

OBSHCHESTVO S OGRANICHENNOY OTVETSTVENNOST'YU  
(LIMITED LIABILITY COMPANY)  
*INTERNET VESHCHEY*

**APPROVED BY**

Chief technical officer of  
OOO *INTERNET VESHCHEY*

\_\_\_\_\_A.F. Vasilenko

\_\_\_\_\_201\_\_

date

**COMMUNICATION PROTOCOL**  
**for devices fitted with**  
**the UMKa200 Reader (version 2.2)**

**DESIGNED BY**

Electronics engineer of  
OOO *INTERNET VESHCHEY*

\_\_\_\_\_V.A. Verchenko

*Krasnodar*

## General information

The UMK-200 reader operates in a “Slave” mode using command requests and has two communication channels with the “Master” device: RS-485 and USB (detected in the system as a COM port) with a common protocol. In addition, the UMK-200 device supports the ADM20 protocol commands (Appendix B).

The protocol is valid for the UMK-200 firmware versions from 0.4.4 and later.

## General packet structure

The packet structure is based on the binary protocol of the following structure:

|                   |   |                        |               |
|-------------------|---|------------------------|---------------|
| ADDRESS<br>1 byte | LENGTH<br>of the “Data” field<br>1 byte | DATA<br>Max. 255 bytes | CRC<br>1 byte |
|-------------------|---|------------------------|---------------|

Where:

ADDRESS – is a device address on the RS-485 bus;

LENGTH – is the length of the “DATA” field;

DATA – is a field of the command/subcommand, protocol and parameters identifier, see below;

CRC – is a checksum generated by the XOR operation gating of all the bytes for the CRC field (a C code example is given in Appendix A).

UMKa200 protocol commands and their examples are given in Appendix B.

### “DATA” field format for UMKa200

Bytes are placed in the big-endian order. The UMKa200 protocol identifier distinguishing feature is a 1-byte prefix “200” (0xC8) in the “DATA” field, which is followed by command code, subcommand code, and data.

#### a. “DATA” field format for UMKa200, request:

|                                |                     |                     |              |
|--------------------------------|---------------------|---------------------|--------------|
| Protocol<br>prefix<br>(1 byte) | Command<br>(1 byte) | Command<br>(1 byte) | Command data |
|--------------------------------|---------------------|---------------------|--------------|

Where:

Protocol prefix – is a binary prefix definitive for the UMKa200 protocol; for UMKa200 protocol, it is “0xC8”;

Command – is the binary command code. It can take on a value within the range of 0x00 to 0x7F;

Subcommand – is the subcommand code taking on a value within the range of 0x00 to 0xFF;

Command data – is the payload (updates and such).

**b. “DATA” field format for UMKa200, response:**

|                             |                     |                     |               |
|-----------------------------|---------------------|---------------------|---------------|
| Protocol prefix<br>(1 byte) | Command<br>(1 byte) | Command<br>(1 byte) | Response data |
|-----------------------------|---------------------|---------------------|---------------|

Where:

Protocol prefix – is a binary prefix definitive for the UMKa200 protocol; for UMKa200 protocol, it is “0xC8”;

Command – is the binary command response code. It can take on a value within the range of 0x00 to 0xFF. Command – the binary command response code. It can take on a value within the range of 0x00 to 0xFF. Command response attribute is the 7th bit set to “TRUE”, the rest of the bit field takes the value of the respective field in the request;

Subcommand – is the subcommand code taking on a value within the range of 0x00 to 0xFF. Takes the value of the respective field in the request;

Command data – is the payload.

**“DATA” field format for the supported ADM20 commands**

A distinguishing feature of the ADM20 protocol commands is the use of ASCII command code when generating a request. For example, the cleanup command for the tag table “TC”:

Request:

| Address | Length of the data field | ASCII command code "T" | ASCII command code "C" | CRC checksum |
|---------|--------------------------|------------------------|------------------------|--------------|
| 0x01    | 0x02                     | 0x54                   | 0x43                   | 0x14         |

Response:

| Address | Length of the data field | ASCII command code "T" | ASCII command code "C" | ASCII command code "K" | CRC checksum |
|---------|--------------------------|------------------------|------------------------|------------------------|--------------|
| 0x01    | 0x03                     | 0x41                   | 0x43                   | 0x4B                   | 0x4B         |

You can find a more detailed description of the ADM20 protocol and commands on the manufacturer's website. The list of supported ADM20 protocol commands is given in Appendix B.

## Appendix A. An example of a C code CRC calculation

```
/* CRC Calculation */
uint8_t calcCRCadm20(const uint8_t *message, size_t size)
{
    uint8_t calcCRC = 0 ;
    for (uint8_t i = 0 ; i < size ; i++)
    {
        calcCRC = calcCRC ^ message[i];
    }
    return calcCRC;
}
```

