



## Technical Data Sheet

RFID Reader

# UW-R4G

UW-R4G-man-1  
For firmware at least v2.3



|             |                                                                              |           |
|-------------|------------------------------------------------------------------------------|-----------|
| <b>1.</b>   | <b>INTRODUCTION .....</b>                                                    | <b>5</b>  |
| <b>2.</b>   | <b>GENERAL SPECIFICATIONS .....</b>                                          | <b>6</b>  |
| <b>3.</b>   | <b>NAMES AND FUNCTIONS OF PARTS .....</b>                                    | <b>7</b>  |
| <b>4.</b>   | <b>SERIAL TRANSMISSION FORMAT .....</b>                                      | <b>9</b>  |
| <b>4.1.</b> | <b>Key management .....</b>                                                  | <b>10</b> |
| 4.1.1.      | Key loading into dynamic key memory .....                                    | 10        |
| 4.1.2.      | Key loading to key static memory.....                                        | 11        |
| <b>4.2.</b> | <b>Commands for communication with transponder .....</b>                     | <b>11</b> |
| 4.2.1.      | On/off switching of reader field .....                                       | 11        |
| 4.2.2.      | Selecting one of many transponders.....                                      | 12        |
| 4.2.3.      | Logging by means of Dynamic Key Buffer to selected sector of transponder     | 13        |
| 4.2.4.      | Logging by means of Static Key Buffer to selected sector of transponder..... | 13        |
| 4.2.5.      | Reading-out the content of transponder block .....                           | 14        |
| 4.2.6.      | Writing the content of transponder block.....                                | 14        |
| 4.2.7.      | Copying the content of transponder block into other block .....              | 15        |
| 4.2.8.      | Writing the page content into Mifare UL.....                                 | 15        |
| 4.2.9.      | Reading the page content in Mifare UL .....                                  | 16        |
| 4.2.10.     | Writing values to transponder block .....                                    | 16        |
| 4.2.11.     | Reading-out the values from transponder block .....                          | 17        |
| 4.2.12.     | Increasing the value included in transponder block .....                     | 17        |
| 4.2.13.     | Decreasing the value included in block transponder.....                      | 18        |
| 4.2.14.     | Setting the transponder in field into sleep mode .....                       | 18        |
| <b>4.3.</b> | <b>Mifare Plus/ Desfire .....</b>                                            | <b>18</b> |
| <b>4.4.</b> | <b>Electrical inputs and outputs .....</b>                                   | <b>19</b> |
| 4.4.1.      | Describing the output state .....                                            | 19        |
| 4.4.2.      | Reading-out the input state.....                                             | 19        |
| 4.4.3.      | Writing the settings to any port .....                                       | 20        |
| 4.4.4.      | Reading-out the configuration of freely selected port.....                   | 22        |
| <b>4.5.</b> | <b>Access password .....</b>                                                 | <b>23</b> |
| 4.5.1.      | Logging to reader .....                                                      | 23        |
| 4.5.2.      | Changing the password .....                                                  | 23        |
| 4.5.3.      | Logging out of the reader .....                                              | 24        |
| <b>4.6.</b> | <b>Operating the transponder internal memory .....</b>                       | <b>24</b> |
| 4.6.1.      | Reading-out the transponder number from memory .....                         | 24        |
| 4.6.2.      | Writing the transponder name to memory.....                                  | 25        |
| <b>4.7.</b> | <b>Operating the built-in access control.....</b>                            | <b>25</b> |

|                       |                                                                  |           |
|-----------------------|------------------------------------------------------------------|-----------|
| 4.7.1.                | Writing the configuration of access control .....                | 25        |
| 4.7.2.                | Reading-out the configuration of access control.....             | 25        |
| 4.7.3.                | Writing the automatic device configuration.....                  | 26        |
| 4.7.4.                | Reading-out the configuration of automatic device .....          | 28        |
| 4.7.5.                | Setting the date and time .....                                  | 28        |
| 4.7.6.                | Reading-out the date and time.....                               | 28        |
| <b>4.8.</b>           | <b>Configuring the RS-485 serial interface.....</b>              | <b>29</b> |
| 4.8.1.                | Writing the configuration of serial port.....                    | 29        |
| 4.8.2.                | Reading the configuration of serial interface .....              | 29        |
| <b>4.9.</b>           | <b>Managing the events.....</b>                                  | <b>30</b> |
| 4.9.1.                | Setting the event recorder.....                                  | 30        |
| 4.9.2.                | Reading the event recorder.....                                  | 31        |
| 4.9.3.                | Reading the counters related to event memory. ....               | 31        |
| 4.9.4.                | Reading the events .....                                         | 32        |
| <b>4.10.</b>          | <b>MAD – Mifare Application Directory .....</b>                  | <b>33</b> |
| 4.10.1.               | Card MAD formatting.....                                         | 33        |
| 4.10.2.               | Adding the application to MAD directory .....                    | 33        |
| 4.10.3.               | Pursuing the sector for given application .....                  | 34        |
| 4.10.4.               | Pursuing the next sector of application .....                    | 34        |
| <b>4.11.</b>          | <b>Other commands .....</b>                                      | <b>35</b> |
| 4.11.1.               | Remote reset of reader.....                                      | 35        |
| 4.11.2.               | Reading-out the reader software .....                            | 35        |
| 4.11.3.               | Setting buzzer volume.....                                       | 35        |
| 4.11.4.               | Reading buzzer volume.....                                       | 35        |
| <b>4.12.</b>          | <b>Meaning of operation code in response frame.....</b>          | <b>37</b> |
| <b>5.</b>             | <b>MODBUS RTU PROTOCOL .....</b>                                 | <b>38</b> |
| <b>5.1.</b>           | <b>Supported MODBUS protocol functions:.....</b>                 | <b>38</b> |
| <b>5.2.</b>           | <b>MODBUS adress (decimal) .....</b>                             | <b>38</b> |
| <b>5.3.</b>           | <b>Encapsulation Netronix protocol inside MODBUS RTU .....</b>   | <b>40</b> |
| 5.3.1.                | Example of using 0x17 function .....                             | 40        |
| <b>6.</b>             | <b>MEANING OF SYMBOLS AND MARKINGS USED IN THE SPECIFICATION</b> | <b>41</b> |
| <b>7.</b>             | <b>MECHANISM OF MASTER ID .....</b>                              | <b>42</b> |
| <b>8.</b>             | <b>CLEARING THE CARD MEMORY AND RESETING TO FACTORY</b>          |           |
| <b>DEFAULTS .....</b> |                                                                  | <b>42</b> |

**9. OPERATION EXAMPLE OF TRANSPONDER..... 43**

## 1 . Introduction

The UW-R4G is RFID card reader, which belongs to Mifare and I-CODE family.

