



Technische Dokumentation

WebControlModul



Copyright Page

Die in diesem Dokument enthaltene Information kann ohne Vorankündigung geändert werden und stellt keine Verpflichtung seitens F. LifeArt Technologie Electronic GmbH dar. Die in diesem Dokument beschriebene Hardware/Software und/oder die verwendeten Bibliotheken bzw. Datenbanken werden unter einer Lizenz- und unter einer Geheimhaltungsvereinbarung zur Verfügung gestellt. Die Software und/oder Bibliotheken und Datenbanken dürfen nur nach Maßgabe der Lizenzvereinbarungen benutzt und kopiert werden. Es ist rechtswidrig, die Software auf ein anderes Medium zu kopieren, soweit das nicht ausdrücklich in der Lizenz- oder Geheimhaltungsvereinbarung erlaubt ist. Der Käufer darf eine Kopie der Software zu Sicherungszwecken anfertigen. Ohne ausdrückliche, schriftliche Erlaubnis von F. LifeArt Technologie Electronic GmbH dürfen für andere Zwecke als dem persönlichen Gebrauch durch den Käufer diese Dokumentation und/oder die Datenbanken nicht in irgendeiner Form, mit irgendwelchen Mitteln, elektronisch oder mechanisch, mittels Fotokopie, durch Aufzeichnung oder mit Informationsspeicherungs- und Informationswiedergewinnungssystemen reproduziert oder übertragen werden.

Copyright 1996 - 2006 F. LifeArt Technologie Electronic GmbH. Alle Rechte vorbehalten.

In dieser Dokumentation sind auch Funktionen beschrieben, die zum Teil Optionen darstellen. Unsere Geräte dürfen nur von Fachpersonal angeschlossen werden!
Unsere Geräte dürfen nicht in lebenserhaltenen Funktionen eingesetzt werden!



Inhaltverzeichnis

1.	Hardware	4
	Abb. 1 Blockschaltbild	4
	Abb. 2 Layout	4
	Abb. 3 Abmessungen	5
2.	Technische Daten	6
3.	Software	9
4.	Bedienung des WCM	10
	4.1 Bedienung über IR	10
	4.2 Bedienung über Webbrowser	10
	4.2.1 Beispiel einer Steuerungswebseite	10
	4.2.2 Relais-Seite	11
	4.2.3 Dimmer-Seite	12
	4.2.4 IR-Seite	13
	4.3 Bedienung über digitale Eingänge	13
	4.4 Bedienung über Zeitablaufplan	14
	Anhang A	15
	Anhang B	17



