



INSTRUCTION MANUAL

ControlPlex[®] CPC12 bus controller –
Rest client endpoints



TABLE OF CONTENTS

1. Introduction	3
2. CPC12 Information (api/)	4
3. Network Configuration (api/network/)	11
4. ELBus Port State (api/#port/)	13
5. GET all Channel Information at Once	15
6. ELBus Channel State (api/#port/#channel/state)	16
7. ELBus Channel Info (api/#port/#channel/info)	20
8. ELBus Channel Measurements (api/#port/#channel/meas)	21
9. ELBus Channel Parameters (api/#port/#channel/param)	24

1. INTRODUCTION

CPC12 webserver is based on a REST-API (Representational State Transfer Application Programming Interface). HTTP-Methods GET and POST are supported. All information about the whole system is provided in JSON-strings. Therefore, the system offers endpoints in which related information is bundled. To control the system, users can post JSON-strings to the same endpoints that are used to read data.

All users have read access to each endpoint. To get write access, the user has to authenticate himself. The HTTP basic authentication scheme is used. The CPC12 supports two different user types namely admins and normal users. Admins are allowed to execute POST-commands on each endpoint. Normal users have limited POST access to the system. The required user privilege to access a specific endpoint is documented below.

The endpoints have the following format:

```
http://host/api/#elbus_port/#channel/
```

where `elbus_port` is a number between 1-P and `channel` is a number between 1-N. Currently P=1 and N=32. The `channel` number 0 returns a JSON-Object with all N `channel` information of a given endpoint (available since software version 1.0.25).

Following customer- and GUI relevant endpoints are defined (click to see its JSON implementation):

- <http://host/api/> Read/Write actual CPC12 related content
- <http://host/api/network/> Read/write network information
- <http://host/api/#port/> Read ELBus port state
- <http://host/api/#port/#channel/state/> Read/write channel state
- <http://host/api/#port/#channel/info/> Read channel infos
- <http://host/api/#port/#channel/meas/> Read channel measurements

- <http://host/api/#port/#channel/param/> Read/write channel parameters

PUT Methods are not supported. POST Methods are used to update information in the related endpoint. GET Methods are used to read information in the related endpoint. The value range of each attribute is listed after each example object.

Following endpoints are customer- und GUI relevant:

2. CPC12 INFORMATION (API/)

GET

```
{
  "name": "CPC12PN-T1",
  "description": "ControlPlex Controller Platform 12 Profinet, DIN-rail, 1 ELBus",
  "info": {
    "productid": "CPC12PN-T1-001",
    "productinfo": "Controller for REX12D and REX22D Devices",
    "intern_fert": 123
    "auftr_spl": 456
    "prod_an1": 789
    "sn": 123
    "version": {
      "hw": "1",
      "sw": "V1.1.1"
    }
  },
  "error": "none",
  "state": "standalone",
  "config": {
    "freeze": true,
    "usb_en": true,
    "power_save": true
  },
  "update_delay": 45,
  "fieldbus_is_writeable": true
}
```

JSON	Fieldbus equivalent						
	key	possible content	Hint	Header	Byte	Bit	Hint
name	"inactive" "CPC12PN-T1"	-	-	-	-	-	-
description	"" "ControlPlex Controller Platform 12 Profinet, DIN- rail, 1 ELBus"	-	-	-	-	-	-
productid	"" "CPC12PN-T1-001"	-	Device Information	0-1	-	-	Fieldbus --> Integer; Json --> String

