brennenstuhl®

_ smart technology.

Premium-Web-Line V3

CGI specification

Version 1.3
30.01.2020

System requirements: 3.1.13 or later

Hugo Brennenstuhl GmbH & Co.
Kommanditgesellschaft
Seestr. 1-3
72074 Tübingen

Germany

Telefon: +49 0 70 71 - 8 80 10
Telefax: +49 0 70 71 - 8 76 57
www.brennenstuhl.com
info@brennenstuhl.de
ILN-Nr. 40 07 123 00000 5
Ust.-IdNr.: DE 146890047

Kommanditgesellschaft, Sitz Tübingen
Registergericht Stuttgart, HRA 380573
Persönlich haftende Gesellschafter:
H. Brennenstuhl GmbH, Sitz Tübingen
Registergericht Stuttgart, HRB 380156
Dietmar Kailer;
Geschäftsführer: Dr. Wolfgang Dehler-Brennenstuhl,
Dipl.-Kfm. Sebastian Brennenstuhl

Contents

CGI functions

---

1. cgiFktConfiguration

CGI function `/cgi/configuration` : Get number of relays and power measurement devices.

Get number of relays and power measurement devices. Request-type: GET.

Syntax:

`/cgi/configuration`

CGI-parameters:

None**.**

Reply:

`<relays>|<powmdev>]`

with:

- `relays` as the number of relays,
- `powmdev` as the number of power measurement devices.

Returns

always true.

2. cgiFktDateTimeGet

CGI function `/cgi/getDateTime` : Request date/time settings.

Retrieve the current date/time settings of the system. Request-type: GET.

Syntax:

`/cgi/getDateTime`

CGI-parameters:

None**.**

Reply:

`<TZone>|<isNTP>|<ntpServer>|<hh>|<mm>|<DD>|<MM>|<YYYY>|<valid>`

with:

- `TZone` is the time zone,
- `isNTP:` bool [0|1],
- `ntpServer` is the ntp servers url,
- `hh, mm` is the time,
- `DD, MM, YYYY` is the date,
- `valid` is a boolean value, '1' if time is valid.

Returns

always true.

3. cgiFktDeleteImage

CGI function `/cgi/delInst` : deletes a downloaded firmware image from serial data flash.

Deletes a downloaded firmware image from serial data flash. Request-type: GET.

Syntax:

`/cgi/delInst`

CGI-parameters:

None.

Reply:

There is no reply value.

Returns

always true.

4. cgiFktGetAccessRights

CGI function `/cgi/checkAcc` : Return access rights to files of current user.

Return the access rights to a list of files of the current user. Request-type: GET.

Syntax:

`/cgi/checkAcc?files=<file1>,<file2>,...,<fileN>`

CGI-parameters:

- `files=<file1>` ,<file2>,...,<fileN> is the list of N files.

Reply:

`<a1>|<a2>|...|<aN>`

with:

- `a1` as the access right to file1: 1 is granted, 0 is denied.
- `aN` as the access right to fileN: 1 is granted, 0 is denied.

Returns

always true.

5. cgiFktGetAclFSettings

CGI function `/cgi/getAclSet` : Get ACL filter settings.

Get ACL filter settings. Request-type: GET.

Syntax:

`/cgi/getAclSet`

CGI-parameters:

None.

Reply:

`<isACL>|<aclF1>|<aclF2>|<aclF3>|<aclF4>|<aclF5>|<aclF6>|<aclF7>|<aclF8>`

with:

- `isACL` being '1', if the the ACL filter is activated,
- `aclF1` ... `aclF8` being the ip address of the device being allowed for access.

Returns

always true.

6. cgiFktGetAllRelays

CGI function "`/cgi/getRelays`" : Return state of all relays.

Return state of a relay. Request-type: GET.

Syntax:

`/cgi/getRelays`

CGI-parameters:

None.

Reply:

- `[['0'|'1']'|']*` for the on/off state of the relays.

Returns

always true.

7. cgiFktGetDataFS

CGI function `/cgi/getDataFS` : Return information about the file system in the external flash memory.

Return power state. Request-type: GET.

Syntax:

`/cgi/getDataFS`

CGI-parameters:

None.

Reply:

`<SizeFS>|<FreeFS>`

with:

- `SizeFS` as the size of the file system,
- `FreeFS` as the size of the free memory in the file system.

Returns

always true.

8. cgiFktGetEnetMode

CGI function `/cgi/getEnetMode` : Get standby settings.

Get standby settings. Request-type: GET.

Syntax:

`/cgi/getEnetMode`

CGI-parameters:

None.

Reply:

`<stbyMode>.`

with:

- `stbyMode` being the standby mode, with
-
    - *0* : uses 100Base-TX or 10Base-T, whichever is the fastest supported by the partner device,
    - *1* : uses 10Base-T,
    - *2* : will start with 100Base-TX and switch back to 10Base-T when in sleep mode.

Returns

always true.

9. cgiFktGetFreeFS

CGI function `/cgi/getFreeFS` : Get free available space of the file system in the external flash memory.

Retrieve the free memory in the file system. Request-type: GET.

Syntax:

`/cgi/getFreeFS`

CGI-parameters:

None.

Reply:

`<FreeFS>`

with:

- `FreeFS` as the free flash memory in Bytes.

Returns

always true.

10. cgiFktGetFtpdSettings

CGI function `/cgi/getFtpdSet` : Get ftp daemon settings.