1. Hardware

Abb. 1 Blockschaltbild

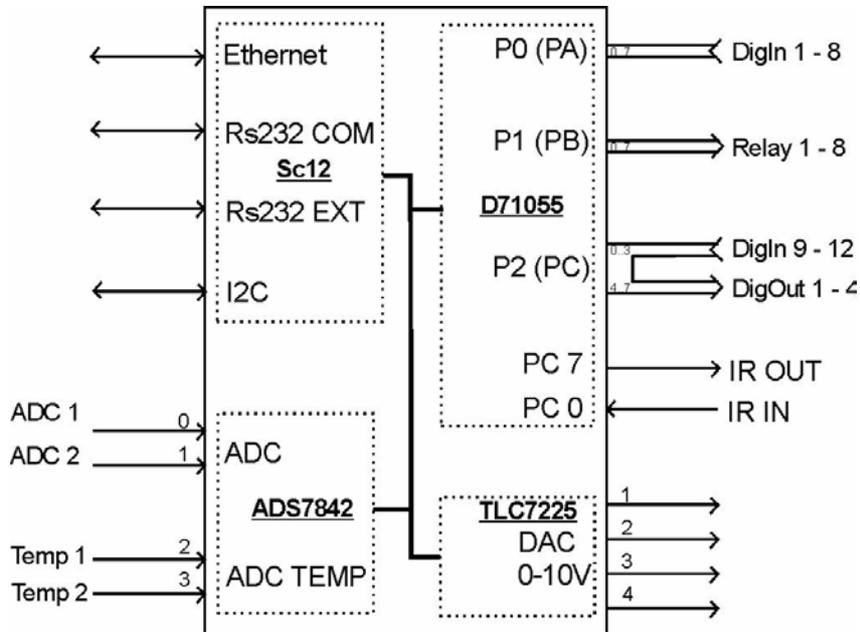


Abb. 2 Layout

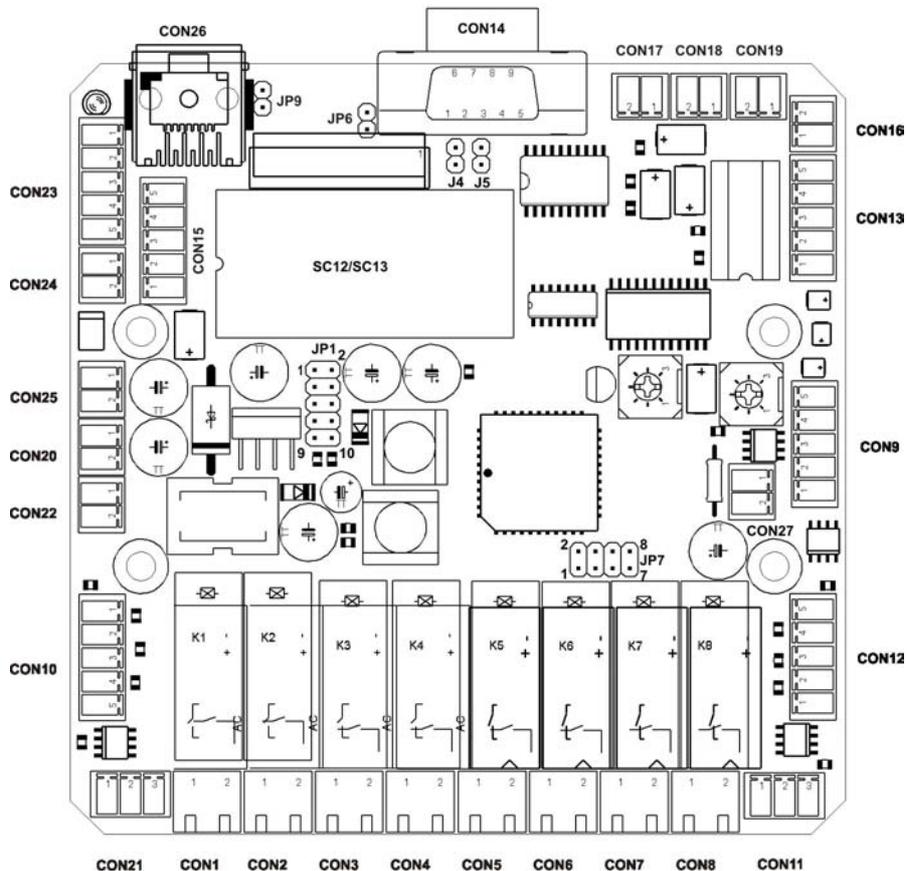
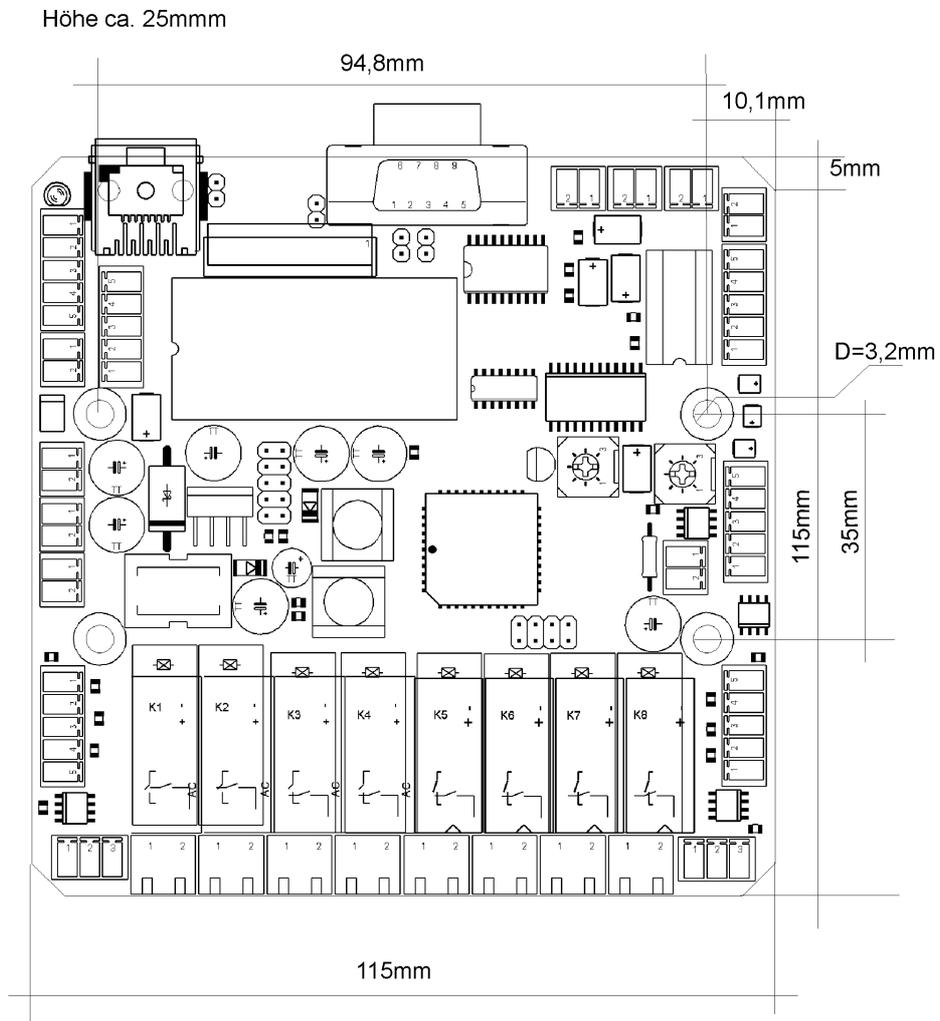




Abb. 3 **Abmessungen**



2. Technische Daten

Netzwerkanschluss:	1 x 10BaseT autosensing, RJ45 (mit SC12 Prozessor) 1 x 10/100BaseT autosensing, RJ45 (mit SC13 Prozessor)
Serielle Ports:	2 x bidirektionale RS-232 Schnittstellen
Anzeigen:	Status-LEDs Netzwerk
Software-Schnittstellen:	WEBSERVER HTTP FTP TELNET POP3 DHCP PPP TCP- und UDP-Sockets, Client und Server SMTP (E-Mail) Watchdog Timer
Weitere Schnittstellen:	Infrarot Schnittstelle (IR) mit Lernfunktion I ² C BUS
Digitale Ausgänge:	8 x Relaisausgänge (230V, 8A, 1S) 4 x OC (ICDC 50mA, BVCEO, 70 V) kurzschlussfest Programmierbare Logikfunktionen
Digitale Eingänge:	12 x optisch isolierte Eingänge In 4 Gruppen galvanisch getrennt (2x4 Eingänge + 2x2 Eingänge) max. Eingangsspannung +/-48V Schwelle 5V +/- 1V "Ein"-Strom = 2,2 mA Programmierbare Ereignis -und Logikfunktionen
Analoge Ausgänge:	4 x Ausgänge 0-10V kurzschlussfest
Analoge Eingänge:	4 x Analogeingänge 12 bit (0-10V) davon 2 als Temperaturmesssensoren-Eingänge
Messsensor:	LM335 1mV/1K° Messbereich: -40C° bis 100C°
Anschlüsse:	8x 2-fach Schraubsteckklemmen RM 5mm 9 x 2-fach Schraubsteckklemmen RM 3,5mm 2 x 3-fach Schraubsteckklemmen RM 3,5mm 6 x 5-fach Schraubsteckklemmen RM 3,5mm 1 x RJ45 Buchse (180° oder 90°) 1 x SUB-D Buchse 9pol.
Galvanische Trennung:	Netzwerkanschluß: 1500 VRMS Digitale Eingänge und Ausgänge: min. 3000 VRMS Relaisausgänge – 4000 VAC
Stromversorgung:	24V DC ca. 300mA@SC13, ca. 250mA@SC12)
Lagertemperatur:	-25°C - 70°C
Betriebstemperatur:	0°C - 60°C
Massen:	115 x 115 x 25 (l x b x h)
Gewicht:	ca. 230 g

