

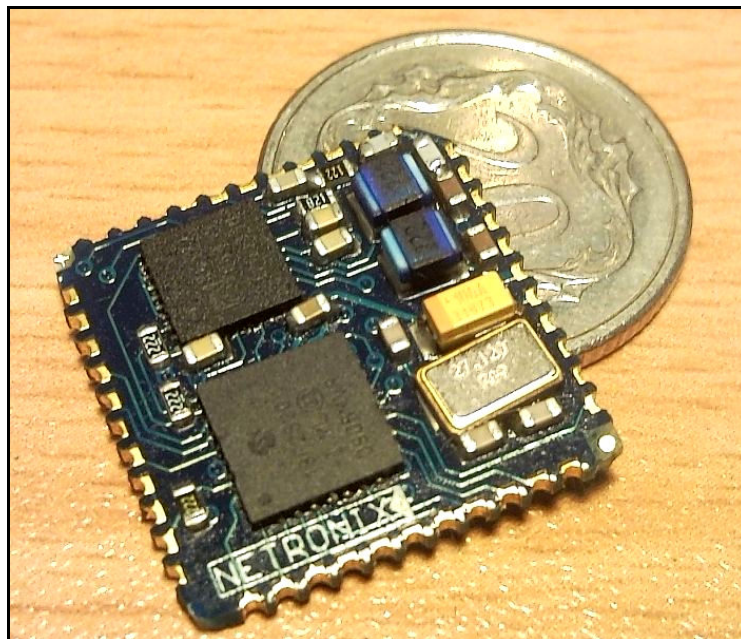


Technical Data Sheet

RFID reader

# NANO-MP NANO-MS

NANO-M-man-eng-v1



*NANO-MS*

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**

<b>1.</b>	<b>INTRODUCTION.....</b>	<b>5</b>
<b>2.</b>	<b>SPECIFICATIONS .....</b>	<b>6</b>
<b>3.</b>	<b>TERMINAL DESCRIPTION .....</b>	<b>7</b>
<b>4.</b>	<b>INTERFACE SELECTION, HARDWARE CONFIGURATION.....</b>	<b>8</b>
<b>4.1.</b>	<b>Configuring inputs of interface select.....</b>	<b>8</b>
<b>4.2.</b>	<b>Selecting external elements of antenna.....</b>	<b>8</b>
<b>4.3.</b>	<b>Application diagrams .....</b>	<b>8</b>
<b>5.</b>	<b>TRANSMISSION PROTOCOLS .....</b>	<b>10</b>
<b>5.1.</b>	<b>RS232 transmission protocol.....</b>	<b>10</b>
<b>5.2.</b>	<b>Protocol for I<sup>2</sup>C transmission .....</b>	<b>10</b>
5.2.1.	Data exchange algorithm .....	10
5.2.2.	I2C Timings.....	11
<b>5.3.</b>	<b>SPI transmission protocol.....</b>	<b>12</b>
5.3.1.	Data exchange algorithm .....	12
5.3.2.	SPI Timings .....	13
<b>5.4.</b>	<b>Protocol for 1WIRE (Dallas) bus.....</b>	<b>14</b>
<b>5.5.</b>	<b>Wiegand protocol.....</b>	<b>15</b>
<b>6.</b>	<b>COMMUNICATION PROTOCOL COMMANDS .....</b>	<b>17</b>
<b>6.1.</b>	<b>Commands for communication with transponders.....</b>	<b>17</b>
6.1.1.	Key management introduction.....	17
6.1.2.	Key loading into dynamic key memory.....	17
6.1.3.	Key loading to key static memory.....	18
<b>6.2.</b>	<b>Commands for communication with transponder.....</b>	<b>18</b>
6.2.1.	On/off switching of reader field .....	18
6.2.2.	Selecting one of many transponders .....	19
6.2.3.	Logging by means of Dynamic Key Buffer to selected sector of transponder.....	20
6.2.4.	Logging by means of Static Key Buffer to selected sector of transponder.....	20
6.2.5.	Reading-out the content of transponder block .....	21
6.2.6.	Writing the content of transponder block .....	21
6.2.7.	Copying the content of transponder block into other block.....	22

6.2.8.	Writing the page content into Mifare UL.....	22
6.2.9.	Reading the page content in Mifare UL.....	23
6.2.10.	Writing values to transponder block.....	23
6.2.11.	Reading-out the values from transponder block.....	24
6.2.12.	Increasing the value included in transponder block.....	24
6.2.13.	Decreasing the value included in block transponder.....	25
6.2.14.	Setting the transponder in field into sleep mode.....	25
<b>6.3.</b>	<b>MAD – Mifare Application Directory .....</b>	<b>26</b>
6.3.1.	Card MAD formatting.....	26
6.3.2.	Adding the application to MAD directory.....	26
6.3.3.	Pursuing the sector for given application.....	27
6.3.4.	Pursuing the next sector of application.....	27
<b>6.4.</b>	<b>Reader inputs and outputs.....</b>	<b>28</b>
6.4.1.	Writing the output state.....	28
6.4.2.	Reading the input state.....	28
6.4.3.	Writing the settings to any port.....	28
6.4.4.	Reading-out the configuration of freely selected port.....	30
<b>6.5.</b>	<b>Access password.....</b>	<b>32</b>
6.5.1.	Logging to reader.....	32
6.5.2.	Changing the password.....	32
6.5.3.	Logging out of the reader.....	33
6.5.4.	Writing the “automatic read” configuration.....	33
6.5.5.	Reading-out the configuration of automatic device.....	34
6.5.6.	Setting the date and time.....	35
6.5.7.	Reading-out the date and time.....	35
<b>6.6.</b>	<b>Configuring the UART serial interface .....</b>	<b>36</b>
6.6.1.	Writing the configuration of serial port.....	36
6.6.2.	Reading the configuration of serial interface.....	36
<b>6.7.</b>	<b>Other commands.....</b>	<b>36</b>
6.7.1.	Remote reset of reader.....	36
6.7.2.	Sleep mode.....	37
6.7.3.	Reading-out the reader software.....	37
<b>6.8.</b>	<b>Code meanings in response frames .....</b>	<b>38</b>
<b>7.</b>	<b>RESET TO DEFAULT SETTINGS.....</b>	<b>39</b>
<b>8.</b>	<b>FOOTPRINT PROPOSED FOR NANO MODULE.....</b>	<b>40</b>



## 1 . Introduction

NANO-M device is OEM miniature RFID card reader operating at frequency of 13,56 MHz.

