NETRONIX

Technical Data Sheet

RFID Reader

MM-R5

Introduction 5

1 Specifications 6

2 Terminal description 7

3 Module dimensions and numbering of terminals 8

4 Serial transmission format 9

Key management 10
4.1.1 Key loading into dynamic key memory 10
4.1.2 Key loading to key static memory 10

Commands for communication with Mifare Classic transponder 11
4.1.3 On/off switching of reader field 11
4.1.4 Selecting one of many transponders 11
4.1.5 Logging by means of Dynamic Key Buffer to selected sector of transponder 12
4.1.6 Logging by means of Static Key Buffer to selected sector of transponder 12
4.1.7 Reading-out the content of transponder block 13
4.1.8 Writing the content of transponder block 13
4.1.9 Copying the content of transponder block into other block 14
4.1.10 Writing values to transponder block 14
4.1.11 Reading-out the values from transponder block 15
4.1.12 Increasing the value included in transponder block 15
4.1.13 Decreasing the value included in block transponder 16
4.1.14 Setting the transponder in field into sleep mode 16

Commands for communication with Mifare Ultralight (C) transponder 17
4.1.15 Ultralight C Authorization 17
4.1.16 Writing the page content into Mifare UL 17
4.1.17 Reading the page content in Mifare UL 17

Mifare Plus commands 18
4.1.18 SL0 level commands 18
4.1.19 SL1 level command set 18
4.1.20 SL3 level command set 19
4.1.21 Mifare Plus operations durance 21

Desfire transponder operation 22
4.1.22 Loading the AES, DES, 3DES keys to reader memory 22
4.1.23 Authorization and logging to an application actually selected 22
4.1.24 Changing the Master key settings in application currently selected 23
4.1.25 Changing the key 24
4.1.26 Creating the application 24
4.1.27 Removing the application 25
4.1.28 Getting the application list 25
4.1.29 Selecting the application 26
4.1.30 Formatting the transponder 26
4.1.31 Initializing the transmission protocol with Desfire transponders 26
4.1.32 Getting the file list of application currently selected 27
4.1.33 Getting the file features 27
4.1.34 Creating the files of *Standard Data* type 28

4.1.35 Creating the files of *Backup Data* type 28
4.1.36 Creating the files of Linear/Cyclic Record type 29
4.1.37 Removing the file 29
4.1.38 Changing the file settings 30
4.1.39 Reading the data from file of *Std/Back Data* type 30
4.1.40 Writing the data to file of *Std/Back Data* type 31
4.1.41 Writing the record to file of *Record Data* type 31
4.1.42 Reading the record from file of *Record Data* type 32
4.1.43 Clearing the files of *Record Data* type 32
4.1.44 Confirmation command - *DesCommit* 32
4.1.45 Deselecting the transponder 33
4.1.46 I-Block transmission T=CL (ISO14443-4) 33

I-CODE SLI transponders 33
4.1.47 Reading ICODE SLI ID 33
4.1.48 SLI page read 34
4.1.49 SLI page write 34

Electrical inputs and outputs 35
4.1.50 Writing output state 35
4.1.51 Reading-out the input state 35
4.1.52 Writing the settings to any port 36
4.1.53 Reading-out the configuration of freely selected port 37

Access password 38
4.1.54 Logging to reader 38
4.1.55 Changing the password 38
4.1.56 Logging out of the reader 39

Autoreader configuration 39
4.1.57 Writing the automatic device configuration 39
4.1.58 Reading-out the configuration of automatic device 41

Configuring the RS232 TTL serial interface 41
4.1.59 Writing the configuration of serial port 41
4.1.60 Reading the configuration of serial interface 42

MAD – Mifare Application Directory 42
4.1.61 Card MAD formatting 42
4.1.62 Adding the application to MAD directory 43
4.1.63 Pursuing the sector for given application 43
4.1.64 Pursuing the next sector of application 44

Other commands 44
4.1.65 Remote reset of reader 44
4.1.66 Reading-out the reader software 45
4.1.67 Setting the date and time 45
4.1.68 Reading-out the date and time 45

Meaning of operation code in response frame 46

Meaning of symbols and markings used in the specification 47

5 Restoring the default settings 47

6 Operation example of transponder 48

An example how to operate the Mifare transponders 48

An example how to operate the Desfire transponders 49

Operation example of Mifare PLUS transponder 50

Introduction

MM-R5 is reader of RFID cards which works on 13,56MHz rated frequency

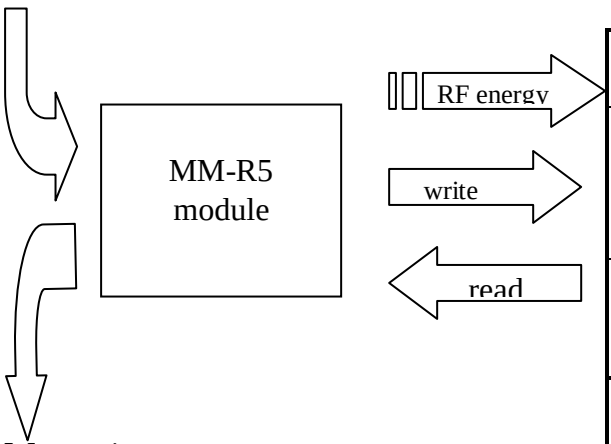
The reader has following functionality:

- Supported transponders: Mifare S50,S70, Mifare Ultralight, , Ultralight C, Mifare Plus S, Mifare Plus X, Mifare DesFire EV1, I-CODE SLI, iClass
- Plain ISO14443-4 supported
- Interface: UART-TTL
- Addressability on UART bus
- Configuring the two-state port inputs/outputs
- Controlling the two-state outputs
- Reading out the two-state inputs
- Possibility of full access to all sectors of Mifare cards on read and write level.
- Module configuration protected with password.

MM-R5 module is a device, which operates on basis of non-contact data reading and writing from and to the Mifare® transponder (RFID). The module is operated via UART interface on voltage levels conforming TTL.
The device operates on basis:
Query (from master device - host) - action (of module) - response (of module).

We send query-command to the MM-R5 module:

module address	frame length	command	data	CRCH,CRCL
xx	xx	xx	xx xx xx	xx xx



sector no.	block no.	Transponder blocks with written data
0	0	B1.....B16
	1	B1.....B16
	2	B1.....B16
	3	B1.....B16
⋮	⋮	⋮
15 or 39		

We receive response:

module address	frame length	response	data	operation code	CRCH,CRCL
xx	xx	xx	xx xx	xx	xx xx

The module has five user ports (one-bit), which can be used for read-out or writing.

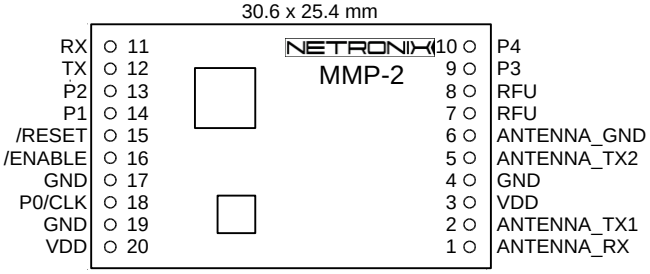
Connect an antenna in form of air-core coil to MM-R5, which will radiate electromagnetic field and supply with it a transponder located in the field.

1 Specifications

Supported functionality depending on transponder / card type:		
Transponder type	**ID reading**	**Full data access**
S50	YES	YES
S70	YES	YES
Ultra Light	YES	YES
DESFire , DESFire EV1	YES	YES
UltraLight C	YES	YES
MIFARE PLUS S MIFARE PLUS X	YES	YES(SL0, SL1, SL3)
I-CODE SLI	YES	YES
I-CLASS	CSN	NO

Module MM-R5 characteristics	
Supply voltage	4.5...5.5 V
Maximum supply current	100 mA
Rated operation frequency of RF module	13.65 MHz
Read distance of transponders	5...10 cm
Antenna	External, including resonance capacitors for 13.56 MHz frequency.
RS-485 transmission	2400, 4800, 9600, 19200, 38400, 57600, 115200 bit/s, 8 data bits, 1 stop bit, no parity compliance with „Netronix Protocol”

