

EZcontrol© XS1

Kommunikationsschnittstelle

Benutzerbefehlssatz (Auszug)

HTTP / JSON

Protokoll Ver. 11

EZcontrol® XS1 Dokumentation der Programmierschnittstelle v11 Ausgabe 2

Rose + Herleth GbR
Wendenweg 17
13595 Berlin
Tel. +49 (0)30 369 91554
Fax +49 (0)30 362 83064
www.ezcontrol.de

Inhaltsverzeichnis

Einleitung

Bei der Programmierschnittstelle des EZcontrol XS1 handelt es sich um eine Kombination von HTTP GET Requests (HTTP URLs) und Antworten in Textform, die das Datenformat *JSON (**J**ava**S**cript **O**bject **N**otation*) benutzen.

Das Datenformat JSON wurde gewählt, da nur dieses eine Kommunikation über Domaingrenzen hinweg ermöglicht.

Dies ist notwendig, damit eine JavaScript/AJAX Anwendung lokal auf dem jeweiligen Client liegen kann („lokale“ Domain) und trotzdem mit dem EZcontrol XS1 (entfernte Domain) kommunizieren kann.

Lange Ladezeiten der Anwendung über die eventuell schmalbandige Internetverbindung entfallen dadurch und auch die Datenmenge wird klein gehalten.

Des Weiteren lassen sich mehrere XS1 Geräte in verschiedenen Domainen steuern, ohne die Anwendung neu laden zu müssen.

Die Sicherheitseinstellungen der Browser lassen in der Regel eine andere Art der Kommunikation, wie zum Beispiel XML, über Domaingrenzen hinweg nicht zu.

Kompatibilität

Cross-Domain JSON wird unterstützt und wurde von uns getestet mit folgenden Webbrowsern:

- Internet Explorer 6 / 7 / 8 (Active Scripting muß erlaubt werden)
- Mozilla Firefox 3.0.x, 3.5.x
- Mozilla Firefox 2.0.0.x (nur mit Plugin http://crypto.stanford.edu/jsonrequest/)
- Opera 9.x
- Google Chrome / SRWare Iron 0.2.152.0
- Safari 3.1.2 (525.21) (Windows)
- Safari 4.0.3 (Mac)

Beispiel

Hier ein einfaches Beipiel:

Anfrage

http://192.168.1.242/control?
callback=cname&cmd=get_state_actor&number=1

Auf dem EZcontrol XS1 mit der IP 192.168.1.242 wird die „virtuelle“ Seite *control* mit den entsprechenden Parametern aufgerufen.

Hierbei bedeutet:

callback : Der Variablen *callback* ist ein frei wählbarer Name (**max. Länge 19 Zeichen**) zuweisbar. Dieser spielt nur eine Rolle bei der Verwendung einer Callbackfunktion in JavaScript. **Wird dieser Parameter weggelassen, bzw. nicht zugewiesen, wird kein JSON Antwortpaket geschickt.** Dies kann zur Einbettung von Schaltlinks in HTML Seiten Sinn machen, bei denen keine Antwort erwünscht ist (wie bei EZcontrol T-10 Schaltlinks).

cmd : Command. Bestimmt den Befehl, den das XS1 ausführen soll.

Antwort

Die Antwort wird als Text mit dem Mimetype *application/json* ausgeliefert.
Dieser Typ wird wahrscheinlich mit keiner Anwendung verknüpft sein, es empfiehlt sich diesen Mimetype einfach mit einem Texteditor zu verknüpfen oder als Datei abspeichern:

```
cname({
      "type": "get_state_actor",
      "actor": {
            "number": 1,
            "name": "Lampe",
            "type": "switch",
            "value": 0.0,
            "unit": "%",
            "utime": 1224135475,
            "date": {
                  "weekday": "th",
                  "day": 16,
                  "month": 10,
                  "year": 2008
            },
            "time": {
                  "hour": 6,
                  "min": 37,
```

```
                    "sec": 55
                }
        }
})
```

Falls die Verarbeitung des Paketes mit JavaScript erfolgt, kann nun die Callbackfunktion beim Erhalt des Paketes automatisch ausgeführt werden. Im Beispiel hat diese den Namen *cname.*

JSON ist strukturell ähnlich aufgebaut wie XML und es existieren Implementierungen schon in diversen Programmiersprachen. Nähere Informationen zur JSON Syntax und Struktur finden Sie z.B. bei Wikipedia und http://www.**json**.org.

