

Ethernet DO and DI module

r0 – relay and transistor outputs only

User manual - draft



Embedded Electronics
&
Solutions, s.r.o.

www.eeas.cz



1. Device parameters

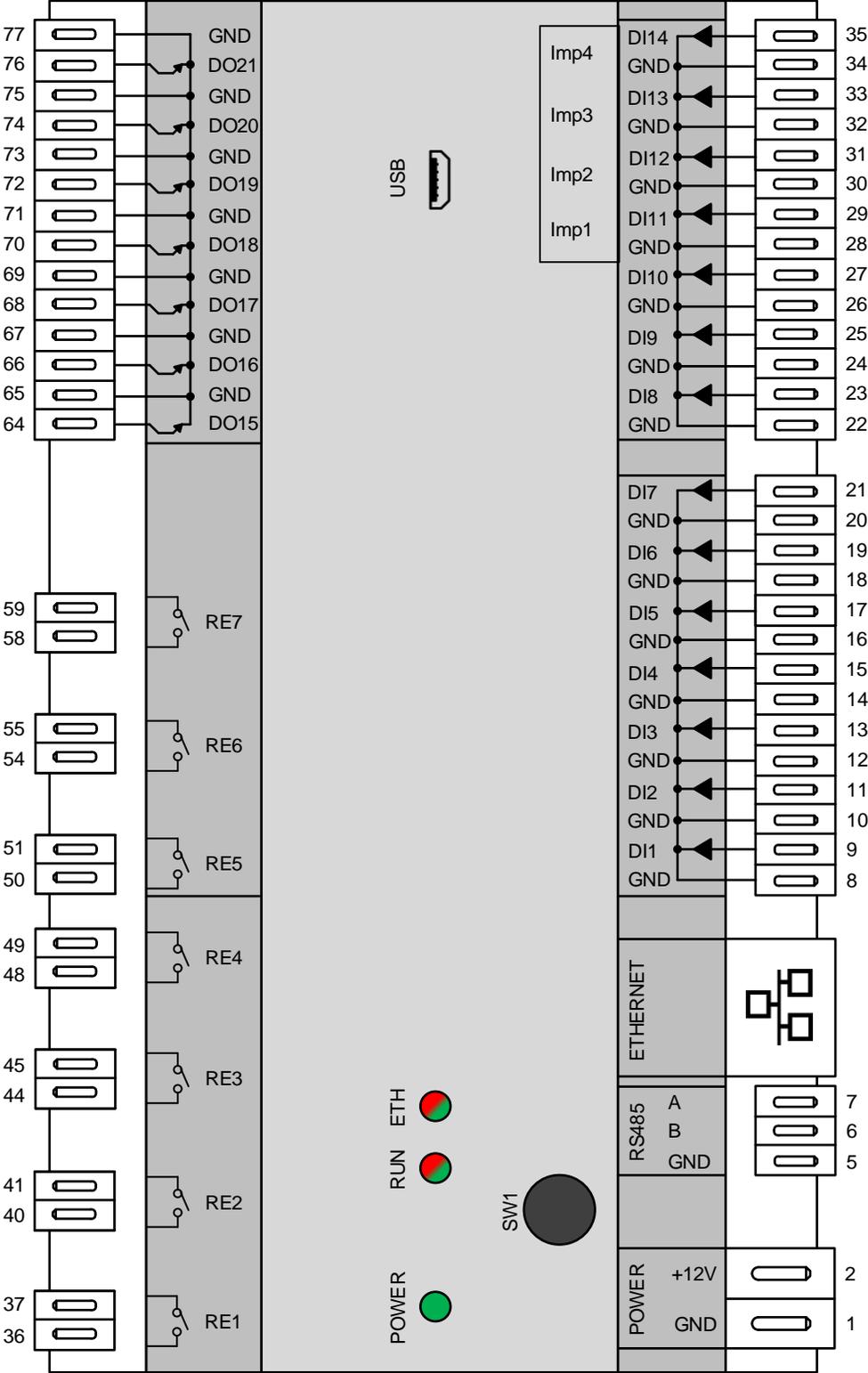
Power supply	typ 24VDC (another on request)
Power consumption	max. 2W
Switch parameters	250 VAC / 2A / 60W
USB interface (VCP)	115200/8/N/1
Impulse measurement	
Minimum impulse duration	1 ms
Impulse output type	S0 (or similar)
Digital input parameters	
Input sensing voltage	9 – 48 VDC
Input current	2,5 mA @ 12VDC input
Output parameters (relay)	
Switching parameters	5 A / 250 VAC
Output parameters (transistors)	
Type	Open-drain
Max. voltage	48 VDC
Max. current	3 A

