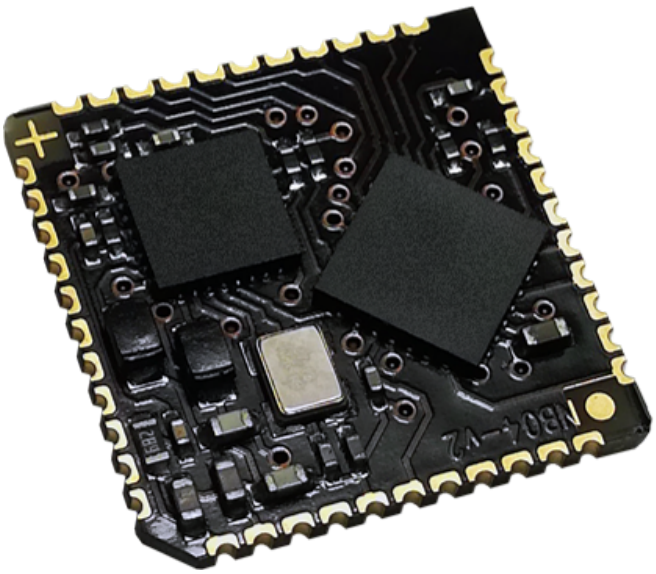
Technical Data Sheet

RFID reader

NANO-RS



1 . Introduction

NANO-RS module is OEM miniature RFID card reader operating at frequency of 13,56 MHz.
Main features:

- Support of Mifare Classic, Mifare UltraLight, NFC NTAG, Mifare Desfire EV, Mifare Plus, NFC UID (Android NX UID),
- Support ISO15693 TAGS (I-CODE SLI, iCLASS CSN, LEGIC CSN),
- Support ISO14443B,
- Support ISO14443A-4 raw communication (SmartMX, etc.),
- Netronix USER ID support (secured ID with AES crypto functionality)
- Netronix NFC ID support
- UART(TTL) interface with RS485 bus transmitter/receiver control output,
- Addressability on bus in RS-485 mode,
- SPI interface,
- 1WIRE (Dallas DS1990) interface,
- Interface WIEGAND,
- Signal terminal for reset to factory defaults,
- 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 UART interface.

2 . Specifications

Transponder operate frequency	13,56 MHz
Supported transponder type	Mifare S50,S70,UL,Desfire, Plus, SmartMX*, ISO15693
Approximate maximal communication range with transponders (using ø 50 mm antenna)	Classic – 8cm Plus/Desfire – 5cm
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 sleep mode	 25 mA 16,5 mA Max. 70mA – zależne od zastosowanej anteny Max. 15 µA
Supported two-way interfaces: Supported one-way interfaces:	- UART 3.3V version with terminal controlling RS485 driver - SPI - 1WIRE (DS1990 pill emulation) - WIEGAND
Common purpose inputs/outputs	5 configurable inputs/outputs
Dimensions	17.5 x 17.5 x 3 mm

3 . Terminal description

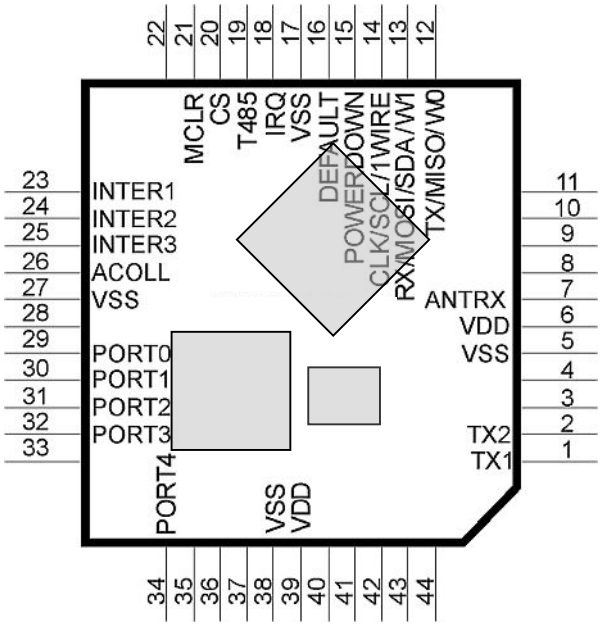


Fig. 1

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	Not used
12	TX/MISO/W0	TX(data out) for UART, MISO(data out) for SPI, ‘0’ for Wiegand
13	RX/MOSI/W1	RX(data in) for UART, MOSI(data in) for SPI, ‘1’ for Wiegand
14	CLK/SCL/1WIRE	CLK signal for SPI 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	RFU	RFU
19,20	T485, /CS	Transmit/receive switching output for RS485 interface transceiver, Chip select input for SPI bus
21	/MLCR	Input of hardware reset of NANO module
23	INTER1	Communication interface select, see diagrams below
24	INTER2	
25	INTER3	
26	/ACOLL	PORT0 (connected with terminal #29)
27	VSS	Ground of module supply
29	PORT0	Input/output port of common purpose
30	PORT1	Input/output port of common purpose
31	PORT2	Input/output port of common purpose
32	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