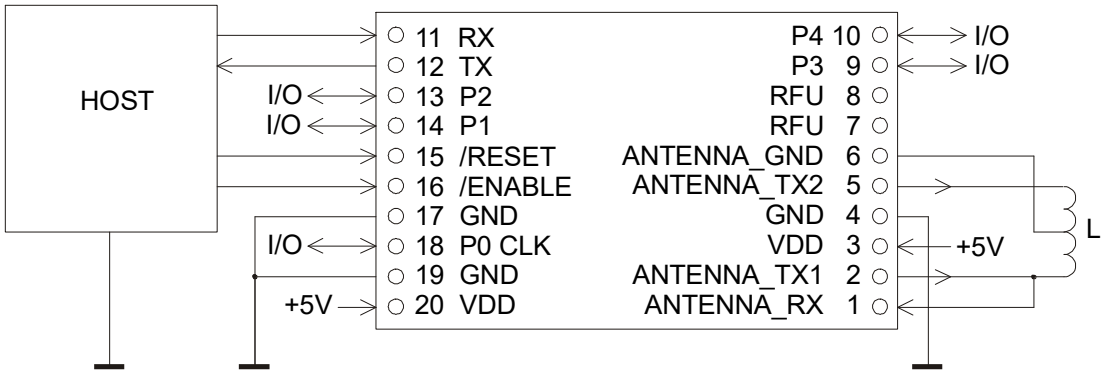
2 Terminal description



Terminal view from component side

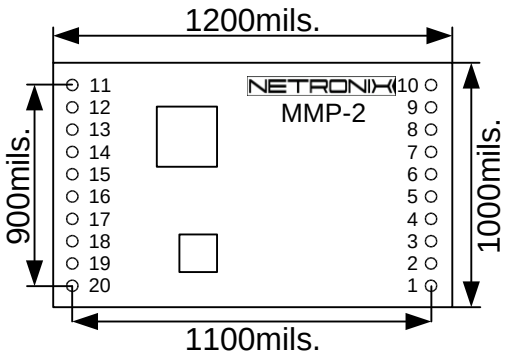
Pin No.	Name	Terminal description
1	ANTENNA_RX	input receiving data from transponder, connected to antenna
2	ANTENNA_TX1	one of outputs supplying energy to antenna
3	VDD	plus of supply voltage
4	GND	circuit ground (minus of supply voltage)
5	ANTENNA_TX2	one of outputs supplying energy to antenna
6	ANTENNA_GND	antenna ground, connected to antenna tap
7	NC	not connected
8	NC	not connected
9	P3	P3 port - user output/input *
10	P4	P4 port – user output/input *
11	UART-RX	UART input on voltage levels complied with TTL *
12	UART-TX	UART output on voltage levels complied with TTL *
13	P2	P2 port - user output/input *
14	P1	P1 port - user output/input *
15	/RESET	input of external reset of the module, active state L *
16	/ENABLE	enable input of the module, active state L *
17,19	GND	circuit ground (minus of supply voltage)
18	P0/CLK	P3 port - user output/input *
20	VDD	plus of supply voltage

* Includes protection resistor 100 Ω (see: Circuit diagram)



Circuit diagram of connections with external components

3 Module dimensions and numbering of terminals



4 Serial transmission format

In this data sheet UART protocol has been confined to descriptions of commands, responses and their parameters. Header and CRC control sum exist always and are compliant with full "Netronix Prtocol" document.

Command frame:

Header	C_CommandName	Response_parameters1…n	CRC

Response frame:

Header	C_CommandName +1	Response_parametrers…m	OperationCode	CRC

RS protocol operation can be tested by means of development tools including free of charge "FRAMER" software".

Key management

Key management feature includes key loading to internal key memory.
For security reasons, these keys cannot be red-out.

To maintain the highest level of data security, employed a particular philosophy of working with these keys.
It allows unit or person who possesses the highest level of confidence to load a key. Such loading operation can be made one time only, or very rarely.
Reader operation in given application is based on using a key not directly, but on recalling key number, to login to sector.
The result is that, in substance, key does not appear in data bus in given application.

Additionally, a user is advised to make sure key should have proper access rights to sectors. This is accomplished by card initialization process, where new confidential keys are loaded to cards with proper access rights, which are assigned to these keys.

Keys A and B are assigned to each sector.
Commands C_LoadKeyToSKB and C_LoadKeyToDKB load these keys to reader memory without information on key type (A or B).
During logging to sector, user has to input as a parameter value of 0xAA or 0xBB, if he wants, the key which is being recalled would be treated as an A or B.

4.1.1 Key loading into dynamic key memory

Dynamic memory features of automatic content delete in case of supply decay. The memory can be overwritten many times.

Command frame:

Header	C_LoadKeyToDKB	Key1…6	CRC

Where:

Parameter name	Parameter description	Value range
C_LoadKeyToDKB	Key loading to key dynamic memory	0x14
Key1…6	6-byte code	whichever

Response frame:

Header	C_LoadKeyToDKB +1		OperationCode	CRC

4.1.2 Key loading to key static memory

Important feature of static memory is that in case of supply decay, data stored in it will not be lost. The memory can be overwritten many times.

Command frame:

Header	C_LoadKeyToSKB	Key1…6, KeyNo	CRC

Where:

Parameter name	Parameter description	Value range
C_LoadKeyToSKB	Key loading to key static memory	0x16
Key1…6	6-byte key	whichever
KeyNo	Key number. It possible to load 32 different keys to a reader.	0x00…0x1f

Response frame:

Header	C_LoadKeyToSKB +1		OperationCode	CRC

Commands for communication with Mifare Classic transponder

4.1.3 On/off switching of reader field

Command frame:

Header	C_TurnOnAntennaPower	State	CRC

Where:

Parameter name	Parameter description	Value range
C_TurnOnAntennaPower	On/off switching of reader field	0x10
State	On state	0x00 – switching the field off 0x01 – switching the field on

Response frame:

Header	C_TurnOnAntennaPower +1		OperationCode	CRC

4.1.4 Selecting one of many transponders

Command frame:

Header	C_Select	RequestType	CRC

Where:

Parameter name	Parameter description	Values
C_Select	Selecting one of many transponders	0x12
RequestType	Type of transponder selection	0x00 - Standard selecting from group of transponders, which are not in stand-by mode 0x01 - Selecting from group of transponders, which are in reader field.

Response frame:

Header	C_Select +1	ColNo, CardType, ID1…….IDn	OperationCode	CRC

Where:

Parameter name	Parameter description	Meaning
ColNo	Number of collisions during one transponder selecting. This figure can be equal to the transponder quantities, which are in the field simultaneously, and which are not in stand-by state.	
CardType	Type of selected transponder	0x50 – S50 0x70 – S70 0x10 – Ultra Light 0xdf – Des Fire
ID1…IDn	Unique number of transponder	ID1 – LSB, IDn – MSB

4.1.5 Logging by means of Dynamic Key Buffer to selected sector of transponder

To complete logging successfully, it is important after any input of the reader, to reload the Dynamic Key Buffer.

Command frame:

Header	C_LoginWithDKB	SectorNo, KeyType, DKNo	CRC

Where:

Parameter name	Parameter description	Value range
C_LoginWithDKB	Logging to sector	0x18
SectorNo	Transponder sector number, to which user wants to login.	0x00 – 0x0f (s50) 0x00 – 0x27 (s70)
KeyType	Key type, which is inside internal Dynamic Key Buffer.	0xAA – key of A type 0xBB – key of B type
DKNo	Dynamic key number	0x00

Response frame:

Header	C_LoginWithDKB +1		OperationCode	CRC

4.1.6 Logging by means of Static Key Buffer to selected sector of transponder

To complete logging successfully, it is important to load Static Key Buffer first.

Command frame:

Header	C_LoginWithSKB	SectorNo, KeyType, SKNo	CRC

Where:

Parmeter name	Parameter description	Value range
C_LoginWithSKB	Logging to sector	0x1a
SectorNo	Transponder sector number, to which user wants to login.	0x00 – 0x0f (s50) 0x00 – 0x27 (s70)
KeyType	Key type, which is inside internal	0xAA – key of A type

	Static Key Buffer.	0xBB – key of B type
SKNo	Static Key number	0x00…0x1F

Response frame:

Header	C_LoginWithSKB +1		OperationCode	CRC

4.1.7 Reading-out the content of transponder block

Command frame:

Header	C_ReadBlock	BlockNo	CRC

Where:

Parameter name	Parameter description	Value range
C_ReadBlock	Read-out of transponder block content	0x1e
BlockNo	Block number within given sector	**Sector and block numeration

Response frame:

Header	C_ReadBlock +1	Data1….. Data16	OperationCode	CRC

Where:

Parameter name	Parameter description	Value range
Data1…. Data16	Red-out of data from transponder block	

4.1.8 Writing the content of transponder block

Command frame:

Header	C_WriteBlock	BlockNo, Data1….. Data116	CRC

Where:

Parameter name	Parameter description	Value range
C_WriteBlock	Write of transponder block content	0x1c
BlockNo	Block number within given sector	**Sector and block numeration
Data1…. Data16	Data, which are to be written into transponder block.	whichever

Response frame:

Header	C_WriteBlock +1		OperationCode	CRC

4.1.9 Copying the content of transponder block into other block

Command frame:

Header	C_CopyBlock	SourceBlockNo, TargetBlockNo	CRC

Where:

Parameter name	Parameter description	Value range
C_CopyBlock	Copying the content of transponder block into other block	0x60
SourceBlockNo	Source block	**Sector and block numeration
TargetBlockNo	Target block for data	

Response frame:

Header	C_CopyBlock +1		OperationCode	CRC

4.1.10 Writing values to transponder block

Command frame:

Header	C_WriteValue	BlockNo, BackupBlockNo,Value1...4,	CRC

Where:

Parameter name	Parameter description	Value range
C_WriteValue	Write of values to transponder block.	0x34
BlockNo	Block number within given sector, into which the Value will be written.	**Sector and block numeration
BackupBlockNo	Declared block number including the Value copy. BackupBlockNo has no influence for system operation, but user can/should make the Value copy by himself.	**Sector and block numeration
Value1...4	The Value, which is written to transponder block.	whichever

Response frame:

Header	C_WriteValue +1		OperationCode	CRC

4.1.11 Reading-out the values from transponder block

Command frame:

Header	C_ReadValue	BlockNo	CRC

Where:

Parameter name	Parameter description	Value range
C_ReadValue	Read-out of the Value from transponder block.	0x36
BlockNo	Block number within given sector, from which the Value will be red-out.	**Sector and block numeration

Response frame:

Header	C_ReadValue+1	Value1...4, BackupBlockNo	OperationCode	CRC

Where:

Parameter name	Parameter description	Value range
Value1...4	Red-out Value from transponder block.	
BackupBlockNo	Block number, which can include the Value copy.	**Sector and block numeration

4.1.12 Increasing the value included in transponder block

To execute a command successfully, format of data included in declared block should be "Value" format.

