Intelligent cloud RFID reader with active PoE supply

Embedded Electronics
&
Solutions, s.r.o.

www.eeas.cz

User manual

Dec 07 2023

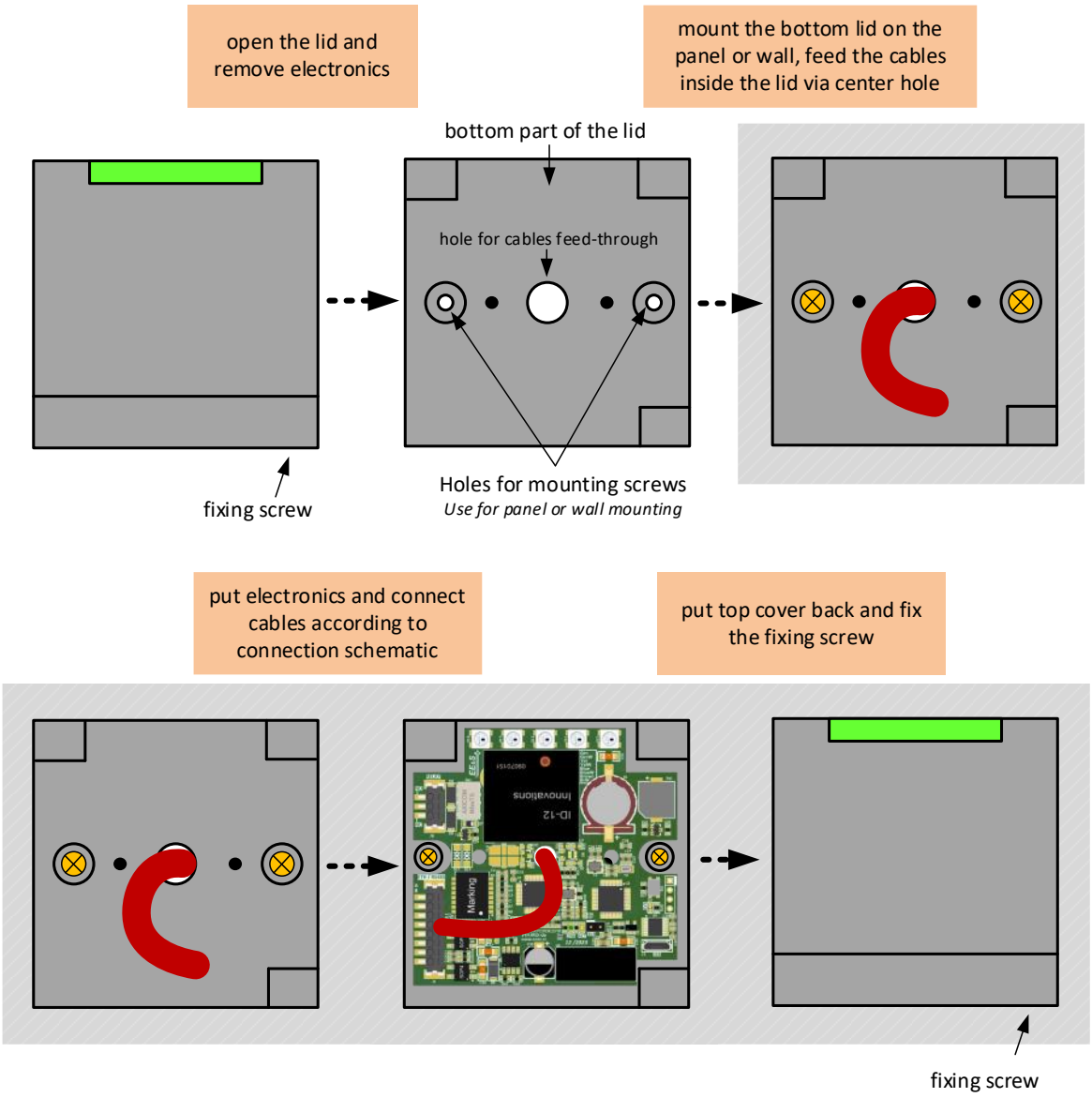
1. Device parameters

Power supply	**Power Over Ethernet PoE** (802.3af / 802.3at Type 1 or Type 2)
Nominal PoE supply voltage	+48 VDC
PoE standard	**active only** based on 802.3af / 802.3at Type 1 or Type 2
Power consumption	max. 2 W **(Powered Device Class 1)**
Relay contact voltage	max 36 VDC
Relay contact current	max 2A
Relay contact power	max 60W / 62,5 VA
Timekeeping battery	CR1220 standard cell (3V)
USB interface (VCP)	115200/8/N/1
RFID standards	125 kHz ISO/EM4001
Ambient temperature	-20 °C up to 50 °C
Ambient humidity	0 up to 95 % non condensating

2. Factory settings

IP adress	192.168.0.100
Subnet mask	255.255.255.0
Gateway	192.168.0.1
PHP port	80
Script URL	
Password	admin
Hash MD5	disabled
Hash salt	salt
Card buffering	disabled
Heart beat period	60 seconds
Reader ID	0x00000000
Configuration port	5000
HTTP request port	80
LED intensity	50