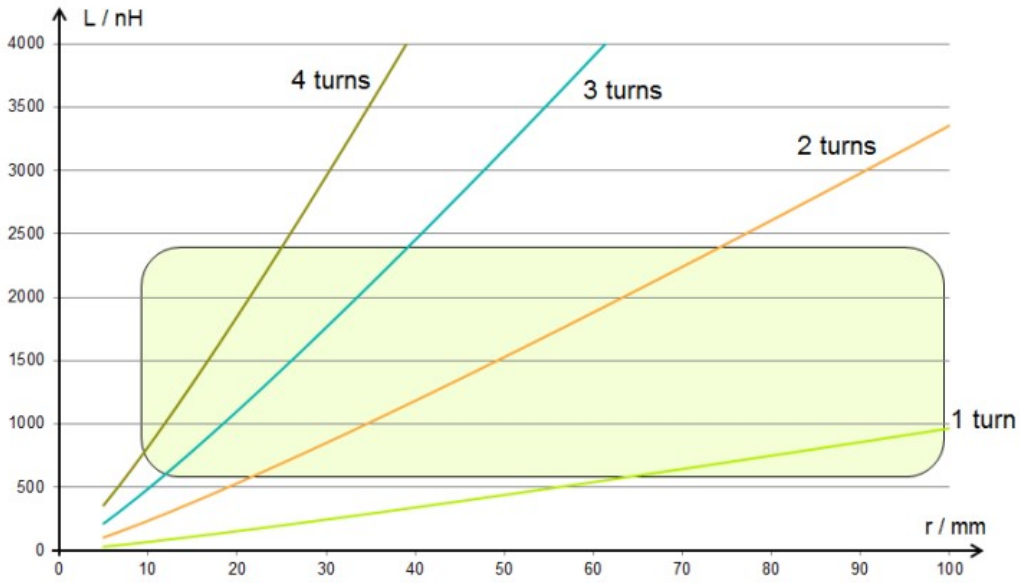
INTER1	INTER2	INTER3	Interface type	Default settings of interface
1	1	1	UART	9600 bps, 8, N, 1
1	0	1	SPI	
0	1	1	DALLAS	Addres: 0x01, family code: 0x01
0	0	1	WIEGAND	37 bits

“1” – connected to +3,3 V
„0” – connected to GND

4 . 2 . Selecting external elements and designing antenna

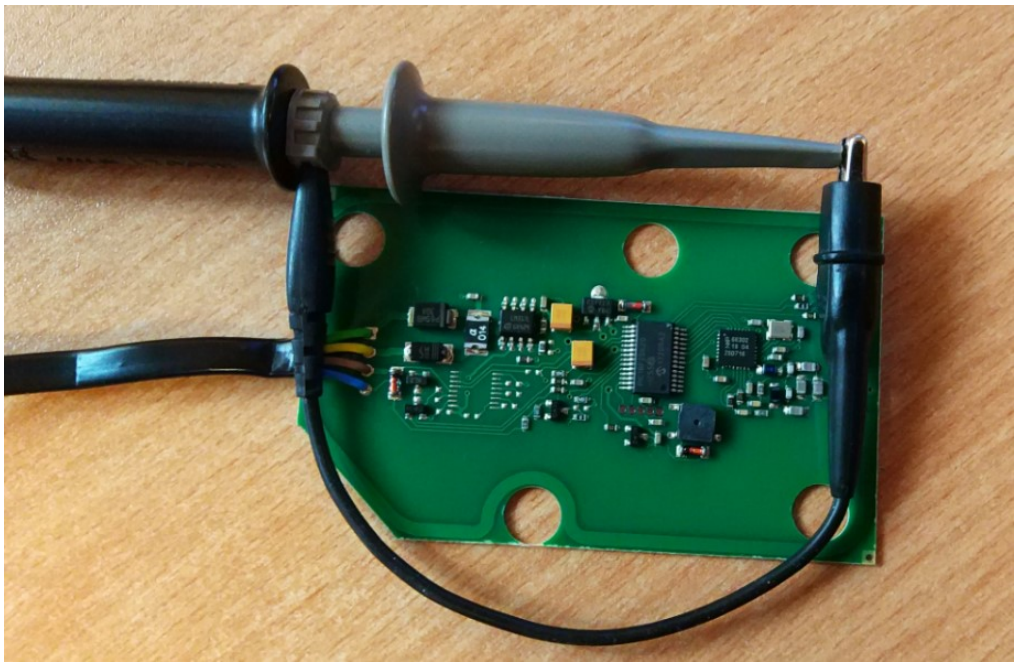
It is recommended that transmit/receive antenna inductance to be 0,5µH – 2µH. Inductance L with capacitor C used should form resonance circuit for 13,56 MHz frequency. Let start with 100pF and tune (typically C range is 47-330pF) for best read range (field amplitude, as in photo below). Resistance R sets the antenna quality which should be 8 to 15, in most cases can be omitted. Be sure to use a capacitor with properly high voltage, at least 50 V.