Anschlussbelegung

Connector	Pin	Bezeichnung	Port	Connector	Pin	Bezeichnung	Port
CON1	1,2	Relaisausgang 1	PB0	CON16	1	Temperatur-Sensor 2	+
CON2	1,2	Relaisausgang 2	PB1		2	(LM135)	-
CON3	1,2	Relaisausgang 3	PB2	CON19	1	Temperatur-Sensor 1	+
CON4	1,2	Relaisausgang 4	PB3		2	(LM135)	-
CON5	1,2	Relaisausgang 5	PB4	CON17	1	Analog In 2	+
CON6	1,2	Relaisausgang 6	PB5		2		-
CON7	1,2	Relaisausgang 7	PB6	CON18	1	Analog In 1	+
CON8	1,2	Relaisausgang 8	PB7		2		-
CON9	5	OC Ausgang 1	PC4	CON20	1	Spannungsausgang 5 - 18V für externe Geräte	-
	4	OC Ausgang 2	PC5		2		+
	3	OC Ausgang 3	PC6	CON22	1	Spannungsausgang 15V für externe Geräte	-
	2	OC Ausgang 4	PC7		2		+
1	Common						
CON11	3	Common		CON23	1	I ² C Bus	GND
	2	Digital In 1	PA0		2		I2C DAT
	1	Digital In 2	PA1	3	I2C CLK		
CON21	1	Common		4	+5V		
	3	Digital In 3	PA2	5	NC		
	2	Digital In 4	PA3	CON24	1,2	VCC-Eingang	+ 24V
CON10	1	Common		CON25	1,2		GND
	5	Digital In 5	PA4	CON26	1-8	Ethernet Anschluss (RJ-45)	
	4	Digital In 6	PA5				
	3	Digital In 7	PA6				
	2	Digital In 8	PA7				
CON12	1	Common					CON27
	5	Digital In 9	PC0	2		Katode	
	4	Digital In 10	PC1				
	3	Digital In 11	PC2				
	2	Digital In 12	PC3				
CON13	5	Analog Out 1	DAC 0				
	4	Analog Out 2	DAC 1				
	3	Analog Out 3	DAC 2				
	2	Analog Out 4	DAC 3				
	1	Analog Masse					
CON14	2	RS232 (COM)	RxD				
	3		TxD				
	5		GND				
	7		RTS				
	8		CTS				
CON15	5	RS232(EXT)	RTS				
	4		CTS				
	3		RxD				
	2		TxD				
	1		GND				



Jumperbelegung

Jumper			
JP 1	1,2	Einstellung für Spannungsausgang 5 - 18V für externe Geräte	5 V
	3,4		9 V
	5,6		12 V
	7,8		15 V
	9,10		18 V
JP4	1,2	PFAIL	
JP5	1,2	RESET	
JP6	1,2	CTS	Jumper offen=> folgende Prozessor-Pins werden freigegeben: 25(CTS1), 26(RTS1) vom SC12 frei
JP9	1,2	RTS	
J7	1,2	Relaisausgang Verriegelung (offen - verriegelt)	K1 + K2
	3,4		K3 + K4
	5,6		K5 + K6
	7,8		K7 + K8

Erste Schritte

Anschließen des WCM:

- Netzteil 24V
- Netzkabel
- 9-pol. D.-Sub Kabel (1:1)

Zugriff auf das WCM:

Terminal Software

Anschlusseinstellungen:

- 19200 baud
- 8 bit
- no parity
- one stop bit
- no handshake

Telnet

Einstellungen:

- Servername(Port) : IP Adresse (z.B. 192.168.0.105)
- Benutzername : tel
- Passwort : tel

FTP

Einstellungen:

- Servername(Port) : IP Adresse (z.B. 192.168.0.105)
- Benutzername : ftp
- Passwort : ftp

HTTP Webserver

:z.B. http://192.168.0.105

CHIPTOOL

Chiptool ist eine Software der Firma Beck IPC GmbH (<http://www.beck-ipc.com>). Mit dieser Software können die o.g. Verbindungen direkt gestartet werden. Die Installationsdatei

„Chiptool_Install_V5.9.2.1.exe“ ist auf der mitgelieferten CD zu finden.

Beim Verbindungsaufbau über Netzwerk TCP/IP muss sich der PC und das WebControlModul im gleichen Subnetz befinden, z.B. 192.168.0 – Auslieferungszustand.

Informationen zur Konfiguration der IP-Adresse sind auf der CD zu finden, („IPCatCHIP_Gettingstart_V23.pdf“, ab Seite 7).



3. Software

Zu dem WebControlModul wird eine Server-Software (wcm.exe) zur Verfügung gestellt. Diese Software dient der Steuerung des WCM. Die Steuerung erfolgt über das WCM-UDP-Protokoll (siehe Anhang A) und die CGI-Schnittstelle.

Das WCM kann entweder manuell oder automatisch, mittels eines programmierten Scheduler, angesteuert werden. Der WCM-Server wird über die Datei „wcm.cfg“ konfiguriert (siehe Anhang B). Die CGI-Schnittstelle zusammen mit den Java-Applets (wcmapp.jar) ermöglicht die Erstellung von individuellen Steuerungswebseiten. Folgende Applets stehen zur Verfügung:

- digin
- switch
- light
- aircondition
- firealarm
- infrared
- transponder

Die Voraussetzung für die Verwendung der Java-Applets ist das installierte „Java Runtime Environment“.

Die Steuerungswebseiten können entweder direkt auf dem WCM gespeichert werden (Dateiname im DOS Format 8 Zeichen + 3 Zeichen Erweiterung z.B. „relais.htm“) oder auf dem PC gespeichert werden. Im Fall der Platzierung auf einem PC, müssen die Rechte für die Applets freigegeben werden. Dieses erfolgt indem die Datei .java.policy entsprechend geändert wird.

Beispiel:

```
// Detailed explanation for policy file:
// http://java.sun.com/j2se/1.4.2/docs/guide/security/PolicyFiles.html
// This file: .java.policy should be placed into user.home property
// user.home can be taken from java console (press 's')
// Usually it is something like: <System Disk>/<Documents>/<User>
// eg. C:\Dokumente und Einstellungen\Hans
// WCM Applet code permissions:
// codeBase should point to directory containing applet jar
```

```
grant codeBase "file:/C:/wcm/server/*" {
    permission java.security.AllPermission;
};
```

Weiterhin wird eine Softwarebibliothek (C++) für den Zugriff auf die WCM Hardware zur Verfügung gestellt:

Bibliothekname: wcm_hal.lib

Funktionen:

DigOut(int channel)		GetAnalogIn()
digital output		analog digital converter
channel	0-7 PortPB 0-7	channel 0 - 3 (0.00V - 10V) 12 bit (0 -4095)
RelaisOut		SetAnalogOut(value)
channel	8-11 PortPC 4-7	digital analog converter
OpenCollectorOut		channel 0-3 (0.00V-10.0V) 8 bit (0 -255)
DigIn (int channel)		
digital input		
channel	0-7 PortPA 0-7	
channel	8-11 PortPC 0-3	



Die APIs für den "SC12 Chip" sind in der Bibliothek "sc12.lib" zusammengefasst. Weitere Informationen sowie Dokumentation und Beispiele für den "SC12 Chip" sind auf der Website <http://www.beck-ipc.com/ipc> zu finden.

