

# **CL-Software**

# Table of Contents

<b>1. CL-Studio Software.....</b>	<b>5</b>
1.1. Titel .....	5
1.2. Cover .....	6
1.3. Allgemeine Hinweise .....	7
1.4. Erste Schritte .....	10
1.5. Software-Installation.....	17
1.6. Einstellungen HomeMatic-CCU.....	20
1.7. Installation FHZ2000-Schnittstelle .....	22
1.8. Einstellungen FHZ-Schnittstellen.....	28
1.9. Benutzung des PCs als Zentrale.....	30
1.10. Installation Hardware .....	33
1.11. Die wichtigsten Begriffe des Systems .....	34
1.12. Makro-Programmierung .....	35
1.13. Hauptfenster .....	41
1.14. Visualisierungsfenster .....	42
1.15. Das Fenster Modulauswahl.....	43
1.15.1. Das Fenster Schaltsensoren.....	45
1.15.2. Das Fenster Aktoren .....	47
1.15.3. Das Fenster Sensoren.....	48
1.16. Einstellungen.....	49
1.16.1. Seite Ansichten .....	50
1.16.2. Seite Objekte.....	51
1.16.3. Seite Makros.....	52
1.16.4. Seite Allgemein.....	53
1.16.4.1. Fenster Sonnenzeiten .....	55
1.16.4.2. Fenster Feiertage .....	56
1.16.4.3. Fenster e-mail Konfiguration .....	57
1.16.5. Seite Verzeichnisse.....	58
1.16.6. Seite Sicherheit.....	59
1.16.7. Seite PC-Modus .....	60
1.17. Fenster Hardware .....	61
1.17.1. PC als Zentrale .....	63
1.18. Objektdefinitionen.....	66
1.18.1. Seite Allgemein der Objektdefinition.....	67
1.18.2. Seite Makro .....	69
1.18.3. Seite Zeit-Tabelle.....	71
1.18.4. Seite Visualisierung .....	72
1.18.4.1. Bildsymbole für Objekte.....	73
1.19. Fenster Anwesenheitssimulation .....	74
1.20. Das Fenster Visualisierung .....	75
1.21. Fenster Makro-Anweisungen erstellen.....	77
1.21.1. Makro-Anweisungen .....	78
1.21.1.1. warten.....	79
1.21.1.2. Audio-Datei abspielen.....	80
1.21.1.3. Makro aufrufen .....	81
1.21.1.4. Makro starten .....	82
1.21.1.5. Windows-Programm starten .....	83
1.22. Die Programmierung des Systems.....	84
1.22.1. Makros.....	85
1.22.2. Variablen .....	87
1.22.3. Zuweisungen.....	89
1.22.4. Bedingungen .....	92
1.22.5. Anweisungen.....	95
1.22.5.1. ABBRECHEN .....	98
1.22.5.2. ABFRAGE .....	99
1.22.5.3. AKTIVIEREN .....	100

1.22.5.4. ALLEABFRAGEN.....	101
1.22.5.5. ALLEWERTESICHERN.....	102
1.22.5.6. ANZEIGEN.....	103
1.22.5.7. AUFRUFEN .....	104
1.22.5.8. DEAKTIVIEREN .....	105
1.22.5.9. ERLEDIGT .....	106
1.22.5.10. GEHEZU .....	107
1.22.5.11. GETCCUSYSVAR .....	108
1.22.5.12. GETSITE .....	109
1.22.5.13. GRUPPENZUWEISUNG .....	111
1.22.5.14. LESEWERTEDATEI.....	112
1.22.5.15. LÖSCHEANZEIGE.....	113
1.22.5.16. LÖSCHEDATEI .....	114
1.22.5.17. PLAY.....	115
1.22.5.18. SCHLIESSEDATEIEN .....	116
1.22.5.19. SCHREIBEDATEI .....	117
1.22.5.20. SENDE .....	119
1.22.5.21. SENDEMAIL .....	120
1.22.5.22. SENDESMS .....	122
1.22.5.23. SETCCUSYSVAR .....	124
1.22.5.24. SETZEHISTORYDIFFERENZ .....	125
1.22.5.25. SETZEKONFIG.....	126
1.22.5.26. SETZEWERT .....	127
1.22.5.27. SETZEZEITTABELLENWERT .....	130
1.22.5.28. SICHTBAR .....	131
1.22.5.29. SPRACHAUSGABE.....	132
1.22.5.30. STARTE.....	133
1.22.5.31. STARTPROGRAMM .....	134
1.22.5.32. STARTUHR .....	135
1.22.5.33. STARTWIN .....	136
1.22.5.34. STOPPE.....	137
1.22.5.35. TEMPERATURABSENKUNG.....	138
1.22.5.36. UNSICHTBAR.....	139
1.22.5.37. VERLASSEN .....	140
1.22.5.38. WARTE .....	141
1.22.5.39. WENN.....	142
1.22.6. Funktionen.....	143
1.22.6.1. AKTIVIERT .....	144
1.22.6.2. BATTERIELEER .....	145
1.22.6.3. COMERROR.....	146
1.22.6.4. DATEIVORHANDEN .....	147
1.22.6.5. DATUM .....	148
1.22.6.6. FEIERTAG .....	149
1.22.6.7. GESCHALTET .....	150
1.22.6.8. JAHR .....	151
1.22.6.9. MAILLOG.....	152
1.22.6.10. MONAT .....	153
1.22.6.11. MONATSTAG.....	154
1.22.6.12. OBJEKTBEZ .....	155
1.22.6.13. OBJEKTNAME.....	156
1.22.6.14. LÖSCHEDATEI .....	157
1.22.6.15. SCHALTDAUER.....	158
1.22.6.16. SELBST .....	159
1.22.6.17. SONNENAUFGANG.....	160
1.22.6.18. SONNENUNTERGANG.....	161
1.22.6.19. STOPPUHR .....	162
1.22.6.20. TAUPUNKT .....	163
1.22.6.21. TRIGGER.....	164
1.22.6.22. UHRZEIT .....	165

1.22.6.23. UHRZEITSEKUNDE .....	166
1.22.6.24. WOCHENTAG .....	167
1.22.6.25. ZEIT .....	168
1.22.6.26. ZUFALLSZEIT .....	169
1.22.6.27. Funktionen zur Textbearbeitung .....	170
1.22.6.27.1. ERSETZEN .....	171
1.22.6.27.2. GROSSBUCHSTABEN .....	172
1.22.6.27.3. KLEINBUCHSTABEN .....	173
1.22.6.27.4. LESETEXTPAR .....	174
1.22.6.27.5. LINKERTEIL .....	175
1.22.6.27.6. RECHTERTEIL .....	176
1.22.6.27.7. TEXTLÄNGE .....	177
1.22.6.27.8. TEXTPOSITION .....	178
1.22.6.27.9. TEXTTEIL .....	179
1.22.6.27.10. TEXTZWISCHEN .....	180
1.23. Hinweise zur Hardware .....	181
1.23.1. Datenpunkte von Geräten ändern .....	182
1.23.2. HomeMatic .....	185
1.23.2.1. Rollladen/Jalousiesteuerungen .....	186
1.23.2.2. HM-Thermostate .....	188
1.23.2.3. 19-Tasten-Fernbedienung .....	191
1.23.2.4. MP3 Funkgong mit Signalleuchte HM-OU-CFM- .....	193
1.23.2.5. MP3-Player HM-OU-CM-PCB .....	195
1.23.2.6. Statusanzeige HM-Dis-WM55 .....	196
1.23.2.7. EPaper-Display HM-Dis-EP-WM55 .....	199
1.23.2.8. WinMatic .....	201
1.23.3. HomeMatic IP .....	202
1.23.3.1. HmIP-Thermostate .....	203
1.23.3.2. HmIP-Rauchmelder .....	206
1.23.3.3. HmIP-ASIR - Alarmsirenen .....	207
1.23.3.4. HmIP-BSL - Schaltaktor mit Signalleuchte .....	209
1.23.3.5. HmIP-DLD - Türschlossantrieb .....	210
1.23.3.6. HmIP-MP3P - Kombisignalgeber .....	211
1.23.3.7. HmIP-SWD Wassersensor .....	212
1.23.3.8. HmIP-WHS2 - Heizungsaktor .....	213
1.23.3.9. HmIP-WRCD EPaper-Display .....	214
1.23.3.10. HmIP-WRCR - Drehtaster .....	217
1.23.3.11. HmIP-USBSM .....	218
1.23.3.12. HmIP-MIOB .....	219
1.23.3.13. HmIP-WUA/ELV-SH-WUA .....	220
1.23.3.14. ELV-SH-WSC .....	221
1.23.3.15. HmIP-WKP Keypad .....	222
1.23.3.16. HmIP-RGBW .....	223
1.23.3.17. HmIPW-WRC6 .....	224
1.23.3.18. HmIP-Wired .....	225
1.23.4. Shelly-System .....	226
1.23.4.1. Allgemeine Gerätekonfiguration .....	227
1.23.4.2. Shelly1(PM) mit Addons .....	229
1.23.4.3. Shelly2.5 als Rollladenaktor .....	230
1.23.5. Spezielle Gateways .....	231
1.23.5.1. Philips Hue Bridge .....	232
1.23.5.2. Osram Lightify Gateway .....	235
1.23.6. FHZ-System .....	237
1.23.6.1. Anmelden von FHT80b-Raumreglern .....	238
1.23.6.2. FS20-Rollladensteuerungen .....	240
1.23.6.3. FS20-Zustandsmelder .....	241
1.23.7. CL-control .....	242
1.23.7.1. WLAN-Schaltsteckdose CL-SDAx .....	243
1.24. Allgemeine Hinweise .....	245



1.24.1. Aktoren.....	246
1.24.2. Ausführungscode erstellen .....	247
1.24.3. Balkendarstellung.....	248
1.24.4. Bild-Symbole .....	249
1.24.5. Bitmaps .....	250
1.24.6. Dateiverwaltung auf der Zentrale.....	251
1.24.7. Dimmersteuerung .....	252
1.24.8. Duty-Cycle .....	253
1.24.9. ExecEngine.....	254
1.24.10. ExecEngineWin .....	255
1.24.11. Freigabe des Programms.....	256
1.24.12. Gruppen.....	257
1.24.13. History und Log-Dateien .....	258
1.24.14. Objekte.....	259
1.24.15. Objektdarstellung.....	260
1.24.16. Objektrahmen und Textrahmen.....	261
1.24.17. Programmierung .....	262
1.24.18. Projekt.....	263
1.24.19. Schieberegler.....	264
1.24.20. Sensoren.....	265
1.24.21. Sonnenzeiten.....	266
1.24.22. Spezielle Objekte .....	267
1.24.22.1. Typdefinitionen .....	269
1.24.23. Sprachausgabe .....	270
1.24.24. Visualisierung .....	271
1.24.25. Visualisierung mit VisuWin .....	272
1.24.26. Zeilenvorschub.....	273
1.24.27. Zustandsmelder .....	274
1.24.28. Variablen über virtuelle Objekte setzen .....	275
1.25. FAQs .....	276
1.26. Online-Informationen .....	278

# 1. CL-Studio Software

## 1.1. Titel

---

### CL-Software 4.1

---

Benutzerhandbuch  
2023

## 1.2. Cover

---

**homeputer CL Studio-Version**

**homeputer CLX**

**c-comatic**

Version 4.1

copyright (c)

CL-control 2022

[www.cl-control.de](http://www.cl-control.de)

# 1.3. Allgemeine Hinweise

---

## Allgemeine Hinweise

In diesem Dokument sind die Programmversionen CL-Studio, CLX und c-comatic von CL-control beschrieben.