Main features:

- Support of Mifare S50, Mifare S70, Mifare Ultra Light, Mifare DesFire\*, Mifare Plus\*\*
- RS232 (TTL) interface with RS485 bus transmitter/receiver control output,
- Addressability on bus in RS-485 mode,
- I<sup>2</sup>C interface,
- SPI interface,
- 1WIRE (Dallas DS1990) interface,
- Interface WIEGAND,
- Signal terminal for reset to factory defaults,
- Anticollision output for multiple readers operating in proximity,
- Low current consumption,
- standby mode,
- 5 configurable inputs/outputs,
- 2-state output control,
- Read-out of 2-state inputs,
- Data password protected,
- Small dimensions 17,5 x 17,5 x 3mm,
- Software update via RS-232 interface using *NEFIR* program.

Reader is available in two versions depending on assembly method:

- NANO-MS – version for surface mounting (SMD)
- NANO-MP – version for through-hole mounting (THT)

## 2 . Specifications

<b>Transponder operate frequency</b>	13,56 MHz
Supported transponder type	Mifare S50,S70,U.L,Desfire*,Plus**, SmartMX*
Approximate maximal communication range with transponders (using ø 50 mm antenna)	S50 – 6cm Plus – 4cm
Module supply voltage	3,3 V ±10%
Operating temperature	-20°C to +70°C
Current consumption: - in „autoreader” mode - during field switch-off - during field switch-on - in standby mode	25 mA 16,5 mA Max. 70mA – zależne od zastosowanej anteny Max. 1 µA
Supported two-way interfaces:	- RS232 3.3V version with terminal controlling RS485 driver - SPI - I <sup>2</sup> C
Supported one-way interfaces:	- 1WIRE (DS1990 pill emulation) - WIEGAND
Common purpose inputs/outputs	5 configurable inputs/outputs
Dimensions	17.5 x 17.5 x 3 mm

\* UID reading only

\*\* SL1 mode and UID reading only

### 3. Terminal description

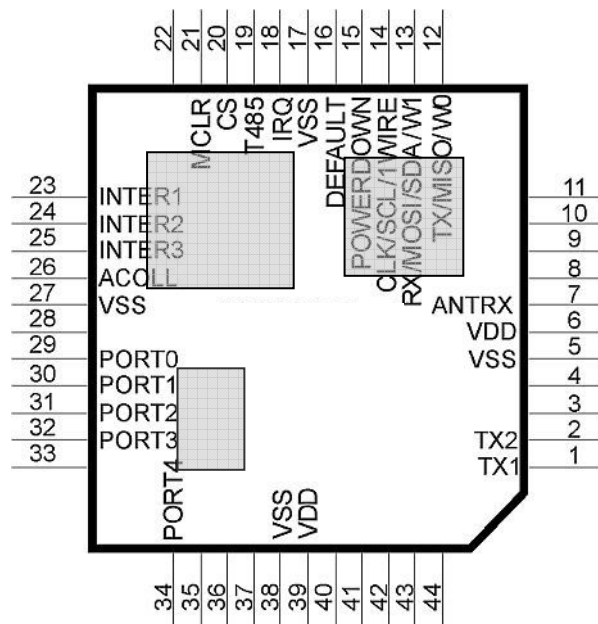


Fig. 1

No.	Label	Description
1	TX1	Antenna 1 output
2	TX2	Antenna 2 output
5	VSS	Ground of module supply
6	VDD	Plus of supply voltage of the module
7	ANTRX	Return signal from antenna
12	TX/MISO/W0	TX(data out) for RS232, MISO(data out) for SPI, '0' for Wiegand
13	RX/MOSI/SDA/W1	RX(data in) for RS232, MOSI(data in) for SPI, '1' for Wiegand
14	CLK/SCL/1WIRE	CLK signal for SPI bus, SCL signal for I <sup>2</sup> C bus, 1WIRE pin
15	/POWERDOWN	Applying logical zero makes the module go to standby mode. If module enters stand-by mode by means C_Sleep command, positive slope wakes the module up.
16	/DEFAULT	Applying logical zero for time 2 sec. or longer makes NANO module return to default settings
17	VSS	Ground of module supply
18	/IRQ	Interrupt output – currently not used
19	T485	Transmit/receive switching output for RS485 interface transceiver
20	/CS	Chip select input for SPI bus
21	/MLCR	Input of hardware reset of NANO module – currently not used
23	INTER1	Communication interface select, see diagrams below
24	INTER2	
25	INTER3	
26	/ACOLL	Field switch on synchronization input for multiple readers operating in proximity
27	VSS	Ground of module supply
28	PORT0	Input/output port of common purpose
29	PORT1	Input/output port of common purpose
30	PORT2	Input/output port of common purpose
31	PORT3	Input/output port of common purpose
34	PORT4	Input/output port of common purpose
38	VSS	Ground of module supply
39	VDD	Plus of supply voltage of the module

## 4 . Interface selection, hardware configuration

### 4 . 1 . Configuring inputs of interface select

INTER1	INTER2	INTER3	Interface type	Default settings of interface
1	1	1	RS232	9600 bps, 8, N, 1
1	1	0	I <sup>2</sup> C	Address 0xC0, fclk: 400 kHz
0	1	0	I <sup>2</sup> C	Address 0xC2, fclk: 400 kHz
1	0	1	SPI	Fclk: 400kHz
0	1	1	DALLAS	Adres: 0x01, family code: 0x01
0	0	1	WIEGAND	37 bits

“1” – connected to +3,3 V

„0” – connected to GND

### 4 . 2 . Selecting external elements of antenna

It is recommended that transmit/receive antenna inductance to be 0,5μH – 1,5μH. Inductance L with capacitor C used should form resonance circuit for 13,56 MHz frequency. Resistance R sets the antenna quality which should be 8 to 15. Be sure to use a capacitor with properly high voltage higher than 100 V.