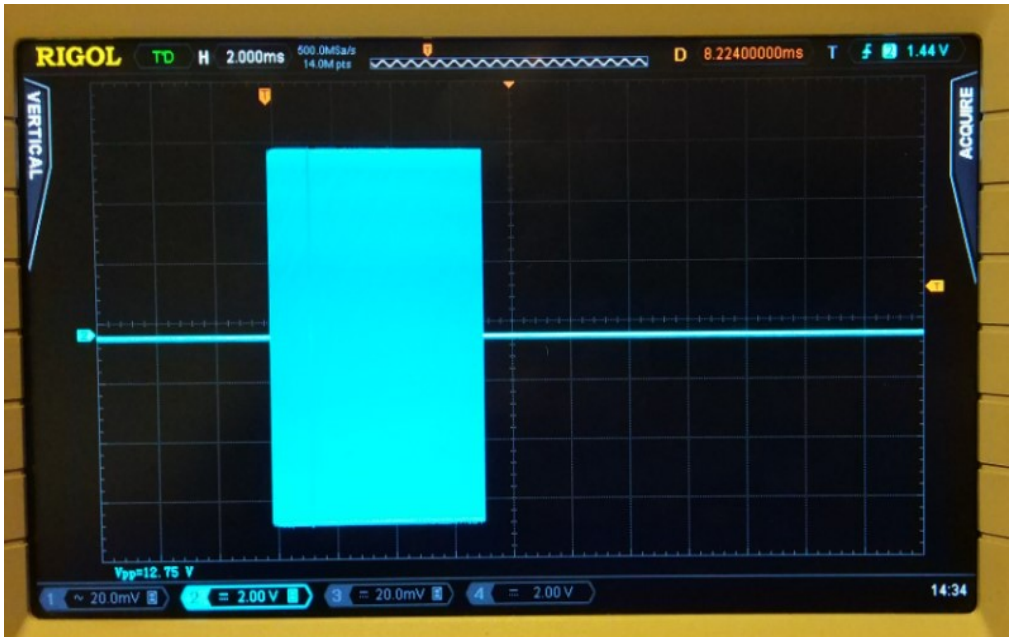
For design antenna please use following relation, trace width is typically 30-60mils:



Inductance versus coil radius

To find best C value, please use below method and find maximum amplitude (1:10 divider on probe should be enabled):