Features of the card reader include:

- Support for: Mifare S50,S70, Plus, UltraLight C, Desfire, iClass(CSN), I-CODE SLI, ISO14443B
- 1000 card memory with built-in lock driver
- RS-485 interface
- RS-485 bus addressability
- Modbus and Netronix protocol
- Built-in relay and buzzer
- Built-in push-button and warning LED's on front panel
- Built-in push-button for reset to default settings
- Built-in tamper with spring
- Set-up capability for two-state inputs and outputs
- Buzzer, relay and LED's setup
- Two-state outputs control
- Read-out of two-state input
- Full access possibility to all card sectors on read and write level
- Data security by password
- Battery backup RTC
- Events memory with trigger mask
- Software update via RS485 interface using *NEFIR* program

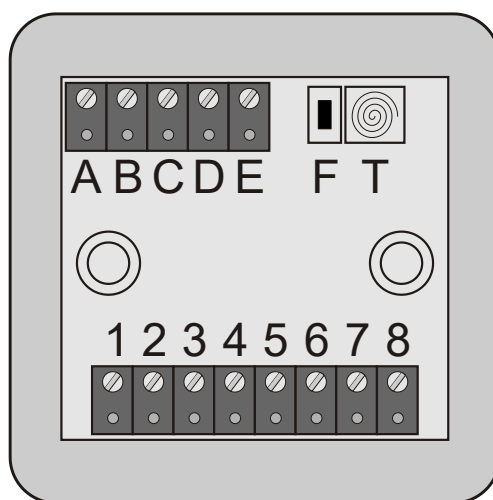
## 2 . General Specifications

| Supported functionality depending on transponder / card type: |                    |                                          |                                   |
|---------------------------------------------------------------|--------------------|------------------------------------------|-----------------------------------|
| Card Type                                                     | ID number read-out | Full write and read-out of memory blocks | Supported by internal lock driver |
| S50                                                           | YES                | YES                                      | YES                               |
| S70                                                           | YES                | YES                                      | YES                               |
| PLUS S, PLUS X                                                | YES                | YES (SL1, SL3)                           | YES                               |
| Ultralight, Ultralight C                                      | YES                | YES                                      | YES                               |
| DesFire                                                       | YES                | YES                                      | YES                               |
| iCLASS                                                        | YES (CSN)          | NO                                       | YES                               |
| I-CODE                                                        | YES                | YES                                      | YES                               |
| ISO14443B                                                     | YES                | NO                                       | YES                               |

| UW-R4G module parameters                  |                                                                                                                                |
|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| Supply voltage                            | 8-16 V                                                                                                                         |
| Max. supply current                       | 200 mA                                                                                                                         |
| Rated operation radio frequency of module | 13.56 MHz                                                                                                                      |
| Read-out distance                         | up to 8 cm                                                                                                                     |
| Maximum output current                    | 1A                                                                                                                             |
| Maximum total output current              | 2A                                                                                                                             |
| RS-485 communication                      | 2400, 4800, 9600, 19200, 38400, 57600, 115200 bps,<br>8 data bits, 1 stop bit, no parity<br>compliant with „Netronix Protocol” |

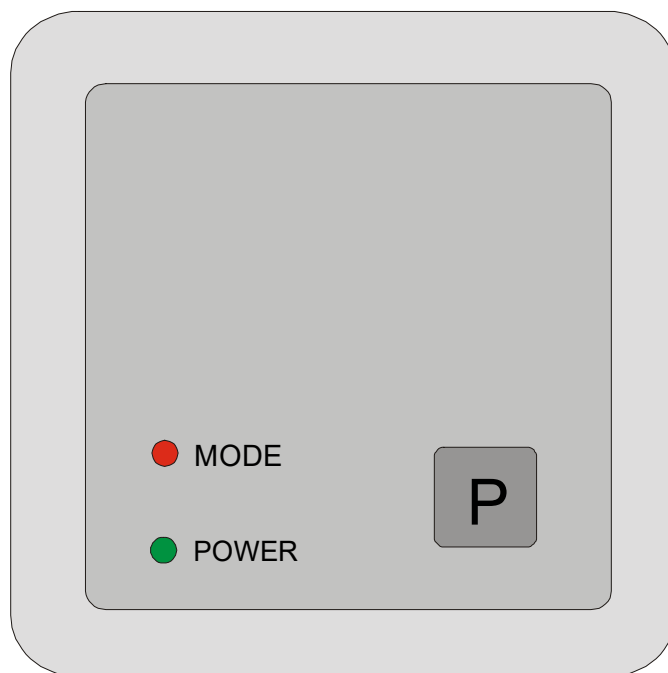
### 3 . Names and functions of parts

Rear view



UW-R4G

| Symbol on drawing | Function                                               |
|-------------------|--------------------------------------------------------|
| 1                 | IO 1                                                   |
| 2                 | Synchronization of two readers, which are in-proximity |
| 3                 | RS-485 pin A                                           |
| 4                 | RS-485 pin B                                           |
| 5                 | Module power supply                                    |
| 6                 | Ground and minus of supply voltage                     |
| 7                 | Relay contacts outputs                                 |
| 8                 |                                                        |
| T                 | Tamper with spring                                     |
| F                 | Push-button for reset to default settings              |
| A                 | GPIO 2                                                 |
| B                 | GPIO 3                                                 |
| C                 | GPIO 4                                                 |
| D                 | GPIO 5                                                 |
| E                 | GPIO 6                                                 |

**Front view**

| Symbol on drawing | Function                                                                               |
|-------------------|----------------------------------------------------------------------------------------|
| LED MODE          | Red LED, green LED and blue LED in one<br>Optical indication of operation/configurable |
| LED POWER         | Optical indication of supply                                                           |
| P                 | Push-button, which state can read via RS-485 bus.                                      |

The LED mode and internal buzzer functions are designed to warn user on state, in which buzzer actually is. Additionally, it is possible to change settings, which will compel extra reactions of indication elements. Extra reactions can be modified by means of port settings.

## 4 . Serial transmission format

In this data sheet RS-485 protocol has been confined to descriptions of commands, responses and their parameters. Header and CRC control sum exist always and are compliant with full “Netronix Prtocol” document.

Command frame:

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

Response frame:

|        |                  |                          |               |     |
|--------|------------------|--------------------------|---------------|-----|
| Header | C_CommandName +1 | Response_parametrers...m | OperationCode | CRC |
|--------|------------------|--------------------------|---------------|-----|

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

## 4.1. Key management

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

To maintain the highest level of data security, employed a particular philosophy of working with these keys.

It allows unit or person who possesses the highest level of confidence to load a key. Such loading operation can be made one time only, or very rarely.

Reader operation in given application is based on using a key not directly, but on recalling key number, to login to sector.

The result is that, in substance, key does not appear in data bus in given application.

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

Keys A and B are assigned to each sector.

Commands C\_LoadKeyToSKB and C\_LoadKeyToDKB load these keys to reader memory without information on key type (A or B).

During logging to sector, user has to input as a parameter value of 0xAA or 0xBB, if he wants, the key which is being recalled would be treated as an A or B.