JSON			Fieldbus equivalent			
key	possible content	Hint	Header	Byte	Bit	Hint
productinfo	"" "Controller for REX12D and REX22D Devices"	-	-	-	-	-
intern_fert	0-4294967295	-	Device Information	4-7	-	-
auftr_spl	0-65535	-	Device Information	8-9	-	-
prod_anl	0-65535	-	Device Information	10-11	-	-
sn	0-4294967295	-	Device Information	12-15	-	-
hw	"0"- "255"	-	Device Information	2-3	-	Fieldbus --> Integer; Json --> String
sw	"V0.0.0"- "V15.15.15"	V=Release Version, R=Release Candidate, T=Testversion	Device Information	16-18	-	Fieldbus --> Integer; Json --> String
error	"" "none" "error_critical" "error_uncritical"	-	Operating modes of the CPC12 bus controller	- - -	- - -	Start-up, Stand-alone, Slave System error Configuration error
state	"" "standalone" "error" "managed"	-	Operating modes of the CPC12 bus controller	- - -	- - -	Start-up, Stand-alone System error, Configuration error Slave
freeze	true false	-	Configuration data CPC12 controller	0	1	-

JSON			Fieldbus equivalent			
key	possible content	Hint	Header	Byte	Bit	Hint
usb_en	true false	-	Configuration data CPC12 controller	0	0	-
power_save	true false	-	Configuration data CPC12 controller	0	2	-
update_delay	35-60	maximal time needed for firmware update installation in seconds	-	-	-	-
fieldbus_is_writable	true false	-	-	-	-	-

POST

This command activates/deactivates energy save mode

 Administrator rights needed

```
{
  "energy_save": 1
}
```

JSON			Fieldbus equivalent			
key	possible content	Hint	Header	Byte	Bit	Hint
energy_save	0 1	-	Konfigurationsdaten CPC12 Controller	0	2	-

 This post will be ignored in case CPC is managed by PLC and usb/ws_enabled is cleared

Example:

```
curl -X POST -u admin:admin -H "Content-Type: application/json" --data "{\"energy_save\":0}" 192.168.1.1/api
```

This command resets all tripped channels

 Administrator rights needed

```
{  
  "reset_all":1  
}
```

JSON		Fieldbus equivalent				
key	possible content	Hint	Header	Byte	Bit	Hint
reset_all	1	-	-	-	-	-

 This post will be ignored in case CPC is managed by PLC and usb/ws_enabled is cleared

Example:

```
curl -X POST -u admin:admin -H "Content-Type: application/json" --data "{\"reset_all\":1}" 192.168.1.1/api
```

This command switches off all channels except for plc_locked channels

 Administrator rights needed

```
{  
  "turn_off_all":1  
}
```

JSON		Fieldbus equivalent				
key	possible content	Hint	Header	Byte	Bit	Hint
turn_off_all	1	-	-	-	-	-

⚠ This post will be ignored in case CPC is managed by PLC and usb/ws_enabled is cleared

Example:

```
curl -X POST -u admin:admin -H "Content-Type: application/json" --data "{\"turn_off_all\":1}" 192.168.1.1/api
```

This command sets following properties to default:

plc_lock for all channels

socket parameters

cpc configuration

 Administrator rights needed

```
{  
  "restore_factory":1  
}
```

JSON		Fieldbus equivalent				
key	possible content	Hint	Header	Byte	Bit	Hint
restore_factory	1	-	Action commands CPC12 Controller	0	-	value: 118

⚠ This post will be ignored in case CPC is managed by PLC and usb/ws_enabled is cleared

Example:

```
curl -X POST -u admin:admin -H "Content-Type: application/json" --data "{\"restore_factory\":1}" 192.168.1.1/api
```

This command resets following properties to default:

plc_lock for all channels

socket parameters

socket configurations

cpc configuration

complete non-volatile-memory including:

min, max and average values

3rd ETH IP settings

user names and passwords

 Administrator rights needed

```
{  
  "back_to_box":1  
}
```

JSON		Fieldbus equivalent				
key	possible content	Hint	Header	Byte	Bit	Hint
back_to_box	1	-	Action commands CPC12 Controller	0	-	value: 131

 This post will be ignored in case CPC is managed by PLC and usb/ws_enabled is cleared