NANO-RS

4 . 3 . Application diagrams

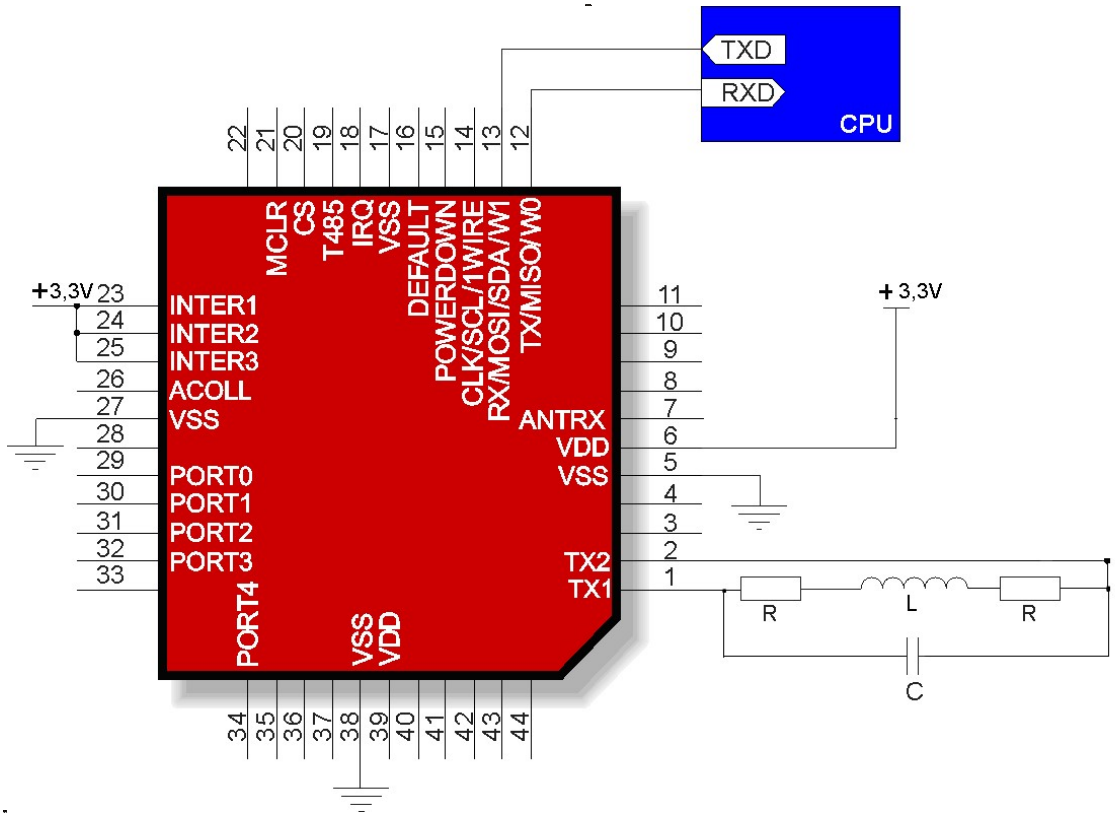


Fig. 2 Minimal configuration for UART interface

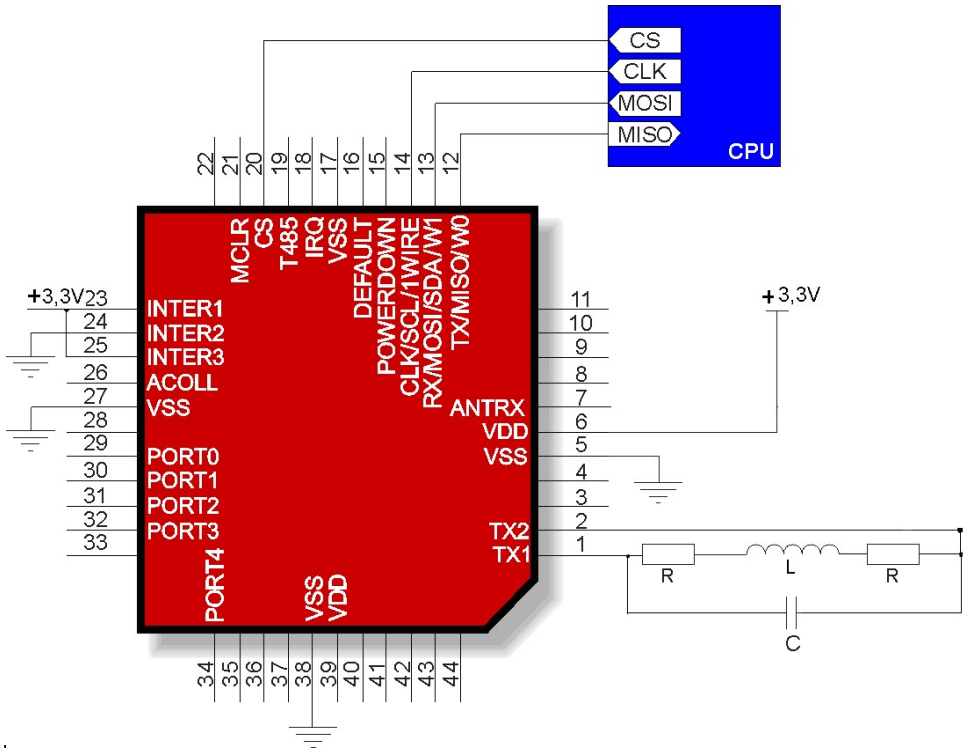


Fig. 3 Minimal configuration for SPI interface

5 . Transmision protocols

5.1. UART transmission protocol

In this data sheet UART protocol has been confined to descriptions of commands, responses and their parameters. Header and CRC control sum (CRC-16 XMODEM) 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

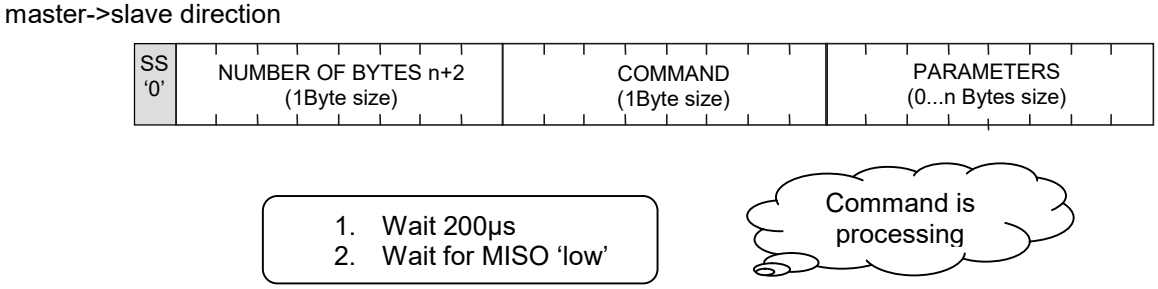
RS protocol operation can be tested by means of development tools including free of charge “FRAMER” software”.