### 4.1.1. Key loading into dynamic key memory

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

Command frame:

|        |                |          |     |
|--------|----------------|----------|-----|
| Header | C_LoadKeyToDKB | Key1...6 | CRC |
|--------|----------------|----------|-----|

Where:

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

Response frame:

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

#### 4.1.2. Key loading to key static memory

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

Command frame:

|        |                |                 |     |
|--------|----------------|-----------------|-----|
| Header | C_LoadKeyToSKB | Key1...6, KeyNo | CRC |
|--------|----------------|-----------------|-----|

Where:

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

Response frame:

|        |                   |  |               |     |
|--------|-------------------|--|---------------|-----|
| Header | C_LoadKeyToSKB +1 |  | OperationCode | CRC |
|--------|-------------------|--|---------------|-----|

#### 4.2. Commands for communication with transponder

##### 4.2.1. 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<br>0x01 – switching the field on |

Response frame:

|        |                         |  |               |     |
|--------|-------------------------|--|---------------|-----|
| Header | C_TurnOnAntennaPower +1 |  | OperationCode | CRC |
|--------|-------------------------|--|---------------|-----|

#### 4.2.2. 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<br>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.<br>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<br>0x70 – S70<br>0x10 – Ultra Light<br>0xdf – Des Fire |
| ID1...IDn      | Unique number of transponder                                                                                                                                                                  | ID1 – LSB,<br>IDn – MSB                                           |

#### 4.2.3. 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)<br>0x00 – 0x27 (s70)       |
| KeyType        | Key type, which is inside internal Dynamic Key Buffer.   | 0xAA – key of A type<br>0xBB – key of B type |
| DKNo           | Dynamic key number                                       | 0x00                                         |

Response frame:

|        |                   |  |               |     |
|--------|-------------------|--|---------------|-----|
| Header | C_LoginWithDKB +1 |  | OperationCode | CRC |
|--------|-------------------|--|---------------|-----|

#### 4.2.4. 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)<br>0x00 – 0x27 (s70)       |
| KeyType        | Key type, which is inside internal Static Key Buffer.    | 0xAA – key of A type<br>0xBB – key of B type |
| SKNo           | Static Key number                                        | 0x00...0x1F                                  |

Response frame:

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

#### 4.2.5. 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 |             |

#### 4.2.6. 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 |
|--------|-----------------|--|---------------|-----|

#### 4.2.7. 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 |
|--------|----------------|--|---------------|-----|

#### 4.2.8. 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 |
|--------|------------------|--|---------------|-----|

#### 4.2.9. 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   |

#### 4.2.10. 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.<br>BackupBlockNo has no influence for system operation, but user can/should make the Value copy by himself. | **Sector and block numeration |
| Value1...4     | The Value, which is written to transponder block.                                                                                                           | whichever                     |

Response frame:

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

#### 4.2.11. 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 |

#### 4.2.12. 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 |
|--------|---------------------|--|---------------|-----|

#### 4.2.13. 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 |
|--------|--------------------|--|---------------|-----|

#### 4.2.14. 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 |
|--------|----------|--|---------------|-----|

#### 4.3. Mifare Plus/ Desfire

Please contact with our support at [netronix@netronix.pl](mailto:netronix@netronix.pl)

#### 4.4. Electrical inputs and outputs

The reader has configurable inputs and outputs. All the outputs are of OC (open drain) type and feature 1 mA load capacity (1.5 A for pulse of width less than 10 ms). Module comprising outputs is fitted with overcurrent protection feature, which switches outputs off in case of excessive current occurs, and switches them on, after current drops below given limit value.

##### 4.4.1. Describing the output state

Command frame:

|        |                |             |     |
|--------|----------------|-------------|-----|
| Header | C_WriteOutputs | IONo, State | CRC |
|--------|----------------|-------------|-----|

Where:

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

Response frame:

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

##### 4.4.2. Reading-out the input state

Command frame:

|        |              |      |     |
|--------|--------------|------|-----|
| Header | C_ReadInputs | IONo | CRC |
|--------|--------------|------|-----|

Where:

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

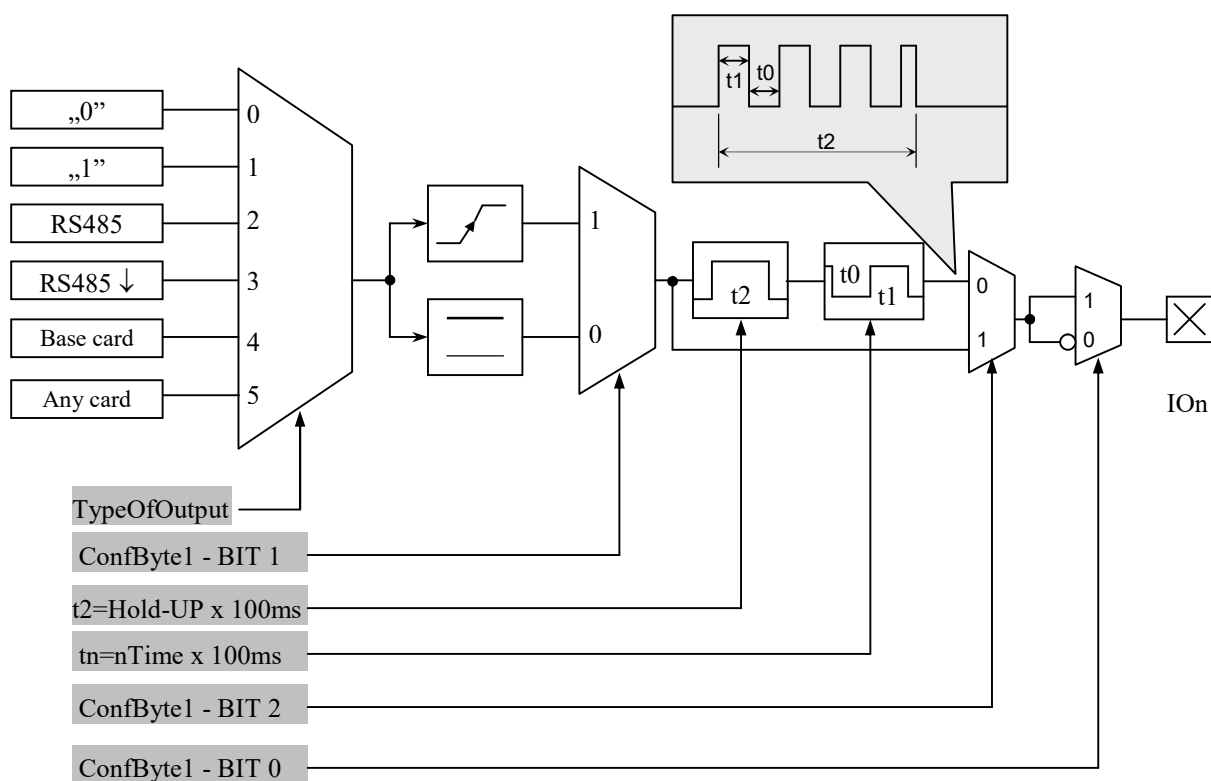
Response frame:

|        |                 |                 |               |     |
|--------|-----------------|-----------------|---------------|-----|
| Header | C_ReadInputs +1 | State,[counter] | OperationCode | CRC |
|--------|-----------------|-----------------|---------------|-----|

Where:

| Parameter name | Parameter description                   | Value range |
|----------------|-----------------------------------------|-------------|
| State          | Red-out of output state                 |             |
| Counter        | Counter state for counter type of input |             |

#### 4.4.3. 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 to any port.                                                                                                                                                                                                     | 0x50                                                                                                                       |
| IONo           | Number of I/O port, which is to be configured.                                                                                                                                                                                             | 0x02...0x06                                                                                                                |
| Dir            | Port direction                                                                                                                                                                                                                             | 0x00 – output                                                                                                              |
| ConfByte1      | One byte, in which younger byte defines output type as a NO or NC.<br>Next byte characterizes response manner of given output, as responding for actuation change (slope responding) or responding for actuation state (state responding). | ConfByte1.BIT 0<br>0-Normally Closed<br>1-Normally Open<br><br>ConfByte1.BIT 1<br>0-level responding<br>1-slope responding |

|              |                                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TypeOfOutput | Source of driving signal                                                                                                                                                                                                                                                                                          | 0x00 – permanently off<br>0x01 – permanently on<br>0x02 – driven via serial interface<br>0x03 – driven via serial with automatic reset<br>0x04 – driven by internal access control mechanism ACM.<br>This output is driven in case of applying the card to reader, which is written into internal card base.<br>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:<br><br>Hold-up x 100 ms<br><br>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 a input, IOConfigData1...n parameters would be as below:**

Dir, Neg, TypeOfInput, RFU1, RFU2, RFU3,

Where:

| Parameter name | Parameter description                                                | Value range              |
|----------------|----------------------------------------------------------------------|--------------------------|
| C_SetIOConfig  | Writing the configuration of freely selected port.                   | 0x50                     |
| IONo           | I/O port number, which is to be configured.                          | 0x00,0x01,0x07,0x08-0x0C |
| Dir            | Port direction                                                       | 0x01 – input             |
| Neg            | 0: return negation state of input<br>1: return direct state of input | 0x0,0x1                  |
| TypeOfInput    | 0x03 – normal input<br>0x04 – counter type input                     | 0x03,0x04                |
| RFU1-RFU3      | Reserved, must be set to ‘0’                                         | 0x00                     |

UW-R4G reader has no possibility to toggle port direction.