Da es Unterschiede zwischen den Versionen gibt, kann es an einigen Stellen sein, dass Funktionen beschrieben sind, die in der benutzten Software-Version nicht zur Verfügung stehen.

Der generelle Unterschied zwischen den Versionen ist, dass mit der CL-Studio-Version nur eine HomeMatic-CCU eingesetzt werden kann, die gleichzeitig auch die Zentrale ist.

Eine Ausnahme ist die CL-Studio-Version für FHZ, in dieser Version kann nur der PC als Zentrale benutzt werden.

Mit der CLX-Version und der c-comatic-Version können je nach System mehrere Gateways (in HomeMatic.-Versionen HomeMatic-CCUs als Schnittstelle) eingesetzt werden und entweder eine separate Zentrale (c-comatic) oder beim HomeMatic-System eine der verwendeten CCUs als Zentrale benutzt werden.

Die **CL-Software** bietet die Möglichkeit komplexe logische Funktionen zur Steuerung des Systems auf einfache Weise zu realisieren. Dies geschieht über eine einfach zu erlernende Makrosprache mit einem grossen Funktionsumfang. Das homeputer-Programm erlaubt zudem umfangreiche individuelle Visualisierungen. In der Studio-, CLX- und c-comatic-Version können beliebig viele individuelle Ansichten erstellt werden, in kleineren Versionen ist die Anzahl der möglichen Ansichten je nach Version beschränkt.

Zur Visualisierung kann das Visualisierungsprogramm für Windows (VisuWin), eine App oder der CL-Webserver benutzt werden. VisuWin und der CL-Webserver sind Bestandteil aller Versionen ab der Studio-Version.

Das Prinzip der CL-Software sieht wie folgt aus:

Jeder Sensor (also z.B. jede Taste einer Fernbedienung) und jeder Aktor (z.B. die einzelnen Ausgänge eines 4-fach-Aktormoduls) werden in der homeputer-Software als „Objekt“ dargestellt. Es können auch Objekte ohne zugehöriges Hardware-Modul erstellt werden, diese werden dann als virtuelle Objekte bezeichnet.

Jedes Objekt hat ein eigenes Makro.

Makros von Sensor-Objekten werden normalerweise ausgeführt, wenn von der Sensor-Hardware eine Meldung empfangen wird. Dazu ist bei Sensoren standardmässig die Option "Ausführen bei Empfang" aktiviert.

Makros von Aktoren können in verschiedenen Ausführungsintervallen von mehrmals in der Sekunde bis einmal täglich aktiviert werden.

Für jedes Objekt können unterschiedliche Darstellungsformen für die einzelnen Ansichten definiert werden. Die Visualisierungen können optional mit Hintergrundbildern, z.B. Grundrissen eines Hauses, versehen werden.

Es kann auch ein PC als Zentrale benutzt werden. In dem Fall läuft die Steuerungssoftware *ExecEngine* auf dem PC und es werden an den PC angeschlossene oder über Netzwerk eingebundene Gateways als Schnittstelle zu den [Sensoren](#) und [Aktoren](#) benutzt. Der Nachteil in dieser Betriebsart ist, dass der PC ständig laufen muss. Aber auch wenn - wie im Normalfall üblich - eine Zentrale ( wie CCU oder c-comatic ) benutzt wird, bietet diese PC-Option einen Vorteil: Man kann optional die Hardwarekommunikation deaktivieren und somit Veränderungen in der Konfiguration und Programmierung seiner Anwendung testen ohne den Echtbetrieb zu stören. Die Einstellung der Option zur Ausführung der Steuerungssoftware auf dem PC kann unter dem Menüpunkt *Konfigurieren->Hardware* eingestellt werden. Lesen Sie dazu bitte unbedingt den Abschnitt [Der PC als HomeMatic-Zentrale](#) und gehen genau nach der dort beschriebenen Anleitung vor.

Aktuelle Informationen zur Software von CL-control für das HomeMatic-System finden Sie im Internet unter:

<http://www.cl-control.de/software.html>

Zum Anlernen und zur Konfiguration von Modulen und auch zur direkten Verknüpfung von Modulen wird die WEB-UI der CCU bzw. bei anderen Steuerungszentralen wie der CL-Box oder bei Benutzung des PCs als Zentrale die mitgelieferte WEB-UI für den PC benutzt.

Falls eine CCU benutzt wird, können die implementierten Programmierungsoptionen der CCU unabhängig von der homeputer-Software weiter benutzt werden. Dabei sollten Sie aber beachten, dass es dann schwierig werden kann den Überblick über die programmierten Funktionen zu behalten.

Falls eine HomeMatic-CCU als Zentrale benutzt und über eine USB-Verbindung an den PC angeschlossen wird, ist standardmässig eine IP-Adresse für die CCU vorgegeben. Zum Zeitpunkt der Erstellung dieser Beschreibung ist das die IP-Adresse 10.101.81.51.

Sollte diese Adresse nicht mehr aktuell sein können Sie die korrekte Adresse im entsprechenden Kapitel über die USB-Verbindung in der Bedienungsanleitung der CCU finden.

In den Teilen, in denen die verschiedenen Software-Versionen identisch sind, sind auch diese Hilfetexte identisch. An einigen Stellen der Hilfetexte kann es vorkommen, dass Funktionen beschrieben sind, die in der verwendeten Software-Version nicht zur Verfügung stehen. An diesen Stellen befindet sich dann ein entsprechender Hinweis.

Sollte dieser Hinweis im Einzelfall fehlen oder missverständlich sein, bitten wir um einen entsprechenden Hinweis.

Die homeputer-Software besteht aus mehreren Programmen:

### 1. Das **homeputer CL-Konfigurationsprogramm**

Mit diesem Programm werden die Anwendungen auf einem PC mit Betriebssystem Windows erstellt. Diese Konfigurationen und Programmierungen umfassen Zeittabellen zur Anwesenheitssimulation und zum Schalten einzelner Objekte, Makros zur Ablaufsteuerung und logischen Verknüpfung von Objekten und Funktionen und Definitionen zur Visualisierung der Anwendung.

Wenn die Anwendung mit allen Funktionen definiert und programmiert ist, wird mit einem Compiler (der aus dem Konfigurationsprogramm aufgerufen wird) eine Datei mit optimiertem binärem Ausführungscode erstellt. Dieser Ausführungscode wird auf die HomeMatic-Zentrale übertragen und wird dort selbständig, ohne dass ein PC eingeschaltet sein muss von der

### 2. **Exec-Engine** verarbeitet.

Die Exec-Engine ist ein Programm, dass auf der Hardware der Zentrale (im PC-Modus ohne externe Zentrale auf dem PC) läuft und aufgrund der erstellten Makros, Zeittabellen und Abhängigkeiten die logische Steuerung übernimmt. Die Exec-Engine kann erst nach Übertragung eines Projekts aktiv werden, solange noch kein Projekt übertragen wurde ist der Status "nicht aktiv". Die Exec-Engine stellt auch die Daten für Visualisierungen zur Verfügung. Die Visualisierungen werden vom Programm

### 3. **VisuWin** angezeigt.

Das Programm homeputer CL VisuWin kann auf einem oder mehreren PCs im Netzwerk installiert werden und erlaubt die Visualisierung, Kontrolle und Steuerung der Anwendung am PC-Bildschirm. Dabei können bis zu drei PC's gleichzeitig benutzt werden (in erweiterten Versionen der Studio-Software auch mehr, auf Anfrage). Alle definierten Ansichten können in unterschiedlichen eigenen Fenstern auf dem PC-Bildschirm angezeigt werden, wobei es auch möglich ist per Mausklick Zustände zu verändern und so Steuerungen direkt am PC-Bildschirm vorzunehmen.

### 4. **ExecEngineWin** (nur bei Benutzung externer Zentralen, wird im Modus "PC als Zentrale" nicht benutzt)

ist ein Programm-Modul, das mit der auf der Zentrale laufenden Exec-Engine kommuniziert, den Status der Exec-Engine anzeigt, die Ausführung starten und anhalten kann und für die Übertragung des erstellten Ausführungscode vom PC auf die HomeMatic-Zentrale zuständig ist. Dies passiert z.B. nach einer Änderung des Ausführungscode mit einem Knopfdruck.

Das Funktionsprinzip ist grundsätzlich einfach:

Der Ausführungsteil der Software (die Exec-Engine) empfängt Meldungen von den Sensoren (=Sendern des Systems). Für diese kann ein Makro (=eine Folge von Anweisungen) erstellt werden. Dieses Makro wird ausgeführt, sobald eine Meldung vom zugehörigen Sensor empfangen wird. In dem Makro können die Aktoren (Empfänger) geschaltet werden, aber auch andere Aktionen ausgeführt werden. So ist es z.B. möglich Windows-Programme zu starten oder als Hinweis für bestimmte Situationen Wave-Dateien auf einem Visualisierungs-PC abzuspielen.