### 4 . 3 . Application diagrams

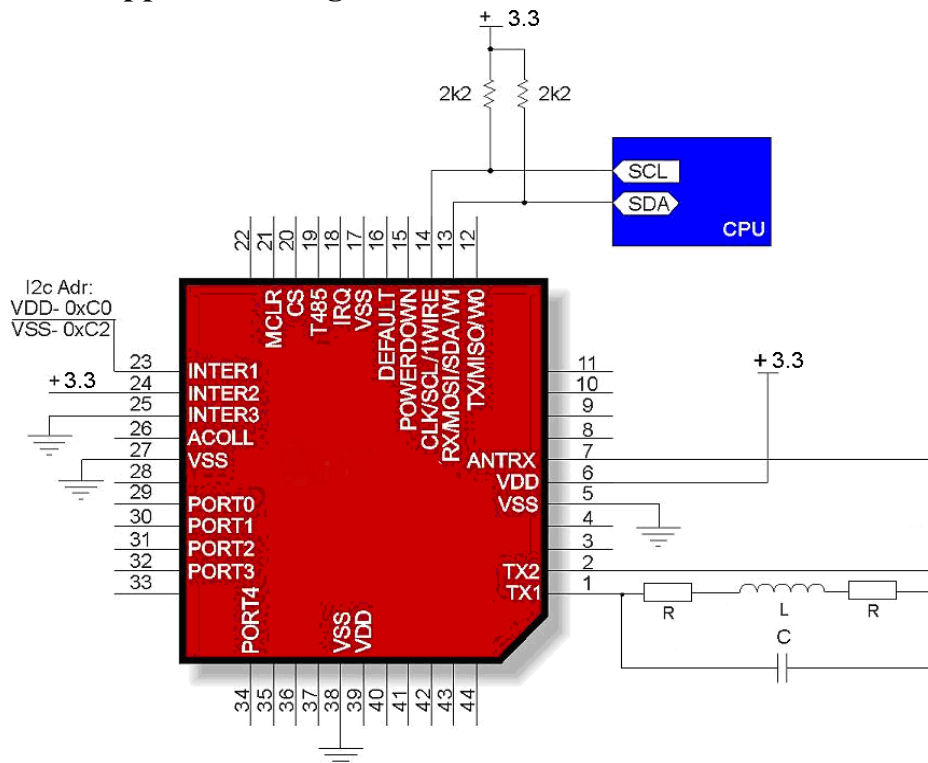


Fig. 2 Minimal configuration for I<sup>2</sup>C interface



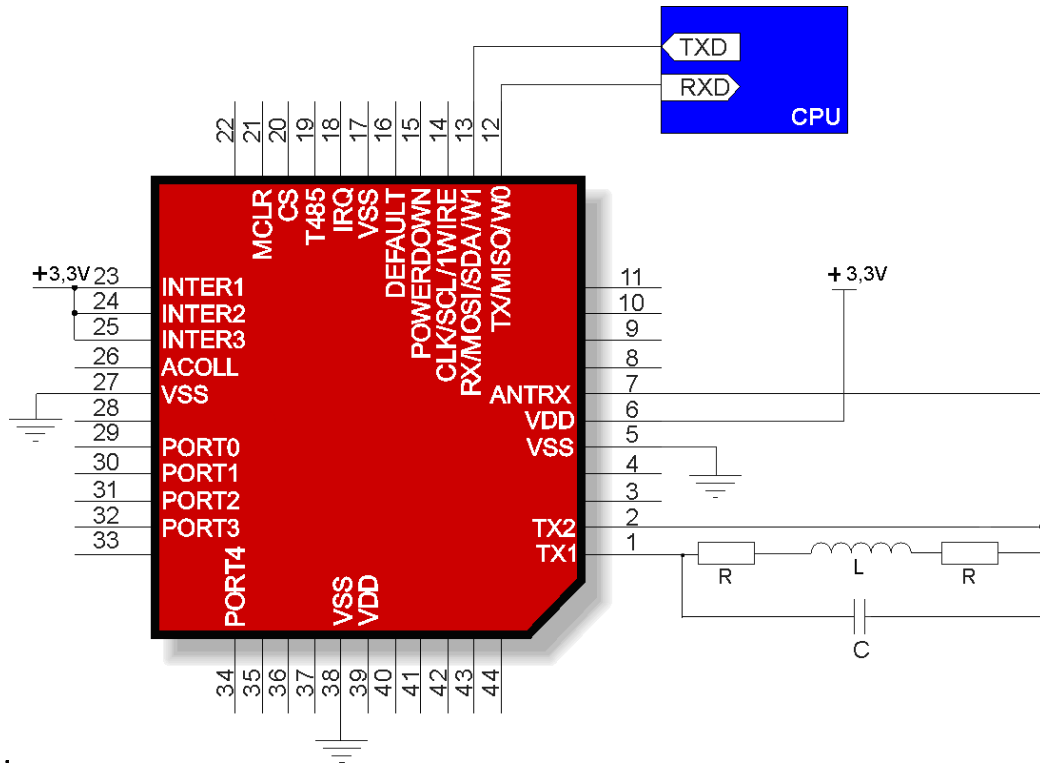


Fig. 3 Minimal configuration for RS232 interface

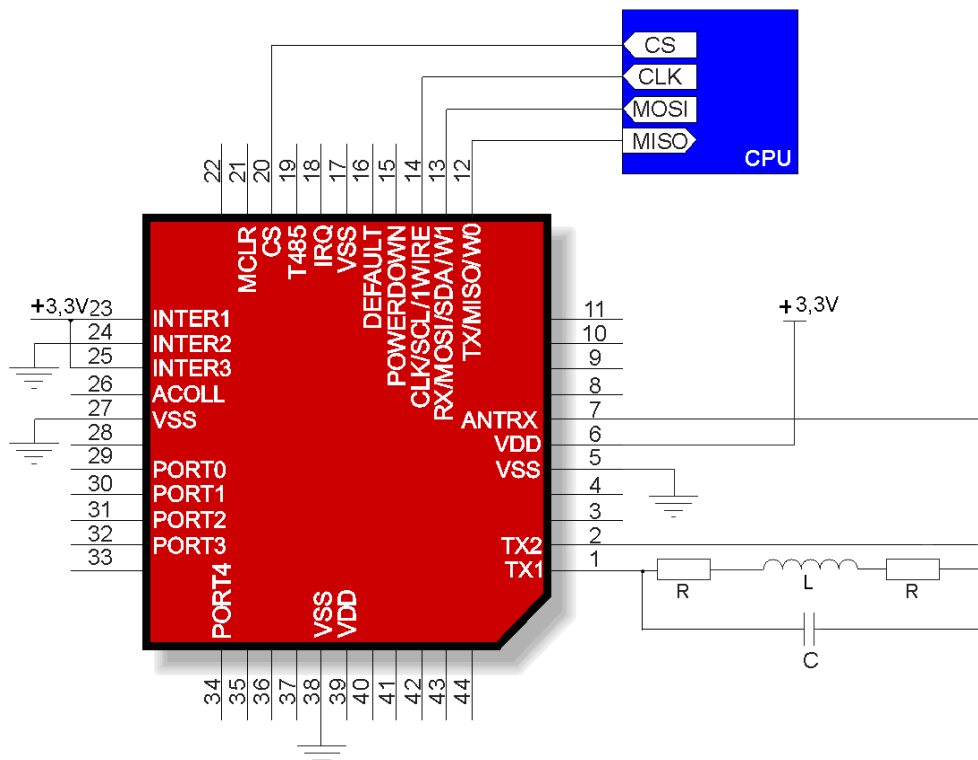


Fig. 4 Minimal configuration for SPI interface

## 5 . Transmission protocols

### 5 . 1 . RS232 transmission protocol

In this data sheet RS-232 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 Protocol” document.

Command frame:

Header	C_CommandName	Response_parameters1...n	CRC
--------	---------------	--------------------------	-----

Response frame:

Header	C_CommandName +1	Response_parametrs...m	OperationCode	CRC
--------	------------------	------------------------	---------------	-----

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

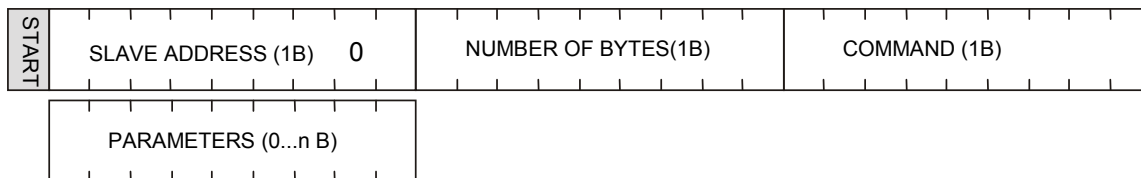