Get ftp daemon settings. Request-type: GET.

Syntax:

`/cgi/getFtpdSet`

CGI-parameters:

None.

Reply:

`<isFtp>|<isFtpActive>|<ftpUser>|<ftpPasswd>|<ftpWrite>`

with:

- `isFtp` being '1', if the ftp daemon is present,
- `isFtpActive` being '1', if the ftp daemon is active,
- `ftpUser` being the ftp username,
- `ftpPasswd` being the ftp password,
- `ftpWrite` being '0', if ftp is in readonly mode, '1' for read/write.

Returns

always true.

11. cgiFktGetGraphData

CGI function `/cgi/graph` : Return the logged power values for graphical presentation.

Return the logged power values for graphical presentation by page index. Request-type: GET.

Syntax:

`/cgi/graph?id=<num>&type=[0|1]`

CGI-parameters:

- `id=<num>` is an integer value containing the 0-based index of the measurement page each packet has 508 values.
- `type=` [0|1] is '0' for 10s values and '1' for 24h values.

Reply:

A binary field of data containing the requested measurement values.

Returns

always true.

12. cgiFktGetGraphDataTS

CGI function `/cgi/graphTs` : Return the logged power values for graphical presentation.

Return the logged power values for graphical presentation by time stamp. Request-type: GET.

Syntax:

```
/cgi/graphTs?id=<timestamp>&type=[0|1]
```

CGI-parameters:

- `id=<timestamp>` is the time stamp of the requested measurement page each packet has 508 values.
- `Type=` [0|1] is '0' for 10s values and '1' for 24h values.

Reply:

A binary field of data containing the requested measurement values.

Returns

always true.

13. cgiFktGetHttpSettings

CGI function `/cgi/getHttpSet` : Get web server settings.

Get web server settings. Request-type: GET.

Syntax:

```
/cgi/getHttpSet
```

CGI-parameters:

None.

Reply:

```
<toState>|<toVal>
```

with:

- `toState` being the state of timeout setting '0' = infinite, '1' = active.
- `toVal` as the value for the inactivity timeout in seconds . This is the time, after which a user has to login again after inactivity.

Returns

always true.

14. cgiFktGetIPSettings

CGI function `/cgi/getIpSet` : Get ip settings.

Get ip settings. Request-type: GET.

Syntax:

`/cgi/getIpSet`

CGI-parameters:

None.

Reply:

`<devName>|<ipMode>|<ipAddr>|<netMask>|<gwAddr>|<dnsServer>`

with:

- `devName` as the device name,
- `ipMode` as the ip mode '0': static ip address; '1': use dhcp; '2': use autoip,
- `ipAddr` as the ip address,
- `netMask` as the network mask,
- `gwAddr` as the gateway address,
- `dnsServer` as the ip address of an optional dns server.

Returns

always true.

15. cgiFktGetMailState

CGI function `/cgi/getMailSt` : requests the current state of the e-mail sending process.

Requests the current state of the e-mail sending process. Request-type: GET.

Syntax:

`/cgi/getMailSt`

CGI-parameters:

None.

Reply:

`['0'|'1'|'2']`

with:

- `'0'` : idle,
- `'1'` : ok,
- `'2'` : nok.

Returns

always true.

16. cgiFktGetPowerState

CGI function `/cgi/getPower` : Return power state.

Return power state. Request-type: GET.

Syntax:

`/cgi/getPower?Pow=<index>`

CGI-parameters:

- `Pow=<index>` is the 0-based index of the power measurement unit i.e. 0 for relay 0.

Reply:

`<power_r>|<power_a>|<energy>`

with:

- `power_r` as relative power value.
- `power_a` as absolute power value.
- `energy` as energy value.

Returns

always true.

17. cgiFktGetPwrLogSettings

CGI function `/cgi/getPwLogSet` : Get power logging settings.

Get the power logging settings. Request-type: GET.

Syntax:

`/cgi/getPwLogSet`

CGI-parameters:

None.

Reply:

`<pwLog10s>|<pwLogMail10s>|<pwLog24h>|<pwLogMail24h> .`

with:

- `pwLog10s` is '1', if 10s logging is enabled.
- `pwLogMail10s` is '1', if 10s values are sent by email.
- `pwLog24h` is '1', if 24h logging is enabled.
- `pwLogMail24h` is '1', if 24h values are sent by email.

Returns

always true.

18. cgiFktGetRelayState

CGI function `"/cgi/relaySt"` : Return state of relay.

Return state of a relay. Request-type: GET.

Syntax:

`/cgi/relaySt?Rel=<index>`

CGI-parameters:

- `Rel=<index>` is the 0-based index of the relay.

Reply:

- `["on"|"off"]` for the on/off relay state.

Returns

always true.

19. cgiFktGetRestRes

CGI function `/cgi/getRestRes` : Return success state of last restore operation `/cgi/restore` .

Return the success of the last restore operation. Request-type: GET.

Syntax:

`/cgi/getRestRes`

CGI-parameters:

None.

Reply:

`["0"|"1"]`

0 if the settings file was invalid, 1 if it was valid.

Returns

always true.

20. cgiFktGetSizeFS

CGI function `/cgi/getSizeFS` : Get the size of the file system in the external flash memory.

Retrieve the size of the external flash memory. Request-type: GET.

Syntax:

`/cgi/getSizeFS`

CGI-parameters:

None.