Sicherheitshinweise

Neben denen in der Bedienungsanleitung+Beiblatt genannten Sicherheitshinweisen zum Betrieb des Gerätes sind folgende speziell bei der Programmierung zu beachten:

- Wie bei allen Funkschaltsysteme im 868-868,6 MHz Bereich ist nur eine durchschnittliche maximale zeitliche Sendedauer von 1% in diesem Bereich rechtlich zulässig. Dies hat der Nutzer bei der Nutzung des Gerätes, d.h. auch durch Programmierungen im Gerät und Ansteuerung von außen, sicherzustellen.
  Dies gilt entsprechend mit einem Maximum von 10% Sendedauer im 433 MHz Bereich.
- Zyklische/automatisierte Schreibvorgänge von Konfigurationsdaten (Aktor-, Sensor-, Timer-, Script-, Raum-Einstellungen) und Uhrzeiteinstellung dürfen nicht vorgenommen werden, da diese langfristig den internen Flashspeicher abnutzen und das Gerät zerstört wird. Dies betrifft natürlich nicht das Setzen oder Auslesen von Aktor- und Sensorzuständen, **der veröffentlichte Benutzerbefehlssatz in diesem Dokument darf uneingeschränkt genutzt werden**. Bei normaler Benutzung ist eine sehr lange Lebensdauer zu erwarten, da das Gerät u.a. über eine interne Verteilung der Schreibvorgänge verfügt (Wear-Leveling). Die vom Chiphersteller garantierte Erhaltungszeit der Konfigurationsdaten liegt bei einem Minimum von 20 Jahren.

Befehlstypen Kurzerklärung

Allgemeine Befehle

get_protocol_info	Version des Kommunikationsprotokolls abfragen
get_config_info	Allgemein Gerätestatusinformationen (nicht änderbar)

Gesamtliste eines Objekttyps auslesen

Kurze Übersichtsliste über alle gespeicherten Objekte liefern die folgenden Befehle.

get_list_actors	Liste aller Aktoren
get_list_sensors	Liste aller Sensoren
get_list_timers	Liste aller Timer

Aktuellen Schaltzustand / Sensorwert auslesen

get_state_actor	Aktuellen Zustand eines Aktors auslesen oder auf der Speicherkarte gespeicherte Werte auslesen
get_state_sensor	Aktuellen Zustand eines Sensors auslesen oder auf der Speicherkarte gespeicherte Werte auslesen

Aktuellen Schaltzustand / Sensorwert setzen

Das Setzen des Zustandes eines Sensors ist nur bei virtuellen Sensoren (System *virtual)* möglich, die keine Daten von empfangenen Sensoren darstellen.
Dieses soll ermöglichen Messwerte/Zustände von Drittprogrammen in das XS1 einspeisen zu können und diese dort in Scripten zu verwenden oder auf der Speicherkarte zu protokollieren.

set_state_actor	Zustand eines Aktors setzen
set_state_sensor	Zustand eines Sensors setzen (nur System *virtual*)

Allgemeine Begrenzungen

Maximale Länge aller Namen (name=) :

19 Buchstaben (intern 20, inklusive Null-Begrenzer)

Maximale Anzahl der jeweiligen Objekttypen (number=):

SYSTEMS	15
ACTORS	64
SENSORS	64
TIMERS	128
SCRIPTS	32

(**Firmwareabhängig**, kann durch den Befehl *get_config_info* abgefragt werden)

Die **Numerierung** der Objekte und auch Aktor Unterfunktionen erfolgt generell **von 1 an**.

D.h. z.B. Aktor 1 bis 64 sind adressierbar.

