



ND154-v5 AMS D180222

Technical documentation
RFID reader

PAC-PUG / PAC-PUB PLA-RUP



PAC-PUG



PLA-RUP

Documentation version: ND154-v5
Date: 24.11.2020
Valid from firmware version: PAC-PU-v4.7.A1.2 (dla urządzeń PAC-PUx)
PLA-RUP-4.7.A1.2 (dla urządzeń PLA-RUP)

TABLE OF CONTENTS

1 Introduction 5

1 General Specifications..... 5

2 Serial transmission format 6

2.1 Key management..... 6

2.1.1 Mifare Classic Key loading into dynamic key memory 6

2.1.2 Mifare Classic Key loading to key static memory 6

2.1.3 AES / DES / 3DES keys loading to key static memory 7

2.2 Commands for communication with transponder 7

2.2.1 On/off switching of reader field..... 7

2.2.2 Selecting one of many transponders 7

2.2.3 Setting the transponder in field into sleep mode 8

2.3 Mifare Classic Commands 8

2.3.1 Logging by means of Dynamic Key Buffer to selected sector of transponder 8

2.3.2 Logging by means of Static Key Buffer to selected sector of transponder 8

2.3.3 Reading-out the content of transponder block 9

2.3.4 Writing the content of transponder block 9

2.3.5 Copying the content of transponder block into other block..... 9

2.3.6 Writing values to transponder block..... 10

2.3.7 Reading-out the values from transponder block 10

2.3.8 Increasing the value included in transponder block 10

2.3.9 Decreasing the value included in block transponder 10

2.4 Mifare Ultralight, Mifare Ultralight C Commands 11

2.4.1 Writing the page content into Mifare UL..... 11

2.4.2 Reading the page content in Mifare UL 11

2.4.3 Ultralight C Authorization 11

2.5 Mifare Plus commands..... 12

2.5.1 SL0 level commands..... 12

2.5.1.1 Write Perso –card initialization..... 12

2.5.1.2 Commit Perso – switch to next SL level 12

2.5.2 SL1 level command set..... 12

2.5.2.1 SL1 AES authorization 12

2.5.2.2 Switch to next SL level / originality check..... 12

2.5.3 SL3 level command set..... 13

2.5.3.1 Establish ISO14443-4 mode 13

2.5.3.2 Login into sector..... 13

2.5.3.3 Read block content 13

2.5.3.4 Write block content 14

2.5.4	Czasy trwania operacji dla Mifare Plus	14
2.6	Mifare DESFire, DESFire EV1 commands	14
2.6.1	Authorization, logging in to the currently selected applications	14
2.6.2	Changing the Master key settings of the currently selected applications	15
2.6.3	Key change	15
2.6.4	Tworzenie aplikacji	16
2.6.5	Deleting application	16
2.6.6	Downloading list of applications	16
2.6.7	Application selection	17
2.6.8	Transponder formatting	17
2.6.9	Initialization of the transmission protocol with DESFire transponders	17
2.6.10	Downloading list of files of the currently selected application	17
2.6.11	Downloading file properties	18
2.6.12	Creating Standard Data Files type	18
2.6.13	Creating Backup Data Files type	19
2.6.14	Creating Linear/Cyclic Record Files type	19
2.6.15	Deleting file	19
2.6.16	Reading-out data from <i>Std/Back Data File</i> type	20
2.6.17	Writing data to <i>Std/Back Data File</i> type	20
2.6.18	Writing record to <i>Record Data File</i> type	21
2.6.19	Reading-out a record from <i>Record Data File</i> type	21
2.6.20	Clearing Record Data File type	21
2.6.21	Confirmation command - <i>DesCommit</i>	22
2.6.22	Transponder deselection	22
2.7	I-Block transmission command according to ISO14443-4 protocol	22
2.8	Mifare Application Directory - MAD	22
2.8.1	FormATTING the MAD card	22
2.8.2	Adding the application to the MAD directory	23
2.8.3	Search sector for a given application	23
2.8.4	Searching for the next application sector	24
2.9	Electrical inputs and outputs	24
2.9.1	Describing the output state	24
2.9.2	Writing the configuration of any port	24
2.9.3	Reading-out the configuration of freely selected port	25
2.10	Access password	25
2.10.1	Logging to reader	25
2.10.2	Changing the password	26
2.10.3	Logging out of the reader	26
2.11	Autoreader function	26
2.11.1	Write an Autoreader configuration	27
2.11.2	Read the Autoadera configuration	28

2.12	Support for ID stored in transponder memory	28
2.12.1	Write/Read configuration	28
2.13	Configuring the USB interface	28
2.13.1	Writing the configuration of serial interface	28
2.13.2	Reading the configuration of serial interface	29
2.14	Other commands.....	29
2.14.1	Remote reset of reader	29
2.14.2	Turning on/off keyboard emulation function	29
2.14.3	Reading-out the reader software	30
2.15	Meaning of operation code in response frame	31
3	Keyboard emulation.....	32
3.1	ID formats.....	32
4	Reset to default settings	32
5	Examples of working with transponders	33
5.1	Operation example of S50, S70 transponder	33
5.2	Example of working with DESFire transponders	33
5.3	Operation example of Mifare PLUS transponder	34

1 INTRODUCTION

PAC-PUx / PLA-RUP is reader of RFID cards which works on 13,56MHz rated frequency.