Reply:

`<SizeFS>`

with:

- `SizeFS` as the size of the file system in Bytes.

Returns

always true.

21. cgiFktGetSlogSettings

CGI function `/cgi/getSlogSet` : Get syslog settings.

Get syslog settings. Request-type: GET.

Syntax:

`/cgi/getSlogSet`

CGI-parameters:

None.

Reply:

`<isSlog>|<slogIP>|<slogPort>`

with:

- `isSlog` being '1', if syslog is activated,
- `slogIP` being the syslog destination ip address,
- `slogPort` being the syslog port.

Returns

always true.

22. cgiFktGetSmtpSettings

CGI function `/cgi/getEmailSet` : Get smtp settings.

Get smtp settings. Request-type: GET.

Syntax:

`/cgi/getEmailSet`

CGI-parameters:

None.

Reply:

`<isSMTP>|<smtpServer>|<smtpPort>|<smtpAccnt>|<smtpPwd>|<smtpFrom>|<smtpTo>`

with:

- `isSMTP` being '1', if smtp is active, '0' if otherwise,
- `smtpServer` as the name of the smtp server,
- `smtpPort` as the smtp port,
- `smtpAccnt` as the smtp account name,
- `smtpPwd` as the smtp password,
- `smtpFrom` as the smtp *from* entry,
- `smtpTo` as the smtp *to* entry.

Returns

always true.

23. cgiFktGetSnmpSettings

CGI function `/cgi/getSnmpSet` : Get snmp settings.

Get snmp settings. Request-type: GET.

Syntax:

`/cgi/getSnmpSet`

CGI-parameters:

None.

Reply:

`<snmpOn>|<snmpTrap>|<snmpTrapIP>|<snmpContact>|<snmpName>|<snmpLocation>|<snmpCommunity>|<snmpWrCommunity>`

with:

- `snmpOn` as being '1', if snmp is activated,
- `snmpTrap` as being '2', if snmp traps are enabled,
- `snmpTrapIP` being the trap destination ip address,
- `snmpContact` being the snmp contact string,
- `snmpName` being the snmp name string,
- `snmpLocation` being the snmp location string,

- `snmpCommunity` being the snmp community string.
- `snmpWrCommunity` being the snmp write community string.

Returns

always true.

24. cgiFktGetSwitchSettings

CGI function `/cgi/getSwitchSet` : Get switching settings.

Get the switching settings: setting after power on, on and off delay. Request-type: GET.

Syntax:

```
/cgi/getSwitchSet
```

CGI-parameters:

None.

Reply:

```
<action_0>|<onDelay_0>|<offDelay_0>|<action_1>|<onDelay_1>|<offDelay_1>
```

with:

- `action_<index>` is the action after power on, with initial state:
  - '0' : off,
  - '1' : on,
  - '2' : last mode.
- `onDelay_<index>` is the on delay in seconds after power on.
- `offDelay_<index>` is the time delay to automatically switch on after a switching off.

With index being the index of the relay.

Returns

always true.

25. cgiFktGetSysSettings

CGI function `/cgi/getSysSet` : Get system settings.

Get the system settings. Request-type: GET.

Syntax:

```
/cgi/getSysSet
```

CGI-parameters:

None.

Reply:

```
<stateMS>|<thresMSOn>|<thresMSOff>|<stateWOL>|<delayWOL>|<macWOL>|
<delayMSOn>|<delayMSOff>
```

with:

- `stateMS` is '1', if master/slave-function is on, '0' if otherwise,
- `thresMSOn` is the on-threshold for the slave in 100mW,
- `thresMSOff` is the off-threshold for the slave in 100mW,
- `stateWOL` is '1', if a WOL package will be sent after master is on, '0' if otherwise,
- `delayWOL` is the delay for WOL in ms,

- `macWOL` is the MAC address to be used in the WOL package in the form `aa.aa.aa.aa.aa.aa` ,
- `delayMSOn` is the delay in seconds before the slave is switched on,
- `delayMSOff` is the delay in seconds before the slave is switched off.

Returns

always true.

26. cgiFktGetTempSettings

CGI function `/cgi/getTempSet` : Get temperature measurement settings.

Get temperature measurement settings. Request-type: GET.

Syntax:

`/cgi/getTempSet`

CGI-parameters:

None.

Reply:

`<isTemp>|<isTActive>|<tempIntf>|<isMail>|<thresMail>|<tempActive>|`

`<tempAction>|<thresRel0>|<thresRel1>`

with:

- `isTemp` being '1', if temperature measurement hardware is present,
- `isTActive` being '1', if temperature measurement is activated,
- `tempIntf` as the temperature sensor type: '0': LM75BI2C; '1': DS18B201-wire,
- `isMail` being '1', if an email has to be sent on over temperature,
- `thresMail` as the threshold temperature for email sending,
- `tempActive` as a bitfield for relays; '1': switch relay based on temperature,
- `tempAction` as a bitfield for relays; '0': switch off, '1': switch on,
- `thresRel0` as the threshold temperature for relay 0,
- `thresRel1` as the threshold temperature for relay 1.

Returns

always true.

27. cgiFktGetTftpSettings

CGI function `/cgi/getTftpSrv` : returns the name of the TFTP server.

Returns the name of the TFTP server for software download as a single string. Request-type: GET.

Syntax:

`/cgi/getTftpSrv`

CGI-parameters:

None.

Reply:

`<servername>`

Returns

always true.

