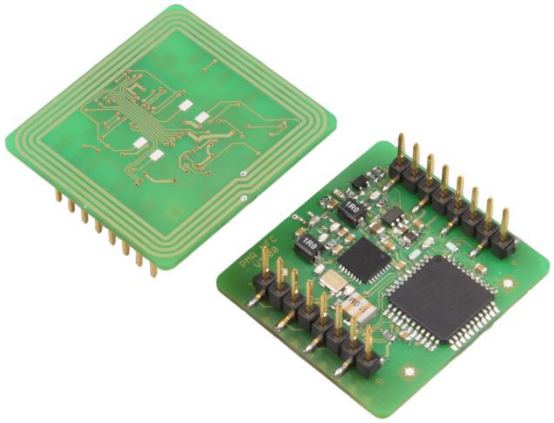
ELATEC

RFID Systems



Transponder Reader MIFARE® NFC Technical Manual

Doc.-Rev. 1.02

Content

1. Introduction

This document is the reference guide for the transponder reader family TWN3 MIFARE® NFC and Mini Reader MIFARE® NFC. The readers are using the same reading technology, so this document is applicable for both devices.

Note:

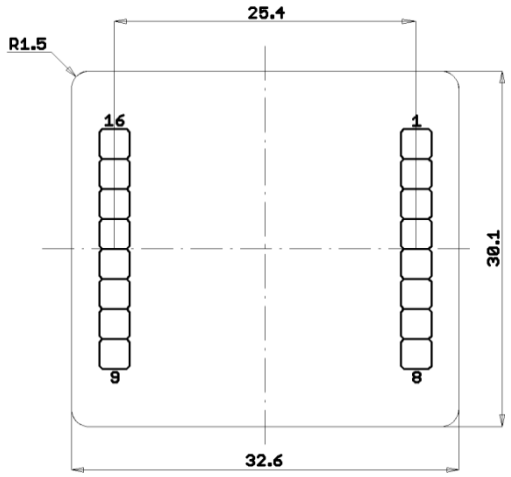
In order to use the functionality, which is described in this document, your MIFARE® NFC reader needs a firmware version V1.07 or above. In order to update the firmware from an elder version, please refer to chapter 11 "Firmware Update".

2. Installation of Mini Reader MIFARE® NFC

This chapter covers the installation of Mini Reader MIFARE® NFC in an embedded environment.

2.1 Mechanical Outline

All dimensions are in mm.



Component side shown, Pin spacing: 2.54mm

2.2 Pinning

Pin	Name	Description	Pin char.	Pin	Name	Description	Pin char.
1	$\overline{\text{RESET}}$	Asynchronous reset	I / PU	9	VSAM	3.0V regulated supply for SAM	S
2	$\overline{\text{PWRDWN}}$	Hard power down	I / PU	10	SAM_IO	Bidirectional SAM I/O line	IO / PU
3	GND	Ground	S	11	GPIO3	General purpose input/output 3	IO
4	VCC	3.3V – 5V	S	12	GPIO2	General purpose input/output 2	IO
5	RXD/MOSI	UART/SPI receiver input	I / PU	13	GPIO1	General purpose input/output 1	IO
6	TXD/MISO	UART/SPI transmitter output	O	14	GPIO0	General purpose input/output 0	IO
7	SCK	SPI serial clock	I	15	SAM_CLK	SAM clock	O
8	$\overline{\text{SS}}$	SPI slave select	I / PU	16	SAM_RST	SAM reset	O

I: Input
O: Output
IO: Input / Output
S: Supply
PU: Integrated Pull-up resistor

2.3 Electrical Characteristics

Frequency	13.56 MHz	
Power supply	3.15V - 5.5V DC	
Current consumption	RF field off: RF field on: Power Down:	10mA Typically 80mA < 2µA

2.4 Asynchronous Serial Connection (UART)

Because Mini Reader MIFARE® NFC is transmitting and receiving TTL levels, it can be directly connected to a microcontroller. If you plan to run the reader at a PC, an appropriate interface converter circuit must be connected.

By default, the reader starts communication at 9600 baud, 8 databits, 1 stopbit, no parity. The baudrate can be increased up to 115200 baud in order to achieve a higher communication speed.

Asynchronous Serial Connection is also the first offered communication interface after power-up or reset of the reader. This means, the reader starts operation in asynchronous mode, as soon as $\overline{SS}$ is tied to low, synchronous mode is selected. The pin $\overline{SS}$ can be left floating if synchronous mode is not required.

2.5 Synchronous Serial Connection (SPI)

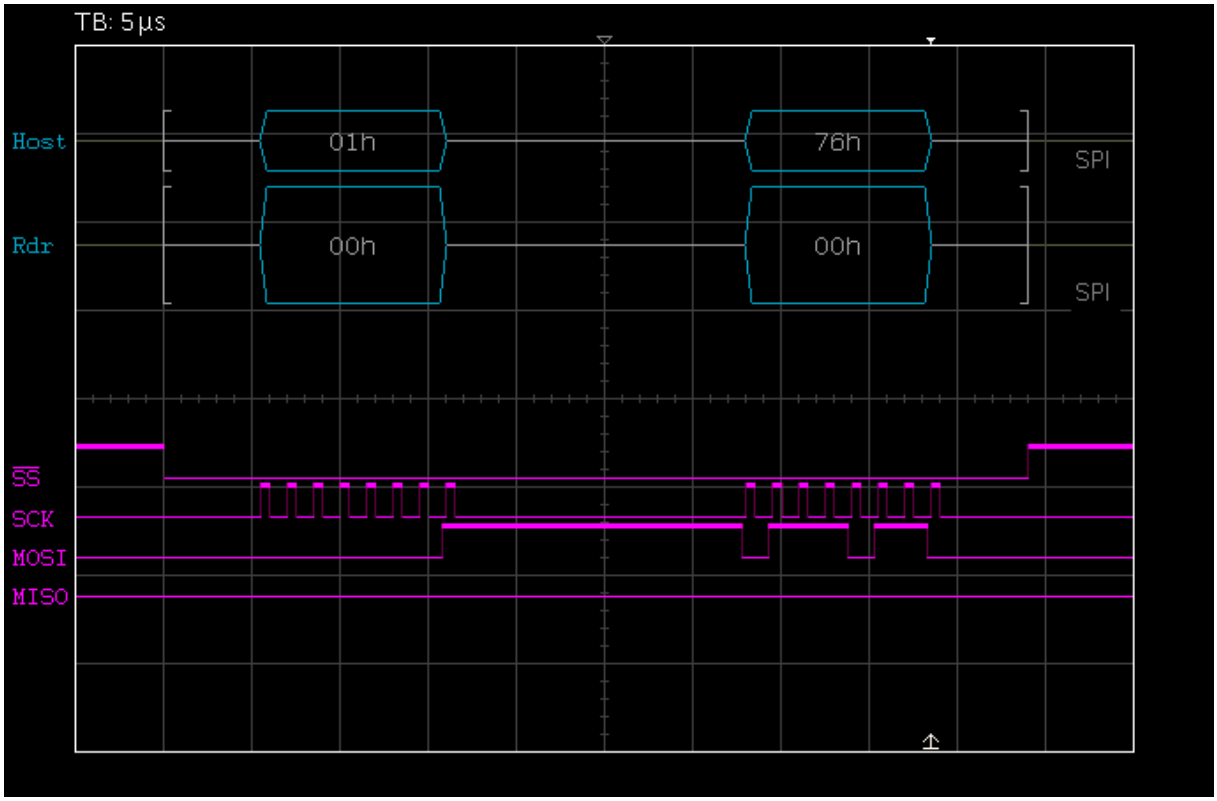
Mini Reader MIFARE® NFC also offers synchronous data transfer via a 4-line SPI. In this case, the reader is always slave, it cannot initiate any data transfer by itself. All data transfer is initiated by the master by pulling the pin $\overline{SS}$ to low. Data is transmitted from the master to the reader on signal MOSI, the reader transmits its response data on signal MISO. The serial clock is issued on signal SCK. All data bytes are transferred MSB first. A data bit is sampled at the rising edge of SCK. A SPI data transfer is completed by releasing the pin $\overline{SS}$ to inactive state.

After activation of $\overline{SS}$ the SPI unit needs a setup time (see 1.5.1, marking 1) before the serial clock is issued in order to handle any incoming or outgoing data transfer. When $\overline{SS}$ is active, the master may send an arbitrary count of data bytes to the reader. It is recommended to deactivate $\overline{SS}$ when a data frame has been transmitted completely.

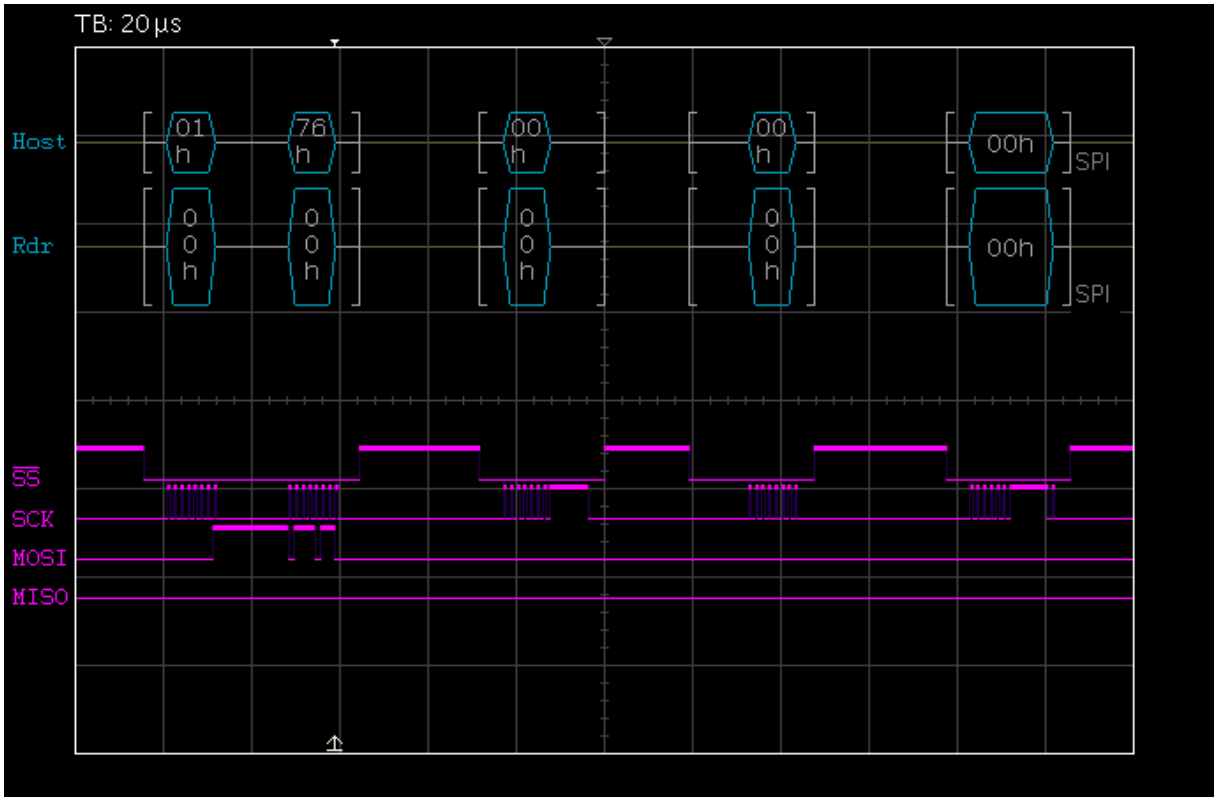
A data byte takes 8 SCK pulses. If $\overline{SS}$ is deactivated before 8 SCK pulses have been issued, the receiver circuit is reset and the partial received data byte is dropped.

SPI communication works data frame oriented, this means that the first byte received by the reader is always interpreted as a “length-byte”; it represents the count of data bytes which will follow. On the same way, the reader transmits its “length-byte” first and tells the master, how many bytes will follow. If one of the communication entities sets the length-byte to zero, it signals that it has no data to send.

