 to KeyAddr 1.



Load default Key A FFFFFFFFFFFF to Keyaddr 0.



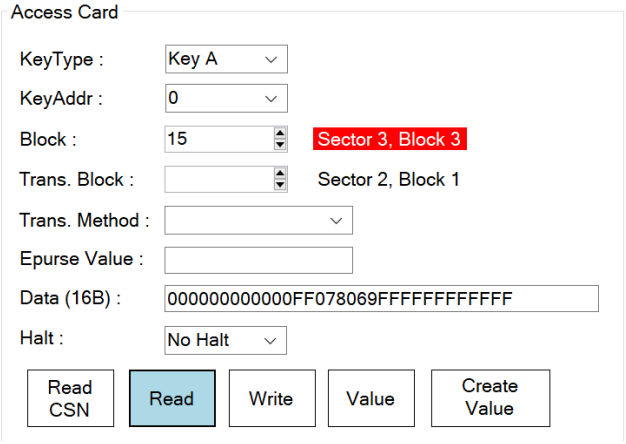
Load new Key A 010203040506 to Keyaddr 1.

Step 2:

The default access condition for a new MF Classic card is FF 07 80 69, which briefly means: Enable Key A for All functions. Please refer card manufacturer's manual for more details. We will use this default access condition in this example.

You may get the access condition by reading the sector trailer, as shown below:

Read the sector trailer of sector 3, which is Block 15, to get the access condition.



Click 

to read Sector3 ‘s Sector Trailer.

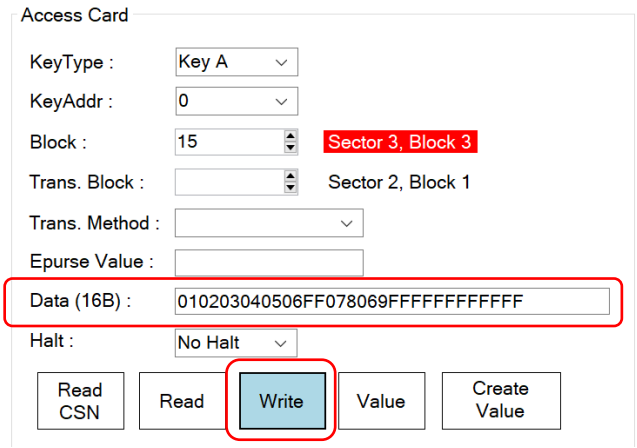
The returned 16B card content is:

- 000000000000 hex (6B Key A, the real Key is hidden, unreadable. Enabled)
- FF 07 80 69 hex (4B Access Condition, refer card manufacturer’s manual for details.)
- FFFFFFFFFFFF hex (6B Key B, since this is readable, Key B is disabled.)

Step 3:

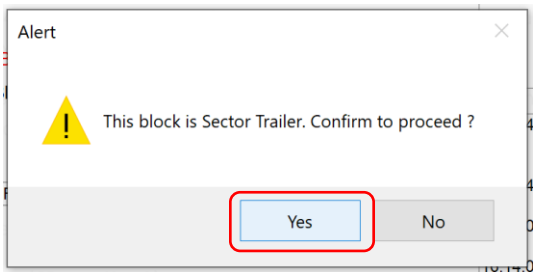
Program the Sector Trailer with new Key:

- 010203040506 hex (6B new Key A)
- FF 07 80 69 hex (4B Access Condition)
- FFFFFFFFFFFF hex (6B default Key B, can be used as data storage.)