To accomplish proper configuration, input proper direction option to given port.

| LIST OF EXISTING PORTS, WHICH CAN BE DRIVEN IN UW-R4G |              |                                         |
|-------------------------------------------------------|--------------|-----------------------------------------|
| Port number                                           | Direction    | Description                             |
| 0                                                     | input        | Switch located on front panel of reader |
| 1                                                     | input/output | GPIO 1                                  |
| 2                                                     | output       | Green LED: „mode”                       |
| 3                                                     | output       | Red LED: „mode”                         |
| 4                                                     | output       | buzzer                                  |
| 5                                                     | output       | relay                                   |
| 6                                                     | output       | Blue LED “mode”                         |
| 7                                                     | input        | Tamper switch                           |
| 8                                                     | input/output | GPIO2                                   |
| 9                                                     | input/output | GPIO3                                   |
| 10                                                    | input/output | GPIO4                                   |
| 11                                                    | input/output | GPIO5                                   |
| 12                                                    | input/output | GPIO6                                   |

Response frame:

|        |                  |  |               |     |
|--------|------------------|--|---------------|-----|
| Header | C_SetIOConfig +1 |  | OperationCode | CRC |
|--------|------------------|--|---------------|-----|

#### 4.4.4. 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. |             |

## 4.5. Access password

### 4.5.1. 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.<br>String length, which can be 0 to 8 bytes |
| 0x00           | Logic Zero, which terminates a string. | 0x00                                                                     |

Response frame:

|        |                |  |               |     |
|--------|----------------|--|---------------|-----|
| Header | C_LoginUser +1 |  | OperationCode | CRC |
|--------|----------------|--|---------------|-----|

### 4.5.2. 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.<br>String length, which can be 0 to 8 bytes |
| 0x00              | Logic Zero, which terminates a string.                          | 0x00                                                                     |

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

Response frame:

|        |                     |  |               |     |
|--------|---------------------|--|---------------|-----|
| Header | C_ChangeLoginUser+1 |  | OperationCode | CRC |
|--------|---------------------|--|---------------|-----|

### 4.5.3. 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 |
|--------|-----------------|--|---------------|-----|

## 4.6. Operating the transponder internal memory

### 4.6.1. Reading-out the transponder number from memory

Command frame:

|        |                  |            |     |
|--------|------------------|------------|-----|
| Header | C_CardMemoryRead | AdrL, AdrH | CRC |
|--------|------------------|------------|-----|

Where:

| Parameter name   | Parameter description                       | Value range     |
|------------------|---------------------------------------------|-----------------|
| C_CardMemoryRead | Read-out of transponder number from memory. | 0x20            |
| AdrL, AdrH       | Younger and older byte respectively.        | 0x0000...0x01fd |

Response frame:

|        |                     |                        |               |     |
|--------|---------------------|------------------------|---------------|-----|
| Header | C_CardMemoryRead +1 | ID1(L)...ID5(H), Right | OperationCode | CRC |
|--------|---------------------|------------------------|---------------|-----|

Where:

| Parameter name  | Parameter description              | Value range |
|-----------------|------------------------------------|-------------|
| ID1(L)...ID5(H) | Five bytes of transponder number   |             |
| Right           | Access rights to given transponder | 0x01        |

#### 4.6.2. Writing the transponder name to memory

Command frame:

|        |                   |                                    |     |
|--------|-------------------|------------------------------------|-----|
| Header | C_CardMemoryWrite | AdrL, AdrH, ID1(L)...ID5(H), Right | CRC |
|--------|-------------------|------------------------------------|-----|

Where:

| Parameter name    | Parameter description                               | Value range       |
|-------------------|-----------------------------------------------------|-------------------|
| C_CardMemoryWrite | Write of transponder number into memory.            | 0x22              |
| AdrL, AdrH        | Younger and older byte respectively                 | 0x00...0x01fd     |
| ID1(L)...ID5(H)   | Five bytes of transponder number                    | Any of five bytes |
| Right             | Access rights or function performed by transponder. | 0x01              |

Response frame:

|        |                     |  |               |     |
|--------|---------------------|--|---------------|-----|
| Header | C_CardMemoryWrite+1 |  | OperationCode | CRC |
|--------|---------------------|--|---------------|-----|

Where:

#### 4.7. Operating the built-in access control

##### 4.7.1. Writing the configuration of access control

Command frame:

|        |                           |      |     |
|--------|---------------------------|------|-----|
| Header | C_AccesControlConfigWrite | Mode | CRC |
|--------|---------------------------|------|-----|

Where:

| Parameter name            | Parameter description                    | Value range                                     |
|---------------------------|------------------------------------------|-------------------------------------------------|
| C_AccesControlConfigWrite | Write of access control configuration.   | 0x74                                            |
| Mode                      | Operation mode of control access module. | 0x00 – module disabled<br>0x01 – module enabled |

Response frame:

|        |                             |  |               |     |
|--------|-----------------------------|--|---------------|-----|
| Header | C_AccesControlConfigWrite+1 |  | OperationCode | CRC |
|--------|-----------------------------|--|---------------|-----|

Where:

##### 4.7.2. Reading-out the configuration of access control

Command frame:

|        |                          |  |     |
|--------|--------------------------|--|-----|
| Header | C_AccesControlConfigRead |  | CRC |
|--------|--------------------------|--|-----|

Where:

| Parameter name           | Parameter description                     | Value range |
|--------------------------|-------------------------------------------|-------------|
| C_AccesControlConfigRead | Read-out of access control configuration. | 0x76        |
|                          |                                           |             |

Response frame:

|        |                            |      |               |     |
|--------|----------------------------|------|---------------|-----|
| Header | C_AccesControlConfigRead+1 | Mode | OperationCode | CRC |
|--------|----------------------------|------|---------------|-----|

Where:

| Parameter name | Parameter description                    | Value range                                     |
|----------------|------------------------------------------|-------------------------------------------------|
| Mode           | Operation mode of access control module. | 0x00 – module disabled<br>0x01 – module enabled |

#### 4.7.3. Writing the automatic device configuration

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

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

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

If the reader will operate in mixed mode i.e.:

- automatic reading device UID is enabled and:
- master device (computer, controller) communicates with reader or with transponders via reader,

it is required, to configure the reader correctly, so in case of communication with a reader or transponder, automatic reading device would hold-on its operation.