2. Factory settings

IP address	192.168.0.100
Subnet mask	255.255.255.0
Gateway	192.168.0.1
Default TCP text port	5000
Default Modbus port	502
PHP port	80
Script URL	/script.php
Password	admin
Hash MD5	disable
Hash salt	salt
Heart beat period	60 seconds
Module ID	0x00000001
HTTP request port	80

To recall the factory settings, press and hold the button on the pannel until the leds start flashing, then release.

3. Front view



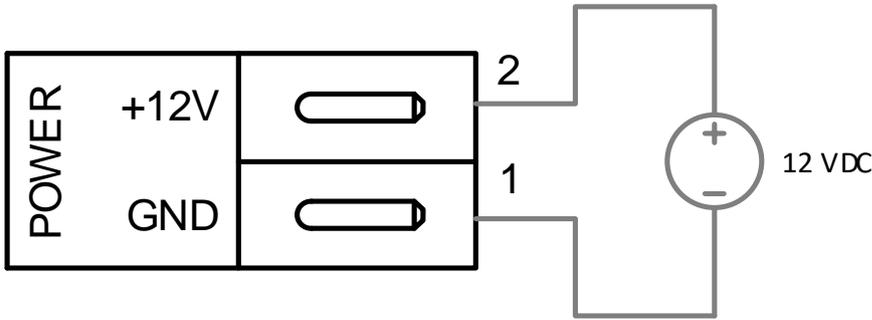
4. Description

The Ethernet DO and DI module is an electronic device providing remote digital outputs and inputs. Depending on a product version, it provides up to 21 transistor outputs or up to 14 relay outputs and 14 digital inputs. The device could be controlled over a raw TCP port or a Modbus TCP interface.

5. Electrical connection

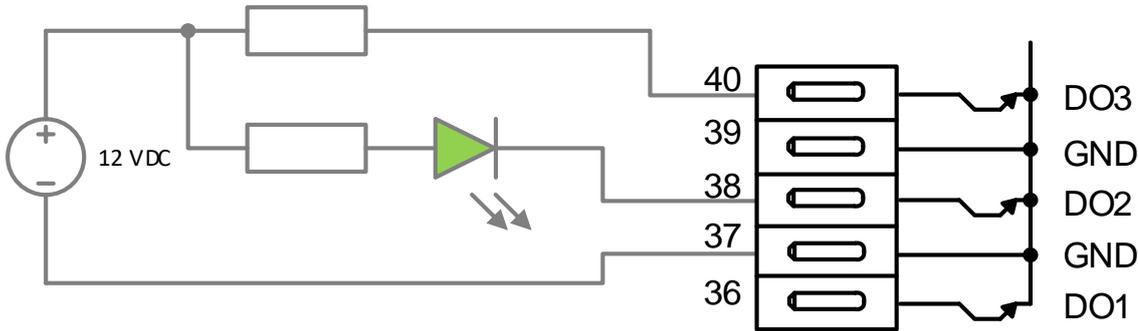
5.1. Power supply

The device is supplied by an external power supply (not included) with output voltage of 12 V DC. The connection circuit is stated on the picture below.



5.2. Transistor outputs

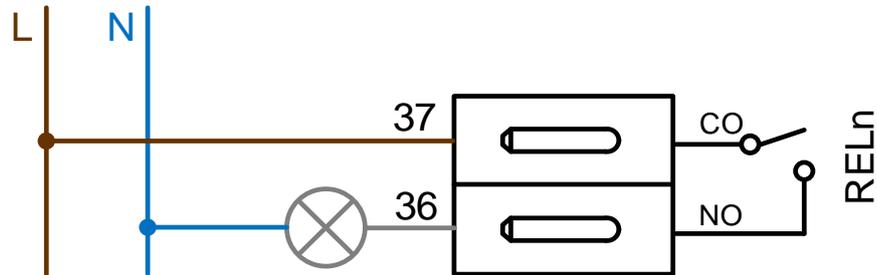
The transistors outputs are DO15 – DO21. An example of the connection is stated on the picture below. **The polarity of the external power source have to be respected!**



Max. switching voltage	48 VDC
Max. switching current	3 A

5.3. Relay outputs

The relay outputs are RE1 – RE7.