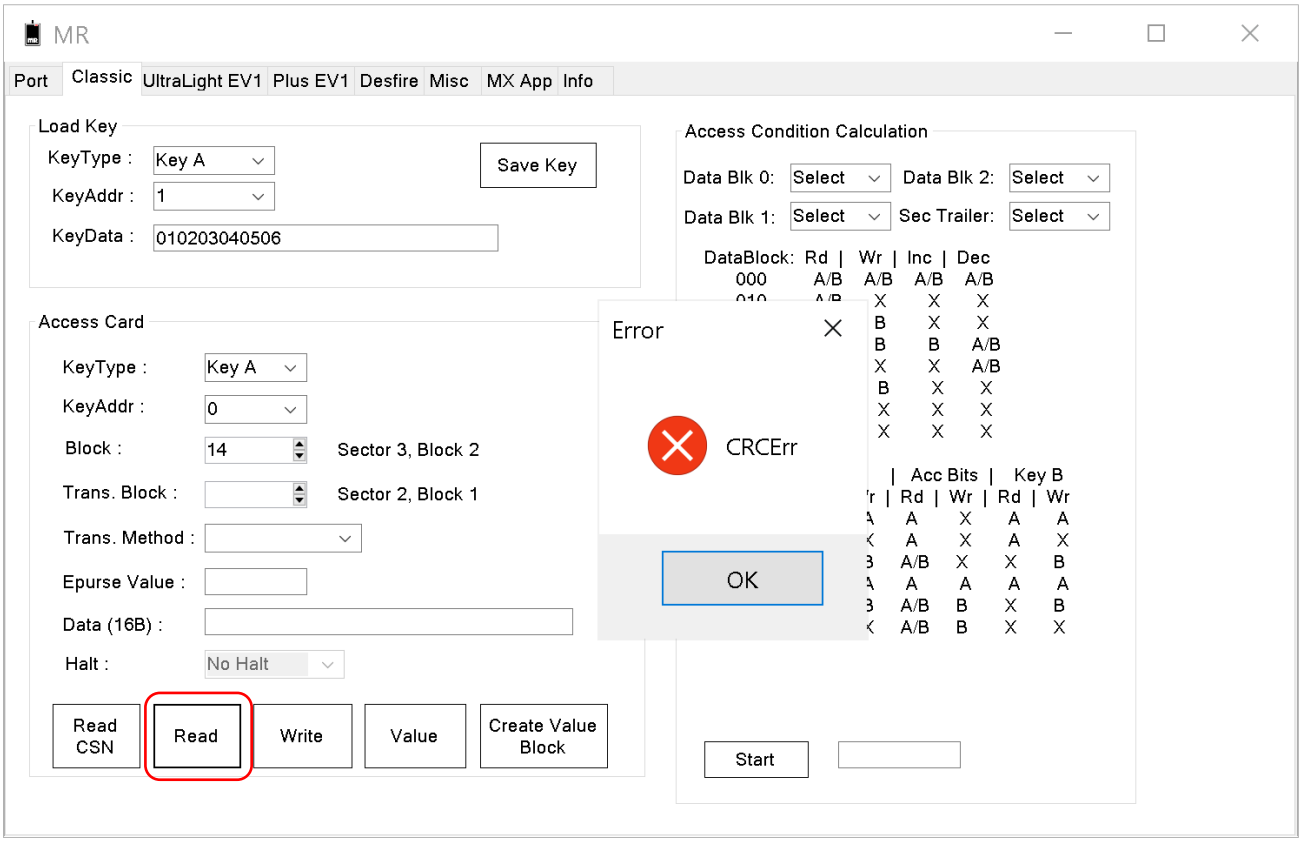
Click 

to program new Key A to block 15, sector 3, block 3.



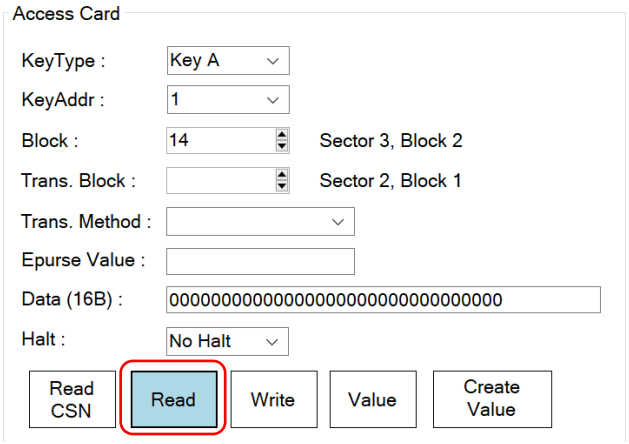
Click ‘Yes’ to write to sector trailer.

After the change, Sector 3 can no longer authenticated by using the default Key A FFFFFFFFFFFF hex. It will return error, as shown below:



Click  to read Block 14 (Sector 3 Block 2) with Key A, Keyaddr 0. Not successful. You may need to reset the RF or card to view this.

It will succeed if it is authenticated with the new Key A 010203040506 hex, the key that we have loaded in Keyaddr 1.



Click  to read Block 14 (Sector 3 Block 2) with Key A, Keyaddr 1. Successful.

Now we have changed the Key A of Sector 3, from default key FFFFFFFFFFFF hex to 010203040506 hex. Whenever we want to access sector 3 (Block 12, 13, 14, and 15), we will need to use this key.