It features following functionality:

- Supported transponders: Mifare S50, Mifare S70, Mifare Ultralight, , Ultralight C, Mifare Plus S, Mifare Plus X, Mifare DesFire, Mifare DESFire EV1
- ISO14443-4 supported
- USB interface (composite device)
 - Class CDC (serial port emulation)
 - Class (HID keyboard emulation)
- Internal buzzer
- Built-in
 - PAC-PUx: two LED's of common purpose and diode as a supply indicator
 - PLA-RUP: two LED's of common purpose
- Built-in switch for reset to factory settings
- Full access possibility to all sectors of Mifare cards on reading and writing level.
- Integrated MAD mechanism (Mifare Application Directory)
- Data protection with password
- Software update via USB interface using NEFIR program

1 GENERAL 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)

PAC-PUx module parameters

Supply voltage	5 V(USB)
Max. supply current	200 mA
Rated operation radio frequency of module	13,56 MHz
Read-out distance between transponders	Up to 7 cm
Dimension	92x146x29
USB communication	CDC class – serial port emulation compliant with „Netronix Protocol” HID calss - keyboard
Temperature	0-50st.c

2 SERIAL TRANSMISSION FORMAT

After drivers (http://www.netronix.pl/attachments/article/155/PAC-PUx_Driver.zip) installation, PAC-PUx reader is seen by PC port as a virtual serial port.

In this data sheet USB serial 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". This application can downloaded form our website: www.netronix.pl/en/software

2.1 KEY MANAGEMENT

Key management feature includes key loading to internal key memory.

For security reasons, these keys cannot be red-out. There are two memory areas, separately for Mifare Classic card keys, separately for AES128bit and 3DES keys.

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.

2.1.1 MIFARE CLASSIC 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	any

Response frame:

Header	C_LoadKeyToDKB +1	OperationCode	CRC
--------	-------------------	---------------	-----

2.1.2 MIFARE CLASSIC 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
--------	-------------------	---------------	-----

2.1.3 AES / DES / 3DES KEYS 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_DesSaveKey	KeyNo, Key0..Key15	CRC
--------	--------------	--------------------	-----

Where:

Parameter name	Parameter description	Value range
C_DesSaveKey	Key loading to key static memory	0x38
KeyNo	Key number. It possible to load 32 different keys to a reader.	0x00...0x1f
Key0...15	16-byte key	any

Response frame:

Header	C_DesSaveKey +1	OperationCode	CRC
--------	-----------------	---------------	-----

2.2 COMMANDS FOR COMMUNICATION WITH TRANSPONDER

2.2.1 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
--------	-------------------------	---------------	-----

2.2.2 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

2.2.3 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
--------	----------	--	---------------	-----

2.3 MIFARE CLASSIC COMMANDS

2.3.1 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
--------	-------------------	--	---------------	-----

2.3.2 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:

Parameter 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 Static Key Buffer.	0xAA – key of A type 0xBB – key of B type
SKNo	Static Key number	0x00...0x1F

Response frame:

Header	C_LoginWithSKB +1		OperationCode	CRC
--------	-------------------	--	---------------	-----

2.3.3 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	

2.3.4 WRITING THE CONTENT OF TRANSPONDER BLOCK

Command frame:

Header	C_WriteBlock	BlockNo, Data1..... Data16		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.	any

Response frame:

Header	C_WriteBlock +1		OperationCode	CRC
--------	-----------------	--	---------------	-----

2.3.5 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
--------	----------------	--	---------------	-----

2.3.6 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
--------	-----------------	--	---------------	-----

2.3.7 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

2.3.8 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
--------	---------------------	--	---------------	-----

2.3.9 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
--------	--------------------	---------------	-----

2.4 MIFARE ULTRALIGHT, MIFARE ULTRALIGHT C COMMANDS

2.4.1 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	any

Response frame:

Header	C_WritePage4B +1	OperationCode	CRC
--------	------------------	---------------	-----

2.4.2 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.	any

2.4.3 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
--------	---------------	--	---------------	-----

2.5 MIFARE PLUS COMMANDS

2.5.1 SLO LEVEL COMMANDS

2.5.1.1 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
--------	----------------	--	---------------	-----

2.5.1.2 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
--------	----------------	--	---------------	-----

2.5.2 SL1 LEVEL COMMAND SET

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

2.5.2.1 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
--------	----------------	--	---------------	-----

2.5.2.2 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
--------	----------------	---------------	-----

2.5.3 SL3 LEVEL COMMAND SET

2.5.3.1 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
--------	---------------------	---------------	-----

2.5.3.2 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
--------	----------------	---------------	-----

2.5.3.3 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:	0x30-0x33

	Cmd.	MAC on command	MAC on response	Plain /encrypted	
	0x30	Yes	No	Encrypted*	
	0x31	Yes	Yes	Encrypted*	
	0x32	Yes	No	Plan	
	0x33	Yes	Yes	Plan	
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	

2.5.3.4 WRITE BLOCK CONTENT

Command frame:

header	C_MfPlusCMD	write_cmd, block, data0..data15	CRC
--------	-------------	---------------------------------	-----

Where:

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

*only Plus X transponders

Response frame:

header	C_MfPlusCMD +1	OperationCode	CRC
--------	----------------	---------------	-----

2.5.4 CZASY TRWANIA OPERACJI DLA MIFARE PLUS

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

2.6 MIFARE DESFIRE, DESFIRE EV1 COMMANDS

2.6.1 AUTHORIZATION, LOGGING IN TO THE CURRENTLY SELECTED APPLICATIONS

Command frame:

header	C_DesAuth (0x42)	KeyNo{0..0x10}, KeyIdx, AuthType	CRC
--------	------------------	----------------------------------	-----

Where:

Parameter name	Parameter description	Value range
C_DesAuth	Authorization command	0x42
KeyNo	Key number in relation to the transponder	0x00..0x10
KeyIdx	Index of the AES key stored in the reader	0x00-0x1F
AuthType	Authorization type: 0x0A – DES 0xAA - AES	0x0A, 0xAA

Response frame:

header	C_DesAuth+1	OperationCode	CRC
--------	-------------	---------------	-----