Command frame:

Header	C_IncrementValue	BlockNo, Value1...4	CRC

Where:

Parameter name	Parameter description	Value range
C_IncrementValue	Increasing the value included in transponder block.	0x30
BlockNo	Block number within given sector, in which the Value will be modified.	**Sector and block numeration
Value1...4	Value, which is being added to existed real value of block transponder.	

Response frame:

Header	C_IncrementValue +1		OperationCode	CRC

4.1.13 Decreasing the value included in block transponder

To execute a command successfully, format of data included in declared block should be "Value" format.

Command frame:

Header	C_DecrementValue	BlockNo, Value1...4	CRC

Where:

Parameter name	Parameter description	Value range
C_DecrementValue	Decreasing the Value included in transponder block.	0x32
BlockNo	Block number within given sector, in which the Value will be modified	**Sector and block numeration
Value1...4	The Value, which is being subtracted from existed real value of block transponder.	whichever

Response frame:

Header	C_DecrementValue+1		OperationCode	CRC

4.1.14 Setting the transponder in field into sleep mode

To set transponder to sleep mode, select it first.

Command frame:

Header	C_Halt		CRC

Parameter name	Parameter description	Value range
C_Halt	Setting the transponder in field into sleep mode.	0x40

Response frame:

Header	C_Halt+1		OperationCode	CRC

Commands for communication with Mifare Ultralight (C) transponder

4.1.15 Ultralight C Authorization

Command frame:

header	C_ULC_Auth	KeyIdx	CRC

Where:

Parameter name	Parameter description	Value range
C_ULC_Auth		0x3C
KeyIdx	Index of key stored In reader	0x00…0x1f

Response frame:

header	C_ ULC_Auth +1		OperationCode	CRC

4.1.16 Writing the page content into Mifare UL

Command frame:

Header	C_WritePage4B	PageAdr, Data1...4	CRC

Where:

Parameter name	Parameter description	Value range
C_WritePage4B	Writing the page content into Mifare UL	0x26
PageAdr	Page number in transponder	0x00…0x0f
Data1...4	Data, which are to be written	whichever

Response frame:

Header	C_WritePage4B +1		OperationCode	CRC

4.1.17 Reading the page content in Mifare UL

Command frame:

Header	C_ReadPage16B	PageAdr	CRC

Where:

Parameter name	Parameter description	Value range
C_ReadPage16B	Read-out of page content in Mifare UL	0x28
PageAdr	Page address, from which read-out of following four pages should start. If PageAdr>0x????, starts read-out process of pages, which are present at memory beginning.	0x00…0x0f

Response frame:

Header	C_ReadPage16B +1	Data1…16	OperationCode	CRC

Where:

Parameter name	Parameter description	Value range
Data1…16	Red-out of data from four subsequent pages.	whichever

Mifare Plus commands

4.1.18 SL0 level commands

4.1.18.1 Write Perso –card initialization

Command frame:

Header	C_MfPlusCMD	0xA8, AdrH, AdrL, Data{0..15}	CRC

Where:

Parameter name	Parameter description	Value range
C_MfPlusCMD	MFPlus common command	0x3A
0xA8	Sub-command ‘Write Perso’	0xA8
AdrH, AdrL	Two bytes number of block or key to init	According to MIFARE PLUS datasheet
Data{0..15}	Keys or data to save	any

Response frame:

header	C_ MfPlusCMD +1		OperationCode	CRC

4.1.18.2 Commit Perso – switch to next SL level

Command frame:

header	C_MfPlusCMD	0xAA	CRC

Where:

Parameter name	Parameter description	Value range
C_MfPlusCMD	MFPlus common command	0x3A
0xAA	Sub-command ‘Commit Perso’	0xAA

Response frame:

header	C_ MfPlusCMD +1		OperationCode	CRC

4.1.19 SL1 level command set

In this level, Mifare PLUS is compatible with Mifare Claasic. All Mifare Claasic commands are available. Additionally new AES authorization command was added.

4.1.19.1 SL1 AES authorization

Command frame:

header	C_MfPlusCMD	0x10, KeyIdx	CRC

Where:

Parameter name	Parameter description	Value range
C_MfPlusCMD	MFPlus common command	0x3A
0x10	Sub-command ‘Authentication SL1’	
KeyIdx	Stored in reader AES key index	0x00-0x1F

Response frame:

header	C_ MfPlusCMD +1		OperationCode	CRC

4.1.19.2 Switch to next SL level / originality check

Switch to next SL level or check originality of transponder is done by successful authorization to specific address.

Command frame:

header	C_MfPlusCMD	0x70, AdrH, AdrL, KeyIdx	CRC

Where:

Parameter name	Parameter description	Value range
C_MfPlusCMD	MFPlus common command	0x3A
0x70	Sub-command ‘First Auth’	
AdrH, AdrL	Two bytes number of key	0x9002 – switch to SL2 0x9003 – switch to SL3 0x8000 – originality check
KeyIdx	Stored in reader AES key index	0x00-0x1F

Response frame:

header	C_ MfPlusCMD +1		OperationCode	CRC

4.1.20 SL3 level command set

4.1.20.1 Establish ISO14443-4 mode

Each SL3 command must be preceded by one-time entry of the transponder into ISO14443-4 mode

Command frame:

header	C_Init_ISO14443-4	CID	CRC

Where:

Parameter name	Parameter description	Value range
C_Init_ISO14443-4		0x3E
CID	CID identifier	0x00

Response frame:

header	C_Init_ISO14443-4+1		OperationCode	CRC

4.1.20.2 Login into sector

Command frame:

header	C_MfPlusCMD	0x1A, Sector, KeyType, KeyIdx	CRC

Where:

Parameter name	Parameter description	Value range
C_MfPlusCMD	MFPlus common command	0x3A
0x1A	Sub-command ‘sector login’	
Sector	Sector number	0x00-0x1f –Plus 2K card 0x00-0x27 –Plus 4k card
KeyType	Key type	0xAA –A key 0xBB –B key
KeyIdx	Stored in reader AES key index	0x00-0x1F

Response frame:

header	C_ MfPlusCMD +1		OperationCode	CRC

4.1.20.3 Read block content

Command frame:

header	C_ MfPlusCMD	read_cmd, block	CRC

Where:

Parameter name	Parameter description	Value range
C_MfPlusCMD	MFPlus common command	0x3A
read_cmd	Read mode type: Cmd. / MAC on command / MAC on resonse / Plain /encrypted 0x30 / Yes / No / Encrypted * 0x31 / Yes / Yes / Encrypted * 0x32 / Yes / No / Plan 0x33 / Yes / Yes / Plan	0x30-0x33
block	block number	0-3 for sector<0x20 0-15 for sector>0x20

*only Plus X transponders

Response frame:

header	C_ MfPlusCMD +1	Data1….. Data16	OperationCode	CRC

Where:

Parameter name	Parameter description	Value range
Data1…. Data16	16 bytes content of block	

4.1.20.4 Write block content

Command frame:

header	C_ MfPlusCMD	write_cmd, block, data0..data15	CRC

Where:

<table>
<tr><th>Parameter name</th><th>Parameter description</th><th>Value range</th></tr>
<tr><td>C_MfPlusCMD</td><td>MFPlus common command</td><td>0x3A</td></tr>
<tr><td>write _cmd</td><td>Write mode type:
<table>
<tr><th>Cmd.</th><th>MAC on command</th><th>MAC on resonse</th><th>Plain /encrypted</th></tr>
<tr><td>0xA0</td><td>Yes</td><td>No</td><td>Encrypted*</td></tr>
<tr><td>0xA1</td><td>Yes</td><td>Yes</td><td>Encrypted*</td></tr>
<tr><td>0xA2</td><td>Yes</td><td>No</td><td>Plain</td></tr>
<tr><td>0xA3</td><td>Yes</td><td>Yes</td><td>Plain</td></tr>
</table></td><td>0xA0-0xA3</td></tr>
<tr><td>block</td><td>Block number</td><td>0-3 for sector<0x20
0-15 for sector>0x20</td></tr>
<tr><td>data0..data15</td><td>16 bytes block data</td><td></td></tr>
</table>

*only Plus X transponders

Response frame:

header	C_ MfPlusCMD +1		OperationCode	CRC

4.1.21 Mifare Plus operations durance

The following specification defines the duration of individual operations, counted from the moment of sending the command frame (RS) to the moment of sending the answer frame (RS).