Fehlermeldungen

```
01 invalid command
02 cmd type missing
03 number/name not found
04 duplicate name
05 invalid system
06 invalid function
07 invalid date/time
08 object not found
09 type not virtual
10 syntax error
11 error time range
12 protocol version mismatch
```

Beispiel:

```
cname({
     "type": "void",
     "error": "01"
})
```

Pakettypen

get_protocol_info – Protokollinformation auslesen

Anfrage (Client -> XS1):
http://192.168.1.242/control?callback=cname&cmd=get_protocol_info

Antwort (XS1 -> Client):

```
cname({
     "type": "get_protocol_info",
     "version": "11"
})
```

Die Versionsnummer ist eine positive16 Bit Ganzzahl.

Den Versionsstand dieser Protokolldokumentation können Sie der Kopfzeile oben entnehmen.

get_config_info – Geräteinformationen auslesen

Alle nicht über die Programmierschnittstelle veränderbaren Gerätedaten.

Anfrage (Client -> XS1):
http://192.168.1.242/control?callback=cname&cmd=get_config_info

Antwort (XS1 -> Client):

```
cname({
        "type": "get_config_info",
        "info": {
                "devicename": "xs1",
                "hardware": "1.2.0.0",
                "bootloader": "0.0.0.0",
                "firmware": "1.0.0.0",
                "systems": 15,
                "maxactuators": 64,
                "maxsensors": 64,
                "maxtimers": 128,
                "maxscripts": 32,
                "maxrooms": 64,
                "mac": "00:11:22:33:44:55",
                "autoip": "off",
                "current": {
                        "ip": "192.168.1.242",
                        "netmask": "255.255.255.0",
                        "gateway": "192.168.1.1",
                        "dns": "192.168.1.1"
                },
                "saved": {
                        "ip": "192.168.1.242",
                        "netmask": "255.255.255.0",
                        "gateway": "192.168.1.1",
                        "dns": "192.168.1.1"
                }
        }
})
```

devicename	Hostname (genutzt u.a. als DHCP Hostname)
hardware, bootloader, firmware	Versionsnummern

systems	Anzahl der uterstützten Funkschaltsysteme
max*	Maximale Anzahl der abspeicherbaren Objekte
mac	MAC Adresse
autoip	DHCP an/aus
current *	Aktuelle IP Einstellungen (wenn DHCP an, dann die bezogene IP)
saved *	Abgespeicherte statische IP Daten, nur wirksam, wenn DHCP ausgeschaltet ist

get_list_actuators - Liste aller Aktoren auslesen

Jeder gespeicherte Aktor beinhaltet vier Funktionen, die jeweils aus der Art der Schaltfunktion (*type*) und einer Bezeichnung (*dsc*, Description) bestehen.

Anfrage (Client -> XS1):
http://192.168.1.242/control?callback=cname&cmd=get_list_actuators

Antwort (XS1 -> Client):

```
cname({
        "type": "get_list_actuators",
        "actuator": [
                {
                        "name": "Schalter",
                        "type": "switch",
                        "value": 0.0,
                        "utime": 0,
                        "unit": "%",
                        "function": [
                                {
                                        "type": "on",
                                        "dsc": "ON"
                                },

                                {
                                        "type": "off",
                                        "dsc": "OFF"
                                },

                                {
                                        "type": "disabled",
                                        "dsc": ""
                                },

                                {
                                        "type": "disabled",
                                        "dsc": ""
                                }
                        ]
                },


...
```

```
                {
                        "name": "Actor_64",
                        "type": "disabled",
                        "value": 0.0,
                        "utime": 1238164275,
                        "unit": "%",
                        "function": [
                                {
                                        "type": "disabled",
                                        "dsc": ""
                                },

                                {
                                        "type": "disabled",
                                        "dsc": ""
                                },

                                {
                                        "type": "disabled",
                                        "dsc": ""
                                },

                                {
                                        "type": "disabled",
                                        "dsc": ""
                                }
                        ]
                }]
})
```

get_list_sensors - Liste aller Sensoren auslesen

Anfrage (Client -> XS1):
http://192.168.1.242/control?callback=cname&cmd=get_list_sensors