## Appendix B. List of supported ADM20 protocol commands

| No   | Command        | Description with examples  |                                    |                      |                      |                             |                             |             |             |                  |  |  |   |   |                                      |            |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |
|--|----------------|--|------------------------------------|----------------------|----------------------|-----------------------------|-----------------------------|-------------|-------------|------------------|--|--|---|---|--------------------------------------|------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| 1  | TC             | Tag table cleanup command. The response is always “ASK”.   |                                    |                      |                      |                             |                             |             |             |                  |  |  |   |   |                                      |            |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |
|  |                | Request example:   |                                    |                      |                      |                             |                             |             |             |                  |  |  |   |   |                                      |            |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |
|  |                | <table border="1"> <thead> <tr> <th><i>Device address</i></th> <th><i>Length</i></th> <th><i>ASCII<br/>“T”</i></th> <th><i>ASCII<br/>“C”</i></th> <th><i>CRC</i></th> </tr> </thead> <tbody> <tr> <td><i>0x01</i></td> <td><i>0x02</i></td> <td><i>0x54</i></td> <td><i>0x43</i></td> <td><i>0x14</i></td> </tr> </tbody> </table> | <i>Device address</i>              | <i>Length</i>        | <i>ASCII<br/>“T”</i> | <i>ASCII<br/>“C”</i>        | <i>CRC</i>                  | <i>0x01</i> | <i>0x02</i> | <i>0x54</i>      | <i>0x43</i>                                  | <i>0x14</i>                                  |   |   |                                      |            |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |
|  |                | <i>Device address</i>  | <i>Length</i>                      | <i>ASCII<br/>“T”</i> | <i>ASCII<br/>“C”</i> | <i>CRC</i>                  |                             |             |             |                  |  |  |   |   |                                      |            |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |
| <i>0x01</i>  | <i>0x02</i>    | <i>0x54</i>  | <i>0x43</i>                        | <i>0x14</i>          |                      |                             |                             |             |             |                  |  |  |   |   |                                      |            |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |
| Response example:  |                |  |                                    |                      |                      |                             |                             |             |             |                  |  |  |   |   |                                      |            |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |
| <table border="1"> <thead> <tr> <th><i>Address</i></th> <th><i>Length</i></th> <th><i>ASCII<br/>“A”</i></th> <th><i>ASCII<br/>“C”</i></th> <th><i>ASCII<br/>“K”</i></th> <th><i>CRC</i></th> </tr> </thead> <tbody> <tr> <td><i>0x01</i></td> <td><i>0x03</i></td> <td><i>0x41</i></td> <td><i>0x43</i></td> <td><i>0x4B</i></td> <td><i>0x4B</i></td> </tr> </tbody> </table>   | <i>Address</i> | <i>Length</i>  | <i>ASCII<br/>“A”</i>               | <i>ASCII<br/>“C”</i> | <i>ASCII<br/>“K”</i> | <i>CRC</i>                  | <i>0x01</i>                 | <i>0x03</i> | <i>0x41</i> | <i>0x43</i>      | <i>0x4B</i>                                  | <i>0x4B</i>                                  |   |   |                                      |            |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |
| <i>Address</i>   | <i>Length</i>  | <i>ASCII<br/>“A”</i>   | <i>ASCII<br/>“C”</i>               | <i>ASCII<br/>“K”</i> | <i>CRC</i>           |                             |                             |             |             |                  |  |  |   |   |                                      |            |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |
| <i>0x01</i>  | <i>0x03</i>    | <i>0x41</i>  | <i>0x43</i>                        | <i>0x4B</i>          | <i>0x4B</i>          |                             |                             |             |             |                  |  |  |   |   |                                      |            |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |
| 2  | LG             | Tag request command. After the tag has been requested, it is deleted from the table.   |                                    |                      |                      |                             |                             |             |             |                  |  |  |   |   |                                      |            |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |
|  |                | Request example:   |                                    |                      |                      |                             |                             |             |             |                  |  |  |   |   |                                      |            |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |
|  |                | <table border="1"> <thead> <tr> <th><i>Address</i></th> <th><i>Length</i></th> <th><i>ASCII<br/>“L”</i></th> <th><i>ASCII<br/>“G”</i></th> <th><i>CRC</i></th> </tr> </thead> <tbody> <tr> <td><i>0x01</i></td> <td><i>0x02</i></td> <td><i>0x4C</i></td> <td><i>0x47</i></td> <td><i>0x08</i></td> </tr> </tbody> </table>        | <i>Address</i>                     | <i>Length</i>        | <i>ASCII<br/>“L”</i> | <i>ASCII<br/>“G”</i>        | <i>CRC</i>                  | <i>0x01</i> | <i>0x02</i> | <i>0x4C</i>      | <i>0x47</i>                                  | <i>0x08</i>                                  |   |   |                                      |            |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |
|  |                | <i>Address</i>   | <i>Length</i>                      | <i>ASCII<br/>“L”</i> | <i>ASCII<br/>“G”</i> | <i>CRC</i>                  |                             |             |             |                  |  |  |   |   |                                      |            |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |
| <i>0x01</i>  | <i>0x02</i>    | <i>0x4C</i>  | <i>0x47</i>                        | <i>0x08</i>          |                      |                             |                             |             |             |                  |  |  |   |   |                                      |            |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |
| Response example:  |                |  |                                    |                      |                      |                             |                             |             |             |                  |  |  |   |   |                                      |            |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |
| <table border="1"> <thead> <tr> <th><i>Address</i></th> <th><i>Length</i></th> <th colspan="4"><i>Serial number<br/>(4 bytes)</i></th> <th colspan="4"><i>Tag ID<br/>(4 bytes)</i></th> <th><i>Subnet ID</i></th> <th colspan="2"><i>Transmission<br/>period<br/>(2 bytes)</i></th> <th><i>Battery<br/>voltage<br/>(1 byte)</i></th> <th><i>Signal<br/>level<br/>(1 byte)</i></th> <th><i>CRC</i></th> </tr> </thead> <tbody> <tr> <td><i>0x01</i></td> <td><i>0x0D</i></td> <td><i>0x11</i></td> <td><i>0x00</i></td> <td><i>0x00</i></td> <td><i>0x00</i></td> <td><i>0x11</i></td> <td><i>0x00</i></td> <td><i>0x00</i></td> <td><i>0x00</i></td> <td><i>0x02</i></td> <td><i>0x01</i></td> <td><i>0x00</i></td> <td><i>0xD9</i></td> <td><i>0x40</i></td> <td><i>0x96</i></td> </tr> </tbody> </table> | <i>Address</i> | <i>Length</i>  | <i>Serial number<br/>(4 bytes)</i> |                      |                      |                             | <i>Tag ID<br/>(4 bytes)</i> |             |             |                  | <i>Subnet ID</i>                             | <i>Transmission<br/>period<br/>(2 bytes)</i> |   | <i>Battery<br/>voltage<br/>(1 byte)</i> | <i>Signal<br/>level<br/>(1 byte)</i> | <i>CRC</i> | <i>0x01</i> | <i>0x0D</i> | <i>0x11</i> | <i>0x00</i> | <i>0x00</i> | <i>0x00</i> | <i>0x11</i> | <i>0x00</i> | <i>0x00</i> | <i>0x00</i> | <i>0x02</i> | <i>0x01</i> | <i>0x00</i> | <i>0xD9</i> | <i>0x40</i> | <i>0x96</i> |
| <i>Address</i>   | <i>Length</i>  | <i>Serial number<br/>(4 bytes)</i>   |                                    |                      |                      | <i>Tag ID<br/>(4 bytes)</i> |                             |             |             | <i>Subnet ID</i> | <i>Transmission<br/>period<br/>(2 bytes)</i> |  | <i>Battery<br/>voltage<br/>(1 byte)</i> | <i>Signal<br/>level<br/>(1 byte)</i>    | <i>CRC</i>                           |            |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |
| <i>0x01</i>  | <i>0x0D</i>    | <i>0x11</i>  | <i>0x00</i>                        | <i>0x00</i>          | <i>0x00</i>          | <i>0x11</i>                 | <i>0x00</i>                 | <i>0x00</i> | <i>0x00</i> | <i>0x02</i>      | <i>0x01</i>                                  | <i>0x00</i>                                  | <i>0xD9</i>                             | <i>0x40</i>                             | <i>0x96</i>                          |            |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |             |