### 5 . 2 . Protocol for I<sup>2</sup>C transmission

#### 5 . 2 . 1 . Data exchange algorithm

A module configured depending on diagram showed on Fig. 2 operates in I<sup>2</sup>C interface mode in following sequences:

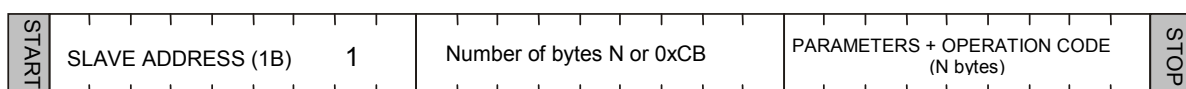
1. Master (external device) writes command with parameters if nessesery into slave device (NANO module)
- 2 The command is performed (immediatly after receiving byte sent quantity declared in frame)
3. Master device reads response, its parameters and operation code. In case of receiving busy byte 0xCB, repeat attempt to read the response after ca. 1 ms (commands connected with write to/and read from transponders can last up to 100 ms).

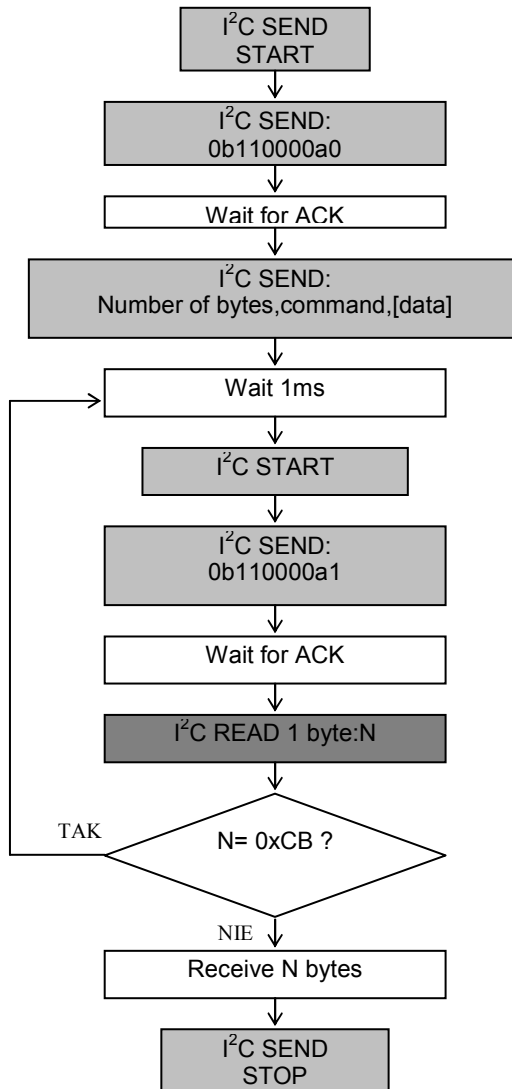
We write inquiry-command to NANO module:



The „number of bytes” field must contain information on byte quantity sent directly “command” fields and “parameters”.