Example:

```
curl -X POST -u admin:admin -H "Content-Type: application/json" --data "{\"back_to_box\":1}"  
192.168.1.1/api
```

This command adopts the device type for all connected devices

 Administrator rights needed

```
{  
  "adopt_device_types":1  
}
```

JSON		Fieldbus equivalent				
key	possible content	Hint	Header	Byte	Bit	Hint
adopt_device_types	1	-	-	-	-	-

 This post will be ignored in case CPC is managed by PLC and usb/ws_enabled is cleared

Example:

```
curl -X POST -u admin:admin -H "Content-Type: application/json" --data "{\"adopt_device_types\":1}"  
192.168.1.1/api
```

3. NETWORK CONFIGURATION (API/NETWORK/)

GET

```
{
  "ethernet":{
    "address":"x.x.x.x",
    "netmask": "x.x.x.x",
    "gateway" : "x.x.x.x",
    "dhcp": true
  },
  "fieldbus":{
    "address":"x.x.x.x",
    "netmask": "x.x.x.x",
    "gateway" : "x.x.x.x",
    "dhcp": false
  },
  "hostname":"CPC12"
}
```

JSON		Fieldbus equivalent				
key	possible content	Hint	Header	Byte	Bit	Hint
address	"x.x.x.x"	-	-	-	-	-
netmask	"x.x.x.x"	-	-	-	-	-
gateway	"x.x.x.x"	-	-	-	-	-
dhcp	true	-	-	-	-	-
	false	-	-	-	-	-

POST

This commands sets IP-settings for 3rd Etehrnet-Port

 Administrator rights needed

```
{
  "ethernet":{
    "address":"x.x.x.x",
    "netmask": "x.x.x.x",
    "gateway" : "x.x.x.x",
    "dhcp": true
  }
}
```

JSON			Fieldbus equivalent			
key	possible content	Hint	Header	Byte	Bit	Hint
address	"x.x.x.x"	-	-	-	-	-
netmask	"x.x.x.x"	-	-	-	-	-
gateway	"x.x.x.x"	-	-	-	-	-
dhcp	true false	-	-	-	-	-

Example:

```
curl -X POST -u admin:admin -H "Content-Type: application/json" --data "{\"ethernet\":{\"address\":\"192.168.1.2\",\"ne
```

This commands sets IP-settings for the Fieldbus-Ports

- ⓘ Administrator rights needed
- ⓘ Only available for the Modbus-TCP (MB) device

```
{
  "fieldbus":{
    "address":"x.x.x.x",
    "netmask": "x.x.x.x",
    "gateway" : "x.x.x.x",
    "dhcp": true
  }
}
```

4. ELBUS PORT STATE (API/#PORT/)

GET

```
{
  "supply": 24.01,
  "total_current": 38.06,
  "channels": [{
    "channel": 1,
    "number_of_channels": 2,
    "inom": 3,
    "state": "OK",
    "event": "None",
    "load_current": 1.2,
    "load_voltage": 24.2
  },
  {
    "channel": 2,
    "number_of_channels": 2,
    "inom": 3,
    "state": "OK",
    "event": "energy_safe",
    "load_current": 1.2,
    "load_voltage": 24.2
  },
  {
    "channel": 3,
    "number_of_channels": 1,
    "inom": 5,
    "state": "OFF",
    "event": "button_pressed",
    "load_current": 1.2,
    "load_voltage": 24.2
  }
  ]
}
```

JSON		Fieldbus equivalent				
key	possible content	Hint	Header	Byte	Bit	Hint
supply	0-655.35	-	-	-	-	-
total_current	0-655.35	-	Submodule total current	0-1	-	Json = Fieldbus/100
channel	1-32	-	-	-	-	-
number_of_channels	1-2	-	-	-	-	-