6. Communication interfaces

The module is equipped with three communication interfaces. A raw TCP/IP server listening at port 5000 (by default) for module configuration. The server also provides configuration of the HTTP request. An HTTP interface is intended for use in standard service. There is also TCP/IP server listening at port 502 (by default) for ModBUS/TCP communication. In addition RS485 Modbus/RTU interface is present. An USB interface is used as backup configuration interface and also for module firmware update.

6.1. Raw/configuration TCP socket

The TCP server serves for a setting the digital outputs and reading the digital inputs as well as the main configuration of the device. The user can connect to the server using a PC terminal such as netcat etc. A factory preset IP address and port are stated in chapter 2. After successful opening, the communication port user can write commands to the module. List of all supported commands is stated in chapter 7. Each command has to be terminated by line feed character ($\backslash n$, 0x0A). An example of a start configuration of the module is stated in a listing below. User commands are stated with **bold** characters and the module reply with plane characters. The line feed character is shown as $\backslash n$.

```
UNLOCK=admin\n
OK\n
PASS=password\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
```

Please note, the socket provides a 15 seconds timeout between two commands. If the timeout is reached, the socket connection is closed automatically.

6.2. HTTP interface

The HTTP interface serves for a normal operation of the module. The interfaces implements a heart beat request and receiving requests from host server.

6.2.1. Heart beat

If a heart beat request is enabled (see chapter 7.13) the module periodically 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&moduleid=00000001&hash=E82E
E47E3F15470F8A6D30C21B80F2C2 HTTP/1.1\n
Host: 216.58.201.99\n
Connection: close\n
\n
```

The request includes following parameters

<i>hb</i>	heart beat indicator,
<i>moduleid</i>	ID of the module sending the heart beat,
<i>hash</i>	hash code computed from all parameters.

If the hash code computation is enabled (see chapter 7.117.11 and 7.12) the hash is computed following way. The MD5 algorithm is used. The input string is assembled from the parameters *hb*, *moduleid* 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

100000001salt

The time period of sending the heart beat is adjustable by command desribed in chapter 7.13. The host server IP address and the script URL are adjustable using the commands described in chapter 7.

6.2.2. Asynchronous HTTP request from host server

If there is a need to address the module asynchronously from the host server the module 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

<i>CMD</i>	specify to the module to expect incoming commands,
<i>TOKEN</i>	token expected in module reply.

After the request is received the module sends another HTTP GET request directly to the host server with IP and URL set by the commands described in chapters 7.6 - 7.9. The format of the request is following

```
GET
/script.php?CO=1&TOKEN=aaaa&moduleid=00000001&has
```

```
h=9AF6601BBA054EE1B6C07E27B9793C8A HTTP/1.1\n
Host: 130.193.9.19\n
Connection: close\n
\n
```

with parameters

<i>CO</i>	specify to the server to prepare commands,
<i>TOKEN</i>	token from host server HTTP request,
<i>moduleid</i>	id of the module sending the request,
<i>hash</i>	hash code computed from all parameters.

The hash code (if is enabled) is computed the same way as was described in chapter 6.2.1 from the parameters *co*, *token* and *moduleid*. The host server IP address and the script URL are adjustable using the commands described in chapter 7.6 and 7.8.

In the body of the HTTP response from the server there could be included commands for the module. The format of the HTTP body is discussed in the chapter 6.3.

6.3. HTTP body commands formatting

The host server sends commands to the module in the body of the HTTP reply. The format of the reply is the same as in case described in chapter 6.2.2. After the host server received an HTTP request from the module the host have to reply to the module with a batch of commands. For example flashing the leds or switching on the relay. The commands have to be included in the body of the HTTP response surrounded by `<EEAS>` and `</EEAS>` tags. All commands have to be terminated by line feed character (`\n`, `0x0A`). The module accepts commands described in the chapter 7.

An example of the reply is shown in the listing bellow. It shows a basic response commanding to set relay 1 output. An example of the script implementation itself is stated in chapter 8.