|                   |          |   |              |              |              |                                     |              |                       |                        |              |              |      |      |      |      |      |      |      |      |
|-------------------|----------|---|--------------|--------------|--------------|-------------------------------------|--------------|-----------------------|------------------------|--------------|--------------|------|------|------|------|------|------|------|------|
| 3                 | RFID125  | 125 KHz RFID tag request. After the tag has been requested, it is deleted from the table. |              |              |              |                                     |              |                       |                        |              |              |      |      |      |      |      |      |      |      |
|                   |          | Request example:  |              |              |              |                                     |              |                       |                        |              |              |      |      |      |      |      |      |      |      |
|                   |          | Address   | Length       | ASCII<br>"R" | ASCII<br>"F" | ASCII<br>"I"                        | ASCII<br>"D" | ASCII<br>"1"          | ASCII<br>"2"           | ASCII<br>"5" | CRC          |      |      |      |      |      |      |      |      |
|                   | 0x01     | 0x07  | 0x52         | 0x46         | 0x49         | 0x44                                | 0x31         | 0x32                  | 0x35                   | 0x29         |              |      |      |      |      |      |      |      |      |
| Response example: |          |   |              |              |              |                                     |              |                       |                        |              |              |      |      |      |      |      |      |      |      |
| Address           | Length   | ASCII<br>"R"  | ASCII<br>"F" | ASCII<br>"I" | ASCII<br>"D" | Manufacturer<br>number<br>(2 bytes) |              | Card UID<br>(3 bytes) |                        |              | CRC          |      |      |      |      |      |      |      |      |
| 0x01              | 0x09     | 0x52  | 0x46         | 0x49         | 0x44         | 0x56                                | 0x5A         | 0x57                  | 0x7C                   | 0x1F         | 0x29         |      |      |      |      |      |      |      |      |
| 4                 | RFID1356 | 13.56 MHz RFID tag request  |              |              |              |                                     |              |                       |                        |              |              |      |      |      |      |      |      |      |      |
|                   |          | Request example:  |              |              |              |                                     |              |                       |                        |              |              |      |      |      |      |      |      |      |      |
|                   |          | Address   | Length       | ASCII<br>"R" | ASCII<br>"F" | ASCII<br>"I"                        | ASCII<br>"D" | ASCII<br>"1"          | ASCII<br>"3"           | ASCII<br>"5" | ASCII<br>"6" | CRC  |      |      |      |      |      |      |      |
| 0x01              | 0x08     | 0x52  | 0x46         | 0x49         | 0x44         | 0x31                                | 0x33         | 0x35                  | 0x36                   | 0x11         |              |      |      |      |      |      |      |      |      |
| Response example: |          |   |              |              |              |                                     |              |                       |                        |              |              |      |      |      |      |      |      |      |      |
| Address           | Length   | ASCII<br>"R"  | ASCII<br>"F" | ASCII<br>"I" | ASCII<br>"D" | ATQ<br>(2 bytes)                    |              | SAK                   | Card UID<br>(10 bytes) |              |              |      |      |      |      |      |      |      | CRC  |
| 0x01              | 0x11     | 0x52  | 0x46         | 0x49         | 0x44         | 0x44                                | 0x00         | 0x00                  | 0x04                   | 0xBD         | 0x17         | 0x42 | 0xCE | 0x20 | 0x84 | 0x00 | 0x00 | 0x00 | 0xCB |
| 5                 | IN1G     | Discrete input status request   |              |              |              |                                     |              |                       |                        |              |              |      |      |      |      |      |      |      |      |
|                   |          | Request example:  |              |              |              |                                     |              |                       |                        |              |              |      |      |      |      |      |      |      |      |
|                   |          | Address   | Length       | ASCII<br>"I" | ASCII<br>"N" | ASCII<br>"1"                        | ASCII<br>"D" | CRC                   |                        |              |              |      |      |      |      |      |      |      |      |
| 0x01              | 0x04     | 0x49  | 0x4E         | 0x31         | 0x47         | 0x74                                |              |                       |                        |              |              |      |      |      |      |      |      |      |      |
| Response example: |          |   |              |              |              |                                     |              |                       |                        |              |              |      |      |      |      |      |      |      |      |
| Address           | Length   | ASCII<br>"I"  | ASCII<br>"N" | ASCII<br>"1" | ASCII<br>"C" | ADC value<br>(2 bytes)              |              | CRC                   |                        |              |              |      |      |      |      |      |      |      |      |
| 0x01              | 0x06     | 0x49  | 0x4E         | 0x31         | 0x43         | 0xFF                                | 0xFF         | 0x3C                  |                        |              |              |      |      |      |      |      |      |      |      |