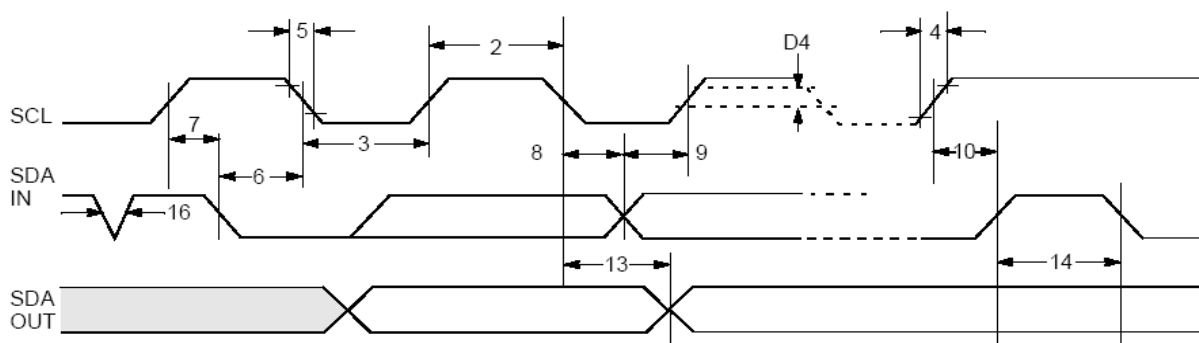
We have then:





**5.2.2. I2C Timings**

Module sends and receives data at 400 kHz clock frequency considering timings showed below.



Param. No.	Sym.	Characteristic	Min.	Max.	Units
1	FCLK	Clock Frequency	— —	400 100	kHz
2	THIGH	Clock High Time	600 4000	— —	ns
3	TLOW	Clock Low Time	1300 4700	— —	ns
4	TR	SDA and SCL Rise Time (Note 1)	— —	300 1000	ns
5	TF	SDA and SCL Fall Time	—	300	ns
6	THD:STA	Start Condition Hold Time	600 4000	— —	ns
7	TSU:STA	Start Condition Setup Time	600 4700	— —	ns
8	THD:DAT	Data Input Hold Time	0	—	ns
9	TSU:DAT	Data Input Setup Time	100 250	— —	ns
10	TSU:STO	Stop Condition Setup Time	600 4000	— —	ns
11	TSU:WP	WP Setup Time	600 4000	— —	ns
12	THD:WP	WP Hold Time	1300 4700	— —	ns
13	TAA	Output Valid from Clock (Note 2)	— —	900 3500	ns
14	TBUF	Bus free time: Time the bus must be free before a new transmission can start	1300 4700	— —	ns
15	TOF	Output Fall Time from $V_{IH}$ Minimum to $V_{IL}$ Maximum	20+0.1Cb —	250 250	ns

**Note 2:** Reader keeps in low state first clock pulse of each byte sent until proper state is placed on SDA line.

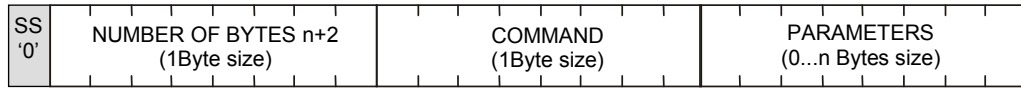
### 5.3. SPI transmission protocol

#### 5.3.1. Data exchange algorithm

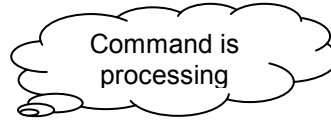
A module configured depending on diagram showed on Fig. 4 operates in SPI interface mode in following sequences:

1. SS pin goes low.
2. Master device sends a command with parameters to slave (NANO) device.
3. Commands is executing
4. NANO module is ready to response, when MISO line is on LOW level. MISO level test should be perform at least after 200us from end of command sending.
5. Master reads data and operation code from NANO,
6. SS pin goes high.

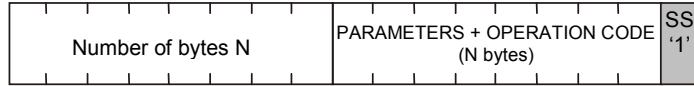
master->slave direction



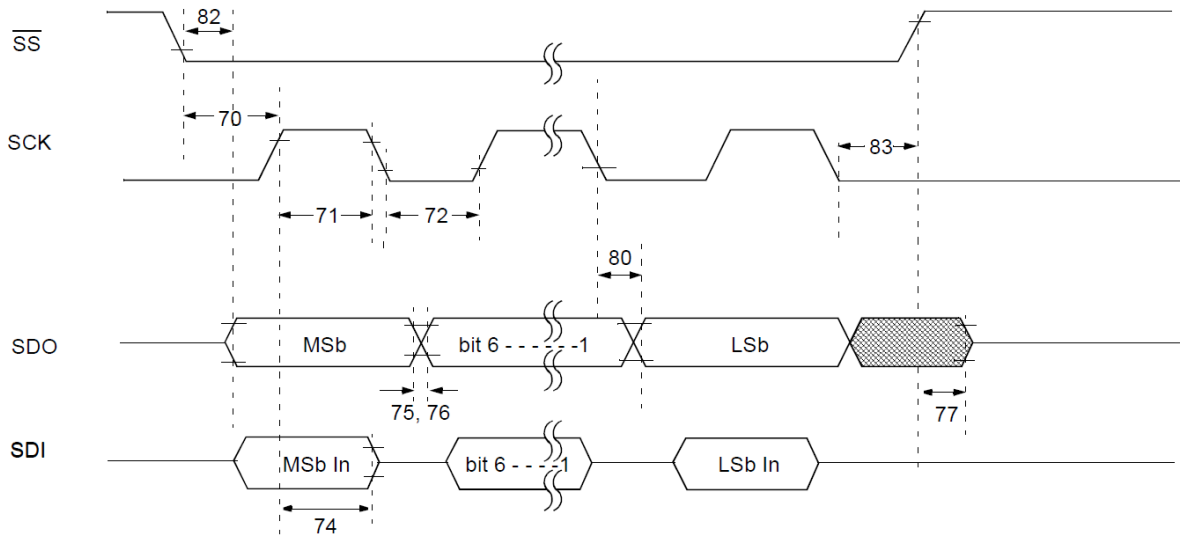
1. Wait 200µs
2. Wait for MISO 'low'



