Technical Data Sheet

RFID reader

PAC-MUG
PAC-MUB

PAC-MUG

ATTENTION! THIS CONFIDENTIAL DOCUMENT IS PROPERTY OF NETRONIX SP. Z O.O.

DISTRIBUTION OF THIS DOCUMENT IN ANY WAY WITHOUT SPECIAL PERMISSION OF ITS OWNER IS STRICTLY FORBIDDEN

1 Introduction

PAC-MUx is RFID desktop card reader for Mifare family.

It features following functionality:

- Supported transponders: Mifare S50, Mifare S70, Mifare Ultra Light, Mifare DesFire
- USB interface
- Internal buzzer
- Built-in two LED's of common purpose and diode as a supply indicator
- Built-in switch for reset to factory settings
- Reading the two-state input
- 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

2 General Specifications

Supported functionality depending on transponder / card type:		
Transponder type	**ID number read-out**	**Full write and read-out of memory blocks**
S50	YES	YES
S70	YES	YES
Ultra Light	YES	YES
DesFire	YES	NO

Pac-MUx 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	2400, 4800, 9600, 19200, 38400, 57600, 115200 bps, 8 data bits, 1 stop bit, no parity compliant with „Netronix Protocol"
Temperature	0-60st.c

3 Serial transmission format

After FTDI drivers installation, PAC-MUx 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".

3.1 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.

3.1.1 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

3.1.2 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

3.2 Commands for communication with transponder

3.2.1 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

3.2.2 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

3.2.3 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

3.2.4 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 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

3.2.5 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	

3.2.6 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

3.2.7 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

3.2.8 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

3.2.9 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

3.2.10 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

3.2.11 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

3.2.12 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

3.2.13 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

3.2.14 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

3.3 Electrical inputs and outputs

3.3.1 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. It should be set as an output.	0x02…0x06
State	Desired output state	0x00 lub 0x01

Response frame:

Header	C_WriteOutputs +1		OperationCode	CRC

3.3.2 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

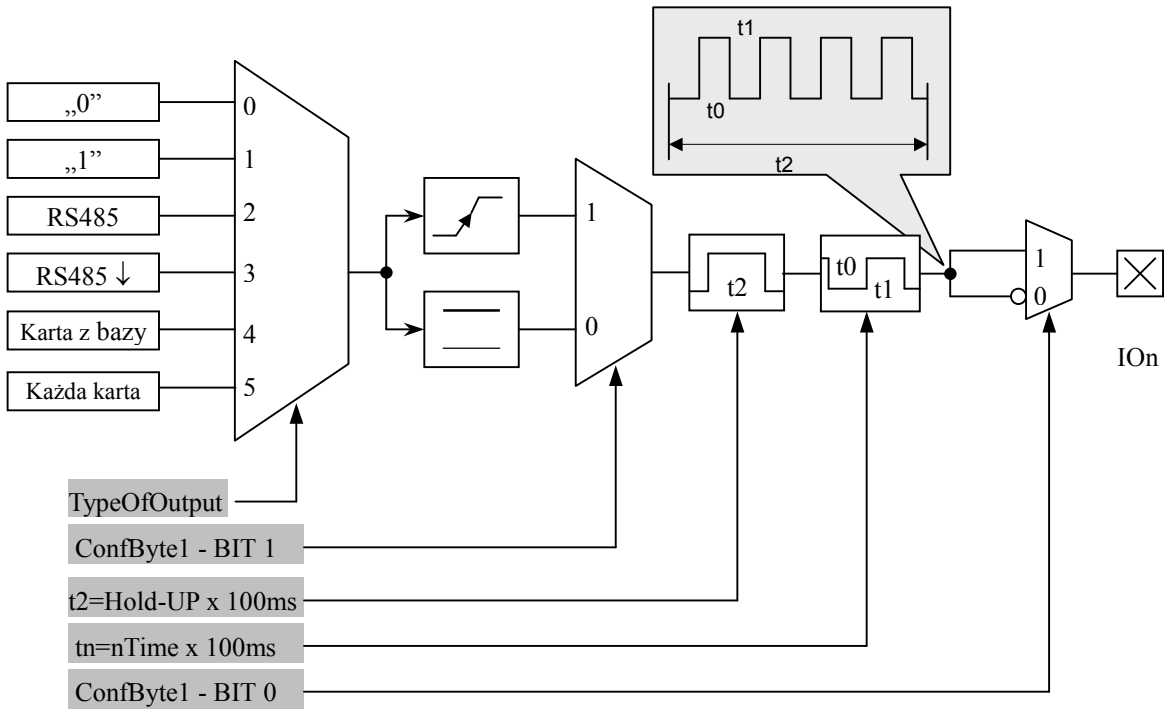
Response frame:

Header	C_ReadInputs +1	State	OperationCode	CRC

Where:

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

3.3.3 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, NormalOpen+ChangeState, 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
NormalOpen+ChangeState	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).	Bit NormalOpen B.0 0-Normally Closed 1-Normally Open Bit ChangeState B.0 0-level responding 1-slope responding

TypeOfOutput	Source of driving signal	0x00 – permanently off 0x01 – permanently on 0x02 – driven via serial interface 0x03 – driven via serial with automatic reset 0x04 – driven by internal access control mechanism ACM. This output is driven in case of applying the card to reader, which is written into internal card base. 0x05 – set in case of applying freely selected card to reader.
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

PAC-MU reader has no possibility to toggle port direction.
To accomplish proper configuration, input proper direction option to given port.

LIST OF EXISTING PORTS, WHICH CAN BE DRIVEN IN PAC-MU		
Port number	Direction	Description
0	Output	Red led 1
1	Output	Red led 2
2	output	BUZZER

Ramka odpowiedzi:

nagłówek	C_SetIOConfig +1		KodOperacji	CRC

Response frame:

Header	C_SetIOConfig +1		OperationCode	CRC

3.3.4 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.	

3.4 Access password

3.4.1 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

3.4.2 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 =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

3.4.3 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

3.5 Autoreader function

3.5.1 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, 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 RS485 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	0x00…0xff

	C_Select	
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

Response frame:

Header	C_ SetAutoReaderConfig +1		OperationCode	CRC

3.5.2 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.

3.6 Configuring the USB interface

3.6.1 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 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	

3.6.2 Reading the configuration of serial interface

Command:

	C_ GetInterfaceConfig		

Where:

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

Odpowiedź:

	C_ GetInterfaceConfig +1	Mode, Adr, Baudrate	OperationCode	

Where:

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

3.7 MAD – Mifare Application Directory

3.7.1 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

3.7.2 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

3.7.3 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.

3.7.4 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.

3.8 Other commands

3.8.1 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		KodOperacji	CRC

3.8.2 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.

3.9 Meaning of operation code in response frame

Operation code name	3.9.1.1.1.1.1 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

4 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 Clearing the card memory and resetting to factory defaults

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:

Name of parameter or its functionality	Value or setting
Address on serial bus	0x01
Baud rate on serial bus	9600 bps
Access password	0x31 32 33 34 00 means „1234” in dot transcription
Port 0 – LED1	Read card indicator
Port 1 – LED2	Read card indicator
Port 2 – BUZZER	Read card indicator

6 Operation example of 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 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.

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