| 6  | OUT1E         | Discrete output switching on   |                      |                      |                      |                      |                      |                      |                       |                |               |                      |                      |                      |                      |                      |                      |                      |                      |                      |                       |             |             |             |             |             |             |             |             |             |             |
|--|---------------|--|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|-----------------------|----------------|---------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|-----------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
|  |               | Request example:   |                      |                      |                      |                      |                      |                      |                       |                |               |                      |                      |                      |                      |                      |                      |                      |                      |                      |                       |             |             |             |             |             |             |             |             |             |             |
|  |               | <table border="1"> <thead> <tr> <th><i>Address</i></th> <th><i>Length</i></th> <th><i>ASCII<br/>"O"</i></th> <th><i>ASCII<br/>"U"</i></th> <th><i>ASCII<br/>"T"</i></th> <th><i>ASCII<br/>"I"</i></th> <th><i>ASCII<br/>"E"</i></th> <th><i>CRC</i></th> </tr> </thead> <tbody> <tr> <td><i>0x01</i></td> <td><i>0x05</i></td> <td><i>0x4F</i></td> <td><i>0x55</i></td> <td><i>0x54</i></td> <td><i>0x31</i></td> <td><i>0x45</i></td> <td><i>0x3E</i></td> </tr> </tbody> </table>   |                      |                      |                      |                      |                      |                      |                       |                |               | <i>Address</i>       | <i>Length</i>        | <i>ASCII<br/>"O"</i> | <i>ASCII<br/>"U"</i> | <i>ASCII<br/>"T"</i> | <i>ASCII<br/>"I"</i> | <i>ASCII<br/>"E"</i> | <i>CRC</i>           | <i>0x01</i>          | <i>0x05</i>           | <i>0x4F</i> | <i>0x55</i> | <i>0x54</i> | <i>0x31</i> | <i>0x45</i> | <i>0x3E</i> |             |             |             |             |
| <i>Address</i>   | <i>Length</i> | <i>ASCII<br/>"O"</i>   | <i>ASCII<br/>"U"</i> | <i>ASCII<br/>"T"</i> | <i>ASCII<br/>"I"</i> | <i>ASCII<br/>"E"</i> | <i>CRC</i>           |                      |                       |                |               |                      |                      |                      |                      |                      |                      |                      |                      |                      |                       |             |             |             |             |             |             |             |             |             |             |
| <i>0x01</i>  | <i>0x05</i>   | <i>0x4F</i>  | <i>0x55</i>          | <i>0x54</i>          | <i>0x31</i>          | <i>0x45</i>          | <i>0x3E</i>          |                      |                       |                |               |                      |                      |                      |                      |                      |                      |                      |                      |                      |                       |             |             |             |             |             |             |             |             |             |             |
| 7  | OUT1D         | Discrete output switching off  |                      |                      |                      |                      |                      |                      |                       |                |               |                      |                      |                      |                      |                      |                      |                      |                      |                      |                       |             |             |             |             |             |             |             |             |             |             |
|  |               | Request example:   |                      |                      |                      |                      |                      |                      |                       |                |               |                      |                      |                      |                      |                      |                      |                      |                      |                      |                       |             |             |             |             |             |             |             |             |             |             |
|  |               | <table border="1"> <thead> <tr> <th><i>Address</i></th> <th><i>Length</i></th> <th><i>ASCII<br/>"O"</i></th> <th><i>ASCII<br/>"U"</i></th> <th><i>ASCII<br/>"T"</i></th> <th><i>ASCII<br/>"I"</i></th> <th><i>ASCII<br/>"E"</i></th> <th><i>CRC</i></th> </tr> </thead> <tbody> <tr> <td><i>0x01</i></td> <td><i>0x05</i></td> <td><i>0x4E</i></td> <td><i>0x55</i></td> <td><i>0x54</i></td> <td><i>0x31</i></td> <td><i>0x45</i></td> <td><i>0x3E</i></td> </tr> </tbody> </table>   |                      |                      |                      |                      |                      |                      |                       |                |               | <i>Address</i>       | <i>Length</i>        | <i>ASCII<br/>"O"</i> | <i>ASCII<br/>"U"</i> | <i>ASCII<br/>"T"</i> | <i>ASCII<br/>"I"</i> | <i>ASCII<br/>"E"</i> | <i>CRC</i>           | <i>0x01</i>          | <i>0x05</i>           | <i>0x4E</i> | <i>0x55</i> | <i>0x54</i> | <i>0x31</i> | <i>0x45</i> | <i>0x3E</i> |             |             |             |             |
| <i>Address</i>   | <i>Length</i> | <i>ASCII<br/>"O"</i>   | <i>ASCII<br/>"U"</i> | <i>ASCII<br/>"T"</i> | <i>ASCII<br/>"I"</i> | <i>ASCII<br/>"E"</i> | <i>CRC</i>           |                      |                       |                |               |                      |                      |                      |                      |                      |                      |                      |                      |                      |                       |             |             |             |             |             |             |             |             |             |             |
| <i>0x01</i>  | <i>0x05</i>   | <i>0x4E</i>  | <i>0x55</i>          | <i>0x54</i>          | <i>0x31</i>          | <i>0x45</i>          | <i>0x3E</i>          |                      |                       |                |               |                      |                      |                      |                      |                      |                      |                      |                      |                      |                       |             |             |             |             |             |             |             |             |             |             |
| 8  | SETADDR       | Setting an address on the RS-485 bus (for example, address = 2). Setting is applied after reboot.  |                      |                      |                      |                      |                      |                      |                       |                |               |                      |                      |                      |                      |                      |                      |                      |                      |                      |                       |             |             |             |             |             |             |             |             |             |             |
|  |               | Request example:   |                      |                      |                      |                      |                      |                      |                       |                |               |                      |                      |                      |                      |                      |                      |                      |                      |                      |                       |             |             |             |             |             |             |             |             |             |             |
|  |               | <table border="1"> <thead> <tr> <th><i>Address</i></th> <th><i>Length</i></th> <th><i>ASCII<br/>"S"</i></th> <th><i>ASCII<br/>"E"</i></th> <th><i>ASCII<br/>"T"</i></th> <th><i>ASCII<br/>"A"</i></th> <th><i>ASCII<br/>"D"</i></th> <th><i>ASCII<br/>"D"</i></th> <th><i>ASCII<br/>"R"</i></th> <th><i>New<br/>addres</i></th> <th><i>CRC</i></th> </tr> </thead> <tbody> <tr> <td><i>0x01</i></td> <td><i>0x08</i></td> <td><i>0x53</i></td> <td><i>0x45</i></td> <td><i>0x54</i></td> <td><i>0x41</i></td> <td><i>0x44</i></td> <td><i>0x44</i></td> <td><i>0x52</i></td> <td><i>0x02</i></td> <td><i>0x3E</i></td> </tr> </tbody> </table> |                      |                      |                      |                      |                      |                      |                       |                |               | <i>Address</i>       | <i>Length</i>        | <i>ASCII<br/>"S"</i> | <i>ASCII<br/>"E"</i> | <i>ASCII<br/>"T"</i> | <i>ASCII<br/>"A"</i> | <i>ASCII<br/>"D"</i> | <i>ASCII<br/>"D"</i> | <i>ASCII<br/>"R"</i> | <i>New<br/>addres</i> | <i>CRC</i>  | <i>0x01</i> | <i>0x08</i> | <i>0x53</i> | <i>0x45</i> | <i>0x54</i> | <i>0x41</i> | <i>0x44</i> | <i>0x44</i> | <i>0x52</i> |
| <i>Address</i>   | <i>Length</i> | <i>ASCII<br/>"S"</i>   | <i>ASCII<br/>"E"</i> | <i>ASCII<br/>"T"</i> | <i>ASCII<br/>"A"</i> | <i>ASCII<br/>"D"</i> | <i>ASCII<br/>"D"</i> | <i>ASCII<br/>"R"</i> | <i>New<br/>addres</i> | <i>CRC</i>     |               |                      |                      |                      |                      |                      |                      |                      |                      |                      |                       |             |             |             |             |             |             |             |             |             |             |
| <i>0x01</i>  | <i>0x08</i>   | <i>0x53</i>  | <i>0x45</i>          | <i>0x54</i>          | <i>0x41</i>          | <i>0x44</i>          | <i>0x44</i>          | <i>0x52</i>          | <i>0x02</i>           | <i>0x3E</i>    |               |                      |                      |                      |                      |                      |                      |                      |                      |                      |                       |             |             |             |             |             |             |             |             |             |             |
| Response example:  |               |  |                      |                      |                      |                      |                      |                      |                       |                |               |                      |                      |                      |                      |                      |                      |                      |                      |                      |                       |             |             |             |             |             |             |             |             |             |             |
| <table border="1"> <thead> <tr> <th><i>Address</i></th> <th><i>Length</i></th> <th><i>ASCII<br/>"A"</i></th> <th><i>ASCII<br/>"C"</i></th> <th><i>ASCII<br/>"K"</i></th> <th><i>CRC</i></th> </tr> </thead> <tbody> <tr> <td><i>0x01</i></td> <td><i>0x03</i></td> <td><i>0x41</i></td> <td><i>0x43</i></td> <td><i>0x4B</i></td> <td><i>0x4B</i></td> </tr> </tbody> </table> |               |  |                      |                      |                      |                      |                      |                      |                       | <i>Address</i> | <i>Length</i> | <i>ASCII<br/>"A"</i> | <i>ASCII<br/>"C"</i> | <i>ASCII<br/>"K"</i> | <i>CRC</i>           | <i>0x01</i>          | <i>0x03</i>          | <i>0x41</i>          | <i>0x43</i>          | <i>0x4B</i>          | <i>0x4B</i>           |             |             |             |             |             |             |             |             |             |             |
| <i>Address</i>   | <i>Length</i> | <i>ASCII<br/>"A"</i>   | <i>ASCII<br/>"C"</i> | <i>ASCII<br/>"K"</i> | <i>CRC</i>           |                      |                      |                      |                       |                |               |                      |                      |                      |                      |                      |                      |                      |                      |                      |                       |             |             |             |             |             |             |             |             |             |             |
| <i>0x01</i>  | <i>0x03</i>   | <i>0x41</i>  | <i>0x43</i>          | <i>0x4B</i>          | <i>0x4B</i>          |                      |                      |                      |                       |                |               |                      |                      |                      |                      |                      |                      |                      |                      |                      |                       |             |             |             |             |             |             |             |             |             |             |