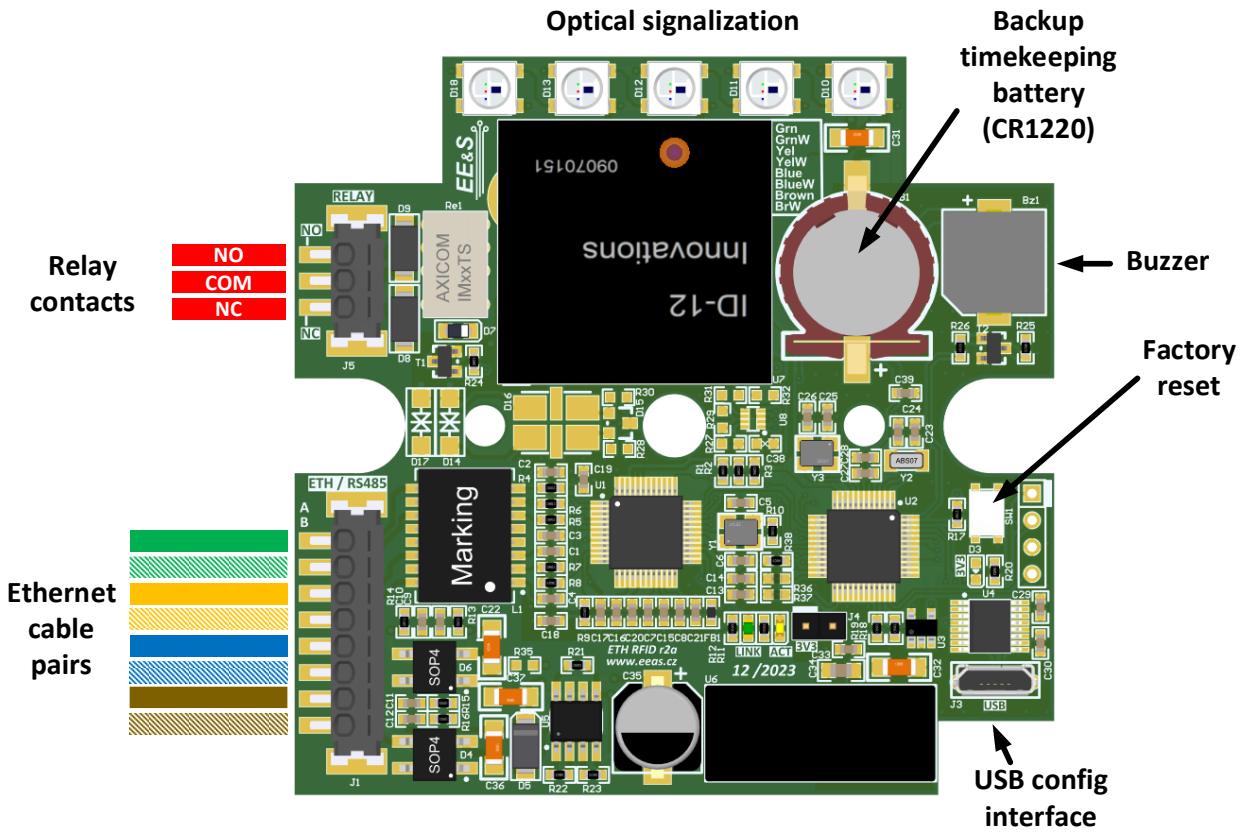
3. Basic mechanical installation



4. Description

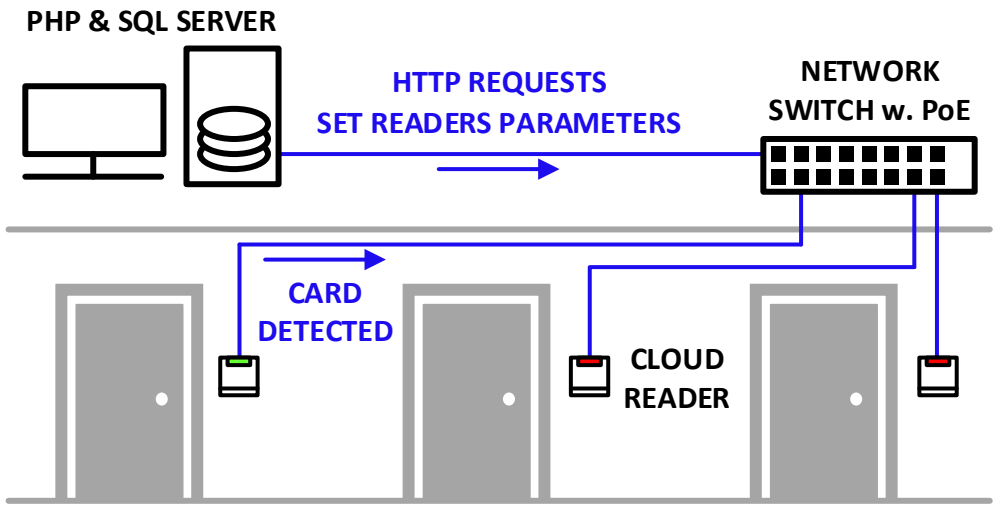
The cloud reader is an electronic device providing a reading of 125 kHz ISO/EM4001. The read card`s IDs are transmitted via HTTP protocol into the cloud or PHP server. Parameters of the HTTP requests are modifiable via raw TCP connetion using proprietar communication protocol described further in the manual. The reader is also listening to the HTTP request from PHP server and reacts to the commands transmitted through the request.

5. Front view and connection schematic



6. Connection

The cloud reader is compatible with the IEEE Power-Over-Ethernet standard in active form. This means that the reader should be connected to the industry standard active PoE injector/router or switch. Typical connection diagram is shown below:



In addition to the LAN / ETH connection, there is a relay contact port on the device for easy controlling of low-voltage DC or AC loads like magnetic locks, light etc.

7. Factory reset

If user wants to issue a factory reset, the marked switch SW1 should be pressed and hold until **RED** signalization starts flashing. Releasing the button after this signalization will recall the factory settings and reset of the device will occur.

8. Timekeeping feature

Device provides internal real-time clock source which is used for timestamping purposes of the communication protocol. The real-time clock has back-up CR1220 battery. The typical lifetime of this battery is more than 7 years, depending on the various parameters such as ambient temperature, quality of lithium battery.

9. Communication interfaces

The reader is equipped with three comunnication interfaces. A raw TCP/IP server listing at port 5000 for reader configuration. The server also provides configuration of the HTTP request. An HTTP interface is intented for use in standard servise. An USB interface is used as backup configuration interface and also for reader firmware update.

9.1. Configuration TCP server

The TCP server serves for a reader configuration. The user can connect to the server using a PC terminal such as netcat etc. A factory preset IP addrress and port are stated in the chapter 2. After succesful opening the comunication port user can write commands to the reader. List of all supported commands is stated in chapter 10. Each command have to be terminated by line feed character (\n, 0x0A). An example of a start configuration of the reader is stated in a listing below. User commands are stated with **bold** characters and the reader reply with plane characters. The line feed character is shown as \n.

