



INSTRUCTION MANUAL

ControlPlex[®] CPC20 bus controller – Rest client endpoints



TABLE OF CONTENTS

1. Introduction	3
2. CPC20 Information (api/)	4
3. Network Configuration (api/network/)	7
4. ELBus Port State (api/#port/)	8
5. ELBus Channel State (api/#port/#channel/state)	9
6. ELBus Channel Info (api/#port/#channel/info)	11
7. ELBus Channel Measurements (api/#port/#channel/meas)	12
8. ELBus Channel Parameters (api/#port/#channel/param)	14
9. ELBus Channel History (api/#port/#channel/histo)	16

1. INTRODUCTION

CPC20 webserver is based on a REST-API (**RE**presentational **State Transfer** **A**pplication **P**rogramming **I**nterface). 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 CPC20 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=2 and N=32.

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

- <http://host/api/> Read/Write actual cpc20 related content
- <http://host/api/network/> Read/write network information
- <http://host/api/#port/> Read/write port related information
- <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

- <http://host/api/#port/#channel/histo/> Read/write history memory

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. CPC20 INFORMATION (API/)

GET

```
{
  "name": "CPC20PN-T2",
  "description": "ControlPlex Controller Platform 20 Profinet, Din-rail, 2 ELBus",
  "info": {
    "productid": "CPC20PN-T2-001",
    "productinfo": "Busklemmen-Controller für 18plus-ELBus mit ESX60D-S200",
    "intern_fert": 4294967295,
    "auftr_spl": 65535,
    "prod_anl": 65535,
    "sn": 4294967295,
    "version": {
      "hw": "1.0.0",
      "sw": "1.0.4"
    }
  },
  "error": "none",
  "state": "standalone",
  "config": {
    "freeze": true,
    "usb_en": true,
    "power_save": false
  }
}
```

```
"name" : "CPC20PN-T2"
"description" : "ControlPlex Controller Platform 20 Profinet, Din-rail, 2 ELBus"
"productid" : "CPC20PN-T2-001"
"productinfo" : "Busklemmen-Controller für 18plus-ELBus mit ESX60D-S200"
"intern_fert" : 0 - 4294967295
"auftr_spl" : 0 - 65535
"prod_anl" : 0 - 65535
"sn" : 0 - 4294967295
"hw" : "0.0.0" - "15.15.15"
"sw" : "0.0.0" - "15.15.15"
"error" : "" ; "none" ; "error_critical" ; "error_uncritical"
"state" : "" ; "standalone" ; "error" ; "managed"
"freeze" : true ; false
If usb_en is set to false by the fieldbus master, any POST request will be rejected
"usb_en" : true ; false
"power_save" : true ; false
```

POST

This command activates/deactivates energy save mode

 No administrator rights needed

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

```
"energy_save" : 0 ; 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 "{\"energy_save\":0}" 192.168.1.1/api
```

This command resets all tripped channels

① No administrator rights needed

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

```
"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

① No administrator rights needed

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

```
"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:

- socket configurations
- board configurations
- application tag
- custom data
- svs config
- fs_version
- min and max values
- pn and eth configurations
- plc lock infos

 No administrator rights needed

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

```
"restore_factory" : 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
{"restore_factory":1} 192.168.1.1/api
```


3. NETWORK CONFIGURATION (API/NETWORK/)

GET

```
{
  "ethernet":{
    "addr":"x.x.x.x",
    "mask": "x.x.x.x",
    "gw" : "x.x.x.x",
    "dhcp": true
  },
  "profinet":{
    "addr":"x.x.x.x",
    "mask": "x.x.x.x",
    "gw" : "x.x.x.x",
    "dhcp": false
  },
  "hostname": "CPC20"
}
```

POST

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

 No administrator rights needed

```
{
  "ethernet":{
    "addr":"x.x.x.x",
    "mask": "x.x.x.x",
    "gw" : "x.x.x.x",
    "dhcp": true
  },
  "hostname": "CPC20"
}
```

Example:

```
curl -X POST -u admin:admin -H "Content-Type: application/json" --data
"{\"ethernet\":{\"address\": \"192.168.1.2\",
\"netmask\": \"255.255.255.0\", \"gateway\": \"0.0.0.0\",
\"dhcp\": false},CPC20}" 192.168.1.1/api/network
```


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

GET

```
{
  "supply": "24",
  "total_current": "43",
  "channels": [{
    "channel": 1,
    "inom": 3,
    "state": "OK",
    "event": "None",
    "load_current": 1.21,
    "load_voltage": 24.2,
    "temperature": 23.0
  },
  {
    "channel": 2,
    "Inom": 3,
    "state": "OK",
    "event": "history_available",
    "load current": 1.21,
    "load voltage": 24.2,
    "temperature": 23.0
  },
  {
    "channel": 3,
    "inom": 5,
    "state": "OFF",
    "event": "button_pressed",
    "load current": 1.21,
    "load voltage": 24.2,
    "temperature": 23.0
  }
  ]
}
```

"supply" : 0 - 655.35

"total current" : 0 - 655.35

"channel" : "1 - 32"

"inom" : "1 - 10"

"state" : "OK"; "OFF"; "Shortcircuit"; "Overload"; "Undervoltage"; "Overtemperature"; "Warning Threshold"; "Overvoltage"; "No device inserted"