| 9              | PING          | <p>Pinging the reader</p> <p>Request example:</p> <table border="1" data-bbox="322 199 1344 296"> <thead> <tr> <th><i>Address</i></th> <th><i>Length</i></th> <th><i>ASCII<br/>"P"</i></th> <th><i>ASCII<br/>"I"</i></th> <th><i>ASCII<br/>"N"</i></th> <th><i>ASCII<br/>"G"</i></th> <th><i>CRC</i></th> </tr> </thead> <tbody> <tr> <td><i>0x01</i></td> <td><i>0x04</i></td> <td><i>0x50</i></td> <td><i>0x49</i></td> <td><i>0x4E</i></td> <td><i>0x47</i></td> <td><i>0x15</i></td> </tr> </tbody> </table> <p>Response example:</p> <table border="1" data-bbox="322 355 804 451"> <thead> <tr> <th><i>Address</i></th> <th><i>Length</i></th> <th><i>ASCII<br/>"O"</i></th> <th><i>ASCII<br/>"K"</i></th> <th><i>CRC</i></th> </tr> </thead> <tbody> <tr> <td><i>0x01</i></td> <td><i>0x02</i></td> <td><i>0x4F</i></td> <td><i>0x4B</i></td> <td><i>0x07</i></td> </tr> </tbody> </table>   | <i>Address</i>       | <i>Length</i>        | <i>ASCII<br/>"P"</i> | <i>ASCII<br/>"I"</i>                         | <i>ASCII<br/>"N"</i> | <i>ASCII<br/>"G"</i>         | <i>CRC</i>  | <i>0x01</i> | <i>0x04</i> | <i>0x50</i> | <i>0x49</i> | <i>0x4E</i> | <i>0x47</i> | <i>0x15</i> | <i>Address</i> | <i>Length</i> | <i>ASCII<br/>"O"</i> | <i>ASCII<br/>"K"</i> | <i>CRC</i>           | <i>0x01</i>          | <i>0x02</i>                                  | <i>0x4F</i> | <i>0x4B</i>                  | <i>0x07</i> |  |            |             |             |             |             |             |             |             |             |             |             |             |             |
|----------------|---------------|---|----------------------|----------------------|----------------------|--|----------------------|------------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|----------------|---------------|----------------------|----------------------|----------------------|----------------------|--|-------------|------------------------------|-------------|--|------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| <i>Address</i> | <i>Length</i> | <i>ASCII<br/>"P"</i>  | <i>ASCII<br/>"I"</i> | <i>ASCII<br/>"N"</i> | <i>ASCII<br/>"G"</i> | <i>CRC</i>                                   |                      |                              |             |             |             |             |             |             |             |             |                |               |                      |                      |                      |                      |  |             |                              |             |  |            |             |             |             |             |             |             |             |             |             |             |             |             |
| <i>0x01</i>    | <i>0x04</i>   | <i>0x50</i>   | <i>0x49</i>          | <i>0x4E</i>          | <i>0x47</i>          | <i>0x15</i>                                  |                      |                              |             |             |             |             |             |             |             |             |                |               |                      |                      |                      |                      |  |             |                              |             |  |            |             |             |             |             |             |             |             |             |             |             |             |             |
| <i>Address</i> | <i>Length</i> | <i>ASCII<br/>"O"</i>  | <i>ASCII<br/>"K"</i> | <i>CRC</i>           |                      |  |                      |                              |             |             |             |             |             |             |             |             |                |               |                      |                      |                      |                      |  |             |                              |             |  |            |             |             |             |             |             |             |             |             |             |             |             |             |
| <i>0x01</i>    | <i>0x02</i>   | <i>0x4F</i>   | <i>0x4B</i>          | <i>0x07</i>          |                      |  |                      |                              |             |             |             |             |             |             |             |             |                |               |                      |                      |                      |                      |  |             |                              |             |  |            |             |             |             |             |             |             |             |             |             |             |             |             |
| 10             | RFID          | <p>125 KHz RFID tag request. After the tag has been requested, it is deleted from the table. There are two types of response to this command: one, as the response to the "RFID125" command, the other, as the "RFID1356" response. If there has been no tags, the answer will be the same as to the RFID125 command with a zero tag (UID is 0x00).</p> <p>Request example:</p> <table border="1" data-bbox="322 657 1444 754"> <thead> <tr> <th><i>Address</i></th> <th><i>Length</i></th> <th><i>ASCII<br/>"R"</i></th> <th><i>ASCII<br/>"F"</i></th> <th><i>ASCII<br/>"I"</i></th> <th><i>ASCII<br/>"D"</i></th> <th><i>CRC</i></th> </tr> </thead> <tbody> <tr> <td><i>0x01</i></td> <td><i>0x07</i></td> <td><i>0x52</i></td> <td><i>0x46</i></td> <td><i>0x49</i></td> <td><i>0x44</i></td> <td><i>0x16</i></td> </tr> </tbody> </table> <p>Response example:</p> <table border="1" data-bbox="322 812 1839 938"> <thead> <tr> <th><i>Address</i></th> <th><i>Length</i></th> <th><i>ASCII<br/>"R"</i></th> <th><i>ASCII<br/>"F"</i></th> <th><i>ASCII<br/>"I"</i></th> <th><i>ASCII<br/>"D"</i></th> <th colspan="2"><i>Manufacturer<br/>number<br/>(2 bytes)</i></th> <th colspan="3"><i>Card UID<br/>(3bytes)</i></th> <th><i>CRC</i></th> </tr> </thead> <tbody> <tr> <td><i>0x01</i></td> <td><i>0x09</i></td> <td><i>0x52</i></td> <td><i>0x46</i></td> <td><i>0x49</i></td> <td><i>0x44</i></td> <td><i>0x56</i></td> <td><i>0x5A</i></td> <td><i>0x57</i></td> <td><i>0x7C</i></td> <td><i>0x1F</i></td> <td><i>0x29</i></td> </tr> </tbody> </table> | <i>Address</i>       | <i>Length</i>        | <i>ASCII<br/>"R"</i> | <i>ASCII<br/>"F"</i>                         | <i>ASCII<br/>"I"</i> | <i>ASCII<br/>"D"</i>         | <i>CRC</i>  | <i>0x01</i> | <i>0x07</i> | <i>0x52</i> | <i>0x46</i> | <i>0x49</i> | <i>0x44</i> | <i>0x16</i> | <i>Address</i> | <i>Length</i> | <i>ASCII<br/>"R"</i> | <i>ASCII<br/>"F"</i> | <i>ASCII<br/>"I"</i> | <i>ASCII<br/>"D"</i> | <i>Manufacturer<br/>number<br/>(2 bytes)</i> |             | <i>Card UID<br/>(3bytes)</i> |             |  | <i>CRC</i> | <i>0x01</i> | <i>0x09</i> | <i>0x52</i> | <i>0x46</i> | <i>0x49</i> | <i>0x44</i> | <i>0x56</i> | <i>0x5A</i> | <i>0x57</i> | <i>0x7C</i> | <i>0x1F</i> | <i>0x29</i> |
| <i>Address</i> | <i>Length</i> | <i>ASCII<br/>"R"</i>  | <i>ASCII<br/>"F"</i> | <i>ASCII<br/>"I"</i> | <i>ASCII<br/>"D"</i> | <i>CRC</i>                                   |                      |                              |             |             |             |             |             |             |             |             |                |               |                      |                      |                      |                      |  |             |                              |             |  |            |             |             |             |             |             |             |             |             |             |             |             |             |
| <i>0x01</i>    | <i>0x07</i>   | <i>0x52</i>   | <i>0x46</i>          | <i>0x49</i>          | <i>0x44</i>          | <i>0x16</i>                                  |                      |                              |             |             |             |             |             |             |             |             |                |               |                      |                      |                      |                      |  |             |                              |             |  |            |             |             |             |             |             |             |             |             |             |             |             |             |
| <i>Address</i> | <i>Length</i> | <i>ASCII<br/>"R"</i>  | <i>ASCII<br/>"F"</i> | <i>ASCII<br/>"I"</i> | <i>ASCII<br/>"D"</i> | <i>Manufacturer<br/>number<br/>(2 bytes)</i> |                      | <i>Card UID<br/>(3bytes)</i> |             |             | <i>CRC</i>  |             |             |             |             |             |                |               |                      |                      |                      |                      |  |             |                              |             |  |            |             |             |             |             |             |             |             |             |             |             |             |             |
| <i>0x01</i>    | <i>0x09</i>   | <i>0x52</i>   | <i>0x46</i>          | <i>0x49</i>          | <i>0x44</i>          | <i>0x56</i>                                  | <i>0x5A</i>          | <i>0x57</i>                  | <i>0x7C</i> | <i>0x1F</i> | <i>0x29</i> |             |             |             |             |             |                |               |                      |                      |                      |                      |  |             |                              |             |  |            |             |             |             |             |             |             |             |             |             |             |             |             |

## Appendix C. Description of the UMKa200 commands

### 4.1 RESET Command (0x02)

Description: Hard reboot of the device. The device will respond to the command and reboot.

Request example:

| Device address | Length | Prefix | Command code | Subcommand code | CRC  |
|----------------|--------|--------|--------------|-----------------|------|
| 0x01           | 0x03   | 0xC8   | 0x02         | 0x00            | 0xC8 |

Response example:

| Device address | Length | Prefix | Command code | Subcommand code | Execution status<br>NO_ERROR | CRC  |
|----------------|--------|--------|--------------|-----------------|------------------------------|------|
| 0x01           | 0x04   | 0xC8   | 0x82         | 0x00            | 0x00                         | 0x4F |

## 4.2 CONFIG RS-485 Command (0x03)

Description: RS-485 Interface Configuration Command.