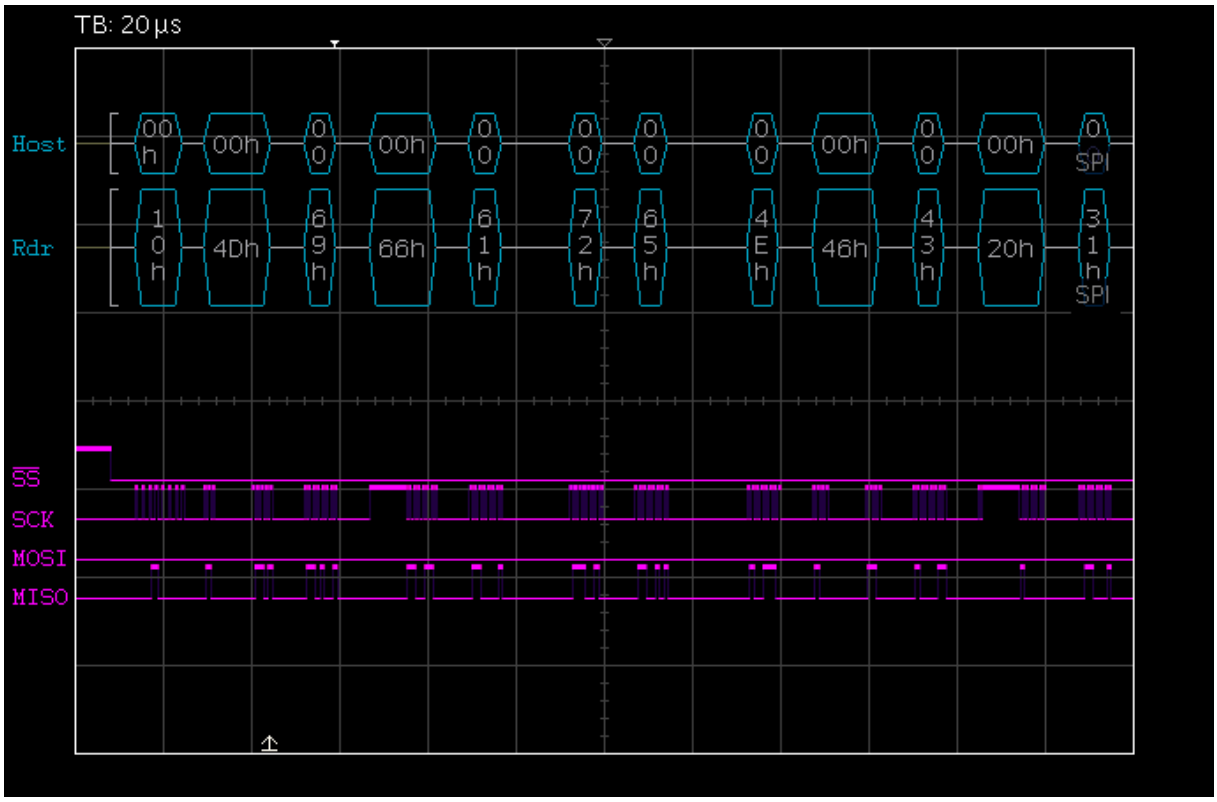
The following images show an exemplary SPI transfer. Hereby, the command ‘Get Version’ (‘v’) is sent to the reader. The master transmits the length-byte 0x01 followed by the command-code 0x76 (ASCII ‘v’) to the reader:



After sending the command frame, the master polls the reader for the response. This is done by cyclically sending 0x00 frames:



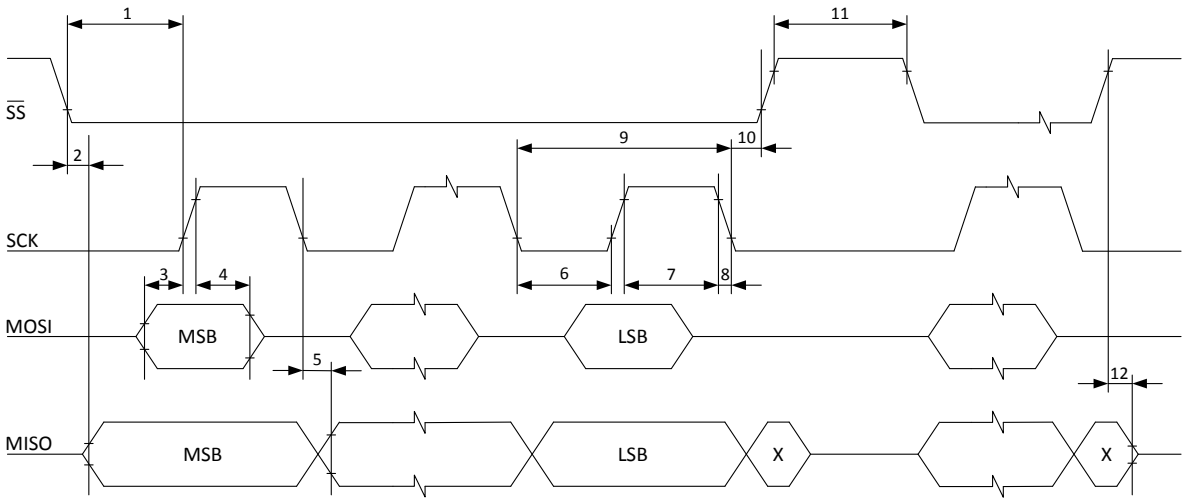
Every command requires its individual processing time until a response is available. This response time also depends on the response polling frequency of the master, so it might be useful to slow down polling in order to increase processing speed of the reader. As soon as there is a response available, the reader sends its response to the master, when polled:



The previous image shows the response of the reader, in this case it is the ASCII-representation of the version string “MifareNFC 1.02<CR><LF>” (with leading length-byte 0x10).

2.5.1 SPI Timing

SPI Interface: Timing requirements. Drawing is not for scale!



Marking	Description	Min	Typ	Max
1	$\overline{SS}$ low to start of clocking	5µs		
2	$\overline{SS}$ low to data output		15ns	
3	Data input setup time	10ns		
4	Data input hold time	90ns		
5	SCK to data output		15ns	
6	SCK low	182ns		
7	SCK high	182ns		
8	SCK rise/fall time			1600ns
9	SCK period	364ns		
10	SCK to $\overline{SS}$ high	4µs		
11	$\overline{SS}$ inactive time	8µs		
12	$\overline{SS}$ high to tri-state		10ns	

Please note: The reader requires a boot time of at least 5ms until full SPI functionality is provided. The boot time is adjustable by the EEPROM to higher values. It is good practice for the master, to issue a hard reset after power-up while $\overline{SS}$ remains inactive and SCK / MOSI have a logic low level. After the boot time has elapsed, $\overline{SS}$ may be tied to low for the first time in order to read the version string.

2.5.2 SPI Software Implementation

The following code demonstrates how to implement SPI communication in software for the host system, e.g. for a microcontroller:

```
byte SPI_ReadWrite(byte Data)
{
	byte i;

	for (i=0; i<8; i++)
	{
		if (Data & 0x80)
			MOSI_HIGH;
		else
			MOSI_LOW;
		SCK_HIGH;
		Data <<= 1;
		if (READ_MISO)
			Data |= 0x01;
		SCK_LOW;
	}
	return Data;
}

int main(void)
{
	byte k;
	byte q;
	byte LengthByteRead;
	byte LengthByteWrite;

// ...

	while(1)
	{
		LengthByteWrite = GetNumBytesInSendBuffer();
		SS_LOW;
		LengthByteRead = SPI_ReadWrite(LengthByteWrite);
		while ((LengthByteRead > 0) || (LengthByteWrite > 0))
		{
			if (LengthByteWrite > 0)
			{
				q = GetByteFromSendBuffer();
				LengthByteWrite--;
			}
			else
				q = 0x00;
			k = SPI_ReadWrite(q);
			if (LengthByteRead > 0)
			{
				LengthByteRead--;
				PushByteToReceiveBuffer(k);
			}
			else
				LengthByteRead = k;
		}
		SS_HIGH;
	}
// ...
}
```

2.6 Usage of GPIOs

The reader provides four general purpose I/Os that can be configured individually. These I/Os can be read and written by commands.

Please consider, that the GPIOs have limited current source and sink capability of max. 25mA. An overcharge of GPIO-pins can damage the module!

2.7 Asynchronous Reset

In usual operating environments, the reset pin can be left floating. For an asynchronous hardware reset pull the reset pin to a logic low level. The reader continues operation as soon as the reset pin is released.

2.8 Power Supply

A power supply of 3.3V – 5V must be applied between the VCC and GND pins. Please make sure that the power supply provides a current capability of at least 100mA.

Please note: The current consumption can be significantly increased if GPIOs and/or SAM module are used. The total current consumption may not exceed 150mA.

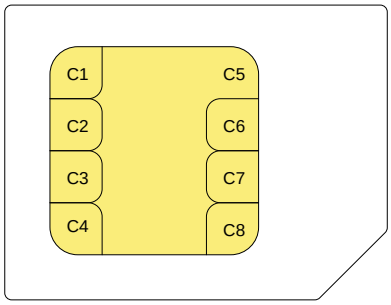
2.9 Hard Power Down

The reader supports a hardware power down feature which enables the user to turn off the entire reader. Hard power down mode is entered by pulling the $\overline{\text{PWRDWN}}$ pin to low; this results in turning off the internal voltage regulator.

Please note: Maximum power saving can be achieved by entering this mode, but this also means that the reader needs the boot time until it is available again.

2.10 SAM

The reader supports connection of ISO7816 compatible SAM cards. The lines SAM_IO, SAM_CLK, SAM_RST, VSAM and GND have to be connected as follows:

SAM card	Pad	Line
	C1	VSAM
	C2	SAM_RST
	C3	SAM_CLK
	C4	Leave open
	C5	GND
	C6	Leave open
	C7	SAM_IO
	C8	Leave open

3. Supported Transponders

ISO14443A	ISO14443B	ISO18092
MIFARE - Classic 1k, 4k - DESFire EV1 - Mini - Plus S, X - Pro X - SmartMX - Ultralight - Ultralight C Legic Advant[1] SLE44R35 SLE66Rxx	Calypso CEPAS Moneo SRI512, SRT512, SRI4K, SRIX4K Inside Contactless PicoPass[1] HID ICLASS[1]	FeliCa[1] NFC

[1] UID only

4. Setting up a Terminal Program

In order to establish a connection between the reader and a terminal program, the following steps have to be done:

- Connect your reader to a PC
- Start your preferred terminal program (for Windows e.g. HyperTerminal)
- Select the serial port (e.g. COM1), where you connected the reader.
- Set up the connection speed and format: 9600 baud (default), 8 data bits, no parity, 1 stop bit
- Select no software handshake and no hardware handshake

5. Register Set

The reader has several system flags customizing its behavior. The flags are stored non-volatile in its EEPROM. It is recommended to keep all bits and bytes labeled RFU at their default state to guarantee further compatibility.

Please consider that if any register is written, the reader needs a reset so that the changes may take effect.

5.1 EEPROM Memory Organization

Register	Access	Default value	Description
00h … 09h	RO	XXh	Internal use
0Ah	R/W	01h	Station ID
0Bh	R/W	41h	Protocol Configuration Register 1
0Ch	R/W	00h	Baudrate Control Register
0Dh	R/W	20h	RFU, never change this register
0Eh	R/W	7Fh	Operation Mode Register 1
0Fh	R/W	0Ah	RFU, never change this register
10h	R/W	00h	Internal use
11h	R/W	00h	Internal use
12h	R/W	00h	Internal use
13h	R/W	00h	Protocol Configuration Register 2
14h	R/W	0Ah	Reset Off Time
15h	R/W	25h	Reset Recovery Time
16h	R/W	00h	RFU, never change this register
17h	R/W	10h	Internal use
18h	R/W	50h	Internal use
19h	R/W	10h	Internal use
1Ah	R/W	20h	Internal use
1Bh	R/W	00h	Protocol Configuration Register 3
1Ch … 1Fh	R/W	00h	Internal use
20h … DFh	R/W	FFh	RFU, never change this register
E0h … EFh	R/W	00h	Installation Identifier
F0h	R/W	03h	Operation Mode Register 2
F1h	R/W	22h	RFU, never change this register
F2h	R/W	B4h	Startup Delay Register

5.2 Station ID (0Ah)

In ASCII mode, the Station ID has principally no influence on the functional behavior of the reader. Nevertheless it can be used to identify a reader in a multi-reader environment.

In Binary mode, the Station ID is used to address the reader in protocol header. The Station ID has the range 01h to FEh and can be set freely. The value 00h is reserved for the bus-master, all readers send their response to this device. The broadcast message (FFh) forces all readers to response to the command.

Default value: 01h

5.3 Protocol Configuration Register 1 (0Bh)

The Protocol configuration register 1 specifies the general behavior of the reader.

Protocol Configuration Register 1							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ExtendID	RFU	RFU	RFU	NewSerial Mode	MultiTag	Binary	AutoStart

5.3.1 AutoStart

If set the reader will start up in continuous read mode.

Default value: 1

5.3.2 Binary

If set the reader uses binary protocol. Refer to binary protocol description for further information.

Default value: 0

5.3.3 MultiTag

The MultiTag flag will enable multi tag recognition in continuous read mode. All tags are detected and displayed. Due to the more complex search algorithm detection speed is decreased in continuous read mode.

Default value: 0

5.3.4 NewSerialMode

The NewSerialMode flag controls the addition of a leading character to the serial number of transponders. This eases the recognition of transponder types.

This setting affects the commands continuous reading (‘c’), single tag select (‘s’) and multi tag select (‘m’).

Default value: 0

Character	Transponder type
M	ISO14443A, e.g. MIFARE®
P	PicoPass
S	ISO14443B, e.g. SRIX4K

5.3.5 ExtendID

If set, the unique serial number (UID) of the transponder is extended by a single prefix byte. This bit has only effect for ISO14443A transponders.

This setting affects the commands continuous reading (‘c’), single tag select (‘s’) and multi tag select (‘m’).