Weiterhin können Schaltungen über Zeittabellen definiert werden, in der Studio-Version können Makros zudem in bestimmten Zeitintervallen (von Sekundenbruchteilen bis stündlich und täglich) aktiviert werden. Wenn Zustände von Objekten durch Makros verändert werden, so wird der neue Zustand nicht bei Ausführung der Makro-Anweisung, sondern erst nach Beendigung des Makros oder bei Ausführung einer WARTE-Anweisung in dem Makro an den Empfänger gesendet. Die umfangreichen Möglichkeiten der Software erlauben die Realisierung sehr grosser und komplexer Anwendungen.

## 1.4. Erste Schritte

---

### Erste Schritte

Wir bedanken uns für die Benutzung unserer Software.  
Im Folgenden finden Sie die Anleitung zur Installation und Inbetriebnahme.

#### **Vorab einige Hinweise und Erklärungen zur CL-Software:**

Die CL-Software bietet umfangreiche Möglichkeiten eigene individuelle Anwendungsfunktionen für ein intelligentes Smart-Home zu erstellen. Grundlage ist dabei eine einfach anzuwendende, leistungsfähige Programmiersprache. Diese Form der Programmierung ist nicht nur einfacher, sondern bietet erheblich mehr Möglichkeiten und Features, als die Programmierung über die integrierte WEB-UI der CCU.

Die CL-Software bietet folgende leistungsfähige Features:

- virtuelle Objekte beliebigen Typs, unabhängig von angelerten Geräten, z.B. zur Ausgabe von Texten in Visualisierungsansichten
- Rechenfunktionen mit Zahlen, Zeiten
- Funktionen zur Bearbeitung von Texten
- Variablen zur Zwischenspeicherung und Berechnung von Werten
- Möglichkeit e-mails zu versenden
- integrierte Anwesenheitssimulation
- Funktionen zur Bearbeitung von Texten
- Schreiben eigener Textdateien z.B für Logs
- Historydatei mit separater Auswertungssoftware (History CL)
- umfangreiche Visualisierungen über
  - Windowsprogramm,
  - CL-Webserver für beliebige Browser
  - optionale Apps für iOS und Android erhältlich in den jeweiligen App-Stores
- Aufruf von Webseiten (http oder https-Requests) um Inhalte von Internetseiten auszulesen oder Geräte anzusprechen
- integrierte Funktion zur Heizungssteuerung mit Schaltaktoren z.B. für Fussbodenheizungen, Infrarotheizungen und Heizlüfter
- Schnittstellen zu Philips Hue-Bridge und Osram-Lightify-Gateway

In den Hilfetexten finden Sie ausführliche Anleitungen und Beschreibungen zur Programmierung.

Die Ausführung bei Benutzung der CL-Software ist erheblich schneller als Programmierungen über die WEB-UI, da die Programmierung in einen optimierten Ausführungscode übersetzt wird, der von der Zusatzsoftware auf der CCU dann sehr schnell ausgeführt werden kann.

Ein weiterer Vorteil der CL-Software ist, dass eine Erweiterung auf mehrere CCUs möglich ist wenn die Kapazitätsgrenze einer CCU bezüglich der Anzahl der angelerten Geräte erreicht ist. Neue Geräte werden dann einfach an einer weiteren CCU angelernt, die nur als Schnittstelle fungiert, und die Software wird upgegradet auf eine CLX-Version. Wenn bei mehreren CCUs auch die Hardware-Kapazität der als Zentrale benutzten CCU überschritten wird, kann eine c-comatic-Zentrale eingesetzt werden. Diese verfügt über erheblich mehr Ressourcen und Leistung als eine CCU.

#### **Nun aber zur aktuellen Installation:**

Das PC-Programm können Sie von folgender Seite herunterladen:

<http://www.cl-control.de/software/downloads.html>

Bitte speichern Sie die Downloaddatei in einem beliebigen Verzeichnis (normalerweise wird sie automatisch im Download-Verzeichnis gespeichert)  
und starten das Installationsprogramm durch Doppelklick auf die Downloaddatei.  
Folgend Sie dann den Anweisungen des Installationsprogramms, Wir empfehlen als

Datenverzeichnis nicht den langen von Windows vorgeschlagenen Pfad zu benutzen, sondern ein eigenes Verzeichnis auszuwählen (z.B. C:\Daten\contronics\CL-Studio).

Als Zentrale könne eine CCU von eQ-3 oder eine alternative Zentrale wie z.B. RaspberryMatic benutzt werden, die Zentrale wird im Folgenden immer als CCU bezeichnet.

Nach der Installation des PC-Programms muss noch eine Zusatzsoftware auf der CCU installiert werden.

Bitte downloaden Sie die Zusatzsoftware für die CCU von der normalen oben schon angegebenen Downloadseite:

<http://www.cl-control.de/software/downloads.html>

Bitte beachten Sie, dass für die unterschiedlichen Versionen der CCUs bzw. alternativer Zentralen unterschiedliche Versionen der Zusatzsoftware verwendet werden müssen.

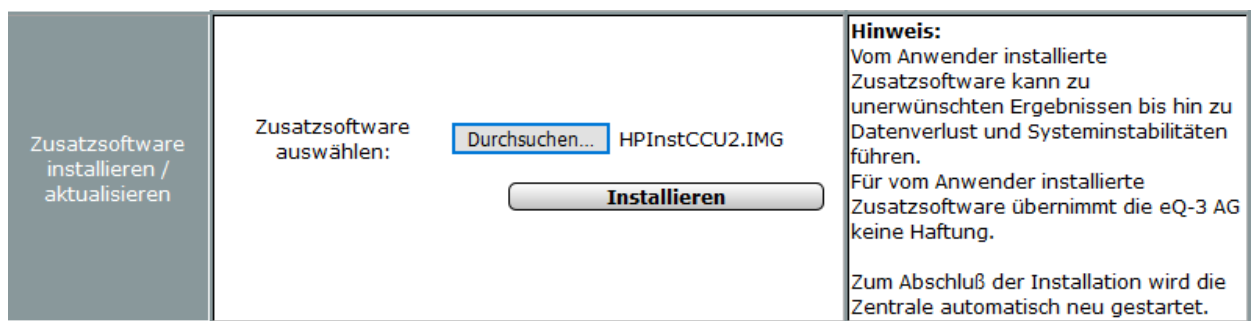
Für die CCU2 wird die Installationsdatei *HPInstCCU2.IMG* benutzt.

Für die CCU3 und RaspberryMatic wird die Installationsdatei *CLInstCCU3RM.IMG* benutzt.

Installieren Sie die Zusatzsoftware auf der Zentrale, indem Sie in der WEB-UI folgenden Menüpunkt aufrufen:

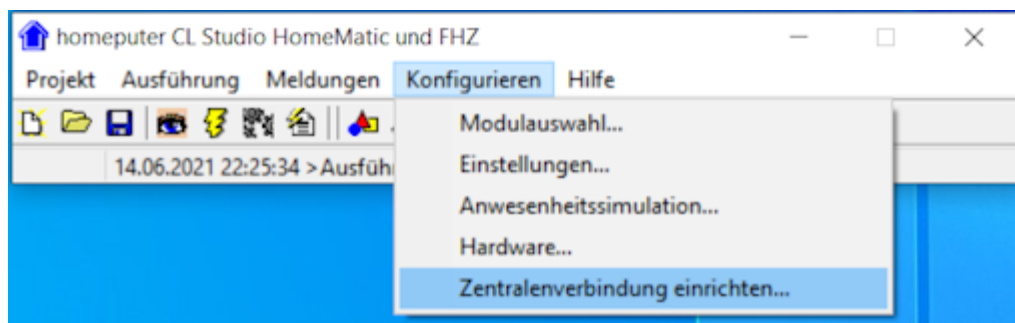
*Einstellungen->Systemsteuerung->Button Zusatzsoftware*

Wählen Sie die heruntergeladene Zusatzsoftware mit dem Button **Durchsuchen** aus und klicken dann auf **Installieren**.



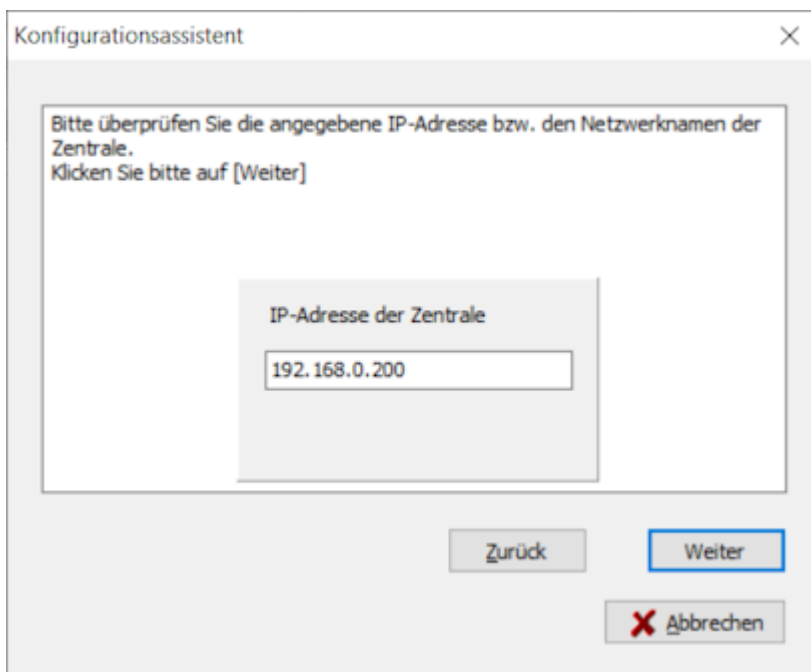
Die CCU wird bei der Installation neu gestartet, was einige Minuten dauern kann.

Nachdem die Installation der Zusatzsoftware auf der CCU abgeschlossen ist rufen Sie im PC-Programm den Menüpunkt **Zentralenverbindung einrichten** auf.

