



relaydroid™ API Documentation

Please note:

This document is for system integrators and programmers. **relaydroid™** devices contain a built-in webserver with a web user interface suitable for most users. This document shows how you can substitute or extend the built-in functionality.

relaydroid™ devices can be controlled externally from a custom program via HTTP or TCP commands.

1. HTTP GET API

1.a) Get the relay states without switching

API v1 command:

```
http://relaydroid_address/api.cgi?p=admin_password
```

API v2 command:

```
http://relaydroid_address/api2.cgi?p=admin_password
```

answer:

relay states (see 1.d section)

relaydroid_address: the IP or NetBios name of the device

admin_password: the password of the admin user

1.b) Switch a relay ON or OFF continuously without a time limit and get the relay states

The state is stored in a non-volatile memory and is remembered in case of a system reboot.

API v1 command:

```
http:// relaydroid_address/api.cgi?p=admin_password&sw=oc_num&v=com
```

API v2 command:

```
http:// relaydroid_address/api2.cgi?p=admin_password&sw=oc_num&v=com
```

answer:

relay states (see 1.d section)



relaydroid_address: the IP or NetBios name of the device

admin_password: the password of the admin user

oc_num: 1, 2, .. n (number of the OC output to switch)

com: 0: switch OFF, 1: switch ON, 2: switch over (ON->OFF; OFF->ON)

1.c) Switch a relay ON with a time limit and get the relay states

The state is NOT stored in a non-volatile memory and in case of a system reboot the output will be OFF.

API v1 command:

`http:// relaydroid_address/api.cgi?p=admin_password&t=minutes&sw=oc_num&v=1`

OR

`http:// relaydroid_address/api.cgi?p=admin_password&t0=seconds&sw=oc_num&v=1`

OR (since v1.06)

`http:// relaydroid_address/api.cgi?p=admin_password&t1=milliseconds&sw=oc_num&v=1`

API v2 command:

`http:// relaydroid_address/api2.cgi?p=admin_password&t=minutes&sw=oc_num&v=1`

OR

`http:// relaydroid_address/api2.cgi?p=admin_password&t0=seconds&sw=oc_num&v=1`

OR (since v1.06)

`http:// relaydroid_address/api2.cgi?p=admin_password&t1=milliseconds&sw=oc_num&v=1`

answer:

relay states (see 1.d section)

relaydroid_address: the IP or NetBios name of the device

admin_password: the password of the admin user

t or *t0* or *t1*^(since v1.06): time limit in *minutes (t)* or *seconds (t0)* or *milliseconds (t1)* (after the given time the selected OC output will switch OFF automatically)

oc_num: 1, 2, .. n (number of the OC output to switch)

1.d) Relay state format of the answers when using api.cgi and api2.cgi

API v1 answer:

Sequence of '0'-s and '1'-s, showing which open collector output or digital input is ON (1) or OFF(0).

3 OC out e.g.:

100 (OC1: ON, OC2-3: OFF)

3 OC out + 3 dig. input e.g.:

100011 (OC1: ON, OC2-3: OFF, IN1: OFF, IN2-IN3: ON)



API v2 answer:

```
RELAY_MAX\r\n
R1_name$R2_name$...$Rmax_name\r\n
R1_default_time$R2_defaul_ttime$...$Rmax_defaul_ttime\r\n
R1_state$R2_state$...$Rmax_state\r\n
INPUT_MAX\r\n
I1_state$I2_state$...$Imax_state
```

- RELAY_MAX: the number of OC outputs
- Rx_name: the name of the OC output
- Rx_default_time: the default ON time set on the web GUI
- Rx_state:
 - pl. ON,0: ON without a time limit
 - pl. ON,-: ON with a time limit of less than 1 seconds
 - pl. ON,100: ON with a time limit of 100 seconds (it will be OFF after 100 seconds)
 - pl. OFF: off
- INPUT_MAX: the number of digital inputs
- Ix_state: ON (closed) or OFF (open)

e.g.:

```
3
R01$R02$R03
1$1$1
ON,10$ON,0$OFF
0
```

- the number of OC outputs is 3
- the name of the outputs is R01, R02 and R03
- the default ON time is 1, 1 and 1 sec
- the OC1 output is ON for another 10 seconds (ON,10),
- the OC2 output is ON without a time limit (ON,0),
- the OC3 output is OFF
- there are 0 digital input ports
- empty line because there are no digital inputs

2. TCP/IP API

Usage: open a TCP port and send the message below.
 Port number to open at the IP address of the device: 80
 Message to send (plain text):
 r[1-n] [0-999999999] [admin_password] \n
 OR (since v1.06)
 r[1-n] [0-999999999]- [admin_password] \n
 OR (since v1.06)



`r[1-n] - [admin_password] \n`

e.g.: `r1 3000 passw\n` - switch on OC1 for 3 seconds

e.g. ^(since v1.06): `r1 1000- passw\n` - switch on OC1 for 1 seconds and ask for current state of outputs

e.g. ^(since v1.06): `r1 - passw\n` - ask for current state of outputs

answer:

OK

OR ^(since v1.06)

[sequence of '0' and '1'] OK

The '0' and '1' characters show which OC output is ON (1) or OFF (0). This character sequence is in the answer only if the request contains the '-' (hyphen) character. Without the '-' the answer is simply 'OK'.

rn: *n* is the number of the OC output to switch

0-999999999: time limit in millisecs. After the given time the selected OC output will switch OFF automatically. The '0', '1' and '2' values have a special meaning:

- '0' means: switch OFF immediately.
- '1' means: switch ON without a time limit.
- '2' means: switch to the opposite state (ON->OFF, OFF->ON) without a time limit.
- Other values (bigger than 2) are rounded to 100 millisec intervals and they mean a time limit.

Since v1.06 version the request can contain the '-' (hyphen) character after the number or as a standalone character. Using the '-' character the answer will contain the state of the OC outputs. Using it with no number you can ask for the OC states without modifying any of them.

admin_password: the password of the admin user.

`\n`: the new-line character (ASCII 0x0A)