Inveo Sp. z o.o.

IND

Instruction Manual

Overview

Warning

This manual applies only to the firmware version v0.88 onwards. Inveo does not guarantee that the information contained in this document applies to previous firmware revisions.



Purpose of the device

The IND LED (with LED indicators) and IND LCD (with an LCD display) are universal devices designed for a wide range of applications such as industry, services, trade, and logistics. Thanks to comprehensive functionality, the device will perform well as a machine log-in device, access control to rooms (eg. warehouses), part of a customer loyalty system or a working time recorder.

The device supports integration with different systems thanks to Modbus TCP/RTU and HTTP (client/server) protocol support. It has numerous inputs and outputs that allow control over the device's elements such as the display, buzzer or additional inputs/outputs. Power can be supplied from various sources, optionally via PoE (Power over Ethernet), which makes it flexible and easily adaptable to specific user requirements and installation conditions.

The reader can read multiple RFID tag types (depending on the device's version), including Mifare Classic®, Mifare Plus® (UID), Mifare DESFire® (UID), Unique EM4100 EM4102, ICODE® (ISO 15693), HITAG (HITAG 2).

Changelog

v0.88 11th of June 2026

- Added the `cardList.csv` resource
- Added `time` and `date` markers to `status.xml`
- Corrected the column names in `log.csv`

v0.86 2nd of April 2025

- Added support of `enter` command.

Table of contents

- Overview
  - Purpose of the device
  - Changelog
- Device construction
  - Technical characteristics
  - Connectors
  - General characteristics
- Network configuration
- Changing the device's IP address - Discoverer programme
- Changing the computer's subnet address for configuration
- Network settings configuration
- Communication with the module from an external network
- Device configuration
  - Device status
  - Reader operation modes
  - Managing cards
    - Adding and removing cards using a web browser
    - Adding cards by using Modbus
    - Own API (http GET)
  - Saving data to Mifare and ICODE cards – HTTP GET command
  - Reading Mifare and ICODE blocks
  - Access – card groups
  - Events logging
    - Events preview via a www page
    - Events preview using an XML file
    - CSV download
  - LCD text
  - Reaction for events (inputs / outputs) – I/O Settings
  - Relay activation on a different Inveo device
  - Audio/visual signaling
  - Real Time Clock (RTC)
  - SNMP server configuration
  - Communication protocols and administration
- Communication protocols
  - HTTP

        - HTTP GET – read state
        - HTTP GET – control
        - Editing cards via HTTP POST
        - HTTP Client
    - Modbus
        - Holding Registers addressing
        - Single Coil addressing
    - Integration with own software
- Factory settings
- Restoring factory settings
- Warranty and manufacturer's liability
    - Storage, operation and transport conditions
    - Disposal and decommissioning

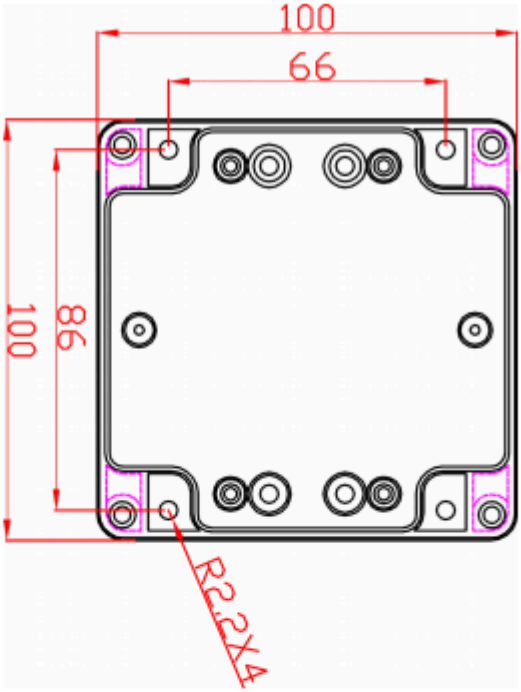
Device construction

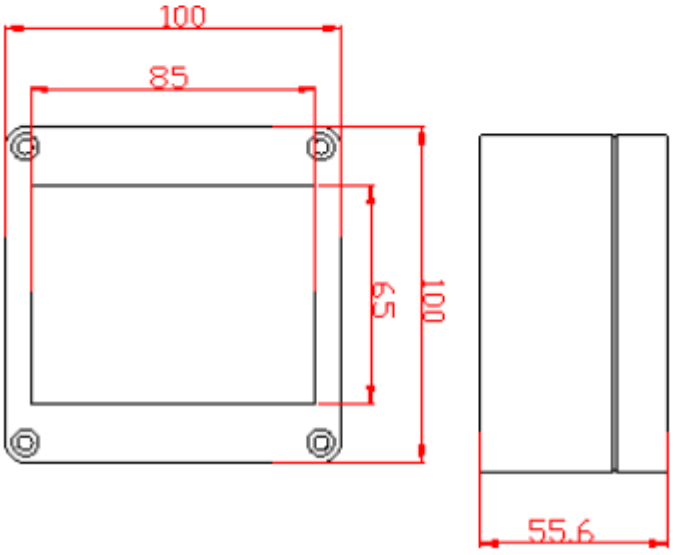
Technical characteristics

Parameter	Description
Power supply	**IND Passive PoE:** **10-24V DC** via a 3,5mm screw terminal block **Passive PoE** 10-24V DC
	IND PoE 802,3af: **10-24V DC** via a 3,5mm screw terminal block **PoE:** 33-57V PoE IEEE 802.3af
Power consumption	**Max 2,5W** (~200mA @ 12V)
Transponders	**The following IND versions are available, supporting various transponder types:** **Mif:** Mifare Classic®, Mifare Plus® (UID) and Mifare DESFire® (UID) **Uni:** Unique EM4100, EM4102 **HT2:** HITAG 2 **Ico:** ICODE® (ISO 15693) **Read distance:** up to 8cm **Built-in memory:** 1000 tags, 30 000 events
Inputs	- **Amount of inputs:** 2 - **Input type:** Normally Open (NO)
Outputs	- **Amount of outputs:** 2 - **Output type:** NO (relay), - **Maximum load:** 1A @ 30V DC
Communication	- **Ethernet** 10Mbps Half Duplex, RJ45 port - **RS485 port**, Modbus RTU
Enclosure ingress protection	**IP65**

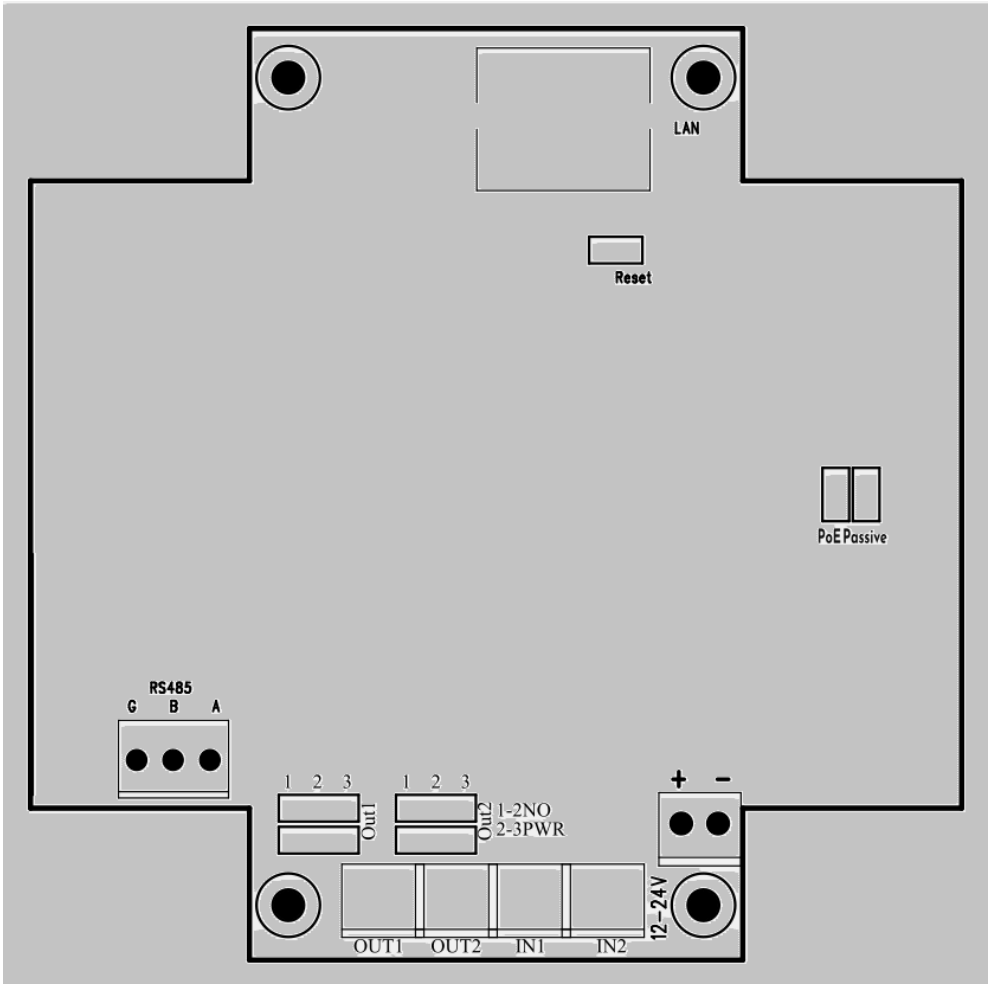
Because of different installation needs, the device does not have any mounting openings in the enclosure. Openings should be made by the customer anywhere in the rear section of the housing - this does not void warranty. Remember to use a proper cable gland in order to maintain the declared ingress protection.

Dimensions

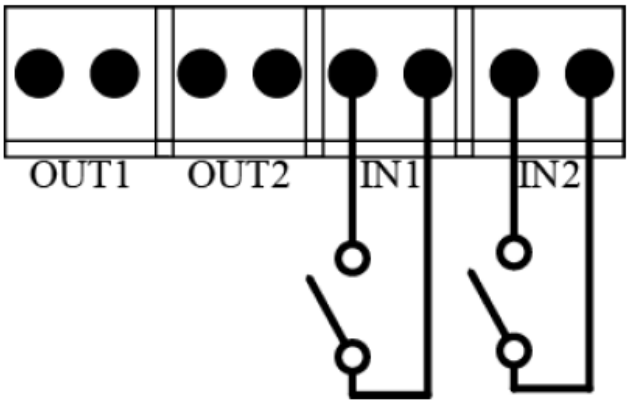




Connectors



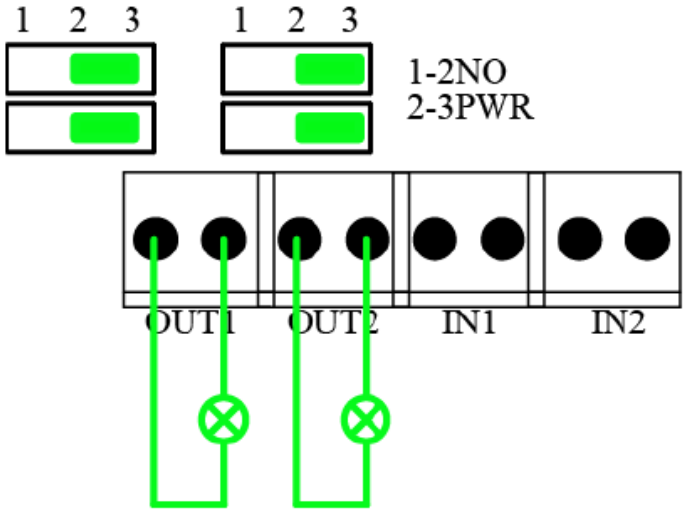
1. **LAN** – LAN connection socket,
2. **Boot/Rst** – Shorting the pins for 10-15 seconds will reset the module to factory settings,
3. **Power jumpers** – Passive PoE power jumpers: **Remove the jumpers if using the screw terminal power!** *(Only in Passive PoE version)*,
4. **RS485** – Modbus RTU connector,
5. **RelayMode** – Relay mode jumpers. In position 1-2 - NO relay output, position 2-3 - PWR output voltage is supplied from the power input (10-24VDC),
6. **OUT1** – Relay 1 connector,
7. **OUT2** – Relay 2 connector,
8. **IN1** – General purpose input No. 1,
9. **IN2** – General purpose input No. 2,
10. **+12V-** – 10-24V DC module power supply input.