28. cgiFktGetTransferState

CGI function `/cgi/isTransfer` : queries the state of a firmware transfer.

Queries the state of the transfer of a firmware file. Request-type: GET.

Syntax:

```
/cgi/isTransfer
```

CGI-parameters:

`None.`

Reply:

```
["0"|"1"]
```

0, if there is no active transfer, 1 if there is an active transfer.

Returns

always true.

29. cgiFktGetUPnPSettings

CGI function `/cgi/getUpnpSet` : Get UPnP settings.

Get UPnP settings. Request-type: GET.

Syntax:

```
/cgi/getUpnpSet
```

CGI-parameters:

`None.`

Reply:

```
<isUPnP>|<upnpIntvl>
```

with:

- `isUPnP` being '1', if UPnP is activated,
- `upnpIntvl` being the interval for UPnP advertisements in seconds.

Returns

always true.

30. cgiFktGetWdSettings

CGI function `/cgi/getWdSet` : Get watchdog settings.

Get watchdog settings. Request-type: GET.

Syntax:

```
/cgi/getWdSet
```

CGI-parameters:

`None.`

Reply:

```
<wdActive[index]>|<wdProto[index]>|<wdIP[index]>|<wdPort[index]>|
<wdIntvl[index]>|<wdRetry[index]>|<wdAction[index]>|<wdWait[index]>|
```

`<wdDelay[index]>`

with:

- `wdActive[index]` being '1', if watchdog is active,
- `wdProto[index]` being '0' for icmp protocol, '1' for tcp protocol,
- `wdIP[index]` as the ip address to be supervised,
- `wdPort[index]` as the port to be supervised,
- `wdIntvl[index]` as the interval in seconds between pings,
- `wdRetry[index]` as the number of retries,
- `wdAction[index]` as the watchdog action, being '0' if the relay has to be switched off, '1' if it has to be switch off and after a delay wdDelay switched on again,
- `wdWait[index]` being '0', if watchdog has to be started immediately, '1' if it will wait for the first response before starting,
- `wdDelay[index]` as the delay for a reset in seconds.

With index being the index of the relay.

Returns

always true.

31. cgiFktIsFlashFs

CGI function `/cgi/isFAT` : Check, if flash file system is present.

Check, if the hardware is prepared for flash file system. Request-type: GET.

Syntax:

`/cgi/isFlashFs`

CGI-parameters:

None.

Reply:

`['0'|'1']`

a boolean value, which is '1', if flash file system is present.

Returns

always true.

32. cgiFktIsTemperature

CGI function `/cgi/isTempHw` : Check, if temperature measurement is present.

Check, if the hardware is prepared for temperature measurement. Request-type:GET.

Syntax:

`/cgi/isTempHw`

CGI-parameters:

None.

Reply:

`['0'|'1']`

a boolean value, which is '1', if temperature measurement is working.

Returns

always true.

33. cgiFktLoadFs

CGI function `/cgi/loadFs` : Upload file system image.

Upload a lmi flash file system image to the system. Request-type: POST.

Syntax:

```
/cgi/loadFs
```

CGI-parameters:

None.

Reply:

The html page `/index.htm` is returned.

Returns

False, if not all data have been receive. True, if the last package has been received and handled.

34. cgiFktLogoff

CGI function `/cgi/logoff` : Logoff current user.

Logs off the current user. Request-type: GET.

Syntax:

```
/cgi/logoff
```

CGI-parameters:

None.

Reply:

None.

Returns

always true.

35. cgiFktMarkForInst

CGI function `/cgi/markInst` : Mark downloaded firmware image for installation.

Mark a downloaded firmware image for installation.

If the flash reserved area is used: The update process will transfer the image into the flash region for the application.

If the image is a file in the file-system: The software update image file will receive the SYSTEM attribute set.

Request-type: GET.

Syntax:

```
/cgi/markInst
```

CGI-parameters:

None.

Reply:

`["0"|"1"]`

0 if file has been armed, 1 if there is no valid file to be armed.

Returns

always true.

36. cgiFktQueryImageVersion

CGI function `/cgi/getImgVer` : queries the image version of the image file in the serial data flash.

Queries the version and arm-state of the downloaded firmware image file in the serial data flash. Request-type: GET.

Syntax:

`/cgi/getImgVer`

CGI-parameters:

None.

Reply:

`<version>|["-"|"A"|"I"]`

with:

- `version` as the image file software version string.
- "-" if the image is not armed, "A" if the image is armed for installation, "I" if the image has been installed.

Returns

always true.

37. cgiFktReboot

CGI function `/cgi/reboot` : reboots the system.

Performs a system reset using SysCtlReset. Request-type: GET.

Syntax:

`/cgi/reboot`

CGI-parameters:

None.

Reply:

There is no reply value.

Returns

always true.

38. cgiFktRestore

CGI function `/cgi/restore` : Restore persistence data.

Restore device settings by setting the persistence data structure. Request-type: POST.

Syntax:

`/cgi/restore`

CGI-parameters:

None.

Reply:

The html page `/set_std.htz` is returned.

Returns

always true.

39. cgiFktSendTestMail

CGI function `/cgi/testMail` : Sends a test e-mail.

Sends a test-email with the current settings. Request-type: GET.

Syntax:

```
/cgi/testMail
```

CGI-parameters:

None.

Reply:

There is no reply value.

Returns

always true.