Command frame:

|        |                       |                                                   |     |
|--------|-----------------------|---------------------------------------------------|-----|
| Header | C_SetAutoReaderConfig | ATrig, AOfflineTime, ASerial, AMode, Abuzz, Amask | 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<br>1-automatic device enabled permanently<br>2=enabled automatically in case of transmission lack on RS485 for a time longer than AOfflineTime<br>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<br>$T = AofflineTime * [100ms]$<br>Lack of transmission can concern to | 0x00...0xff                                                                                                                                                                                                                                                                                                            |

|         |                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                |                                                                                          |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
|         | <div>any commands (Atrig=2), or<br/>commands for communication with<br/>transponder (Atrig=3).</div> <div>Commands for communication with<br/>transponder:<br/>C_TurnOnAntennaPower<br/>C_Select<br/>C_LoginWithDKB<br/>C_LoginWithSKB)<br/>C_ReadBlock<br/>C_WriteBlock<br/>C_CopyBlock<br/>C_WritePage4B<br/>C_ReadPage16B<br/>C_IncrementValue<br/>C_DecrementValue<br/>C_WriteValue<br/>C_ReadValue<br/>C_Halt</div> |                                                                                |                                                                                          |
| ASerial | Automatic sending the UID<br>transponder number, after reading it<br>automatically from transponder.                                                                                                                                                                                                                                                                                                                     | 0-never<br>1-for the first applying the<br>transponder only<br>2-sends all     |                                                                                          |
| AMode   | <div>Selection the format of sending<br/>number</div> <div>8 bits:</div> <div><div>MSB</div><div><div><div>R</div><div>R</div><div>R</div><div>CR</div><div>M</div><div>E</div><div>I</div><div>A</div></div></div><div>LSB</div></div>                                                                                                                                                                                  | R                                                                              | Reserved, always 0                                                                       |
|         |                                                                                                                                                                                                                                                                                                                                                                                                                          | CR=1                                                                           | Number which is ended with<br>line end mark CR+LF                                        |
|         |                                                                                                                                                                                                                                                                                                                                                                                                                          | M=1                                                                            | Number which begins with<br>“M”sign                                                      |
|         |                                                                                                                                                                                                                                                                                                                                                                                                                          | E=1                                                                            | information extended with cards<br>umber in filed and card type<br>(UW-M4x readers only) |
|         |                                                                                                                                                                                                                                                                                                                                                                                                                          | I=1                                                                            | Number in reversed order                                                                 |
|         |                                                                                                                                                                                                                                                                                                                                                                                                                          | A=1                                                                            | Number sent in ASCII format                                                              |
|         |                                                                                                                                                                                                                                                                                                                                                                                                                          | A=0                                                                            | Number sent in Nertonix format                                                           |
| ABuzz   | Automatic indication of reading by<br>means of buzzer, after automatic UID<br>read-out from transponder.                                                                                                                                                                                                                                                                                                                 | 0-never<br>1-for the first applying the<br>transponder only<br>2-indicates all |                                                                                          |
| AMask   | <div>Defines which transponder will be<br/>read:</div> <div><div>MSB</div><div><div><div>R</div><div>R</div><div>R</div><div>R</div><div>S</div><div>I</div><div>B</div><div>M</div></div></div><div>LSB</div></div>                                                                                                                                                                                                     | R                                                                              | Reserved, always 0                                                                       |
|         |                                                                                                                                                                                                                                                                                                                                                                                                                          | M=1                                                                            | Mifare (ISO14443A)                                                                       |
|         |                                                                                                                                                                                                                                                                                                                                                                                                                          | B=1                                                                            | ISO14443B                                                                                |
|         |                                                                                                                                                                                                                                                                                                                                                                                                                          | I=1                                                                            | IClass CSN                                                                               |
|         |                                                                                                                                                                                                                                                                                                                                                                                                                          | S=1                                                                            | ICode SLI (ISO15693)                                                                     |

Response frame:

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

#### 4.7.4. 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.

#### 4.7.5. Setting the date and time

Following setting has no influence for reader operation today.

Command frame:

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

Where:

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

Response frame:

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

#### 4.7.6. 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  | 0xb6        |

|  |      |  |
|--|------|--|
|  | time |  |
|--|------|--|

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.

## 4.8. Configuring the RS-485 serial interface

### 4.8.1. 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<br>0x02=4800 bps<br>0x03=9600 bps<br>0x04=19200 bps<br>0x05=38400 bps<br>0x06=57600 bps<br>0x07=115200 bps |

Response:

|                         |  |               |  |
|-------------------------|--|---------------|--|
| C_SetInterfaceConfig +1 |  | OperationCode |  |
|-------------------------|--|---------------|--|

### 4.8.2. 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.

## 4.9. Managing the events

The UW-x4x series readers has equipped with event memory of capacity 4400 records. Reason of event can be operation related to card or state changing on reader outputs. The readers have RTC clock with battery back-up. After supply failure above 7 days, clock is reset to defaults: date: 1 January 2019, time: 15:01:01. Event counter is reset.

### 4.9.1. Setting the event recorder

Command frame:

|        |                |                                              |     |
|--------|----------------|----------------------------------------------|-----|
| Header | C_SetEventTrig | CardTrig, In4Trig, In3Trig, In2Trig, In1Trig | CRC |
|--------|----------------|----------------------------------------------|-----|

Wherein:

| Parameter name         | Parameter description                               | Value range |
|------------------------|-----------------------------------------------------|-------------|
| C_SetEventTrig<br>0x7C | Setting the event masking                           | 0x7C        |
| CardTrig               | Masking the events related to card<br>(see below)   | 0x00 - 0xFF |
| In1Trig-In4Trig        | Masking the events related to inputs<br>(see below) | 0x00-0xFF   |

Response frame:

|        |                  |  |               |     |
|--------|------------------|--|---------------|-----|
| Header | C_SetEventTrig+1 |  | OperationCode | CRC |
|--------|------------------|--|---------------|-----|

### Masking byte of events related to card

| Bit 8   | Bit 7     | Bit 6        | Bit 5       | Bit 4   | Bit 3       | Bit 2             | Bit 1            |
|---------|-----------|--------------|-------------|---------|-------------|-------------------|------------------|
| Reserve | No memory | Card removal | Card adding | Reserve | Master card | Outside base card | Inside base card |

E.g. byte 0x25(00100101) means that events will be written in case of:

- inside base card has been red-out,
- card written as master has been written,
- inside base card has been removed

### Masking bytes related to inputs state change

| Byte    | Bit 8   | Bit 7   | Bit 6   | Bit 5   | Bit 4   | Bit 3   | Bit 2   | Bit 1   |
|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| In1Trig | IO[3]F  | IO[3]R  | IO[2]F  | IO[2]R  | IO[1]F  | IO[1]R  | IO[0]F  | IO[0]R  |
| In2Trig | IO[7]F  | IO[7]R  | IO[6]F  | IO[6]R  | IO[5]F  | IO[5]R  | IO[4]F  | IO[4]R  |
| In3Trig | IO[11]F | IO[11]R | IO[10]F | IO[10]R | IO[9]F  | IO[9]R  | IO[8]F  | IO[8]R  |
| In4Trig | IO[15]F | IO[15]R | IO[14]F | IO[14]R | IO[13]F | IO[13]R | IO[12]F | IO[12]R |

Bits IO[n]R denote reaction to the input **n** positive slope,

Bits IO[n]F denote reaction to the input **n** negative slope

*E.g. In4Trig-In1Trig configuration byte sequence: 0x00,0x31,0x40,0x08, causes, that events will be written in case of:*

- Any state change of input with index 10 occurs
- Positive slope appears on input with index 8
- Positive slope appears on input with index 7
- Negative slope appears on input with index 1

During configuring the event triggers, decide which port is configured as an input. Do not configure events for those I/O's, which are outputs.