Antwort (XS1 -> Client):

```
cname({
        "type": "get_list_sensors",
        "sensor": [
                {
                        "name": "Aussentemperatur",
                        "type": "temperature",
                        "value": 10.0,
                        "utime": 1238164313,
                        "unit": "°C"
                },

                {
                        "name": "Aussenfeuchte",
                        "type": "hygrometer",
                        "value": 42.5,
                        "utime": 1238164313,
                        "unit": "%"
                },

...
```

get_list_timers - Liste aller Timer auslesen

Anfrage (Client -> XS1):
http://192.168.1.242/control?callback=cname&cmd=get_list_timers

Antwort (XS1 -> Client):

```
cname({
        "type": "get_list_timers",
        "timer": [
                {
                        "name": "Schalter_ein",
                        "type": "time",
                        "next": 1238251680
                },

                {
                        "name": "Timer_2",
                        "type": "disabled",
                        "next": 0
                },



...

       {
                        "name": "Timer_128",
                        "type": "disabled",
                        "next": 0
                }]
})
```

name	Timername
type	Timertype (*disabled, time, sunrise, sunset*)
next	Nächstes Schaltereignis (in 32 Bit Unix Zeit, UTC)

get_state_actuator - Aktor auslesen

Auslesen des aktuellen Wertes

Anfrage (Client -> XS1):

http://192.168.1.242/control?callback=cname&cmd=get_state_actuator&number=1

Antwort (XS1 -> Client):

```
cname({
      "type": "get_state_actuator",
      "actuator": {
            "number": 1,
            "name": "Lampe",
            "type": "switch",
            "value": 0.0,
            "unit": "%",
            "utime": 1224135475,
            "date": {
                  "weekday": "th",
                  "day": 16,
                  "month": 10,
                  "year": 2008
            },
            "time": {
                  "hour": 6,
                  "min": 37,
                  "sec": 55
            }
      }
})
```

Der Timestamp gibt den Zeitpunkt des letzten Schaltens per XS1 bei unidirektionalen Aktoren an, bzw. den aktuellen Aktorstatus bei bidirektionalen.

utime	Zeit in Unix Zeit (Sekunden seit 01.01.1970) bei letztem Empfang **(UTC Zeit)**
date	Datum bei letztem Empfang **(lokale Zeit)**
time	Zeit bei letztem Empfang **(lokale Zeit)**

set_state_actuator - Aktor Zustand setzen

PRESET MODE (Abgespeicherte Funktion eines abgespeicherten Aktors auslösen)

Es wird die Nummer des Aktors und die Nummer der Aktorfunktion (1..4) angegeben.

Anfrage (Client -> XS1):
http://192.168.1.242/control?callback=cname&cmd=set_state_actuator&number=1&function=1

Antwort (XS1 -> Client):

```
cname({
     "type": "set_state_actuator",
     "actuator": {
          "number": 1,
          "name": "Lampe",
          "type": "switch",
          "value": 100.0,
          "unit": "%",
          "utime": 1224135475,
          "date": {
               "weekday": "th",
               "day": 16,
               "month": 10,
               "year": 2008
          },
          "time": {
               "hour": 6,
               "min": 37,
               "sec": 55
          }
     }
})
```

utime	Zeit in Unix Zeit (Sekunden seit 01.01.1970) bei letztem Empfang **(UTC Zeit)**
date	Datum bei letztem Empfang **(lokale Zeit)**
time	Zeit bei letztem Empfang **(lokale Zeit)**

set_state_actuator - DIRECT MODE (Wert eines abgespeicherten Aktors direkt setzen)

Es wird die Nummer des Aktors und der zu setzende Wert (value) angegeben.

Anfrage (Client -> XS1):
http://192.168.1.242/control?callback=cname&cmd=set_state_actuator&number=1&value=50

Antwort (XS1 -> Client):


```
cname({
     "type": "set_state_actuator",
     "actuator": {
          "number": 1,
          "name": "Lampe",
          "type": "dimmer",
          "value": 50.0,
          "unit": "%",
          "utime": 1224135475,
          "date": {
               "weekday": "th",
               "day": 16,
               "month": 10,
               "year": 2008
          },
          "time": {
               "hour": 6,
               "min": 37,
               "sec": 55
          }
     }
})
```


utime	Zeit in Unix Zeit (Sekunden seit 01.01.1970) bei letztem Empfang **(UTC Zeit)**
date	Datum bei letztem Empfang **(lokale Zeit)**
time	Zeit bei letztem Empfang **(lokale Zeit)**

set_state_actuator - EZcontrol T-10 kompatibler „PRESET MODE“ zum Aktorschalten

Beispielaufruf:

http://192.168.1.242/preset?switch=1&value=on

Schaltet den im EZcontrol XS1 oder T-10 auf Speicherplatz 1 konfigurierten Schalter ein.