40. cgiFktSetDateTime

CGI function `/cgi/setDTSet` : set date/time of the device, including ntp settings. Return html page `/set_std.htz` .

This cgi-function sets the date and time settings of the device. Also the usage of the ntp protocol can be selected. Return the `/set_std.htz` html page. Request-type: GET.

Syntax:

```
/cgi/setDTSet?tzone=<tz>&BtnTIME=[NTP|MAN]&NTPserver=<host>&tHH=<hour>&
tMM=<minute>&dDD=<day>&dMM=<month>&dYY=<year>&SUB=Apply
```

CGI-parameters:

- `SUB=Apply` is the submit code.
- `tzone=<tz>` is the time zone string.
- `BtnTime=` [NTP|MAN] selects between manual and ntp time setting.

Conditional parameters:

- If `BtnTime` == NTP:
  - `NTPserver=<host>` is the ntp server as ip-address or url.
- If `BtnTime` = MAN:
  - `tHH=<hour>` are the hours of the current time.
  - `tMM=<minute>` are the minutes of the current time.
  - `dDD=<day>` is the day of the current date.
  - `dMM=<month>` is the month of the current date.
  - `dYY=<year>` is the year of the current date.

Reply:

The html page `/set_std.htz` is returned.

Returns

always true.

41. cgiFktSetEnetMode

CGI function `/cgi/setEnetMode` : Set the ethernet PHY speed. Return html page `/set_exp.ssi` .

This cgi-function sets the ethernet PHY speed. Return the `/set_exp.ssi` html page. Request-type: GET.

Syntax:

```
/cgi/setEnetMode?SB=[100BT|10BT|sleep]&SUB=Apply
```

CGI-parameters:

- `SUB=Apply` is the submit code.
- `SB=` [100BT|10BT|sleep] is the selected speed of the ethernet PHY:
  - `100BT` stands for 10Base-T or 100Base-T,
  - `10BT` stands for 10Base-T,
  - `sleep` stands for 10Base-T or 100Base-T; 10Base-T in sleep-mode.

Reply:

The html page `/set_exp.ssi` is returned.

Returns

always true.

42. cgiFktSetFactory

CGI function `/cgi/setFactory` : Reset device to factory settings.

Reset device to factory settings. Request-type: GET.

Syntax:

```
/cgi/setFactory?Reset=Reset
```

CGI-parameters:

- `Reset=Reset` is the only parameter to indicate a request for the factory reset.

Reply:

The html page `/set_std.htz` is returned.

Returns

always true.

43. cgiFktSetFileSystem

CGI function `/cgi/formatFs` : Format the flash file-system. Return html page `/set_exp.ssi` .

This cgi-function formats the flash file system and returns the available size in the flash memory in kByte. Request-type: GET.

Syntax:

```
/cgi/formatFs?Format=Fmt
```

CGI-parameters:

- `Format=Fmt` is the submit code to start the action.

Reply:

`<SizeFS>|<FreeFS>`

with:

- `SizeFS` as the size of the formatted file system,
- `FreeFS` as the size of the free memory in the file system.

Returns

always true.

44. cgiFktSetFtpdSettings

CGI function `/cgi/setFtpdSet` : Set-up the ftp daemon. Return html page `/set_lan.htz` .

This cgi-function sets-up the ftp daemon settings. Return the /set_lan.htz html page. Request-type: GET.

Syntax:

`/cgi/setFtpdSet?ftp=ftp&ftpU=<user>&ftpP=<pwd>&ftpW=ftpW&SUB=Apply`

CGI-parameters:

- `SUB=Apply` is the submit code.
- `ftp=ftp` is present, if the ftp daemon is activated.
- `ftpU=<user>` represents the ftp user name.
- `ftpP=<pwd>` represents the ftp user password.
- `ftpW=ftpW` is present, if write access is granted.

Reply:

The html page `/set_lan.htz` is returned.

Returns

always true.

45. cgiFktSetIPACLSettings

CGI function `/cgi/setIpaclSet` : Set-up the ip access control list ACL. Return html page `/set_lan.htz` .

This cgi-function sets the ip access control list ACL. Return the `/set_lan.htz` html page. Request-type: GET.

Syntax:

`/cgi/setIpaclSet&ACLF=ACLON&ACL<n>=<ip>&SUB=Apply`

CGI-parameters:

- `SUB=Apply` is the submit code.
- `ACLF=ACLON` is present, if the ACL is activated.
- `ACL<n>=<ip>` .

  Where:

  - `<n>` is the index of the ACL entry,
  - `<ip>` is the ip address of an allowed client.

Reply:

The html page `/set_lan.htz` is returned.

Returns

always true.

46. cgiFktSetIPData

CGI function `/cgi/setIP` : set the ip-settings of the device. Set the ip settings of the device.

CGI function to set the ip settings of the device. Request-type: GET.

Syntax:

```
/cgi/setIP?DevName=<name>&GetIP=<type>&hostip=<h_ip>&
netmask=<mask>&gateway=<g_ip>&SUB=Apply
```

CGI-parameters:

- `SUB=Apply` is the submit code.
- `DevName=<name>` is the name of the device.
- `GetIP=` [DHCP|Man] is the address resolution mode. **DHCP** activates the dhcp client, **Man** addresses a manual entry of data.
- `hostip=<h_ip>` is the host ip-address of the device, transferred as a string.
- `netmask=<mask>` is the network mask, transferred as a string.
- `gateway=<g_ip>` is the gateway ip-address of the device, transferred as a string.
- `dns=<d_ip>` is the name of the dns server, transferred as string.