### 4.2.1 SET SPEED RS485 Subcommand (0x00)

Description: Setting the Baud Rate for the RS-485 Interface. Possible variants: 9600, 19200, 115200.

Request example:

| Device address | Length | Prefix | Command code | Subcommand code | Baud Rate<br>(4 bytes)<br>Example: 19200 | CRC  |
|----------------|--------|--------|--------------|-----------------|--|------|
| 0x01           | 0x07   | 0xC8   | 0x03         | 0x00            | 0x00004B00                               | 0x86 |

Response example:

| Device address | Length | Prefix | Command code | Subcommand code | Execution Status<br>NO_ERROR | CRC  |
|----------------|--------|--------|--------------|-----------------|------------------------------|------|
| 0x01           | 0x04   | 0xC8   | 0x83         | 0x00            | 0x00                         | 0x4E |

Possible execution statuses:

0x00 NO\_ERROR

0x01 WRONG\_SPEED

0x04 WRITE\_ERROR

## 4.2.2 GET INFO RS485 Subcommand (0x01)

Description: Requesting the Baud Rate for the RS-485 Interface (Baud rate).

Request example:

| Device address | Length | Prefix | Command code | Subcommand code | CRC  |
|----------------|--------|--------|--------------|-----------------|------|
| 0x01           | 0x03   | 0xC8   | 0x03         | 0x01            | 0xC8 |

Response example:

| Device address | Length | Prefix | Command code | Subcommand code | Execution status<br>NO_ERROR | Baud Rate<br>(4 bytes)<br>Example: 19200 | Address<br>(1 byte) | CRC  |
|----------------|--------|--------|--------------|-----------------|------------------------------|--|---------------------|------|
| 0x01           | 0x09   | 0xC8   | 0x83         | 0x01            | 0x00                         | 0x00004B00                               | 0x01                | 0x08 |

Possible execution statuses:

0x00 NO\_ERROR

0x07 DEFAULT\_DATA

### 4.2.3 SET ADDRESS RS485 Subcommand (0x02)

Description: Address setting on the RS-485 bus.

Request example:

| Device address | Length | Prefix | Command code | Subcommand code | Saved address | CRC  |
|----------------|--------|--------|--------------|-----------------|---------------|------|
| 0x01           | 0x04   | 0xC8   | 0x03         | 0x02            | 0x01          | 0xCD |

Response example:

| Device address | Length | Prefix | Command code | Subcommand code | Execution status<br>NO_ERROR | CRC  |
|----------------|--------|--------|--------------|-----------------|------------------------------|------|
| 0x01           | 0x04   | 0xC8   | 0x83         | 0x02            | 0x00                         | 0x4C |

Possible execution statuses:

0x00 NO\_ERROR

0x04 WRITE\_ERROR

#### 4.2.4 SET LLS ID RS485 Subcommand (0x03)

Description: Enable/Disable Sending ID via LLS. “Sending ID” parameter:

0x00 – disable transmission,

0x01 – enable transmission of the RFID 125kHz,

0x02 – enable transmission of the RFID 125kHz to the current address and of the UMKa100 tag ID to the current address +1.

Request example:

| Device address | Length | Prefix | Command code | Subcommand code | ID Transmission | CRC  |
|----------------|--------|--------|--------------|-----------------|-----------------|------|
| 0x01           | 0x05   | 0xC8   | 0x03         | 0x03            | 0x01            | 0xCD |

Response example:

| Device address | Length | Prefix | Command code | Subcommand code | Execution status<br>NO_ERROR | CRC  |
|----------------|--------|--------|--------------|-----------------|------------------------------|------|
| 0x01           | 0x04   | 0xC8   | 0x83         | 0x03            | 0x00                         | 0x4D |

Possible execution statuses:

0x00 NO\_ERROR

0x04 WRITE\_ERROR

#### 4.2.5 GET LLS ID RS485 Subcommand (0x04)

Description: Receiving settings for data transmission via the LLS protocol. “ID Transmission” – enable ID transmission:

0x00 – disable transmission,

0x01 – enable transmission of the RFID 125kHz,

0x02 – enable transmission of the RFID 125kHz to the current address and of the UMKa100 tag ID to the current address +1.

“ID Transmission Format” – if the value is 0x00 then the short format is transmitted (three bytes of the ID card in the temperature and fuel fields) or, if 0x01, the full format (five bytes of the ID card instead of all the LLS protocol fields).

Request example:

| Device address | Length | Prefix | Command code | Subcommand code | CRC  |
|----------------|--------|--------|--------------|-----------------|------|
| 0x01           | 0x03   | 0xC8   | 0x03         | 0x04            | 0xCD |

Response example:

| Device address | Length | Prefix | Command code | Subcommand code | Execution status<br>NO_ERROR | ID Transmission | ID Transmission<br>Format | CRC  |
|----------------|--------|--------|--------------|-----------------|------------------------------|-----------------|---------------------------|------|
| 0x01           | 0x06   | 0xC8   | 0x83         | 0x04            | 0x00                         | 0x00            | 0x00                      | 0x48 |

Possible execution statuses:

0x00 NO\_ERROR

0x07 DEFAULT\_DATA

#### 4.2.6 SET LLS FORMAT RS485 Subcommand (0x05)

Description: Setting the format of ID transmission via the RS-485 LLS is executed by the “ID Transmission Format” command parameter: 0x00 – transmit the short format (three bytes of the ID card in the temperature and fuel fields), 0x01 – the full format (five bytes of the ID card instead of all the LLS protocol fields).

Request example:

| Device address | Length | Prefix | Command code | Subcommand code | ID Transmission Format | CRC  |
|----------------|--------|--------|--------------|-----------------|------------------------|------|
| 0x01           | 0x05   | 0xC8   | 0x03         | 0x05            | 0x00                   | 0xCA |

Response example:

| Device address | Length | Prefix | Command code | Subcommand code | Execution status<br>NO_ERROR | CRC  |
|----------------|--------|--------|--------------|-----------------|------------------------------|------|
| 0x01           | 0x04   | 0xC8   | 0x83         | 0x05            | 0x00                         | 0x4B |

Possible execution statuses:

0x00 NO\_ERROR

0x04 WRITE\_ERROR



### 4.3 UMKa100 CONFIG Command (0x04)

Description: UMKa100 Configuration Procedure Command.

#### 4.3.1 TRANSMIT FSK WITH TIMEOUT Subcommand(0x00)

Description:

Sending commands to the UMKa100 RFID tag. Command response: command received (NO\_ERROR). For the broadcast message, the DevEUI value should be zero. During the “Timeout” period, the UMKa200 is waiting for the UMKa100 RFID tag to communicate and respond to the command.

Request example:

| Device address | Length | Prefix | Command code | Subcommand code | UMKa100 DevEUI (8 bytes) | Execution Timeout (2 bytes)<br>Example: 100 seconds | Reserved (8 bytes) | Commands and parameters<br>Example: ASCII code of the UMKa100<br>“PASS 0,100” command | CRC  |
|----------------|--------|--------|--------------|-----------------|--------------------------|---|--------------------|---|------|
| 0x01           | 0x1F   | 0xC8   | 0x04         | 0x00            | 0x0000000012020061       | 0x0064  | ...                | 0x5041535320302C313030  | 0xDB |

Response example:

| Device address | Length | Prefix | Command code | Subcommand code | Execution status<br>NO_ERROR | CRC  |
|----------------|--------|--------|--------------|-----------------|------------------------------|------|
| 0x01           | 0x04   | 0xC8   | 0x84         | 0x00            | 0x00                         | 0x49 |

### 4.3.2 GET STATUS Subcommand (0x01)

#### Description:

Requesting the execution status of the last command. Possible answers are “no error” (0x00), “the command is running” (0x05) and “the execution timed out” (0x06).