Function : Classic Save Key
8:39:39 PM, TX : 02 00 00 09 46 00 00 FF FF FF FF FF FF 4D 03
8:39:39 PM, RX : 02 00 00 01 00 03 03
Function : Classic Save Key
8:39:47 PM, TX : 02 00 00 09 46 01 00 01 02 03 04 05 06 4B 03
8:39:47 PM, RX : 02 00 00 01 00 03 03
Function : Classic Read
8:43:23 PM, TX : 02 00 00 04 10 0E 00 00 18 03
8:43:23 PM, RX : 02 00 00 01 02 01 03
Function : Classic Read
8:44:06 PM, TX : 02 00 00 04 10 0E 01 00 19 03
8:44:06 PM, RX : 02 00 00 1A 00 07 04 07 92 B2 B3 57 80 08 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 50 03

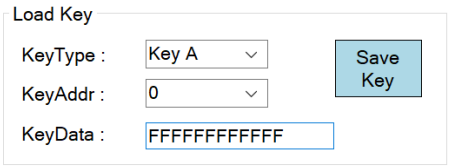
14. Application 2 (Key A and Key B) – Change Access Condition

Some application, especially the E-Purse applications may require higher security, that involve Key A and B. Thus, card personalization is needed before project deployment. Here, we will show you how to change the card default key A to a new Key A 010203040506 hex, and enable the Key B with key 0A0B0C0D0E0F hex, for sector 4.

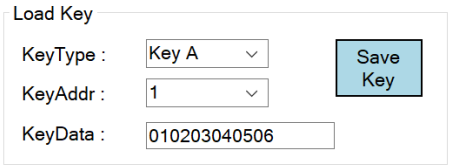
Step 1:

Load the following key to reader:

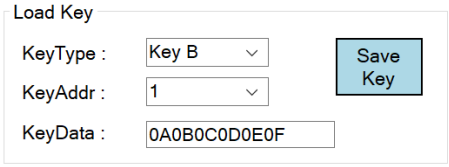
- default key A FFFFFFFFFFFF to Keyaddr 0,
- new Key A 010203040506 to Keyaddr 1,
- new Key B 0A0B0C0D0E0F to Keyaddr 1, (select KeyType Key B).



Load default Key A FFFFFFFFFFFF to Keyaddr 0.



Load new Key A 010203040506 to Keyaddr 1.



Load new Key B 0A0B0C0D0E0F to Keyaddr 1.

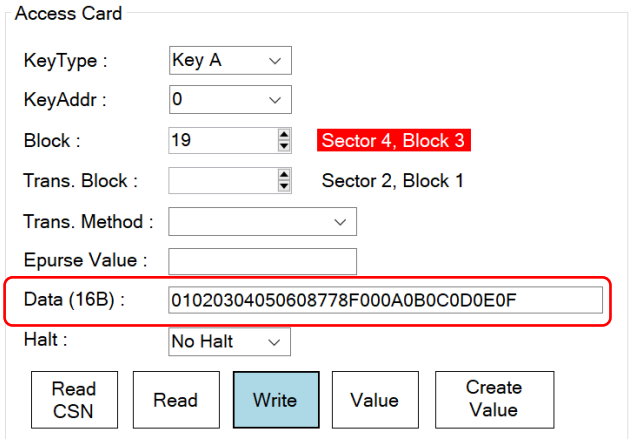
Step 2:

We will use these values for our new access condition 08 77 8F 00, which briefly means: Enable Key A and B, where Key A is used for Read / Decrement, and Key B for All functions. Please refer card manufacturer's manual for more details.

Step 3:

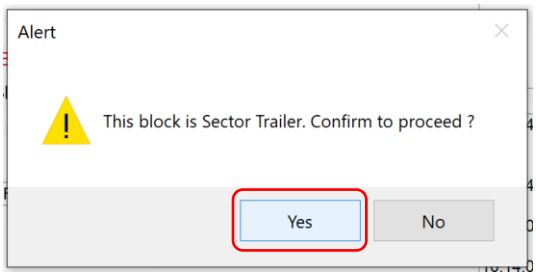
Program the Sector Trailer with new Key and access condition:

- 010203040506 hex (6B new Key A)
- 08 77 8F 00 hex (4B Access Condition)
- 0A0B0C0D0E0F hex (6B new Key B.)