Default value: 0

Possible values for the prefix byte are:

Prefix	Description
01h	Cascade Level 1 transponder, e.g. MIFARE® Classic
02h	Cascade Level 2 transponder, e.g. MIFARE® Ultralight
03h	Cascade Level 3 transponder

5.4 Baudrate Control Register (0Ch)

The Baudrate control register defines the start-up baudrate of the reader. For baudrate values, refer to the tables below:

Baudrate Control Register							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RFU	RFU	RFU	RFU	RFU	BS2	BS1	BS0

BS2	BS1	BS0	Resulting Baudrate
0	0	0	9600 bps (default)
0	0	1	19200 bps
0	1	0	38400 bps
0	1	1	57600 bps
1	0	0	115200 bps
1	0	1	reserved
1	1	0	reserved
1	1	1	reserved

5.4.1 Resetting the baudrate to default

The reader has a built-in mechanism to reset the baudrate to the default value of 9600 bps. This is useful, if no communication is possible any more due to an errant setting of the Baudrate control register. The reset can be done by sending a byte sequence at 9600 bps to the reader. If the sequence matches, the reader turns its baudrate setting to 9600 bps. This configuration remains valid until a subsequent reset or power-down event occurs. The user may now write the desired baudrate setting to the Baudrate Control Register. After that, a reset must be executed, so that the new settings may take effect.

In order to reset the baudrate, the host has to send the value FFh at 9600 bps three times to the reader. The bytes must have a temporal distance of at least 100ms. If a valid sequence is recognized, the reader switches to 9600 bps and sends an acknowledge to the host (‘B<CR><LF>’) using the ASCII protocol.

5.5 Operation Mode Register 1 (0Eh)

The Operation Mode Register 1 defines which tag types the reader shall support. Including or excluding transponder types has direct influence in the transponder detection speed of the reader.

Operation Mode Register 1							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RFU	RFU	RFU	RFU	RFU	SRX	ISO14443B	ISO14443A

5.6 Protocol Configuration Register 2 (13h)

The Protocol Configuration Register 2 specifies the further general behavior of the reader.

Protocol Configuration Register 2							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RFU	RFU	RFU	RFU	Noisy Environment	RFU	Disable Startup Message	RFU

5.6.1 Disable Startup Message

If set, the reader will not print the startup message (**`MifareNFC 1.07`**) when powered or if a reset occurs.

Default value: 0

5.6.2 Noisy Environment

If set, the reader will only quit continuous read mode if the '**`.`**' character is received.

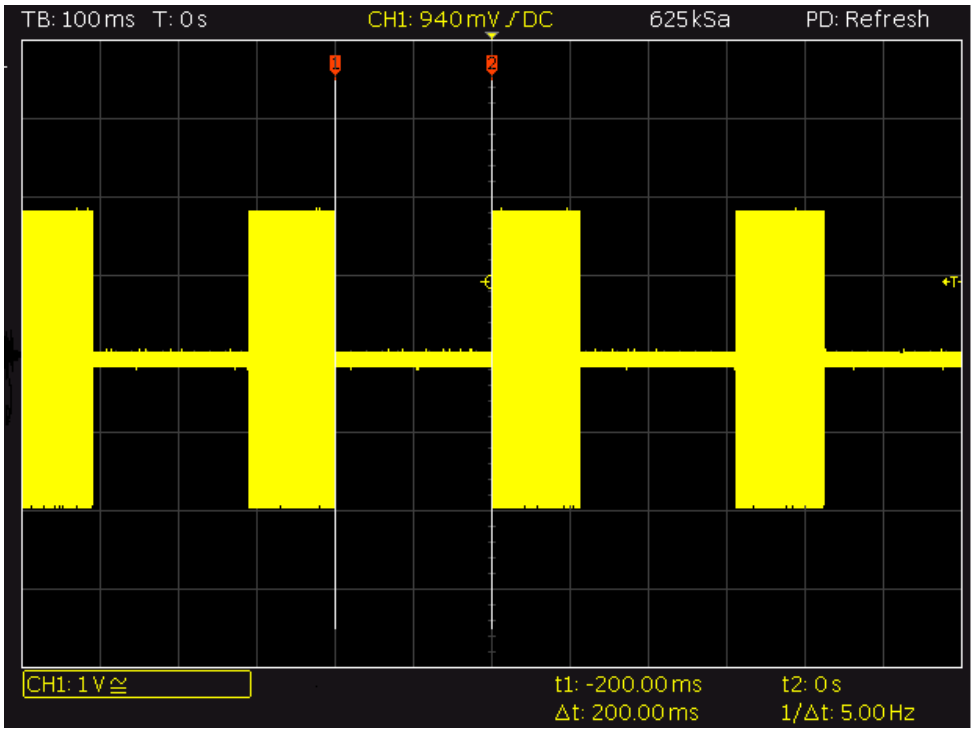
Default value: 0

5.7 Reset Off Time (14h)

The Reset Off Time register represents the time in milliseconds, the RF-field is switched off before and after a reading attempt. This register is used for the continuous read mode (‘c’).
The higher the value of the register, the more energy can be saved. Keep in mind that increased saving of energy results in decreased detection speeds.

Default value: 0Ah

The image below shows the activity of the RF-field. The Reset Off Time register has been set to 64h, this results in a pause of 200ms:

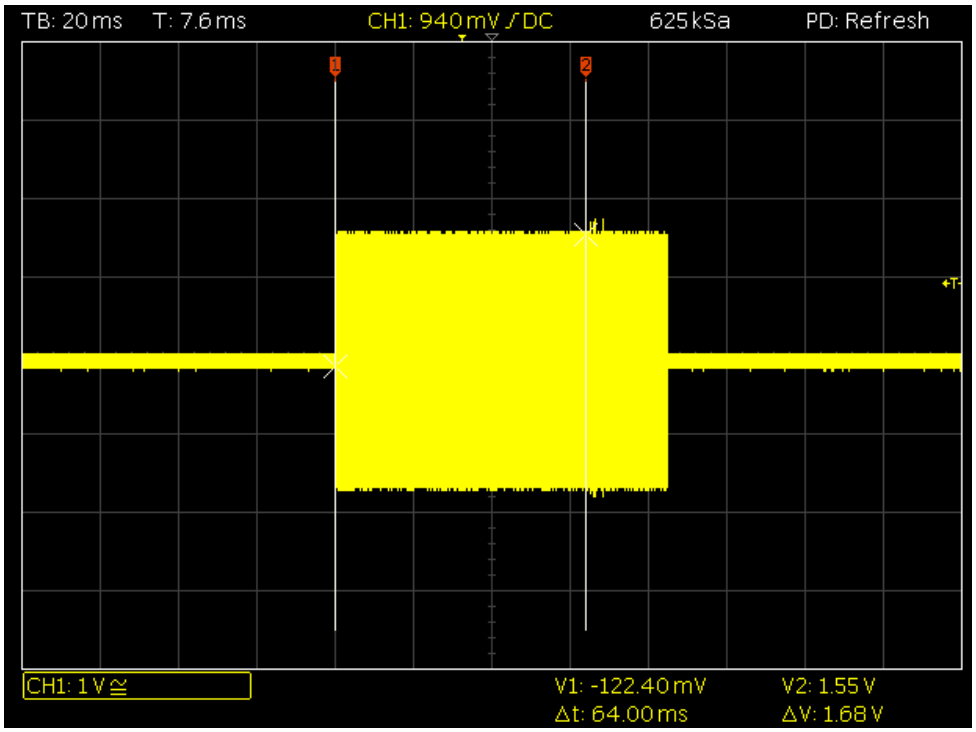


5.8 Reset Recovery Time (15h)

The Reset Recovery Time register represents the recovery time in milliseconds after the RF-field is turned on. This register is used for the continuous read mode (‘c’), single tag select (‘s’) and multi tag select (‘m’) commands.
The value of the register determines the time the reader waits before any reading attempt. Keep in mind that a higher value results in increased energy consumption.

Default value: 25h

The image below shows the activity of the RF-field. The Reset Recovery Time register has been set to 40h (64ms):



5.9 Protocol Configuration Register 3 (1Bh)

The Protocol Configuration register 3 specifies the further general behavior of the reader.

Protocol Configuration Register 3							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TagInfo	ReqA Extended ID	Internal use / do not change			RFU	RFU	RFU

5.9.1 ReqA Extended ID

If set, the reader will replace the cascade level information (1 byte) by the ReqA response (2 bytes). This bit has only effect in combination with ISO14443A transponders.

Default value: 0

5.9.2 TagInfo

If set, the reader extends the UID by a single byte. The value of the byte gives information about the tag type.

Default value: 0

Refer to the table below for tag types:

Byte value	Corresponding transponder type
00	MIFARE® Mini
01	MIFARE® Classic 1K
02	MIFARE® Classic 4K
03	MIFARE® Plus S
04	MIFARE® Plus X
05	MIFARE® Ultralight / Ultralight C
06	MIFARE® DESFire
0F	Unknown ISO14443A transponder
30	PicoPass / ICLASS transponder
40	SRX transponder
60	FeliCa transponder
80	ISO14443B transponder
FF	Unknown transponder

5.10 Installation Identifier (E0h … EFh)

The Installation Identifier is used for MIFARE® PLUS in virtual card environment. The Installation Identifier must match with the one stored in the MIFARE® PLUS transponder that shall be identified using the Virtual Card Select command.

5.11 Operation Mode Register 2 (F0h)

The Operation Mode Register 2 defines which tag types the reader shall support. Including or excluding transponder types has direct influence in the transponder detection speed of the reader.

Operation Mode Register 2							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RFU	RFU	RFU	RFU	RFU	RFU	FeliCa	PicoPass / ICLASS

5.12 Startup Delay Register (F2h)

The Startup Delay Register influences the boot time of the reader. The minimum required boot time after power up or a hardware reset is approximately 5ms. By storing a non-zero value in the Startup Delay Register, the user can influence e.g. the moment of sending the version string at startup. The resulting boot time is 5ms plus the value stored in the register in milliseconds.

Default value: B4h

6. Communication Protocol

6.1 ASCII Protocol

The ASCII protocol has been designed for easy handling. Data is always transmitted in hexadecimal notation, i.e. **5E**.

Every time the reader is powered up, a startup message is displayed. On the terminal screen this should look like this:

MifareNFC 1.07<CR><LF>

The reader is now ready for reception of commands. By default, the reader starts in continuous reading mode, this means the reader is scanning for transponders and prints the present UIDs.

Please note that pseudo-tetrade values always must be submitted in capital letters, e.g. **1234ABCD**

Commands must be submitted in lower-case letters, e.g. **wp0C03**

6.2 Binary Protocol

The binary protocol has been designed for industrial applications with synchronization and frame checking. The reader uses a watchdog timer internally to ensure correct framing. A binary frame is built up as follows:

STX	Station ID	Length	Data	BCC	ETX
1 byte	1 byte	1 byte	Various length	1 byte	1 byte

In binary mode, the reader only gives a response if a command is issued. This means, the reader does not show the start-up string and also continuous read does not work.

6.2.1 STX

Start of transmission (02h)

6.2.2 Station ID

00h: Reserved for the bus master. Responses are sent with Station ID set to 00h.
FFh: Broadcast message. All devices will execute the command and send their response.

6.2.3 Length

Denotes the length of the data block in bytes.

6.2.4 Data

The data block contains the command including its arguments. The command values are the same as in ASCII protocol mode whereas the arguments are transmitted in binary. If a command requires a subsequent carriage return (0Dh) it also must be submitted.

6.2.5 Block Check Character (BCC)

The BCC is used to detect transmission errors. The BCC is calculated by XOR-ing each byte of the transmission frame except the STX/ETX characters.

6.2.6 ETX

End of Transmission (03h)

6.2.7 Example

This example shows how to log into sector 0 of a MIFARE® card, using transport key FFh:

STX	Station ID	Length	Data	BCC	ETX
02	01	04	6C 00 FF 0D	9B	03

6.2.8 Remarks

If an invalid instruction frame is received (e.g. BCC is wrong) or if the requested Station ID does not match the internal ID of the reader, the command is not executed.