#### Request example:

| Device address | Length | Prefix | Command code | Subcommand code | CRC  |
|----------------|--------|--------|--------------|-----------------|------|
| 0x01           | 0x03   | 0xC8   | 0x04         | 0x01            | 0xCF |

#### Response example:

| Device address | Length | Prefix | Command code | Subcommand code | Execution status<br>NO_ERROR | Received Data Size<br>(2 bytes) | Data<br>Example:<br>«PASS=OK» | CRC  |
|----------------|--------|--------|--------------|-----------------|------------------------------|---------------------------------|-------------------------------|------|
| 0x01           | 0x0F   | 0xC8   | 0x84         | 0x01            | 0x00                         | 0x0007                          | 0x504153533D4F4B              | 0x02 |

#### Possible execution statuses:

0x00 NO\_ERROR

0x05 COM\_RUN

0x06 COM\_TIMEOUT

### 4.3.3 TRANSMIT FSK NOW Subcommand (0x02)

#### Description:

Sending urgent commands to the UMKa100 RFID tag without waiting for communication established. Command response: command received (NO\_ERROR). For the broadcast message, the DevEUI value should be zero. During the “Timeout” period, the command is waiting for the UMKa100 RFID tag to respond.

#### Request example:

| Device address | Length | Prefix | Command code | Subcommand code | UMKa100 DevEUI (8 bytes) | Response Waiting Timeout (2 bytes) | Reserved (8 bytes) | Commands and parameters Example: «EXIT» | CRC  |
|----------------|--------|--------|--------------|-----------------|--------------------------|------------------------------------|--------------------|---|------|
| 0x01           | 0x25   | 0xC8   | 0x04         | 0x02            | 0x0000000012020061       | 0x0064                             | ...                | 0x45584954C3                            | 0xC3 |

#### Response example:

| Device address | Length | Prefix | Command code | Subcommand code | Execution status NO_ERROR | CRC  |
|----------------|--------|--------|--------------|-----------------|---------------------------|------|
| 0x01           | 0x04   | 0xC8   | 0x84         | 0x02            | 0x00                      | 0x4B |

#### 4.3.4 GET UMKA100 PACK Subcommand(0x03)

##### Description:

Requesting the RFID tag data. After the data has been requested, the tag is deleted from the table. If there is no more tags then there is an error code DATA\_NONE (0x09) in the “Execution status” field.

##### Request example:

| Device address | Length | Prefix | Command code | Subcommand code | CRC  |
|----------------|--------|--------|--------------|-----------------|------|
| 0x01           | 0x03   | 0xC8   | 0x04         | 0x03            | 0xCD |

##### Response example:

| Device address | Length | Prefix | Command code | Subcommand code | Execution status<br>NO_ERROR | UMKa100DevEUI<br>(8 bytes) | RSSI<br>(2 bytes) | Service packet from the tag<br>See the structure in the OM<br>for the UMKa100 | CRC  |
|----------------|--------|--------|--------------|-----------------|------------------------------|----------------------------|-------------------|---|------|
| 0x01           | 0x16   | 0xC8   | 0x84         | 0x03            | 0x00                         | 0x0000000012020061         | 0xFFAD            | 0x07008C120200618800  | 0x09 |

##### Possible execution statuses:

0x00 NO\_ERROR

0x09 DATA\_NONE

#### 4.4 INPUT CONTROL Command (0x05)

Description: Discrete input control command.

##### 4.4.1 GET STATE Subcommand (0x00)

Description:

Getting the discrete input status.

Request example:

| Device address | Length | Prefix | Command code | Subcommand code | CRC  |
|----------------|--------|--------|--------------|-----------------|------|
| 0x01           | 0x03   | 0xC8   | 0x05         | 0x00            | 0xCF |

Response example:

| Device address | Length | Prefix | Command code | Subcommand code | Execution status (NO_ERROR) | Input Status (SET) | Pull-up (PULL-UP) | CRC  |
|----------------|--------|--------|--------------|-----------------|-----------------------------|--------------------|-------------------|------|
| 0x01           | 0x06   | 0xC8   | 0x85         | 0x00            | 0x00                        | 0xFF               | 0xFF              | 0x4A |

Possible input statuses:

0x00 RESET

0xFF SET

Possible input pull-up configurations:

0x00 PULL-DOWN

0xFF PULL-UP

#### 4.4.2 PULL-UP Subcommand (0x01)

Description:

Setting the pull-up to VDC.

Request example:

| Device address | Length | Prefix | Command code | Subcommand code | CRC  |
|----------------|--------|--------|--------------|-----------------|------|
| 0x01           | 0x03   | 0xC8   | 0x05         | 0x01            | 0xCE |

Response example:

| Device address | Length | Prefix | Command code | Subcommand code | Execution status (NO_ERROR) | CRC  |
|----------------|--------|--------|--------------|-----------------|-----------------------------|------|
| 0x01           | 0x04   | 0xC8   | 0x85         | 0x01            | 0x00                        | 0x49 |

#### 4.4.3 PULL-DOWN Subcommand (0x02)

Description:

Setting pull-down to GND.

Request example:

| Device address | Length | Prefix | Command code | Subcommand code | CRC  |
|----------------|--------|--------|--------------|-----------------|------|
| 0x01           | 0x03   | 0xC8   | 0x05         | 0x02            | 0xCD |

Response example:

| Device address | Length | Prefix | Command code | Subcommand code | Execution status (NO_ERROR) | CRC  |
|----------------|--------|--------|--------------|-----------------|-----------------------------|------|
| 0x01           | 0x04   | 0xC8   | 0x85         | 0x02            | 0x00                        | 0x4A |

## 4.5 OUTPUT CONTROL Command (0x06)

Description: Discrete output control command.

### 4.5.1 RESET STATE Subcommand (0x00)

Description:

Reset the discrete output status.

Request:

| Device address | Length | Prefix | Command code | Subcommand code | CRC  |
|----------------|--------|--------|--------------|-----------------|------|
| 0x01           | 0x03   | 0xC8   | 0x06         | 0x00            | 0xCC |

Response:

| Device address | Length | Prefix | Command code | Subcommand code | Execution status (NO_ERROR) | CRC  |
|----------------|--------|--------|--------------|-----------------|-----------------------------|------|
| 0x01           | 0x04   | 0xC8   | 0x86         | 0x00            | 0x00                        | 0x4B |



## 4.5.2 SET STATE Subcommand (0x01)

### Description:

Set the discrete output status.

### Request example:

| Device address | Length | Prefix | Command code | Subcommand code | CRC  |
|----------------|--------|--------|--------------|-----------------|------|
| 0x01           | 0x03   | 0xC8   | 0x06         | 0x01            | 0xCD |

### Response example:

| Device address | Length | Prefix | Command code | Subcommand code | Execution status (NO_ERROR) | CRC  |
|----------------|--------|--------|--------------|-----------------|-----------------------------|------|
| 0x01           | 0x04   | 0xC8   | 0x86         | 0x01            | 0x00                        | 0x4A |

### 4.5.3 GET STATE Subcommand (0x02)

#### Description:

Getting the discrete output status. For the switched off output, the value of the “Status” byte is 0x00, for the switched on – 0xFF.