To guarantee correctness of event write process, time between two subsequent triggers must be longer than 20 ms.

#### 4.9.2. Reading the event recorder

Command frame:

|        |                |  |     |
|--------|----------------|--|-----|
| Header | C_GetEventTrig |  | CRC |
|--------|----------------|--|-----|

Wherein:

| Parameter name         | Parameter description                       | Value range |
|------------------------|---------------------------------------------|-------------|
| C_GetEventTrig<br>0x7E | Reading the configuration of event recorder | 0x7E        |

Response frame:

|        |                  |                                              |               |     |
|--------|------------------|----------------------------------------------|---------------|-----|
| Header | C_GetEventTrig+1 | CardTrig, In4Trig, In3Trig, In2Trig, In1Trig | OperationCode | CRC |
|--------|------------------|----------------------------------------------|---------------|-----|

Response bytes (CardTrig, In4Trig, In3Trig, In2Trig, In1Trig) are related to bytes from point 10.1.

#### 4.9.3. Reading the counters related to event memory.

Command frame:

|        |                 |  |     |
|--------|-----------------|--|-----|
| Header | C_GetEventParam |  | CRC |
|--------|-----------------|--|-----|

Wherein:

| Parameter name          | Parameter description                       | Value range |
|-------------------------|---------------------------------------------|-------------|
| C_GetEventParam<br>0x78 | Reading the configuration of event recorder | 0x78        |

Response frame:

|        |                   |                                                            |               |     |
|--------|-------------------|------------------------------------------------------------|---------------|-----|
| Header | C_GetEventParam+1 | CapL, CapH, PointerL, PointerH, TotB3, TotB2, TotB1, TotB0 | OperationCode | CRC |
|--------|-------------------|------------------------------------------------------------|---------------|-----|

CapH:CapL – two-byte value, which defines event memory capacity.

PointerH:PointerL – two-byte value, which marks from first free event.

TotB3:TotB2:TotB1:TotB0 – four-byte value, which defines number of events recorded from the moment of counter reset.

Events are recorded in sequence from index 0 up to Cap-1. In the moment memory gets full, the counter is being “overturned”, and older inputs are overwritten.

*Example:*

*If using C\_GetEventParam command, we have read that event memory capacity is 4400 inputs; the total value of input events is 5678. For instance, if we want to read the event with no. 5600, event index event of interest will be  $5678-4400=1278$ .*

*If we want to read the last event, we can use Pointer value. The last event index will be Pointer-1.*

#### 4.9.4. Reading the events

Command frame:

|        |            |              |     |
|--------|------------|--------------|-----|
| Header | C_GetEvent | EvNoL, EvNoH | CRC |
|--------|------------|--------------|-----|

Wherein:

| Parameter name     | Parameter description      | Value range |
|--------------------|----------------------------|-------------|
| C_GetEvent<br>0x7a | Reading the event          | 0x7a        |
| EvNoL,EvNoH        | LSB and MSB of event index |             |

Response frame:

|        |              |                                        |                   |     |
|--------|--------------|----------------------------------------|-------------------|-----|
| Header | C_GetEvent+1 | YY,MM,DD, hh,mm,ss,type,B1,B2,B3,B4,B5 | OperationC<br>ode | CRC |
|--------|--------------|----------------------------------------|-------------------|-----|

YY,MM,DD – year, month, day of event occurrence

hh,mm,ss – hour, minute, second of event occurrence

type – event type

Depending on value “type” the 8-th bit of byte, there are distinguished two assignments:

| Bit8      | Bit7      | Bit6     | Bit5     | Bit4     | Bit3   | Bit2         | Bit1        |
|-----------|-----------|----------|----------|----------|--------|--------------|-------------|
| 0 – card  | No memory | Removed  | Added    | reserved | Master | Outside base | Inside base |
| 1 -inputs | reserved  | reserved | reserved | N4       | N2     | N1           | N0          |

N4:N0 –number of input, on which event trigger signal appeared.

- If given event was related to card, the B1-B5 bytes comprise card ID number.

| B1   | B2   | B3   | B4   | B5            |
|------|------|------|------|---------------|
| UID1 | UID2 | UID3 | UID4 | UID5 (Unique) |

- If event is related to input change, B1-B5 bytes comprise information regarding input state, and have format:

| B1  |     |     |     | B2  |     |     |     | B3  |     |      |      | B4   |      |      |      | B5       |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|------|----------|
| IO0 | IO1 | IO2 | IO3 | IO4 | IO5 | IO6 | IO7 | IO8 | IO9 | IO10 | IO11 | IO12 | IO13 | IO14 | IO15 | Reserved |

## 4.10. MAD – Mifare Application Directory

### 4.10.1. Card MAD formatting

Command frame:

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

Wherein:

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

Response frame:

|        |               |  |               |     |
|--------|---------------|--|---------------|-----|
| Header | C_FormatMad+1 |  | OperationCode | CRC |
|--------|---------------|--|---------------|-----|

Notes:

Before you run C\_FormatMad command:

- switch AutoReader mode off (using C\_SetAutoReaderConfig command)
- load the keys (default 0xff,0xff,0xff,0xff,0xff,0xff)
- turn antenna supply on (using C\_TurnOnAntennaPower)
- select the cart (using C\_Select command)
- login to sector with number 0, using key of AA type

### 4.10.2. Adding the application to MAD directory

Command frame:

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

Wherein:

| Parameter name                  | Parameter description                                       | Value range                        |
|---------------------------------|-------------------------------------------------------------|------------------------------------|
| C_AddApplication<br><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<br>0x01-0x1F :MAD2 |

Response frame:

|        |                    |  |               |     |
|--------|--------------------|--|---------------|-----|
| Header | C_AddApplication+1 |  | OperationCode | CRC |
|--------|--------------------|--|---------------|-----|

Notes:

Application number should be other than 0x0000

Before you run C\_AddApplication command:

- switch AutoReader mode off (using command C\_SetAutoReaderConfig)
- load the keys (default 0xff,0xff,0xff,0xff,0xff,0xff)

- turn antenna supply on (using C\_TurnOnAntennaPower command)
- select the card (using C\_Select command)
- login to sector with number 0, using key of AA type

#### 4.10.3. Pursuing the sector for given application

Command frame:

|        |                |          |     |
|--------|----------------|----------|-----|
| Header | C_GetSectorMad | LSB, MSB | CRC |
|--------|----------------|----------|-----|

Wherein:

| Parameter name         | Parameter description     | Value range |
|------------------------|---------------------------|-------------|
| C_GetSectorMad<br>0xAC | Pursuing the sector       | 0xAC        |
| LSB                    | LSB of application number | 0x00 - 0xFF |
| MSB                    | MSB of application number | 0x00 - 0xFF |

Response frame:

|        |                  |        |               |     |
|--------|------------------|--------|---------------|-----|
| Header | C_GetSectorMad+1 | Sector | OperationCode | CRC |
|--------|------------------|--------|---------------|-----|

Notes:

Before you run C\_GetSectorMad command:

- switch AutoReader mode off (using C\_SetAutoReaderConfig command)
- load the keys (using 0xff,0xff,0xff,0xff,0xff,0xff)
- turn antenna supply on (using C\_TurnOnAntennaPower command)
- select the card (using C\_Select command)
- login to sector with number 0, using key of AA type

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

#### 4.10.4. Pursuing the next sector of application

Command frame:

|        |                    |          |     |
|--------|--------------------|----------|-----|
| Header | C_GetSectorMadNext | LSB, MSB | CRC |
|--------|--------------------|----------|-----|

Wherein:

| Parameter name         | Parameter description    | Value range |
|------------------------|--------------------------|-------------|
| C_GetSectorMad<br>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.