```
<EEAS>\n
OUTSET=1,1\n
</EEAS>
```

6.4. Modbus TCP

The device is also equipped with a Modbus TCP communication interface. A specification of the interface could be found at <http://www.modbus.org/>. The memory map of the device outputs and inputs is stated in the table below.

Please note, the socket provides a 15 seconds timeout between two commands. If the timeout is reached, the socket connection is closed automatically.

Input registers	
0	FW version year
1	FW version month
2	FW version day
16-19	Impulse count on input DI11 (impulse counter 1)
20-23	Impulse count on input DI12 (impulse counter 2)
24-27	Impulse count on input DI13 (impulse counter 3)
28-31	Impulse count on input DI14 (impulse counter 4)

Holding registers	
0,1,2,3	IP address of the device (default is: 192,168,0,100)
4,5,6,7	IP subnet mask (default is: 255,255,255,0)
8,9,10,11	IP gateway (default is: 192,168,0,1)
12	TCP port of ModBUS communication (default is: 502)
13,14,15,16,17,18	MAC address (default is unique from factory)
19	TCP port of Text protocol communication (default is: 5000)
20	Modbus RTU RS485 address (applicable after set)
21	Modbus RTU RS485 baud rate (applicable after device reset) 0- 9600 1- 19200 2- 38400 3- 57600 4- 115200
22	Modbus RTU RS485 parity (applicable after device reset) 0- None 1- Odd 2- Even
23-26	Impulse setpoint count on input DI11 (impulse counter 1)
27-30	Impulse setpoint count on input DI12 (impulse counter 2)
31-34	Impulse setpoint count on input DI13 (impulse counter 3)
35-38	Impulse setpoint count on input DI14 (impulse counter 4)

Coils	
0-6	Relay outputs (Re1-Re7)
14-20	Transistor outputs (DO15-DO21)
24	If set module will load new IP settings and automatically make reset (ETH)
25	reset device

28	set imp. counter 1
29	set imp. counter 2
30	set imp. counter 3
31	set imp. counter 4

Input discrete	
0-13	digital inputs

6.5. USB interface

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

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

The USB interface can be used to a device firmware upgrade. In this case, a special PC utility is necessary to load the firmware.

7. List of commands

7.1. Unlock

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

UNLOCK=<password>

password password, default value = admin

7.2. IP address

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

IPADR=<ip1>,<ip2>,<ip3>,<ip4>

IPADR=192,168,0,100

<i>ip1</i>	first byte of IP address
<i>ip2</i>	second byte of IP address
<i>ip3</i>	third byte of IP address
<i>ip4</i>	fourth byte of IP address

Query:

IPADR?

Reply:

<ip1>,<ip2>,<ip3>,<ip4>

7.3. Subnet mask

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

SNMASK=<sm1>,<sm2>,<sm3>,<sm4>

SNMASK=255,255,255,0

<i>sm1</i>	first byte of subnet mask
<i>sm2</i>	second byte of subnet mask
<i>sm3</i>	third byte of subnet mask

sm4 fourth byte of subnet mask

Query:

SNMASK?

Reply:

<sm1>,<sm2>,<sm3>,<sm4>

7.4. Default gateway

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

GWADR=<gw1>,<gw2>,<gw3>,<gw4>

GWADR=192,168,0,1

<i>gw1</i>	first byte of gateway IP address
<i>gw2</i>	second byte gateway of IP address
<i>gw3</i>	third byte of gateway IP address
<i>gw4</i>	fourth byte of gateway IP address

Query:

GWADR?

Reply:

<gw1>,<gw2>,<gw3>,<gw4>

7.5. TCP ports

Set TCP ports of the raw TCP socket and the Modbus socket.

TCPPORT=<raw>,<modbus>

TCPPORT=5000,502

<i>raw</i>	TCP port of the raw TCP socket
<i>modbus</i>	TCP port of the Modbus interface

Query:

TCPPORT?

Reply:

<raw>,<modbus>

7.6. PHP/Cloud server IP address

Set IP address of the requested PHP/Cloud server.

WSADR=<ip1>,<ip2>,<ip3>,<ip4>

WSADR=192,168,0,2

<i>ip1</i>	first byte of IP address
<i>ip2</i>	second byte of IP address
<i>ip3</i>	third byte of IP address
<i>ip4</i>	fourth byte of IP address

Query:

WSADR?

Reply:

<ip1>,<ip2>,<ip3>,<ip4>

7.7. PHP/Cloud server port

Set port of the requested PHP/Cloud server.

WSPRT=<port>

WSPRT=5000

<i>port</i>	TCP port for communication with a cloud
-------------	---

Query:

WSPRT?

Reply:

<port>