Slave->master



### 5.3.2. SPI Timings



Param No.	Symbol	Characteristic	Min	Max	Units	
70	TssL2scH, TssL2scL	$\overline{SS}$ ↓ to SCK ↓ or SCK ↑ Input	Tcy	—	ns	
71 71A	TscH	SCK Input High Time (Slave mode)	Continuous Single Byte	1.25 Tcy + 30 40	— —	ns ns
72 72A	TscL	SCK Input Low Time (Slave mode)	Continuous Single Byte	1.25 Tcy + 30 40	— —	ns ns
73A	Tb2b	Last Clock Edge of Byte 1 to the First Clock Edge of Byte 2	1.5 Tcy + 40	—	ns	
74	Tsch2diL, TscL2diL	Hold Time of SDI Data Input to SCK Edge	100	—	ns	
75	TdoR	SDO Data Output Rise Time	XXXX	—	25	ns
76	TdoF	SDO Data Output Fall Time	—	25	ns	
77	TssH2doZ	$\overline{SS}$ ↑ to SDO Output High-Impedance	10	50	ns	
78	TscR	SCK Output Rise Time (Master mode)	XXXX	—	25	ns
79	TscF	SCK Output Fall Time (Master mode)	—	25	ns	
80	Tsch2doV, TscL2doV	SDO Data Output Valid after SCK Edge	XXXX	—	50	ns
82	TssL2doV	SDO Data Output Valid after $\overline{SS}$ ↓ Edge	XXXX	—	50	ns
83	Tsch2ssH, TscL2ssH	$\overline{SS}$ ↑ after SCK Edge	1.5 Tcy + 40	—	ns	

Tcy = 150ns

#### 5.4. Protocol for 1WIRE (Dallas) bus.

Family code	ID1...ID5	Address	CRC
1 byte	5 bytes	1 byte	1 bytet

#### ID1...5 – unique ID number of transponder

CRC\_DAL- check sum of data send

The format conforms 1-WIRE Dallas (e.g.. DS1990A). It means, that described module could be used as a replacement of DS1990A drop.

During operation, a module tries to read-out transponder periodically. If it fails (no successful read-out), module does not response for pulses sent from 1-WIRE master unit. Bus does not "see" the module, which corresponds with lack of reader applying, it means applying the DS1990A drop to drop reader. If module reads out the transponder, the module starts to send data via 1-WIRE bus.

Calculate the CRC value

According to DS1990A specification C value is calculated from equation  $x^8+x^5+x^4+1$  with initial value equal to 0x00. The CRC is calculated on basis of all frame bytes excluding the last one.

An example of CRC value calculation procedure written in C language

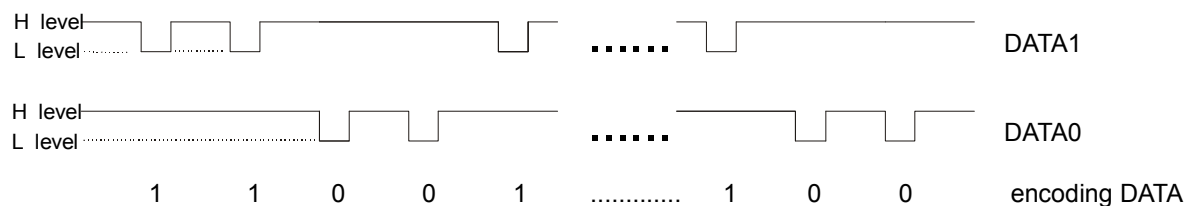
```
unsigned char CalcCRCDallas(unsigned char *SourceAdr)
{
  unsigned char i,k,In,CRC=0;
  for(i=0;i<7;i++)
  {
    In=*SourceAdr;
    for(k=0;k<8;k++)
    {
      if((In^CRC)&1) CRC=((CRC^0x18)>>1)|0x80;
      else CRC=CRC>>1;
      In>>=1;
    }
    SourceAdr++;
  }
  return(CRC);
}
```

where \*SourceAdr is beginning flag of data buffer

### 5.5. Wiegand protocol

The format conforms WIEGAND protocol specification for N bits. During operation, a module tries to read-out transponder periodically. If it fails (no successful read-out), module does not send data (bus does not "see" the module). If module reads out the transponder, the module starts to send data via Wiegand bus.

Pulse sequence from left to right.



Total number of pulses (level L) is equal to N. The first being bit sent complements up to parity the bits from first half of total bits. The last bit N complements up to non-parity the bits from second half of bits being sent.

It means, that two bits out of N bits assure the transmission correctness. Information is being sent is written by means bits 2 to N-1, it gives N information bits.

Check sums for bit sequence:

for even N:

EXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

or for odd N:

XXXXXXXXXXXXXXXXXXXXXXXXXXXXX.....  
.....YYYYYYYYYYYYYYYYYYYYYO

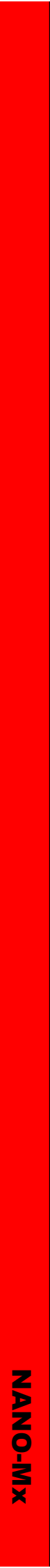
Where:

E = bit complementing up to parity

O = bit complementing up to non-parity

X = mask for parity calculation

Y = mask for non-parity calculation





## 6 . Communication protocol commands

### 6 . 1 . Commands for communication with transponders

#### 6 . 1 . 1 . Key management introduction

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.