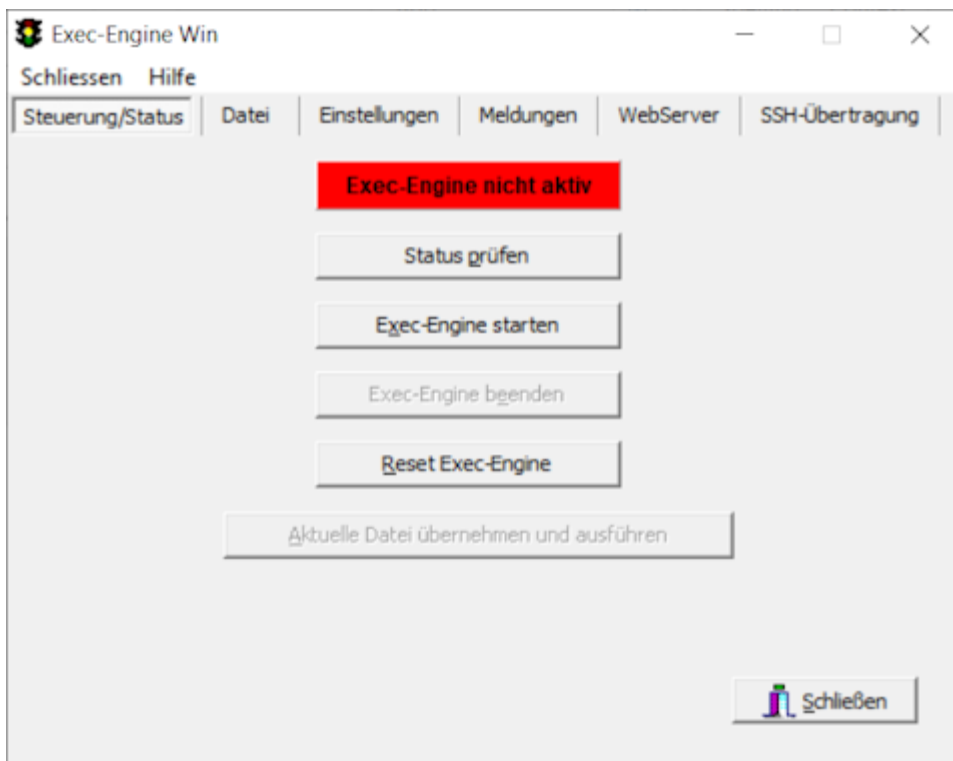


Nach einem Hinweisenfenster erscheint das Fenster zur Eingabe der IP-Adresse der CCU.





Als nächstes stellt der Verbindungsassistent eine Verbindung zur CCU her und zeigt dann den Status der Exec-Engine an.



Der Status der Exec-Engine wird bei der ersten Konfiguration als nicht aktiv angezeigt.  
**Das ist richtig so, die Exec-Engine kann erst aktiv werden, nachdem ein Projekt auf die CCU übertragen wurde!**

Damit ist die Verbindungskonfiguration zur CCU erfolgreich beendet.  
 Unter diesem Link können Sie sich auch ein Video zur Verbindungskonfiguration ansehen:  
<http://www.cl-control.de/videos/VerbindungskonfigurationCCU.mp4>

Im Folgenden ist eine erste kleine Anwendung beschrieben.

Bitte beachten Sie bei der Verwendung von externen Zentralen (CCU, c-comatic):  
 Das Anlernen der Geräte an externe Zentralen wird ebenso wie die Einstellungen von Grundparametern an den Geräten mit der WEB-UI der jeweiligen Zentrale vorgenommen.  
 Informationen zur WEB-UI und zum Anlernen und Konfigurieren der Geräte finden Sie in der

jeweiligen Bedienungsanleitung der Zentrale.

Nach dem Anlernen werden die angelernten Module in die CL-Software importiert. Nach der Erstellung der Anwendungsfunktionen und der optionalen Erstellung individueller Visualisierungsansichten für die PC-Visualisierung VisuWin mit der CL-Software wird eine Ausführungsdatei mit den erstellten Konfigurationen und Programmierungen auf die externe Zentrale übertragen. Der PC mit der CL-Software kann dann zur Visualisierung benutzt werden, ist aber für die Automationsanwendung nicht mehr erforderlich und kann ausgeschaltet werden.

Wenn der PC als Zentrale verwendet werden soll lesen Sie unbedingt das Kapitel [Benutzung des PCs als Zentrale](#).

## Prinzipielles

Jeder Sensor (also z.B. jede Taste einer Fernbedienung) und jeder Aktor (z.B. eine Schaltsteckdose oder die einzelnen Ausgänge eines 4-fach-Aktormoduls) werden in der CL-Software als „Objekt“ dargestellt.

Es können auch eigene Objekte ohne zugehöriges Hardware-Modul erstellt werden, diese werden dann als virtuelle Objekte bezeichnet.

Jedes Objekt hat ein eigenes Makro.

Makros von Sensor-Objekten werden normalerweise ausgeführt, wenn von der Sensor-Hardware eine Meldung empfangen wird. Dazu ist bei Sensoren standardmässig die Option "Ausführen bei Empfang" aktiviert.

Makros von Aktoren können in verschiedenen Ausführungsintervallen aktiviert werden, das gewählte Ausführungsintervall sollte dabei nicht kleiner als nötig sein. Ausführungsintervalle unter 5 Sekunden sollten nur in speziellen Ausnahmefällen verwendet werden.

Die Visualisierungen können optional mit Hintergrundbildern, z.B. Grundrissen eines Hauses, versehen werden.

Im Folgenden ist beschrieben wie Sie sich um sich mit den Grundlagen vertraut machen können.

## Eine einfache Visualisierung

ohne logische Funktionen und eigene Ansichten ist sehr einfach machbar. Als erstes muss eine Verbindung zur Zentrale konfiguriert werden. Die Einrichtung der Verbindung kann wie oben beschrieben mit dem Menüpunkt *Zentralenverbindung einrichten* im Menü *Konfigurieren* durchgeführt werden. Falls diese automatische Verbindungseinrichtung Fehlermeldungen anzeigt und nicht funktioniert nehmen Sie bitte die einzelnen Schritte der Einrichtung manuell vor. Diese Einstellungen sind abhängig von der verwendeten Hardware und in den Kapiteln hinter *Software-Installation* im Inhaltsverzeichnis der Hilfetexte beschrieben.

- Lernen Sie einige Module an der Zentrale an. Übernehmen Sie die Module mit dem Knopf "Module importieren" im Fenster Modulauswahl.

- Klicken Sie auf den "Blitz"-Button oder wählen im Menü *Ausführung* den Punkt *Compilieren und übertragen*.

- Starten Sie das Programm *VisuWin* (Ausführung->Visualisierung oder Klick auf den "Auge"-Button) und wählen die Ansicht "Alle Objekte".

Nun sehen Sie alle Objekte in einer Ansicht und können diese mit der Maustaste bedienen.

Wenn das alles funktioniert hat, sind alle Grundeinstellungen bezüglich der Kommunikation mit der Zentrale in Ordnung und wir können uns grösseren Aufgaben zuwenden. Wenn keine Ansichten und Objekte angezeigt werden kontrollieren Sie bitte die Einstellungen in VisuWin unter dem Menüpunkt Konfiguration->Verbindung. Wenn das Objekt Uhr nicht alle 5 Sekunden aktualisiert wird, wurde wahrscheinlich eine falsche IP-Adresse unter "eigene IP-Adresse" angegeben, dann weiss die ExecEngine nicht, an welche Adresse Aktualisierungen geschickt werden sollen.

.

- Wenn in VisuWin keine Objekte angezeigt werden können Sie mit dem Programm [ExecEngineWin](#) kontrollieren ob die ExecEngine auf der Zentrale aktiv ist. ExecEngineWin wird wie folgt gestartet: Menü Konfigurieren -> Hardware-Schnittstelle dann auf den Knopf "Kontrollprogramm für Zentrale" klicken.

Bitte beachten Sie, dass die ExecEngine auf der Zentrale nur aktiv sein kann nachdem ein Projekt zur Zentrale übertragen wurde, vorher steht der Status auf "nicht aktiv".

Auf der Seite muss nach der Übertragung eines Projekts der Status Aktiv angezeigt werden (Knopf oben links ist grün).

Das wichtigste Werkzeug zur Erstellung leistungsfähiger Anwendungen sind die Makros.

## Grundlagen der Makro-Programmierung

Mit den Anweisungen in den [Makros](#) werden die Aktionen, logischen Verknüpfungen, also alle wichtigen Funktionen einer Anwendung programmiert.

Makros werden aktiviert durch Empfang einer Meldung von einem Sensor (das kann eine z.B. Fernbedienung oder ein Temperatursensor sein).

Um Aktionen also z.B. bei Betätigung einer Fernbedienungstaste auszulösen, werden entsprechende Anweisungen im Makro der Fernbedienungstaste programmiert.

In der Studio-Version können Makros zusätzlich in einem individuell einstellbarem Zeitintervall ausgeführt werden. Die Aktivierung von Makros im Zeitintervall wird normalerweise bei Aktoren verwendet, z.B. um eine Lampe nach einer bestimmten Zeit auszuschalten.

Bitte lesen Sie auch die Seiten [Allgemeine Hinweise](#) und [Die wichtigsten Begriffe](#)

Um in die Praxis einzusteigen sollten zuerst mit kleinen Testprojekten etwas üben.

### Ein kleines Beispielprojekt Teil 1

In diesem Teil des kleinen Beispielprojekts erstellen wir ein Makro für eine Fernbedienungstaste zum Schalten einer Lampe.

Falls Sie schon Module in der Zentrale angelernt haben, können Sie diese im Fenster Modulauswahl mit dem Knopf "Module aus Zentrale importieren" übernehmen und die vorhandenen Module benutzen, soweit diese benutzt werden können (also eine Fernbedienungstaste und ein Schaltaktor wie z.B. eine Schaltsteckdose).

Als erstes schreiben wir ein kleines Makro für eine Fernbedienungstaste, mit der eine Lampe geschaltet wird. Falls in der Zentrale noch keine Fernbedienung angelernt und übernommen worden ist, wählen Sie einfach im Fenster Modulauswahl eine Fernbedienung aus und bringen diese durch Anklicken des Knopfes mit dem roten Pfeil in das rechte Fenster. Nun wird noch eine Lampe benötigt. Wenn es noch von der Zentrale übernommenes Modul zum Schalten einer Lampe gibt (z.B. eine Schaltsteckdose) wählen Sie in der Modulauswahl eine Schaltsteckdose aus und bringen diese durch Anklicken des Knopfes mit dem roten Pfeil in das rechte Fenster. Gehen Sie in die Modulbearbeitung und nennen das Objekt "Lampe1". Wenn es schon ein geeignetes Modul gibt benennen Sie dieses um auf "Lampe1".