7. Instruction Set

7.1 Command Overview

7.1.1 Common Commands

Command	Description	For details view chapter
`' '`	Cancel command	7.6.6
`'c'`	Continuous read mode	7.3.1
`'.'`	Stop continuous read mode (in noisy environment)	7.6.6
`'ei'`	Init SAM card	7.5.9.1
`'et'`	SAM card transport data	7.5.9.2
`'g'`	Get station ID	7.6.1
`'ir'/'iw'`	Read / write GPIO0…3[1]	7.6.7 and 7.6.8
`'m'`	MultiTag select / tag list	7.3.3
`'of'`	Set configuration flags	7.5.3
`'og'`	Set configuration registers	7.5.4
`'o+a'/'o-a'`	Include / Exclude tag type ISO14443A	7.5.6 and 7.5.7
`'o+b'/'o-b'`	Include / Exclude tag type ISO14443B	7.5.6 and 7.5.7
`'o+j'/'o-j'`	Include / Exclude tag type FeliCa	7.5.6 and 7.5.7
`'o+p'/'o-p'`	Include / Exclude tag type PicoPass / ICLASS	7.5.6 and 7.5.7
`'o+s'/'o-s'`	Include / Exclude tag type SRX	7.5.6 and 7.5.7
`'oa'`	Set tag type ISO14443A	7.5.5
`'ob'`	Set tag type ISO14443B	7.5.5
`'oj'`	Set tag type FeliCa	7.5.5
`'op'`	Set tag type PicoPass / ICLASS	7.5.5
`'os'`	Set tag type SRX	7.5.5
`'ot'`	Search for all supported tag types	7.5.5
`'poff'`	Antenna power off	7.6.2
`'pon'`	Antenna power on	7.6.3
`'pr'/'pw'`	Read / write GPIO1[1]	7.6.9 and 7.6.10
`'r'/'rb'`	Read data block	7.4.2
`'rp'`	Read EEPROM register	7.5.1
`'s'`	Select single tag	7.3.2
`'t'`	Transparent ISO14443-4 transponder access	7.4.6
`'v'`	Get version	7.6.4
`'w'/'wb'`	Write data block	7.4.3
`'wd'`	Write AES / Triple DES key	7.5.8.2
`'wp'`	Write EEPROM register	7.5.2
`'x'`	Reset	7.6.5

[1] No function on TWN3

7.1.2 MIFARE® Classic Specific Commands

Command	Description	For details view chapter
'+'	Increment value block	7.4.4.3
'-'	Decrement value block	7.4.4.4
'='	Copy value block	7.4.4.5
'l'	Login (authenticate tag)	7.4.1.1 and 7.4.1.2
'rv'	Read value block	7.4.4.2
'wm'	Write MIFARE® Classic key	7.5.8.1
'wv'	Write value block	7.4.4.1

7.1.3 MIFARE® PLUS Specific Commands

Command	Description	For details view chapter
'ncp'	Commit Personalization	7.4.5.2
'nl'	Perform MIFARE® PLUS authentication	7.4.1.3
'nvcs'	Virtual Card Select	7.3.4
'nw'	Write configuration data	7.4.5.3
'nwp'	Write Personalization Data	7.4.5.1

7.1.4 MIFARE® DESFire Specific Commands 'f'

Sub-command	Description	For details view chapter
00	Authenticate	7.4.6.1
08	Select Application	7.4.6.2
0D	Select File	7.4.6.3
15	Read File	7.4.6.4
16	Write File	7.4.6.5
20	Get card UID	7.4.6.6

7.2 Error Codes

Error Code	Description
'?<CR><LF>'	Unknown command
'E<CR><LF>'	Invalid key format
'F<CR><LF>'	General failure
'I<CR><LF>'	Invalid value block (block does not match the value format)
'N<CR><LF>'	No tag in the field / SAM card not found
'O<CR><LF>'	Operation mode failure
'R<CR><LF>'	Parameter out of range
'X<CR><LF>'	Block already locked / Error during value operation

7.3 Transponder Serial Number Related Commands

7.3.1 Continuous read mode ‘c’

The reader reads and displays serial numbers continuously while one or more tags remain in the field. This command stops as soon as any character is sent to the reader.

The reader supports different tag types at the same time. In order to increase the reading performance, switch to single tag mode. If more than one tag shall be detected at the same time, the MultiTag flag must be activated. The response data length mainly depends on the tag type.

Command: `'c'`

Answer

Answer	Description
`Data<CR><LF>`	Serial number (UID, n bytes)

7.3.2 Select Single Tag ‘s’

This command selects a single tag in the antenna field. It shall only be used in single tag environments (use the ‘m’ command in multiple tag environments). In case of success, the command returns the UID of the selected card. The length of the UID is detected automatically.

As soon as a transponder has been selected, it is ready for further data transactions, e.g. authentication, read block or write block.

Command: `'s'`

Answer

Answer	Description
`Data<CR><LF>`	Serial number (UID, n bytes)
`'N<CR><LF>'`	Error: No tag in the field

7.3.3 MultiTag Selection / Tag List ‘m’

This command detects several tags at the same time. It replaces the fast select command (‘s’) in multiple tag surroundings. The MultiTag list command lists all present tags with their serial numbers. Use the MultiTag select command to select a single tag. Each tag has to be selected separately.

After selection, the transponder is ready for further data transactions, e.g. authentication, read block or write block.

Keep in mind that each transponder consumes its individual amount of energy, provided by the reader. Due to the limited availability of emitted energy, the operating distance is decreased, the more transponders are present. Principally the operating distance depends on the used transponders, the total amount of transponders and the ambient conditions, e.g. if metal is surrounding the reader.

Command: `'m<CR>' / 'm[UID]<CR>'`

Command	Description
`'m<CR>'`	List all present tags
`'m[UID]<CR>'`	Select tag by its UID

Answer

Answer	Description
`Data<CR><LF>`	Serial number (UID, n bytes)
`'N<CR><LF>'`	Error: No tag in the field

Examples

Command	Answer	Description

`'m<CR>'`	`044A3A11FC1E80`
	`56AB3798`
	`02`
		First tag
		Second tag
		Number of detected tags

`'m56AB3798000000<CR>'`	`56AB3798`	Second tag selected

Note:

The MultiTag selection command always requires a 7 byte UID in case of a MIFARE® card. If the UID of the desired transponder possesses only 4 bytes (e.g. MIFARE® Classic), the passed UID must be filled up with zeros – see the example above.

7.3.4 MIFARE® PLUS Virtual Card Select ‘nvcs’

Use this command to retrieve the ‘real’ UID of the transponder in a virtual card environment. If a card is configured to show random IDs, the virtual card select command offers the possibility to retrieve the UID in a fast and secure way. In order to get the UID, the 16 bytes Installation Identifiers of the card and the reader must match. Furthermore, two AES keys are required. These are the Virtual Card Polling Encryption key and the Virtual Card Polling MAC key. Please note, that only EEPROM keys can be used.

Command: `'nvcs[VCPEKey][VCPMACKey]'`

Parameters	Description
`[VCPEKey]`	Virtual Card Polling Encryption EEPROM AES key (1 byte)
`[VCPMACKey]`	Virtual Card Polling MAC EEPROM AES key (1 byte)

Answer

Answer	Description
`UID<CR><LF>`	Real UID of the transponder
`'F<CR><LF>'`	Error: Proximity check failed
`'R<CR><LF>'`	Error: Parameter out of range

Example

Command	Description
`'nvcs5051<CR>'`	Perform virtual card select, using EEPROM AES keys 00 and 01

7.4 Data-Transaction Related Commands

7.4.1 Authenticate Tag

7.4.1.1 MIFARE® Classic Login 'l'

This command performs an authentication into a specific sector of a MIFARE® Classic transponder. Only one transponder and only one sector can be accessed at the same time. Prior access, the transponder must be selected by either single tag or MultiTag selection commands.

For authentication into a sector, the matching login key is needed. The key may either be entered by command, or can be stored in the readers' EEPROM. The reader is able to store up to 32 MIFARE® Classic keys. Principally every stored key may act either as key A or key B, the selection is done via parameter list.

Command: `'l[Sector][KeyType][Key / <CR>]'`

Parameters	Description
`[Sector]`	Sector number, valid range 00h – 3Fh
`[KeyType]`	AAh: authenticate with key type A FFh: authenticate with key type A, transport key FFFFFFFFFFFFh BBh: authenticate with key type B 10h … 2Fh: authenticate with stored key type A (00h … 1Fh) 30h … 4Fh: authenticate with stored key type B (00h … 1Fh)
`[Key / <CR>]`	Enter key manually (6 bytes) or tell the reader to login with a transport key by submitting a carriage return `<CR>` (1 byte)

Answer

Answer	Description
`'L<CR><LF>'`	Login success
`'F<CR><LF>'`	Error: general failure
`'N<CR><LF>'`	Error: no tag in the field or wrong key

Examples

Command	Description
`'101AA<CR>'`	Authenticate into sector 01 using transport key type A A0A1A2A3A4A5h
`'102BB<CR>'`	Authenticate into sector 02 using transport key type B B0B1B2B3B4B5h
`'103FF<CR>'`	Authenticate into sector 03 using transport key type A FFFFFFFFFFFFh
`'104AA1234567890AB'`	Authenticate into sector 04 using specified key type A 1234567890ABh
`'10510'`	Authenticate into sector 05 using EEPROM key 0, key type A
`'10637'`	Authenticate into sector 06 using EEPROM key 7, key type B

7.4.1.2 MIFARE® Ultralight C Login 'l'

This command performs an authentication into a MIFARE® Ultralight C transponder. The authentication scheme into this transponder type is different to MIFARE® Classic due to the fact, that Ultralight C uses Triple-DES cryptography.

Unlike MIFARE® Classic the authentication concerns the entire transponder. This means, once an authentication succeeds, the complete transponder is accessible.

Prior login, the transponder must be selected by either single tag or MultiTag selection commands.

For authentication, the matching login key is needed. The key may either be entered by command, or can be stored in the readers' EEPROM. The reader is able to store up to 16 Triple-DES keys.

Command: `'l[00][KeyType][Key / <CR>]'`

Parameters	Description
`[00]`	To maintain backward compatibility the first parameter shall be set to 00h
`[KeyType]`	CCh: denotes Triple-DES (Ultralight C) key 50h … 5Fh: authenticate with stored Triple-DES key (00h … 0Fh)
`[Key / <CR>]`	Enter key manually (16 bytes) or tell the reader to login with a transport key by submitting a carriage return `<CR>` (1 byte)

Answer

Answer	Description
`'L<CR><LF>'`	Login success
`'F<CR><LF>'`	Error: general failure
`'N<CR><LF>'`	Error: no tag in the field or the tag does not respond

Examples

Command	Description
`'l00CC<CR>'`	Authenticate into transponder using Triple-DES transport key: K1 = 49454D4B41455242h K2 = 214E4143554F5946h
`'l00CC00010203040506070809 0A0B0C0D0E0F'`	Authenticate into transponder using Triple-DES key: K1 = 0001020304050607h K2 = 08090A0B0C0D0E0Fh
`'l0055'`	Authenticate into transponder using Triple-DES EEPROM key 05h

7.4.1.3 MIFARE® PLUS Login 'nl'

MIFARE® PLUS offers various authentication features to the user. Depending on the desired transaction or configured security level of the transponder, the appropriate authentication procedure must be chosen by the user. The reader always demands an AES key for authentication; if the transponder is configured to security level 2, optionally a Crypto1 key can be passed to the reader. This enables the reader to compute the appropriate session key for further communication ciphered by the MIFARE® Crypto1 algorithm. The Crypto1 login is performed automatically after AES authentication.

The login procedure is also used to switch a transponder to a higher security level. Furthermore, an optional AES-authentication can be done in SL1 using the SL1 card authentication key.

Command: **`'nl[Sector / KeyNumber][KeyType][KeyAES][KeyCrypto1]'`**

Parameters	Description
`[Sector / KeyNumber]`	Sector- or Keynumber, valid range 00h – 2Ch. 00h … 27h: Parameter addresses a data sector 28h: Card master key 29h: Card configuration key 2Ah: Level 2 switch key 2Bh: Level 3 switch key 2Ch: SL1 card authentication key
`[KeyType]`	AAh: authenticate with key type A BBh: authenticate with key type B
`[KeyAES]`	Enter key manually (16 bytes) or tell the reader to login with a stored EEPROM key (1 byte). EEPROM keys are range 50h … 5Fh
`[KeyCrypto1]`	Enter key manually (6 bytes), or tell the reader to generate the Crypto1 session key using either a stored EEPROM key or a transport key (1 byte). EEPROM keys are range 10h … 2Fh, Transport keys are indicated by AA/BB/FF

Answer

Answer	Description
`'L<CR><LF>'`	Login / Authentication / Security level switch success
`'F<CR><LF>'`	Error: general failure
`'N<CR><LF>'`	Error: no tag in the field or the tag does not respond

Examples

Command	Description
'nl00AA50<CR>'	Authenticate to sector 00 using EEPROM AES key 00, type A
'nl02BB51<CR>'	Authenticate to sector 02 using EEPROM AES key 01, type B
'nl28AA52<CR>'	Authenticate with card master key using EEPROM AES key 02, KeyType is ignored but has to be transmitted to the reader.
'nl2BAA53<CR>'	Switch card to SL3 using EEPROM AES key 03, KeyType is ignored but has to be transmitted to the reader.
'nl03AA0011223344556677 8899AABBCCDDEEFF<CR>'	Authenticate into sector 03 using specified AES key 00112233445566778899AABBCCDDEEFFh, key type A
'nl04AA0011223344556677 8899AABBCCDDEEFFAA<CR>'	Authenticate into sector 04 using specified AES key 00112233445566778899AABBCCDDEEFFh, key type A, compute Crypto1 session key, key base is transport key A0A1A2A3A4A5h
'nl05BB0011223344556677 8899AABBCCDDEEFFBB<CR>'	Authenticate into sector 05 using specified AES key 00112233445566778899AABBCCDDEEFFh, key type B, compute Crypto1 session key, key base is transport key B0B1B2B3B4B5h
'nl06AA0011223344556677 8899AABBCCDDEEFF16<CR>'	Authenticate into sector 06 using specified AES key 00112233445566778899AABBCCDDEEFFh, key type A, compute Crypto1 session key, key base is EEPROM Crypto1 key 06
'nl07AA54A0A1A2A3A4A5<CR>'	Authenticate into sector 07 using EEPROM AES key 04, compute Crypto1 session key, key base is specified key A0A1A2A3A4A5h
'nl08AA55FF<CR>'	Authenticate into sector 08 using EEPROM AES key 05, compute Crypto1 session key, key base is transport key FFFFFFFFFFFFh
'nl09AA5617<CR>'	Authenticate into sector 09 using EEPROM AES key 06, compute Crypto1 session key, key base is EEPROM Crypto1 key 07
'nl0AAA0011223344556677 8899AABBCCDDEEFFA0A1A2A3A4A5'	Authenticate into sector 0A using specified AES key 00112233445566778899AABBCCDDEEFFh, compute Crypto1 session key, key base is specified key A0A1A2A3A4A5h. Note that this command is not followed by a <CR>

7.4.1.4 Login in multiple Tag Surroundings

In order to log into different tags, list all present tags first and then select the desired tag. After that, perform the login procedure.