Reply:

The html page `/set_std.htz` is returned.

Returns

always true.

47. cgiFktSetPwrLogSettings

CGI function `/cgi/setPwLogSet` : set-up power logging feature. Return html page `/set_std.htz` .

This cgi-function sets the behavior of power logging. Return the `/set_std.htz` html page. Request-type: GET.

Syntax:

```
/cgi/setPwLogSet?Plog=[10s|24h|all]&Pmail=[10s|24h|all]&SUB=Apply
```

CGI-parameters:

- `SUB=Apply` is the submit code.
- `Plog=` [10s|24h|all] switches the power logging.
  - `10s` sets logging to 10s-values only,
  - `24h` sets logging to 24h-values only,
  - `all` sets logging for all values.
- `Pmail=` [10s|24h|all] switches email setting.
  - `10s` enables email for 10s-values only,
  - `24h` enables email for 24h-values only,
  - `all` enables email for all values.

Reply:

The html page `/set_std.htz` is returned.

Returns

always true.

48. cgiFktSetRelay

CGI function `/cgi/setRelay` : Set relay.

Set a relay. Request-type: GET.

Syntax:

```
/cgi/cgiFktSetRelay?Rel=<index>&State=[0|1]
```

CGI-parameters:

- `Rel=<index>` is the 0-based index of the relay.
- `State=` [0|1] is the action off/on.

Reply:

There is no reply value.

Returns

always true.

49. cgiFktSetScheduler

CGI function `/cgi/setScheduler` : Set-up the schedulers.

CGI function to set-up the schedulers. Request-type: POST.

Syntax:

```
/cgi/setScheduler?SUB=<action>&Rel<n>=<index>&Act<n>=<[0|1]>&
Mod<n>=<[0|1]>&Mav=<rand>&MSqE=<seqOn>&MSqA=<seqOff>&MSqD=<delay>
```

CGI-parameters:

- `SUB=Apply` is the submit code.
- `Rel<n>=` [0|1] is the relay number.
- `Act<n>=` [0|1] is the action off/on.
- `Mod<n>=` [0|1] is the scheduler mode day/weekday.
- `vcf<n>=<x>` switches on vacation function with x='V'.
- `Mav=<rand>` is the randomize number of minutes for the vacation function between 0 and rand minutes.
- `MSqE=<seqOn>` is the number of minutes to switch sequence on.
- `MSqA=<seqOff>` is the number of minutes to switch sequence off.
- `MSqD=<delay>` is the number of minutes until sequence starts.

Conditional parameters:

- If `Mod` == 0:
  - `Ddd<n>=<day>` is the day of month.
  - `Dmm<n>=<month>` is the month of year.
  - `Dyy<n>=<year>` is the year.
- If `Mod` == 1:
  - `Day<n>=<weekday-bitfield>` is the bit field for the weekday, with 0=Monday, 1=Tuesday, 2=Wednesday, ...
  - `Thh<n>=<hour>` is the hour.
  - `Tmm<n>=<minute>` is the minute.

  Where:

  `<n>` is the timer number 0-based index.

Reply:

The html page `/timer.ssi` is returned.

Returns

always true.

50. cgiFktSetSmtpSettings

CGI function `/cgi/setEmailSet` : Set-up the e-mail client. Return html page `/set_std.htz` .

This cgi-function sets the parameters for the e-mail client. Return the `/set_std.htz` html page. Request-type: GET.

Syntax:

```
/cgi/setEmailSet?smtp=smtp&srvr=<server>&port=<port>&accnt=<account>&
pwd=<passwd>&eFrom=<from>&eTo=<to>&SUB=Apply
```

CGI-parameters:

- `SUB=Apply` is the submit code.
- `smtp=smtp` is present, if the e-mail client is activated.
- `srvr=<server>` is the smtp e-mail server url.
- `port=<port>` is the smtp e-mail server port.
- `accnt=<account>` is the e-mail account.
- `pwd=<passwd>` is the e-mail account password.
- `eFrom=<from>` is the e-mail originator.
- `eTo=<to>` is the e-mail recipient.

Reply:

The html page `/set_std.htz` is returned.

Returns

always true.

51. cgiFktSetSnmpSettings

CGI function `/cgi/setSnmpSet` : Set-up the snmp settings. Return html page `/set_lan.htz` .

This cgi-function sets up the snmp agent settings. Return the /set_lan.htz html page. Request-type: GET.

Syntax:

```
/cgi/setSnmpSet?StSNMP=SS1&StTrap=ST1&TDIP=<ip>&Cnt=<contact>&
Name=<name>&Loc=<position>&Comm=<community>&WrComm=<community>&SUB=Apply
```

CGI-parameters:

- `SUB=Apply` is the submit code.
- `StSNMP=SS1` is present, if SNMP is activated.
- `StTrap=ST1` is present, if SNMP traps are activated.
- `TDIP=<ip>` is the trap destination ip address.
- `Cnt=<contact>` is the contact string.
- `Name=<name>` is the name string.
- `Loc=<position>` is the position string.
- `Comm=<community>` is the community string.
- `WrComm=<community>` is the write community string.

Reply:

The html page `/set_lan.htz` is returned.

Returns

always true.

52. cgiFktSetSwitchSettings

CGI function `/cgi/setSwitchSet` : set behavior of the relays. Return html page `/set_std.htz` .