Wenn Sie eine Fernbedienung an der Zentrale angelernt haben beachten Sie bitte, dass die einzelnen Tasten vor der Verwendung "registriert werden müssen. Infos dazu finden Sie auf der Seite [Schaltsensoren](#) am Ende der Seite.

Wählen Sie dann die neu angelegte Fernbedienung (oder eine beliebige schon in der Zentrale vorhandene Fernbedienung) aus und klicken auf den Knopf "Bearbeiten".

Wählen Sie in dem Fenster der Fernbedienung eine Taste aus und klicken dann auf "Makro". In dem Fenster zur Makrobearbeitung erstellen Sie dann folgende Anweisung:

Lampe1 einschalten für 10 Sekunden

Diese Anweisung bewirkt, dass das Objekt "Lampe1" für 10 Sekunden eingeschaltet wird. Um das Makro direkt in der Visualisierung ohne Fernbedienung testen zu können, aktivieren Sie auf der Seite Allgemein des Objektfensters die Option "Ausführen bei Eingabe". Wenn diese Option aktiviert ist, wird das Makro ausgeführt, wenn das Icon mit der Maus angeklickt wird. (Diese Funktion ist normalerweise standardmässig nicht aktiviert, da ein Anklicken des Icons in der Visualisierung auch eine Funk-Sendung verursachen kann, je nach Programmierung und Konstellation könnte das dann zu unbeabsichtigten Reaktionen führen.)

Klicken Sie auf den "Blitz"-Button oder wählen im Menü *Ausführung* den Punkt *Compilieren und übertragen*.

Falls es bei der Übertragung Probleme gibt überprüfen Sie bitte die Einstellungen und Eintragungen zur [Hardwareverbindung](#).

Starten Sie dann das Programm VisuWin (Ausführung->Visualisierung oder Klick auf den Auge-Knopf) und wählen die Ansicht "Alle Objekte".

Klicken Sie auf das Icon der Fernbedienungstaste oder auf die Fernbedienungstaste und prüfen ob alles funktioniert.

Nun ändern wir das Makro der Fernbedienung etwas ab, wir löschen die vorhandene Anweisung und erstellen diese Anweisung:

```
Lampel umschalten
```

Diese Anweisung bewirkt, dass das Objekt "Lampe1" umgeschaltet wird (d.h. wenn es ausgeschaltet war wird es eingeschaltet, wenn es eingeschaltet war wird es ausgeschaltet). Mit dieser Anweisung ist es möglich eine Lampe mit beliebig vielen Tastern mit jeweils nur einer Tastenbetätigung zu schalten (in der konventionellen Elektroinstallation wird diese Funktion "Wechselschaltung" genannt).

Klicken Sie auf den "Blitz"-Button oder wählen im Menü *Ausführung* den Punkt *Compilieren und übertragen*.

## Ein kleines Beispielprojekt Teil 2

Makros können auch für Aktoren erstellt werden. Wir erweitern nun das Beispielprojekt um eine zusätzliche Funktion, indem wir auch für die Lampe ein Makro erstellen. Die Lampe soll nach 5 Minuten ausgeschaltet werden, unabhängig davon wie sie eingeschaltet wurde. Dazu wählen wir das Modul "Lampe1" im Fenster Modulauswahl aus und klicken auf "Bearbeiten" und im Fenster der Lampe auf "Makro".

Im Makro erstellen Sie nun folgende Anweisungen:

```
wenn Lampel eingeschaltet und Schaltdauer(Lampel) > "00:00:20" dann  
    Lampel ausschalten  
endewenn
```

Dieses Makro soll bewirken, dass die Lampe1 immer 20 Sekunden nachdem sie eingeschaltet wurde wieder ausgeschaltet wird.

Um die Lampe auszuschalten muss das Makro ausgeführt werden, in dem Fall sollte für die Ausführung ein Zeitintervall von 5 Sekunden eingestellt werden. Das Ausführungsintervall sollte immer unter Berücksichtigung der im Makro verwendeten Zeitabfragen gewählt werden. Im Zweifelsfall ist 5 Sekunden eine gute Einstellung, kleinere Ausführungsintervalle sollten vermieden werden. Wenn die im Makro verwendeten Zeitabfragen im Minuten- oder Stundenbereich ist es empfehlenswert entsprechend grössere Zeitintervall auszuwählen.

### Bitte beachten Sie dabei:

Uhrzeitabfragen müssen immer durch 5 Sekunden teilbar sein. Wenn ein Makro Uhrzeitabfragen beinhaltet, ist es wichtig darauf zu achten, dass das Zeitintervall entsprechend eingestellt wird. Wenn in einem Makro Sonnenaufgang oder Sonnenuntergang abgefragt werden, muss dieses mindestens jede Minute ausgeführt werden, weil Sonnenauf- und -untergang immer volle Minuten haben.

Übertragen Sie das Projekt mit dem "Blitz"-Button auf die Zentrale und spielen ein wenig damit.

Erweitern Sie dieses erste kleine Projekt und spielen ein wenig mit den Möglichkeiten. Sehen Sie sich die Anweisungen und Funktionen der Makrosprache an und lassen Sie Ihrer Phantasie freien Lauf bezüglich der Realisierung Ihrer individuellen Projekte.

Nochmal eine Checkliste zu Projekterstellung:

### Schritte zur Erstellung von Projekten

1. Lernen Sie die HomeMatic-Module in der CCU an.
2. Übernehmen Sie die HomeMatic Module aus der CCU durch Anklicken des Knopfes "Module aus Zentrale importieren" bzw. legen bei anderen Modulen Gateways und Module an
3. Erstellen Sie Ihre individuellen Ansichten auf der Seite *Ansichten* des Einstellungsfensters wenn Sie die PC-Visualisierung benutzen wollen. Dabei können Sie auch individuelle Hintergrundbilder (z.B. Grundrisse) festlegen. Dieser Schritt ist optional.

4. Erstellen Sie die Makros für die einzelnen Objekte.

5. Erstellen und übertragen Sie die Ausführungsdatei auf die Zentrale. Klicken Sie dazu auf den "Blitz"-Button oder wählen im Menü *Ausführung* den Punkt *Compilieren und übertragen*

6. Starten Sie das Programm VisuWin und prüfen Sie ob in den Ansichten alles so dargestellt wird, wie Sie es wollten. Es gibt standardmässig immer zusätzlich zu evtl. selbst erstellten Ansichten auch eine Ansicht "Alle Objekte", in der alle benutzten Objekte angezeigt werden - wenn Sie diese Ansicht in der Konfiguration des Visualisierungsprogramms nicht deaktiviert haben.

Wenn Sie Objekte mit der Maus anklicken können Sie die Zustände ändern bzw. neue Werte angeben und die Geräte so steuern.

Zur Visualisierung können auch die Apps pocket control CL (iOS) der Firma Penzler für iPhone/iPad oder mobileControl CL für Android-Geräte verwendet werden. Zur Visualisierung mit jedem beliebigem Browser können Sie den CL-Webserver benutzen. Der CL-Webserver ist Bestandteil der Software und kann kostenfrei im Downloadbereich der CL-Software geladen werden.

# 1.5. Software-Installation

---

## Software-Installation

### Systemvoraussetzungen des PC

- Standard-Windows-PC
- Grafikkarte mit einer Auflösung von mindestens 800x600 Pixel
- Betriebssystem Windows XP oder höher
- Maus oder ein kompatibles Zeigegerät

### Installation der Software auf dem PC:

Das Installationsprogramm kopiert alle Dateien und richtet die Anwendung auf Ihrem Computer ein. Sie müssen die Installation nur einmal durchführen.

Das PC-Programm ist ein Windows-Programm. Sie benötigen das Betriebssystem Windows 2000, XP oder höher auf Ihrem Rechner.

Eventuell laufende Programme sollten Sie beenden, da sie den Installationsvorgang behindern können.

Downloaden Sie die Datei von der Downloadseite:

<http://www.cl-control.de/software/downloads.html>

- Starten Sie das heruntergeladene Installationsprogramm durch einen Doppelklick auf die Datei.  
Falls eine Warnung kommt, da die Datei aus dem Internet geladen wurde lassen Sie sich weitere Informationen anzeigen und wählen dann "trotzdem ausführen".  
(je nach Windows-Version)
- Folgen Sie den Hinweisen des Installationsprogramms.
- Nach Abschluss der Installation starten Sie das Programm bitte und beenden es sofort wieder. Dann starten Sie es erneut und rufen den Menüpunkt Projekt->Update auf. Prüfen Sie bitte im Updatefenster ob eine neuere Version zur Verfügung steht und machen einen Update wenn das der Fall ist. Da dauernd neue Module zum System erscheinen und für diese oftmals ein Softwareupdate bzw. ein Update der Modultabelle erforderlich ist, ist auf der CD oftmals nicht die aktuellste Version.

**Die folgenden Schritte sind nur erforderlich falls Sie eine CCU als Zentrale benutzen, für andere Zentralen von CL-control überspringen Sie diese Schritte bitte.**

### Installation der Software auf der HomeMatic-Zentrale (CCU):

Auf der HomeMatic-Zentrale muss auch eine entsprechenden Software installiert werden.

Die aktuelle Version der CCU-Zusatzsoftware können Sie von dieser Seite downloaden:

<http://www.cl-control.de/software/downloads.html>

**CLInstCCU3RM.IMG** für die CCU3 und RaspberryMatic

**HPInstCCU2.IMG** für die CCU2

**HPInst.IMG** für die CCU1

Speichern Sie die Downloaddatei in einem beliebigen Verzeichniss des PCs.

Starten Sie die WEB-Oberfläche (WEB-UI) der Zentrale durch Eingabe der IP-Adresse der Zentrale in einem Internet-Browser.

Wählen Sie den Menüpunkt **Einstellungen** und dort den Unterpunkt **Systemsteuerung**.

Klicken Sie auf die Box **Zusatzsoftware**

Es öffnet sich ein Fenster in dem der Speicherort für die Software angegeben werden muss. Geben Sie das Verzeichnis an, in das Sie die Downloaddatei mit der Zusatzsoftware abgespeichert haben.

Klicken Sie dann auf den Knopf **[Installieren]**.