5.2. SPI transmission protocol

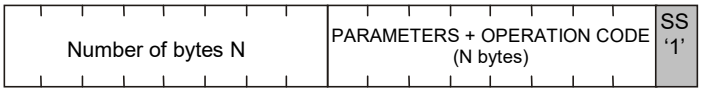
5.2.1. Data exchange algorithm

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

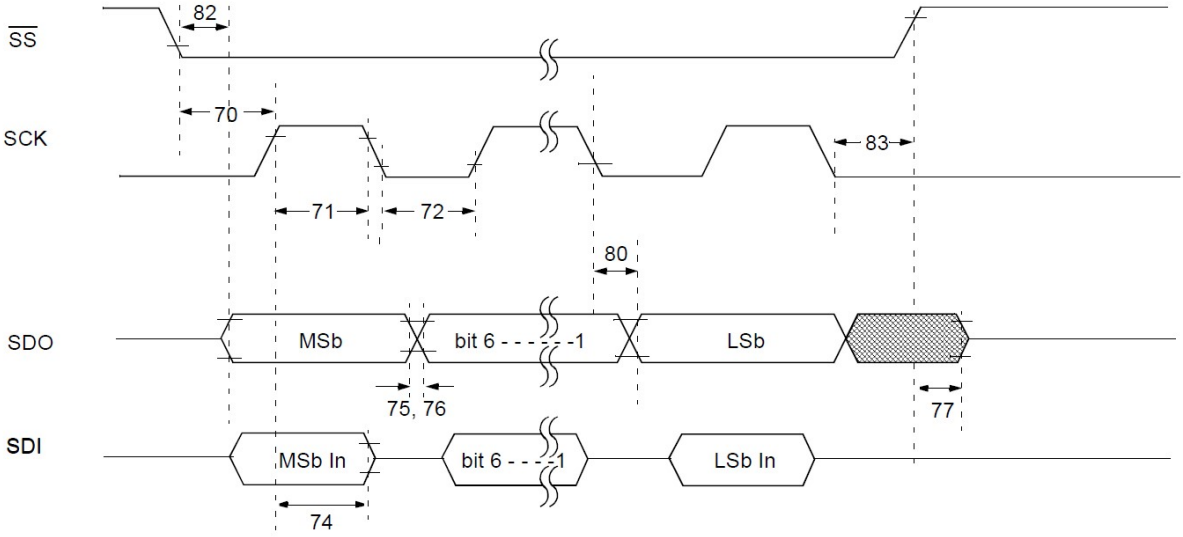
1. SS pin goes low.
2. Master device sends a command with parameters to slave (NANO) device.
3. Commands is executing
4. NANO module is ready to response, when MISO line is on LOW level. MISO level test should be perform at least after 200us from end of command sending.
5. Master reads data and operation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



Slave->master



5.2.2. SPI Timings



Param No.	Symbol	Characteristic		Min	Max	Units	
70	TssL2scH, TssL2scL	$\overline{SS}$ ↓ to SCK ↓ or SCK ↑ Input		TCY	—	ns	
71	TscH	SCK Input High Time (Slave mode)	Continuous	1.25 TCY + 30	—	ns	
71A			Single Byte	40	—	ns	
72	TscL	SCK Input Low Time (Slave mode)	Continuous	1.25 TCY + 30	—	ns	
72A			Single Byte	40	—	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