2.6.2 CHANGING THE MASTER KEY SETTINGS OF THE CURRENTLY SELECTED APPLICATIONS

Command frame:

header	C_DesChangeKeySett (0x44)	KeySettings	CRC
--------	---------------------------	-------------	-----

Where:

Parameter name	Parameter description	Value range
C_DesChangeKeySett	Command for changing key settings	0x44
KeySettings	Configurational byte	0x00..0x0f

Response frame:

header	C_DesChangeKeySett+1	OperationCode	CRC
--------	----------------------	---------------	-----

Structure of configurational byte KeyStettings:

Bit	Meaning
0	0 – PICC Master key is non-modifiable
	1* – PICC Master key is modifiable
1	0 – calling C_DesGetAppIDs function requires authorization using PICC Master key
	1* – calling C_DesGetAppIDs function does not require authorization
2	0 – creating/deleting the file requires authorization using PICC MasterKey
	1* -creating/deleting the file does not require authorization using PICC MasterKey
3	0 – it is not possible to change the configuration of the PICC Master Key
	1* - change of configuration of the PICC Master Key is allowed in case of authorization using this key
4	RFU – 0
5	RFU – 0
6	RFU – 0
7	RFU – 0

* - default setting

2.6.3 KEY CHANGE

Command frame:

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

Where:

Parameter name	Parameter description	Value range
C_DesChangeKey	Key change command	0x46
KeyNo	Key number to be changed	0x00..0x0D
NewEESavedKey	Index of the new key stored in the reader's memory	0x00..0x13
PrevEESavedKey	If the changed key is not the one after which the actual authorization occurred, we give the index of the current key that will be changed If the changed key is the same, after which the current authorization occurred, this parameter is left blank	0x00..0x13

Response frame:

header	C_DesChangeKey+1		OperationCode	CRC
--------	------------------	--	---------------	-----

2.6.4 TWORZENIE APLIKACJI

Command frame:

header	C_DesCreateApp (0x48)	Ald1..3,KeySettings1, KeySettings2		CRC
--------	-----------------------	------------------------------------	--	-----

Where:

Parameter name:	Parameter description	Value range
C_DesCreateApp	Application creation command	0x48
Ald1..3	3-byte application ID	0x00..0xFF
KeySettings1	Configurational byte (look below)	0x00..0x0F
KeySettings2	Bit3..bit0: Number of keys assigned to the application Bit7..Bit6: 00 – DES authorization for the entire application 10- AES authorization for the entire application	0x00..0x0D

Response frame:

header	C_DesCreateApp +1		OperationCode	CRC
--------	-------------------	--	---------------	-----

Structure of configurational byte KeyStettings:

Bit	Meaning
0	0 – Application Master key is unmodified
	1* – Application Master key is modifiable, requires authorization using the existing AppMasterKey key
1	0 – calling C_DesGetApIDs function requires authorization using PICC Master key
	1* – calling C_DesGetApIDs function does not require authorization
2	0 - creating/deleting the file requires authorization using AppMasterKey
	1* - creating/deleting the file does not require authorization with the use of AppMasterKey
3	0 - it is not possible to change the configuration of the Application Master Key
	1* - changing the configuration of the Application Master Key allowed in the case of authorization using this key
4	Bit7-Bit4: determine the rights to change key parameters
5	0x0*:application Master key is necessary to change the key settings
6	0x1-0xD : authorization with a key with this index is necessary to change the key settings
7	0xE :changing the key settings requires authorization using the same key

* - default setting

2.6.5 DELETING APPLICATION

Command frame:

header	C_DesDeleteApp (0x4a)	Ald1..3		CRC
--------	-----------------------	---------	--	-----

Where:

Parameter name	Parameter description	Value range
C_DesDeleteApp	Delete application command	0x4a
Ald1..3	3-byte application ID	0x00..0xFF

Response frame:

header	C_DesCreateApp +1		OperationCode	CRC
--------	-------------------	--	---------------	-----

2.6.6 DOWNLOADING LIST OF APPLICATIONS

Command frame:

header	C_DesGetAppIDs (0x4c)			CRC
--------	-----------------------	--	--	-----

Where:

Parameter name	Parameter description	Value range
C_DesGetAppIDs	Application list download command	0x4c

Response frame:

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

List of Aid numbers, currently existing in applications, is returned.

2.6.7 APPLICATION SELECTION

Command frame:

header	C_DesSelectApp (0x4e)	Aid1..3	CRC
--------	-----------------------	---------	-----

Where:

Parameter name	Parameter description	Value range
C_DesSelectApp	Application selection command	0x4e
Aid1..3	3-byte application ID	0x00-0xff

Response frame:

header	C_DesSelectApp+1	OperationCode	CRC
--------	------------------	---------------	-----

2.6.8 TRANSPONDER FORMATTING

Command frame:

header	C_DesFormatPICC (0x60)	CRC
--------	------------------------	-----

Where:

Parameter name	Parameter description	Value range
C_DesFormatPICC	Transponder formatting command	0x60

Execution of this command requires authorization using the PICC Master key.

Response frame:

header	C_DesFormatPICC +1	OperationCode	CRC
--------	--------------------	---------------	-----

2.6.9 INITIALIZATION OF THE TRANSMISSION PROTOCOL WITH DESFIRE TRANSPONDERS

Command frame:

header	C_DesNitProtocol	CID	CRC
--------	------------------	-----	-----

Where:

Parameter name	Parameter description	Value range
C_DesNitProtocol	Transponder formatting command	0x3E
CID	Logical number of the selected transponder	0x00-0x0E

This command must appear immediately after selecting the transponder with the C_Select command. The current version of the reader allows you to work with one Desfire transponder simultaneously. The CID logical number does not currently matter, it is recommended to enter the number 0.