JSON		Fieldbus equivalent				
key	possible content	Hint	Header	Byte	Bit	Hint
inom	0-255	-	Device parameters for one channel	0	-	-
state	"No device inserted" "wrong_productid" "OK" "OFF" "Shortcircuit" "Overload" "Undervoltage" "Warning Threshold"	-	Submodule circuit protectors	0 or 5	- - 0 0 1 2 3 6	Byte 0 for odd channel-numbers, Byte 5 for even channel-numbers
event	"None" "device_error" "button_pressed" "waiting_for_param" "energy_safe"	-	Event message for one channel	1	- 7 3 0 -	Byte 0 for odd channel-numbers, Byte 5 for even channel-numbers
load_current	0-655.35	-	Submodule circuit protectors	1-2 or 6-7	-	Byte 1-2 for odd channel-numbers, Byte 6-7 for even channel-numbers
load_voltage	0-655.35	-	Submodule circuit protectors	3-4 or 8-9	-	Byte 3-4 for odd channel-numbers, Byte 8-9 for even channel-numbers

5. GET ALL CHANNEL INFORMATION AT ONCE

The `channel` number 0 returns a JSON-Object with all 32 `channel` information of the following endpoints (available since software version 1.0.25):

- `api/#port/#channel/state`
- `api/#port/#channel/info`
- `api/#port/#channel/meas`
- `api/#port/#channel/param`

GET with `#channel == 0`

```
{
  "channel_1": {
    "state": "OK",
    "reason": "None",
    "event": "None",
    "error": "None",
    "trip_cnt": 0,
    "last_trip": "None"
  },
  "channel_2": {
    "state": "OK",
    "reason": "None",
    "event": "None",
    "error": "None",
    "trip_cnt": 0,
    "last_trip": "None"
  },
  ...
}
```

6. ELBUS CHANNEL STATE (API/#PORT/#CHANNEL/STATE)

GET

```
{
  "state": "OK",
  "reason": "shortcircuit",
  "error": "watchdog_reset",
  "event": "None",
  "trip_cnt": 200,
  "last_trip": "overload"
}
```

JSON		Fieldbus equivalent				
key	possible content	Hint	Header	Byte	Bit	Hint
state	"inactive"	-	Submodule circuit protectors	0 or 5	-	Byte 0 for odd channel-numbers, Byte 5 for even channelnumbers
	"OK"				0	
	"OFF"				0	
	"trip"				1-2	
	"threshold"				6	
reason	"wrong_productid"				-	
reason	"None"	channel is on	Submodule circuit protectors	0 or 5	-	Byte 0 for odd channel-numbers, Byte 5 for even channelnumbers
reason	"shortcircuit" "overload" "undervoltage"	-	Submodule circuit protectors	0 or 5	1	Byte 0 for odd channel-numbers, Byte 5 for even channelnumbers
					2	
					3	
reason	"button_pressed"	off by button	Submodule circuit protectors	0 or 5	7	Byte 0 for odd channel-numbers, Byte 5 for even channelnumbers
reason	"elbus"	off by bus	-	-	-	-

JSON		Fieldbus equivalent				
key	possible content	Hint	Header	Byte	Bit	Hint
error	"None"				-	
	"no_sys_params"				0	
	"chksum_err_param"	-	Diagnostic data for one channel	0-1	1	
	"chksum_err_program"				2	-
	"intern_storage_err"				3	
	"controller_err"				4	
"watchdog_reset"	5					
event	"None" "waiting_for_param"				-	Event message for one channel
event	"history_available"	-	-	-	-	not available on CPC12
event	"energy_safe"	-	-	-	-	-
event	"device_error"	-	Event message for one channel	0	7	-
trip_cnt	0-65535	-	Diagnostic data for one channel	2-3	-	-
last_trip	"None"				0	
	"shortcircuit"				1	
	"overload"	-	Diagnostic data for one channel	4	2	-
	"overtemp"				3	
"intern_err"	4					