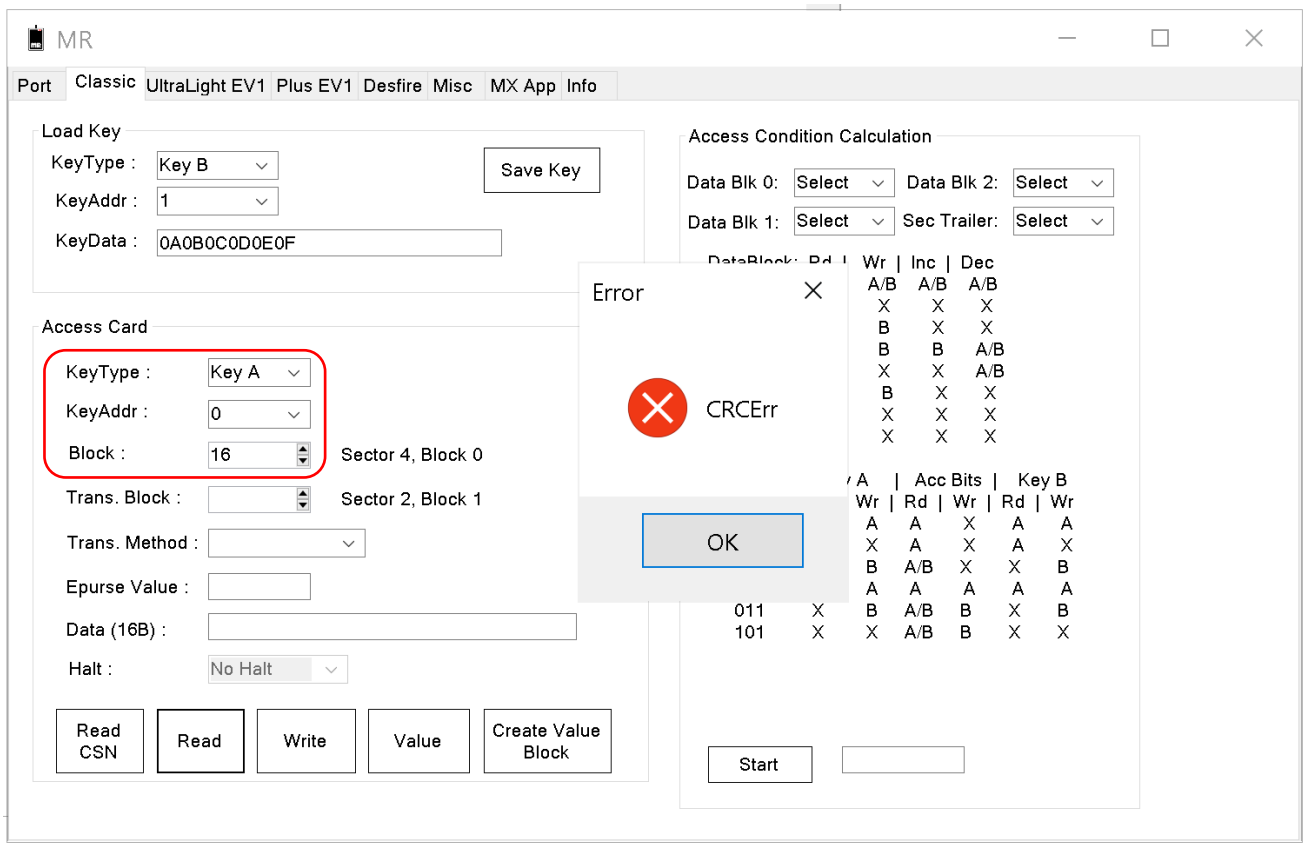
Click 

to program new Key A, Key B, and access condition to Block 19 (Sector 4, Block 3).



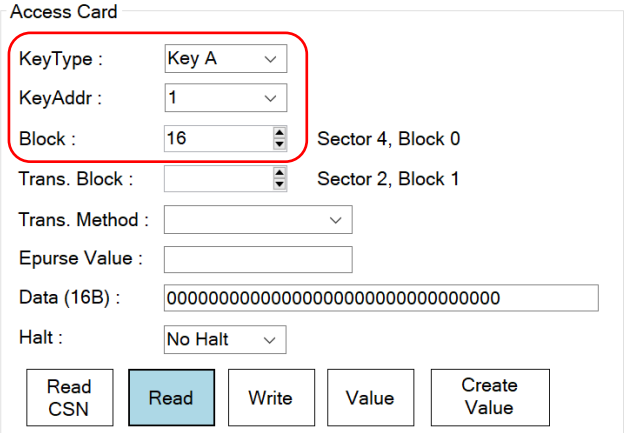
Click 'Yes' to write to sector trailer.