4. Bedienung des WCM

Die Bedienung des WebControlModuls kann über folgende Plattformen erfolgen:

- Infrarot Schnittstelle (IR)
- Webbrowser (PC, LCD Wandeinbau-Bedientableau, PDA)
- Digitale Eingänge (herkömmliche Schalter oder Taster)
- Zeitablaufplan (Scheduler)

4.1 Bedienung über IR

Das WCM mit dem optionalen IR Sender-Empfänger Modul kann über IR-Signale gesteuert werden sowie definierte Geräte über IR-Signale steuern.

Mit dem Softwaretool „IR-Lernmodul“ werden die Infrarot-Signale empfangen, dekodiert und gespeichert. Die gespeicherten IR-Signale können den Tasten auf der Steuerungswebseite zugeordnet werden. Damit kann eine Steuerung der definierten Geräte über Infrarot mittels eines Webbrowsers erfolgen. Weiterhin ist es möglich zu den gespeicherten IR-Signalen Funktionen des WCM zuzuordnen, z.B. Relais Schalten, Werte an ein Analogausgang setzen. Empfängt das WCM ein IR-Signal dem eine Funktion bereits zugeordnet ist, wird diese Funktion ausgeführt.

4.2 Bedienung über Webbrowser

Zum Abrufen einer Steuerungswebseite müssen nachstehende Arbeitsschritte erfolgen:

- Erstellung der Steuerungswebseite
- Platzierung der Steuerungswebseite entweder direkt auf dem WCM oder auf einem PC
- Abrufen im Webbrowser

Die Steuerungswebseiten können mittels HTML, Javascript, Java-Applets erstellt werden. Die Beschreibung der Javascript-Funktionen „WCM Javascript Dokumentation.html“ sind im „javadoc“-Katalog zu finden. Ein Beispiel mit Javascript ist in der Datei „jstest.htm“ zu finden

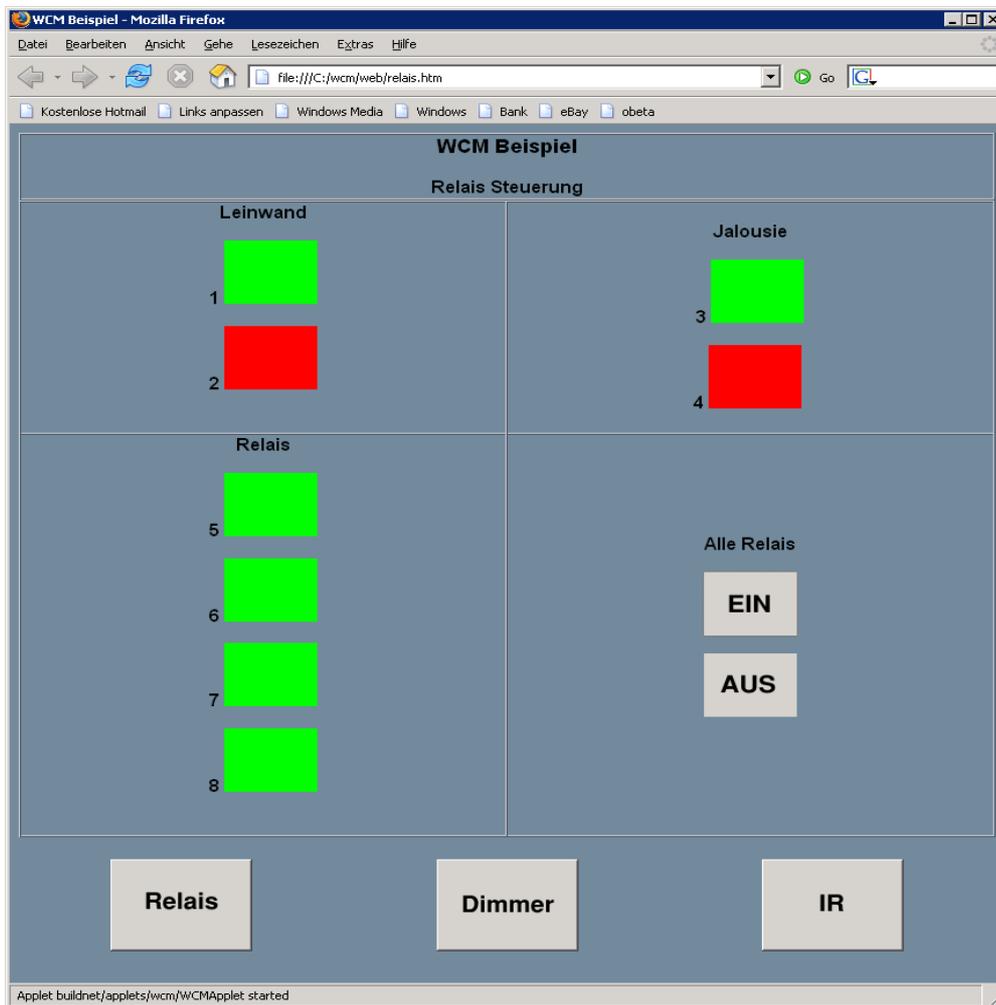
4.2.1 Beispiel einer Steuerungswebseite

Das Beispiel besteht aus drei Webseiten die über Navigationstasten abrufbar sind.

- Relais-Seite „relais.htm“
- Dimmer-Seite „dimmer.htm“
- IR-Seite „ir.htm“

Direkter Abruf der Seite unter: <http://192.168.0.105/relais.htm> oder über den Link [Hier gehts zur Beispielseite](#) von der Hauptseite <http://192.168.0.105>.