Operation	The result is correct [ms]	The result is incorrect [ms]
SELECT	14	12
LOGIN SL3	25	100
READ BLOCK	10	100
WRITE BLOCK	13	100

Desfire transponder operation

4.1.22 Loading the AES, DES, 3DES keys to reader memory

Command frame:

header	C_DesSaveKey(0x38)	KeyNo, Key1..n	CRC

Wherein:

Parameter name	Parameter description	Value range
C_DesSaveKey	Key write command to EEPROM memory	0x38
KeyNo	Number of key which is written	0x00..0x1f
Key1..n	8-byte DES key or 16byte AES/3DES key	0x00-0xff

Response frame:

header	C_DesSaveKey +1		OperationCode	CRC

Attention!

Key write to reader memory is a single process, during that the keys are sent via communication interface in an open way. For security purposes, it is recomended to establish keys individually by person with highest confidence level.

During login to individual Desfire card applications or during key change on Desfire card, refer always to key index saved in EEPROM memory, not mentioning it in an open way. For security purposes, there is no possibility of reading the keys, which are written in reader.

4.1.23 Authorization and logging to an application actually selected

Command frame:

header	C_DesAuth (0x42)	KeyNo, EESavedKey, AuthType	CRC

Wherein:

Parameter name	Parameter description	Value range
C_DesAuth	Authorization command	0x42
KeyNo	Key number referring to transponder	0x00..0x10
EESavedKey	Key position in reader memory	0x00..0x1F
AuthType	Typ autoryzacji : 0x0A – DES 0x3A – 3DES 0xAA – AES	0x0A, 0x3A, 0xAA

Response frame:

header	C_DesAuth +1		OperationCode	CRC

4.1.24 Changing the Master key settings in application currently selected

Command frame:

header	C_DesChangeKeySett (0x44)	KeySettings	CRC

Wherein:

Parameter name	**Parameter description**	**Value range**
C_DesChangeKeySett	Command of key setting change	0x44
KeySettings	Configuration byte	0x00..0x0f

Response frame:

header	C_DesChangeKeySett+1		OperationCode	CRC

Configuration byte structure *KeySettings*:

Bit	**Meaning**
0	0 – PICC Master key cannot be modified
	1* – PICC Master key can be modified
1	0 – recall of C_DesGetAppIDs function requires authorization using PICC Master key
	1* – recall of C_DesGetAppIDs function does not require authorization
2	0 – creating/removing the application requires authorization using PICC Master key
	1* - creating new application does not require authorization, removing the application requires authorization using key of given application or by means of PICC Master key
3	0 – changing of PICC Master Key configuration is not possible
	1* - changing of PICC Master Key configuration is possible in case of authorization using this key
4	RFU – 0
5	RFU – 0
6	RFU – 0
7	RFU – 0

* - default setting

4.1.25 Changing the key

Command frame:

header	C_DesChangeKey (0x46)	KeyNo, NewEESavedKey,[PrevEESavedKey]	CRC

Wherein:

Parameter name	Parameter description	Value range
C_DesChangeKey	Key change command	0x46
KeyNo	Key number to change	0x00..0x0D
NewEESavedKey	New key index saved in reader memory	0x00..0x13
PrevEESavedKey	• If key to be changed is not the one in which, current authorization took place, we mention index of current key, which will be changed. • If key to be changed is the same one in which, current authorization took place, the parameter should be left as an empty.	0x00..0x13

Response frame:

header	C_DesChangeKey+1		OperationCode	CRC

4.1.26 Creating the application

Command frame:

header	C_DesCreateApp (0x48)	AId1..3,KeySettings,NumberOfKeys{0..0x0D}	CRC

Wherein:

Parameter number	Parameter description	Value range
C_DesCreateApp	Application creation command	0x48
AId1..3	3-byte identifier of application	0x00..0xFF
KeySettings	Configuration byte (see below)	0x00..0x0F
NumberOfKeys	Number of keys assigned to given application	0x00..0x0D

Response frame:

header	C_DesCreateApp +1		OperationCode	CRC

Configuration byte structure *KeySettings*:

Bit	Meaning
0	0 – Application Master key cannot be modified

	1* – Application Master key can be modified, but it requires authorization using current AppMasterKey.
1	0 – recalling of C_DesGetAppIDs function requires authorization using PICC Master key
	1* – recalling of C_DesGetAppIDs function does not require authorization.
2	0 – creating/removing of file requires authorization using AppMasterKey
	1* - creating/removing of file does not requires authorization using AppMasterKey
3	0 – change of Application Master Key is not possible
	1* - change of Application Master Key configuration is allowed in case of authorization using the key
4	Bit7-Bit4: determine rights to change key parameters
5	0x0* : Master key of application is required to change key settings
6	0x1-0xD : authorization using key with his index is required to change key settings
7	0xE :change of key settings requires authorization using the same key

* - default setting

4.1.27 Removing the application

Command frame:

header	C_DesDeleteApp (0x4a)	AId1..3	CRC

Wherein:

Parameter name	Parameter description	Value range
C_DesDeleteApp	Command of removing the application	0x4a
AId1..3	3-byte identifier of application	0x00..0xFF

Response frame:

header	C_DesCreateApp +1		OperationCode	CRC

4.1.28 Getting the application list

Command frame:

header	C_DesGetAppIDs (0x4c)		CRC

Wherein:

Parameter name	Parameter description	Value range
C_DesGetAppIDs	Command of application list getting	0x4c

Response frame:

header	C_DesGetAppIDs +1	N*{Aid3,Aid2,Aid1}	OperationCode	CRC

"Aid" list with numbers of applications which currently exist is being returned.

4.1.29 Selecting the application

Command frame:

header	C_DesSelectApp (0x4e)	Aid1..3	CRC

Wherein:

Parameter name	Parameter description	Value range
C_DesSelectApp	Command of selecting the application	0x4e
Aid1..3	3-byte identifier of application	0x00-0xff

Response frame:

header	C_DesSelectApp+1		OperationCode	CRC

4.1.30 Formatting the transponder

Command frame:

header	C_DesFormatPICC (0x60)		CRC

Wherein:

Parameter name	Parameter description	Value range
C_DesFormatPICC	Command of transponder formatting	0x60

Using of this command requires authorization by means of PICC Master key.

Response frame:

header	C_DesFormatPICC +1		OperationCode	CRC

4.1.31 Initializing the transmission protocol with Desfire transponders

Command frame:

header	C_DesInitProtocol	CID	CRC

Wherein:

Parameter name	Parameter description	Value range
C_DesInitProtocol	Command of transponder formatting	0x3E
CID	Logic number of selected transponder	0x00-0x0E

Use this command right after selecting the transponder by means of C_Select command. Current reader version allows operation with one Desfire transponder at once. At his time, CID logic number does not mean anything, so it is recommended to use 0 number.

Response frame:

header	C_DesInitProtocol +1		OperationCode	CRC

4.1.32 Getting the file list of application currently selected

Command frame:

header	C_DesGetFileIDs (0x64)		CRC

Wherein:

Parameter name	Parameter description	Value range
C_DesGetFileIDs	Command of getting the file list	0x64

Response frame:

header	C_DesGetAppIDs +1	N*FileNo	OperationCode	CRC

List with numbers of files currently existing in selected application is being returned.

4.1.33 Getting the file features

Command frame:

header	C_DesGetFileSett (0x66)	FileNo	CRC

Wherein:

Parameter name	Parameter description	Value range
C_DesGetFileSett	Command of getting the file features	0x66
FileNo	File identifier	0x00-0x0f

Response frame:

header	C_DesGetAppIDs +1	File params...	OperationCode	CRC

According to type of file, information with following format is being returned:

- For *Standard Data Files* and *Backup Data Files*

1 byte	1 byte	2 bytes	3 bytes
File type	Comm. Sett.	Access right	File size
		LSB MSB	LSB MSB

- For *Value Files* (this type is currently not implemented)

1 byte	1 byte	2 bytes	4 bytes	4 bytes	4 bytes	1 byte
File type	Comm. Sett.	Access right	Lower limit	Upper limit	Limited credit value	Limited credit enable
		LSB MSB	LSB MSB	LSB MSB	LSB MSB	

- For *Linear/Cyclic record files*

1 byte	1 byte	2 bytes	3 bytes	3 bytes	3 bytes
File type	Comm. Sett.	Access right	Record size	Maximum number of records	Current number of records
		LSB MSB	LSB MSB	LSB MSB	LSB MSB

4.1.34 Creating the files of *Standard Data* type

Command frame:

header	C_DesCreateSTDataFile (0x68)	FileNo,ComSett,AccRight1..2,FileSize1..3	CRC

Wherein:

Parameter name	**Parameter description**	**Value range**
C_DesCreateSTDataFile	Command of creating the STD file	0x68
FileNo	File identifier	0..0x0F
ComSett	Transmission type: 0x01 – non-coded 0x03 –DES coded	0x00,0x03
AccRight1..2	Access rights to file, see table below	0x00..0xff
FileSize1..3	3-byte size of file in bytes, in LSB..MSB order	0x00-0xff

Bytes which determine access rights:

15 12	11 8	7 4	3 0
Read Access	**Write Access**	**Read & Write Access**	**Change Right Access**
MBS ***1st byte***		***2nd byte***	LSB

Two bytes of access rights are divided into four 4-bits fields. Each filed can include values from 0x0 – 0xF range.