Response frame:

header	C_DesNitProtocol +1	OperationCode	CRC
--------	---------------------	---------------	-----

2.6.10 DOWNLOADING LIST OF FILES OF THE CURRENTLY SELECTED APPLICATION

Command frame:

header	C_DesGetFileIDs (0x64)	CRC
--------	------------------------	-----

Where:

Parameter name	Parameter description	Value range
C_DesGetFileIDs	File list download command	0x64

Response frame:

header	C_DesGetAppIDs +1	N*FileNo	OperationCode	CRC
--------	-------------------	----------	---------------	-----

List of file numbers currently existing in the selected application is returned.

2.6.11 DOWNLOADING FILE PROPERTIES

Command frame:

header	C_DesGetFileSett (0x66)	FileNo	CRC
--------	-------------------------	--------	-----

Where:

Parameter name	Parameter description	Value range
C_DesGetFileSett	File properties download command	0x66
FileNo	File ID	0x00-0x0f

Response frame:

header	C_DesGetAppIDs +1	File params...	OperationCode	CRC
--------	-------------------	----------------	---------------	-----

Depending on the type of file, information is returned in the following format:

For files: 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 files: 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 enable	credit
		LSB	MSB	LSB	MSB	LSB	MSB	LSB	MSB		

For files: 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

2.6.12 CREATING STANDARD DATA FILES TYPE

Command:

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

Where:

Parameter name	Parameter description	Value range
C_DesCreateSTDDataFile	Creating STD file command	0x68
FileNo	File ID	0..0x0F
ComSett	Transmission type: 0x00 – unencrypted 0x03 – encrypted DES	0x00,0x03
AccRight1..2	Access rights to the file, see table below	0x00..0xff
FileSize1..3	3-byte file size in bytes, in the order of LSB..MSB	0x00-0xff

Bytes specifying 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-bit fields. Each field can contain values from the range 0x0 - 0xF

- Values from the range 0x0 - 0xD specify the key number, which will have the right to perform the given operation,
- The value 0xE means that the operation does not require authorization
- The value 0xF means that there is no access to the operation, regardless of the key used

Response frame:

Header	C_DesCreateSTDataFile +1		OperationCode	CRC
--------	--------------------------	--	---------------	-----

2.6.13 CREATING BACKUP DATA FILES TYPE

Command frame:

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

Where:

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

Response frame:

header	C_DesCreateBACKDataFile +1		OperationCode	CRC
--------	----------------------------	--	---------------	-----

Access rights are defined in the same way as for Standard Data Files.

Saving of the Backup Data file type must end with the issuance of the C_DesCommit command.

2.6.14 CREATING LINEAR/CYCLIC RECORD FILES TYPE

Command frame:

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

Where:

Parameter name	Parameter description	Value range
C_DesCreateRecordFile	Create command for Record File type	0x6c
FileNo	File ID	0..0x0F
ComSett	Transmission type: 0x01 – unencrypted 0x03 – DES encrypted	0x00,0x03
AccRight1..2	Access rights to the file	0x00..0xff
RecSize1..3	3-byte record size in bytes, in the order of LSB..MSB	0x00-0xff
RecNumb1..3	3-byte parameter specifying the number of records, order of LSB..MSB	
Cy/Li	0x0c- cyclical type 0x0l – linear type	0x0C,0x01

Response frame:

header	C_DesCreateRecordFile+1		OperationCode	CRC
--------	-------------------------	--	---------------	-----

Access rights are defined in the same way as for Standard Data Files.

2.6.15 DELETING FILE

Command frame:

header	C_DesDeleteFile (0x6e)	FileNo		CRC
--------	------------------------	--------	--	-----

Where:

Parameter name	Parameter description	Value range
C_DesDeleteFile	Delete file command	0x6e
FileNo	File ID	0x00..0x0F

Response frame:

header	C_DesDeleteFile+1		OperationCode	CRC
--------	-------------------	--	---------------	-----

Change file settings

Command frame:

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

Where:

Parameter name	Parameter description	Value range
C_DesChangeFileSett	Command to change the properties of the file	0x80
FileNo	File ID	0..0x0F
ComSett	Transmission type: 0x01 – unencrypted 0x03 – DES encrypted	0x00,0x03
AccRight1..2	Access rights to the file	0x00..0xff

Response frame:

Header	C_DesChangeFileSett+1		OperationCode	CRC
--------	-----------------------	--	---------------	-----

Access rights are defined in the same way as for Standard Data Files.

2.6.16 READING-OUT DATA FROM STD/BACK DATA FILE TYPE

Command frame:

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

Where:

Parameter name	Parameter description	Value range
C_DesReadData	Read-out command from a file	0x82
FileNo	File ID	0..0x0F
Offset1..3	3-byte parameter specifying the place from which we start to read the file, order of LSB..MSB	0x00-0xFF
Length1..3	3-byte parameter specifying the number of bytes to be read, order of LSB..MSB (up to 58 bytes can be read at once)	0x00-0x3A

Response frame:

header	C_DesReadData +1	n Bytes	OperationCode	CRC
--------	------------------	---------	---------------	-----

2.6.17 WRITING DATA TO STD/BACK DATA FILE TYPE

Command frame:

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

Where:

Parameter name	Opis parametru	Zakres wartości
C_DesWriteData	File write command	0x84
FileNo	File ID	0..0x0F
Offset1..3	3-byte parameter specifying the place from which we start to write, order of LSB..MSB	0x00-0xFF
Data1..58	Data that we intend to save to a file, (one time you can save up to 58Byte)	0x00-0xFF

Response frame:

header	C_DesWriteData+1		OperationCode	CRC
--------	------------------	--	---------------	-----

2.6.18 WRITING RECORD TO RECORD DATA FILE TYPE

Command frame:

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

Where:

Parameter name	Parameter description	Value range
C_DesWriteRecord	Write record command	0x86
FileNo	File ID	0..0x0F
Offset1..3	3-byte parameter specifying the place from which we start to write, order of LSB..MSB (this value must be smaller than the size of a single record)	0x00-0xFF
Data1..58	Data that we intend to save to a file, (one time you can save up to 58 bytes, the sum of this value and the offset must be smaller than the size of a single record)	0x00-0xFF

Response frame:

header	C_DesWriteRecord+1		OperationCode	CRC
--------	--------------------	--	---------------	-----

Note:

Writing a record to a Record File type must end with the issuance of the C_DesCommit command.