7.8. 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>

7.9. 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>

7.10. Module ID

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

MODULEID=<id>

MODULEID=1A2B3C4D

id module id in hex format (%08X)

HBRATE=<period>

HBRATE=60

period

period in seconds

Query:

HBRATE?

Reply:

<period>

7.14. Password

Set new password.

PASS=<password>

PASS=amin

password

string, max. 20 characters, no space

7.15. Reset

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

RESET

7.16. Firmware version

Get firmware version.

Query:

VER?

Reply:

<version>

version

string describing a FW version

7.17. Is ready

Query used to detect the device is ready to operate. It can be used to reset the TCP timeout.

Query:

RDY?

Reply:

RDY

7.18. Help

Prints a description of the device commands.

HELP

Reply:

<device commands description>

7.19. Set output

Sets specified output to specified state.

OUTSET=<output>,<state>

OUTSET=5,1

output

position of the digital output

state

logical state to be set

Command OUTSET=5,1 sets output DO5 to log. 1 and all other outputs remain in the previous state.

Reply:

OK

Output was set

ERROR

Error occurred

7.20. Get output

Get state of specified output.

Query:

OUTGET=<output>?

Reply:

<state>

output

position of the digital output

state

logical state to be set

7.21. Set output word

Sets specified output word to the output register.

OUT=<output word>

OUT=5

output word

a word in decadic form assembled from particular binary output states where DO1 correspond to the LSB, see example bellow

Command OUT=5 sets outputs DO1 and DO3 to log. 1 and all other outputs to log. 0.

Reply:

OK

Outputs were set

ERROR

Error occurred

Query:

OUT?

Reply:

<output word>

7.22. Set output mask

Turn on specified outputs using an output mask.

OUTON=<output mask>

OUTON=5

output mask

a word in decadic form assembled from particular binary positions that should be set to log.1, where DO1 correspond to the LSB, see example bellow

Command OUT=5 sets outputs DO1 and DO3 to log. 1 and all other outputs remain in previous states.

Reply:

OK

Output was set

ERROR

Error occurred

7.23. Set output and automatically clear after defined time

Turn on specified output and after desired time automatically turn off output

OUTIMP=<output >,<time>

OUTON=1,10

output

position of the digital output

time

on time of the output in steps of 100 ms

Reply:

OK

Output was set

ERROR

Error occurred

7.24. Reset output mask

Turn off specified outputs using an output mask.

OUTOFF=<output mask>

OUTOFF=5

output mask

a word in decadic form assembled from particular binary positions that should be set to log.0, where DO1 correspond to the LSB, see example bellow

Command OUT=5 sets outputs DO1 and DO3 to log. 0 and all other outputs remain in previous states.

Reply:

OK

7.25. Get input word

Get the input word of the digital inputs states.

Query:

DINPUT?

Reply:

<input word>

input word

a word in decadic form assembled from particular binary output states where DI1 correspond to the LSB, see example bellow

Input word 9 represents the inputs DI1 and DI4 are set to log. 1 and all other inputs are sets to leg. 0.

7.26. Get impulse count

Get the impulse count on all impulse inputs.

Query:

IMPULSE?

Reply:

<im1>,<im2>,<im3>,<im4>

im1-im4

count of impulses on the specific inputs

7.27. Clear impulse count

Clear impulse count on the specified input

Query:

IMPCLR=<impinput>

IMPCLR=1

Reply:

OK

impulse counter cleared

impinput

specified the particular impulse counter to be cleared

8. Example of a PHP script

The following example shows how to implement a PHP script reacting to the module. The script checks the command tag and returns commands OUTSET and OUTIMP.

```
<?php
echo "<EEAS>\n";
$command = $_GET["CO"];
if( $command == "1" )
{
echo "OUTSET=1,1\n";
echo "OUTIMP=5,50\n";
}
else
{
}
echo "</EEAS>";
?>
```


***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

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