After the change, Sector 4 can no longer be authenticated using the default Key A FFFFFFFFFFFF hex. It will return error, as shown below:

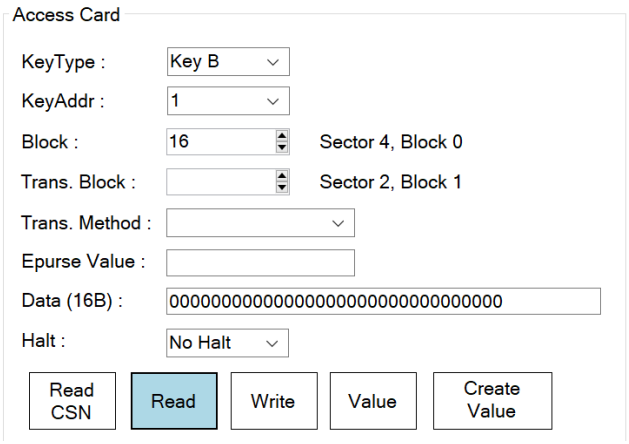


Click  to read block 16 (Sector 4, block 0), not successful. You may need to reset RF or card to view this.

It will succeed if it is authenticated with the new Key A 010203040506 hex, or Key B 0A0B0C0D0E0F hex, which we have loaded in Keyaddr 1.



Click  to read Block 16 (Sector 4 Block 0) with Key A, Keyaddr 1. Successful.



Click 

to read Block 16 (Sector 4 Block 0) with Key B, Keyaddr 1. Successful.

We have programmed the sector 4 trailer to limit its access to certain functions. It should be allowed to perform Read / Decrement using Key A, and perform all function using Key B. We will try them now.

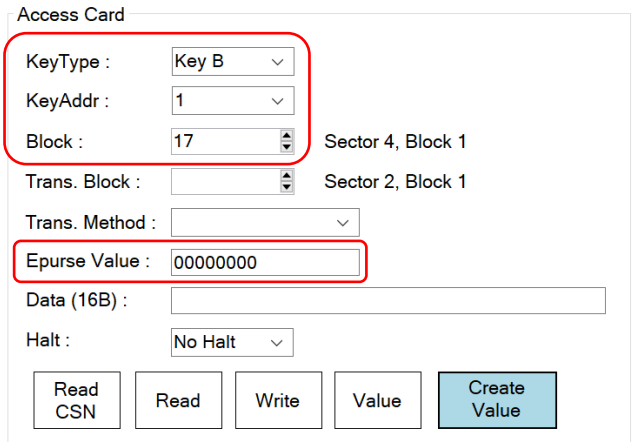
We are going to test this sector 4 now by performing a top-up, and then a deduct value. There are few steps in the top-up:

- Step 1, use Key B to create a value block, (Key A is prohibited to do 'Write', skip this step if the block is already a value block),
- Step 2, use Key B to increase a value of 35000000 (LSB to MSB) hex to block 17, (Key A is prohibited to do 'Increment').

And then the deduct value involves few steps too:

- Step 3, use Key A to deduct 10000000 (LSB to MSB) hex from block 17,
- Step 4, use Key A to read block 17.

Step 1: Use Key B to create a value block at block 17.

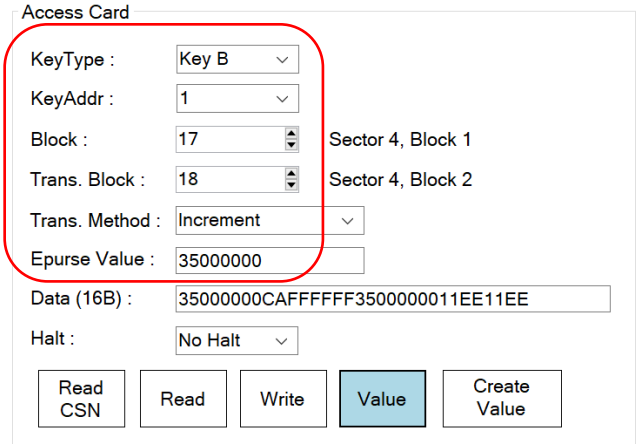


Click 

to convert block 17 to value block. Successful!