2.6.19 READING-OUT A RECORD FROM RECORD DATA FILE TYPE

Command frame:

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

Where:

Parameter name	Parameter description	Value range
C_DesReadRecord	Record read-out command	0x88
FileNo	File ID	0..0x0F
WhichRecord1..3	3-byte parameter specifying the record from which we start to read, order of LSB..MSB	0x00-0xFF
NoOfRecords1..3	3-byte parameter specifying the number of records to read-out, order of LSB..MSB	0x00-0xFF

Response frame:

header	C_DesReadRecord +1	Record data...	OperationCode	CRC
--------	--------------------	----------------	---------------	-----

The number of read-out data can not be more than 58 bytes, so keep the rule: {NoOfRecords1..3} * size_crumb <58bytes

2.6.20 CLEARING RECORD DATA FILE TYPE

Command frame:

header	C_DesClearRecordFile (0x8a)	FileNo	CRC
--------	-----------------------------	--------	-----

Where:

Parameter name	Parameter description	Value range
C_DesClearRecordFile	Clearing command of the record file	0x8a
FileNo	File ID	0..0x0F

Response frame:

header	C_DesClearRecordFile+1		OperationCode	CRC
--------	------------------------	--	---------------	-----

Note:

This operation must end with the issuance of the C_DesCommit command.

2.6.21 CONFIRMATION COMMAND - *DESCOMMIT*

Command frame:

header	C_DesCommit (0x8c)		CRC
--------	--------------------	--	-----

Where:

Parameter name	Parameter description	Value range
C_DesCommit	Confirmation command	0x8c

Response frame:

header	C_DesCommit+1		OperationCode	CRC
--------	---------------	--	---------------	-----

2.6.22 TRANSPONDER DESELECTION

Command frame:

header	C_DesDeselect (0x8e)		CRC
--------	----------------------	--	-----

Where:

Parameter name	Parameter description	Value range
C_DesDeselect	Komenda de-selekcjonująca transponder	0x8e

Response frame:

header	C_DesDeselect+1		OperationCode	CRC
--------	-----------------	--	---------------	-----

2.7 I-BLOCK TRANSMISSION COMMAND ACCORDING TO ISO14443-4 PROTOCOL.

This command enables exchange plain data using I-BLOCK structure according to ISO14443-4. This functionality provide operating many transponders types like SMART MX, JCOP. Before, it is necessary to enable ISO1444-3 mode using C_Init_ISO14443-4 command.

Command frame:

header	C_TransclBlock	data	CRC
--------	----------------	------	-----

Where:

Parameter name	Parameter description	Value range
C_TransclBlock		0xC8
data	I-Block plain data	any

Response frame:

header	C_TransclBlock+1	Data from transponder	OperationCode	CRC
--------	------------------	-----------------------	---------------	-----

2.8 MIFARE APPLICATION DIRECTORY - MAD

2.8.1 FORMATTING THE MAD CARD

Command:

header	C_FormatMad	Type, Infobyte	CRC
--------	-------------	----------------	-----

Where:

Parameter name	Parameter description	Value range
C_FormatMad 0xa8	Formatting to MAD	0xa8
Type	1 - MAD1 (15sectors) 2 – MAD2 (30sectors)	0x01,0x02
Infobyte	Indicator on the issuer sector (default 0x00)	0x00-0x1F

Response frame:

header	C_FormatMad+1		OperationCode	CRC
--------	---------------	--	---------------	-----

Note:

Before executing the C_FormatMad command:

- disable the AutoReader mode (via the C_SetAutoReaderConfig command)
- load keys (default 0xff, 0xff, 0xff, 0xff, 0xff, 0xff)
- turn on antenna power (via C_TurnOnAntennaPower command)
- select the card (using the C_Select command)
- log in to sector 0 using an AA type key

2.8.2 ADDING THE APPLICATION TO THE MAD DIRECTORY

Command frame:

header	C_AddApplication	LSB, MSB, Sector	CRC
--------	------------------	------------------	-----

Where:

Parameter name	Parameter description	Value range
C_AddApplication 0xaa	Adding application	0xaa
LSB	less significant byte of the application number	0x00 - 0xFF
MSB	more significant byte of the application number	0x00 - 0xFF
Sector	Sector number, where the application should be located	0x01-0x0F :MAD1 0x01-0x1F :MAD2

Response frame:

header	C_AddApplication+1		OperationCode	CRC
--------	--------------------	--	---------------	-----

Note:

Application number must be different from 0x0000

Before executing the C_AddApplication command:

- disable the AutoReader mode (via the C_SetAutoReaderConfig command)
- load keys (default 0xff, 0xff, 0xff, 0xff, 0xff, 0xff)
- turn on antenna power (via C_TurnOnAntennaPower command)
- select the card (using the C_Select command)
- log in to sector 0 using an AA type key

2.8.3 SEARCH SECTOR FOR A GIVEN APPLICATION

Command frame:

header	C_GetSectorMad	LSB, MSB	CRC
--------	----------------	----------	-----

Where:

Parameter name	Parameter description	Value range
C_GetSectorMad 0xac	Search sector	0xac
LSB	less significant byte of the application number	0x00 - 0xFF
MSB	more significant byte of the application number	0x00 - 0xFF

Response frame:

header	C_GetSectorMad+1	Sector	OperationCode	CRC
--------	------------------	--------	---------------	-----

Note:

Before executing the C_GetSectorMad command:

- disable the AutoReader mode (via the C_SetAutoReaderConfig command)
- load keys (default 0xff, 0xff, 0xff, 0xff, 0xff, 0xff)
- turn on antenna power (via C_TurnOnAntennaPower command)
- select the card (using the C_Select command)
- log in to sector 0 using an AA type key

If the response byte is 0x00, then the application is not in the MAD directory.