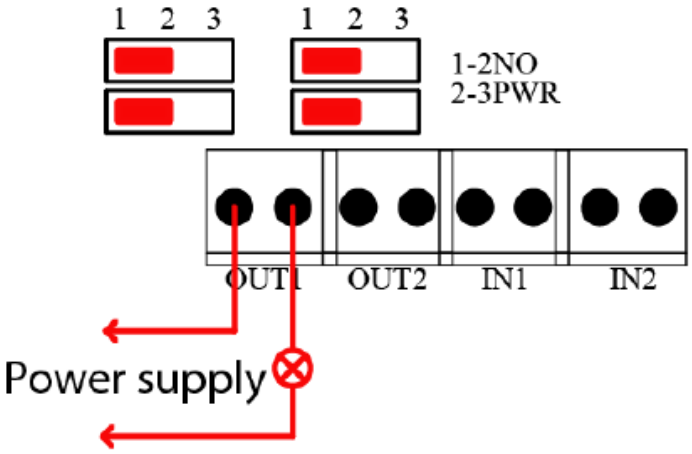
OUT1 and **OUT2** outputs can operate in two modes:

- **PWR mode** – Configuration jumpers set in position 2-3 (as shown in the image below). In this mode, an activated relay will provide the output with the power supply voltage eg. 12 VDC. If the reader is powered with 24 VDC, 24 VDC will be provided.

> **Warning**
>
> **If the reader is powered with PoE 802.3af and does not have a supply connected to the power input, outputs OUT1 and OUT2 won't have any voltage supplied to them.**



- **Normally Open mode** – Jumpers set in position 1-2 (as shown in the image below). In this mode an external power source is necessary.



General characteristics

IND LCD general view



IND LED general view



The module communicates via LAN or RS485.

The following methods can be used to access the read RFID tag:

- Built-in WWW server, available from a standard web browser,
- HTTP in server mode,
- HTTP in client mode,
- Modbus TCP,
- Modbus RTU (RS485),
- SNMP.

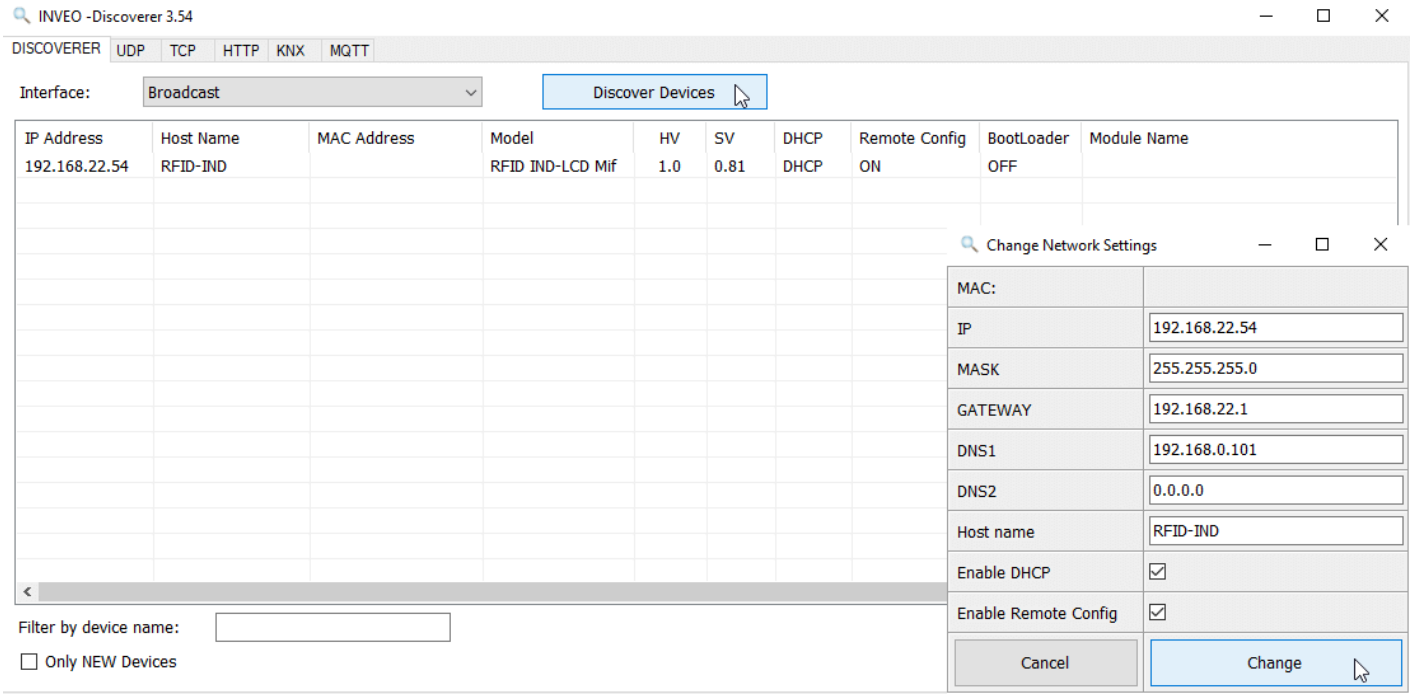
The module is equipped with an LCD display or LED indicators, informing about power supply and current device state.

The IND LED can optionally be equipped with a card slot – enabling easy machine access control or hotel room power management.

Network configuration

After the initial power-up, it's needed to configure the device. It can be done in two ways, the simplest method is to use the Inveo Discoverer programme.

Changing the device's IP address - Discoverer programme



After running the **Discoverer** programme and searching for the appropriate device:

1. Right-click the device,
2. Click the **Change settings** button.

After the dialogue window is opened you can:

- Set the IP address, mask, gateway, DNS1/DNS2,
- Change the Host name.

> **Note**
>
> **The device will be configured after the Change button is pressed.**

If **Remote Config** is disabled (enabled by default), the device needs to be configured by changing the computer's subnet:

To enable remote configuration:

1. Go to the **Administration** tab,
2. In the **Access configuration** window, select **Enable Remote Config**.

Enable Remote Network Config	☑	Enable Remote Network Config by Inveo Discover Software.

Tip

After making changes, apply them using the Save button

Changing the computer's subnet address for configuration

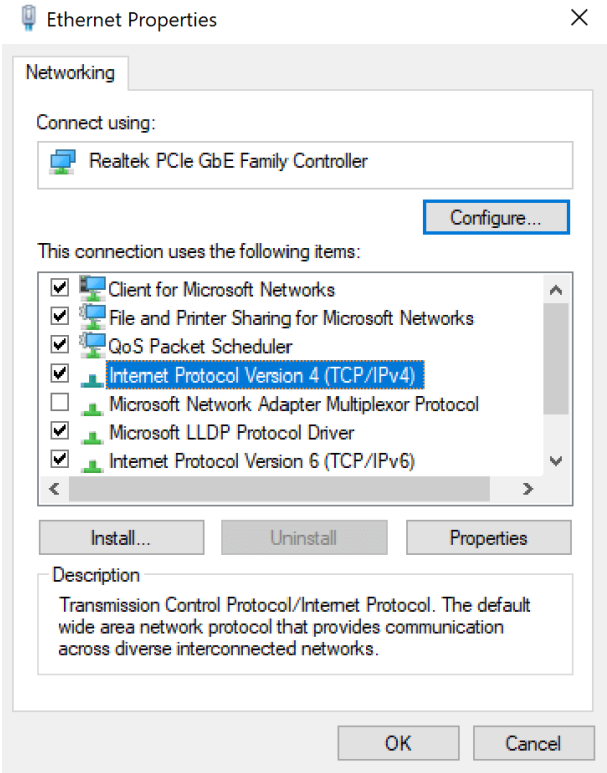
During configuration without **Discoverer** programme, it's needed to change the subnet address of the computer on the same network.

To access the computer's network configuration, do one of the following:

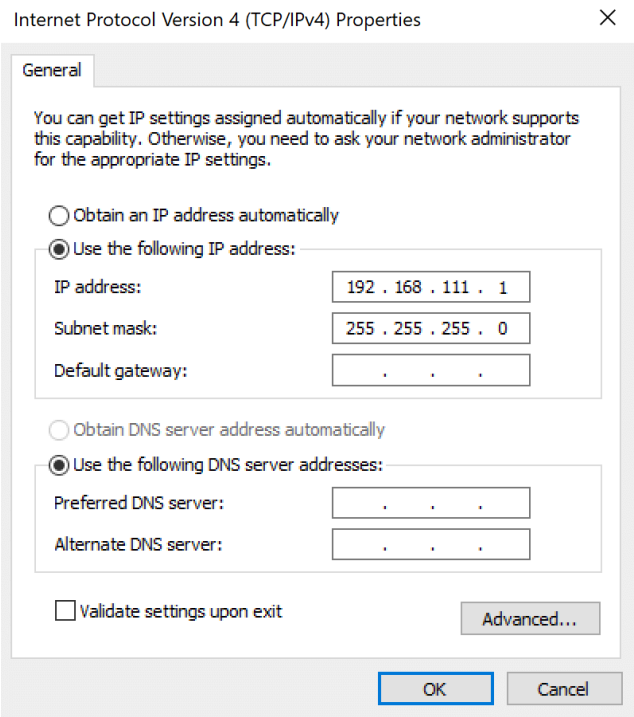
- Press **Win + R**, type in `ncpa.cpl`, and then press **Enter**,
- Alternatively, go to:
  Start → Control Panel → Network and Internet → Network and sharing center → Change adapter settings.

Then:

1. Select the network connection,
2. Right-click it and select **Properties**,
3. After selecting this option, network connection configuration screen will appear,



Select **Internet protocol version 4 (TCP/IPv4)** and type in the following parameters:



Note

After applying the configuration with OK, start a web browser and type in the following address: 192.168.111.15. (Default user and password: admin/admin00)

Network settings configuration

To customise the device's network settings, go to the **Network** tab. Parameters such as the IP address, subnet mask, gateway, DNS, and other network-specific options can be configured here.

Network Configuration

This page allows the configuration of the device's network settings.

IP Configuration

Name	Value	Description
Host Name	RFID-IND	0..15 characters
DHCP	☑	Enable DHCP Client
IP Address	192.168.22.54	A.B.C.D
IP Mask	255.255.255.0	A.B.C.D
Gateway	192.168.22.1	A.B.C.D
DNS1	192.168.0.101	A.B.C.D
DNS2	0.0.0.0	A.B.C.D

Save

HTTP Client Configuration

Server IP	0.0.0.0	A.B.C.D
Server Name		Server DNS-registered name. Set 'Server IP' to 0.0.0.0 if used.
Port	0	
Resource		HTTP pool resource ie. / or /something.php
Poll time	50	x100ms, 0-send only changes

Save

- **DHCP** – Enable/disable DHCP server,
- **IP** – Device IP address,
- **Netmask** – IP subnet mask,
- **Gateway** – Network gateway,
- **DNS1, DNS2** – DNS server addresses.

Communication with the module from an external network

If the module is located in a different LAN, port forwarding is required. Depending on the communication method utilized, contact with the network administrator and forwarding of certain ports is needed.

WWW webpage and HTTP protocol:

- TCP/IP port 80

MODBUS TCP:

- TCP/IP port 502

SNMP protocol:

- UDP port 161

Device configuration

Device status

Device status is available via the www page (default 192.168.111.15).

Status	
Name	**Value**
Input 1:	Off
Input 2:	Off
Relay 1:	Off
Relay 2:	Off
Last read ID:	00000000000000
ID known as:	None
Number of read ID:	0
Active Groups:	1234
Other info:	

After the tab is selected, the **Status** table will be displayed:

- **Input 1** – Current binary input No. 1 state,
- **Input 2** – Current binary input No. 2 state,
- **Relay 1** – Current relay output No. 1 state,
- **Relay 2** – Current relay output No. 2 state,
- **Last read ID** – Last read tag in HEX,
- **ID known as** – Read tag type (unknown/user),
- **Number of read ID** – Number of tags read since the last device restart,
- **Active Groups** – Currently active user groups,
- **Other info** – This field informs about server response timeout (`Protocol Timeout!`).

Tip

When the `(LOCK!)` flag is displayed in Last ID, this means that reading of subsequent tags will be blocked until the `releaseId` (HTTP GET) command is sent. If using Modbus, send a 0 to Holding Register 1 or to Single Coil 1004.

Reader operation modes

The device allows for operating mode selection. Extensive control options enable stand-alone operation and program control.

Services		
Name	**Value**	**Description**
Autonomic	☑	
Enable MODBUS TCP	☐	
Enable MODBUS RTU	☐	
Enable SNMP	☐	
Enable HTTP GET	☐	
Enable HTTP Client	☐	
Timeout	0	(x0,1s) a reader communication loss timeout after which the reader switches to Autonomic mode.

Autonomous mode

The device operates in stand-alone mode, the reader compares the tags with internal memory, opens the door in response to application of a saved card etc.

Program control mode

The module can be controlled with protocols. In this setting, the communication timeout time can be defined, after which the reader switches back to autonomous mode. It should be noted that the reader will return to program control mode after connection is restored.