- Values from 0x0 – 0xD range determine key number, which will have rights to use given operation,
- 0xE value means that given operation doses not require authorization,
- 0xF value means that there is no access to given operation, regardless which key is being used.

Response frame:

header	C_DesCreateSTDataFile +1		OperationCode	CRC

4.1.35 Creating the files of *Backup Data* type

Command frame:

header	C_DesCreateBACKDataFile (0x6a)	FileNo,ComSett,AccRight1..2,FileSize1..3	CRC

Wherein:

Parameter name	Parameter description	Value range
C_DesCreateBACKDataFile	Command of creating BACKUP file	0x6a
FileNo	File identifier	0..0x07
ComSett	Transmission type: 0x01 – non-coded 0x03 –DES coded	0x00,0x03
AccRight1..2	Access rights to file	0x00..0xff
FileSize1..3	3-byte size of file in bytes in LSB..MSB order	0x00-0xff

Response frame:

header	C_DesCreateBACKDataFile +1		OperationCode	CRC

Access rights are determined in the same way as in case of *Standard Data Files*.

Saving the file of *Backup Data* type must be ended by performing the C_DesCommit command.

4.1.36 Creating the files of Linear/Cyclic Record type

Command frame:

header	C_DesCreateRecordFile (0x6c)	FileNo, ComSett, AccRight1..2, RecSize1..3, RecNumb1..3, Cy/Li{0x0C,0x01}	CRC

Wherein:

Parameter name	Parameter description	Value range
C_DesCreateRecordFile	Command of creating the file of *Record* type	0x6c
FileNo	File identifier	0..0x0F
ComSett	Transmission type: 0x01 – non-coded 0x03 –DES coded	0x00,0x03
AccRight1..2	Access rights to file	0x00..0xff
RecSize1..3	3-byte size of record in bytes, in LSB..MSB order	0x00-0xff
RecNumb1..3	3-bytes parameter which determines number of records, in LSB..MSB order	
Cy/Li	0x0c- cycle type 0x0l – linear type	0x0C,0x01

Response frame:

header	C_DesCreateRecordFile+1		OperationCode	CRC

Access rights are determined in the same way as in case of *Standard Data* files.

4.1.37 Removing the file

Command frame:

header	C_DesDeleteFile (0x6e)	FileNo	CRC

Wherein:

Parameter name	Parameter description	Value range
C_DesDeleteFile	Command of deleting the file	0x6e
FileNo	File identifiers	0x00..0x0F

Response frame:

header	C_DesDeleteFile+1		OperationCode	CRC

4.1.38 Changing the file settings

Command frame:

header	C_DesChangeFileSett (0x80)	FileNo, ComSett, AccRight1..2	CRC

Wherein:

Parameter name	Parameter description	Value range
C_DesChangeFileSett	Command of changing the file features	0x80
FileNo	File identifier	0..0x0F
ComSett	Transmission type: 0x01 – non-coded 0x03 – DES coded	0x00,0x03
AccRight1..2	Access rights to file	0x00..0xff

Response frame:

header	C_DesChangeFileSett+1		OperationCode	CRC

Access rights are determined in the same way as in case of creating the *Standard Data* files.

4.1.39 Reading the data from file of *Std/Back Data* type

Command frame:

header	C_DesReadData (0x82)	FileNo, Offset1..3, Length1..3	CRC

Wherein:

Parameter name	Parameter description	Value range
C_DesReadData	Command of reading from file	0x82
FileNo	File identifier	0..0x0F
Offset1..3	3-bytes parameter, which determines location from which we begin to read the file, in LSB..MSB order.	0x00-0xFF
Length1..3	3-bytes parameter which determines byte quantity, which we want to read, in LSB..MSB order	0x00-0x3A

	(it is possible to read up to 58 bytes once)	

Response frame:

header	C_DesReadData +1	n Bytes	OperationCode	CRC

4.1.40 Writing the data to file of *Std/Back Data* type

Command frame:

header	C_DesWriteData (0x84)	FileNo, Offset1..3,Data1..58	CRC

Wherein:

Parameter name	Parameter description	Value range
C_DesWriteData	Command of writing to file	0x84
FileNo	File identifier	0..0x0F
Offset1..3	3-bytes parameter which determines location from which we begin to write, in LSB..MSB order.	0x00-0xFF
Data1..58	Data, we want to write to file, (it is possible to write up to 58 bytes once)	0x00-0xFF

Response frame:

header	C_DesWriteData+1		OperationCode	CRC

4.1.41 Writing the record to file of *Record Data* type

Command frame:

header	C_DesWriteRecord (0x86)	FileNo, Offset1..3,Data1..58	CRC

Wherein:

Parameter name	Parameter description	Value range
C_DesWriteRecord	Command of record writing	0x86
FileNo	File identifier	0..0x0F
Offset1..3	3-bytes parameter which determines location from which we begin to write, in LSB..MSB order (this value should be less than size of one record)	0x00-0xFF
Data1..58	Data, we want write to file, (it is possible to write up to 58 bytes once), sum of this value and offset must be less than one record size)	0x00-0xFF

Response frame:

header	C_DesWriteRecord+1		OperationCode	CRC

Note: Writing the record to file of *Record* type must be ended by performing the C_DesCommit command.

4.1.42 Reading the record from file of *Record Data* type

Command frame:

header	C_DesReadRecord (0x88)	FileNo, WhichRecord1..3,NoOfRecords1..3	CRC

Wherein:

Parameter name	Parameter description	Value range
C_DesReadRecord	Command of record reading	0x88
FileNo	File identifier	0..0x0F
WhichRecord1..3	3-bytes parameter which determines record from which we begin to read, in LSB..MSB order	0x00-0xFF
NoOfRecords1..3	3-bytes parameter which determines number of records for reading, in LSB..MSB order	0x00-0xFF

Response frame:

header	C_DesReadRecord +1	Record data...	OperationCode	CRC

Number of data which has been red can not be higher than 58 bytes, so it should be observed following principle: {NoOfRecords1..3}*record_size < 58bytes

4.1.43 Clearing the files of *Record Data* type

Command frame:

header	C_DesClearRecordFile (0x8a)	FileNo	CRC

Wherein:

Parameter name	Parameter description	Value range
C_DesClearRecordFile	Command of record file clearing	0x8a
FileNo	File identifier	0..0x0F

Response frame:

header	C_DesClearRecordFile+1		OperationCode	CRC

Note: This operation must be ended by performing the C_DesCommit command.

4.1.44 Confirmation command - *DesCommit*

Command frame:

header	C_DesCommit (0x8c)		CRC

Wherein:

Parameter name	Parameter description	Value range
C_DesCommit	Confirmation command	0x8c

Response frame:

header	C_DesCommit+1		OperationCode	CRC

4.1.45 Deselecting the transponder

Command frame:

header	C_DesDeselect (0x8e)		CRC

Wherein:

Parameter name	Parameter description	Value range
C_DesDeselect	Command of deselecting the transponder	0x8e

Response frame:

header	C_DesDeselect+1		OperationCode	CRC

4.1.46 I-Block transmission T=CL (ISO14443-4)

This command allows to transceive data with tranponder that is in ISO14443-4 mode. Only payload must be present in request and only payload is returned.

Command frame:

header	C_TranscIBlock	data	CRC

Where:

Parameter name	Parameter description	Value range
C_TranscIBlock	Transceive command	0xC8
data	I-Block payload	any

Response frame:

header	C_TranscIBlock+1	data	OperationCode	CRC

I-CODE SLI transponders

4.1.47 Reading ICODE SLI ID

Command frame:

Header	C_Inventory		CRC

Where:

Parameter name	Parameter description	Value range
C_Inventory	Reading SLI ID	0x04

Response frame:

header	C_Inventory +1	0,CardType,ID1...ID8	OperationCode	CRC

4.1.48 SLI page read

Command frame:

header	C_SLIReadPage	PageAdr	CRC

Where:

Parameter name	Parameter description	Value range
C_ SLIReadPage	SLI page read	0x2C
PageAdr	Page address	

Response frame:

nagłówek	C_ SLIReadPage +1	Data1…4	OperationCode	CRC

Where:

Parameter name	Parameter description	Value range
Data1…4	Read data.	any

4.1.49 SLI page write

Command frame:

nagłówek	C_SLIWritePage	PageAdr, Data1...4	CRC

Where:

Parameter name	Parameter description	Value range
C_ SLIWritePage	SLI page write	0x2E
PageAdr	Page number	
Data1...4	4 byte of data to write	dowolne

Response frame:

nagłówek	C_ SLIWritePage +1		OperationCode	CRC

Electrical inputs and outputs

The reader has configurable inputs and outputs. All the outputs are TTL levels with 100Ohm resistor in series. Input has TTL levels

4.1.50 Writing output state

Command frame:

Header	C_WriteOutputs	IONo, State	CRC

Where:

Parameter name	Parameter description	Value range
C_WriteOutputs	Description of output state	0x70
IONo	Number of I/O port. It should be set as an output.	0x02…0x06
State	Desired output state	0x00 lub 0x01

Response frame:

Header	C_WriteOutputs +1		OperationCode	CRC

4.1.51 Reading-out the input state

Command frame:

Header	C_ReadInputs	IONo	CRC

Where:

Parameter name	Parameter description	Value range
C_ReadInputs	Read-out of input state	0x72
IONo	Number of I/O port. It should be set as an input.	0x00,0x01,0x07

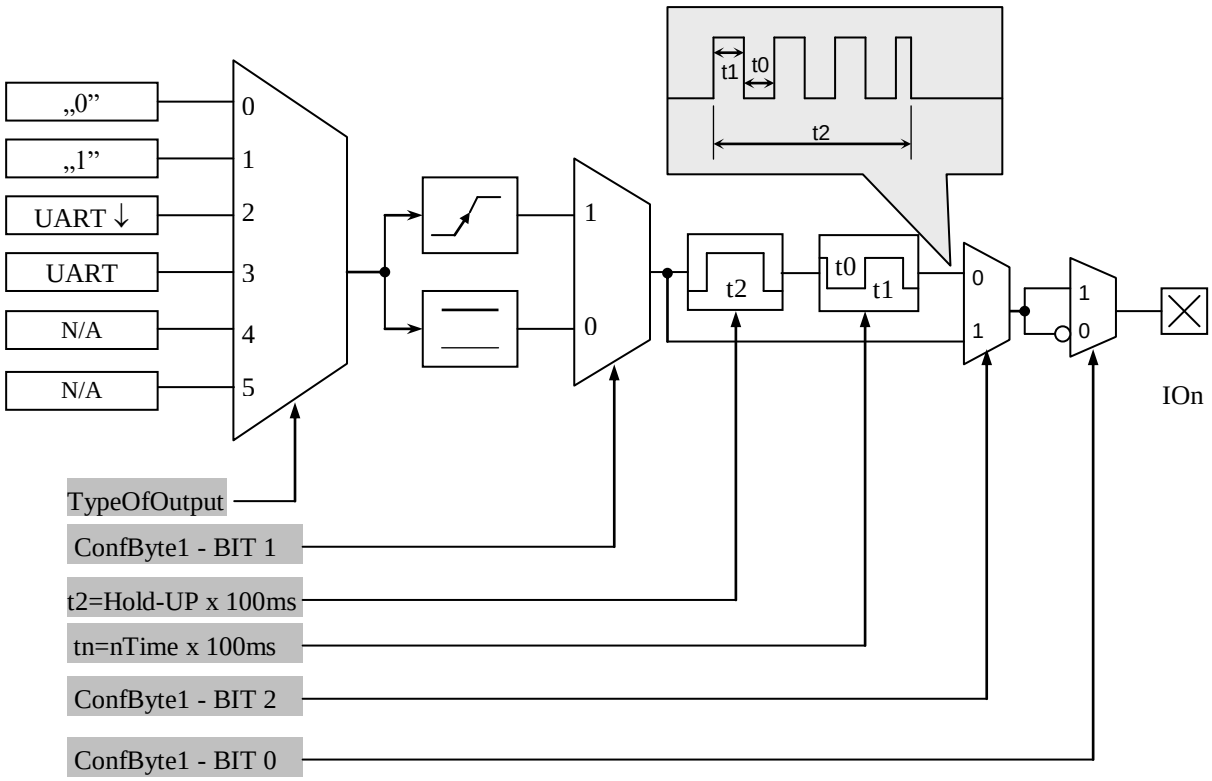
Response frame:

Header	C_ReadInputs +1	State	OperationCode	CRC

Where:

Parameter name	Parameter description	Value range
State	Red-out of output state	

4.1.52 Writing the settings to any port



Command frame:

Header	C_SetIOConfig	IONo, IOConfigData1…n	CRC

If we set a port as output, IOConfigData1…n parameters are as below:

Dir, ConfByte1, TypeOfOutput, Hold-up, 0Time, 1Time

Where:

Parameter name	Parameter description	Value range
C_SetIOConfig	Writing the configuration to any port.	0x50
IONo	Number of I/O port, which is to be configured.	0x02…0x06
Dir	Port direction	0x00 – output
ConfByte1	One byte, in which younger byte defines output type as a NO or NC. Next byte characterizes response manner of given output, as responding for actuation change (slope responding) or responding for actuation state (state responding).	ConfByte1.BIT 0 0-Normally Closed 1-Normally Open ConfByte1.BIT 1 0-level responding 1-slope responding
TypeOfOutput	Source of driving signal	0x00 – permanently off

		0x01 – permanently on 0x02 – driven via serial interface with automatic reset 0x03 – driven via serial
Hold-up	Time of maintaining the on state after actuation stopped. This time is specified as: Hold-up x 100 ms During "hold-up" time, it is possible to configure the output, whish is able to generate rectangular wave. By means of following parameters are configured "Logic 1" time and "Logic 0" time:	
0Time	Logic 0 time	
1Time	Logic 1 time	

If we set a port as a input, IOConfigData1…n parameters would be as below:
Dir, Triger, TypeOfInput, Delay,

Where:

Parameter name	Parameter description	Value range
C_SetIOConfig	Writing the configuration of freely selected port.	0x50
IONo	I/O port number, which is to be configured.	0x00,0x01,0x07
Dir	Port direction	0x01 – input
TypeOfInput	Input type	0x03
Opoznienie	Delay	0x00

4.1.53 Reading-out the configuration of freely selected port

Command frame:

Header	C_GetIOConfig	IONo	CRC

Where:

Parameter name	Parameter description	Value range
C_GetIOConfig	Reading-out the configuration of freely selected port.	0x52
IONo	I/O port number, which configuration is to be red-out.	0x00…0x05

Response frame:

Header	C_GetIOConfig +1	IOConfigData1…n	OperationCode	CRC

Where:

Parameter name	Parameter description	Value range
IOConfigData1…n	This is the same, as in case of configuration write.	

Access password

4.1.54 Logging to reader

Command frame:

Header	C_LoginUser	Data1…n, 0x0	CRC

Where:

Parameter name	Parameter description	Value range
C_LoginUser	Logging to reader	0xb2
Data1…n	This is any byte string	Any from range: 0x01…0xff. String length, which can be 0 to 8 bytes
0x00	Logic Zero, which terminates a string.	0x00

Response frame:

Header	C_LoginUser +1		OperationCode	CRC

4.1.55 Changing the password

Command frame:

Header	C_ChangeLoginUser	Data1…n, 0x0	CRC

Where:

Parameter name	Parameter description	Value range
C_ChangeLoginUser	Password change	0xb4
Data1…n	This is any byte string, which will form valid access password.	Any from range: 0x01…0xff. String length, which can be 0 to 8 bytes
0x00	Logic Zero, which terminates a string.	0x00

If =0x00, a reader will not be protected by password. At any moment, there is possible to set new password later on, to protect the reader by it.

Response frame:

Header	C_ChangeLoginUser+1		OperationCode	CRC

4.1.56 Logging out of the reader

This command sets latest password as an invalid.

Command frame:

Header	C_LogoutUser		CRC

Parameter name	Parameter description	Value range
C_LogoutUser	Logging out of the reader.	0xd6

Response frame:

Header	C_LogoutUser +1		OperationCode	CRC

Autoreader configuration

4.1.57 Writing the automatic device configuration

This command sets operation method of automatic device, reading the unique transponder number UID.

Because of high security level provided by Milfare transponders, there is no possibility of operation of UID reading automatic device and communication with transponders via RS-485 simultaneously.

The reader described below makes possible to hold-on operation of automatic device for a while, in case of suitable transmission via serial interface.

If the reader will operate in mixed mode i.e.:
- automatic reading device UID is enabled and:
- master device (computer, controller) communicates with reader or with transponders via reader,
it is required, to configure the reader correctly, so in case of communication with a reader or transponder, automatic reading device would hold-on its operation.

Command frame:

Header	C_SetAutoReaderConfig	ATrig, AOfflineTime, ASerial, Amode,Abuzz, Amulti	CRC

Where:

Parameter name	Parameter description	Value range
C_SetAutoReaderConfig	Writing the automatic device configuration.	0x58
ATrig	Defines, when automatic reading device UID will operate.	0-automatic device disabled permanently 1-automatic device enabled permanently 2=enabled automatically in case of transmission lack on UART for a time longer than AOfflineTime 3=enabled automatically, in case of no recall of communication commands with transponder for a time longer than AOfflineTime
AOfflineTime	Lack of transmission time on RS485 bus T= AofflineTime * [100ms] Lack of transmission can concern to any commands (Atrig=2), or commands for communication with transponder (Atrig=3). Commands for communication with transponder: C_TurnOnAntennaPower C_Select C_LoginWithDKB C_LoginWithSKB) C_ReadBlock C_WriteBlock C_WritePage4B C_ReadPage16B C_Halt	0x00…0xff
ASerial	Automatic sending the UID transponder number, after reading it automatically from transponder.	0-never 1-for the first applying the transponder only 2-sends all
AMode	Selection the format of sending number 8 bits: MSb LSb R R R CR M E I A	R: Reserved, always 0 CR=1: Number which is ended with line end mark CR+LF M=1: Number which begins with “M”sign E=1: information extended with cards umber in filed and card type (UW-M4x readers only) I=1: Number in reversed order A=1: Number sent in ASCII format

		A=0	Number sent in Nertonix format
ABuzz	Automatic indication of reading by means of buzzer, after automatic UID read-out from transponder.	0-never 1-for the first applying the transponder only 2-indicates all	
AMulti	Transponder to scan MSb LSb S I C M	M- Mifare family C- Calypso (ISO14443B) I – IClass (CSN) S – I-CODE (ISO15693)	

Response frame:

Header	C_ SetAutoReaderConfig +1		OperationCode	CRC

4.1.58 Reading-out the configuration of automatic device

Command frame:

Header	C_ GetAutoReaderConfig		CRC

Where:

Parameter name	Parameter description	Value range
C_GetAutoReaderConfig	Read-out of automatic device configuration.	0x5a

Response frame:

Header	C_ GetAutoReaderConfig +1	ATrig, AOfflineTime, ASerial, ABuzz	OperationCode	CRC

Where:

The meaning of response parameters is the same as described before.