7.4.2 Read Data Block ‘r’ / ‘rb’

This command reads an entire data block from a transponder. The range of valid block addresses depends on the used transponder. If you are working with MIFARE® cards, keep in mind that any read- or write access requires a successful login into the respective sector. The length of returned data depends on the tag-type, e.g. a MIFARE® Classic card returns 16 bytes of data, a SRX transponder returns 4 bytes per block.

Command: **`'r[BlockAddr]' / 'rb[BlockAddr]'`**

Answer

Answer	Description
`Data<CR><LF>`	Block data
`'F<CR><LF>'`	Error: read failure
`'N<CR><LF>'`	Error: no tag in the field, or the tag does not respond
`'R<CR><LF>'`	Error: Block address out range; the block address of the ‘r’ command is higher than 40h, use the ‘rb’ command instead.

Example

Command	Description
`r08`	Reads block 08 (sector 02, block 00)

7.4.2.1 MIFARE® Ultralight / Ultralight C

Though the page size of this transponder family is 4 bytes, the read command always returns 16 bytes. This is achieved by reading the three subsequent data pages, e.g. if page 04 is to be read, the reader also returns the content of page 05, 06 and 07.

7.4.3 Write Data Block ‘w’ / ‘wb’

This command writes an entire data block to the transponder. The range of valid block addresses depends on the used transponder. A read after write is done automatically to ensure data integrity. If you are working with MIFARE® cards, keep in mind that any read- or write access requires a successful login into the respective sector. The length of required data to be written to the transponder depends on the tag-type, e.g. a MIFARE® Standard block has 16 bytes, a SRX transponder has 4 bytes per block.

Command: **`'w[BlockAddr]' / 'wb[BlockAddr]'`**

Answer

Answer	Description
`Data<CR><LF>`	Block data
`'F<CR><LF>'`	Error: write failure
`'N<CR><LF>'`	Error: no tag in the field, or the tag does not respond
`'R<CR><LF>'`	Error: Operation mode failure; the block address of the ‘w’ command is higher than 40h, use the ‘wb’ command instead.

Example

Command	Description
`w0800010203040506070809 0A0B0C0D0E0F`	Writes data **`00010203040506070809 0A0B0C0D0E0F`** to block 08 (sector 02, block 00)
`w1A11223344`	Writes data **`11223344`** to block 1Ah

7.4.3.1 MIFARE® Ultralight / Ultralight C

If data is written to a page of this transponder family, the reader performs the so-called “Compatibility-write” procedure. This means, the reader expects 16 bytes of data, but only the first 4 bytes are written to the specified data page; the remaining 12 bytes are ignored by the transponder.

The command executes an automatic read-after-write check, where all 16 bytes are included into the comparison. So the risk of getting a write failure is very high, if the content of subsequent memory-pages is unknown and the remaining 12 bytes are filled with dummy bytes! So it could be useful, to make a manual read-before-write, to construct the parameter list correctly in order not to get such errors.

Example

Command	Description
`r04`	Reads pages 04 … 07, content: **`00000000040506070809 0A0B0C0D0E0F`**
`w0400010203040506070809 0A0B0C0D0E0F`	Writes data **`00010203`** to page 04

7.4.4 MIFARE® Value Block Related commands

A sector of a MIFARE® transponder may contain so-called ‘value blocks’. A value block is a usual data block, where the information is stored in a certain format. Special MIFARE® commands like increment or decrement may be applied to such a block, e.g. for electronic purse functionality. Value block commands only work in combination with MIFARE® Classic and MIFARE® PLUS transponders. Please note, that only MIFARE® PLUS X transponders support value block commands in SL3.

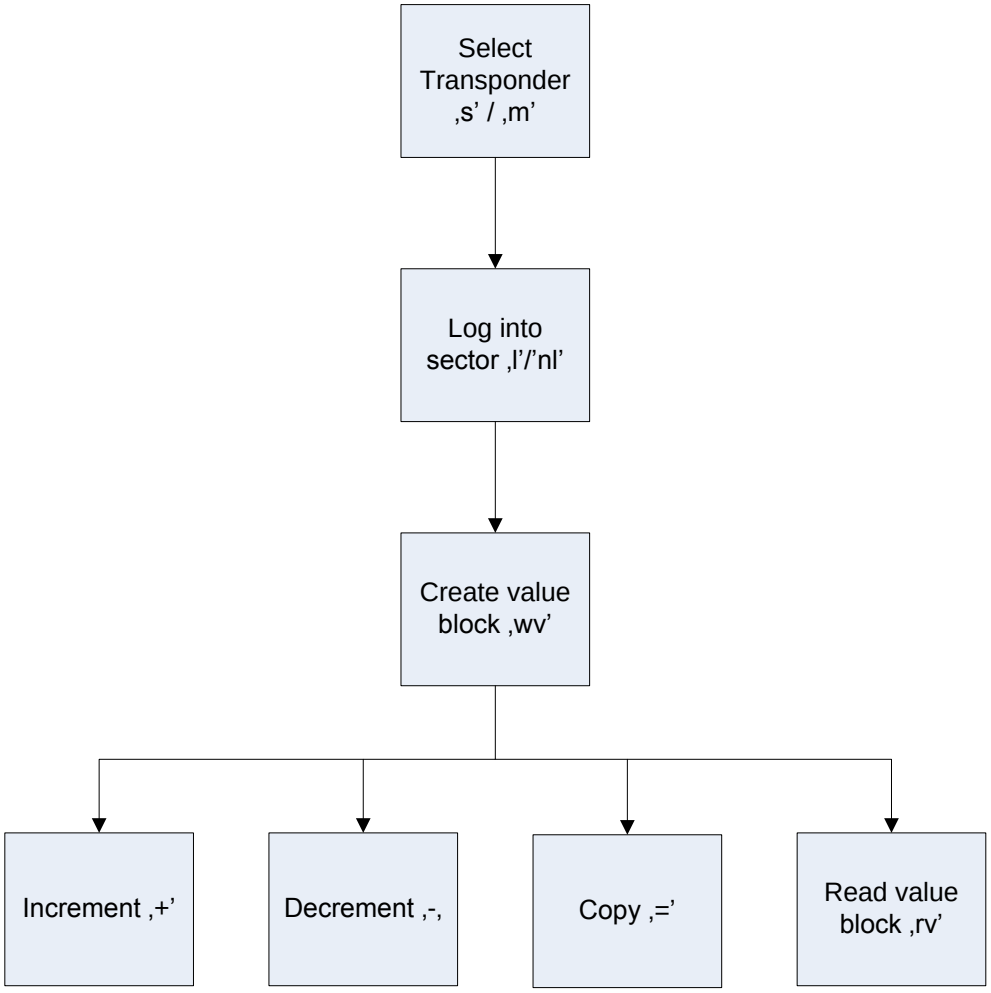
A value block contains 4 bytes user data, the remaining 12 bytes are processed internally by the MIFARE® transponder for increased data integrity.

A value block is formatted as follows:

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Data	Value				$\overline{\text{Value}}$				Value				A	$\overline{\text{A}}$	A	$\overline{\text{A}}$

The value data is stored three times: twice non-inverted and once inverted. The lowest significant byte is stored in the lowest address byte. The last four bytes are for internal use and shall not be altered.

The following diagram shows the typical command-flow in order to work with MIFARE® value blocks:



7.4.4.1 Create Value Block ‘wv’

Use this command to format a common data block as a value block. The range of valid block addresses depends on the used transponder type. You must log into the respective sector before this command can be executed.

Command: `'wv[BlockAddr][Value]'`

Parameters	Description
`[BlockAddr]`	Block address
`[Value]`	Initial value stored on the value block (4 bytes). The value is stored signed (most significant bit). Negative values are stored in 2’s complement format.

Answer

Answer	Description
`Data<CR><LF>`	Written value
`'F<CR><LF>'`	Error: write failure
`'I<CR><LF>'`	Error: invalid block format

Examples

Command	Description
`wv0800000000`	Formats block 08 as a value block. Initial value: 00000000h
`wv0912345678`	Formats block 09 as a value block. Initial value: 12345678h
`wv0AFFFFFFFE`	Formats block 0A as a value block. Initial value: FFFFFFFEh (-2)

7.4.4.2 Read Value Block ‘rv’

Use this command to read a value block. This command checks if data of the specified block is stored in value format. You must log into the respective sector before this command can be executed.

Command: `'rv[BlockAddr]'`

Parameters	Description
`[BlockAddr]`	Block address

Answer

Answer	Description
`Data<CR><LF>`	Value of block
`'F<CR><LF>'`	Error: read failure
`'I<CR><LF>'`	Error: invalid block format

Example

Command	Description
`rv08`	Reads value block 08

7.4.4.3 Increment Value Block '+'

Use this command to increment a value block by a defined value. A read after increment is performed automatically. The command fails if the specified block is not formatted as value block. You must log into the respective sector before this command can be executed.

Command: `'+[BlockAddr][Value]'`

Parameters	Description
`[BlockAddr]`	Block address (1 byte)
`[Value]`	Value to be added (4 bytes)

Answer

Answer	Description
`Data<CR><LF>`	Value of block
`'F<CR><LF>'`	Error: read failure
`'I<CR><LF>'`	Error: invalid block format
`'X<CR><LF>'`	Error: Bad value operation, e.g. sign overflow

Examples

Command	Description
`+0800000001`	Increments block 08 by 00000001h
`+0812345678`	Increments block 08 by 12345678h

7.4.4.4 Decrement Value Block '-'

Use this command to decrement a value block by a defined value. A read after decrement is performed automatically. The command fails if the specified block is not formatted as value block. You must log into the respective sector before this command can be executed.

Command: `'-[BlockAddr][Value]'`

Parameters	Description
`[BlockAddr]`	Block address (1 byte)
`[Value]`	Value to be subtracted (4 bytes)

Answer

Answer	Description
`Data<CR><LF>`	Value of block
`'F<CR><LF>'`	Error: read failure
`'I<CR><LF>'`	Error: invalid block format
`'X<CR><LF>'`	Error: Bad value operation, e.g. sign overflow

Examples

Command	Description
`-0800000001`	Decrements block 08 by 00000001h
`-0812345678`	Decrements block 08 by 12345678h

7.4.4.5 Copy Value Block '='

Use this command to copy a value block to another value block of the same sector. A read after copy is performed automatically. The command fails if one of the specified blocks is not in value format. You must log into the respective sector before this command can be executed.

Command: `'=[SourceBlock][TargetBlock]'`

Parameters	Description
`[SourceBlock]`	Source block address (1 byte)
`[TargetBlock]`	Target block address (1 byte)

Answer

Answer	Description
`Data<CR><LF>`	New value of target block
`'F<CR><LF>'`	Error: general failure
`'I<CR><LF>'`	Error: invalid block format

Examples

Command	Description
`=0809`	Copy value block 08 to block 09
`=080A`	Copy value block 08 to block 0A

7.4.5 MIFARE® PLUS Related Commands

7.4.5.1 Write Personalization Data 'nwp'

Use this command to transmit personalization data to the transponder in SL0. At least Card Master key, Card configuration key, Level 2 switch key (MIFARE® PLUS X only) and Level 3 switch key must be written in order to personalize the transponder. The personalization is finished by sending the command 'Commit Personalization'. Personalization shall happen at a secure place because all data is transmitted in plain.

Command: **`'nwp[DataBlock / SpecialBlock][KeyType][Data]'`**

Parameters	Description
`[DataBlock / SpecialBlock]`	If KeyType is present and set to AA or BB, the first parameter is interpreted as SpecialBlock. This makes it possible to address e.g. AES keys and Configuration blocks. Every sector can hold two AES keys (type A and type B). Otherwise the standard data blocks are accessed (00h … FFh) 00h … 27h: Parameter addresses an AES key of data sector 28h: Card master key 29h: Card configuration key 2Ah: Level 2 switch key 2Bh: Level 3 switch key 2Ch: SL1 card authentication key 2Dh: Select Virtual Card key 2Eh: Proximity Check key 2Fh: Virtual Card Polling Encryption key 30h: Virtual Card Polling MAC key 40h: MFP Configuration Block 41h: Installation Identifier 42h: ATS Information 43h: Field Configuration Block
`[KeyType]`	AAh: indicate key type A BBh: indicate key type B
`[Data]`	Data to be written to the transponder (16 bytes)

Answer

Answer	Description
`Data<CR><LF>`	Written data
`'F<CR><LF>'`	Error: General failure
`'R<CR><LF>'`	Error: Parameter out of range

Examples

Command	Description
`nwp010011223344556677 8899AABBCCDDEEFF<CR>`	Writes 00112233445566778899AABBCCDDEEFFh to data block 01
`nwp00AA0011223344556677 8899AABBCCDDEEFF`	Set AES key of sector 00, key type A, key is: 00112233445566778899AABBCCDDEEFFh
`nwp01BB0011223344556677 8899AABBCCDDEEFF`	Set AES key of sector 01, key type B, key is: 00112233445566778899AABBCCDDEEFFh
`nwp28AA0001020304050607 08090A0B0C0D0E0F`	Set Card master key to 000102030405060708090A0B0C0D0E0Fh

7.4.5.2 Commit Personalization 'ncp'

Use this command to finish personalization and to switch the transponder from SL0 to SL1. If the personalized transponder is a so-called 'L3'-card, the transponder is switched to SL3. This command is only supported if the transponder is in SL0.