Erforderliche Parameter:

Parametername	**Wertebereich EZcontrol T-10**	**Wertebereich EZcontrol XS1**	
switch	1..16	1..64	Aktorspeicherplatznummer
value	on/off **ab FW 2.25 zusätzlich:** 0..100	on/off **oder:** 0...100	Schaltwert **T-10:** bei „on“ der gespeicherte Wert gesendet **XS1:** bei „on“ wird 100 % (an) gesendet

get_state_sensor - Sensor auslesen

Anfrage (Client -> XS1):

http://192.168.1.242/control?
callback=cname&cmd=get_state_sensor&number=1

Antwort (XS1 -> Client):


```
cname({
        "type": "get_state_sensor",
        "sensor": {
                "number": 1,
                "name": "Sensor_1",
                "type": "unknown",
                "value": 0.00,
                "status": [ ],
                "unit": "none",
                "timestamp": {
                        "utime": 0,
                        "date": {
                                "weekday": "tu",
                                "day": 1,
                                "hour": 1,
                                "month": 1,
                                "year": 1970
                        },
                        "time": {
                                "hour": 1,
                                "min": 0,
                                "sec": 0
                        }
                }
        }
})
```


status	Optionale Meldungen des Sensors („Battery empty“ etc.)
utime	Zeit in Unix Zeit (Sekunden seit 01.01.1970) bei letztem Empfang **(UTC Zeit)**
date	Datum bei letztem Empfang **(lokale Zeit)**
time	Zeit bei letztem Empfang **(lokale Zeit)**

get_state_sensor – Auslesen des aktuellen Wertes und alter, auf der Speicherkarte gespeicherter Werte

Anfrage (Client -> XS1):

http://192.168.1.242/control?callback=cname&cmd=GET_STATE_SENSOR&number=4&sutime=1224108000&eutime=1224151199

sutime	start utime Zeitpunkt in der Vergangenheit ab dem die aufgezeichneten Werte abgefragt werden sollen, Zeit in Unix Zeit (Sekunden seit 01.01.1970) **(UTC Zeit)**
eutime	end utime Zeitpunkt bis zu dem die Werte ausgeliefert werden sollen (exklusiv der letzten Sekunde) Zeit in Unix Zeit (Sekunden seit 01.01.1970) **(UTC Zeit)**

Alternativ ist die Abfrage auch mit detailierter Zeitangabe in lokaler Zeit möglich:

sday, smonth, syear, shour smin, ssec,	Startzeit Zeitpunkt bis zu dem die Werte ausgeliefert werden sollen (exklusiv der letzten Sekunde) **(lokale Zeit)**
eday, emonth, eyear, ehour emin, esec,	Endzeitzeit Zeitpunkt bis zu dem die Werte ausgeliefert werden sollen (exklusiv der letzten Sekunde) **(lokale Zeit)**

Antwort (XS1 -> Client):

```
cname({
      "type": "get_state_sensor",
      "sensor": {
            "number": 4,
            "name": "Aussentemperatur",
            "type": "temperature",
            "value": 9.7,
            "status": [ ],
            "unit": "°C",
            "utime": 1224174070,
            "date": {
                  "weekday": "th",
                  "day": 16,
                  "month": 10,
                  "year": 2008
                  },
            "time": {
                  "hour": 17,
                  "min": 21,
                  "sec": 10
            },

            "data": [
            { "utime": 1224117198,"value": 9.1 },
            { "utime": 1224117375,"value": 9.7 },
            { "utime": 1224117552,"value": 9.8 },
            { "utime": 1224117729,"value": 9.3 },
...

            { "utime": 1224150304,"value": 18.5 },
            { "utime": 1224150481,"value": 18.8 },
            { "utime": 1224150658,"value": 17.4 },
            { "utime": 1224150835,"value": 17.0 },
            ( "utime": 1224151189,"value": 13.2 }
      ]
}
})
```