Configuring the RS232 TTL serial interface

4.1.59 Writing the configuration of serial port

Command:

	C_SetInterfaceConfig	Mode, Adr, Baudrate	

Where:

Parameter name	Parameter description	Value range

C_SetInterfaceConfig	Serial interface configuration write	0x54
Mode		0x01
Adr	Address on RS interface	0x01…0xfe
Baudrate	Data baud rate on RS bus	0x01=2400 bps 0x02=4800 bps 0x03=9600 bps 0x04=19200 bps 0x05=38400 bps 0x06=57600 bps 0x07=115200 bps

Response:

	C_SetInterfaceConfig +1		OperationCode	

4.1.60 Reading the configuration of serial interface

Command:

	C_ GetInterfaceConfig		

Where:

Parameter name	Parameter description	Value range
C_GetInterfaceConfig	Serial interface configuration read-out	0x56

Odpowiedź:

	C_ GetInterfaceConfig +1	Mode, Adr, Baudrate	OperationCode	

Where:

The meaning of response parameters is the same as described before.

MAD – Mifare Application Directory

4.1.61 Card MAD formatting

Command frame:

Header	C_FormatMad	Type, Infobyte	CRC

Wherein:

Parameter name	Parameter description	Value range
C_FormatMad **0xA8**	Formatting to MAD	0xA8
Type	1 - MAD1 (15 sectors) 2 – MAD2 (30 sectors)	0x01,0x02
Infobyte	Mark in emitent sector (default 0x00)	0x00-0x1F

Response frame:

Header	C_FormatMad+1		OperationCode	CRC

Notes:

Before you run C_FormatMad command:

- switch AutoReader mode off (using C_SetAutoReaderConfig command)
- load the keys (default 0xff,0xff,0xff,0xff,0xff,0xff)
- turn antenna supply on (using C_TurnOnAntennaPower)
- select the cart (using C_Select command)
- login to sector with number 0, using key of AA type

4.1.62 Adding the application to MAD directory

Command frame:

Header	C_AddApplication	LSB, MSB, Sector	CRC

Wherein:

Parameter name	Parameter description	Value range
C_AddApplication **0xAA**	Adding application	0xAA
LSB	LSB of application number	0x00 - 0xFF
MSB	MSB of application number	0x00 - 0xFF
Sector	Number of sector, in which the application is to be present	0x01-0x0F :MAD1 0x01-0x1F :MAD2

Response frame:

Header	C_AddApplication+1		OperationCode	CRC

Notes:

Application number should be other than 0x0000

Before you run C_AddApplication command:

- switch AutoReader mode off (using command C_SetAutoReaderConfig)
- load the keys (default 0xff,0xff,0xff,0xff,0xff,0xff)
- turn antenna supply on (using C_TurnOnAntennaPower command)
- select the card (using C_Select command)
- login to sector with number 0, using key of AA type

4.1.63 Pursuing the sector for given application

Command frame:

Header	C_GetSectorMad	LSB, MSB	CRC

Wherein:

Parameter name	Parameter description	Value range
C_GetSectorMad **0xAC**	Pursuing the sector	0xAC
LSB	LSB of application number	0x00 - 0xFF
MSB	MSB of application number	0x00 - 0xFF

Response frame:

Header	C_GetSectorMad+1	Sector	OperationCode	CRC

Notes:

Before you run C_GetSectorMad command:

- switch AutoReader mode off (using C_SetAutoReaderConfig command)
- load the keys (using 0xff,0xff,0xff,0xff,0xff,0xff)
- turn antenna supply on (using C_TurnOnAntennaPower command)
- select the card (using C_Select command)
- login to sector with number 0, using key of AA type

If response byte is 0x00, it will mean, that given application is not present in MAD catalogue.

4.1.64 Pursuing the next sector of application

Command frame:

Header	C_GetSectorMadNext	LSB, MSB	CRC

Wherein:

Parameter name	Parameter description	Value range
C_GetSectorMad **0xAE**	Pursuing the next sector	0xAE

Response frame:

Header	C_GetSectorMadNext+1	Sector	OperationCode	CRC

Notes:
Before you run C_GetSectorMadNext command, perform sector searching operation using C_GetSectorMad, command, of which pursuing result was other than 0.

If response byte is 0x00, it will mean, than no more sectors have been found for given application.

Other commands

4.1.65 Remote reset of reader

Command frame:

Header	C_Reset		CRC

Where:

Parameter name	Parameter description	Value range
C_Reset	Remote reader reset	0xd0

Response frame:

Header	C_Reset +1		KodOperacji	CRC

4.1.66 Reading-out the reader software

Command frame:

Header	C_FirmwareVersion		CRC

Where:

Parameter name	Parameter description	Value range
C_FirmwareVersion	Read-out of reader software version	0xfe

Response frame:

Header	C_FirmwareVersion+1	Data1…..n	KodOperacji	CRC

Where:

Data1…n is sequence of dots, which are written as an ASCII codes.

4.1.67 Setting the date and time

Following setting has no influence for reader operation today.

Command frame:

Header	C_SetRtc	Year, Month, Day, Hour, Minute, Second	CRC

Where:

Parameter name	Parameter description	Value range
C_SetRtc	Date and time set-up	0xb8
Year	year	0…99
Month	month	1…12
Day	day	1…31
Hour	hour	0…23
Minute	minute	0…59
Second	second	0…59

Response frame:

Header	C_SetRtc +1		OperationCode	CRC

4.1.68 Reading-out the date and time

Command frame:

Header	C_GetRtc		CRC

Where:

Parameter name	Parameter description	Value range
C_GetRtc	Read-out of date and time	0xb6

Response frame:

Header	C_GetRtc+1	Year, Month, Day, Hour, Minute, Second	OperationCode	CRC

Where:

The meaning of response parameters is the same as described before.

Meaning of operation code in response frame

Operation code name	Description	Value
OC_Error	Error	0x00
OC_ParityError	Parity error	0x01
OC_RangeError	Parameter range error	0x02
OC_LengthError	Data quantity error	0x03
OC_ParameterError	Parameter Error	0x04
OC_Busy	Momentary occupation status of internal modules	0x05
OC_NoACKFromSlave	No internal communication	0x22
OC_CommandUnknown	Unknown command	0x07
OC_WrongPassword	Wrong password or last password expired i.e. automatic LogOut occurred.	0x09
OC_NoCard	No transponder	0x0a
OC_BadFormat	Wrong data format	0x18
OC_FrameError	Transmission error. Noise occurrence possibility.	0x19
OC_NoAnswer	No response from transponder	0x1E
OC_TimeOut	Operation time limit exceeded. Possible the lack of transponder in reader field.	0x16
OC_Successful	Operation finished successfully	0xff

Operation codes connected with DESFIRE transponders		
OC_DesNoChanges	Commit operation does not make any changes	0x0c
OC_DesOutOfEEprom	No EEPROM memory	0x0e
OC_DesIllegalCommand	Not allowed command	0x1c
OC_DesIntegrityError	Error of CRC/ transmission with card	0x1e
OC_DesNoSuchKey	Wrong key number	0x40
OC_DesLengthError	Wrong command length	0x7e
OC_DesPermisionDenied	No rights to perform given operation	0x9d
OC_DesParameterError	Error of command parameter	0x9e
OC_DesApplNotFound	No application about selected Aids	0xa0
OC_DesApplIntegrError	Application error, the application is aborted	0xa1
OC_DesAuthError	Authorization error / wrong key	0xae
OC_DesBoundaryError	Writing/reading from the record crossed the limit	0xbe
OC_DesPICCIntegError	Internal error of transponder, transponder is aborted	0xc1
OC_DesCountError	The limit of 28 applications has been crossed	0xce
OC_DesDuplicateError	Application/File with this identifier does not exist any longer	0xde
OC_DesEepromError	Error of write/ read to/from EEPROM memory	0xee
OC_DesFileNotFound	File with this identifier does not exist any longer	0xf0
OC_DesFileIntegrError	Non-reversible error, file is aborted	0xf1

Meaning of symbols and markings used in the specification

**Sectors and block numeration

For S50 cards:
SectorNo=0x00…0x0f
BlockNo=0x00…0x03

For S70 cards:
SectorNo=0x00…0x20 BlockNo=0x00…0x03
SectorNo=0x21…0x27 BlockNo=0x00…0x0f

5 Restoring the default settings

To restore default settings:
- turn power supply of the module off or set it in reset state
- short P3 and P4 terminals
- turn power supply on or to bring module out of reset state.
- open P3 i P4 terminals

During restoring defaults settings, are fixed following reader parameters:

Parameter name or functionality	Value or setting
Address on serial bus	0x01
Baud rate on serial bus	9600 bps
Access password	00 – no password
Port 0	Output, response for any card during automatic readings, constant H state during presence of a card in field
Port 1	Output, response for any card during automatic readings, rectangular wave during reading the card for the first time
Port 2	Output, response for any card during automatic readings, H state of 100 ms during first read-out of a card
Port 3	Input
Port 4	Input

6 Operation example of transponder

An example how to operate the Mifare transponders

After correct connection of reader and achieving the bi-directional communication between the reader and master computer, it is possible to perform read-out and write operation of transponder memory.
Following operation assumes, that reader is in default condition, and applied S50 card is in default condition too. It means this card has full access rights and both 0xff ff ff ff ff ff keys.