Command: `'ncp'`

Answer

Answer	Description
`'L<CR><LF>'`	Personalization successfully completed
`'F<CR><LF>'`	Error: General failure

7.4.5.3 Write AES Sector Key / Special Block 'nw'

Use this command to write an AES sector key or a special block. If an AES sector key shall be written, you have to be authenticated to the respective sector and the respective key type. If the transponder has been switched to SL2, an authentication with the Card master key is mandatory prior authentication to the respective sector. If a Special block shall be written, you must be authenticated either with the Card master key or the Card configuration key:

Key	Change
Card master key	- Level switch keys - Card configuration key - MFP Configuration block - Installation Identifier - ATS - Card master key
Card configuration key	- Field configuration block - Virtual card keys - Proximity check key - Card configuration key

Command: `'nw[AESSectorKey / SpecialBlock][KeyType][Data]'`

Parameters	Description
`[AESSectorKey / SpecialBlock]`	If KeyType is present and set to AA or BB, the first parameter is interpreted as AESKey. This makes it possible to address the AES keys and their respective key type. Every sector can hold two AES keys (type A and type B). 00h … 27h: Parameter addresses an AES key of a data sector 28h: Card master key 29h: Card configuration key 2Ah: Level 2 switch key 2Bh: Level 3 switch key 2Ch: SL1 card authentication key 2Dh: Select Virtual Card key 2Eh: Proximity Check key 2Fh: Virtual Card Polling Encryption key 30h: Virtual Card Polling MAC key 40h: MFP Configuration Block 41h: Installation Identifier 42h: ATS Information 43h: Field Configuration Block
`[KeyType]`	AAh: indicate key type A BBh: indicate key type B
`[Data]`	Data to be written to the transponder (16 bytes)

Answer

Answer	Description
Data<CR><LF>	Written data
'F<CR><LF>'	Error: General failure
'R<CR><LF>'	Error: Parameter out of range

Examples

Command	Description
nw00AA001122334455 66778899AABBCCDDEEFF	Change AES sector key, sector 00, key type A to 00112233445566778899AABBCCDDEEFFh
nw01BB000000000000 00000000000000000000	Change AES sector key, sector 01, key type B to 00000000000000000000000000000000h
nw2800102030405060 708090A0B0C0D0E0F0<CR>	Change Card master key to 000102030405060708090A0B0C0D0E0Fh

7.4.6 MIFARE® DESFire related commands

The following commands can be used to perform basic data transactions on DESFire transponders. A typical command-flow would be:

Select Tag → Select Application → Authenticate → Select File → Read/Write data

7.4.6.1 Authenticate

For DESFire transponders, only AES is supported for authentication.

Command: **'f[Length][00][KeyNoTag][KeyNoEEPROM / Key]<CR>'**

Parameters	Description
[Length]	Depends on used key. Key resides in EEPROM: Length = 03h Key is passed in command: Length = 12h
[KeyNoTag]	Reference to a key on tag
[KeyNoEEPROM]	Login with a stored EEPROM key. EEPROM keys are range 00h … 0Fh
[Key]	Login with this key

Answer

Answer	Description
'L<CR><LF>'	Login success
'F<CR><LF>'	Error: general failure
'N<CR><LF>'	Error: no tag in the field or the tag does not respond

Examples

Command	Description
'f03000102<CR>'	Authenticate with EEPROM key 02, key number on tag is 01
'f12000111223344556677889 9AABBCCDDEEFF<CR>'	Authenticate with passed key, key number on tag is 01

7.4.6.2 Select Application

Command: **'f[Length][08][AID]<CR>'**

Parameters	Description
[Length]	Set this value to 04h
[AID]	Application ID (3 bytes)

Answer

Answer	Description
'00<CR><LF>'	AID successfully selected
'N<CR><LF>'	Error: no tag in the field or the tag does not respond

Example

Command	Description
'f0408010203<CR>'	Select Application with AID 010203

7.4.6.3 Select File

Command: `'f[Length][0D][FileID]<CR>'`

Parameters	Description
[Length]	Set this value to 02h
[FileID]	File ID (1 byte)

Answer

Answer	Description
'00<CR><LF>'	File successfully selected
'F<CR><LF>'	Error: general failure

Example

Command	Description
'f020D12<CR>'	Select File with ID 12

7.4.6.4 Read File

Using this command, you can read a maximum of 32 bytes by a single transaction from a data file.

Command: `'f[Length][15][Offset][BytesToRead]<CR>'`

Parameters	Description
[Length]	Set this value to 05h
[Offset]	Offset within selected file (3 bytes)
[BytesToRead]	Number of bytes to be read (1 byte)

Answer

Answer	Description
'Data<CR><LF>'	File successfully selected
'N<CR><LF>'	Error: no tag in the field or the tag does not respond
'C<CR><LF>'	Error: checksum or data integrity invalid
'R<CR><LF>'	Error: attempt to read out of range
'F<CR><LF>'	Error: general failure

Examples

Command	Description
'f051500000010<CR>'	Read 16 bytes from file starting at offset 000000h
'f051500000A08<CR>'	Read 8 bytes from file starting at offset 00000Ah

7.4.6.5 Write File

Using this command, you can write a maximum of 32 bytes by a single transaction to a data file.

Command: `'f[Length][16][Offset][Data]<CR>'`

Parameters	Description
[Length]	Set this value to 04h + Length of Data
[Offset]	Offset within selected file (3 bytes)
[Data]	Data to be written to the file (variable length)

Answer

Answer	Description
'00<CR><LF>'	Data successfully written
'N<CR><LF>'	Error: no tag in the field or the tag does not respond
'C<CR><LF>'	Error: checksum or data integrity invalid
'R<CR><LF>'	Error: attempt to write out of range
'O<CR><LF>'	Error: operating mode failure

Examples

Command	Description
'f0716000000112233<CR>'	Write 3 bytes of data to file starting at offset 000000h, data is 112233h
'f06160000034455<CR>'	Write 2 bytes of data to file starting at offset 000003h, data is 4455h

7.4.6.6 Get Card UID

Use this command to retrieve the real card UID when the transponder is configured to generate random UIDs. A successful authentication with any valid key is required prior issuing this command.

Command: `'f[Length][20]<CR>'`

Parameters	Description
[Length]	Set this value to 01h

Answer

Answer	Description
'UID<CR><LF>'	UID of transponder
'F<CR><LF>'	Error: general failure

7.4.7 Transparent ISO14443-4 Transponder Access ‘t’

This command is used to communicate with ISO14443-4 Type A/B compatible transponders. Data passed as argument is forwarded to the transponder, any response received is transmitted to the host as answer.

The transponder must be selected prior further data access. Basic communication parameters like communication timeout and maximum frame size are negotiated by the reader with the card automatically; this means the commands RATS (ISO14443A) and ATTRIB (ISO14443B) are part of the selection process. The maximum allowed frame size is 40 bytes.

The passed data must comply with ISO14443-4 data framing. For details regarding the MIFARE® DESFire command set, please contact NXP Semiconductors in order to receive the appropriate documentation.

Command: **`'t[DataLength][0F][ISO14443-4 Frame]'`**

Answer

Answer	Description
`'Response<CR><LF>'`	ISO 14443-4 compliant response from transponder
`'N<CR><LF>'`	Error: No response from transponder

Example

Command	Description
`t020FABCD`	Sends two bytes **`ABCD`** to the transponder, received data will be forwarded to the host.

7.5 Setup Related Commands of the Reader

7.5.1 Read Byte from EEPROM ‘rp’

Use this command to read a byte from the internal reader EEPROM. Only the configuration bank is accessible by the user, it is not possible to read any stored keys. The address byte ranges from 00h to FFh.

Command: `'rp[EEPROMAddr]'`

Answer

Answer	Description
`Data<CR><LF>`	EEPROM data (1 byte)

Example

Command	Description
`rp0A`	Reads register 0Ah (Station ID)

7.5.2 Write Byte to EEPROM ‘wp’

Use this command to write a byte to the internal reader EEPROM. Only the configuration bank is accessible by the user, if any keys shall be written, the respective commands have to be used. The address byte ranges from 00h to FFh. A read after write check is performed automatically.

Command: `'wp[EEPROMAddr][Data]'`

Answer

Answer	Description
`Data<CR><LF>`	EEPROM data (1 byte)
`'F<CR><LF>'`	Error: read after write failure

Example

Command	Description
`wp0A15`	Writes 15h to register 0A (Station ID)

7.5.2.1 Factory Reset

The ‘wp’ command contains a special feature that resets the reader to factory configuration. In order to do this, the following command shall be issued:

Command: `'wp0055'`

The reader reloads its factory configuration and performs a software reset automatically.

7.5.3 Set Configuration Flags ‘of’

Use this command to set or clear the Configuration Flags of the reader. In contrast to the EEPROM registers, the settings of the ‘of’ command are volatile, this means, they are taking effect on the behavior of the reader at once and they are lost, when a reset occurs or the reader is powered down.

Command: **`'of[FlagAddr][State]'`**

Answer

Answer	Description
`Data<CR><LF>`	State of the modified flag
`'R<CR><LF>'`	Error: Flag out of range

Flag address	Description
00h	MultiTag
01h	NewSerialMode
05h	ExtendID
07h	NoisyEnvironment
11h	ReqAExtendID
12h	TagInfo

Examples

Command	Description
`of0001`	Set MultiTag flag
`of0501`	Set ExtendID flag
`of0700`	Clear NoisyEnvironment flag

7.5.4 Set Configuration Registers ‘og’

Use this command to set the Configuration registers of the reader. In contrast to the EEPROM registers, the settings of the ‘og’ command are volatile, this means they are taking effect on the behavior of the reader at once and they are lost, when a reset occurs or the reader is powered down.

Command: **`'og[RegAddr][Value]'`**

Answer

Answer	Description
`Data<CR><LF>`	Value of the modified register
`'R<CR><LF>'`	Error: Register out of range

Register address	Description
03h	Reset Off Time
04h	Reset Recovery Time

Example

Command	Description
`og0310`	Set Reset Off Time to 10h (16ms)
`og0420`	Set Reset Recovery Time to 20h (32ms)

7.5.5 Set Tag Type ‘oa’ / ‘ob’ / ‘oj’ / ‘op’ / ‘os’ / ‘ot’

Use this command to search for only one specific transponder type (e.g. only ISO14443A). In contrast to the EEPROM registers, the settings of the ‘o’ command are volatile, this means they are taking effect on the behavior of the reader at once and they are lost when a reset occurs or the reader is powered down.

Command: `'oa' / 'ob' / 'oj' / 'op' / 'os'/ 'ot'`

Answer

Answer	Description
`'OA<CR><LF>'`	Search for ISO14443A transponders
`'OB<CR><LF>'`	Search for ISO14443B transponders
`'OJ<CR><LF>'`	Search for FeliCa transponders
`'OP<CR><LF>'`	Search for PicoPass / ICLASS transponders
`'OS<CR><LF>'`	Search for SRX transponders
`'OT<CR><LF>'`	Search for all supported transponders

7.5.6 Include Tag Type ‘o+a’ / ‘o+b’ / ‘o+j’ / ‘o+p’ / ‘o+s’

Use this command to include specific transponder types to the transponder search. In contrast to the EEPROM register, the settings of the ‘o+’ command are volatile, this means they are taking effect on the behavior of the reader at once and they are lost when a reset occurs or the reader is powered down.

Command: `'o+a' / 'o+b' / 'o+j' / 'o+p' / 'o+s'`

Answer

Answer	Description
`'O+A<CR><LF>'`	Search for ISO14443A transponders
`'O+B<CR><LF>'`	Search for ISO14443B transponders
`'O+J<CR><LF>'`	Search for FeliCa transponders
`'O+P<CR><LF>'`	Search for PicoPass / ICLASS transponders
`'O+S<CR><LF>'`	Search for SRX transponders

7.5.7 Exclude Tag Type ‘o-a’ / ‘o-b’ / ‘o-j’ / ‘o-p’ / ‘o-s’

Use this command to exclude specific transponder types from the transponder search. In contrast to the EEPROM register, the settings of the ‘o-’ command are volatile, this means they are taking effect on the behavior of the reader at once and they are lost when a reset occurs or the reader is powered down.

Command: `'o-a' / 'o-b' / 'o-j' / 'o-p' / 'o-s'`

Example

Answer	Description
`'O-A<CR><LF>'`	Don’t search for ISO14443A transponders
`'O-B<CR><LF>'`	Don’t search for ISO14443B transponders
`'O-J<CR><LF>'`	Don’t search for FeliCa transponders
`'O-P<CR><LF>'`	Don’t search for PicoPass / ICLASS transponders
`'O-S<CR><LF>'`	Don’t search for SRX transponders

7.5.8 Key Management

7.5.8.1 Write MIFARE® Classic Key 'wm'

Use this command to store a MIFARE® Classic authentication key into the EEPROM of the reader. The reader is able to store up 32 keys. The key locations are write-only, so the keys can't be read back. Each key is 6 bytes long and is stored redundantly to increase data integrity.