## 4.11. Other commands

### 4.11.1. 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 |
|--------|------------|--|-------------|-----|

### 4.11.2. Reading-out the reader software

Command frame:

|        |                   |  |     |
|--------|-------------------|--|-----|
| Header | C_FirmwareVersion |  | CRC |
|--------|-------------------|--|-----|

Where:

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

Response frame:

|        |                     |             |               |     |
|--------|---------------------|-------------|---------------|-----|
| Header | C_FirmwareVersion+1 | Data1.....n | OperationCode | CRC |
|--------|---------------------|-------------|---------------|-----|

Where:

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

### 4.11.3. Setting buzzer volume

Command frame:

|        |                   |        |     |
|--------|-------------------|--------|-----|
| Header | C_SetBuzzerConfig | volume | CRC |
|--------|-------------------|--------|-----|

Gdzie:

| Parameter name    | Parameter description      | Value range |
|-------------------|----------------------------|-------------|
| C_SetBuzzerConfig | Settings buzzer parameters | 0xD8        |
| volume            | volume                     | 0 - 0xA     |

Response frame:

|        |                      |  |               |     |
|--------|----------------------|--|---------------|-----|
| Header | C_SetBuzzerConfig +1 |  | OperationCode | CRC |
|--------|----------------------|--|---------------|-----|

### 4.11.4. Reading buzzer volume

Command frame:

|        |                   |  |     |
|--------|-------------------|--|-----|
| Header | C_GetBuzzerConfig |  | CRC |
|--------|-------------------|--|-----|

Where:

| Parameter name    | Parameter description     | Value range |
|-------------------|---------------------------|-------------|
| C_GetBuzzerConfig | Reading buzzer parameters | 0xDC        |

Response frame:

|        |                      |        |               |     |
|--------|----------------------|--------|---------------|-----|
| Header | C_GetBuzzerConfig +1 | volume | OperationCode | CRC |
|--------|----------------------|--------|---------------|-----|

Where,

Volume: buzzer volume In range 0-10

#### 4.12. Meaning of operation code in response frame

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

## 5. MODBUS RTU protocol

### 5.1. Supported MODBUS protocol functions:

|      |                               |
|------|-------------------------------|
| 0x01 | Read coils                    |
| 0x03 | Read holding register         |
| 0x05 | Write single coil             |
| 0x06 | Write single register         |
| 0x17 | Write read multiple registers |

### 5.2. MODBUS address (decimal)

| Lp  | Address | Type        | R/W | Description                                                                  |
|-----|---------|-------------|-----|------------------------------------------------------------------------------|
| 1   | 1000    | Holding Reg | R   | Transpoder code [0]                                                          |
| 2   | 1001    | Holding Reg | R   | Transpoder code [1]                                                          |
| 3   | 1002    | Holding Reg | R   | Transpoder code [2]                                                          |
| 4   | 1003    | Holding Reg | R   | Transpoder code [3]                                                          |
| 4.1 | 1004    | Holding Reg | R   | Transpoder code [4]                                                          |
| 4.2 | 1005    | Holding Reg | R   | Transpoder code [5]                                                          |
| 4.3 | 1006    | Holding Reg | R   | Transpoder code [6]                                                          |
| 4.4 | 1007    | Holding Reg | R   | Transpoder code [7]                                                          |
| 6   | 1011    | Holding Reg | R/W | Relay mode<br>0 – inactive,<br>1 – bi-stable<br>2 – a-stable<br>3 – 1 impuls |
| 7   | 1012    | Holding Reg | R/W | Relay ON time (*100ms, max 255)                                              |
| 8   | 1013    | Holding Reg | R/W | Relay OFF time (*100ms, max 255)                                             |
| 9   | 1014    | Holding Reg | R/W | RED LED mode (see p.6)                                                       |
| 10  | 1015    | Holding Reg | R/W | GREEN LED mode (see p.6)                                                     |
| 11  | 1016    | Holding Reg | R/W | BLUE LED mode (see p.6)                                                      |
| 12  | 1017    | Holding Reg | R/W | BUZZER mode (see p.6)                                                        |
| 13  | 1020    | Holding Reg | R/W | On-time of RED LED (*100ms)                                                  |
| 14  | 1021    | Holding Reg | R/W | Off-time of RED LED (*100ms)                                                 |
| 15  | 1022    | Holding Reg | R/W | On-time of GREEN LED (*100ms)                                                |
| 16  | 1023    | Holding Reg | R/W | Off-time of GREEN LED (*100ms)                                               |
| 17  | 1024    | Holding Reg | R/W | On-time of BLUE LED (*100ms)                                                 |
| 18  | 1025    | Holding Reg | R/W | Off-time of BLUE LED (*100ms)                                                |
| 19  | 1026    | Holding Reg | R/W | On-time of buzzer (*100ms)                                                   |
| 20  | 1027    | Holding Reg | R/W | Off-time of buzzer(*100ms)                                                   |
| 21  | 1028    | Holding Reg | R/W | Mode of IO1 (jak w p.6)                                                      |
| 22  | 1029    | Holding Reg | R/W | Mode of IO2 (jak w p.6)                                                      |
| 23  | 1030    | Holding Reg | R/W | Mode of IO3 (jak w p.6)                                                      |
| 24  | 1031    | Holding Reg | R/W | Mode of IO4 (jak w p.6)                                                      |
| 25  | 1032    | Holding Reg | R/W | Mode of IO5 (jak w p.6)                                                      |
| 26  | 1033    | Holding Reg | R/W | Mode of IO6 (jak w p.6)                                                      |