UNLOCK=admin\n
OK\n

PASS=pasword\n
OK\n
CLOCK=2016,10,1,12,34,56\n
OK\n
READID=1A2B3C4D\n
OK\n
IPADR=192,168,1,25\n
OK\n
SNMASK=255,255,255,0\n
OK\n
GWADR=192,168,0,1\n
OK\n
WSADR=216,58,201,99\n
OK\n
WSURL=/rfid_script.php\n
OK\n
HASH=1\n
OK\n
SALT=salt\n
OK\n
RESET\n

9.2. HTTP interface

The HTTP interface serves for a normal operation of the reader. The interfaces implements a heart beat request, an asynchornous request when an RFID card was detected and receiving requests from host server.

9.2.1. Heart beat

If a heart beat request is enabled (see chapter 10.13) the reader periodicaly sends a HTTP request to the host server with a couple of parameters. An example of the HTTP heart beat request is shown on a listing bellow.

```
GET
/rfid_script.php?hb=1&ts=20161018092311&readerid=
00000001&hash=E82EE47E3F15470F8A6D30C21B80F2C2
HTTP/1.1\n
Host: 216.58.201.99\n
```

```
Connection: close\n
\n
```

The request includes following parameters

hb	heart beat indicator,
ts	time stamp in format yyyyMMddHHmmss,
readerid	ID of the reader sending the heart beat,
hash	hash code computed from all parameters.

If the hash code computation is enabled (see chapter 10.11 and 10.12) the hash is computed following way. The MD5 algorithm is used. The input string is assembled from the parameters *hb*, *ts*, *readerid* and *salt* in ACSII form without any separating characters. If salt string is set to “salt” then the hash input string from the HTTP request example above is

120161018092311**00000001**salt

The time period of sending the heart beat is adjustable by command desribed in chapter 10.13. The host server IP address and the script URL are adjustable using the commands described in chapter 10.6 and 10.8.

9.2.2. Asynchronous HTTP request from reader

When an RFID card was detected and succesfully read an anysonchornous HTTP request is sent to the host server. An example of the request is show on a listing bellow.

```
GET
/script.php?card=09002518D4E0&card_i=2431188&ts=2
0161018094102&readerid=00000001&auth=1&hash=93148
62E0A78E896BC887E887085091B HTTP/1.1\n
Host: 130.193.9.19\n
Connection: close\n
\n
```

The request includes following parameters

card	RFID card ID in hex format,
card_i	RFID card ID in decimal format,
auth	*DESFire authentication result*
ts	time stamp in format yyyyMMddHHmmss,
readerid	id of the reader sending the request,

hash hash code computed from all parameters.

The *card_i* parameter is computed as a decimal form of middle four bytes of the paramter *card*. For numbers from example above the procedure is following.

In the case of of LF RFID reader version the *card_i* paramters is computed as follows:

card 09**002518D4**E0

to decimal

card_i 2431188

In the case of of HF RFID reader version the *card_i* paramters is computed from the whole UID:

card 8BC91DA0

to decimal

card_i 2345213344

or

card 04314EFA3C4B80

to decimal

card_i 1180115182308224

In the case of HF RFID reader version the *auth* parameter provides information about DESFire authentication results. The parameter can get three values as follows:

auth=0	card is not desfire type or authentification not applicable
auth=1	card is desfire type and authentification passed
auth=2	card is desfire type and authentification failed

The reader implements the 2k3DES algorithm with a fixed length 16 bytes key. The DESFire authentiction key is set via command described in chapter 10.16. This feature is available only after Nov 10 2017 FW build date. The FW build date culd be find via command described in the chapter 10.31. In the case of LF RFID version the *auth* paramter is not transmitted at all.

The hash code (if is enabled) is computed the same way as was described in chapter 9.2.1 from the parameters *card*, *card_i*, *ts* and *readerid*. The host server IP address and the script URL are adjustable using the commands described in chapter 10.6 and 10.8.