4.2.2 Relais-Seite



Funktionen	Hardware
Leinwand up	Relais 2 OFF + Relais 1 ON
Leinwand down	Relais 1 OFF + Relais 2 ON
Jalousie up	Relais 4 OFF + Relais 3 ON
Jalousie down	Relais 3 OFF + Relais 4 ON

Die Funktion „alle Relais ein/aus“ wird mit Hilfe von relon.txt und reloff.txt Scripts erstellt.

Skripte:

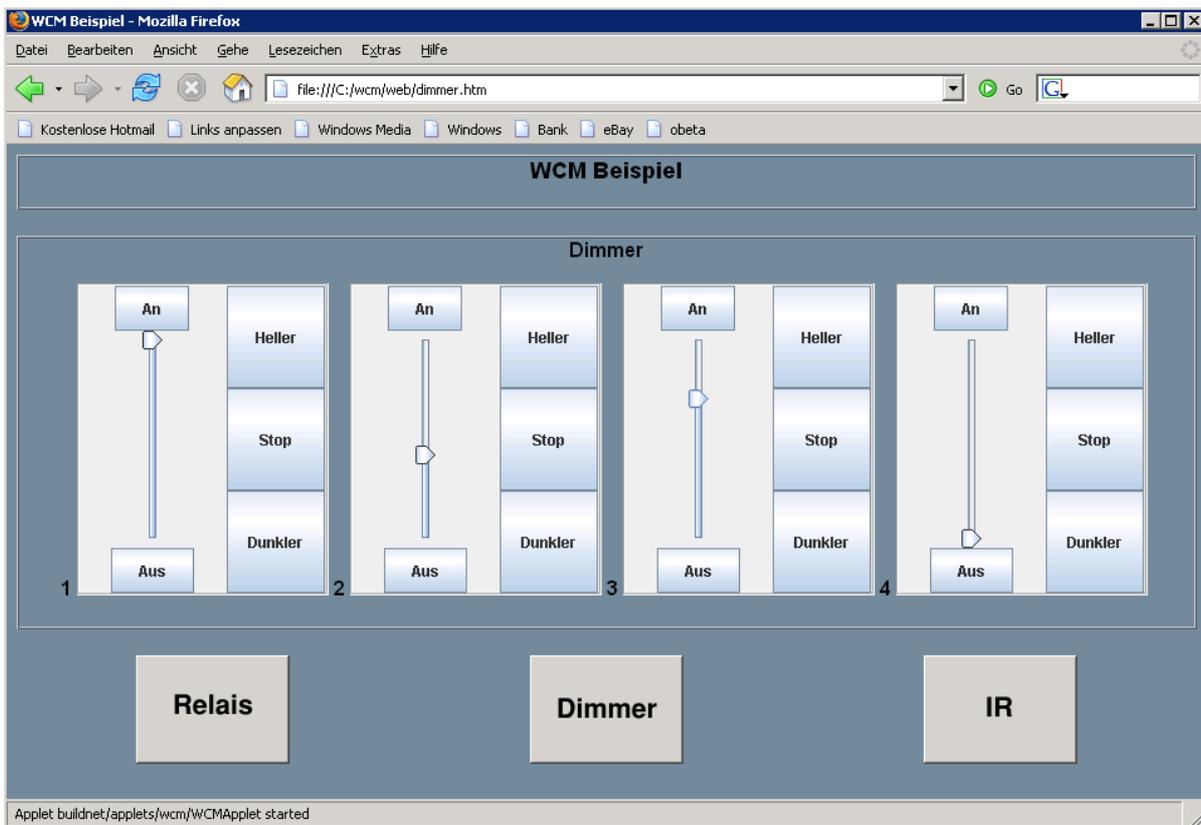
Relon.txt:

```
switch5 SWITCH ON  
switch6 SWITCH ON  
switch7 SWITCH ON  
switch8 SWITCH ON
```

Reloff.txt:

```
switch5 SWITCH OFF  
switch6 SWITCH OFF  
switch7 SWITCH OFF  
switch8 SWITCH OFF
```

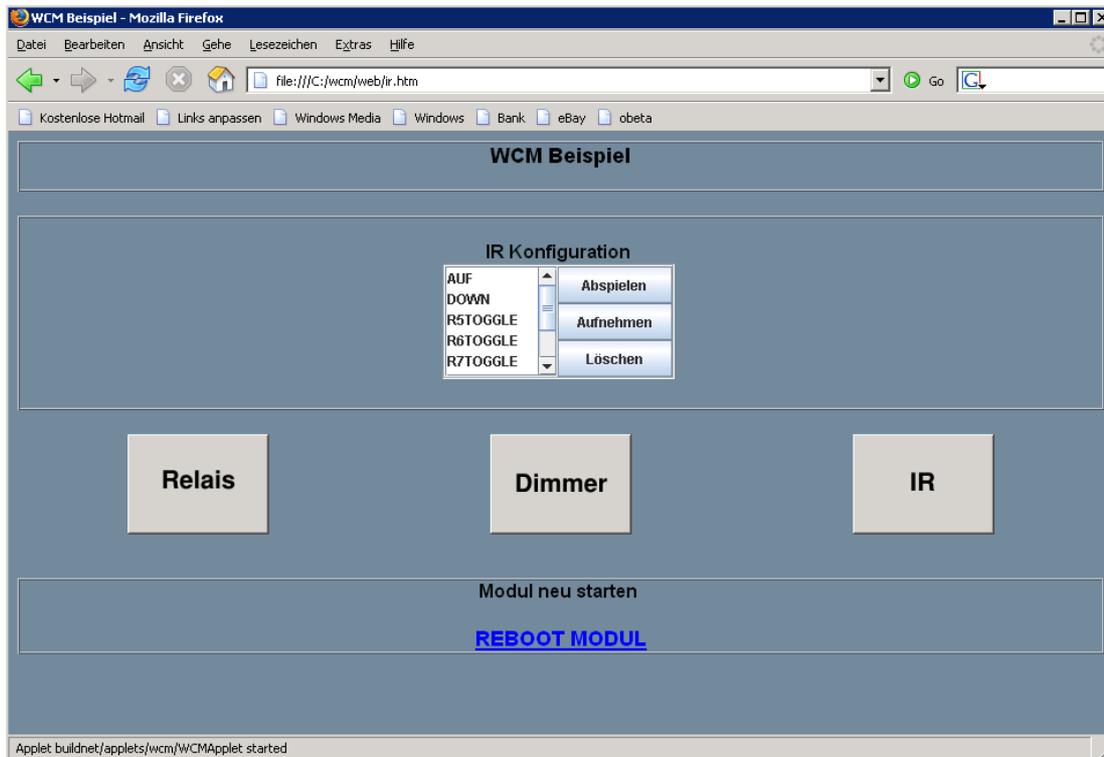
4.2.3 Dimmer-Seite



Über die Dimmer-Seite können 4 Dimmer mit der Spannung von 0-10 Volt (0-100 %) angesteuert werden.