Step 2: Use Key B to top up 35000000 hex to block 17.

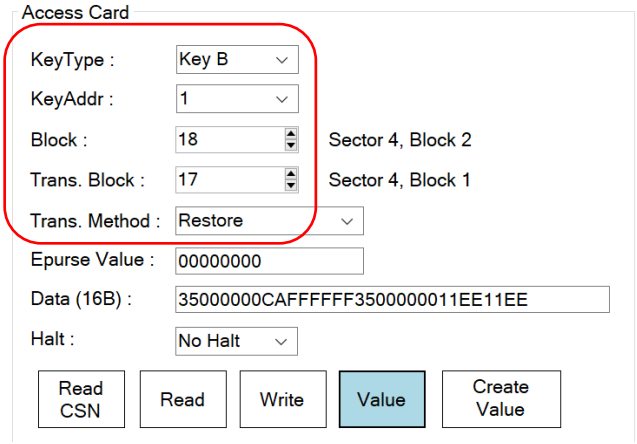
Block 17 is the original block, and block 18 is the temp value block.



Click 

to top up 35000000 hex to block 17, but transfer the balance to block 18. Successful!

Now we restore the value from block 18 to block 17.



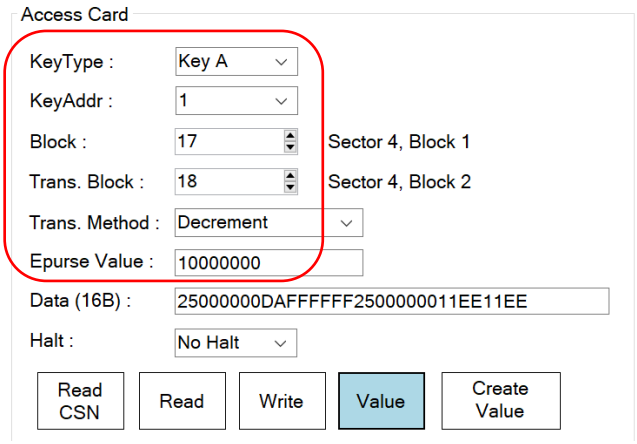
Click 

to restore the value from block 18 to block 17. Successful!

Now block 17 has a value of 35000000 hex. You may read block 17 to verify the content.

Having done the top-up with Key B, we will now use Key A to deduct a value from Block 17.

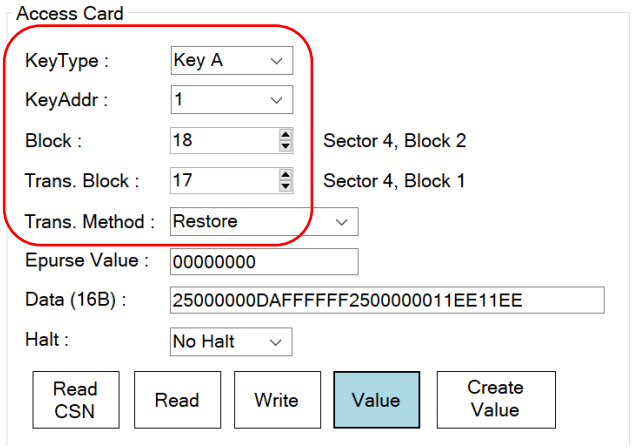
Step 3: Deduct 10000000 hex from Block 17, and transfer the balance to Block 18.



Click 

to deduct 10000000 hex from block 17, and transfer the resulted value to block 18. Successful!

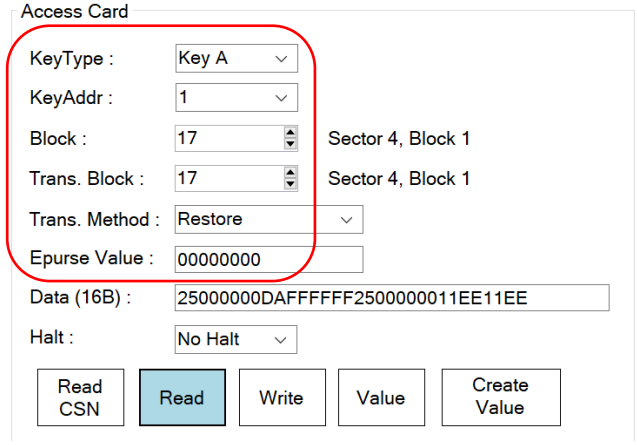
Now we restore the value from block 18 to block 17.