#### 6 . 1 . 2 . 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
--------	-------------------	---------------	-----

### 6.1.3. 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
--------	-------------------	--	---------------	-----

## 6.2. Commands for communication with transponder

### 6.2.1. 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
--------	-------------------------	--	---------------	-----

### 6.2.2. 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

### 6.2.3. 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
--------	-------------------	---------------	-----

### 6.2.4. 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:

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
--------	-------------------	---------------	-----

### 6.2.5. 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	

### 6.2.6. 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
--------	-----------------	--	---------------	-----

### 6.2.7. 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
--------	----------------	--	---------------	-----

### 6.2.8. 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
--------	------------------	--	---------------	-----

### 6.2.9. 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

### 6.2.10. 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	whichever

	transponder block.	
--	--------------------	--

Response frame:

Header	C_WriteValue +1		OperationCode	CRC
--------	-----------------	--	---------------	-----

### 6.2.11. 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

### 6.2.12. 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
--------	------------------------	--	---------------	-----

### 6.2.13. 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
--------	--------------------	--	---------------	-----

### 6.2.14. 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
--------	----------	--	---------------	-----

### 6.3. MAD – Mifare Application Directory

#### 6.3.1. Card MAD formatting

Command frame:

Header	C_FormatMad	Type, Infobyte	CRC
--------	-------------	----------------	-----

Wherein:

Parameter name	Parameter description	Value range
C_FormatMad <b>0xA8</b>	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

#### 6.3.2. Adding the application to MAD directory

Command frame:

Header	C_AddApplication	LSB, MSB, Sector	CRC
--------	------------------	------------------	-----

Wherein:

Parameter name	Parameter description	Value range
C_AddApplication <b>0xAA</b>	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

### 6.3.3. 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.

### 6.3.4. 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.

#### 6.4. Reader inputs and outputs

Reader has inputs and outputs which are configurable. Inputs are controlled directly from microcontroller outputs. Output load current is up to 20 mA.

##### 6.4.1. Writing the output state

Command frame:

C_WriteOutputs	IONo, State
----------------	-------------

Where:

Parameter name	Parameter description	Value range
C_WriteOutputs	Output state write	0x70
IONo	I/O port number. The port should be configured as an output	0x0..0x4
State	Requested output state	0x00 or 0x01

Response frame:

C_WriteOutputs +1		OperationCode
-------------------	--	---------------

##### 6.4.2. Reading the input state

Command frame:

C_ReadInputs	IONo
--------------	------

Where:

Parameter name	Parameter description	Value range
C_ReadInputs	Input state read-out	0x72
IONo	I/O port number. Should be configured as an input.	0x0..0x4

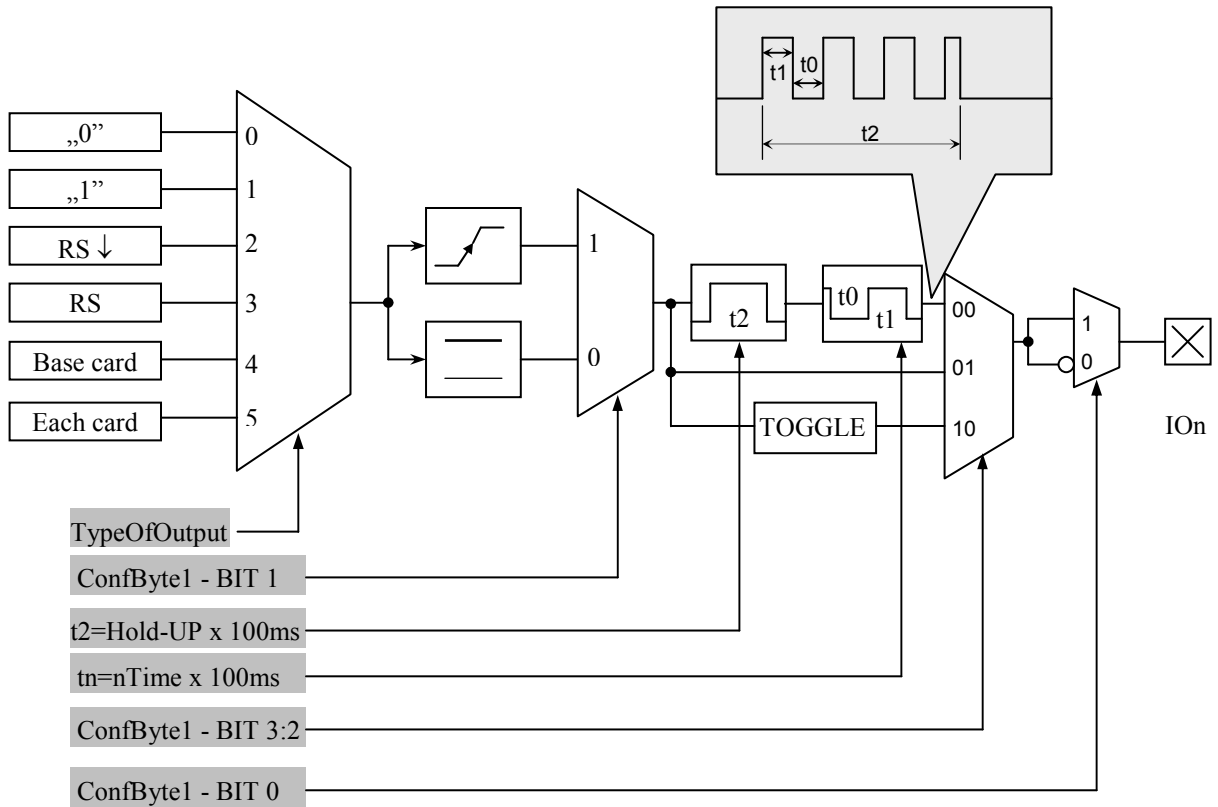
Response frame:

C_ReadInputs +1	State,[COUNTER]	OperationCode
-----------------	-----------------	---------------

Where:

Parameter name	Parameter description	Value range
State	Input state which has been read	
Counter	Counter state for counter type input.	

##### 6.4.3. 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, ConfByte1, TypeOfOutput, Hold-up, 0Time, 1Time

Where:

Parameter name	Parameter description	Value range
C_SetIOConfig	Writing the configuration of every port	0x50
IONo	I/O port number, which is to be configured	0x0..0x4
Dir	Port direction	0x00 – output
ConfByte1	One byte in which: BIT0 assigns output type as normally open or normally closed. BIT 1 determines reaction method of each output as sensitive for simulation changing (slope sensitive) or as sensitive for simulation state (state sensitive). BIT3:2 determines operation method of output referring to trigger signal state.	<b>ConfByte1 Bit 0</b> 0-Normally closed 1-Normally open <b>ConfByte1 Bit 1</b> 0-level sensitive 1-slope sensitive <b>ConfByte1 Bit 3:2</b> 00 – rectangular wave generator 01- directly 10 – output state change

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, which 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 an 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.	0x0-0x4
Dir	Port direction	0x01 – input
TypeOfInput	Input type	0x03
Delay	Delay	0x00

**6.4.4. 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 read-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.	