Es erscheint ein neues Fenster, klicken Sie in diesem Fenster auf den Knopf [Installation starten]

Die Software für die Zentrale wird nun automatisch installiert, nach der Installation wird die Zentrale automatisch neu gestartet.

Rufen Sie den Punkt Zusatzsoftware erneut auf und prüfen ob eine aktuellere Version zum Download zur Verfügung steht, Falls das der Fall ist machen Sie bitte einen Update. Da dauernd neue Module zum System erscheinen und für diese oftmals ein Softwareupdate erforderlich ist, ist auf der CD oftmals nicht die aktuellste Version der Software.

#### **Bitte unbedingt beachten:**

Bei einer CCU3/RaspberryMatic müssen noch die Standard-Sicherheitseinstellungen angepasst werden, damit auf diese zugegriffen werden kann.

Hinweise dazu finden Sie hier:

<http://www.cl-control.de/downloads/pdf/SicherheitseinstellungenCCU3.pdf>

## **Freigabe der Software zur zeitlich unbeschränkten Nutzung**

Die Software muss über die PLN (Persönlichen Lizenz-Nummer) aktiviert werden, um ohne zeitliche Einschränkung benutzt werden zu können.

Je nach Version wird bei ersten Starten des PC-Programms nach der PLN oder nach einem Freigabecode gefragt.

Sie können die Freigabe auch direkt machen, bevor diese nach Ablauf einer eventuellen Testzeit (je nach Programmversion) erforderlich wird.

Das Freigabefenster wird geöffnet durch den Menüpunkt Projekt->Freigabe im Hauptmenü. Nach erfolgter Freigabe wird dieser Menüpunkt je nach Programmversion und -status nicht mehr angezeigt.

Die Freigabe ist nur einmalig erforderlich und wird gespeichert.

Wenn Sie mit einer Zentrale arbeiten wird die Freigabe dort gespeichert und ist nicht PC-abhängig.

**Bitte bewahren Sie Ihre PLN unbedingt gut auf, bei Verlust kann kein Ersatz geliefert werden! .**

Sie brauchen die PLN für erneute Freigaben bei einem eventuellen Hardwarewechsel.

## **Genaue Beschreibung des Freigabevorgangs**

### **Vorbereitung der Freigabe bei einer separaten Zentrale:**

Stellen Sie auf der Zentrale eine feste IP-Adresse ein und geben diese in der Hardwarekonfiguration an.

Das Fenster dazu öffnen Sie mit dem Menüpunkt Konfigurieren->Hardware.

Im Reiter Zentrale geben Sie dann die IP-Adresse der Zentrale an.

Öffnen Sie nun das Freigabefenster mit dem Menüpunkt Projekt->Freigabe auf.

Dort wird oben auf der Seite die Seriennummer der Zentrale angezeigt.

Die Seriennummer wird nur angezeigt wenn die Verbindung zur Zentrale eingerichtet ist und aufgebaut werden kann.

Nur dann kann auch der Freigabevorgang ausgeführt werden.

## Vorbereitung der Freigabe bei Benutzung des PCs als Zentrale:

### Bei Benutzung von HomeMatic-LAN- oder USB-Interface:

Im PC-Modus muss der BidCos-Service vor der Freigabe korrekt eingerichtet sein.  
Das Fenster dazu öffnen Sie mit dem Menüpunkt Konfigurieren->Hardware.  
Im Reiter BidCos klicken Sie dann auf den Button BidCos-Interfaces konfigurieren.

### Bei Benutzung eines FHZ-Interface:

Öffnen Sie das Fenster Hardware mit dem Menüpunkt Konfigurieren->Hardware.  
Legen Sie im Reiter FHZ die benutzten FHZ-Interfaces an.  
Lesen Sie dazu bitte auch das Kapitel Einstellungen FHZ-Schnittstellen

Öffnen Sie nun das Freigabefenster mit dem Menüpunkt Projekt->Freigabe auf.  
Dort wird oben im Fenster die Seriennummer des ersten konfigurierten Interfaces angezeigt.  
Wenn keine Seriennummer angezeigt wird konnte keine Verbindung zu einer Zentrale aufgebaut werden, schiessen Sie in dem Fall das Freigabefenster und rufen den Menüpunkt *Konfigurieren->Zentralenverbindung einrichten* auf.  
Danach öffnen Sie wieder das Freigabefenster.

## **Freigabe aktivieren**

Tragen Sie im Eingabefeld *PLN* Ihre PLN ein und klicken auf den Button **[automatische Internetfreigabe]**.

Wenn der Rechner keine Verbindung zum Internet hat, muss die Freigabe manuell durchgeführt werden.

Rufen Sie dazu in einem Browser die folgende Seite auf und tragen PLN und Seriennummer ein:  
<http://www.contronics-software.de/cgi-bin/Freigabe.php>

Die PLN tragen Sie dann im Feld PLN des Freigabefensters ein, den errechneten Freigabecode im Feld Freigabecode des Freigabefensters ein und klicken auf **[Freigabe ausführen]**.

Sie erhalten nun eine Meldung über die erfolgreiche Freigabe.

Wenn eine Fehlermeldung erscheint überprüfen Sie bitte die in der Fehlermeldung angegebenen Einstellungen.

Eine ausführliche Beschreibung zum Freigabevorgang finden Sie unter diesem Link:  
[www.cl-control.de/downloads/pdf/FreigabeCLStudio.pdf](http://www.cl-control.de/downloads/pdf/FreigabeCLStudio.pdf)



## 1.6. Einstellungen HomeMatic-CCU

---

### Einstellungen HomeMatic-CCU

Nach der Installation der Software müssen einige Einstellungen vorgenommen werden. Diese betreffen hauptsächlich die Kommunikation mit der HomeMatic-Zentrale.

Auf der HomeMatic-CCU muss die ExecEngine installiert werden, bevor eine Verbindung zur Zentrale hergestellt werden kann. Wie das geschieht ist unter [Software-Installation](#) beschrieben. Ob die Software ExecEngine auf der Zentrale installiert ist, können Sie in der WEB-UI überprüfen mit dem Menüpunkt:

Systemsteuerung->Systemkonfiguration->Zusatzsoftware

In dem Fenster Zusatzsoftware muss "homeputer CL" angezeigt werden.

Wenn der PC-Modus benutzt werden soll lesen Sie bitte unbedingt den Abschnitt [Der PC als HomeMatic-Zentrale](#) und gehen genau nach der dort beschriebenen Anleitung vor.

Auf der Zentrale muss eine feste IP-Adresse eingestellt werden, bevor die Verbindung konfiguriert wird. Wie das gemacht wird ist in der Bedienungsanleitung der CCU beschrieben. Falls die Verbindung zur CCU nicht über Netzwerk, sondern über USB-Kabel hergestellt wird werden standardmässig spezielle IP-Adressen verwendet. Diese lauten normalerweise 10.101.82.51 für die CCU und 10.101.82.52 für den PC und müssen bei USB-Verbindung in der Konfiguration entsprechend angegeben werden. Diese IP-Adresse bei USB-Verbindung können sich je nach Firmware-Version eventuell ändern, entnehmen Sie diese dann bitte der Bedienungsanleitung der CCU.

#### **Bitte unbedingt beachten:**

Bei einer CCU3/RaspberryMatic müssen noch die Standard-Sicherheitseinstellungen angepasst werden, damit auf diese zugegriffen werden kann.

Hinweise dazu finden Sie hier:

<http://www.cl-control.de/downloads/pdf/SicherheitseinstellungenCCU3.pdf>

Im Konfigurationsprogramm muss im [Fenster Hardware](#) die IP-Adresse der Zentrale und der Port für die [Exec-Engine](#) eingegeben werden. Standardmässig wird Port 2110 benutzt, diesen sollten Sie nicht ohne Grund ändern. Weiterhin muss in diesem Fenster auf der Seite *eigene IP* die eigene IP-Adresse ausgewählt werden, falls der PC über mehr als eine Netzwerkkarte verfügt oder die Verbindung über das USB-Kabel hergestellt wird.

Nachdem das passiert ist, können Module aus der Zentrale importiert werden.

Bitte stellen Sie auf der Seite "Verzeichnisse" des Einstellungsfensters die Verzeichnisse für Projektdateien und Bitmaps ein. Im Normalfall sind diese Verzeichnisse für Projektdateien und Bitmaps bei der Installation schon eingerichtet und eingetragen worden.

Diese Einstellungen müssen vor der ersten Erstellung eines eigenen Projekts vorgenommen werden, damit dieses in dem Verzeichnis gespeichert werden kann und dort auch die Ausführungsdatei für die Zentrale abgelegt werden kann.

Bevor die Datei mit dem Ausführungscode eines erstellten Projektes zur Zentrale übertragen werden kann muss die Verbindung zur Zentrale konfiguriert werden und es muss eine Konfigurationsdatei zur Zentrale übertragen werden. Die Einrichtung der Verbindung kann mit dem Menüpunkt *Zentralenverbindung einrichten* im Menü *Konfigurieren* durchgeführt werden.

Das Erstellen und Übertragen des Ausführungscode zur Zentrale kann mit dem "Blitz"-Button oder dem Menüpunkt *Compilieren und übertragen* im Menü *Ausführung* erfolgen. Dabei werden alle notwendigen Einstellungen und Aktionen automatisch durchgeführt. Sollte nach einer Übertragung die Statusanzeige in *ExecEngineWin* nach einiger Zeit nicht die grüne Aktiv-Meldung bringen, kann es helfen die Zentrale einmal zu reseten. Bei der HomeMatic-CCU kann man das mit dem kleinen schwarzen Knopf über dem Netzwerkanschluss tun.

Je nach eventuell notwendigen Einstellungen bezüglich der Netzwerkverbindungen (Firewall, Portfreigabe im Router) kann es vorkommen, dass über den Menüpunkt *Zentralenverbindung einrichten* die Verbindung nicht hergestellt werden kann oder die automatische Übertragung des Ausführungscode nicht funktioniert.

Dann kann die Einrichtung der Verbindung und die Übertragung manuell vorgenommen werden.

Wie das funktioniert ist im Folgenden beschrieben:

Vor der Übertragung einer Ausführungsdatei an die Exec-Engine müssen im Programm [ExecEngineWin](#) die folgenden Schritte durchgeführt werden. Das Programm *ExecEngineWin* wird aufgerufen mit dem Button **[Kontrollprogramm für Zentrale]** im Fenster *Hardware*.