Click 

to restore value from block 18 to block 17. Successful!

Step 4: Now block 17 has a value of 25000000 hex. We will read block 17 to verify the content.



Click 

to read block 17. The data shows the value is 25000000 hex, which is correct.

The combined use of Key A and B, enhance the security of application. It should be adopted in all MF Classic application.

Function : Classic Save Key
8:50:01 PM, TX : 02 00 00 09 46 00 00 FF FF FF FF FF FF 4D 03
8:50:01 PM, RX : 02 00 00 01 00 03 03
Function : Classic Save Key
8:50:12 PM, TX : 02 00 00 09 46 01 00 01 02 03 04 05 06 4B 03
8:50:12 PM, RX : 02 00 00 01 00 03 03
Function : Classic Save Key
8:50:22 PM, TX : 02 00 00 09 46 01 01 0A 0B 0C 0D 0E 0F 4C 03
8:50:22 PM, RX : 02 00 00 01 00 03 03
Function : Classic Write
8:51:33 PM, TX : 02 00 00 14 11 13 00 00 01 02 03 04 05 06 08 77 8F 00 0A 0B 0C 0D 0E 0F E2 03
8:51:33 PM, RX : 02 00 00 0A 00 07 04 07 92 B2 B3 57 80 08 40 03
Function : Classic Read
8:52:04 PM, TX : 02 00 00 04 10 10 00 00 06 03
8:52:04 PM, RX : 02 00 00 01 02 01 03
Function : Classic Read
8:53:28 PM, TX : 02 00 00 04 10 10 01 00 07 03
8:53:28 PM, RX : 02 00 00 1A 00 07 04 07 92 B2 B3 57 80 08 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 50 03
Function : Classic Read
8:53:39 PM, TX : 02 00 00 04 10 10 01 01 06 03
8:53:39 PM, RX : 02 00 00 1A 00 07 04 07 92 B2 B3 57 80 08 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 50 03
Function : Classic Create Value Block
8:55:45 PM, TX : 02 00 00 14 11 11 01 01 00 00 00 00 FF FF FF FF 00 00 00 00 11 EE 11 EE 16 03
8:55:45 PM, RX : 02 00 00 0A 00 07 04 07 92 B2 B3 57 80 08 40 03
Function : Classic Value
8:56:11 PM, TX : 02 00 00 0A 12 11 01 01 12 C1 35 00 00 00 ED 03
8:56:11 PM, RX : 02 00 00 1A 00 07 04 07 92 B2 B3 57 80 08 35 00 00 00 CA FF FF FF 35 00 00 00 11 EE 11 EE 65 03
Function : Classic Value
8:56:29 PM, TX : 02 00 00 0A 12 12 01 01 11 C2 00 00 00 00 DB 03
8:56:29 PM, RX : 02 00 00 1A 00 07 04 07 92 B2 B3 57 80 08 35 00 00 00 CA FF FF FF 35 00 00 00 11 EE 11 EE 65 03
Function : Classic Value
8:57:07 PM, TX : 02 00 00 0A 12 11 01 00 12 C0 10 00 00 00 C8 03
8:57:07 PM, RX : 02 00 00 1A 00 07 04 07 92 B2 B3 57 80 08 25 00 00 00 DA FF FF FF 25 00 00 00 11 EE 11 EE 75 03
Function : Classic Value
8:57:33 PM, TX : 02 00 00 0A 12 12 01 00 11 C2 00 00 00 00 DA 03
8:57:33 PM, RX : 02 00 00 1A 00 07 04 07 92 B2 B3 57 80 08 25 00 00 00 DA FF FF FF 25 00 00 00 11 EE 11 EE 75 03
Function : Classic Read
8:57:50 PM, TX : 02 00 00 04 10 11 01 00 06 03
8:57:50 PM, RX : 02 00 00 1A 00 07 04 07 92 B2 B3 57 80 08 25 00 00 00 DA FF FF FF 25 00 00 00 11 EE 11 EE 75 03

Remarks:

In this sector 4, since we have changed the sector trailer, only Key B is allowed to change the sector's Key A and B. Therefore, remember to load the correct Key B to reader, and select Key B for KeyType, when you want to change key.

15. End