Funktionen	Hardware
Aus	Light OFF – Analog Ausgang = 0V
Ein	Light ON – Analog Ausgang = 10V
Heller	Light BRIGHTER – Analog Ausgang = Spannung steigt
Dunkler	Light DARKER – Analog Ausgang = Spannung fällt
Stopp	Light STOP – Analog Ausgang = Erwünschter Wert erreicht

4.2.4 IR-Seite



Über diese Seite können IR-Sequenzen aufgenommen, abgespeichert und abgespielt werden.
Zwei Funktionsansätze

- WCM in der Funktion der Fernbedienung
Aufgenommene IR-Signale werden über IR-Applets oder CGI-Schnittstelle
<ahref="http://192.168.0.105/wcmserver?dev=infrared1&com=IR&arg1=PLAY&arg2=up">up
abgespielt
- WCM gesteuert über Fernbedienung
In der Konfigurationsdatei „wcm.cfg“ wird dem IR-Signal (z.B. up.ir) ein Skript zugeordnet
infrared.signal.up = up.txt
Das Skript (up.txt) beschreibt die Aktion nach Erkennung des IR-Signals.
up.txt:

```
switch2 SWITCH OFF  
switch1 SWITCH ON
```

Bei Aufnahme neuer IR-Signale bzw. Erstellung oder Änderung von Skriptdateien ist es notwendig das WCM neuzustarten, (z.B. über die REBOOT-Funktion).

4.3 Bedienung über digitale Eingänge

Die Konfiguration der Eingänge erfolgt in der Datei „wcm.cfg“. Die Änderung der „wcm.cfg“ Datei wird vom Server erkannt und das System automatisch neugestartet.

Remote Switches

Im folgenden Eintrag wird dem Eingang ein remote Digitalausgang zugeordnet. Bei Angabe eigener IP-Adresse kann auch ein lokaler Ausgang zugeordnet werden.

```
#remote switches  
digin.10.remote.relay.ip = 192.168.0.106  
digin.10.remote.relay = 5
```



Digin Light

Im folgenden Eintrag wird dem Eingang ein Analogausgang (Dimmer) zugeordnet. Die Steuerung des Analogausganges funktioniert nach dem Tastdimmer-Prinzip.

Kurz tasten - AN/AUS, länger drücken – Dimmern.

```
# digin controlling light switches
digin.1.light = 1
digin.2.light = 2
digin.3.light = 3
digin.4.light = 4
```

Digin Scripts

Im folgenden Eintrag wird dem Eingang eine Skriptdatei zugeordnet. In den Skriptdateien können alle WCM Befehle eingesetzt werden. (siehe Anhang A)

```
# digin using scripts
digin.6.script.on = blink.txt
```

```
blink.txt
light1 LIGHT SET 100
module1 PAUSE 500
light1 LIGHT SET 0
module1 PAUSE 500
light1 LIGHT SET 100
module1 PAUSE 500
light1 LIGHT SET 0
module1 PAUSE 500
light1 LIGHT SET 100
module1 PAUSE 500
light1 LIGHT SET 0
module1 PAUSE 500
light1 LIGHT SET 100
module1 PAUSE 500
light1 LIGHT SET 0
module1 PAUSE 500
light1 LIGHT SET 100
module1 PAUSE 500
light1 LIGHT SET 0
```

4.4 Bedienung über Zeitablaufplan

Zeitplanbefehle für den WCM Server werden in der Datei „wcmsched.txt“ gespeichert.

Dazu wird folgendes Format verwendet:

```
<day> <time> <command>
day          - Wochentag, monday-sunday oder * für jeden Tag
time         - Zeitangabe hh:mm:ss
command      - WCM Befehle in WCM Protokollformat
```

Beispiele:

Das Licht (light1) täglich um 9:00 Uhr einschalten

```
* 09:00:00 light1 LIGHT SET 100
```

Das Licht (light1) täglich um 18:00 Uhr ausschalten

```
* 18:00:00 light1 LIGHT SET 0
```

Ausführen des Skripts „blink.txt“ jeden Montag um 12:00 Uhr

```
monday 12:00:00 module1 SCRIPT blink.txt
```

Zeituhr mit dem SNTP Server täglich um 0:30 synchronisieren

```
* 00:30:00 module1 SYNCHRONISE_TIME
```



Anhang A

Simple text based protocol for communication between the BuildNet Server and WebControlModul.
BNS (BuildNet Server) and WCM (WebControlModul) exchange text messages.

The communication is socket based.

Public ports are 1024-65535.

BNS listens on port 1970

WCM listens on port 1971

The communication is UDP based.

Each message contains one or more lines of text.

Lines are terminated with the LF character (\n).

Each line contains text string in defined format (ASCII, 7 bit).

Communication: BNS -> WCM and wCM->BNS uses the same format

Format: deviceID COMMAND ARG1 ARG2

deviceID - string id for device. Predefined values (current hardware):

Defined commands for devices:

module1

ACCESS DENIED (only BNS->WCM)

ACCESS CONTROL <code(string)> (only WCM->BNS)

HTTP <uri> (only BNS->WCM)

send HTTP request to given URI (URL without http://)

SEND <ip> <deviceID&command&arg1&arg2> (only BNS->WCM)

send WCM command to another module.