5.3. 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);
```

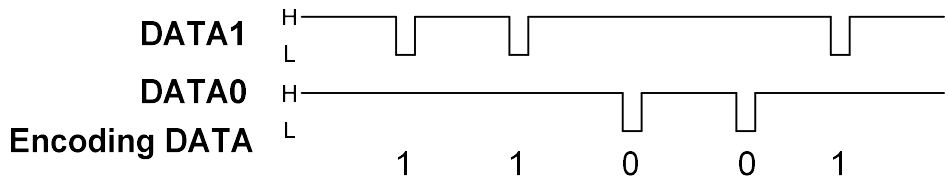
}

where *SourceAdr is beginning flag of data buffer

5.4. Wiegand protocol

Reader, after being configured to operate in WIEGAND mode, sends a unique ID number of the read card in accordance with the Wiegand protocol with the following parameters:

Pulse duration (L level) 100us
Interval between impulses (H level) 1ms



Reader allows you to change the length of the WIEGAND frame and to select the part of the ID of the card to be sent on the bus.
Exmaples:
ID cards = 0x123456789A = 0b00010010000110100010101100111100010011010

WIEGAND parameters	Card ID / responding WIEGAND frame	
P1=26, P2=0	0b00010010000110100010101100111100010011010 P00010010000110100010101 10N	Card ID WIEGAND frame
P1=37, P2=0	0b00010010000110100010101100111100010011010 P00010010000110100010101100111100010 0N	Card ID WIEGAND frame
P1=26, P2=1	0b00010010000110100010101100111100010011010 P01010110011110001001101 0N	Card ID WIEGAND frame

P,N – parity bits,
Another format of WIEGAND, can be obtained by changing the configuration using *C_SetInterfaceConfig* command. For change bytes order, please set 'I' bit of 'Amode' byte for C_SetAutoreaderConfig command.

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. Obtaining ID and selecting ISO14443A TAG.

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:

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

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 transponder block.	whichever

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. GET I-CODE SLI/ ISO15693 ID (inventory)

Command frame:

Header	C_ Inventory		CRC

Gdzie:

Parameter name	Parameter description	Value range
C_Inventory	Get ID	0x04

Response frame:

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

6.2.15. SLI Reading PAGE

Command frame:

Header	C_SLIReadPage	PageAdr	CRC

Where:

Parameter name	Parameter description	Value range
C_ SLIReadPage	Read page from SLI	0x2C
PageAdr	Page address	Depend on TAG type

Response frame:

Header	C_ SLIReadPage +1	Data1…4	OperationCode	CRC

Gdzie:

Parameter name	Parameter description	Value range
Data1…4	Data out.	any

6.2.16. SLI Witting PAGE

Command frame:

Header	C_SLIWritePage	PageAdr, Data1...4	CRC

Where:

Parameter name	Parameter description	Value range
C_ SLIWritePage		0x2E
PageAdr	Page address	Depend on TAG type
Data1...4	Data to write	any

Response frame:

Header	C_ SLIWritePage +1		OperationCode	CRC

6.2.17. Setting the transponder in field into sleep mode

To set transponder to sleep mode, select it first.

Command frame:

Header	C_Halt		CRC

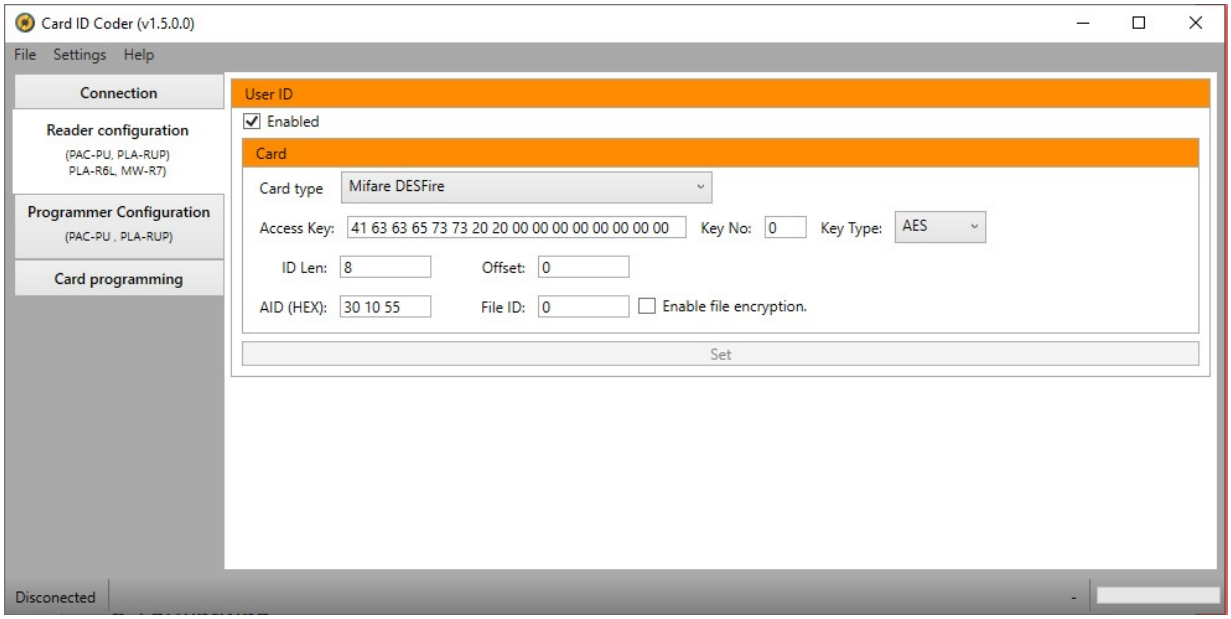
Parameter name	Parameter description	Value range
C_Halt	Setting the transponder in field into sleep mode.	0x40

Response frame:

Header	C_Halt+1		OperationCode	CRC

6.3. USER ID feature

Using USER_ID functionality, reader is able to read ID stored on secured part of memory Mifare PLUS/ Mifare Desfire and Mifare Classic transponder. Few security algorithm (DES/3DES/AES) and flexible data structure can be set. To setup reader, use free CardIDCoder tool https://netronix.pl/en/software/cardidcoder



6.4. NFC ID feature

Using NFC ID functionality, reader is able to read ID which is generated on Android device with NFC support. Device should have installed free application: https://netronix.pl/en/software/nfc-id

Generated ID is unique and based on device MAC address.

6.5. MAD – Mifare Application Directory

6.5.1. 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

6.5.2. 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

6.5.3. 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.

6.5.4. 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.

6.6. 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.

6.6.1. 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

6.6.2. Reading the input state

Command frame:

C_ReadInputs	IONo

Where:

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

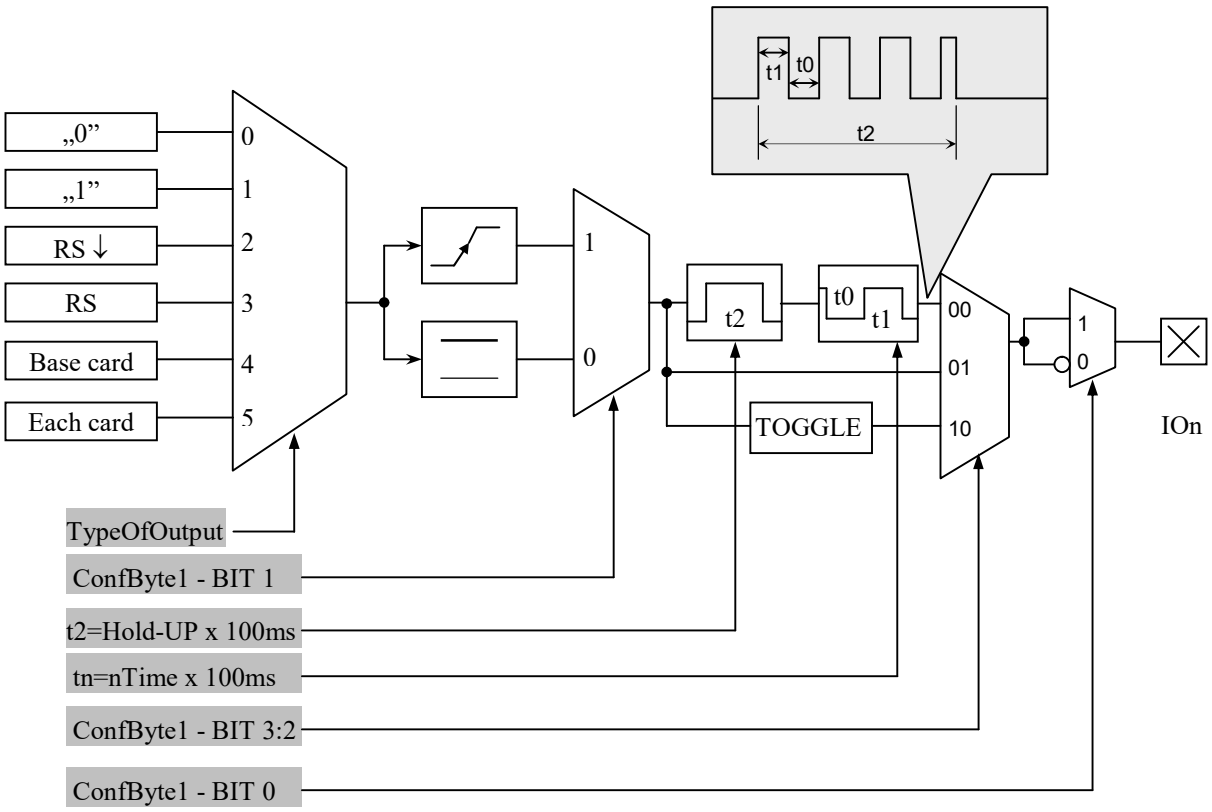
Response frame:

C_ReadInputs +1	State,[COUNTER]	OperationCode

Where:

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

6.6.3. 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.	**ConfByte1 Bit 0** 0-Normally closed 1-Normally open **ConfByte1 Bit 1** 0-level sensitive 1-slope sensitive **ConfByte1 Bit 3:2** 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, 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.	0x0-0x4