Advanced operation mode settings

RFID Settings

Settings

Name	Value	Description
ID Compare Length	4 bytes	Length of tag ID to compare with database.
Swap ID	☐	Swap bytes in ID
Variable ID length	7B-Left padding with zeros	Mode for 4 and 7 UID length (Mifare).
Read Delay	0	Read Delay: x0.1s, 0-disable delay. Delay time for a next card reading.
Continuous reading	☐	For non Autonomic mode: Read a card continuously without release of the newFlagId.
Prevent auto IO control	☐	For non Autonomic mode: Prevent auto IO control (Door relay, LED, sound) while protocol communication is active.
HTTPClient IO control by code	☐	For HTTPClient mode: Control LED and sound by HTTP response code (200, 401,404).

- **ID Compare Length** – ID length will be compared with the card database,
- **Swap ID** – ID bytes swap function,
- **Variable ID length** – Set UID length (only in Mifare version),
- **Read Delay** – Subsequent tags read delay (0,1s; 150=15s),
- **Continuous reading** – Hotel mode – During protocol control operation, the card is visible only while it is applied to the reader, its ID is replaced with zeros after it's removed.

Information

In autonomous operation mode, the card will be read constantly and the assigned relay outputs will remain active for the entire duration of the card's presence in the read field.

Additionally, **Pulse** output mode can be set in the **I/O Settings / Output Relay 1** table. This means that the output will still be activated for a specified period of time (eg. 40*0.1s=4s) after a card is removed.

This function is most commonly used to provide power to a machine that should be active only when an authorized worker inserts a card into the reader. Similar to hotel systems – Room power is on after the card is inserted into the reader (special purpose enclosures),

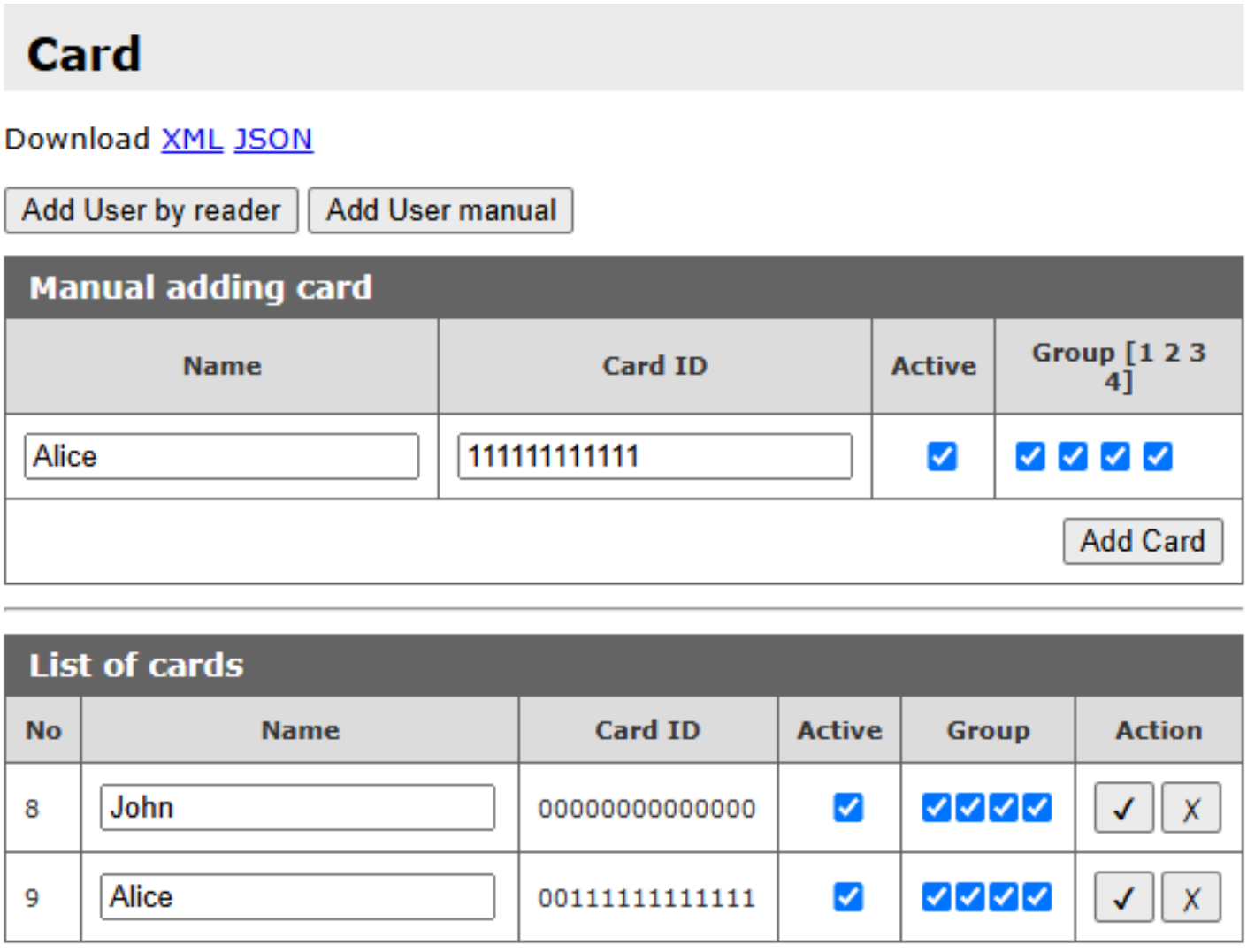
- **Prevent auto IO control** – Forces program control of outputs, LED's and the LCD display when the reader is controlled via a protocol. This option is useful in conjunction with communication timeout, after the reader switches to autonomous mode. In result, the server/controller must control interaction with the user. After communication is lost and the reader automatically enters autonomous mode, the reader will react according to the **I/O Settings**,
- **HTTPClient IO control by code** – In HTTP Client mode, the LED's and sound signal will depend on the response codes (eg. 200, 401, 404).

Managing cards

The reader allows for managing cards both via implemented communication protocols and external software.

Adding and removing cards using a web browser

Tags can be assigned manually in the **Cards** tab.



To add a new user's card, click **Add User by reader** in the **Cards** tab, then apply the card to the reader. Card assignment will be indicated by an appropriate sound signal. If the user doesn't physically hold the tag, but it's needed to add a new user, click **Add User manual**.

In the window, assign a user name, input the card ID and specify group membership. Apply the data using **Add Card**. A new record will appear in the **List of cards** after a successful card add operation. The table allows for card data editing. The **Active** field activates or deactivates the selected card. Thanks to this, a blocked user will not be recognized after his card is applied. The **Group** option is responsible for activating the card on a specified time.

The ✓ button saves the current settings.
✗ deletes the card from memory.

All of the card data stored in the reader can be downloaded by recalling `cardList.xml`, `cardList.json` or `cardList.csv` (e.g. `http://192.168.111.15/cardList.xml`).

XML file

```
<cardList>
    <cardItem>
        <no>6</no>
        <name>John</name>
        <active>1</active>
        <group>1111</group>
        <cardId>0000000000</cardId>
    </cardItem>
    <cardItem>
        <no>7</no>
        <name>Alice</name>
        <active>1</active>
        <group>1010</group>
        <cardId>1111111111</cardId>
    </cardItem>
</cardList>
```

JSON file

```
{
    "cards": [
        {
        "id":6,
        "name":"John",
        "active":"1",
        "group":"1111",
        "cardId":"0000000000"
        },
        {
        "id":7,
        "name":"Alice",
        "active":"1",
        "group":"1010",
        "cardId":"1111111111"
        }
    ]
}
```

CSV file

```
UserId;Name;Active;Group;CardId
6;John;1;1111;0000000000
7;Alice;1;1010;1111111111
```

The card can also be added by using the RFID-card Manager PC programme. To save the cards to the reader's memory, select a CSV file with a card number, user name, and card ID. RFID-card Manager will initiate communication with the module and save the cards to the device's memory.

Adding cards by using Modbus

The module is equipped with the Modbus protocol. Card ID and user name can be saved to the memory by using this protocol. In the following example, the values are sent to the Holding Registers in decimal.

Mifare card example

- **Card number:** 8,
- **Card ID:** 4923267D,
- **User name:** John.
- Enable Modbus TCP or RTU in the module,
- Connect the module to the PC using a USB to RS485 converter or TCP to make Modbus communication possible,
- Run the Modbus service programme and set the PC as a Master. For easier configuration, set the display option for Holding Registers from 1100 to 1149 (50 records),
- After configuration is complete, open the address edition window. To save the card number, send a value minus 1 to Holding Register 1100. For number 8, send 7,
- The next step is to select the card settings (such as: card activation, group membership) by sending an appropriate value to Holding Register 1101.

Dividing card ID to bytes and converting it to decimal:

49 (hex)=73 (dec)
23 (hex)=35 (dec)
26 (hex)=38 (dec)
7D (hex)=125 (dec)

4 bytes for Mifare cards, 5 bytes for Unique cards

Sending of card IDs happens by sending subsequent bytes to Holding Registers 1102-1105.

1. Assign a user name by converting it to ASCII code:

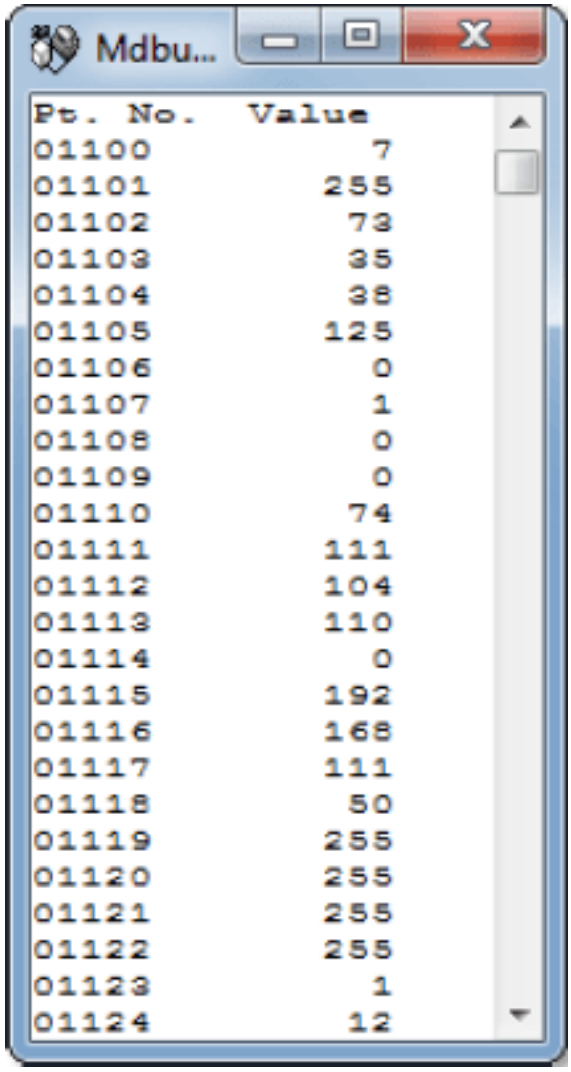
J=74
o=111
h=104
n=110

After the user name is converted to ASCII, send the code values to Holding Registers starting from 1110 (1110-1113 in this example). The name must end with NULL – after the last character, send a 0 to the next address (1114 in this example).

Example

In this example, we save a Mifare card with number 8, card ID 4923267D, and user name John. Holding Register values:
- 1100: 7 (card number - 1),
- 1102-1105: 73, 35, 38, 125 (card ID),
- 1110-1113: 74, 111, 104, 110 (user name),
- 1114: 0 (NULL).

The following screenshot features the Holding Registers with values from the example:



Own API (http GET)

A card can be added to the reader's memory by using HTTP GET commands.

To add a card, type one of the following addresses into a web browser:

- In HEX: `<IP_address>/msg.php?addId="card_ID"`

Example

`http://192.168.111.15/msg.php?addId=0600ADDA62`

- In decimal: `<IP_address>/msg.php?addDecId="card_ID"`

Example

`http://192.168.111.15/msg.php?addDecId=10000000098`

The device returns a number, under which the card was saved: `Return: 8`

Using this number, assign a user name by typing: `<IP_address>/msg.php?changeName=<card_ID>!<username>`

Example

`http://192.168.111.15/msg.php?changeName=8!John`

A card can also be added by using the command: `<IP_address>/msg.php?addCard`

Read the card first, and then type the above command in to the browser. The device saves the last read tag. After using this command, the card number will be returned. To assign a name, use the command: `<IP_address>/msg.php?changeName=<card_ID>!<username>`

More HTTP GET commands are described in the **HTTP GET – control** chapter.

Saving data to Mifare and ICODE cards – HTTP GET command

The reader can save Mifare Classic 1k/4k and ICODE card blocks. Data can be saved to any memory block.

Saving data to an ICODE card

ICODE cards are equipped with an EEPROM. Data can both be written and read. The tag memory is divided into blocks, 4 bytes each.

Tip

In order for the read/write via HTTP GET to operate correctly, make sure that the Read Block Module option in RFID Settings is DISABLED!