Command: **`'wm[KeyNumber][Key]'`**

Answer

Answer	Description
Key<CR><LF>	Written key

Example

Command	Description
wm1A1234567890AB	Writes key **1234567890AB** to location 1Ah

7.5.8.2 Write AES / Triple DES Key 'wd'

Use this command to store a 16-Byte Triple-DES authentication key into the EEPROM of the reader. The reader is able to store up to 16 keys. The key locations are write-only, so the keys can't be read back.

Command: **`'wd[KeyNumber][Key]'`**

Answer

Answer	Description
Key<CR><LF>	Written key
'F<CR><LF>'	Error: Writing of key failed
'R<CR><LF>'	Error: Key number out of range

Example

Command	Description
wd0B000102030405060708090A0B0C0D0E0F	Writes Triple-DES key **000102030405060708090A0B0C0D0E0F** to location 0Bh

7.5.9 SAM related Commands

The reader supports the connection of an ISO7816 compatible SAM card (Secure Access Module). The SAM card is driven by the reader via an ISO7816 compatible UART, using the following default parameters: f_{CLK} = 1.1 MHz; F_i = 372; D_i = 1 (resulting baudrate f_{BAUD} = 2957bps).

If the SAM supports PPS exchange (Protocol and Parameter Selection), the card can be switched to higher baudrates.

7.5.9.1 Init SAM 'ei'

Use this command to perform a warm reset procedure of the SAM card according to ISO7816-3. If a SAM card is recognized the reader answers with the card's ATR (Answer To Reset). The command is equipped with a parameter byte which shall be set to 00h.

Command: **`'ei[00]'`**

Answer

Answer	Description
`'ATR<CR><LF>'`	ISO7816 compliant Answer To Reset of the SAM card
`'N<CR><LF>'`	Error: No response from SAM card

7.5.9.2 SAM Transmit Data 'et'

Use this command to transmit data to the SAM card. The card's response is returned as answer. The UART supports T=0 and T=1 protocols according to ISO 7816-3. The command is equipped with a parameter byte, which shall be set to 00h.

The so-called PPS exchange can be done using this command. If the first data byte represents FFh (which is the PPSS), the whole data packet is interpreted as PPS request according to ISO7816-3. The following bytes are interpreted according to the ISO standard. If PPS_1 is present, F_i and D_i are set respectively. If F_i equals 372, f_{CLK} is set to 2.75MHz, in all other cases f_{CLK} is set to 5.5MHz. The ISO7816 UART is reconfigured accordingly, this means all further data exchange is done using the new parameters. The selection of T=0/T=1 protocol is done in PPS_0.

The resulting baudrate is calculated as: $f_{BAUD} = f_{CLK} \times D_i / F_i$

For further details, please refer to ISO7816-3.

Command: **`'et[DataLength][00][Data]'`**

Answer

Answer	Description
`'Response<CR><LF>'`	Response from the card
`'N<CR><LF>'`	SAM card not initialized, perform 'Init SAM' command before
`'F<CR><LF>'`	No response from SAM card

Examples

Command	Description
`et0200ABCD`	Sends two bytes **`ABCD`** to the card, received data will be returned to the host.
`et0400FF119779`	Perform PPS exchange: F_i=512, D_i=64, f_{CLK}=5.5MHz, resulting baudrate: 687500bps Select T=1 protocol

7.6 Miscellaneous Commands

7.6.1 Get Station ID ‘g’

Use this command to retrieve the station ID of the reader.

Command: `'g'`

Answer	Description
`Data<CR><LF>`	Station ID of the reader (1 byte)

7.6.2 Antenna Power Off ‘poff’

Use this command to turn off the RF-field and save energy. All present tags in the antenna field are powered down and reset.

Command: `'poff'`

Answer	Description
`'P<CR><LF>'`	Power off command performed

7.6.3 Antenna Power On ‘pon’

Use this command to turn on the RF-field manually. If a tag-related command is submitted, the RF-field is also turned on.

Command: `'pon'`

Answer	Description
`'P<CR><LF>'`	Power on command performed

7.6.4 Get Version ‘v’

Use this command to receive the current firmware version of the reader.

Command: `'v'`

Answer	Description
`'MifareNFC 1.07<CR><LF>'`	Version string

7.6.5 Reset ‘x’

Use this command to perform a software reset of the reader.

Command: `'x'`

Answer	Description
`'MifareNFC 1.07<CR><LF>'`	Version string

7.6.6 Break

Send a SPACE (DOT if NoisyEnvironment flag is set) to cancel e.g. Continuous Read Mode.

7.6.7 Read GPIO ‘ir’

Use this command to read a specified GPIO. The GPIO is switched to input prior reading.

Command: **‘ir[GPIONumber]’**

Answer

Answer	Description
Data<CR><LF>	Status of specified GPIO: 00 means Low; 01 means High.

Example

Command	Description
‘ir02’	Returns status of GPIO2.

7.6.8 Write GPIO ‘iw’

Use this command to write a specified GPIO. The GPIO is switched to output prior writing. Please consider the reader’s maximum current source and sink capability!

Command: **‘iw[GPIONumber][Status]’**

Answer

Answer	Description
Data<CR><LF>	Status of specified GPIO: 00 means Low; 01 means High.

Example

Command	Description
‘iw0201’	Sets GPIO2 to logical state High
‘iw0300’	Sets GPIO3 to logical state Low

7.6.9 Read GPIO1 ‘pr’

Use this command to read GPIO1. GPIO1 is switched to input prior reading. The command has the same effect as command ‘ir01’. It has been implemented in order to adhere backward compatibility.

Command: **‘pr’**

Answer

Answer	Description
Data<CR><LF>	Status of GPIO1: 00 means Low; 01 means High.

7.6.10 Write GPIO1 ‘pw’

Use this command to write GPIO1. GPIO1 is switched to output prior writing. Please consider the reader’s maximum current source and sink capability! The command has the same effect as command ‘iw010X’. It has been implemented in order to maintain backward compatibility.

Command: **‘pw[Status]’**

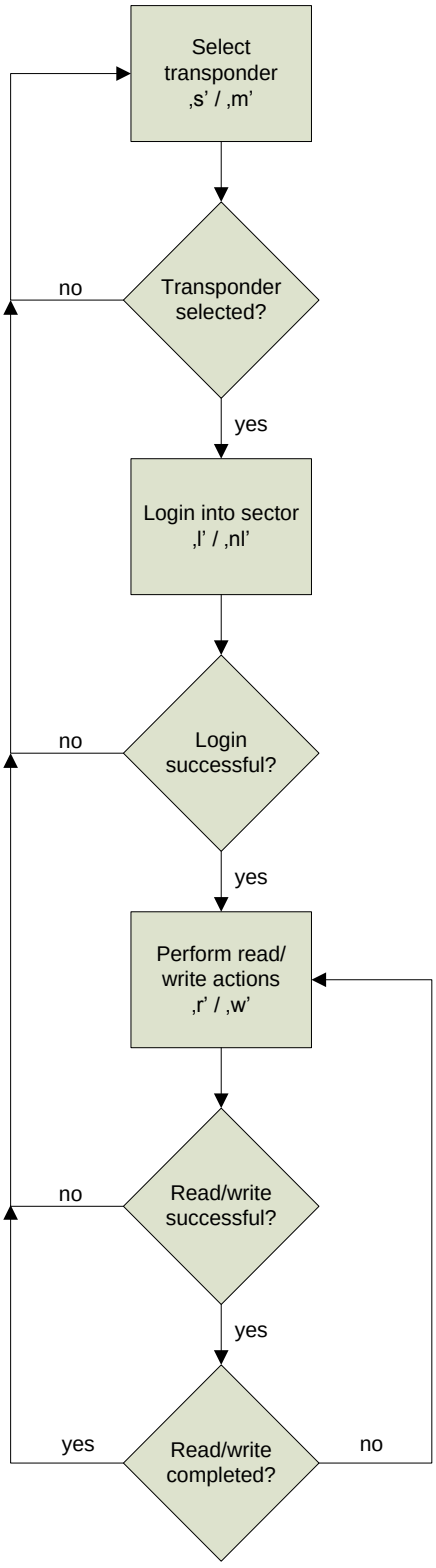
Answer

Answer	Description
Data<CR><LF>	Status of GPIO1: 00 means Low; 01 means High.

8. Typical Data Transaction Procedures

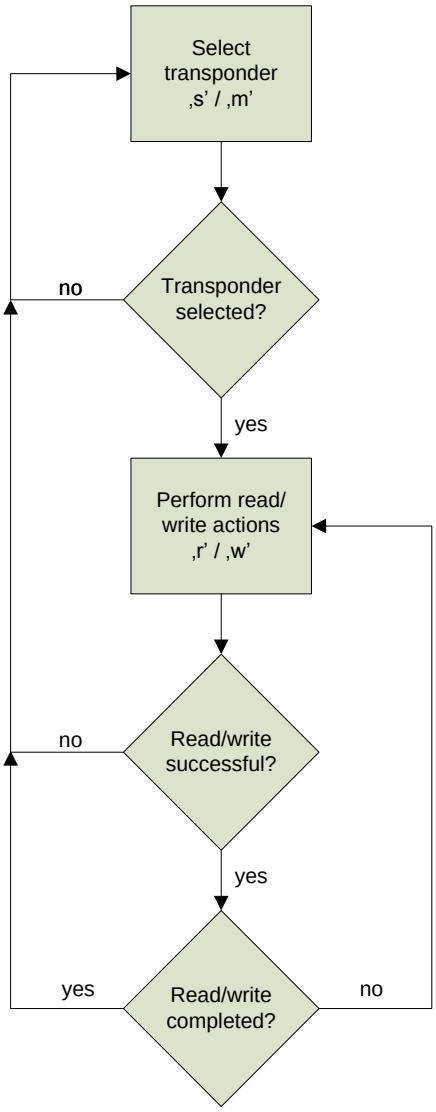
8.1 MIFARE® Classic / MIFARE® PLUS

The following diagram shows the typical command-flow in order to access the data area of a MIFARE® Classic / PLUS transponder:



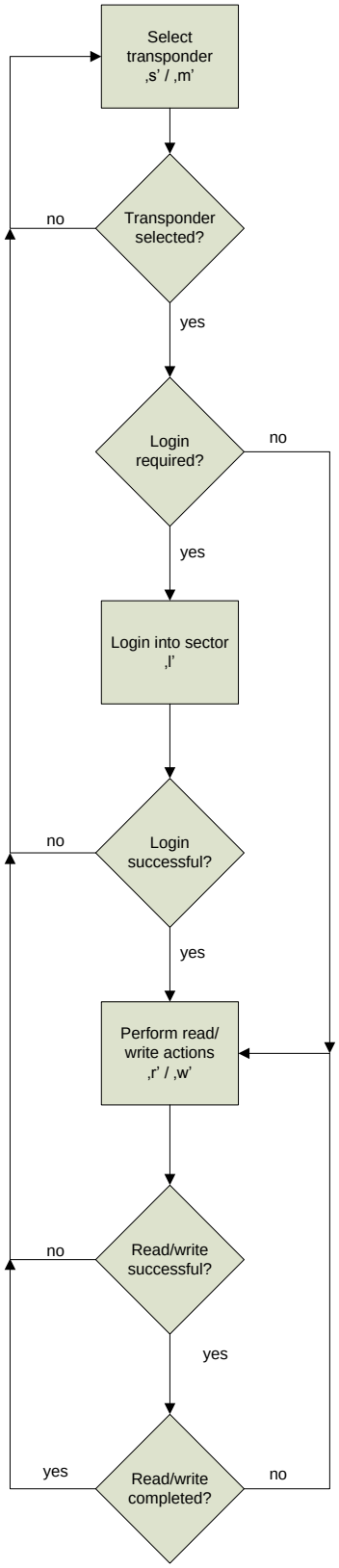
8.2 MIFARE® Ultralight

In contrast to MIFARE® Classic transponders, no login is required to access the EEPROM memory:



8.3 MIFARE® Ultralight C

Accessing the EEPROM depends on the settings of the transponder, e.g. if an authentication is required. The following diagram shows the typical command-flow in order to access the data area of a MIFARE® Ultralight C transponder:



9. Memory Organization of MIFARE® Transponders

9.1 MIFARE® Classic 1k

A MIFARE® Classic 1k transponder consists of 16 sectors. Each sector is organized in four data blocks. A data block stores 16 bytes of data. The following table shows the memory organization of a MIFARE® Classic 1k card:

Sector address	Block addresses			
00h	03h	02h	01h	00h
01h	07h	06h	05h	04h
02h	0Bh	0Ah	09h	08h
03h	0Fh	0Eh	0Dh	0Ch
04h	13h	12h	11h	10h
05h	17h	16h	15h	14h
06h	1Bh	1Ah	19h	18h
07h	1Fh	1Eh	1Dh	1Ch
08h	23h	22h	21h	20h
09h	27h	26h	25h	24h
0Ah	2Bh	2Ah	29h	28h
0Bh	2Fh	2Eh	2Dh	2Ch
0Ch	33h	32h	31h	30h
0Dh	37h	36h	35h	34h
0Eh	3Bh	3Ah	39h	38h
0Fh	3Fh	3Eh	3Dh	3Ch