This cgi-function sets the behavior of the relays: name, delays, behavior after a system startup etc. Return the `/set_std.htz` html page. Request-type: GET.

Syntax:

```
/cgi/setSwitchSet?Name<index>=<name>&Btn<index>=[Off|On|Last]&
Sw<index>s=<s1>&SwO<index>s=<s2>&SUB=Apply
```

CGI-parameters:

- `SUB=Apply` is the submit code.
- `Name<index>=<name>` is the name of the relay.
- `Btn<index>=` [Off|On|Last] is the behavior after power on:
  - `Off` leaves the relay in off mode after power on,
  - `On` switches the relay on after power on,
  - `Last` restores the state of the relay before power off.
- `Sw<index>s=<s1>` is the delay time in seconds to switch the relay on after a power on.
- `SwO<index>s=<s2>` is the period in seconds after which the relay is switched on again after switching it off.

Where `<index>` is the 0-based index of the relay.

Reply:

The html page `/set_std.htz` is returned.

Returns

always true.

53. cgiFktSetSysFunction

CGI function `/cgi/setSysSet` : set system settings of the device. Return html page `/set_std.htz` .

This cgi-function sets the system settings for Master/Slave as well as Wake-On-LAN. Return the `/set_std.htz` html page. Request-type: GET.

Syntax:

```
/cgi/setSysSet?BtnWOL=WOL&WOLval>=<time>&WOLmac=<mac>&BtnMS=MS&
MSOn=<threshold>&MSOff=<threshold>&SlaveOn=<time>&SlaveOff=<time>&
Sub=Apply
```

CGI-parameters:

- `SUB=Apply` is the submit code.
- `BtnWOL=WOL` activates the **Wake-On-LAN** function.
- `WOLval=<time>` is the delay time between switching relay 0 and sending the WOL package in seconds.
- `WOLmac=<mac>` is the target MAC address of the Wake-On-LAN function. It has to follow the form xx.xx.xx.xx.xx.xx, with xx as a hexadecimal value.
- `BtnMS=MS` activates the **Master/Slave** function.

- `MSOn=<threshold>` is the threshold in Watt to switch the slave on.
- `MSOff=<threshold>` is the threshold in Watt to switch the slave off.
- `SlaveOn=<time>` is the delay time in seconds to switch the slave on.
- `SlaveOff=<time>` is the delay time in seconds to switch the slave off.

Reply:

The html page `/set_std.htz` is returned.

Returns

always true.

54. cgiFktSetSyslogSettings

CGI function `/cgi/setSyslogSet` : Set-up the syslog service. Return html page `/set_lan.htz` .

This cgi-function sets the parameters for the syslog service. Return the `/set_lan.htz` html page. Request-type: GET.

Syntax:

```
/cgi/setSyslogSet?SLState=SL1&SLip=<ip>&SLport=<port>&SUB=Apply
```

CGI-parameters:

- `SUB=Apply` is the submit code.
- `SLState=SL1` is present, if the syslog service is activated.
- `SLip=<ip>` is the ip-address of the syslog server.
- `SLport=<port>` is the port for the syslog service.

Reply:

The html page `/set_lan.htz` is returned.

Returns

always true.

55. cgiFktSetTempSettings

CGI function `/cgi/setTempSet` : Set-up temperature measurement. Return html page `/set_std.htz` .

This cgi-function sets-up temperature measurement: the interface type and actions on crossing temperature thresholds. Return the `/set_std.htz` html page. Request-type: GET.

Syntax:

```
/cgi/setTempSet?Tmss=Tmss&Tif=Tif[0|1]&Tmail=Tmail&Tm=<temp>&
T1=T1&T1V=<t1>&T1S=T1S[0|1]&T2=T2&T2V=<t2>&T2S=T2S[0|1]&SUB=Apply
```

CGI-parameters:

- `SUB=Apply` is the submit code.
- `SubmT=` [R|S] is the action:
  - R: requests a reset of the temperature min/max values,
  - S: requests to set new parameters.
- `Tmss=Tmss` is present, if temperature measurement is switched on.
- `Tif=Tif` [0|1] represents the interface type:
  - Tif0 stands for the I²S interface with a LM75B sensor,
  - Tif1 stands for the 1-wire interface with a DS18B20 sensor.
- `Tmail=Tmail` is present, if e-mail sending is activated.

- `Tm=<temp>` represents the temperature which results in sending an e-mail.
- `T1=T1` is present, if relay 1 reacts on a temperature threshold.
- `T1V=<t1>` represents the temperature on which relay 1 reacts.
- `T1S=T1S` [0|1] is the action for relay 1:
  - T1S0: the relay will be switched on,
  - T1S1: the relay will be switched off.
- `T2=T2` is present, if relay 2 reacts on a temperature threshold.
- `T2V=<t2>` represents the temperature on which relay 2 reacts.
- `T2S=T2S` [0|1] is the action for relay 2:
  - T2S0: the relay will be switched on,
  - T2S1: the relay will be switched off.

Reply:

The html page `/set_std.htz` is returned.

Returns

always true.

56. cgiFktSetTFTPServer

CGI function `/cgi/swUpdateSrv` : initiate a software update. Return html page `/update.ssi` .

This cgi-function initiates a software update. Return the `/update.ssi` html page. Request-type: GET.