In the body of the HTTP response from the host server there could be included commands for the reader. The format of the HTTP body is discused in the chapter 9.3.

The reader optionally provides a buffering of the detected cards. This function could be eneable by a command desribed in chapter 10.14. If the card buffering is enabled the reader buffers the detected cards and sequentially sends the ansychronous requests as the server reply of the previuos card was received. If the buffer is full the detected cards are discarded. If the card buffering is disabled all cards detected during the wait for reply period (see chapter 10.30) are discarded. The next card is proceed after the server reply is received or the reply timeout is exceeded.

9.2.3. Asynchronous HTTP request from host server

If there is a need to address the reader asynchronously from the host server the reader is listening to the port 80 for incoming HTTP requests. For example the host server executes the following link

```
http://192.168.0.25/?CMD=1&TOKEN=aaaa
```

with parameters

CMD	specify to the reader to expect incoming commands,
TOKEN	token expected in reader reply.

After the request is received the reader sends another HTTP GET request directly to the host server with IP and URL set by the commands described in chapters 10.6 and 10.8. The format of the request is following

```
GET
/script.php?CO=1&TOKEN=aaaa&ts=20161018094348&rea
derid=00000001&hash=9AF6601BBA054EE1B6C07E27B9793
C8A HTTP/1.1\n
Host: 130.193.9.19\n
Connection: close\n
\n
```

with parameters

CO	specify to the server to prepare commands,
TOKEN	token from host server HTTP request,
ts	time stamp in format yyyyMMddHHmmss,

readerid	id of the reader sending the request,
hash	hash code computed from all parameters.

The hash code (if is enabled) is computed the same way as was described in chapter 9.2.1 from the parameters *co*, *token*, *ts* and *readerid*. The host server IP address and the script URL are adjustable using the commands described in chapter 10.6 and 10.8.

In the body of the HTTP response from the server there could be included commands for the reader. The format of the HTTP body is discused in the chapter 9.3.

9.3. HTTP body commands formating

The host server sends commands to the reader in the body of the HTTP reply. The format of the reply is the same for both cases described in chapters 9.2.2 and 9.2.3. After the host server received an HTTP request from the reader the host have to reply to the reader with a batch of commands. For example flashing the leds or switching on the relay. The commands have to be inlcuded in the body of the HTTP response surrouded by <EEAS> and </EEAS> tags. All commands have to be terminated by line feed character (\n, 0x0A). The reader accepts commands described in the chapter 10.

An example of the reply is shown in the listing bellow. It shows a basic response commanding to flash green led for 1 second and beep the buzzer. An example of the script implementation itself is stated in chapter 11.

```
<EEAS>\n
LEDG=10\n
BUZZER=10\n
</EEAS>
```

9.4. USB interface

The USB interface is used only for service reasons. In a normal operation it is not neccesary to use it.

If you connect the reader via USB cable to the PC, the reader acts as a virtual COM port with paramters 115200/8/N/1. The reader implements the same set of commands as in the case of TCP configuration server, so user could also use the USB interface to configure the reader.

10. List of commands

10.1. Unlock

Unlock read-only mode and allow to set up reader parameters.

UNLOCK=<*password*>

password	password, default value = admin

10.2. Clock

Update internal real time clock timer bz current date and time.

CLOCK=<*yyyy*>,<*MM*>,<*dd*>,<*HH*>,<*mm*>,<*ss*>
CLOCK=2016,1,1,12,0,0

yyyy	year
MM	month
dd	day
HH	hours
mm	minutes
ss	seconds

Query:
CLOCK?

Reply:
<*yyyy*>,<*MM*>,<*dd*>,<*HH*>,<*mm*>,<*ss*>

10.3. IP address

Set reader IP address. The changes will be applied after Reset command.

IPADR=<*ip1*>,<*ip2*>,<*ip3*>,<*ip4*>
IPADR=192,168,0,100

ip1	first byte of IP address
ip2	second byte of IP address
ip3	third byte of IP address

ip4	fourth byte of IP address

Query:
IPADR?