Logging to the reader is to make changes in its factory configuration.
C_LoginUser, 0x31, 0x32, 0x33, 0x34, 0x00

Because during manual experiments, time between subsequent commands sent via serial interface is large and reaches values from some second to some minutes, it is required to disable internal UID automatic read-out device.
It should be done by means of command:
SetAutoReaderConfig with parameters: 0x00, 0x00, 0x00, 0x00.

To read-out the transponder, first load key to key memory.
So load the key to SKB, by means of:
C_LoadKeyToSKB, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00

Enable the field.
TurnOnAntennaPower, 0x01
Apply transponder to reader.

Select transponder
C_Select, 0x00

Login to e.g. sector 3.
C_LoginWithSKB, 0x03, 0xAA, 0x00

Read-out 2nd block content in 3rd sector.
C_ReadBlock, 0x02

If all Operation Codes in response frames were marked as OC_Successful, so obtained values are the values which have been red-out from the block.

An example how to operate the Desfire transponders

After correct connecting the reader and establishing two-way transmission between reader and host computer, it is possible to perform reading/writing operation from/to transponder memory.
Following operations assume that reader is in its defaults state and that settings of Desfire card which is being used are full default settings, it means they have full access rights, and value of PICC master key is 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00.
Aim of this example is to create new application, change standard key of application, create file with data, write data to file and then read data from the file.

Login to reader to make changes in its default configuration.

1. C_LoginUser 0x31, 0x32, 0x33, 0x34, 0x00

Because, during manual experiments, time between subsequent commands sent via RS is relatively high and achieves value from some seconds to some minutes, turn the internal automatic read function UID off.
It should be done by means of command:

2. SetAutoReaderConfig 0x00, 0x00, 0x00, 0x00.

To read the transponder, load keys to key memory first.
We load then standard key of Desfire transponders to e.g. position "3" of the reader memory, and to position "4" of our key, which will be assigned to new application:

3. C_DesSaveKey 0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00

4. C_DesSaveKey 0x04, 0x01, 0x02, 0x03, 0x04, 0x0a, 0x0b, 0x0c, 0x0d

Turn the field on.

5. C_TurnOnAntennaPower 0x01

Apply transponder to the reader and select the transponder.

6. C_Select 0x00

Initialize data exchange protocol ISO with logic number of transponder 0.

7. C_DesInitProtocol 0x00

Perform authorization using „0” key, it means using PICC Master key. This key is written in reader memory under index „3”. DES authorization

8. C_DesAuth 0x00,0x03,0x0A

Create application with identification number e.g. 0x30, 0x10, 0x55, with default setting of ApplicationMasterKey, and with place for 4 keys held in reserve.

9. C_DesCreateApp 0x30,0x10,0x55,0x0F,0x04

Change newly created, default ApplicationMasterKey for one, which is written in reader memory under position 4. In connection with it, select new application:

10. C_DesSelectApp 0x30,0x10,0x55

Login to application using Application Master Key, change it, and then repeat login using new key.

11. C_DesAuth 0x00,0x03
12. C_DesChangeKey 0x00,0x04
13. C_DesAuth 0x00,0x04

Create standard file including full access rights for Application Master Key and read-out rights for key „3". The file will have index „2", non-coded data exchange and size of 1500 bytes.

14. C_DesCreateSTDataFile 0x02,0x00,0x30,0x00,0xDC,0x05,0x00

Write data to newly created file beginning with position 0.

15. C_DesWriteData 0x02,0x00,0x00,0x00, $TuSaNaszeDaneDoZapisu

Read 21 bytes of newly written data.

16. C_DesReadData 0x02,0x00,0x00,0x00, 0x15,0x00,0x00

Operation example of Mifare PLUS transponder

After correct connection of the reader and establishing mutual communication between it and the host computer, read and write operations can be performed on the transponder's memory.
The following operations assume that the reader has factory settings and that an uninitialized new Mifare Plus S 2kB / 4kB card is used.

Below examples presents:

- Loading AES key to reader,
- Loading necessary AES keys to transponder,
- Switching to SL1 level,
- AES authorization on SL1 level,
- Writing lock on SL1,
- Reading block on SL1,
- Switching to SL3 level,
- AES sector authorization,
- Write block data using MAC on command, MAC on response (only available in Mifare Plus S),

- Read block data using MAC on command, MAC on response (only available in Mifare Plus S)

Examples can be realized using free Netronix tool **Framer4** lub **MFPlus Tool**.

Because during manual tests the time between successive commands sent after RS is relatively large and reaches from a few seconds to several minutes, it is necessary to disable the internal UID reading machine.
This should be done using the order:

SetAutoReaderConfig 0x00, 0x00, 0x00, 0x00, 0x00

The first step is loading the keys to the reader's memory. They will then be used to initialize the card, change the SL level and log in to specific sectors of the card.

C_DesSaveKey 0x01, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF

C_DesSaveKey 0x03, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x88, 0x99, 0x00, 0xAA, 0xBB, 0xCC, 0xDD, 0xEE, 0xFF

C_DesSaveKey 0x04, 0x01, 0x02, 0x03, 0x04, 0x0a, 0x0b, 0x0c, 0x0d, 0x0e, 0x0f, 0x10, 0x12, 0x14, 0x16, 0x18, 0x20

And default Mifare Classic key on '0' poison in reader.

C_LoadKeyToSKB 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00

RF field must be turned on.

TurnOnAntennaPower 0x01

Card should be put close to antenna

Transponder must be selected

C_Select 0x00

To write master key 'Card Master Key' (same as we stored on reader at index 0x03)

C_ MfPlusCMD 0xA8 0x90 0x00 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x88, 0x99, 0x00, 0xAA, 0xBB, 0xCC, 0xDD, 0xEE, 0xFF

To write 'SL1 Auth Key' (same as we stored on reader at index 0x04)

C_ MfPlusCMD 0xA8 0x90 0x04 0x01, 0x02, 0x03, 0x04, 0x0a, 0x0b, 0x0c, 0x0d, 0x0e, 0x0f, 0x10, 0x12, 0x14, 0x16, 0x18, 0x20

To write 'Level 3 Switch Auth Key' (same as we stored on reader at index 0x04)

C_ MfPlusCMD 0xA8 0x90 0x03 0x01, 0x02, 0x03, 0x04, 0x0a, 0x0b, 0x0c, 0x0d, 0x0e, 0x0f, 0x10, 0x12, 0x14, 0x16, 0x18, 0x20

To write AES type A key for sector 0x01(same as we stored on reader at index 0x03)
C_ MfPlusCMD 0xA8 0x40 0x02 0x01, 0x02, 0x03, 0x04, 0x0a, 0x0b, 0x0c, 0x0d,

0x0e, 0x0f, 0x10, 0x12, 0x14, 0x16, 0x18, 0x20

Switch to level SL1 is done by command COMMIT PERSO
C_ MfPlusCMD 0xAA

Now card must be reset by sending below command twice
C_Select 0x00

To perform AES authorization using key 4:
C_ MfPlusCMD 0x10 0x04

To login into sector 3 using 'A' key at index 0
C_LoginWithSKB 0x03, 0xAA, 0x00

To write data on 2 block and 3 sector send:

C_WriteBlock 0x02 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88 0x99 0xaa 0xbb 0xcc 0xdd 0xee 0xff 0x00

To write data from 2 block and 3 sector send:

C_ReadBlock 0x02

To switch into ISO14443-4 mode, we must reset card by selecting it agin
C_Select 0x00

Now switch into ISO14443-4 mode is necessary

C_Init_ISO14443-4 0x00

To switch card to level SL3, authorization must be performed:
C_ MfPlusCMD 0x70 0x90 0x03 0x0x04

Now card must be reset by sending below command twice

C_Select 0x00

now switch into ISO14443-4 mode is necessary

C_Init_ISO14443-4 0x00

To login into sector 1 using A key (stored in reader at index 3):
C_ MfPlusCMD 0x1A, 0x01, 0xAA, 0x03

To write block 2 of sector 1 by some examples data:

C_ MfPlusCMD 0xA3 0x02 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88 0x99 0xaa 0xbb 0xcc 0xdd 0xee 0xff 0x00

To read block2 of sector 1:

C_ MfPlusCMD 0x33 0x02

Latest news concerning to NETRONIX products
http://www.netronix.pl/