Syntax:

```
/cgi/swUpdateSrv?TFTPserver=<server>&SUB=Apply
```

CGI-parameters:

- `SUB=Apply` is the submit code.
- `TFTPserver=<server>` is the name of the tftp update-server.

Reply:

The html page `/update.ssi` is returned.

Returns

always true.

57. cgiFktSetUPnPSettings

CGI function `/cgi/setUpnpSet` : Set-up the UPnP settings. Return html page `/set_lan.htz` .

This cgi-function configures the UPnP function. Return the `/set_lan.htz` html page. Request-type: GET.

Syntax:

```
/cgi/setUpnpSet?upnp=upnp&intv=<intv>&SUB=Apply
```

CGI-parameters:

- `SUB=Apply` is the submit code.
- `upnp=upnp` is present, if the UPnP function is active.
- `intv=<intv>` is the UPnP advertisement interval in seconds.

Reply:

The html page `/set_lan.htz` is returned.

Returns

always true.

58. cgiFktSetUser

CGI function `/cgi/setUser` : Set user name and password.

CGI function to set user name and password. Request-type: GET.

Syntax:

```
/cgi/setUser?User1=<username>&Pwd11=<password>&Pwd12=<password>&
User2=<username>&Pwd21=<password>&Pwd22=<password>&
User3=<username>&Pwd31=<password>&Pwd32=<password>&Apply="Apply"
```

CGI-parameters:

- `User<num>=<username>` is the user name.
- `Pwd<num>1=<password>` is the password.
- `Pwd<num>2=<password>` is the repetition of the password
- `Apply="Apply"` is the submit code.

Where `<num>` is the user number (1 = "guest", 2 = "standard", 3 = "admin").

Reply:

The html page "/user.ssi" is returned.

Returns

always true.

59. cgiFktSetWatchdog

CGI function `/cgi/setWdSet` : Set-up the watchdog function. Return html page `/set_lan.htz` .

This cgi-function sets up the watchdog function to supervise external ip based devices. Return the `/set_exp.htz` html page. Request-type: GET.

Syntax:

```
/cgi/setWdSet?WD<n>state=WD&WD<n>Type=[WDICMP|WDTCP]&WD<n>ip=<ip>&

WD<n>p=<port>&WD<n>Intv=<intv>&WD<n>Retry=<retry>&
WD<n>Wait=WA&WD<n>Act=WD<n>Act[0|1]&SUB=Apply
```

CGI-parameters:

- `SUB=Apply` is the submit code.
- `WD<n>state=WD` is present, if the watchdog function is activated.
- `WD<n>Type=` [WDICMP|WDTCP] stands for the watchdog type:
  - `WDICMP` stands for an ICMP ping,
  - `WDTCP` stands for a TCP ping.
- `WD<n>ip=<ip>` is the supervised ip address.
- `WD<n>p=<port>` is the port of the supervised device.
- `WD<n>Intv=<intv>` is the ping interval.
- `WD<n>Retry=<retry>` is the retry count.
- `WD<n>Wait=WA:` if present, the watchdog function starts after a first ping has been answered. This avoids a permanent reset of devices with long boot-times.
- `WD<n>Act=WD<n>Act` [0|1] is the watchdog action:
  - `0` stands for the "switch off" action,
  - `1` stands for the "reboot" action.

Reply:

The html page `/set_lan.htz` is returned.

Returns

always true.

60. cgiFktSetWebSrvSettings

CGI function `/cgi/setWSTimeout` : Set-up the web server. Return html page `/set_exp.ssi` .

This cgi-function sets the parameters for the web server. Return the /set_exp.ssi html page. Request-type: GET.

Syntax:

```
/cgi/setWSTimeout?Wact=Wact&IATime=<interval>&SUB=Apply
```

CGI-parameters:

- `SUB=Apply` is the submit code.
- `Wact=Wact` is present, if the web server has an inactivity timeout.
- `IATime=<interval>` defines the inactivity timeout in seconds.

Reply:

The html page `/set_exp.ssi` is returned.

Returns

always true.

61. cgiFktSwUpdate

CGI function `/cgi/downld` : download a firmware image.

Downloads a firmware image from tftp-server for software update, if available. Request-type: GET.

Syntax:

```
/cgi/downld
```

CGI-parameters:

None.

Reply:

There is no reply value.

Returns

always true.

62. cgiFktTemperatureGet

CGI function `/cgi/getTemp` : Request temperature values.

Retrieve the current temperatures of the system. Request-type:GET.

Syntax:

```
/cgi/getTemp
```

CGI-parameters:

None.

Reply:

```
<TAct>|<TMin>|<TMax>]
```

with:

- `TAct` is the current temperature,
- `TMin` is the minimum temperature measured,
- `TMax` is the maximum temperature measured.

Returns

always true.

63. cgiFktTempReset

CGI function `/cgi/TempReset` : Resets the Min/Max values of temperature measurement.

Resets the Min/Max values of temperature measurement. Request-type: GET.

Syntax:

```
/cgi/TempReset
```

CGI-parameters:

None.

Reply:

There is no reply value.

Returns

always true.

64. cgiFktToggleRelay

CGI function `/cgi/toggleRelay` : Toggle relay.

Toggle a relay. Request-type: GET.

Syntax:

```
/cgi/toggleRelay?Rel=<index>
```

CGI-parameters:

- `Rel=<index>` is the 0-based index of the relay.

Reply:

There is no reply value.

Returns

always true.