1. Auf der Seite *Einstellungen* müssen IP-Adresse der Zentrale und Port der Exec-Engine (2110) eingetragen werden. Danach muss die Konfigurationsdatei durch Mausklick auf den Knopf **[Konfigurationsdatei übertragen]** zur Exec-Engine übertragen werden.
2. Auf der Seite *Dateien* wird der Pfad für die Ausführungsdatei und die Ausführungsdatei eingetragen und durch Mausklick auf den Knopf **[Übertragen auf Zentrale]** zur Exec-Engine übertragen. Im Feld *Ausführungsdatei* darf nur der Dateiname stehen, ohne den Pfad zur Datei. Das ist erforderlich, weil die Datei im Dateisystem der Zentrale unabhängig vom Pfad der Datei auf dem PC abgespeichert wird.
3. Auf der Seite *Steuerung/Status* die Exec-Engine mit dem Knopf **[Exec-Engine starten]** starten bzw. mit dem Knopf **[Reset Exec-Engine]** neu starten wenn sie schon aktiv ist.

Nachdem diese Schritte erledigt sind, sind alle erforderlichen Daten auf der Zentrale gespeichert. Bei einem Neustart der Zentrale startet die Exec-Engine automatisch mit der aktuell eingestellten Ausführungsdatei.

Wenn die aktuelle Ausführungsdatei verändert wird, kann diese durch Anklicken des Knopfes **[Aktuelle Datei übernehmen und ausführen]** übertragen werden. Dabei wird automatisch die aktuelle Ausführung beendet, die eingestellte aktuelle Datei übertragen und die Ausführung wieder gestartet.

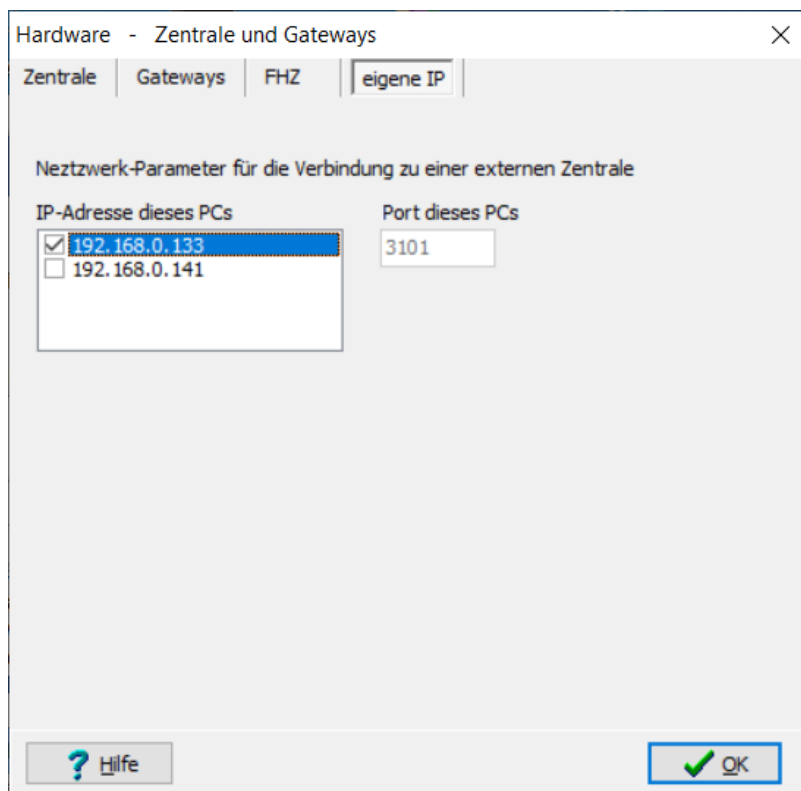
Wenn sich der Name der aktuellen Ausführungsdatei ändert ist etwas mehr zu tun (nur erforderlich wenn die Übertragung der Ausführungsdatei mit dem Blitz-Button des Hauptfensters nicht funktioniert! ):

1. Exec-Engine beenden mit dem entsprechenden Knopf auf der Seite *Steuerung/Status*,
2. Den neuen Dateinamen auf der Seite *Datei* *Ausführungsdatei* eintragen und mit dem Knopf **[Übertragen auf Zentrale]** zur Zentrale übertragen.
3. Exec-Engine wieder starten mit dem entsprechenden Knopf auf der Seite *Steuerung/Status*.

## 1.7. Installation FHZ2000-Schnittstelle

### Installation FHZ2000-Schnittstelle

Bei Verwendung eines PCs als Zentrale muss bei der Konfiguration der FHZ2000 die IP-Adresse des PCs angegeben werden. Das ist in der Standard-Software für die FHZ2000 immer der Fall, da damit keine externe Zentrale verwendet werden kann. Wenn Sie nicht wissen welche Adresse Ihr PC hat, können Sie das im Fenster *Hardware* unter dem Reiter *eigene IP* nachsehen (Menüpunkt *Konfigurieren->Hardware*). Wenn mehrere IP-Adressen angezeigt werden achten Sie bitte darauf die IP-Adresse auszuwählen, die zum Netzwerk gehört, in dem die FHZ2000 angeschlossen ist. Das ist normalerweise das Netzwerk in dem die ersten drei Zahlen der IP-Adresse identisch sind mit den ersten drei Zahlen der IP-Adresse der FHZ2000.



Die Einstellung der Konfiguration in der FHZ2000 wird über einen normalen Browser vorgenommen. Zum Zugriff auf die FHZ2000 mit einem Browser lesen Sie bitte die Bedienungsanleitung, die der FHZ2000 beiliegt.

Geben Sie in einem Browser die IP-Adresse der FHZ2000 an oder den Netzwerknamen der FHZ2000, das sind die Buchstaben "FHZ" gefolgt von der Seriennummer.

Der Zugriff über den Netzwerknamen ist nicht mit allen Routern möglich. Sollte es mit Ihrem Router nicht möglich sein, müssen Sie die IP-Adresse der FHZ2000 im Router nachsehen.

Diese Möglichkeit gibt es in allen Routern, Details dazu entnehmen Sie bitte der Bedienungsanleitung Ihres Routers.

Wenn eine Verbindung zur FHZ2000 im Netzwerk hergestellt werden kann erscheint eine Kennwortabfrage.

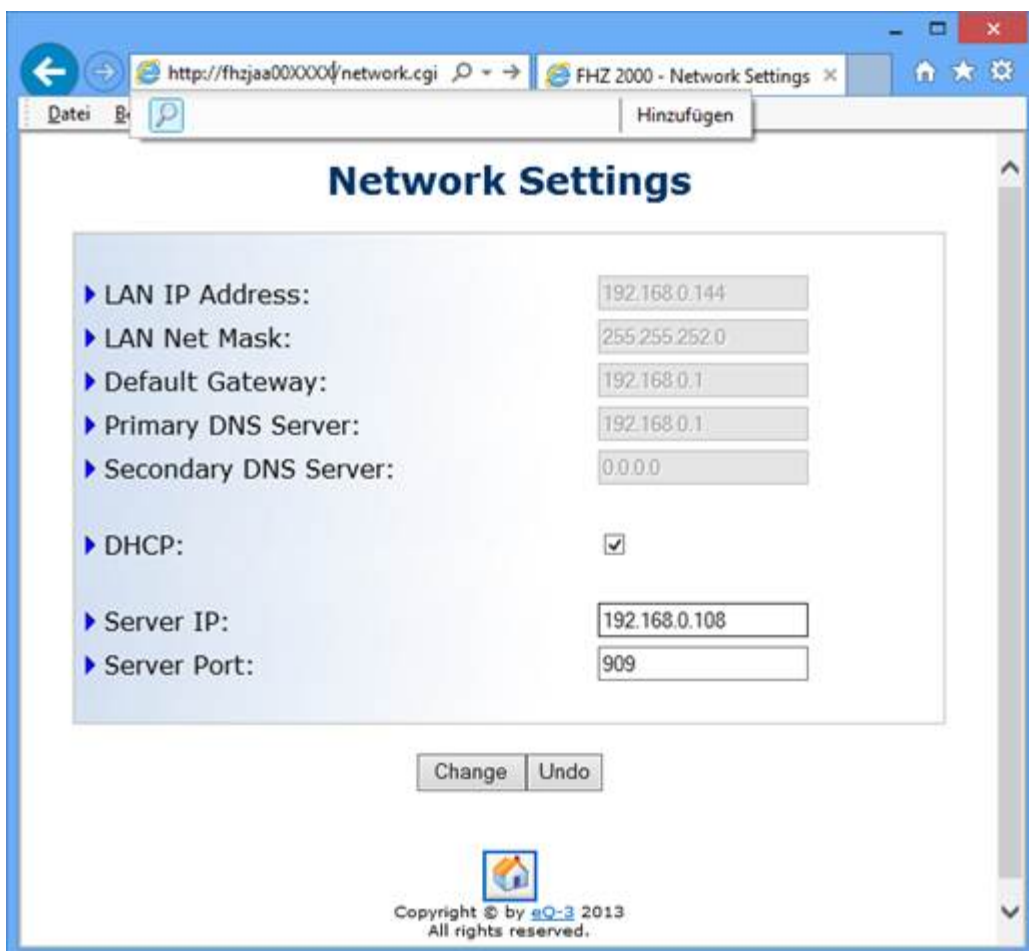
Geben Sie als Benutzer "admin" ein und lassen das Kennwortfeld frei und drücken die Eingabetaste. Es erscheint das Hauptmenü der FHZ2000.

#### Bitte unbedingt beachten:

Bei der Konfiguration kann es passieren, dass die Windows-Firewall eine Meldung auf den Bildschirm bringt, in der gefragt wird, ob die Verbindungen blockiert werden sollen. Diese Meldungen müssen immer so beantwortet werden, dass Verbindungen nicht blockiert werden!



Klicken Sie mit der Maus auf Network Settings.



Bitte geben Sie im Eingabefeld **Server-IP** die IP-Adresse der verwendeten Zentrale ein, also des PCs oder der externen Zentrale (CCU, CL-Box usw.), als Port muss 909 angegeben werden.

Klicken Sie auf den Button **Change** um die Daten zu speichern.

Klicken Sie dann auf das „home“-Symbol um ins Hauptmenü zu gelangen.