ICODE Block Reader

Name	Value	Description
Enable		Enable Read Block Module. (Only for advanced user!)
Block address	0	Address of block to read.
Overwrite UID		Overwrite UID by read data.
Offset for UID	0	Start copy offset for override UID.
Add data to HTTPClient		Add read data to HTTPClient.

The user can write and read the card contents. Using relevant parameters, 16 bytes of tag memory (4 data blocks) can be written at a time. If writing less bytes than a multiple of 4 (length of a single ICODE block), the remaining bytes will be overwritten with zeros. – **example 1**

Syntax: `<IP_address>/block.xml?block=<block_no>&data=<data>`

- `block` – Block to be written/read,
- `data` – Up to 16 bytes of data in HEX format.

After the command is sent, apply the card to the reader. The reader will respond in the following way:

```
<cardContent>
    <result>0</result>
    <uid>9D720004000004E0</uid>
    <block>33333333444444445555555566666666</block>
</cardContent>
```

If the **result** value is different than 0 – the write operation failed.

Tip

If the data field is present, write will occur. If it's absent, a read will occur.

Example 1

Sending 7 bytes of data:

`http://192.168.111.15/block.xml?block=3&data=1A999999111118`

Reader returned the following data:

```
<cardContent>
    <result>0</result>
    <uid>9D720004000004E0</uid>
    <block>1A999999</block>
</cardContent>
```

The reader is assigned with an IP address: 192.168.111.15. The data parameter can contain up to 16 bytes, but 7 bytes were sent. Data will be written to the card starting from block 3: - block 3 – 1A999999, - block 4 – 11111800 (remaining byte was filled with 0s), - block 5 – 98765421 (block unchanged), - block 6 – AF251536 (bblock unchanged).

Example 2

Failed write operation:

`http://192.168.111.15/block.xml?block=34&data=ABA00D&keyType=B&keyIdx=1`

Reader returned the following data:

```
<cardContent>
    <result>10</result>
    <uid>0000001CC60D0D</uid>
    <block>ABA00D00000000000000000000000000</block>
</cardContent>
```

Data write failed because the value of result is not 0.
Authorization with 'B' key was used (`keyType=B`) from the location No. '1' (`keyIdx=1`).

Reading Mifare and ICODE blocks

The reader can read the memory blocks of Mifare Classic 1k/4k and ICODE cards.
Any block can be read from the memory. The read data is available via HTTPClient and Modbus (from Holding Register 100). In HTTP Client, a "block" field is added to the HTTP GET request, containing the hex value of data read (16 bytes). Additionally, the read block can be assigned as a card UID. In this case, the number of copied block bytes corresponds to the length of the card UID (eg. 8 bytes for ICODE tags).

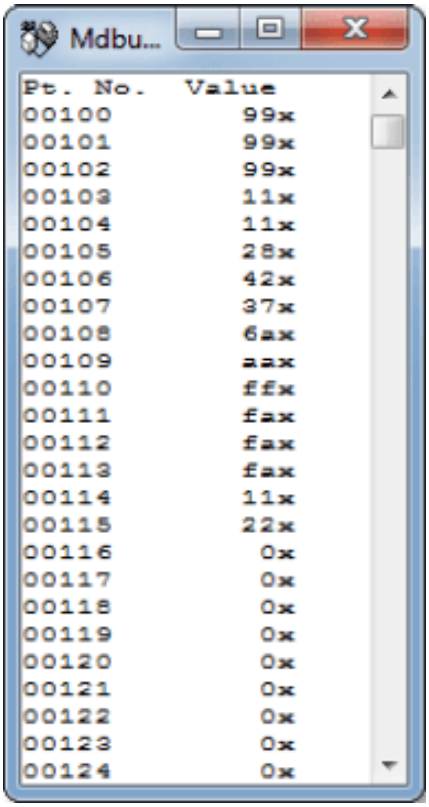
Reading ICODE block contents – Block Reader

In the **RFID Settings** tab, **ICODE Block Reader** table, ICODE card blocks readout can be configured – HTTP Client, Modbus, autonomous operation.

ICODE Block Reader		
Name	**Value**	**Description**
Enable	☑	Enable Read Block Module. (Only for advanced user!)
Block address	3	Address of block to read.
Overwrite UID	☑	Overwrite UID by read data.
Offset for UID	0	Start copy offset for override UID.
Add data to HTTPClient	☐	Add read data to HTTPClient.

- **Enable** – Enable the block reading function,
- **Block address**,
- **Override UID** – Override card ID with read data,
- **Offset for UID** – Number of the byte in a block the data read operation starts from,
- **Add data to HTTPClient** – Read block will be added to a GET request in HTTP Client mode.

Read block using Modbus – data contain Holding Registers 100 - 115:



Reading ICODE tag contents – HTTP GET command

ICODE cards are equipped with EEPROM. They can both be read and written. 1024 memory bits (128B) divided into 32 blocks 4 bytes each, are available to the user.

Tip

In order for the read/write via HTTP GET to operate correctly, make sure that the Read Block Module option in RFID Settings is DISABLED!

ICODE Block Reader

Name	Value	Description
Enable	☐	Enable Read Block Module. (Only for advanced user!)
Block address	0	Address of block to read.
Overwrite UID	☐	Overwrite UID by read data.
Offset for UID	0	Start copy offset for override UID.
Add data to HTTPClient	☐	Add read data to HTTPClient.

The user can write and read tag contents. To read the card, use the command: `<IP_address>/block.xml?block=<block_no>`.

After the command is sent, apply the card to the reader. The reader will respond in the following way:

```
<cardContent>
    <result>0</result>
    <uid>9D720004000004E0</uid>
    <block>33333333444444445555555566666666</block>
</cardContent>
```

If the **result** value is different than 0 – the write operation failed.

Reading Mifare Classic 1k/4k blocks – Block Reader

In the **RFID Settings** tab, **Mifare Block Reader** table, Mifare card block readout can be configured – HTTP Client, Modbus, autonomous operation.

Mifare		
Name	**Value**	**Description**
Enable	☐	Enable Read Block Module. (Only for advanced user!)
Block address	0	Address of block to read.
Auth key		Key in hex format (12 characters). Leave empty when not change. Default key is: FFFFFFFFFFFF.
Auth key B	☐	Authenticate by key B. Default is key A.
Overwrite UID	☐	Overwrite UID by read data.
Offset for UID	0	Start copy offset for override UID.
Add data to HTTPClient	☐	Add read data to HTTPClient.

- **Enable** – Enable the block reading function,
- **Block address**,
- **Auth key** – Authentication key in HEX format - the key is saved in secure memory (write only). The key should be written only when a change is required. Default key: 6xff (FFFFFFFFFFFF),
- **Auth key B** – Key selection. By default, key "A". If key "B" must be used, select this option,
- **Override UID** – Override card ID with read data,
- **Offset for UID** – Number of the byte in a block the data read operation starts from,
- **Add data to HTTPClient** – Read block will be added to a GET request in HTTP Client mode.

Tip

The reader will not emit a sound if the wrong key is inputted. This condition is treated as a card read failure.

Example block read using HTTPClient:

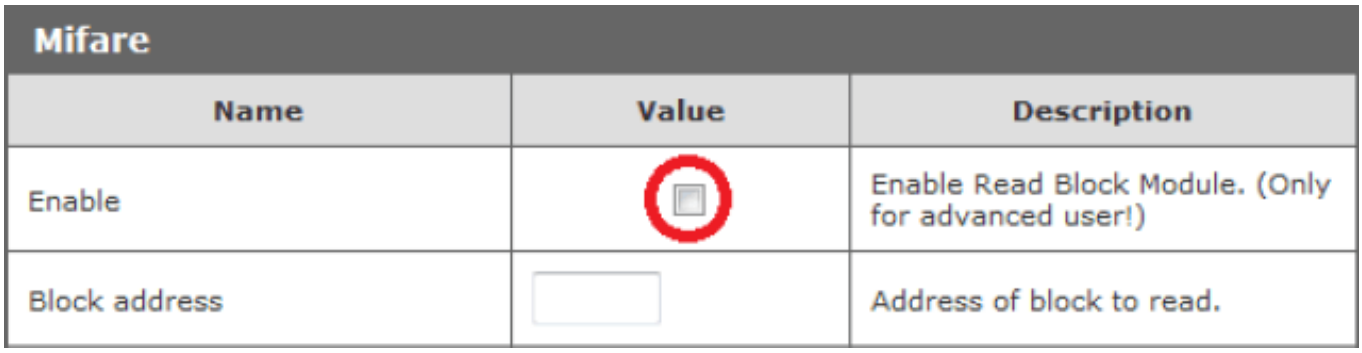
```
GET?
mac=000000000000&ip=192.168.0.236&name=&id=697C0004000004E0&inout=0&block=99999911112842376AAAFFFAFA
FA1122&io=0&put=1
```

Block read via Modbus – data is located in the Holding Registers 100 – 115.

Reading Mifare Classic 1k/4k data – HTTP GET command

Tip

In order for the read/write via HTTP GET to operate correctly, make sure that the Mifare Block Reader option in RFID Settings is DISABLED!



Mifare		
Name	**Value**	**Description**
Enable		Enable Read Block Module. (Only for advanced user!)
Block address		Address of block to read.

The user can write and read tag contents. 16 bytes can be read at one time by sending appropriate parameters (1 memory block of the tag).

Syntax:

```
<IP_address>/block.xml?block=<block_no>&keyType=<A_or_B>&keyIdx=<0-7>
```

- **block** – Number of block to read data from,
- **keyType** – Authorization type: key A or key B,
- **keyIdx** – Memory key index (0-7) – described below.

After the command is sent, apply the card to the reader.

Tip

The reader will not emit a sound if the wrong key is inputted. This condition is treated as a card read failure.

The reader will respond in the following way:

```
<cardContent>
    <result>0</result>
    <uid>0000001CC60D0D</uid>
    <block>ABA00D0000000000000000000000000</block>
</cardContent>
```

If the **result** value is different than 0 – the write operation failed.

The user can set up to 8 Mifare keys. Specific key should be reacalled by keyIdx. By default, all keys (0-7) have the value of FFFFFFFFFFFF.
We suggest that the key '0' remains unchanged. Mifare tags have default keys A and B set to FFFFFFFFFFFF.

To set a Mifare key, send 2 parameters:

- **keyIdx** – Location number,
- **key** – Key in HEX format (6 bytes → 12 hex characters).

Syntax:
`<IP_address>/block.xml?keyIdx=<location_no>&key=<key>`

After the command is sent, apply the card to the reader.

Example

Key write example: `http://192.168.111.15/block.xml?keyIdx=2&key=FFB7FFB7FFB7`

The reader returns the following data:

```
<cardContent>
    <result>12</result>
    <uid>00000000000000</uid>
    <block>00000000000000000000000000000000</block>
</cardContent>
```

Warning! (for writing a key) - If result is equal to **12** – key write operation was successful. - If result is equal to **11** – key write operation has failed.

Example 1

Read block No. 14: `http://192.168.111.15/block.xml?block=14&keyType=A&keyIdx=3`

The reader returns the following data:

```
<cardContent>
    <result>0</result>
    <uid>0000001CC60D0D</uid>
    <block>99000000000000000000000000000000</block>
</cardContent>
```

result is equal to '0', data read operation successful. Authorization with 'A' key was used (`keyType=A`) from the location No. '3' (`keyIdx=3`).

Example 2

Failed data read operation: `http://192.168.111.15/block.xml?block=14&keyType=B&keyIdx=3`

The reader returns the following data:

```
<cardContent>
    <result>10</result>
    <uid>0000001CC60D0D</uid>
    <block>00000000000000000000000000000000</block>
</cardContent>
```

result is different than '0', data read operation failed. Authorization with 'B' key was used (`keyType=B`) from the location No. '3' (`keyIdx=3`).

Access – card groups

The reader allows activation of select cards at a specified time.
This means that you need to set the card activation time from which the user will be recognized in the system.
Group configuration can be done in the **Access** tab.

Access Groups

No	Enable rule	Start time	Group 1	Group 2	Group 3	Group 4	Save
1	☑	07:00	☑	☐	☐	☐	✓
2	☑	15:00	☑	☑	☐	☐	✓
3	☑	21:00	☐	☐	☑	☑	✓
4	☐	06:00	☐	☐	☐	☐	✓

In the **Access Groups** table, 4 conditions can be defined that can be assigned to 4 groups. The table fields have the following meaning:

- **Enable rule** – Enable selected comdition,
- **Start time** – The time from which the respective groups are activated and the previous ones deactivated (at 21:00, groups No. 1 and No. 2 are deactivated, and groups No. 3 and No. 4 are activated),
- **Group 1-4** – Assignment of condition to groups.

Save the configuration of each condition.