POST

This command turns corresponding channel on/off

 No administrator rights needed

```
{
  "state":1
}
```

JSON		Fieldbus equivalent				
key	possible content	Hint	Header	Byte	Bit	Hint
state	0 1	-	Submodule circuit protec- tors	0 or 1	0	Byte 0 for odd channelnumbers, Byte 1 for even channelnumbers

⚠ This post will be ignored in case CPC is managed by PLC and usb/ws_enabled is cleared

Example:

```
curl -X POST -u admin:admin -H "Content-Type: application/json" --data "{\"state\":1}"
192.168.1.1/api/1/1/state
```

This command reset corresponding channel

 No administrator rights needed

```
{
  "reset":1
}
```

JSON		Fieldbus equivalent				
key	possible content	Hint	Header	Byte	Bit	Hint
reset	0 1	-	Submodule circuit protec- tors	0 or 1	1	Byte 0 for odd channelnumbers, Byte 1 for even channelnumbers

⚠ This post will be ignored in case CPC is managed by PLC and usb/ws_enabled is cleared

Example:

```
curl -X POST -u admin:admin -H "Content-Type: application/json" --data "{\"reset\":1}"
192.168.1.1/api/1/1/state
```

This command resets the trip counter of the corresponding device

 No administrator rights needed

```
{  
  "reset_trip_cnt":1  
}
```

JSON		Fieldbus equivalent				
key	possible content	Hint	Header	Byte	Bit	Hint
re- set_trip_cnt	1	-	Action com- mands for one channel	0	-	value: 116

 This post will be ignored in case CPC is managed by PLC and usb/ws_enabled is cleared

Example:

```
curl -X POST -u admin:admin -H "Content-Type: application/json" --data "{\"reset_trip_cnt\":1}"  
192.168.1.1/api/1/1/sta
```

7. ELBUS CHANNEL INFO (API/#PORT/#CHANNEL/INFO)

GET

```
{
  "name": "REX12D-TA1-100",
  "productid": 36873,
  "productinfo": "Manufacturer info string",
  "serial_number": 712401002,
  "version": {
    "hw": "0.1",
    "sw": "1.3.0"
  },
  "channels": 1
}
```

JSON		Fieldbus equivalent				
key	possible content	Hint	Header	Byte	Bit	Hint
name	"Unknown Device" "REX12D-TA1- 100" "..." "REX12D-TE2-101 (Class2)"	-	Configuration data for one channel	0-1	-	Fieldbus --> Integer; Json --> String
productid	0-65535	-	Configuration data for one channel	0-1	-	-
productinfo	"" "Manufacturar info string"	-	-	-	-	-
serial_number	0-4294967295	-	Device information for one channel	12-15	-	-
hw	"0.0"-"255.255"	-	Device information for one channel	2-3	-	Fieldbus --> Integer; Json --> String
sw	"0.0.0"-"15.15.15"	-	Device information for one channel	16-18	-	Fieldbus --> Integer; Json --> String
channels	0-4	-	-	-	-	-

8. ELBUS CHANNEL MEASUREMENTS (API/#PORT/#CHANNEL/MEAS)

GET

```
{
  "load_current":{
    "actual": 2.32,
    "min": 1.11,
    "max": 4.34,
    "average": 1.23
  },
  "load_voltage": {
    "actual": 24.3,
    "min": 15.3,
    "max": 28.2,
    "average":25.1
  }
}
```

JSON			Fieldbus equivalent			
key	possible content	Hint	Header	Byte	Bit	Hint
actual	0-655.35	voltage	Submodule circuit protectors	3-4 or 8-9	-	Byte 3-4 for odd channel-numbers, Byte 8-9 for even channelnumbers
min	0-655.35	voltage	Diagnostic data for one channel	5-6	-	-
max	0-655.35	voltage	Diagnostic data for one channel	7-8	-	-
average	0-655.35	voltage	Diagnostic data for one channel	9-10	-	-
actual	0-655.35	current	Submodule circuit protectors	1-2 or 6-7	-	Byte 1-2 for odd channel-numbers, Byte 6-7 for even channelnumbers