PAUSE <time> <ms/s/min> (only BNS->WCM)

wait given time, time unit is optional (ms is default)

SCRIPT <filename> (only BNS->WCM)

DIGIN_SCRIPT_LOCK (only BNS->WCM)

lock execution of other digin scripts until end of this script,

last pending script will be then executed unless cleared

CLEAR_PENDING_SCRIPT (only BNS->WCM)

clear pending digin script

SYNCHRONISE_TIME (only BNS->WCM)

synchronise time with an SNTP server, as defined in wcm.cfg

REBOOT (only BNS->WCM)

reboot module

switch1 – switch12

SWITCH ON

set switch to on

SWITCH OFF

set switch to off

SWITCH TOGGLE

change switch to opposite state

STATE ON/OFF

only for pulse switches - inform of real state



light1 - light4

LIGHT BRIGHTER

LIGHT DARKER

LIGHT STOP

LIGHT SET <brightness 0-100%(integer)>

aircondition1 - aircondition2

AC WANTED <wanted temperature(integer)>

AC MODE ON/OFF/AUTO

AC WORKING OK/FAILURE

AC TEMP <current temperature(integer)>(only WCM->BNS)

firealarm1 - firealarm2

FIREALARM ON/OFF

infrared1

IR PLAY <filename>

play IR signal from file

IR REC <filename>

record next IR signal to file

IR REMOVE <filename>

remove IR signal file

IR LIST <file1,file2,file3>

get list of saved IR signals, in comma separated format

BNS->WCM requests the list, without args

WCM->BNS delivers list of recorded signals

digin1 - digin12

DIGIN ON/OFF (only WCM->BNS)

SCRIPT_LOCK

lock execution of other scripts for this digin until end of this script,
last pending script will be then executed unless cleared

CLEAR_PENDING_SCRIPT

clear pending digin script

Defined for each device:

GET_STATUS

get status for this device

ACTIVE ON/OFF/TOGGLE

activate, deactivate or toggle state

Beispiele:

"switch2 SWITCH OFF"

"switch5 SWITCH TOGGLE"

"light3 LIGHT SET 70"

"light1 BRIGHTER"

"aircondition1 AC WANTED 22"

"infrared1 IR PLAY up"

(up ist Dateiname mit IR Informationen die mit
„infrared1IR REC up“ aufgenommen wurde)



Anhang B

WCM Server Konfiguration (wcm.cfg)

Der WCM Server liest beim Start die Konfiguration aus der Datei „WCM.CFG“ im aktuellen Verzeichnis.

Die Datei hat einen Standard "Java Properties" Format. Es ist eine einfach lesbare und editierbare Textdatei.

Die Zeilen mit "#" Zeichen enthalten Kommentare und werden beim einlesen ignoriert.

Die Konfiguration enthält Werte für mehrere Parameter des WCM Servers, im folgenden Format:

<Parameter> = <Wert>

Die Worte in Parameter-Namen werden durch Punkt getrennt und alle klein geschrieben. Alle Parameter sind optional, Standardwerte sind in der Anwendung definiert.

Allgemeine Parameter des WCM Servers

udp.receive.port

UDP Port für den Empfang der Kommunikation vom Buildnet Server und WCM Applet.

Standardwert: 1970

udp.server

Die IP-Adresse des Buildnet Servers, der dieses WCM überwacht. Wenn kein Buildnet Server vorhanden/gewollt ist, bitte keine IP-Adresse angeben, damit unnötige Kommunikation vermieden wird.

Die IP-Adresse wird im "dotted decimal notation" angegeben, z.B. 192.168.0.1

Standardwert: kein (Buildnet Server nicht vorhanden)

udp.server.port

Empfangsport vom Buildnet Server

Standardwert: 1971

sc12.active.xxx

Dieser Parameter aktiviert und deaktiviert die WCM-Komponenten: true =aktiv, false = nicht aktiv (standardmäßig ist alles aktiviert)

sc12.active.digins = true

sc12.active.switches = true

sc12.active.lights = true

sc12.active.airconditions = true

sc12.active.firealarms = true

sc12.active.infrared = true

sc12.active.transponder = true

sc12.update.timeout

Mit diesem Parameter wird der Zeitintervall für das Ablesen des Zustands der angeschlossenen Geräte festgelegt (in Millisekunden).

Minimaler Wert: 10 Millisekunden

Standardwert: 3000

digin.click.timeout

Zeit für die "Klick" Erkennung für Digin-Lichtschalter, in Millisekunden.

Wenn ein Lichtschalter innerhalb dieser Zeit gedrückt und losgelassen wird, wird das als Umschalten interpretiert. Das Licht wird dann nicht gedimmt, sondern entweder an- oder ausgeschaltet. Wird der Schalter länger gedrückt, so wird das Licht gedimmt.

Standardwert: 500



light.dimming.delay

Dieser Parameter stellt die Geschwindigkeit des Dimmvorgangs ein.

Wert 1 = Geschwindigkeit des Dimmvorgangs ca. 2 Sek. beim dimmen von 0 bis 100%. Die Dauer des Dimmvorgangs nimmt mit steigenden Werten zu.

Standardwert: 1

sntp.server

Die IP-Adresse vom SNTP Server wird benötigt für Zeitsynchronisation. Ohne Angabe der Adresse erfolgt keine Synchronisation.

Die IP-Adresse wird im "dotted decimal notation" angegeben, z.B. 192.168.0.1

Standardwert: kein (keine Zeitsynchronisation)

sntp.timezone

Zeitzone für die Zeitsynchronisation: für Deutschland gilt MEZ=UTC+1, also +1 in dem Fall.

Standardwert: 0 (UTC Zeit wird dann angenommen)

Digin Parameter

Über diese Parameter können einzelne Digin Geräte konfiguriert und mit Funktionen versehen werden. Einzelner Digin Gerät kann nur eine Funktion übernehmen.

In dem Parameternamen muss die Digin Nummer (1-12) enthalten sein, in der Beschreibung als X symbolisiert.

digin.X.inverse

Der Digin Zustand kann invertiert werden.

Die Zuordnung der Eingangswerte werden vertauscht 0 für 1 und 1 für 0.

Beispiel:

digin.2.inverse = true

Zustand vom Digin 2 wird umgekehrt gemeldet.

digin.X.light

Mit diesem Parameter kann dem Digin die Lichtschalter- und Dimmfunktion zugeordnet werden.

Siehe "digin.click.timeout".

Der Wert bestimmt die Nummer des Lichtreglers (1-4).

Beispiel:

digin.2.light = 3

Digin Nr. 2 kontrolliert Licht Nr. 3.

digin.X.script.on

digin.X.script.off

Mit diesem Parameter wird dem Digin ein Skript zugeordnet. Es können zwei Skripte einem Eingang zugeordnet werden, bei positiven bzw. negativen Eingangssignalwechsel. Die Skripte sind Textdateien mit Kommandos aus dem WCM Protokoll (siehe Beschreibung und Beispiele), die ausgeführt werden.