Tip

If the Enable rule field is not selected, assigning a group to this condition will not cause the card to be unlocked.
If multiple groups with different activation conditions are assigned to a card, the card activation condition has priority.
Not assigning a card to any group is equivalent to card deactivation.

Example 1

Only conditions 1 and 2 are enabled.

Access Groups

No	Enable rule	Start time	Group 1	Group 2	Group 3	Group 4	Save
1	☑	07:00	☑	☐	☐	☐	✓
2	☑	14:00	☑	☑	☑	☑	✓

Cards with group 1 selected will be active all day. Cards with groups 2, 3 and 4 will be active from 2PM to 7AM.

Example 2

Only conditions 1, 3 and 4 are enabled.

Access Groups

No	Enable rule	Start time	Group 1	Group 2	Group 3	Group 4	Save
1	☑	07:00	☑	☐	☐	☐	✓
2	☐	14:00	☑	☑	☐	☐	✓
3	☑	15:00	☐	☐	☑	☑	✓
4	☑	21:00	☐	☐	☐	☑	✓

List of cards

No	Name	Card ID	Active	Group	Action
6	Olaf	000000717AA11B	☑	☐☑☑☐	✓ ✗
7	John	00000031C19E1B	☑	☑☐☐☑	✓ ✗
8	Rex	00000061F3911B	☑	☑☑☑☐	✓ ✗

Groups: Before 7:00 AM, group 4 is active from 9:00 PM. At 7:00 AM group 4 is deactivated and group 1 is activated. At 2:00 PM condition 2 starts, which should remove condition 1. The Enable rule field is not checked, so the condition is not taken into account by the reader. At 3:00 pm, group 1 is deactivated, while groups 3 and 4 are activated. From 9:00 AM to 7:00 AM, only group 4 is active. Group 2 is inactive all day.

Users: User Olaf is assigned to groups 2 and 3. The user will be recognized by the system between 3:00 PM and 9:00 PM. Only then is group 3 active.
User John is assigned to groups 1 and 4. This means that the card will be active all day. From 7:00 AM the condition for group 1 is met. At 3:00 PM group no. 1 is deactivated, but group 4 is activated, which is deactivated at 7:00 AM.
User Rex is assigned ot groups 1, 2 and 3. This user's card will be active from 7:00 AM to 9:00 PM. At 7:00 AM the condition for group 1 is met. At 3:00 PM group no. 1 is deleted, but group 3 is activated at the same time. At 9:00 PM the condition for group 3 is removed.

Events logging

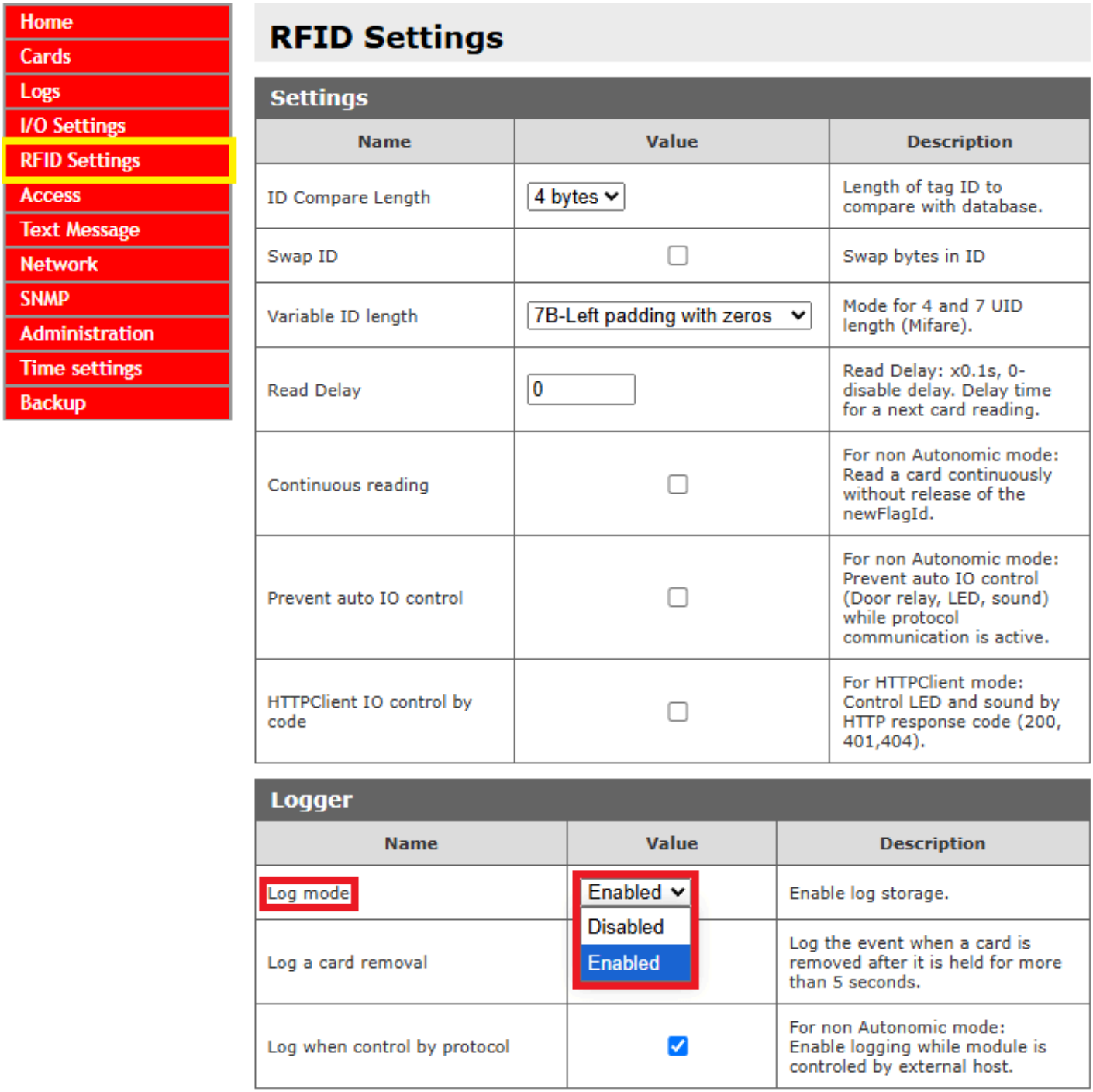
The reader can log tag application events.

For the reader to register RFID tag application to its memory, select an appropriate Log mode in **RFID Settings** → **Logger**.

Event preview is available via:

- www page (**Logs** tab).
- XML file,
- Modbus - read from appropriate addresses,
- raw file.

Events configuration



- **Log mode:**
  - **Disabled**,
  - **Enabled**.
- **Log a card removal** – Selecting this option will additionally register a card removal event after it was held for longer than 5 seconds. Option available only in **Continuous reading** mode,
- **Log when control by protocol** – This option allows logging of events even when the reader is controlled by a protocol. By default, when protocol control is enabled, the reader does not store events.

Events preview via a www page

The **Log** table contains the card read history.

Log

Time status	
Name	**Value**
Current Time	09:54:34
Current Date	2025-04-01

Log		
No	**Name/ID**	**Time**
1	Rex [00000061F3911B]	Tue, 01 Apr 2025 09:54:17
2	John [00000031C19E1B]	Tue, 01 Apr 2025 09:54:19
3	??? [000000D57E2014]	Tue, 01 Apr 2025 09:54:23

The marked flag indicates an attempt to read a card that is not present in the device's memory or is set as inactive.

Clicking **Remove logs** will remove all logs from the device's memory.

Events preview using an XML file

The user can download all device logs by recalling the `<IP_address>/logList.xml` resource or by clicking **XML** in the **Logs** tab:

XML file

```
<logList>
    <logItem>
        <no>1</no>
        <id>6</id>
        <name>John</name>
        <cardId>0000000000</cardId>
        <state>10000000</state>
        <time>1743501257</time>
    </logItem>
    <logItem>
        <no>2</no>
        <id>7</id>
        <name>Alice</name>
        <cardId>1111111111</cardId>
        <state>10000000</state>
        <time>1743501259</time>
    </logItem>
</logList>
```

- `no` – Reading No.,

- `id` – Card number defined in **Cards** minus 1,
- `name` – Name assigned to the user. For unknown or inactive cards, this tag is empty,
- `cardId` – Card UID number,
- `state` – Applies only to RFID RCP with buttons – flag specifying which function keys have been selected:

BIT	dec	Description
0	1	Function key 1 (eg. private)
1	2	Function key 2 (eg. official)
2	4	Backlit key 1 (eg. entry)
3	8	Backlit key 2 (eg. exit)
4	16	Reserve
5	32	Reserve
6	64	Reserve
7	128	Card recognized by the reader

- `time` – Unix timestamp.

CSV download

The option to download logs in CSV formati is available by clicking **CSV** in the **Logs** tab.

CSV file

```
No;UserId;Name;CardId;Date;Time;Flags;Timestamp
1;6;John;0000000000;25-04-01;128;1743501257
2;7;Alice;1111111111;25-04-01;128;1743501259
```

The CSV file contains event records, such as a subsequent number, card user ID (if identified), card ID number, card user name (if defined), date and time of the event, and card mode flag. Additionally, it contains a Unix timestamp for precision. The following table explaining the card mode flags can be used to interpret the data:

		bin	dec	hex
FLAG_PRIV	Private	0000 0001	1	1
FLAG_WORK	Official (work)	0000 0010	2	2

FLAG_IN	Entry	0000 0100	4	4
FLAG_OUT	Exit	0000 1000	8	8
LOG_CARD_ON	Card application	0100 0000	64	40
LOG_CARD_AWAY	Card removal	1100 0000	192	C0
LOG_CARD_KNOWN_ON	Known card application	1000 0000	128	80
LOG_POWER_ON	Power supply on	1111 1111	255	FF
LOG_LONG_OPEN	Card hold action	0001 0000 0000	256	100
LOG_EXT_READER	Log from an external reader	0010 0000 0000	512	200
LOG_OUT1	Current output 1 state	0100 0000 0000	1024	400
LOG_OUT2	Current output 2 state	1000 0000 0000	2048	800
LOG_IN1	Current input 1 state	01 0000 0000 0000	4096	1000
LOG_IN2	Current input 2 state	10 0000 0000 0000	8192	2000

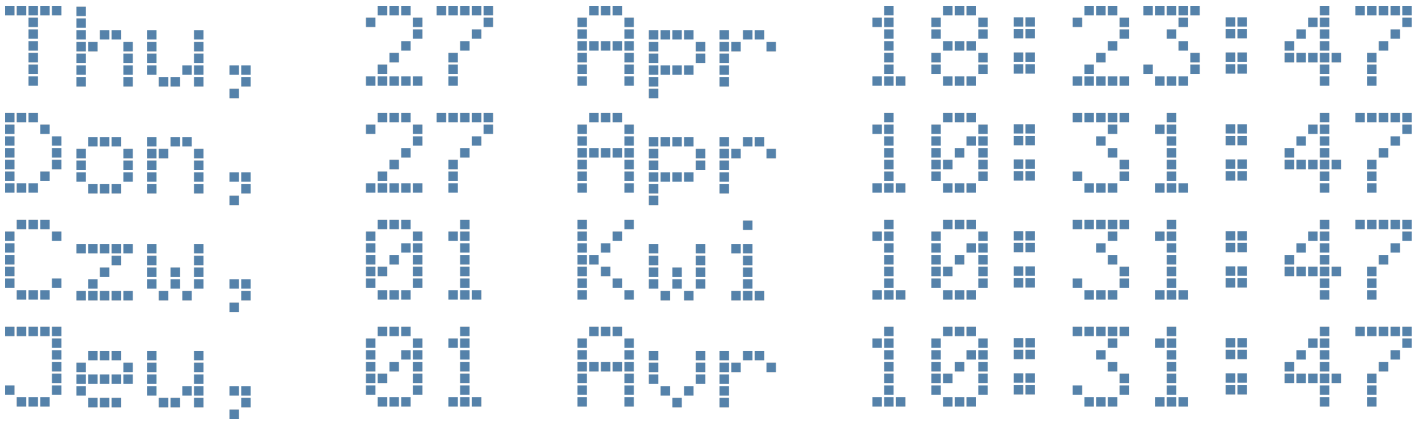
For example, if the flag value is 128, this indicates that a card has been applied. Flag 1024 indicates the current output 1 state. Adding those values (for a flag 1152), we can assume that the card application caused the output 1 state to change.
Thanks to this option, the user can easily download and interpret the logs registered by the device in CSV format, enabling further analysis and data management.

LCD text

In the **Text Message** tab, it's possible to configure the device's display settings.

Time table

In this table, the display language of the time shown in the first row of the display can be configured.



LCD Message table In this table, displaying of notifications on the LCD screen can be activated and a **Large font size** enabled.

Prompt Message, Enter Accept Message and Enter Rejest Message tables

In the tables, it's possible to configure the texts displayed on the screen during various events: application of a known card, application of an inactive card, hold time.