Dir	Port direction	0x01 – input
TypeOfInput	Input type	0x03
Delay	Delay	0x00

6.6.4. 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.	

6.7. Access password

6.7.1. 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

6.7.2. 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

6.7.3. 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

6.7.4. 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 UART 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_SetAutoReader Config	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 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 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	0x00…0xff
ASerial	Automatic sending the UID transponder number, after reading it automatically from transponder.	0-never 1-for the first applying the transponder only 2-sends all
AMode	Selection the format of sending number 8 bits: MSB … LSB R R H CR R E I A	R: Reserved, always 0 CR=1: Number which is ended with line end mark CR+LF E=1: information extended with cards umber in filed and card type

		I=1	Number in reversed order
		A=1 H=0	Number sent in ASCII format
		A=0 H=0	Number sent in Nertonix format
		A=0 H=1	Number sent in binary 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	
AMask	MSB LSB R R R R S I C M	R RFU, always 0 M=1 Mifare (ISO14443-3A) C=1 ISO14443B (Calypso) I=1 IClass CSN S=1 ICodeSLI (ISO15693)	

Response frame:

Header	C_ SetAutoReaderConfig +1		OperationCode	CRC

6.7.5. 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.

6.7.6. 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

6.7.7. 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.

6.8. Configuring the UART serial interface

6.8.1. Writing the configuration of serial port

Command:

	C_SetInterfaceConfig	Mode, P1, P2	

Where:

Parameter name	Parameter description	Value range
C_SetInterfaceConfig	Serial interface configuration write	0x54
Mode		0x01
P1	For UART: Address on RS-485 bus For WIEGAND: number of bits	0x01…0xfe
P2	For UART: Data baud rate on RS-485 bus For WIEGAND: left/right justify	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	

6.8.2. 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.

6.9. Other commands

6.9.1. 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

6.9.2. Sleep mode

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

- For UART interface, awaking will occur, after positive slope is applied to /POWERDOWN terminal
- For I^2C 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

6.9.3. 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.

6.10. 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.	0x1F
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:

Parameter name or functionality	Value or setting
Address on serial bus	0x01
Baud rate on serial bus	9600 bps
Access password	0x0 - no password
Port 0	Direct output controlling via command
Port 1	Direct output controlling via command
Port 2	Direct output controlling via command
Port 3	0,2s pulse output controlling via command
Port 4	0,2s pulse output controlling via command
“Autoreader” configuration	0x02,0x14,0x01,0x01,0x01

8 . Operation example of transponder

8 . 1 . Mifare Classic transponders

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

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, 0x00,0x00, 0x00.

01	0B	0x58 C_SetAutoReaderConfig	00 00 00 00 00 00		44 B6

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

01	0C	0x16 C_LoadKeyToSKB	FF FF FF FF FF FF 00		4B 74

Enable the field.
TurnOnAntennaPower, 0x01

01	06	0x10 C_TurnOnAntennaPower	01		D7 46

Apply transponder to reader.

Select transponder
C_Select, 0x00

01	06	0x12 C_Select	00		A1 05

In result, card UID is return and card is ready for next operations,

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

01	08	0x1A C_LoginWithSKB	03 AA 00		9F 64

Write some data(00,11,22…FF) to 2nd block in 3rd sector.
C_WriteBlock, 0x02, 0x00,0x11,0x22,0x33,0x44,0x55,0x66,0x77,0x88,0x99,0xAA,0xBB,0xCC,0xDD,0xEE,0xFF

01	16	0x1C C_WriteBlock	02 00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF		21 55

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

01	06	0x1E C_ReadBlock	02		C4 2A

8.2. NTAG/Ultralight transponders