#### Request example:

| Device address | Length | Prefix | Command code | Subcommand code | CRC  |
|----------------|--------|--------|--------------|-----------------|------|
| 0x01           | 0x03   | 0xC8   | 0x06         | 0x02            | 0xCE |

#### Response example:

| Device address | Length | Prefix | Command code | Subcommand code | Execution status (NO_ERROR) | Status (false) | CRC  |
|----------------|--------|--------|--------------|-----------------|-----------------------------|----------------|------|
| 0x01           | 0x05   | 0xC8   | 0x86         | 0x02            | 0x00                        | 0x00           | 0x48 |

## 4.6 1-WIRE CONTROL Command (0x07)

Description: 1-Wire Control Command.

### 4.6.1 GET ROM Subcommand (0x00)

Description:

Get the device ID on the 1-Wire bus from the table. If there is no data the “Execution status” is «DATA\_NONE», and “Device code” and “Device ID” fields will not be sent.

Request example:

| Device address | Length | Prefix | Command code | Subcommand code | CRC  |
|----------------|--------|--------|--------------|-----------------|------|
| 0x01           | 0x03   | 0xC8   | 0x07         | 0x00            | 0xCD |

Response example:

| Device address | Length | Prefix | Command code | Subcommand code | Execution status (NO_ERROR) | Device code (1 byte) | Device ID (3 bytes) | CRC  |
|----------------|--------|--------|--------------|-----------------|-----------------------------|----------------------|---------------------|------|
| 0x01           | 0x07   | 0xC8   | 0x87         | 0x00            | 0x00                        | 0x01                 | 0xD7A36A            | 0x56 |

Possible statuses:

0x00 NO\_ERROR

0x09 DATA\_NONE

## 4.7 GET VERSION Command (0x01)

### Description:

Requesting IDs and versions of the bootloader, updates and current firmware. All IDs and software versions are placed in the big-endian order. If there is no update, then the values of the “Update ID” and “Update Version” fields will be zeros. Software versions have a three-digit value, for example, if the bootloader version is 4.3.0, then it corresponds to the value 0x040300 in the transmitted packet.

### Request example:

| Device address | Length | Prefix | Command code | Subcommand code | CRC  |
|----------------|--------|--------|--------------|-----------------|------|
| 0x01           | 0x03   | 0xC8   | 0x01         | 0x00            | 0xCB |

### Response example:

| Device address | Length | Prefix | Command code | Subcommand code | Execution status (NO_ERROR) | Current Firmware ID (2 bytes) | Current Firmware Version (3 bytes) | Update ID (2 bytes) | Update version (3 bytes) | Bootloader ID (2 bytes) | Bootloader Version (3 bytes) | CRC  |
|----------------|--------|--------|--------------|-----------------|-----------------------------|-------------------------------|------------------------------------|---------------------|--------------------------|-------------------------|------------------------------|------|
| 0x01           | 0x12   | 0xC8   | 0x81         | 0x00            | 0x00                        | 0x00C8                        | 0x000400                           | 0x0000              | 0x000000                 | 0x0000                  | 0x040300                     | 0x91 |

## 4.8 DEV\_INFO Command (0x08)

### 4.8.1 GET KEYS Subcommand (0x00)

Description: Requesting LoRa encryption keys. The DEFAULT\_DATA status means that the data could not be read from the configuration file and, therefore, the default settings have been loaded.

Request example:

| Device address | Length | Prefix | Command code | Subcommand code | CRC  |
|----------------|--------|--------|--------------|-----------------|------|
| 0x01           | 0x03   | 0xC8   | 0x08         | 0x00            | 0xC2 |

Response example:

| Device address | Length | Prefix | Command code | Subcommand code | Execution status (NO_ERROR) | DevEUI (8 bytes)      | AppKey (16 bytes)                          | AppSkey (16 bytes)                         | NtwSkey (16 bytes)                         | CRC  |
|----------------|--------|--------|--------------|-----------------|-----------------------------|-----------------------|--|--|--|------|
| 0x01           | 0x3C   | 0xC8   | 0x88         | 0x00            | 0x00                        | 0x00000000<br>b0b0457 | 0x2b7e151628a<br>ed2a6abf7158<br>809cf4f3c | 0x2b7e151628a<br>ed2a6abf715<br>8809cf4f3c | 0x2b7e151628a<br>ed2a6abf7158<br>809cf4f3c | 0xFE |

Possible execution statuses:

0x00 NO\_ERROR

0x07 DEFAULT\_DATA

## 4.8.2 GET FACTORY NUMBER Subcommand (0x01)

Description: Requesting the serial number and DevEUI of the device. The DEFAULT\_DATA status means that the data could not be read from the configuration file and, therefore, the default settings have been loaded.

Request example:

| Device address | Length | Prefix | Command code | Subcommand code | CRC  |
|----------------|--------|--------|--------------|-----------------|------|
| 0x01           | 0x03   | 0xC8   | 0x08         | 0x01            | 0xC2 |

Response example:

| Device address | Length | Prefix | Command code | Subcommand code | Execution status (NO_ERROR) | DevEUI (8bytes)   | Serial number in ASCII (8 bytes) | CRC  |
|----------------|--------|--------|--------------|-----------------|-----------------------------|-------------------|----------------------------------|------|
| 0x01           | 0x3C   | 0xC8   | 0x88         | 0x00            | 0x00                        | 0x00000000b0b0457 | 0x3131313131313131               | 0xFE |

Possible execution statuses:

0x00 NO\_ERROR

0x07 DEFAULT\_DATA

### 4.8.3 GET SUPPLY VOLTAGE Subcommand (0x02)

Description: Requesting the supply voltage level. The “Voltage” field is the value of the ADC code (twelve bits) in which 0x0000 = 0 Volts, and 0x0FFF = 40 Volts.

Request example:

| Device address | Length | Prefix | Command code | Subcommand code | CRC  |
|----------------|--------|--------|--------------|-----------------|------|
| 0x01           | 0x03   | 0xC8   | 0x08         | 0x02            | 0xC0 |

Response example:

| Device address | Length | Prefix | Command code | Subcommand code | Execution status (NO_ERROR) | Voltage (4 bytes) | CRC  |
|----------------|--------|--------|--------------|-----------------|-----------------------------|-------------------|------|
| 0x01           | 0x08   | 0xC8   | 0x88         | 0x02            | 0x00                        | 0x000b0457        | 0x13 |

Possible execution statuses:

0x00 NO\_ERROR

## 4.9 CONTROL 125 TAG Command (0x09)

### 4.9.1 GET 125 TAG Subcommand (0x00)

Description: Requesting the 125 KHz RFID tag ID. The DATA\_NONE status means that there is no tag data available.

Request example:

| Device address | Length | Prefix | Command code | Subcommand code | CRC  |
|----------------|--------|--------|--------------|-----------------|------|
| 0x01           | 0x03   | 0xC8   | 0x09         | 0x00            | 0xC3 |

Response example:

| Device address | Length | Prefix | Command code | Subcommand code | Execution status (NO_ERROR) | Customer ID (1 byte) | UID (4 bytes) | CRC  |
|----------------|--------|--------|--------------|-----------------|-----------------------------|----------------------|---------------|------|
| 0x01           | 0x09   | 0xC8   | 0x89         | 0x00            | 0x00                        | 0x75                 | 0x00853737    | 0xB9 |

Possible statuses:

0x00 NO\_ERROR

0x09 DATA\_NONE