JSON		Fieldbus equivalent				
key	possible content	Hint	Header	Byte	Bit	Hint
min	0-655.35	current	Diagnostic data for one channel	11-12	-	-
max	0-655.35	current	Diagnostic data for one channel	13-14	-	-
average	0-655.35	current	Diagnostic data for one channel	15-16	-	-

POST

This post resets min-value

 No administrator rights needed

```
{
  "reset_min":1
}
```

JSON		Fieldbus equivalent				
key	possible content	Hint	Header	Byte	Bit	Hint
reset_min	1	-	Action commands for one channel	0	-	value: 192

⚠ This post will be ignored in case CPC is managed by PLC and usb/ws_enabled is cleared

Example:

```
curl -X POST -u admin:admin -H "Content-Type: application/json" --data "{\"reset_min\":1}"
192.168.1.1/api/1/1/meas
```

This command resets the max-value

 No administrator rights needed

```
{
  "reset_max":1,
}
```

JSON		Fieldbus equivalent				
key	possible content	Hint	Header	Byte	Bit	Hint
reset_max	1	-	Action commands for one channel	0	-	value: 196

⚠ This post will be ignored in case CPC is managed by PLC and usb/ws_enabled is cleared

Example:

```
curl -X POST -u admin:admin -H "Content-Type: application/json" --data "{\"reset_max\":1}" 192.168.1.1/api/1/1/meas
```

This command resets average-value

 No administrator rights needed

```
{
  "reset_average":1,
}
```

JSON		Fieldbus equivalent				
key	possible content	Hint	Header	Byte	Bit	Hint
re-set_average	1	-	Action commands for one channel	0	-	value: 220

⚠ This post will be ignored in case CPC is managed by PLC and usb/ws_enabled is cleared

Example:

```
curl -X POST -u admin:admin -H "Content-Type: application/json" --data "{\"reset_average\":1}" 192.168.1.1/api/1/1/meas
```

9. ELBUS CHANNEL PARAMETERS (API/#PORT/#CHANNEL/PARAM)

GET/POST

This command sets the parameters of the corresponding channel:

 Administrator rights needed

```
{
  "productid":36873,
  "inom": 1,
  "curlim": 80,
  "lock": false
}
```

JSON		Fieldbus equivalent				
key	possible content	Hint	Header	Byte	Bit	Hint
product_id	0-65535	-	Configuration data for one channel	0-1	-	-
inom	1-10	-	Device parameters for one channel	0	-	-
curlim	50-100	-	Device parameters for one channel	6	-	-
lock	true false	-	Device parameters for one channel	1-4	-	Fieldbus -> 1 bit per channel; Json --> Boolean

Select a expected productid from `/api/productids/` for this channel. This will also set the productid for the second channel: Fx.1 and Fx.2

```
curl -X POST -u admin:admin -H "Content-Type: application/json"
--data "{\"productid\":36873,\"inom\":1,\"curlim\":80,\"lock\":false}" 192.168.1.1/api/1/1/param
```

 This post will be ignored in case CPC is managed by PLC and usb/ws_enabled is cleared

Example:

This command sets default parameters to the corresponding channel

 Administrator rights needed

```
{  
  "set_default":1  
}
```

JSON		Fieldbus equivalent				
key	possible content	Hint	Header	Byte	Bit	Hint
set_default	1	-	Action commands for one channel	0	-	value: 118

 This post will be ignored in case CPC is managed by PLC and usb/ws_enabled is cleared

Example:

```
curl -X POST -u admin:admin -H "Content-Type: application/json" --data "{\"set_default\":1}"  
192.168.1.1/api/1/1/param
```

E-T-A Elektrotechnische Apparate GmbH

Industriestraße 2-8

90518 Altdorf

Tel. +49 9187 10-0

Fax +49 9187 10-397

E-Mail: info@e-t-a.de

global.e-t-a.com