"event" : "device_error"; "button_pressed"; "waiting_for_param"; "history_available"; "energy_safe"

"load current" : 0 - 655.35

"load voltage" : 0 - 655.35

"temperature" : 0 - 655.35

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

GET

```
{
  "state": "OK",
  "reason": "shortcircuit",
  "error": "watchdog_reset",
  "event": "history_available",
  "trip cnt": 200,
  "last trip": "overload"
}
```

```
"state" : "inactive"; "OK"; "OFF"; "trip"; "threshold"
"reason" : "none"; "shortcircuit"; "overload"; "undervoltage"; "overvoltage"; "button_pressed"; "overtemp"; "elbus"
"error" : "None"; "fail_safe"; "cur_sens"; "no_sys_params"; "chksum_err_param"; "chksum_err_program"; "intern_storage_err"; "controller_err"; "watchdog_reset"
"event" : "None"; "waiting_for_param"; "history_available"; "energy_safe"; "device_error"; "button_pressed"
"trip_cnt" : 0 - 65535
"last_trip" : "None"; "shortcircuit"; "overload"; "overtemp"; "intern_err"
```


POST

This command turns corresponding channel on/off

 No administrator rights needed

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

```
"state" : 0 ; 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
{"state":1} 192.168.1.1/api/1/1/state
```

This command reset corresponding channel

 No administrator rights needed

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

```
"reset" : 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":1} 192.168.1.1/api/1/1/state
```

This command resets the trip counter of the corresponding device

① No administrator rights needed

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

```
"reset_trip_cnt" : 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_trip_cnt":1} 192.168.1.1/api/1/1/state
```

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

GET

```
{
  "name": "ESX60D-SA2-100",
  "productid": "0x901E",
  "productinfo": "Two channel electronic circuit protector",
  "serial_number": 50462976,
  "version": {
    "hw": 0,
    "sw": "1.0.1"
  },
  "channels": 2
}
```

```
"name" : "inactive"; "ESX60D-SA2-100"
"productid" : "0x901E"
"productinfo" : "Two channel electronic circuit protector"
"serial_number" : 0 - 4294967295
"hw" : "0.0" - "255.255"
"sw" : "0.0.0" - "15.15.15"
"channels" : 2
```

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

GET

```
{
  "load_current": {
    "actual": 0,
    "min": 0,
    "max": 3.7,
    "average": 0
  },
  "load_voltage": {
    "actual": 0,
    "min": 0,
    "max": 31.8,
    "average": 0
  },
  "supply_voltage": 24.2,
  "temperature": 34.6
}
```

```
"actual" : 0 - 655.35
"min" : 0 - 655.35
"max" : 0 - 655.35
"average" : 0 - 655.35
```

POST

This post resets min-value

 No administrator rights needed

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

```
"reset_min" : 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_min\":1}" 192.168.1.1/api/1/1/meas
```

This command resets the max-value

ⓘ No administrator rights needed

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

```
"reset_max" : 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_max\":1}" 192.168.1.1/api/1/1/meas
```

This command resets average-value

ⓘ No administrator rights needed

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

```
"reset_average" : 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_average\":1}" 192.168.1.1/api/1/1/meas
```

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

GET/POST

This command sets the parameters of the corresponding channel:

 Administrator rights needed

```
{
  "inom": 1,
  "pon_behav": "Last State",
  "overloadshtd": 120,
  "overloadshtdtime": 3000,
  "shortshutdtime": 400,
  "pon_delay": 100,
  "curlim": 80,
  "hyst": 5,
  "lock": true
}
```

```
"inom" : 1 - 10
"pon_behav" : "Last State"; "Off"; "On"
"overloadshtd" : 105 - 135 (increment 1)
"overloadshtdtime" : 50 - 10000 (increment 50)
"shortsutdtime" : 400
"pon_delay" : 50 - 2500 (increment 10)
"curlim" : 50 - 100 (increment 1)
"hyst" : 5 - 20 (increment 1)
"lock" : true ; false
```

 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
{"\inom\": 1, \"pon_behav\": \"Last State\", \"overloadshtd\":120, \"overloadshtdtime\":3000,
 \"shortshutdtime\":400, \"pon_delay\":100, \"curlim\":80,
 \"hyst\":5, \"lock\":true}" 192.168.1.1/api/1/1/param
```

This command sets default parameters to the corresponding channel

 Administrator rights needed

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

```
"set_default": 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  
{"set_default":1} 192.168.1.1/api/1/1/param
```


9. ELBUS CHANNEL HISTORY (API/#PORT/#CHANNEL/HISTO)

GET

```
{  
  "voltage": [1.1, 23.3, ..., 34.3],  
  "current": [1.2, 34.2, ..., 12.3]  
}
```

"voltage" : 0 - 655.35 (Array of 400 values)
"current" : 0 - 655.35 (Array of 400 values)

```
{  
  "error": "no history"  
}
```

"error" : "no history"

POST

This command clears history data of the corresponding channel

 Administrator rights needed

```
{  
  "reset": "true"  
}
```

"reset" : "true"

⚠ 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\":\"true\"} 192.168.1.1/api/1/1/histo
```


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