Reply:
<ip1>,<ip2>,<ip3>,<ip4>

10.4. Subnet mask

Set reader subnet mask. The changes will be applied after Reset command.

SNMASK=<*sm1*>,<*sm2*>,<*sm3*>,<*sm4*>
SNMASK=255,255,255,0

sm1	first byte of subnet mask
sm2	second byte of subnet mask
sm3	third byte of subnet mask
sm4	fourth byte of subnet mask

Query:
SNMASK?

Reply:
<sm1>,<sm2>,<sm3>,<sm4>

10.5. Default gateway

Set reader default gateway. The changes will be applied after Reset command.

GWADR=<*gw1*>,<*gw2*>,<*gw3*>,<*gw4*>
GWADR=192,168,0,1

gw1	first byte of gateway IP address
gw2	second byte gateway of IP address
gw3	third byte of gateway IP address
gw4	fourth byte of gateway IP address

Query:
GWADR?

Reply:
<gw1>,<gw2>,<gw3>,<gw4>

10.6. PHP/Cloud server IP address

Set IP address of the requested PHP/Cloud server.

WSADR=<*ip1*>,<*ip2*>,<*ip3*>,<*ip4*>
WSADR=192,168,0,2

ip1	first byte of IP address
ip2	second byte of IP address
ip3	third byte of IP address
ip4	fourth byte of IP address

Query:
WSADR?

Reply:
<ip1>,<ip2>,<ip3>,<ip4>

10.7. PHP/Cloud server port

Set port of the requested PHP/Cloud server.

WSPRT=<*port*>
WSPRT=5000

port	TCP port for cummunication with a cloud

Query:
WSPRT?

Reply:
<port>

10.8. PHP/Cloud script URL

Set URL of the processing PHP script.

WSURL=<*url*>
WSURL=/script.php

url string of script URL with max. 64 charecters

Query:
WSURL?

Reply:
<url>

10.9. PHP/Cloud host name

Set Host name of the server providing PHP script.

HOST=<*string*>
WSURL=216.58.209.67
WSURL=www.eeas.cz

string string of the host with max. 64 charecters

Query:
HOST?

Reply:
<string>

10.10. Reader ID

Set reader ID that is used to identify reader in HTTP requests.

READID=<*id*>
READID=1A2B3C4D

id reader id in hex format (%08X)

Query:
READID?

Reply:
<id>

10.11. Enable HASH

Enable or disable HASH computation.

HASH=<enable *0*|*1*>
HASH=0

enable 0 = disable; 1 = disable

Query:
HASH?

Reply:
<enable>

10.12. Salt

Set salt string added to the computed hash.

SALT=<string>
SALT=salt

string string of salt, max. 20 characters, no space

Query:
SALT?

Reply:
<string>

10.13. Heart beat period

Set period of sending a heart beat request.

HBRATE=<period>
HBRATE=60

period	period in seconds

Query:
HBRATE?

Reply:
<period>

10.14. Enable card buffering

Enable card buffering. The depth of the buffer is 64 card IDs. Reset must be applied after this command.

BUFF=<enable>
BUFF=1

enable	0 – disable; 1 – enable

Query:
BUFF?

Reply:
<enable>

10.15. Password

Set new password.

PASS=<password>
PASS=amin

password	string, max. 20 characters, no space

10.16. Set DESFire key

Set the DESFire key used for a card authentication.

DESFIREKEY=<key>
DESFIREKEY=ABCDEF0123456789ABCDEF0123456789

key	hexadecimal number, exactly 32 characters, each 2 characters represent one byte

Reply:
OK

10.17. Reset

Reset device and apply new settings. It may take a few seconds

RESET

10.18. Hold relay

Set relay for a specified time.

RELAY=<time>
RELAY=10

time	time in 100 ms (10 == 1 second)

Reply:
OK

10.19. Set relay

Set relay permanently.

RELON

Reply:
OK

10.20. Reset relay

Release relay permanently.

RELOFF

Reply:
OK

10.21. Flash green led

Flash green led for a specified time.