Prompt Message		
Name	**Value**	**Description**
Line 1	Touch with RFID card	LCD First line 0..20 characters
Line 2		LCD Second line 0..20 characters

Enter Accept Message		
Name	**Value**	**Description**
Line 1	Accepted!	LCD First line 0..20 characters
Line 2		LCD Second line 0..20 characters

Enter Reject Message		
Name	**Value**	**Description**
Line 1	Rejected!	LCD First line 0..20 characters
Line 2		LCD Second line 0..20 characters

Reaction for events (inputs / outputs) – I/O Settings

The reader can autonomously control sound and optical signaling, door bolt relay, and alarm state. Functions are described in the tables below.

> **Tip**
>
> **If the reader is set to program control, configure additional output control and sound/visual signaling.**

Selecting **Door unlock** in the **Input 1** table forces automatic activation of relay 1 (door bolt) after input 1 is activated (shorted):

Input 1		
Name	**Value**	**Description**
Door unlock	☑	Trigger an Open state for Output 1.

The **Input 2** table allows the user to set a reaction for the opening of input No, 2:

Input 2		
Name	**Value**	**Description**
Open Timeout Alarm	☑	Override 'Output 2' for alarm state when door is open for long time.
Max Time	5	x 1s

Selecting **Open Timeout Alarm** will activate the function of automatic activation of relay 2 in the event input 2 remains open for a time longer than **Max Time**.

By using this function, an open door sensor can be connected to input 1 and an alarm siren to output 2. Thanks to this, if someone opens the door and it is not shut within the defined time or remains partially open, an alarm will sound.

Remember to **Save** all changes.

In the **Output Relay 1**, **Output Relay 2** tables, module outputs can be configured independently.

Output Relay 1		
Name	**Value**	**Description**
Mode	○ Disabled ● Pulse ○ Toggle	
Time-on	40	x 0.1s
Action	Known Cards	
Invert output	☐	Invert output relay state to NC
Long Open	10	Enable relay for given time (x 1s) when card is hold for 5 seconds.

- **Mode:**
  - **Disabled** – Disable relay control,
  - **Pulse** – After activation, the relay remains on for a specified period of time (eg. door control),
  - **Toggle** – Tag read event causes the output to invert its state.
- **Time-on** – Relay activation time in Pulse mode. In 0,1s units (20 means 2 seconds),
- **Action** – Action that causes the output to activate:
  - **None** – Control via HTTP, SMNP, Modbus,
  - **All Cards** – Output is activated after any RFID tag application (doesn't matter whether it's saved to the module's memory or not),
  - **Known Cards** – Output is activated only after an active tag is applied,
  - **Unknown Cards** – Output is activated only after an unknown/inactive tag is applied.
- **Invert output** – Inverts the output type from NO to NC,
- **Long Open** – Output is activated for a time period defined in this field after a card has been applied to more than 5 seconds:

Function available only for output No. 1, allowing its activation for a longer period of time, up to 38912s.

Function activation:

- Option operates only when the output is set to Pulse mode,
- User's card must be active and assigned to group 4,
- Group 4 does not need to be active, but the card must be assigned to it.

Tip

If the card is not assigned to group 4, Long Open will not be activated.

Example

- Setting the output in **I/O Settings**:

Output Relay 1

Name	Value	Description
Mode	○ Disabled ◉ Pulse ○ Toggle	
Time-on	40	x 0.1s
Action	Known Cards	
Invert output	☐	Invert output relay state to NC
Long Open	10	Enable relay for given time (x 1s) when card is hold for 5 seconds.

- **User2 card:** Assigned to groups 2 and 4 (in the **Cards** tab). Groups 1 and 2 are active at the given moment.

9	User2	000000D57E2014	☑	☐☑☐☑	✓ X

After the User2 card is applied to the reader, output 1 is activated for 4 seconds (0,1*40). If the card remains in the read field for more than 5 seconds, the output will be activated for additional 10 seconds: - Output No. 1 is activated for 4 seconds, - Output is deactivated for 1 second, - Output is activated again for 10 seconds.

```
Sound signaling
- The user will hear the stadard read sound while applying the card,
- If the card is held in the read field form more than 5 seconds, the device will emit single-
tone sounds,
- The last high tone signals the Long Open function activation.

Thanks to this option, relay output No. 1 can be configured for a longer time, which may be
useful in various applications that need a longer opening time.
```

Relay activation on a different Inveo device

The function **Bridge Output Relay 1 to external Inveo device** allows sending of a request to change the output state of another Inveo device. This allows, for example, to open a door bolt that isn't connected to the reader directly. Communication is carried over a LAN.

Bridge Output Relay 1 to external Inveo device		
Name	**Value**	**Description**
Enable	☐	
Server IP	192.168.0.239	A.B.C.D
Port	9761	Default 9761 (UDP)
Channel	1	

- **Enable** – Enable the relay state forwarding function,
- **Server IP** – IP address of the Inveo device that operates the output be controlled by the reader,
- **Port** – Port, on which the remote device listens – default 9761 (UDP),
- **Channel** – Number of the channel in the remote device that will be controlled.

The service forwards the state of relay output 1 to a remote device (eg. LanTick). The relay in the remote device is activated when output 1 is activated in the relay, that's why the settings from **Output relay 1** table are important.

Output Relay 1		
Name	**Value**	**Description**
Mode	○ Disabled ◉ Pulse ○ Toggle	
Time-on	40	x 0.1s
Action	Known Cards	
Invert output	☐	Invert output relay state to NC
Long Open	10	Enable relay for given time (x 1s) when card is hold for 5 seconds.

> **Tip**
>
> **If relay 1 in the reader is set to Pulse mode, it's required to configure the output in the remote device (Time Based mode) in order for the output to operate the same way as the reader. Card application will only cause the output to activate. The remote device must deactivate the relay after defined time.**

Remote module configuration

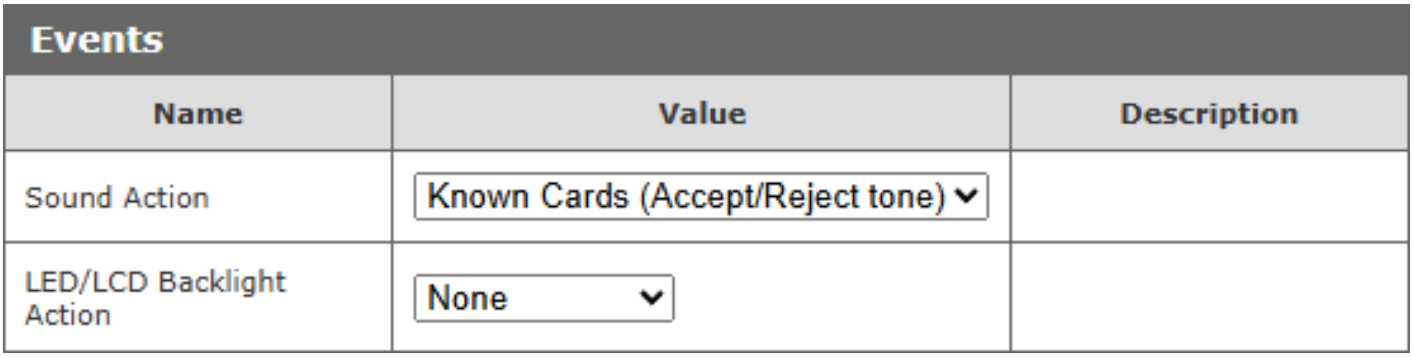
To enable the service, go to the **Administration** tab and select **Enable Program Access**, then **Save Config**.

Audio/visual signaling

The reader has 3 kinds of sound signal:

- **Accept sound** – Two consecutive tones: first one low, second one higher and longer. Positive sound,
- **Reject sound** – Two consecutive tones: first tone like in accept (short tone), second tone lower and longer. Negative sound,
- **Short beep** – Only the short tone from accept/reject.

In the **Events** table, reader behaviour after and RFID tag application can be programmed.



- **Sound action** – Action that triggers the sound signal:
  - **None**,
  - **All Cards (Accept tone)** – Module emits the accept sound after an RFID tag is applied (known or unknown),
  - **All Cards (Short beep)** – Short signal for all tags*,
  - **Known Cards** – Emits the positive sound for known and negative for unknown tags.
- **LED/LCD Backlight Action** – Action that triggers LCD or LED activation:
  - **None** – Control by HTTP, SMNP, Modbus,
  - **All Cards**,
  - **Known Cards**.

* the short sound signal can be used for communication between the reader and server. In this case, the reader indicates tag readout with a short sound, but it's up to the server to signal the processed information.

LCD screen backlight can be controlled by the reader.

LCD		
Name	**Value**	**Description**
Backlight time	255	x1s, 0-always off, 255-always on

Setting a value of 255 will make the backlight glow constantly.
Setting a value of 0 will disable the backlight. Setting a value of eg. 5 will cause the display to light up for 5 seconds after tag application.

Real Time Clock (RTC)

The readers are equipped with a built-in Real Time Clock (RTC).

The time can be set manually in the **Time settings** tab, **Time status** section, using the **Update time** button. The time will be taken off the system. The time zone can be configured with the list. Selecting **Daylight saving** forces the use of summer and winter time.

Time status	
Name	**Value**
Current Time	11:02:39
Current Date	2025-04-01

Update time

Time zone		
Name	**Value**	**Description**
Time Zone:	(GMT +1:00 hour) Brussels, Warsaw, Paris	
Daylight	☑	Enable Daylight Saving
Logger local time	☑	Add local time offset to timestamp and time when log event. If unchecked then log time is GMT.

The module can also synchronize the internal clock with a time server utilizing SNTP. The settings can be configured in **Network** → **SNTP**.

To enable SNTP sychronization, input the SNTP server's address into the **Server** field. Typing in 0.0.0.0 disables time sychronization.

SNTP		
Server	194.146.251.100	A.B.C.D (0.0.0.0 to disable)

SNMP server configuration

The module is equipped with an SNMP v2c server. To enable this function, go to **Administration** → **Services** and **Enable SNMP**.

Enable SNMP	☑	

SNMP allows for:

- Downloading of input state,
- Setting output states,
- Downloading read RFID tag ID's.

MIB file describing the SNMP structure can be downloaded from the **SNMP** tab by clicking **Download MIB file**.

Communication protocols and administration

The **Administration** menu enables configuration of active services and password change.

Module name

Each reader can be assigned with a unique name used for identification.

Module name		
Name	**Value**	**Description**
Module name		0..15 characters

Change password

To change the password:

1. Input the **Current Password**,
2. Input the **New Password** an **Re-type Password**,
3. **Save Config**.

Services settings

Services		
Name	**Value**	**Description**
Autonomic	☑	
Enable MODBUS TCP	☐	
Enable MODBUS RTU	☐	
Enable SNMP	☐	
Enable HTTP GET	☐	
Enable HTTP Client	☐	
Timeout	0	(x0,1s) a reader communication loss timeout after which the reader switches to Autonomic mode.

- **Autonomic** – Autonomous reader mode,
- **Enable Modbus TCP**,
- **Enable Modbus RTU**,
- **Enable SNMP**,
- **Enable HTTP GET**,
- **Enable HTTP Client**,
- **Timeout** – Time, after which the module will enter autonomous mode if it doesn't receive information from a server,
- **Enable Remote Network Config** – Enable remote module configuration (Discoverer programme),
- **Enable TFTP Bootloader**.

Tip

For security reasons TFTP Bootloader and Remote Network Config should be disabled during routine operation. Activate them only before updating firmware.

Communication protocols

HTTP

HTTP GET – read state

RFID modules can be controlled via HTTP on port 80. To read the current state of the module, complete the following steps:

1. Open a web browser,
2. Enter the address: `http://<IP_address>/status.xml`.

An XML resource with basic reader information will display in the browser.

Example XML file

```
<status>
    <model>RFID IND-LCD Mif</model>
    <name/>
    <mac>00:00:00:00:00:00</mac>
    <id>000000D57E2014</id>
    <newId>0</newId>
    <known>1</known>
    <cnt>14</cnt>
    <time>08:50:59</time>
    <date>2026-06-11</date>
    <out0>Off</out0>
    <out1>Off</out1>
    <in0>Off</in0>
    <in1>Off</in1>
    <resetFlag>1</resetFlag>
    <enable>1</enable>
    <group>1234</group>
    <httpClientStatus>0</httpClientStatus>
    <n_logs>4</n_logs>
    <timeout>0</timeout>
    <fw>0.86</fw>
    <hw>1.0</hw>
</status>
```

Section	Description
`<model>`	Module name
`<name>`	User-assigned module name
`<mac>`	Module's MAC address
`<id>`	Last read RFID tag stored in HEX
`<newId>`	In Control only by HTTP GET mode: 1 – a new tag has been read 0 – no new RFID tags
`<known>`	Read tag type 1 – unknown tag 2 – user 3 – master
`<cnt>`	Amount of RFID tags read since last device restart
`<time>`	Current device time
`<date>`	Current device date

▼ Details

`<out0>`	Current relay output No. 1 state
`<out1>`	Current relay output No. 2 state
`<in0>`	Current input No. 1 state
`<in1>`	Current input No. 2 state
`<resetFlag>`	1 – a module reset has occured
`<enable>`	1 – RFID module enabled 0 – RFID module disabled
`<group>`	Currently active group numbers
`<httpClientStatus>`	Current TCP connection state in Control only by HTTP Client mode 1 – connected with a server – socket open 2 – received data from a server 3 – connection terminated 100 – no connection with server
`<n_logs>`	Current amount of logs in the reader
`<timeout>`	Time since last connected with a server
`<fw>`	Firmware version
`<hw>`	Hardware revision

HTTP GET – control

Controlling the module in **Enable HTTP GET** mode involves sending appropriate commands to the reader using HTTP.