|    |      |             |     |                                                                                                                                       |
|----|------|-------------|-----|---------------------------------------------------------------------------------------------------------------------------------------|
| 27 | 1034 | Holding Reg | R/W | On-time of IO1 (*100ms)                                                                                                               |
| 28 | 1035 | Holding Reg | R/W | Off-time of IO1 (*100ms)                                                                                                              |
| 29 | 1036 | Holding Reg | R/W | On-time of IO2 (*100ms)                                                                                                               |
| 30 | 1037 | Holding Reg | R/W | Off-time of IO2 (*100ms)                                                                                                              |
| 31 | 1038 | Holding Reg | R/W | On-time of IO3 (*100ms)                                                                                                               |
| 32 | 1039 | Holding Reg | R/W | Off-time of IO3 (*100ms)                                                                                                              |
| 33 | 1040 | Holding Reg | R/W | On-time of IO4 (*100ms)                                                                                                               |
| 34 | 1041 | Holding Reg | R/W | Off-time of IO4 (*100ms)                                                                                                              |
| 35 | 1042 | Holding Reg | R/W | On-time of IO5 (*100ms)                                                                                                               |
| 36 | 1043 | Holding Reg | R/W | Off-time of IO5 (*100ms)                                                                                                              |
| 37 | 1044 | Holding Reg | R/W | On-time of IO6 (*100ms)                                                                                                               |
| 38 | 1045 | Holding Reg | R/W | Off-time of IO6 (*100ms)                                                                                                              |
| 39 | 1050 | Holding Reg | R/W | RS485 address                                                                                                                         |
| 40 | 1051 | Holding Reg | R   | Firmware version                                                                                                                      |
| 41 | 1000 | Single Coil | W   | Enable output (relay)                                                                                                                 |
| 42 | 1001 | Single Coil | W   | Enable RED LED                                                                                                                        |
| 43 | 1003 | Single Coil | R   | Front switch state                                                                                                                    |
| 44 | 1004 | Single Coil | R/W | New card state*<br>Reading: 1-new transponder detect<br>Writing: 0 – clear this flag<br>*flag is clearing aquatically after 6 seconds |
| 45 | 1010 | Single Coil | W   | Enable GREEN LED                                                                                                                      |
| 46 | 1011 | Single Coil | W   | Enable BLUE LED                                                                                                                       |
| 47 | 1012 | Single Coil | W   | Enable Buzzer                                                                                                                         |
| 48 | 1020 | Single Coil | R/W | W- Enable IO1, R- read state of IO1                                                                                                   |
| 49 | 1021 | Single Coil | R/W | W- Enable IO2, R- read state of IO2                                                                                                   |
| 50 | 1022 | Single Coil | R/W | W- Enable IO3, R- read state of IO3                                                                                                   |
| 51 | 1023 | Single Coil | R/W | W- Enable IO4, R- read state of IO4                                                                                                   |
| 52 | 1024 | Single Coil | R/W | W- Enable IO5, R- read state of IO5                                                                                                   |
| 53 | 1025 | Single Coil | R/W | W- Enable IO6, R- read state of IO6                                                                                                   |

### 5.3. Encapsulation Netronix protocol inside MODBUS RTU

Each frame from Netronix protocol can be sent using below schematic:

MODBUS frame for function 0x17:

#### Request

|                        |                      |                  |
|------------------------|----------------------|------------------|
| Function code          | 1 Byte               | <b>0x17</b>      |
| Read Starting Address  | 2 Bytes              | 0x0000 to 0xFFFF |
| Quantity to Read       | 2 Bytes              | 0x0001 to 0x007D |
| Write Starting Address | 2 Bytes              | 0x0000 to 0xFFFF |
| Quantity to Write      | 2 Bytes              | 0x0001 to 0x0079 |
| Write Byte Count       | 1 Byte               | 2 x <b>N</b> *   |
| Write Registers Value  | <b>N</b> * x 2 Bytes |                  |

\***N** = Quantity to Write

#### Response

|                      |                       |                 |
|----------------------|-----------------------|-----------------|
| Function code        | 1 Byte                | <b>0x17</b>     |
| Byte Count           | 1 Byte                | 2 x <b>N'</b> * |
| Read Registers value | <b>N'</b> * x 2 Bytes |                 |

\***N'** = Quantity to Read

|                        |                      |
|------------------------|----------------------|
| Read Starting Address  | Function code        |
| Quantity to Read       | -                    |
| Write Starting Address | Always 0x0000        |
| Quantity to Write      | -                    |
| Write Byte Count       | Length of parameters |
| Write Registers Value  | Parameters           |

#### 5.3.1. Example of using 0x17 function

To send Netronix C\_ReadBlock 0x01 command, below frame must be sent:

|                           |      |
|---------------------------|------|
| Address                   | 0x01 |
| Command                   | 0x17 |
| ReadStartingAddress Hi    | 0x00 |
| ReadStartingAddress Lo    | 0x1e |
| Quantity to Read Hi       | 0x00 |
| Quantity to Read Lo       | 0x00 |
| Write Starting Address Hi | 0x00 |
| Write Starting Address Lo | 0x00 |
| Quantity to Write Hi      | 0x00 |
| Quantity to Write Lo      | 0x00 |
| Write Byte Count          | 0x01 |
| Write Register Value Hi   | 0x00 |
| Write Register Value Lo   | 0x01 |
| CRC Lo                    | 0xC5 |
| CRC Hu                    | 0x3E |

## 6 . Meaning of symbols and markings used in the specification

**\*\*Sectors and block numeration**

For S50 cards:

SectorNo=0x00...0x0f

BlockNo=0x00...0x03

For S70 cards:

SectorNo=0x00...0x20    BlockNo=0x00...0x03

SectorNo=0x21...0x27    BlockNo=0x00...0x0f

## 7 . Mechanism of Master ID

Master ID mechanism is based on principle the quick adding/removing of user card to/out of reader memory by means of „master card”.

**If you want to register a card as a „master card”,** it is required to clear card memory first by means of reset function to factory defaults.

After clearing the memory, apply selected card to module, whenever you like. This moment, the card becomes “master card”. It is impossible to remove or add the master card by means of other card.

**If you want to register a card as a “user card”,** apply “master card” to reader first, and next during five seconds, apply registered card.

**If you want to remove “user card” from memory,** apply “master card” to reader first, and next during five seconds apply card which is being removed.

After applying to a reader the “user card”, the reader enables electric output, which has been programmed as a controlled by internal access control mechanism.

## 8 . Clearing the card memory and resetting to factory defaults

To reset the device to factory defaults, push for ca. 5 seconds button “F” on rear panel. During reset to factory defaults, following parameters of reader are fixed:

| Name of parameter or its functionality                 | Value or setting                                   |
|--------------------------------------------------------|----------------------------------------------------|
| Address on serial bus                                  | 0x01                                               |
| Baud rate on serial bus                                | 9600 bps                                           |
| Whole internal memory of transponders with Master card | 0xff ff ff ff ff means “memory cleared”            |
| Access password                                        | 0x31 32 33 34 00 means „1234” in dot transcription |
| Port 0 – front switch                                  | every purpose                                      |
| Port 1 – two-state input/output                        | every purpose input                                |
| Port 2 – green LED                                     | lock “on” indication                               |
| Port 3 – red LED                                       | lock “on” indication                               |
| Port 4 – buzzer                                        | lock “on” indication                               |
| Port 5 – relay                                         | lock “on” indication                               |
| Port 6 – blue LED                                      | every purpose                                      |
| Port 7 - tamper                                        | Input                                              |
| Port 8 – IO2                                           | every purpose input                                |
| Port 9 – IO3                                           | every purpose input                                |
| Port 10 – IO4                                          | every purpose input                                |
| Port 11– IO5                                           | every purpose input                                |

|               |                               |
|---------------|-------------------------------|
| Port 12 - IO6 | every purpose input           |
| Master card   | no Master card in card memory |

## 9 . Operation example of transponder

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

Following operation assumes, that reader is in default condition, and applied S50 card is in default condition too. It means this card has full access rights and both 0xff ff ff ff ff keys.

Logging to the reader is to make changes in its factory configuration.

C\_LoginUser, 0x31, 0x32, 0x33, 0x34, 0x00

Because during manual experiments, time between subsequent commands sent via serial interface is large and reaches values from some second to some minutes, it is required to disable internal UID automatic read-out device.

It should be done by means of command:

SetAutoReaderConfig with parameters: 0x00, 0x00, 0x00, 0x00.

To read-out the transponder, first load key to key memory.

So load the key to SKB, by means of:

C\_LoadKeyToSKB, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00

Enable the field.

TurnOnAntennaPower, 0x01

Apply transponder to reader.

Select transponder

C\_Select, 0x00

Login to e.g. sector 3.

C\_LoginWithSKB, 0x03, 0xAA, 0x00

Read-out 2nd block content in 3rd sector.

C\_ReadBlock, 0x02

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

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