2.8.4 SEARCHING FOR THE NEXT APPLICATION SECTOR

Command frame:

header	C_GetSectorMadNext	LSB, MSB	CRC
--------	--------------------	----------	-----

Where:

Parameter name	Parameter description	Value range
C_GetSectorMad	Searching for another sector	0xae

Response frame:

header	C_GetSectorMadNext+1	Sector	OperationCode	CRC
--------	----------------------	--------	---------------	-----

Note:

Before you can execute the C_GetSectorMadNext command, perform a search of the sector with the C_GetSectorMad command whose search result was different from 0

If the response byte is 0x00, then no more sectors were found for the application.

2.9 ELECTRICAL INPUTS AND OUTPUTS

2.9.1 DESCRIBING THE 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.	0x00...0x03
State	Desired output state	0x00 or 0x01

Response frame:

Header	C_WriteOutputs +1	OperationCode	CRC
--------	-------------------	---------------	-----

2.9.2 WRITING THE CONFIGURATION OF ANY PORT

Command frame:

header	C_SetIOConfig	IONo, IOConfigData1...n	CRC
--------	---------------	-------------------------	-----

If Configuring the port as an output, the parameters IOConfigData1 ... n have the form:

Dir, ConfByte1, TypeOfOutput, Maintain, 0Time, 1Time

Where:

Parameter name	Parameter description	Value range
C_SetIOConfig	Writing the configuration of any port	0x50
IONo	IO port number to be configured	0x00...0x03
Dir	Port direction	0x00 – wyjście
ConfByte1	one byte in which the youngest bit defines the output type as Normally Open or Normally Closed. The next bit determines how the output responds as a response to a change in arousal (responding to a flank) or responsive to a state of arousal (responsive to the state).	ConfByte1.BIT0 0-Normally Closed 1-Normally Open ConfByte1.BIT1 0-responding to a level 1-responding to a flank
TypeOfOutput	Source of the control signal	0x00 – permanently disabled 0x01 – permanently attached 0x02 – controlled via a USB serial interface

		0x03 - controlled via a serial USB interface that automatically returns to zero
Maintain	<p>Time of maintaining the switching state after the activation stops. This time is expressed as:</p> <p>Maintain x 100ms</p> <p>During the "Hold up" time, you can configure the output that can generate a square wave. Time of one and the zero time is set by the following parameters:</p>	
0Time	Logical „0” time	
1Time	Logical „1” time	

If we configure port as input, the parameters IOConfigData1 ... n have the form: Dir, Triger, TypeOfInput, Delay,

Where:

Parameter name	Parameter description	Value range
C_SetIOConfig	Writing the configuration of any port	0x50
IONo	IO port number to be configured	0x00...0x01,0x07
Dir	Port direction	0x01 – wejście
TypeOfInput	Input type	0x03
Delay	Delay	0x00

PAC_PUx reader can't switch the direction of ports. For correct configuration, the correct direction should be given for the given port.

List of existing ports that can be controlled in the PAC-PU		
Port number	direction	Description
0	output	LED RED1
1	output	LED RED2
3	output	BUZZER

Response frame:

header	C_SetIOConfig +1		OperationCode	CRC
--------	------------------	--	---------------	-----

2.9.3 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...0x07

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.	

2.10 ACCESS PASSWORD

2.10.1 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
--------	----------------	---------------	-----

2.10.2 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 Data1=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
--------	---------------------	---------------	-----

2.10.3 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
--------	-----------------	---------------	-----

2.11 AUTOREADER FUNCTION

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 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.

2.11.1 WRITE AN AUTOREADER CONFIGURATION

Command frame:

Header	C_SetAutoReaderConfig	ATrig, AMode, AOfflineTime, ASerial, Abuzz	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 communication interface 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 communication interface 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_CopyBlock C_WritePage4B C_ReadPage16B C_IncrementValue C_DecrementValue C_WriteValue C_ReadValue C_Halt	0x00...0xff																				
ASerial	Automatic sending of the UID transponder number after automatic reading from the transponder	0-never 1-only for the first application of the transponder 2-sends all																				
AMode	Selecting the format of the number to be sent 8 bits: MSB <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>R</td><td>L</td><td>H</td><td>C</td><td>M</td><td>E</td><td>I</td><td>A</td></tr></table> LSB	R	L	H	C	M	E	I	A	<table border="1" style="width: 100%;"> <tr> <td>R</td> <td>Reserved, always 0</td> </tr> <tr> <td>C=1 L=0</td> <td>number ends with the end of the number ends with the end of the line CR + LP</td> </tr> <tr> <td>C=1 L=1</td> <td>The number ends with the end of the LF line</td> </tr> <tr> <td>C=0 L=1</td> <td>The number ends with the end of the CR line</td> </tr> <tr> <td>M=1</td> <td>number starts with the "M" sign</td> </tr> <tr> <td>E=1</td> <td>extended information about the number of cards in the field and the type of card</td> </tr> </table>	R	Reserved, always 0	C=1 L=0	number ends with the end of the number ends with the end of the line CR + LP	C=1 L=1	The number ends with the end of the LF line	C=0 L=1	The number ends with the end of the CR line	M=1	number starts with the "M" sign	E=1	extended information about the number of cards in the field and the type of card
R	L	H	C	M	E	I	A															
R	Reserved, always 0																					
C=1 L=0	number ends with the end of the number ends with the end of the line CR + LP																					
C=1 L=1	The number ends with the end of the LF line																					
C=0 L=1	The number ends with the end of the CR line																					
M=1	number starts with the "M" sign																					
E=1	extended information about the number of cards in the field and the type of card																					