> **Note**
>
> **The `/status.xml` resource enables control over basic reader functions, such as LED indicators, sound signal. Additionally, the read flags can be deleted via this resource.**

`http://<IP_address>/status.xml?`

Command	Value	Description
`enable`	BOOLEAN	Enable/disable the RFID antenna
`resetFlag`	BOOLEAN	Flag set to 1 after module restart. Writing `0` removes the flag
`releaseId`	1	Delete the read flag and await RFID tag application
`ledr`	TimeOn,TimeOff,Cnt	Activate the red LED. eg. `"ledr":5,5,5` will activate the LED for 0,5s, deactivate for 0,5s and repeat the whole sequence five times **Applies only to RFID IND LED**
`ledg`	TimeOn,TimeOff,Cnt	Activate the green LED. eg. `"ledr":5,5,5` will activate the LED for 0,5s, deactivate for 0,5s and repeat the whole sequence five times **Applies only to RFID IND LED**
`buzz`	a - accept r - reject	Manual buzzer control
`open`	INTEGER	Activate the relay output No. 1 (Mode defined in **I/O Settings**)
`open2`	INTEGER	Activate the relay output No. 2 (Mode defined in **I/O Settings**)
`out0`	BOOLEAN	Relay output No. 1 control
`out1`	BOOLEAN	Relay output No. 2 control
`takeLcd`	INTEGER	Assume LCD control
`showTime`	INTEGER	Display current time in the first LCD row

`lcdClr`	INTEGER	Clear the LCD contents
`lcd1`	STRING	Display text in the 1st LCD line
`lcd2`	STRING	Display text in the 2nd LCD line
`lcd3`	STRING	Display text in the 3rd LCD line
`lcd4`	STRING	Display text in the 4th LCD line
`ring`	1 – deactivate function key backlight 2/3 – left/right function key backlight 4/5 – flash the left/right function key backlight 6/7 – continuous blinking of the left/right function key 8 – continuous blinking of both function keys	Function key backlight. **Applies only to the RFID RCP with keys**
`enter`	1 - private 2 - official 4 - entry 8 - exit	Change the Entry/Exit type. Sum of particular flags. np. Sending `"enter":9` will change the flag to a private exit. **Applies only to the RFID RCP with keys** **Warning!** Sending this command will only cause the entry/exit mode to change. To additionally change the key backlight colour, send the `ring` command too.
`bl`	2 - resume autonomic control 1 - enabled 0 - disabled	LCD backlight control

Note

The `/msg.php` resource allows the user to conduct card database and log operations. Reader settings, such as RTC time, can also be amended.

`http://<IP_address>/msg.php?`

Command	Value	Description
`setLog`	INTEGER	Set current log marker. eg. `"setLog":10` sets the marker to 10
`removeCard`	INTEGER	Remove selected card from memory. eg. `"removeCard":5` deletes card No. 6 (card number minus 1)
`removeAllCards`	INTEGER	Remove all RFID tags from the memory
`removeLog`	INTEGER	Remove all logs from the reader
`removeNLog`	INTEGER	Remove x oldest logs from the device's emmory eg. `"removeNLog":3` will delete 3 oldest logs
`clkY`	x=[0-99]	Set RTC year
`clkM`	x=[0-11] 0 - January 1 - February etc.	Set RTC month
`clkD`	x=[1-31]	Set RTC day
`clkH`	x=[0-23]	Set RTC hour
`clkm`	x=[0-59]	Set RTC minutes
`clkS`	x=[0-59]	Set RTC seconds
`clkd`	x=[0-6] 0 - Sunday 1 - Monday etc.	Set RTC weekday
`factory`	INTEGER	Restore factory settings

`addId`	INTEGER	Manually add a card to memory. ID in HEX format. eg. `"addId":1122334455` will add a card with ID 1122334455. **Note:** If sending an ID (PIN code) with an odd number of characters, precede them with a 0
`changeNam e`	STRING	Manually change the card name. x - card number from **Cards** minus 1 eg. `"changeName":9!Jan` changes the name of card No. 10 to Jan
`flag`	INTEGER	Manually edit card permissions. x - card number from **Cards** minus 1 y - sum of permissions: 1 - specifies whether a card is added to the reader 2 - card active (2) or inactive (0) 4 - assign card to group 1 8 - assign card to group 2 16 - assign card to group 3 32 - assign card to group 4

Editing cards via HTTP POST

The reader allows editing of cards via HTTP POST method. The data structure is located in `edit.xml`:

id: 6
card: 00
name: User2
act: 1
grp: 0100

- `id` – Card number from **Cards** minus 1,
- `card` – TAG,
- `name` – User name,
- `act` – Specifies whether the card is **Active** (0 or 1),
- `grp` – Card group assignment (0 means a card is not assigned, 1 means card assigned to a group):

0001 – card assigned to group No. 4,
0111 – card assigned to groups No. 2, 3 and 4.

The POST method data structure has the following form:
`id=value&card=00&name=value&act=value&grp=value`

Example

`id=6&card=00&name=User2&act=1&grp=0100`

Card can be edited by changing appropriate parameters. HTTP POST allows the user to add a new card by using `id=-1`. The reader will add the card to the first available position.

Tip

It's not possible to edit the UID of a card stored in the reader.

Card edit example

To edit the card, use the cURL programme. cURL enables sending of HTTP requests from the Windows command line. Syntax:

```
curl -u admin:admin00 -d "id=6&card=00&name=John&act=1&grp=0111" -X POST
http://192.168.0.222/edit.html
```

- `-u admin:admin00` – User authorization, user:password,
- `-d "id=6&card=00&name=John&act=1&grp=0111" -X POST` – Card edit request:
  - `id=6` – Card No. 7,
  - `card=0` – Card UID can't be changed,
  - `name=John` – Assign new user name,
  - `act=1` – Force new card activation,
  - `grp=0111` – Assign card to groups No. 2, 3 i 4,
  - `http://192.168.0.222/edit.html` – IP address of the module and the resource that should be recalled.

Add a card example

To add a card to the reader's memory, use the cURL programme. Syntax:

```
curl -u admin:admin00 -d "id=-1&card=0600A4638D&name=Ahr&act=1&grp=1001" -X POST
http://192.168.0.222/edit.html
```

The request shown above will add a new card to the first available position in the reader's memory.

- `id=-1` – Force add new card.

HTTP Client

Module control in HTTP Client mode is possible after selecting **Enable HTTP Client** in**Administration**. If the reader has **Enable HTTP Client** mode activated, configure the address the data will be sent to. To do that, use the **Network → HTTP Client Configuration** table.

HTTP Client Configuration		
Server IP	0.0.0.0	A.B.C.D
Server Name		Server DNS-registered name. Set 'Server IP' to 0.0.0.0 if used.
Port	0	
Resource		HTTP pool resource ie. / or /something.php
Poll time	50	x100ms, 0-send only changes

- **Server IP** – IP address the reader will send data to,
- **Server name** – Domain address name – if it's registered on a DNS. If the domain address is inputted ino **Server IP**, fill this field with 0.0.0.0,
- **Port** – Port, the server is listening on,
- **Resource** – Resource the reader recalls, eg. /somefile.php,
- **Poll time**:
  - Poll time=0 - data is sent only when an RFID tag is applied,
  - Poll time>0 - data is sent periodically and after an RFID tag is applied. eg. Poll time=50 - data will be sent every 5 seconds and after an RFID tag is applied.

Sending data to the server

The module sends data to the corresponding server resource depending on whether the reader has read the TAG or because of periodic information exchange.

If the reader has read an RFID tag, the data will be sent in the following format:

- A – MAC address,
- B – IP address,
- C – Reader name (assigned in the **Administration** tab),
- D – UID of the applied RFID card,
- E – Applies only to RFID RCP with buttons – function key flag (private/official entry/exit),
- F – Timestamp,
- G – Active inputs/outputs information,
- H – Card application or removal flag.

```
mac=A&ip=B&name=C&id=D&inout=E&ts=F&io=G&H=1
```

Sample data sent to the server after a card has been applied

```
?mac=000000000000&ip=192.168.111.15&name=&id=00000000&inout=6&ts=1589505038&io=8&put=1
?mac=000000000000&ip=192.168.111.15&name=&id=00000000&inout=6&ts=1589505038&io=12&put=1&away=1
```

During periodic information exchange, the data is sent to the server in the following format:

- A – MAC address,
- B – IP address,
- C – Reader name (assigned in the **Adminsitration** tab),
- F – Timestamp,
- G – Active inputs/outputs information.

```
mac=A&ip=B&name=C&ts=F&io=G
```

Sample data sent periodically

```
?mac=000000000000&ip=192.168.111.15&name=&ts=1589505038&io=0
?mac=000000000000&ip=192.168.111.15&name=&ts=1589505038&io=7
```

- E – Flag specifying which function keys have been selected (private/official entry/exit) The value is sent as a sum of the components:

Private **0 or 1**	**Officail** **0 or 2**	**Entry** **0 or 4**	**Exit** **0 or 8**	**D**
1	0	4	0	5
0	2	4	0	6
1	0	0	8	9
0	2	0	8	10

Examples

1. Private entry. `inout=5`,
2. Official entry. `inout=6`,
3. Private exit. `inout=9`,
4. Official exit. `inout=10`.

- G – State of activated inputs and outputs sent as a sum of the components:

In1 0 or 1	In2 0 or 1	Out1 0 or 4	Out2 0 or 8	E 0 - 15
1	0	4	0	5
0	0	4	8	12
1	2	4	8	15

Examples

1. Input 1 and output 1 are activated. `io=5`, 2. Outputs 1 and 2 are activated. `io=12`,
   3. Inputs 1 and 2, and outputs 1 and 2 are activated. `io=15`.

- H – Card application or removal state:
- `put=1` – Card applied,
- `away=1` – Card removed (only in **Continuous reading** mode),
- `put=1&away=1` – Card applied and instantly removed (only in **Continuous reading** mode).

Only for Mifare RFID readers:

If the module has block readout enabled, read block information can be added: `block=1234567890ABCDEF1234567890ABCDEF`.

In response, the server can send nothing or reply with a XML resource:

Command	Description
`<time>`	1 – display time on the LCD 0 – do not display the time on LCD
`<clear>`	1 – clear the LCD
`<text>`	Display text on the LCD. The next line is started after 20 subsequent characters. If, for example, the second line should start with HELLO, insert 20 space characters before the word HELLO
`<textxy>`	Display text on the LCD in a defined position. Example syntax: xXXyYY_TEXT np. x05y02_HELLO
`<ledg>`	Activate green LED Syntax: TimeOn,TimeOff,Cnt (like in HTTP GET) **Applies only to RFID IND LED**
`<ledr>`	Activate red LED Syntax: TimeOn,TimeOff,Cnt (like in HTTP GET) **Applies only to RFID IND LED**

`<open>`	1 – activate relay No. 1 in the mode defined in **I/O Settings**
`<open2>`	1 – activate relay No. 2 in the mode defined in **I/O Settings**
`<out0>`	0 – deactivate relay No. 1 1 – activate relay No. 1
`<out1>`	0 – deactivate relay No. 1 1 – activate relay No. 1
`<buzz>`	1 – activate sound signal (ACCEPT sound) 2 – activate sound signal (REJECT sound)
`<ring>`	**Applies only to RCP with buttons** 1 – deactivate function key backlight 2/3 – left/right function key backlight 4/5 – flash the left/right function key backlight 6/7 – continuous blinking of the left/right function key 8 – continuous blinking of both function keys
`<bl>`	LCD backlight on – backlight activated off – backlight deactivated time [s] – timed backlight

When operating the reader in HTTP Client mode, the HTTP Response code can be used to control interaction with the user. In the **RFID Settings** tab, automatic reaction to the 3 basic HTTP response codes (200, 401, 404) can be set – **HTTPClient IO control by code**

In this configuration, the reader will react depending on the code number that it receives:

- Code 200: generate the „accept” sound and blink the green LED,
- Code 401: generate the „reject” sound and blink the red LED,
- Code 404: generate the „reject” sound an blink both LEDs.