After correct connection of reader and achieving the bi-directional communication between the reader and master computer, it is possible to perform read-out and write operation of transponder memory.

Following operation assumes, that reader is in default condition, and applied card is in default condition too. Frames, for test purpose, should be sending using Framer tool.

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, 0x00,0x00, 0x00.

01	0B	0x58 C_SetAutoReaderConfig	00 00 00 00 00 00		44 B6

Enable the field.
TurnOnAntennaPower, 0x01

01	06	0x10 C_TurnOnAntennaPower	01		D7 46

Apply transponder to reader.

Select transponder
C_Select, 0x00

01	06	0x12 C_Select	00		A1 05

In result, card UID is return and card is ready for next operations,

Write some data(00,11,22,33) on 4th page. Only 4 bytes can be written once.
C_WritePage4B 0x04 0x00 0x11 0x22 0x33

01	0A	0x26 C_WritePage4B	04 00 11 22 33		97 95

Read-out 4th page, 16 bytes (pages:4-7) are read-out at once.
C_ReadPage16, 0x04

01	06	0x28 C_ReadPage16B	04		0B DF

8.3. 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 AES key, create a data file, save and then read the data from the file.

Because during manual tests the time between successive commands sent on interface 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, 0x00, 0x00.

To read the transponder, first load the keys to the key memory.
So we load the default desfire transponder key into the "3" position of the reader's memory, and for position 4 we load our own AES key, which we will give the new application:

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

3. C_DesSaveKey 0x04, 0x01, 0x02, 0x03, 0x04, 0x0a, 0x0b, 0x0c, 0x0d, 0x49, 0xa2, 0x22, 0x94, 0xe3, 0x10, 0xaa, 0xbc

Enable RF 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 AES keys

8. C_DesCreateApp 0x30,0x10,0x55,0x0F,0x84

We change the default, newly created ApplicationMasterKey key to the one we have saved in the reader in position 4. Therefore, we must 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, 0xAA
11. C_DesChangeKey 0x00, 0x04
12. C_DesAuth 0x00, 0x04, 0xAA

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

8.4. Mifare PLUS transponders

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, 0x00, 0x00

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

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

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

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

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

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

RF field must be turned on.

TurnOnAntennaPower 0x01

Card should be put close to antenna

Transponder must be selected

C_Select 0x00

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

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

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

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

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

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

To write AES type A key for sector 0x01(same as we stored on reader at index 0x03)
C_ MfPlusCMD 0xA8 0x40 0x02 0x01, 0x02, 0x03, 0x04, 0x0a, 0x0b, 0x0c, 0x0d, 0x0e, 0x0f, 0x10, 0x12, 0x14, 0x16, 0x18, 0x20

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

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

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

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

To write data on 2 block and 3 sector send:

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

To write data from 2 block and 3 sector send:

C_ReadBlock 0x02

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

Now switch into ISO14443-4 mode is necessary

C_Init_ISO14443-4 0x00

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

Now card must be reset by sending below command twice

C_Select 0x00

now switch into ISO14443-4 mode is necessary

C_Init_ISO14443-4 0x00

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

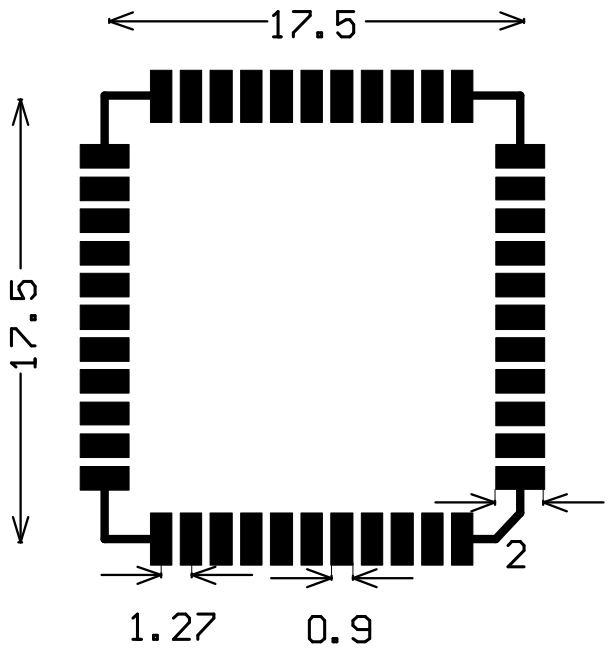
To write block 2 of sector 1 by some examples data:

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

To read block2 of sector 1:

C_ MfPlusCMD 0x33 0x02

9 . Footprint proposed for NANO module.



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

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