		I=1	Number in reverse order
		A=1 H=0	Number sent in the format ASCII
		A=0 H=0	Number sent in the Netronix frame format
		A=0 H=1	Number sent in HEX format
ABuzz	Automatic signaling by means of a buzzer after automatic reading of the UID from the transponder.	0-never 1-only for the first application of the transponder 2-signals all	

Response frame:

header	C_SetAutoReaderConfig +1		OperationCode	CRC
--------	--------------------------	--	---------------	-----

2.11.2 READ THE AUTOADARA CONFIGURATION

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.

2.12 SUPPORT FOR ID STORED IN TRANSPONDER MEMORY

2.12.1 WRITE/READ CONFIGURATION

Command frame:

Header	C_ConfigUserID	[CardType, ID_Len, ID_Offset, SecNo, KeyType, SKBKeyNo]		CRC
--------	----------------	---	--	-----

Where:

Parameter name	Parameter description	Value range
C_X_ConfigUserID	Command code	0x5C
CardType	Typ karty	0x50 – Mifare S50
ID_Len	Długość zapisanego ID	0x01...0x20
ID_Offset		0x20-ID_Len
SecNo	Number of the sector in which the ID is stored	
KeyType	Typ klucza	0xAA – Key A 0xBB – Key B
SKBKeyNo	The key number in the memory of the static keys that is used to log into the sector.	0-0x1F

Response frame:

	C_X_ConfigUserID+1	CardType, ID_Len, ID_Offset, SecNo, KeyType, SKBKeyNo	KodOperacji	
--	--------------------	---	-------------	--

The meaning of the fields the same as above.

2.13 CONFIGURING THE USB INTERFACE

2.13.1 WRITING THE CONFIGURATION OF SERIAL INTERFACE

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 USB bus	0x01...0xfe
Baudrate	Data baud rate on USB 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
-------------------------	--	---------------

2.13.2 READING THE CONFIGURATION OF SERIAL INTERFACE

Command:

C_GetInterfaceConfig	
----------------------	--

Where:

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

Response:

C_GetInterfaceConfig +1	Mode, Adr, Baudrate	OperationCode
-------------------------	---------------------	---------------

Where:

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

2.14 OTHER COMMANDS

2.14.1 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		OperationCode	CRC
--------	------------	--	---------------	-----

2.14.2 TURNING ON/OFF KEYBOARD EMULATION FUNCTION

Command frame:

Header	C_Keyboard	[Param]	CRC
--------	------------	---------	-----

Where:

Parameter name	Parameter description	Value range
C_Keyboard	Read-out of reader software version	0x04

[Param]	One byte:							MSB	LSB
	0	0	F2	F1	F0	INV	ENTER		

F2	F1	F0	Format
1	0	0	ASCII
0	1	1	1WIRE frame format. ID justified to the right
0	1	0	1WIRE frame format. ID justified to the left
0	0	1	Hexadecimal
0	0	0	Decimal

INV=1 – reverse order

ENTER = 1 – simulation of pressing the ENTER key after entering the ID

EN = 1 – enabled keyboard

Response frame:

header	C_Keyboard+1	Param	OperationCode	CRC
--------	--------------	-------	---------------	-----

Where:

Param - the meaning of response parameters is the same as described before.

2.14.3 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 a string of bytes specifying the firmware name.

2.15 MEANING OF OPERATION CODE IN RESPONSE FRAME

Name of operation code	Description	Value
OC_Error	error	0x00
OC_ParityError	Parity error	0x01
OC_RangeError	Range parameter error	0x02
OC_LengthError	Data lenght error	0x03
OC_ParameterError	Parameter error	0x04
OC_Busy	Momentary occupancy of internal modules	0x05
OC_BadCRC	Communication error with the card	0x20
OC_CommandUnknown	Unknown command	0x07
OC_WrongPassword	Wrong password or the last password has expired, i.e. an automatic Log-Out has taken place.	0x09
OC_NoCard	No transponder	0x0a
OC_BadFormat	Wrong data format	0x18
OC_FrameError	Transmission error. It may be indicative of existing interference	0x19
OC_NoAnswer	No response from transponder	0x1E
OC_TimeOut	Operation time exceeded. It may indicate a lack of a transponder in the field of the reader	0x16
OC_Successful	Operation completed correctly	0xff
Operation codes related to DESFIRE transponders		
OC_DesNoChanges	Commit operation did not bring any changes	0x0c
OC_DesOutOfEeprom	Lack of eeprom memory	0x0e
OC_DesIllegalCommand	Illegal command	0x1c
OC_DesIntegrityError	CRC error / transmission with the card	0x1e
OC_DesNoSuchKey	Invalid key number	0x40
OC_DesLengthError	Invalid command lenght	0x7e
OC_DesPermissionDenied	No permission to perform the operation	0x9d
OC_DesParameterError	Command parameter error	0x9e
OC_DesApplNotFound	No application for selected Aid	0xa0
OC_DesApplIntegrError	Application error, application is blocked	0xa1
OC_DesAuthError	Authorization error/incorrect key	0xae
OC_DesBoundaryError	Writing/reading-out from the record went beyond the size	0xbe
OC_DesPICCIntegError	Internal transponder error is blocked	0xc1
OC_DesCountError	28 applications limit has been exceeded	0xce
OC_DesDuplicateError	Application / File with this identifier already exists	0xde
OC_DesEepromError	Error during reading-out/writing to EEPROM memory	0xee
OC_DesFileNotFound	File with this ID does not exist	0xf0
OC_DesFileIntegrError	Irreversible file error, the file is blocked	0xf1

3 KEYBOARD EMULATION

The device PAC-DUX can emulate USB keyboard (HID). During emulating the keyboard, each reading of the ID transponder through the mechanism AutoReader'a will simulate entering his ID.