## 6.5. Access password

### 6.5.1. 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
--------	----------------	--	---------------	-----

### 6.5.2. 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
--------	---------------------	--	---------------	-----



### 6.5.3. 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
--------	-----------------	---------------	-----

### 6.5.4. Writing the “automatic read” configuration

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

Because of high security level provided by Mifare transponders, there is no possibility of operation of UID reading automatic device and communication with transponders via RS-232 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, RFU,AMulti	CRC
--------	-----------------------	---	-----

Where:

Parameter name	Parameter description	Value range
C_SetAutoReaderConfig	Writing the automatic device configuration.	0x58
ATrig	Defines, when automatic reading device UID will operate.	0-automatic device disabled permanently 1-automatic device enabled permanently 2=enabled automatically in case of

		transmission lack on 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 C_Select	0x00...0xff
ASerial	Automatic sending the UID transponder number, after reading it automatically from transponder.	0-never 1-for the first applying the transponder only 2-sends all
AMode	Selection the format of sending number  8 bits: MSB <span style="float:right">LSB</span> 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 number 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
RFU	reserved	0
AMulti	Multi type of transponders read mode	0 – read a only selected by CSetTransponderType command transponder type 0xff – read all known transponder types

Response frame:

Header	C_SetAutoReaderConfig +1	OperationCode	CRC
--------	--------------------------	---------------	-----

**6.5.5. 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.

### 6.5.6. Setting the date and time

Following setting has no influence for reader operation today.

Command frame:

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

Where:

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

Response frame:

Header	C_SetRtc +1		OperationCode	CRC
--------	-------------	--	---------------	-----

### 6.5.7. Reading-out the date and time

Command frame:

Header	C_GetRtc		CRC
--------	----------	--	-----

Where:

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

Response frame:

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

Where:

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

## 6.6. Configuring the UART serial interface

### 6.6.1. 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 RS-485 bus	0x01...0xfe
Baudrate	Data baud rate on RS-485 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
-------------------------	---------------

### 6.6.2. 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.

## 6.7. Other commands

### 6.7.1. 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

### 6.7.2. Sleep mode

This commands set a module into sleep mode. Depending on communication interface used, there are various methods of awaking, and then:

- For RS232 interface, awaking will occur, after positive slope is applied to /POWERDOWN terminal
- For I<sup>2</sup>C interface, awaking will occur, after sending the byte with proper number of SLAVE device. After this byte, awaking process begins which lasts 2 ms. Sending the subsequent data to NANO module should occur after this time elapses. Applying positive slope to /POWERDOWN terminal also causes a device to leave sleep mode.
- For SPI interface, awaking will occur, after receiving of one complete byte. After that, next data sending is allowed, after awaking process is completed, i.e. after 2 ms.

Command frame:

C_Sleep	
---------	--

Where:

Parameter name	Parameter description	Value range
C_Sleep	Entering sleep mode	0xda

Response frame:

C_Sleep +1		OperationCode
------------	--	---------------

### 6.7.3. 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 sequence of dots, which are written as an ASCII codes.

## 6.8. Code meanings in response frames

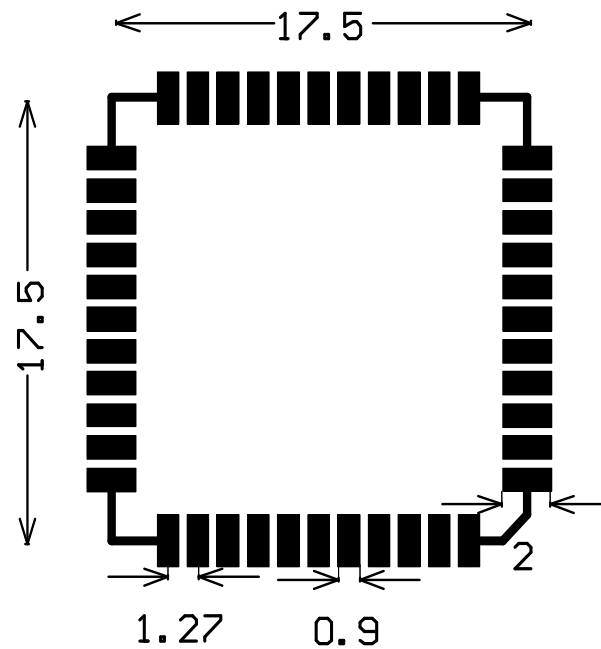
Operation code name	Description	Value
OC_Error	Error	0x00
OC_ParityError	Parity error	0x01
OC_RangeError	Parameter range error	0x02
OC_LengthError	Data quantity error	0x03
OC_ParameterError	Parameter error	0x04
OC_Busy	Internal modules are busy at the moment.	0x05
OC_NoACKFromSlave	No internal communication	0x22
OC_CommandUnknown	Unknown command	0x07
OC_WrongPassword	Wrong password or last password terminated i.e. automatic LogOut occurred.	0x09
OC_NoCard	No transponder	0x0a
OC_BadFormat	Wrong data format.	0x18
OC_FrameError	Transmission error. Noise occurrence possible.	0x19
OC_NoAnswer	No response from transponder.	0x1E
OC_TimeOut	Operation time out. No transponder in reader field possible.	0x16
OC_Successful	Operation completed successfully.	0xff

## 7 . Reset to default settings

To restore default settings, connect reset terminal with ground for 2 s or longer. During restoring the defaults following reader parameters are fixed:

<b>Parameter name or functionality</b>	<b>Value or setting</b>
Address on serial bus	0x01
Baud rate on serial bus	9600 bps
Access password	0x0 - no password
Port 0	Common purpose input
Port 1	Common purpose input
Port 2	Common purpose input
Port 3	Common purpose input
Port 4	Common purpose input
“Autoreader” configuration	0x2,0x14,0x1,0x1,0xff

## 8 . Footprint proposed for NANO module.



Footprint dimensions recommended for SMD NANO-MS version of the module

Latest news concerning the NETRONIX products  
<http://www.netronix.pl/>