The XML file can contain the following fields (the example sounds the accept sound, opens the door and displays text on the LCD)

```
<buzz>1</buzz>
<clear>0</clear>
<text>Enter please</text>
<open>1</open>
```

Activate the green LED for 2 seconds:

```
<ledg>20,0,1</ledg>
```

Blink the red LED 2 times with a time of 0.5/0.5 seconds:

```
<ledr>5,5,2</ledr>
```

The XML file syntax is not checked. Only information between the known tags is browsed.

Php server support example:

```php
<?php
if( $_GET["id"] ) // module send id and MAC - $_GET["mac"]
{
    $who=$_GET["id"];
    // you can check id in DB and do some action
    echo "<buzz>1</buzz>"; // sound signal
    echo "<clear>1</clear>"; // clear LCD
    echo "<text>Card ID: $who</text>" // print ID on LCD
    echo "<open>1</open>"; // door open
}
else // no id - default state polling
{
    echo "<clear>1</clear>"; // clear LCD
    echo "<text>Hello</text>"; //print prompt text
}
```

Modbus

Modbus is available via RS485 (Modbus RTU) or TCP (Modbus TCP). RS485 port settings are located in the **Administration** tab.

> **Information**
>
> **For Modbus TCP, the device supports up to two simultaneous connections.**

RS485 Parameters (Modbus RTU)		
Name	**Value**	**Description**
PDU	1	
Baudrate	9600	bps
Parity	None	

- **PDU** – Device's Modbus address,
- **Baudrate** – Transmission speed:
  - 1200,
  - 2400,
  - 4800,
  - 9600,
  - 19200,
  - 38400,
  - 57600,

    - 115200.
- **Parity**:
    - None,
    - None and 2 stop,
    - Even,
    - Odd,
    - Mark,
    - Space.

After all parameters have been set, enable the Modbus RTU service and **Save** the configuration.

Services		
Name	**Value**	**Description**
Autonomic	☐	
Enable MODBUS TCP	☐	
Enable MODBUS RTU	☑	

The device supports the following Modbus RTU functions:

- 0x01 Read Coils
- 0x03 Read Holding Register
- 0x05 Write Single Coil
- 0x06 Write Single Register
- 0x0F Write Multiple Coils
- 0x10 Write Multiple Registers

Holding Registers addressing

RFID control

Address	R/W	Description
1	R/W	Read flag Read: 1 – a new transponder has been read Write: 0 – read flag zeroize
2	R	ID_LEN
3	R	Card ID/UID/CSN [0]

4	R	Card ID/UID/CSN [1]
5	R	Card ID/UID/CSN [2]
6	R	Card ID/UID/CSN [3]
7	R	Card ID/UID/CSN [4]
8	R	Card ID/UID/CSN [5]
9	R	Card ID/UID/CSN [6]
10	R	Card ID/UID/CSN [7]
11	R	Card ID/UID/CSN [8]
12	R	Card ID/UID/CSN [9]
13	R	Tag type
14	R	ID_MODEL
15	R	ID_SW
16	R	ID_HW
17	R	Read tag type: 0 – none 1 – unknown TAG 2 – USER 3 – MASTER
100-115	R	Only RFID modules supprorting Mifare – read a block of 16 subsequent bytes
1000	R	Card ID/UID/CSN [0]
1001	R	Card ID/UID/CSN [1]
1002	R	Card ID/UID/CSN [2]
1003	R	Card ID/UID/CSN [3]
1004	R	Card ID/UID/CSN [4]

1005	R	Device model (IND-U4 → 0x5534)
1006	R	Software version
1007	R	Hardware version

The Holding Registers 1000-1007 addresses have been distributed this way for backwards compatibility.

Card records amendment

Addres	R/W	Description
1100	R/W	Amended card record address
1101	R/W	Card settings: described below
1102	R/W	Card ID/UID/CSN [0]
1103	R/W	Card ID/UID/CSN [1]
1104	R/W	Card ID/UID/CSN [2]
1105	R/W	Card ID/UID/CSN [3]
1106	R/W	Card ID/UID/CSN [4]
1107	R/W	Card ID/UID/CSN [5]
1108	R/W	Card ID/UID/CSN [6]
1109	R/W	Card ID/UID/CSN [7]
1110-1149	R/W	User name, must end with NULL (0x00)

Card settings (1101 Holding Registers):

If the user card is read or modified, Holding Register 1101 is responsible for storing the card in reader's memory, its activation and the groups that are assigned to this card.

Bit	dec	Function
0	1	Specifies whether a card is stored in the reader's memory

1	2	Specifies whether a card is active: 0 – card inactive 1 – card active
2	4	Group No. 1 assignment
3	8	Group No. 2 assignment
4	16	Group No. 3 assignment
5	32	Group No. 4 assignment
6	64	Reserve
7	128	Reserve

Example of values with explanation:

- Delete card from reader's memory – value 0,
- Simultaneous card deactivation and assignment to group No. 1 – value 5,
- Asignment of card to all groups – value 63.

Log records

Address	**R/W**	**Description**
1200	R/W	Amount of records [MSB], sending 0 deletes all logs
1201	R	Amount of records [LSB]
1202	R	Selected record to be read [MSB]
1203	R	Selected record to be read [LSB]
1204	R	Timestamp [LSB]
1205	R	Timestamp
1206	R	Timestamp
1207	R	Timestamp [MSB]

1208	R	Event Event 0x0080 – known card Event 0x0040 – card removed
1210	R	User ID (from the cards table)
1212	R	Card ID/UID/CSN [0]
1213	R	Card ID/UID/CSN [1]
1214	R	Card ID/UID/CSN [2]
1215	R	Card ID/UID/CSN [3]
1216	R	Card ID/UID/CSN [4]
1217	R	Card ID/UID/CSN [5]
1218	R	Card ID/UID/CSN [6]
1219	R	Card ID/UID/CSN [7]

Read card control

Addres	R/W	Description
1300	R	Card record address
1301	R	Card settings
1302	R	Card ID/UID/CSN [0]
1303	R	Card ID/UID/CSN [1]
1304	R	Card ID/UID/CSN [2]
1305	R	Card ID/UID/CSN [3]
1306	R	Card ID/UID/CSN [4]
1307	R	Card ID/UID/CSN [5]
1308	R	Card ID/UID/CSN [6]

1309	R	Card ID/UID/CSN [7]
1310-1349	Holding Reg	R

> **Tip**
>
> **1300-1349 Holding Registers contain the same data as 1100-1149 Holding Registers. The difference is that the addresses 1300-1349 are read-only.**

LCD

Address	R/W	Description
2000	R/W	LCD Mode: 0 – display controlled autonomically 1 – Modbus display control
2001	R/W	LCD TIME, if (LCD Mode=1): 0 – do not display time on LCD 1 – display time on LCD
2002	R/W	LCD Clear: 1 – clear LCD contents
2010-2019	W	LCD 1st line (only if LCD TIME = 0)
2020-2029	W	LCD 2nd line
2030-2039	W	LCD 3rd line
2040-2049	W	LCD 4th line
2050	W	LCD backlight (LCD Mode=1): 0 – disable backlight 1 – enable backlight >1 - backlight time in 0.1 second units

2 characters should be sent to one Holding Register address. To write "John" from the beginning of the second display row, convert the characters using an ASCII table:

- J = hex 4A,
- o = hex 6F,
- h = hex 68,
- n = hex 6E.

After that, send 4A6F to address 2020 and 686E to address 2021. Space character is represented by 0x1E.

Single Coil addressing

I/O

Address	R/W	Description
1	R/W	State/set output No. 1
2	R/W	State/set output No. 2
3	R	Output 1 state
4	R	Output 2 state
11	R	Output 3 state
12	R	Output 4 state
13	R/W	LED control
14	R/W	LED control
15	W	1 – „Accept” sound
16	W	1 – „Reject” sound
17	R/W	Read flag Read: 1 – a new transponder has been read Write: 0 – zeroize read flag
18	R/W	Reset status Read: 1 – a reader restart has occured (eg. power loss) Write: 1 – force device reset 0 – zeroize restart flag
1000	R	Output 1 state – (door)
1001	R	Input 1 state
1002	R	Input 2 state
1003	R	Reserve
1004	R/W	Read flag Read: 1 – a new transponder has been read

		Write: 0 – zeroize read flag
1005	R/W	Reset status Read: 1 – a reader restart has occured (eg. power loss) Write: 1 – force device reset 0 – zeroize restart flag
1010	W	1 – activate door bolt relay
1011	W	1 – „Accept” sound
1012	W	1 – „Reject” sound
1013	R/W	LED control
1014	R/W	LED control

Tip

After a tag has been read correctly, the Single Coil register 1004 will be written with 1 – a new tag has been read. Reading of next transponder is possible only after the read flag has been reset (by writing Single Coil 1004 with a 0). Flag reset is not required if the module is set to Continuous Reading.

Integration with own software

Inveo RFID readers can be integrated with own software, operating in server mode (**Administration > Enable HTTP GET**) or client mode (**Administration > Enable HTTP Client**)

Server mode (HTTP Server)

In this mode, the reader operates as a server. A remote host (client) connects with the module and manages it using HTTP GET. To enable this mode, select **Administration → Enable HTTP GET**

- Read status: the client periodically reads the `status.xml` file, decodes the XML tag it's interested in and returns commands to the reader when needed. The `status.xml` tag contains all required information that can be saved to a database, such as the reader's MAC address, input and output state, and card ID.
- User interaction: After using the data, the client can send commands to the reader, such as accept/reject sound, LED blink, door bolt release or LCD text. The reader blocks the readout of subsequent tags until the client recalls `releaseId=1`.
- Sound signal: If the communication isn't happening in real-time, the reader can emit a short sound after a tag has been read, informing the user about the fact. The user does not need to hold the card in the read field indefinitely – only until they hear a sound effect.

Client mode (HTTP Client)

In this mode, the reader operates similarly to a web browser. It sends requests to the server and awaits replies. The server awaits requests, processes them and sends an appropriate answer. To activate this mode, go to

Administration → Enable HTTP Client.

- Sending data: the module automatically connects with the server and sends the read tag information via HTTP GET. The server processes the request and sends a reply in the form of an XML file containing commands that define module operation: LED control, sound signal, LCD screen.
- Direct integration: The module sends a code to the server or a controlling application by itself right after a card is read. The client-server mode enables easy integration with PHP, Node JS, MySQL servers.

Factory settings

Restoring factory settings

To restore the factory settings:

- Power up the device,
- Short the jumper (RFID IND) or press the **RESET** button (RFID RCP) for a time period between 10 and 15 seconds,
- Remove the jumper or release the button during the sound signal.

Once the above steps have been completed, the device will be set to the following parameters:

- **IP address:** 192.168.111.15
- **IP mask:** 255.255.255.0
- **User:** admin
- **Password:** admin00

Warranty and manufacturer's liability

> **Warning**
>
> **The manufacturer provides a two-year warranty for the device and post-warranty service for a period of 10 years from the date of introduction of the device to the market. The warranty covers all material and production defects.**

The manufacturer undertakes to respect the warranty agreement, if the following conditions are met:

- All repairs, changes, expansions and device calibrations are carried out by the manufacturer or an authorized service center,
- The power supply system meets the applicable standards,
- The device is operated in accordance with the suggestions presented in this manual,
- The device is operated in accordance with its intended purpose.

The manufacturer assumes no responsibility for consequences resulting from improper installation, improper use of the device, failure to comply with the instruction manual, and repairs made by unauthorized personnel.

> **Warning**
>
> **The device contains no user serviceable parts inside.**

Storage, operation and transport conditions

The device should be stored in enclosed rooms, where the atmosphere is free from vapours and corrosive substances:

- Environment temperature from -30°C to +60°C (-22°F - 140°F),
- Humidity from 25% to 90% (condensation unacceptable),
- Atmospheric pressure from 700 to 1060 hPa.

The device is intended to operate in the following conditions:

- Environment temperature from -10°C do +55°C (14°F - 131°F),
- Humidity from 30% to 75%,
- Atmospheric pressure from 700 to 1060 hPa.

Recommended transport conditions:

- Environment temperature from -40°C do +85°C (-40°F - 185°F),
- Humidity from 5% to 95%,
- Atmospheric pressure from 700 to 1060 hPa.

Installation and device operation:

- The module should be operated in accordance with recommendations provided later in this manual.

Disposal and decommissioning

In an event the device needs to be decommissioned (eg. after its intended life period is surpassed), it is recommended to contact the manufacturer or his representative, who are responsible to respond appropriately, i.e., to collect the device from the user. The user can alternatively contact companies specializing in electronic device or computer equipment disposal and/or decommissioning. Under no condition should the device be placed with other waste.