9.2 MIFARE® Classic 4k

A MIFARE® Classic 4k transponder consists of 40 sectors. In contrast to the MIFARE® standard 1k transponder, the 4k version has a different memory organization, shown in the following table:

	Sector	Block addresses	Login sector	Sector	Block addresses	Login sector
000h - 7FFh	00h	00h...03h	00h	10h	40h...43h	10h
	01h	04h...07h	01h	11h	44h...47h	11h
	02h	08h...0Bh	02h	12h	48h...4Bh	12h
	03h	0Ch...0Fh	03h	13h	4Ch...4Fh	13h
	04h	10h...13h	04h	14h	50h...53h	14h
	05h	14h...17h	05h	15h	54h...57h	15h
	06h	18h...1Bh	06h	16h	58h...5Bh	16h
	07h	1Ch...1Fh	07h	17h	5Ch...5Fh	17h
	08h	20h...23h	08h	18h	60h...63h	18h
	09h	24h...27h	09h	19h	64h...67h	19h
	0Ah	28h...2Bh	0Ah	1Ah	68h...6Bh	1Ah
	0Bh	2Ch...2Fh	0Bh	1Bh	6Ch...6Fh	1Bh
	0Ch	30h...33h	0Ch	1Ch	70h...73h	1Ch
	0Dh	34h...37h	0Dh	1Dh	74h...77h	1Dh
	0Eh	38h...3Bh	0Eh	1Eh	78h...7Bh	1Eh
	0Fh	3Ch...3Fh	0Fh	1Fh	7Ch...7Fh	1Fh
800h - FFFh	20h	80h...8Fh	20h	24h	C0h...CFh	30h
	21h	90h...9Fh	24h	25h	D0h...DFh	34h
	22h	A0h...AFh	28h	26h	E0h...EFh	38h
	23h	B0h...BFh	2Ch	27h	F0h...FFh	3Ch

9.3 MIFARE® PLUS

MIFARE® PLUS 2K/4K transponders have a similar memory organization like MIFARE® Classic 4K. However, a 2Kbyte card only has the lower 32 Sectors built-in:

Bytes	Sector	Block addresses	Sector	Block addresses
000h - 7FFh	00h	00h...03h	10h	40h...43h
	01h	04h...07h	11h	44h...47h
	02h	08h...0Bh	12h	48h...4Bh
	03h	0Ch...0Fh	13h	4Ch...4Fh
	04h	10h...13h	14h	50h...53h
	05h	14h...17h	15h	54h...57h
	06h	18h...1Bh	16h	58h...5Bh
	07h	1Ch...1Fh	17h	5Ch...5Fh
	08h	20h...23h	18h	60h...63h
	09h	24h...27h	19h	64h...67h
	0Ah	28h...2Bh	1Ah	68h...6Bh
	0Bh	2Ch...2Fh	1Bh	6Ch...6Fh
	0Ch	30h...33h	1Ch	70h...73h
	0Dh	34h...37h	1Dh	74h...77h
	0Eh	38h...3Bh	1Eh	78h...7Bh
	0Fh	3Ch...3Fh	1Fh	7Ch...7Fh
800h - FFFh	20h	80h...8Fh	24h	C0h...CFh
	21h	90h...9Fh	25h	D0h...DFh
	22h	A0h...AFh	26h	E0h...EFh
	23h	B0h...BFh	27h	F0h...FFh

9.4 MIFARE® Ultralight

A MIFARE® Ultralight transponder consists of 16 pages with 4 bytes each. The UID is read-only and is also mapped into the memory area.

Page	Byte 0	Byte 1	Byte 2	Byte 3	Access rights
00h	SN0	SN1	SN2	BCC0	Read only
01h	SN3	SN4	SN5	SN6	Read only
02h	BCC1	Internal	Lock0	Lock1	Read only / Lock
03h	OTP0	OTP1	OTP2	OTP3	One time programmable
04h	Data0	Data1	Data2	Data3	Read / Write
05h	Data4	Data5	Data6	Data7	Read / Write
06h	Data8	Data9	Data10	Data11	Read / Write
07h	Data12	Data13	Data14	Data15	Read / Write
08h	Data16	Data17	Data18	Data19	Read / Write
09h	Data20	Data21	Data22	Data23	Read / Write
0Ah	Data24	Data25	Data26	Data27	Read / Write
0Bh	Data28	Data29	Data30	Data31	Read / Write
0Ch	Data32	Data33	Data34	Data35	Read / Write
0Dh	Data36	Data37	Data38	Data39	Read / Write
0Eh	Data40	Data41	Data42	Data43	Read / Write
0Fh	Data44	Data45	Data46	Data47	Read / Write

9.5 MIFARE® Ultralight C

A MIFARE® Ultralight C transponder consists of 48 pages with 4 bytes each. The UID is read-only and is also mapped into the memory area. The 16 bytes Triple-DES key is stored in the last 4 pages of the transponder.

Page	Byte 0	Byte 1	Byte 2	Byte 3	Access rights
00h	SN0	SN1	SN2	BCC0	Read only
01h	SN3	SN4	SN5	SN6	Read only
02h	BCC1	Internal	Lock0	Lock1	Read only / Lock
03h	OTP0	OTP1	OTP2	OTP3	One time programmable
04h	Data0	Data1	Data2	Data3	Read / Write
05h	Data4	Data5	Data6	Data7	Read / Write
⋮	⋮	⋮	⋮	⋮	Read / Write
27h	Data140	Data141	Data142	Data143	Read / Write
28h	Lock2	Lock3	RFU	RFU	Read / Write
29h	CNT	CNT	RFU	RFU	Read / Write
2Ah	AUTH0	RFU	RFU	RFU	Read / Write
2Bh	AUTH1	RFU	RFU	RFU	Read / Write
2Ch	Key1/K0	Key1/K1	Key1/K2	Key1/K3	Write only
2Dh	Key1/K4	Key1/K5	Key1/K6	Key1/K7	Write only
2Eh	Key2/K0	Key2/K1	Key2/K2	Key2/K3	Write only
2Fh	Key2/K4	Key2/K5	Key2/K6	Key2/K7	Write only

9.5.1 Example: Writing the Triple-DES Key

This chapter shows how to write the Triple-DES key into the transponder.

On example of Key1 = 0001020304050607h and Key2 = 08090A0B0C0D0E0Fh the command sequence shown in the table below has to be executed for writing the key. The following preconditions shall be assumed:

UID: 04169BE1ED2580h

K1: 49454D4B41455242h

K2: 214E4143554F5946h

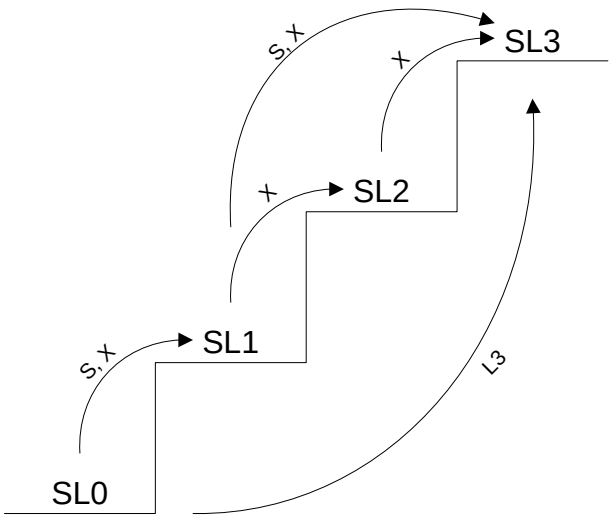
Please note:
Pages 2Ch … 2Fh are write-only, so the ‘w’ command will always report an error, because the automatic read-after-write check fails! Furthermore, the transponder looses its authenticated state. So it is necessary, that each write-cycle is followed by a subsequent login with the updated part of the new key:

Command	Response
s	04169BE1ED2580<CR><LF>
l00CC49454D4B41455242214E4143554F5946	L<CR><LF>
w2C0706050400000000000000000000000000	F<CR><LF>
s	04169BE1ED2580<CR><LF>
l00CC49454D4B04050607214E4143554F5946	L<CR><LF>
w2D0302010000000000000000000000000000	F<CR><LF>
s	04169BE1ED2580<CR><LF>
l00CC0001020304050607214E4143554F5946	L<CR><LF>
w2E0F0E0D0C000000000000000000000000	F<CR><LF>
s	04169BE1ED2580<CR><LF>
l00CC0001020304050607214E41430C0D0E0F	L<CR><LF>
w2F0B0A0908000000000000000000000000	F<CR><LF>
s	04169BE1ED2580<CR><LF>
l00CC000102030405060708090A0B0C0D0E0F	L<CR><LF>

10. MIFARE® PLUS

10.1 Security Levels

MIFARE® PLUS supports four security levels. The initial delivery configuration is SL0. Within this level, cards must be personalized. After that, the card is switched to a higher security level. Once a card has been switched to a higher level, it cannot be switched back to a lower level. Depending on card type, different ways of switching are possible:



SL0: Initial delivery configuration. Only commands ‘Select’, ‘Write Personalization Data’ and ‘Commit Personalization’ work. Depending on transponder type, the ‘Commit Personalization’ command switches the card either to SL1 or SL3.

SL1: MIFARE® Classic compatibility mode. The transponder behaves like a MIFARE® Classic 2K/4K card. If the transponder is a MIFARE® PLUS X card, it can be switched to SL2 or directly to SL3, otherwise only SL3 is possible.

SL2: AES authentication required, Crypto1 session key is generated, communication is furthermore secured by Crypto1.

SL3: Authentication and MACing based on AES. If the transponder is a MIFARE® PLUS X card, the entire communication between reader and transponder is secured by AES cryptography, MIFARE® PLUS S uses plain communication.

10.2 Application Hints

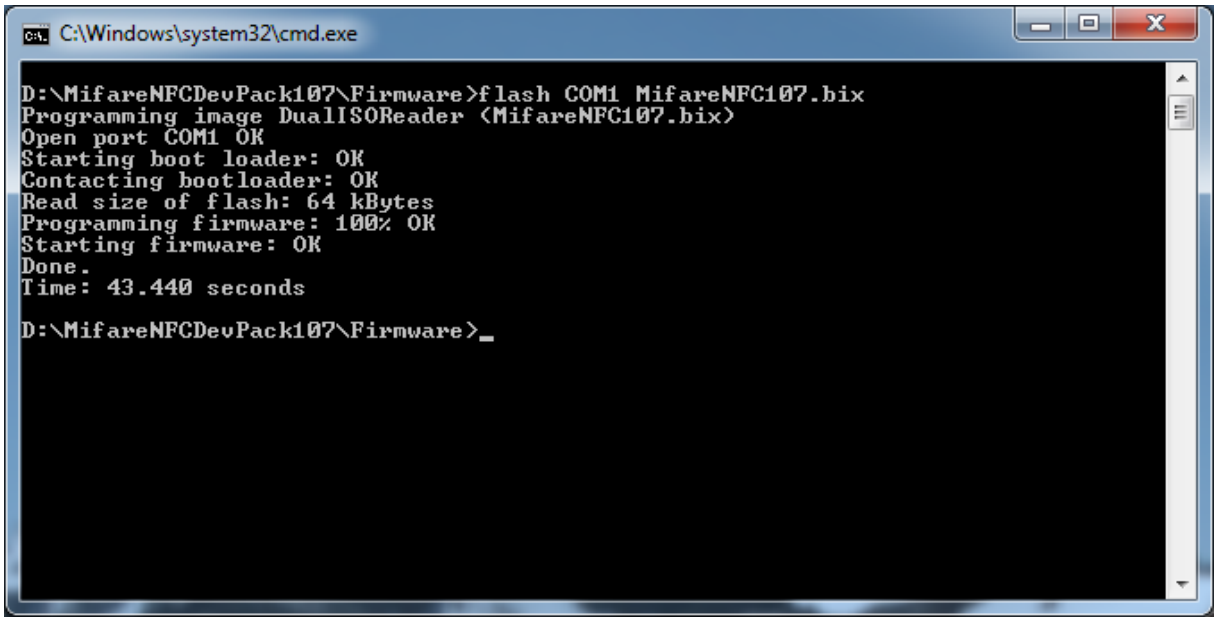
Have a look at the following hints when developing your MIFARE® PLUS application. They will help you choosing the correct transponder and reader in order to satisfy your needs:

- In SL3, value formats are only supported by MIFARE® PLUS X cards
- In SL3, MIFARE® PLUS S transponders use plain communication on the RF-field secured by MACing. MIFARE® PLUS X uses fully AES enciphered communication on the RF-field including MACing.

11. Firmware Update

The reader contains a bootloader program which makes firmware updates possible. Complete the following steps to update the firmware:

- Power up the reader
- Ensure that the reader runs at 9600 bps
- Open the subdirectory “Firmware” in a new command line window and enter the following command: **`flash [COMPort] MifareNFC107.bix`**
- You should see the following screen after successful firmware update:



12. Firmware History

Version	Changes
V1.00	• Initial release
V1.02	• New EEPROM configuration byte StartupDelay
V1.05	• DESFire command set - Authenticate - Select Application - Select File - Read Data - Write Data - Get Card UID
V1.07	• Support of PN512 V2

13. Trademarks

All referenced brands, product names, service names and trademarks mentioned in this document are the property of their respective owners.