3.1 ID FORMATS

Configuration byte form for the *C_Keyboard* command:

MSB				LSB			
0	0	F2	F1	F0	INV	ENTER	EN

The meaning of bits::

- Bit **EN** – if set, the keyboard emulation is enabled.
- Bit **ENTER** – if set, then after simulation of entering the ID, the ENTER key is simulated
- Bit **INV** – if set, the ID order is inverted before further formatting.
- Bits **F<0-2>** – specify the output format.

Examples of formatting:

Bit ENTER=0, bit INV=0

F2	F1	F0	Format	Read ID	Sent ID
1	0	0	ASCII	“Tekst0123”	TEKST0132 ⁽¹⁾
0	1	1	1WIRE frame format. ID justified to the right	3BFBB064	9700003BFBB06401
0	1	0	1WIRE frame format. ID justified to the left	3BFBB064	3B3BFBB064000001
0	0	1	Hexadecimal	1234ABCD	1234ABCD
0	0	0	Decimal	1234ABCD	305441741

⁽¹⁾ – Lowercase letters are converted to uppercase. Allowed characters: A-Z and numbers 0-9.

4 RESET TO DEFAULT SETTINGS

To reset the device to factory defaults, push for ca. 5 seconds button on bottom panel.

During reset to factory defaults, following parameters of reader are fixed:

Parameter name or functionality	Value or setting
Access password	No password
Port 0 – LED1	card reading signal, serial port control
Port 1 – LED2	Controlling the serial port
Port 3 – BUZZER	card reading signal, serial port control
Autoreader	activated
Keyboard	disactivated

5 EXAMPLES OF WORKING WITH TRANSPONDERS

5.1 OPERATION EXAMPLE OF S50, S70 TRANSPONDER

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 keys.

Because during manual experiments, time between subsequent commands sent via serial interface is large and reaches values from some seconds 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, 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 read-out from the block.

5.2 EXAMPLE OF WORKING WITH DESFIRE TRANSPONDERS

After correctly connecting the reader and establishing mutual communication between it and the host computer, it is possible to proceed with the read and write operation of the transponder's memory.

The following operations assume that the reader has factory settings and that the Desfire card used has factory settings, ie full access rights, and the PICC Master key has the value 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00.

The result of this example is to create a new application, change the standard application key, create a data file, save and then read the data from the file.

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:

1. SetAutoReaderConfig 0x00, 0x00, 0x00, 0x00.

To read the transponder, first load the keys to the key memory.

So we load the standard desfire transponder key into the "3" position of the reader's memory, and for position 4 we load our own key, which we will give the new application:

2. C_DesSaveKey 0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00

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

Enclose field.

4. C_TurnOnAntennaPower 0x01

We apply the transponder to the reader, we select the transponder.

5. C_Select 0x00

We initiate the ISO data exchange protocol with the transponder logical number 0

6. C_DesInitProtocol 0x00

We authorize using the key "0" or PICC Master key, this key is stored in the reader's memory under the index "3"

7. C_DesAuth 0x00, 0x03, 0x0A

We create an application with an identification number, e.g. 0x30, 0x10, 0x55, with the default settings of the ApplicationMasterKey key, with reservation for 4 keys

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

We change the default, newly created ApplicationMasterKey key to the one we have saved in the reader in position 4. Therefore, we select a new application:

9. C_DesSelectApp 0x30, 0x10, 0x55

We log in to the application using the Application Master Key, then change it and then log in again using the new key

10. C_DesAuth 0x00, 0x03, 0x0A

11. C_DesChangeKey 0x00, 0x04

12. C_DesAuth 0x00, 0x04, 0x0A

We create a standard data file with full access rights for the Application Master Key and read rights for the key "3". The file will have the index "2", the unencrypted data exchange and the size of 1500 bytes

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

We now save data to the file just created from position 0

14. C_DesWriteData 0x02,0x00,0x00,0x00, \$TuSaNaszeDaneDoZapisu

We read 21 bytes of just saved data

14. C_DesReadData 0x02,0x00,0x00,0x00, 0x15,0x00,0x00

5.3 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

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 again

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

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for [Netronix](#) manufacturer:

Other Similar products are found below :

[ACTS-2](#) [AGEDI-B](#) [ANTENNA L112 50 MM](#) [ANTENNA L113 30 MM](#) [COTER-E4I](#) [COTER-ECI](#) [CTU-D2R](#) [CTU-D5N](#) [CTU-D5R](#) [CTU-M5RM](#) [CTU-R2RM](#) [CTU-R5RL](#) [CTU-R5RM](#) [MICRODEM3](#) [MM-R5](#) [MW-D7G](#) [MW-R4B](#) [MW-R7G](#) [MW-UDB](#) [NANO-MS](#) [NANO-US](#) [NBL-2](#) [NBL-3](#) [PAC-DUG](#) [PAC-PUB](#) [PLA-D6L](#) [PLA-MDK](#) [PLA-R6L](#) [PLA-UDK](#) [UW-DAL](#) [UW-RES](#) [UW-U4A](#) [UW-U4G](#)