Beispiel:

digin.7.script.on = dig7on.txt

digin.7.script.off = dig7off.txt

Beim Einschalten von Digin 7 wird "dig7on.txt" ausgeführt, beim Ausschalten "dig7off.txt"



digin.X.script.on.minTime

digin.X.script.off.minTime

Dieser Parameter stellt eine zusätzliche Erweiterung für die Funktion Digin Skript dar. Das Skript wird nur dann ausgeführt, wenn Digin eine definierte Mindestzeit (in Sekunden) in einem Zustand verbleibt.

Beispiel:

```
digin.7.script.on = dig7on.txt
```

```
digin.7.script.on.minTime = 3
```

Wenn Digin 7 eingeschaltet wird und Minimum 3 Sekunden in dem Zustand verbleibt, wird "dig7on.txt" ausgeführt.

digin.X.script.on.minSwitches

digin.X.script.off.minSwitches

Dieser Parameter stellt eine zusätzliche Erweiterung für die Funktion Digin Skript dar. Das Skript wird nur dann ausgeführt, wenn in einer definierten Zeitspanne (in Sekunden) ein Minimum definierter Signalwechsel am Digin stattfindet.

Beispiel:

```
digin.7.script.on = dig7on.txt
```

```
digin.7.script.on.minSwitches = 3,5
```

Wenn Digin 7 mindestens 3 mal innerhalb von 5 Sekunden eingeschaltet wird, wird dann der "dig7on.txt" Skript ausgeführt.

digin.X.remote.relay.ip

digin.X.remote.relay

Mit diesem Parameter wird dem Digin die Funktion „Digout - Fernsteuerung“ zugeordnet.

Dieser Digin wird einen Digout vom beliebigen WCM, der über TCP/IP Netzwerk erreichbar ist, steuern. Bei Angabe eigener IP-Adresse wird der lokale Ausgang gesteuert.

Die IP-Adresse ("dotted decimal notation") sowie die Nummer des gesteuerten Digouts (1-12) muss angegeben werden.

Beispiel:

```
digin.10.remote.relay.ip = 192.168.0.247
```

```
digin.10.remote.relay = 5
```

Digin 10 steuert Digout Nr. 5 vom WCM mit IP-Adresse 192.168.0.247.

Switch Parameter

Über diese Parameter können einzelne Digouts konfiguriert und mit Funktionen versehen werden.

In dem Parameternamen muss die Digout Nummer (1-12) enthalten sein, in der Beschreibung als X symbolisiert.

switch.X.crossed

Mit diesem Parameter werden zwei Digouts verriegelt in master / slave Beziehung. Beim Umschalten des Masters wird auch der "Slave" in die gegenseitige Position geschaltet.

(master ein -> slave aus, master aus -> slave ein).

Beispiel:

```
switch.1.crossed = 2
```

Digout 1 wird zum master und Digout 2 zum slave.

switch.X.remote

Mit diesem Parameter wird dem Digout die Funktion „Digout - Fernsteuerung“ zugeordnet. Dieser Digout wird einen Digout vom beliebigen WCM, der über TCP/IP Netzwerk erreichbar ist, steuern. Bei Angabe eigener IP-Adresse wird der lokale Ausgang gesteuert. Die IP-Adresse ("dotted decimal notation") sowie die Nummer des gesteuerten Digouts (1-12) muss angegeben werden.

Beispiel:

```
switch.3.remote.switch = 192.168.0.247/5
```

Switch 3 steuert Switch Nr. 5 vom WCM mit IP-Adresse 192.168.0.247.

switch.x.pulse

Mit diesem Parameter wird der Digout in Impulsmodus umgestellt. Beim Schalten des Digouts wird ein Impuls mit definierter Länge generiert. (um z.B. Stromstoßschalter anzusteuern) Impulslänge wird in Millisekunden angegeben.

Beispiel:

```
switch.1.pulse = 500
```

Bei Umschalten des Digout 1 wird ein Impuls in 500ms Länge generiert.

Infrarot Parameter

infrared.frequency

Sampling Frequenz in Hz. Standardmäßig: 4000

infrared.modulation

Modulation Frequenz in Hz. Standardmäßig: 36000

infrared.length

Maximale Signallänge in Samples. Standardmäßig: 10000

infrared.endlength

Minimale Länge vom Signalende (0). Standardmäßig: 1000

infrared.digin

Digin Nr.(0-11) der für Infrarot Signalaufnahme benutzt wird, Standardmäßig: 8

infrared.digout

Digout Nr.(0-11) der für Infrarot Signalausgabe benutzt wird, Standardmäßig: 11

Logging Parameter

Mit dem Logging Parameter wird das Loggingsystem konfiguriert. Das Loggingsystem dient zu Serverüberwachung und Fehleranalyse.

debug.level

Mit diesem Parameter wird festgelegt welche Ereignisse angezeigt werden sollen.

Detail-Stufen der Ausgabe in absteigender Wichtigkeit:

- fatal - fataler Fehler, Anwendung wird beendet (immer an)
- assert - Assertion(Programm) Fehler, Anwendung wird beendet (immer an)
- error - sonstiger Fehler (immer an)
- allerrors - alle obigen Fehlern
- warn - Warnungen (standardmäßig an)
- trace – Ausgabe der Serveraktivitäten
- time - Zeitmessung für besondere Abschnitte
- debug - detaillierte Informationen für Fehlersuche
- all - alle obigen Stufen

Mehrere Parameter werden mit einem Doppelpunkt getrennt.

Beispiel für alle Fehler und Warnungen:

```
debug.level = allerrors:warn
```



debug.format

Mit diesem Parameter wird der Format der Loggingausgabe definiert.

Folgende Informationen können ausgegeben werden:

- type - Typ der Ausgabe (Fehler, Warnung, Information, usw.)
- file - Modulname (Quellcode, Dateiname)
- func - Funktionsname
- line - Zeilennummer im Quellcode
- text - der eigentliche Loggingtext
- short - kurz Format der obigen Informationen
- date - aktuelles Datum
- time - aktuelle Uhrzeit
- difftime - Zeitunterschied (ms)
- all - alle Informationen

Mehrere Parameter werden mit einem Doppelpunkt getrennt.

debug.include

debug.exclude

Die beiden Parameter begrenzen Logging auf bestimmte Module. Diese Einstellung wird für gezielte Analysen benutzt.

Die in der include Liste aufgestellten Module werden geloggt.

Die in der exclude Liste aufgestellten Module werden nicht geloggt.

Mehrere Parameter werden mit einem Doppelpunkt getrennt.