Das Antwortpaket entspricht dem Antwortpaket bei der Abfrage des aktuellen Status, hinzukommen die jeweils für die Vergangenheit gespeicherten Daten im *data* Block.

data utime	Zeit des Empfangs in Unix Zeit (Sekunden seit 01.01.1970) **(UTC Zeit)**
data value	Empfangener Sensorwert

get_state_sensor – Auslesen des aktuellen Wertes und der auf der MMC Karte gespeicherten Statistikdaten

Das Gerät berechnet zu jeder vollen Stunde das Minimum (min), Maximum (max) und den Durschschnittswert eines aufgezeichneten Sensors/Aktors.
Diese können mit den gleichen Zeitangaben der normalen Datenabfrage abgerufen werden.

Erklärungen zu den sonstigen Parametern finden Sie bei

get_state_sensor – Sensor auslesen, Seite 23.

Anfrage (Client -> XS1):

http://192.168.1.242/control?callback=cname&cmd=get_state_sensor&number=1&sutime=1238122800&eutime=1238140800&statistics

Antwort (XS1 -> Client):


```
cname({
        "type": "get_state_sensor",
        "sensor": {
                "number": 1,
                "name": "Aussentemperatur",
                "type": "temperature",
                "value": 9.0,
                "state": [ ],
                "unit": "°C",
                "utime": 1238168561,
                "date": {
                        "weekday": "fr",
                        "day": 27,
                        "month": 3,
                        "year": 2009
                },
                "time": {
                        "hour": 16,
                        "min": 42,
                        "sec": 41
                },
```

```
                "statistics": [
                {
                        "utime": 1238122800,
                        "avg": 6.6,
                        "min": 6.5,
                        "max": 6.8
                },

                {
                        "utime": 1238126400,
                        "avg": 6.6,
                        "min": 6.5,
                        "max": 6.8
                },

                {
                        "utime": 1238130000,
                        "avg": 6.9,
                        "min": 6.8,
                        "max": 7.0
                },

                {
                        "utime": 1238133600,
                        "avg": 7.6,
                        "min": 7.0,
                        "max": 8.8
                },

                {
                        "utime": 1238137200,
                        "avg": 9.7,
                        "min": 8.6,
                        "max": 11.5
                },

                {
                        "utime": 1238140800,
                        "avg": 10.3,
                        "min": 9.8,
                        "max": 11.0
                }]
        }
})
```

get_state_sensor – Download gespeicherter Sensordaten in CSV Format

Hier findet zusätzlich der Parametet `&format=csv` verwendung.

Erklärungen zu den sonstigen Parametern finden Sie bei

get_state_sensor – Sensor auslesen, Seite 23.

Anfrage (Client -> XS1):

http://192.168.1.242/control?callback=cname&cmd=GET_STATE_SENSOR&number=4&sutime=1224108000&eutime=1224151199&format=csv

Antwort (XS1 -> Client):

Die Antwort erfolgt mit Mimetype *application/octet-stream:*

1251756076;1.9.2009;0:1:16;+0200,S,1,Aussentemperatur;temperature;12.6
1251756253;1.9.2009;0:4:13;+0200,S,1,Aussentemperatur;temperature;12.3
1251756430;1.9.2009;0:7:10;+0200,S,1,Aussentemperatur;temperature;12.5
1251756607;1.9.2009;0:10:7;+0200,S,1,Aussentemperatur;temperature;12.6

Datensatzaufbau:

Zeit in UTC, Datum, Zeit, UTC Offset, Typ, Nummer, Name, Aktor-/Sensortyp, Wert

Typ: S = Sensor, A = Aktor

set_state_sensor - Sensor setzen

Nur in Ausnahmefällen bei „virtuellen“ Sensoren sinnvoll. Es können so z.B. auch externe Datenquellen auf der Speicherkarte mitgeloggt werden oder in Skriptentscheidungen miteinbezogen werden.