LEDG=<time>
LEDG=10

time time in 100 ms (10 == 1 second)

Reply:
OK

10.22. Set green led

Turn on green led permanently.

LEDGON

Reply:
OK

10.23. Reset green led

Turn off green led permanently.

LEDGOFF

Reply:
OK

10.24. Flash red led

Flash red led for a specified time.

LEDR=<time>
LEDR=10

time time in 100 ms (10 == 1 second)

Reply:
OK

10.25. Set red led

Turn on red led permanently.

LEDRON

Reply:
OK

10.26. Reset red led

Turn off red led permanently.

LEDROFF

Reply:
OK

10.27. Beep buzzer

Beep buzzer for a specified time.

BUZZER=<time>
BUZZER=10

time time in 100 ms (10 == 1 second)

Reply:
OK

10.28. Set buzzer

Turn on buzzer permanently.

BUZZON

Reply:
OK

10.29. Reset buzzer

Turn off buzzer permanently.

BUZZOFF

Reply:
OK

10.30. Wait for reply

Wait for reply indication for a specified time. After a card is detected then orange LEDs starts to flash until the specified timeout is exceeded or the server reply is received.

RWAIT=<time>
RWAIT=50

time time in 100 ms (10 == 1 second)

Reply:
OK

Query:
RWAIT?

Reply:
<time>

10.31. Get FW version

Returns reader firmware build date.

Query:
VER?

Reply:
Ethernet RFID r1 build: <date MMM d yyyy>
Ethernet RFID r1 build: Nov 10 2017

10.32. Ready – introduced in version r2

Replies with READY, can be used for HW readiness detection.

Query:
READY?

Reply:
READY

10.33. Last RFID card detected – introduced in version r2

Sends last ID of card detected. After query the command, the command will reply with zero length string unless new card is detected.

Query:
CARD?

Reply:
CARD=<ASCII decimal string of RFID>
CARD=38658633589

ASCII decimal string of RFID	RFID converted to decimal number and printed as ASCII string

10.34. Flash blue led – introduced in version r2

Flash blue led for a specified time.

LEDB=<time>
LEDB=10

time time in 100 ms (10 == 1 second)

Reply:
OK

10.35. Set blue led – introduced in version r2

Turn on blue led permanently.

LEDBON

Reply:
OK

10.36. Reset blue led – introduced in version r2

Turn off blue led permanently.

LEDBOFF

Reply:
OK

10.37. Set RGB led – introduced in version r2

Set RGB led color and time for flash. If zero time is set, then led stays illuminated util turned off with zeros collor component.

LEDRGB=<red>,<green>,<blue><time>
LEDRGB=255,255,0,10

red red component from range 0 … 255
green green component from range 0 … 255

blue	blue component from range 0 … 255
time	time in 100 ms (10 == 1 second)

Reply:
OK

10.38. LED intensity – introduced in version r2

Sets LED intensity for all color componets.

LEDINTENSITY=<intensity>
LEDINTENSITY=80

intensity	LED intensity in range 0 ... 255

Reply:
OK

Query:
LEDINTENSITY?

Reply:
LEDINTENSITY=<*time*>

---

11. Example of a PHP script

The following example shows how to implement a PHP script reacting to the reader. The script checks the read card_id and depending on this paramters decides to flash red or green led.

```
<?php
echo "<EEAS>\n";
$card_id = $_GET["card"];
if( $card_id == "09002518D4E0" )
{
    echo "LEDG=1\nBUZZER=1\n";
}
else
{
    echo "LEDR=1\nBUZZER=1\n";
}
echo "</EEAS>";
?>
```

Document updates

Date	Update
7. 12. 2023	Initial revision

On behalf of

Embedded Electronics & Solutions, s.r.o.

we would like to thank you.

Manufacturer:

Embedded Electronics & Solutions, s.r.o.
Primátorská 296/38
180 00 Praha 8

www.eeas.cz
Phone: +420 731480348 / +420 737980953

Distributor in Slovakia:

T-Industry, s.r.o.
Hoštáky 910/49
907 01 Myjava
tind@tind.sk
www.tind.sk
Phone: +421 907565722