Das **Sensorsystem** muss vom Type ***virtual*** sein, damit die Werte gesetzt werden dürfen. (XS1 interne Skripte dürfen alle Sensoren setzen.)

Anfrage (Client -> XS1):

http://192.168.1.242/control?callback=cname&cmd=set_state_sensor&number=8&value=5.00

Antwort (XS1 -> Client):


```
cname({
        "type": "set_state_sensor",
        "sensor": {
                "number": 8,
                "name": "Sensor_8",
                "type": "temperature",
                "value": 5.00,
                "unit": "°C",
                "timestamp": {
                "utime": 1224174070,
                "date": {
                        "weekday": "th",
                        "day": 16,
                        "month": 10,
                        "year": 2008
                  },
                  "time": {
                        "hour": 17,
                        "min": 21,
                        "sec": 10
                  },
                }
        }
})
```

`utime`	Zeit in Unix Zeit (Sekunden seit 01.01.1970) bei letztem Empfang, bzw. gesendeten Befehls **(UTC Zeit)**
`date`	Datum bei letztem Empfangs, bzw. gesendeten Befehls **(lokale Zeit)**
`time`	Zeit bei letztem Empfangs, bzw. gesendeten Befehls **(lokale Zeit)**

subscribe - Aktor/Sensor Abonnement

Startet ein Abonnement von allen Aktor / Sensor Ereignissen.
Das XS1 behält die aktuelle Verbindung offen und liefert bei einen Schaltereignis, bzw. empfangenen Sensorwert eine Zeile mit Zeitstempel und Wert aus.

Es können verschiedene Ausgabeformate gewähl werden:

HTXT	Human readable Text, lesbarer Klartext (Tab separiert) für direkte Darstellung im Webbrowser
TXT	Daten mit Trennung durch Leerzeichen
CSV	Daten mit Trennung durch Komma
TSV	Daten mit Trennung durch Tab
JSON	Daten im JSON Format

Die Übertragung der Daten bei den Formaten HTXT / TXT / CSV / TSV erfolgt mit dem Mimetype ***text/plain; charset=UTF-8*** und im **Chunked Transfermode** (HTTP/1.1 , RFC 2616), dies sorgt dafür, dass die empfangenen Daten unmittelbar „live“ im Browser dargestellt werden und die Verbindung permanent offengehalten wird.

Anfrage (Client -> XS1):
http://192.168.1.242/control?callback=cname&cmd=subscribe&format=htxt

Antwort (XS1 -> Client):

Format *HTXT* (Human readable Text):

```
Fri, 17 Oct 2008 10:18:40 Luftdruck        barometer       997.0   hPa
Fri, 17 Oct 2008 10:18:40 Innenfeuchte     hygrometer      30.2    %
Fri, 17 Oct 2008 10:18:40 Innentemperatur  temperature     25.1    °C
Fri, 17 Oct 2008 10:18:43 FS20_FB           remotecontrol   0.0
Fri, 17 Oct 2008 10:18:56 Aussenfeuchte    hygrometer      76.8    %
Fri, 17 Oct 2008 10:18:56 Aussentemperatur temperature     10.1    °C

Zeitangabe erfolgt in lokaler Zeit.
```

Format *TXT / CSV / TSV:*

```
UNIX_Zeit(UTC) Jahr Monat Tag Wochentag Stunde Minute Sekunde Zeitzone Art Nummer Name Wert

1224247936 2008 10 17 Fri 13 52 16 +100 S 7 FS20_FB remotecontrol 0.0
1224248041 2008 10 17 Fri 13 54 1 +100 S 5 Luftdruck barometer 999.0
1224248041 2008 10 17 Fri 13 54 1 +100 S 4 Innenfeuchte hygrometer 27.6
1224248041 2008 10 17 Fri 13 54 1 +100 S 2 Innentemperatur temperature 25.1
```

Art (des Objekts):

A	Aktor
S	Sensor

EZcontrol® XS1 Protokolldokumentation

www.ezcontrol.de · Rose + Herleth GbR