Wenn Sie FHT80-Raumregler verwenden geben Sie bitte unter dem Menüpunkt FHT-Options einen FHT-Code an. Sie können denselben FHT-Code verwenden, den Sie evtl. mit den FHTs schon

benutzen. In diesem Fall müssen Sie aber unbedingt darauf achten eine andere FHZ, die schon mit diesem Code arbeitet, stromlos zu machen, sonst funktioniert die Kommunikation mit den Raumreglern nicht.

Hinweise zum Anmelden von FHT-Raumreglern finden Sie im Kapitel Anmelden von FHT80b-Raumreglern.

## Einstellungen im PC-Programm

Wählen Sie im PC-Programm den Menüpunkt Konfigurieren->Hardware und im Hardwarefenster den Reiter *FHZ*.

Klicken Sie auf den Button **Neu** und geben eine Bezeichnung und die Seriennummer der FHZ2000 ein.

Hardware - Zentrale und Gateways

Zentrale | Gateways | **FHZ** | eigene IP

FHZ - Interfaces

Bezeichnung  
FHZ2000

Seriennummer  
JAAA0001234

Interface - Typ  
☒ FHZ2000 (Port 909)  
☐ FHZ1xxxPC  
☐ FHZ1300WLAN (Port991)

Abbrechen OK

? Hilfe

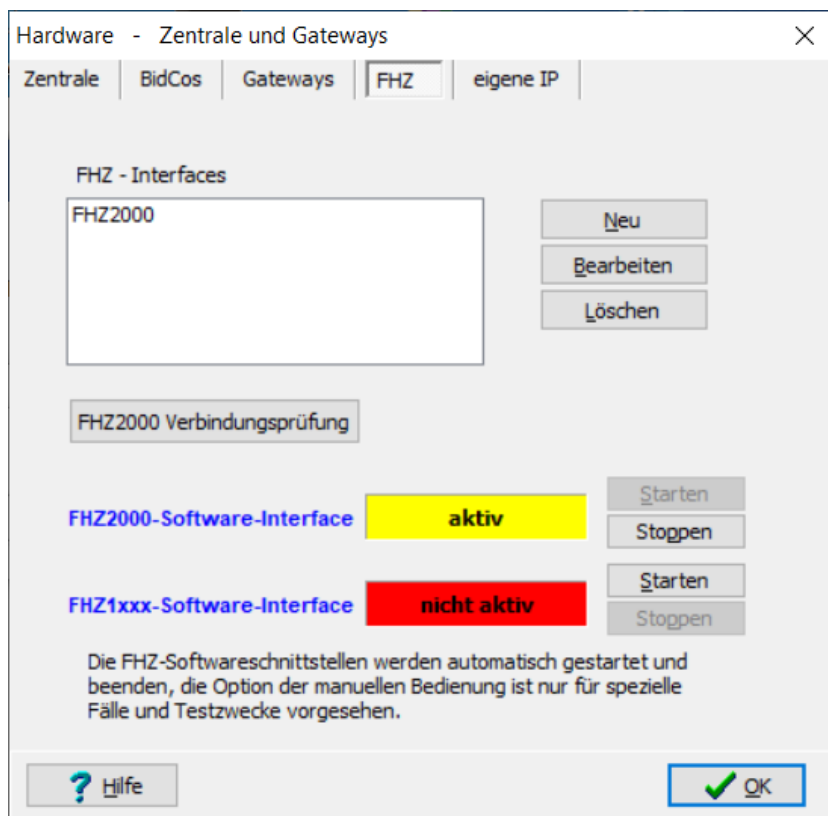
✓ OK

Achten Sie unbedingt darauf die Seriennummer korrekt anzugeben (ohne den Zusatz FHZ vor der Seriennummer !), sonst funktioniert die Kommunikation nicht. Das Programm

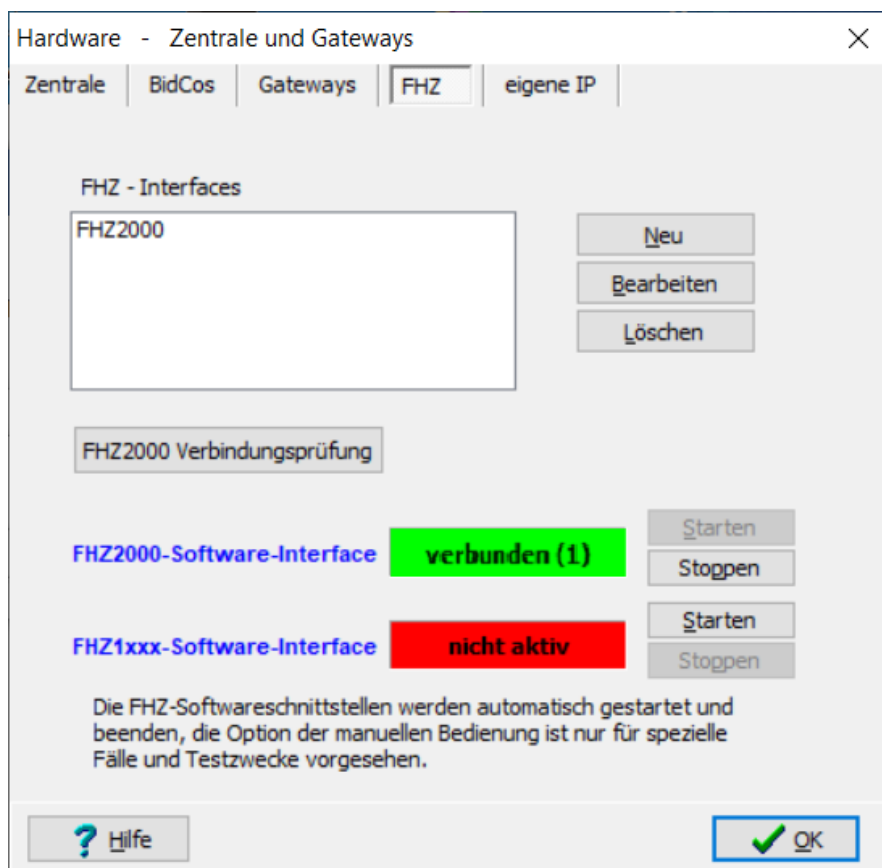
FHZ2000-Software-Interface wird normalerweise bei Bedarf immer automatisch gestartet und beendet. Manueller Start und Beendigung sind nur optional für Testzwecke vorgesehen.

Im folgenden Abschnitt ist die Verwendung eines PCs als Zentrale beschrieben, wenn Sie eine CCU oder c-comatic benutzen können Sie diesen Abschnitt überspringen, lesen Sie dann aber bitte unbedingt den nächsten Abschnitt.

## Benutzung eines PCs als Zentrale



Nachdem ein FHZ2000-Interface angelegt wurde wird automatisch das Schnittstellenprogramm für FHZ2000-Interfaces gestartet. Das Statusfeld für FHZ2000-Interfaces ist jetzt gelb, d.h. eine FHZ2000 kann sich anmelden.



Nach einigen Sekunden meldet sich die FHZ2000 automatisch an, wenn dort die richtigen Konfigurationsdaten eingetragen sind.

Sollte sich die FHZ2000 nicht anmelden so kann das folgende Ursachen haben:

- Die IP-Adresse des PCs wurde bei der Konfiguration der FHZ2000 nicht korrekt eingetragen.

- Die Verbindung wird durch Firewall oder Virensch scanner behindert. In diesem Fall müssen entsprechende Einstellungen in Firewall und/oder Virensch scanner vorgenommen werden, diese können ggfs. testweise auch kurzzeitig deaktiviert werden.

Zum Testen der Verbindung kann das FHZ2000-Software-Interface über die entsprechenden Buttons testweise gestoppt und gestartet werden.

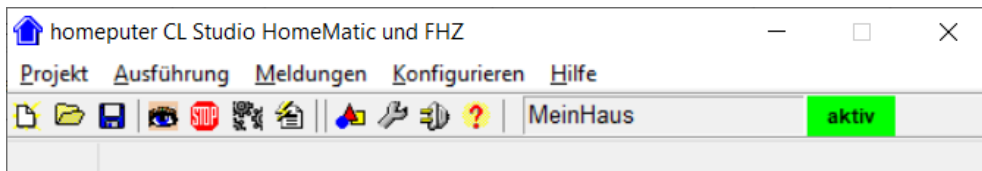
Das Software-Interface für die FHZ-Schnittstellen der Baureihe FHZ1xxxPC steht in der Standard-Version der Software nicht zur Verfügung, daher sind die entsprechenden Buttons deaktiviert und das Programm wird nicht aktiviert.

Bitte beachten Sie, dass die FHZ2000 sich erst einige Sekunden (normalerweise weniger als 10) nach dem Start der Software verbindet.

Bei der ersten Konfiguration von Geräten nach Programmstart oder nach Start des Ausführungsprogramms kann es also sein, dass es etwas dauert bis die FHZ2000 die Verbindung hergestellt hat und Konfigurationen und Steuerungen vorgenommen werden können.

Informationen zur Konfiguration der FHZ2000 finden Sie auch in den Hilfetexten der PC-Software. Nachdem Sie im PC-Programm Ihre Hardwaremodule angelegt haben, die Adressen der Module eingestellt und Programmierungen vorgenommen haben, starten Sie die Programmausführung durch Klick auf den Blitz-Button.

Bei Benutzung einer externen Zentrale (nur Versionen ab CL-Studio) wird die korrekte Übertragung nach Abschluss der Übertragung angezeigt, bei Ausführung auf dem PC erscheint im rechten Teil des Hauptfensters ein grünes Aktiv-Feld.



Nun können Sie das Visualisierungsprogramm aufrufen (Klick auf den Knopf mit dem Auge) und sehen dort in der Standardansicht **Alle Objekte** alle Module und können Aktoren steuern.

Weitere Informationen zur CL-Software finden Sie in den Hilfetexten des PC-Programms.

Wenn etwas nicht funktioniert wie erwartet überprüfen Sie bitte anhand dieser Anleitung und der Hilfetexte ob alle Einstellungen richtig vorgenommen wurden.

#### **Bitte beachten Sie:**

Wenn keine Verbindung mit der FHZ2000 hergestellt wird, kann das daran liegen, dass der Firewall oder ein Virenschutzprogramm die Netzwerkverbindung blockiert.

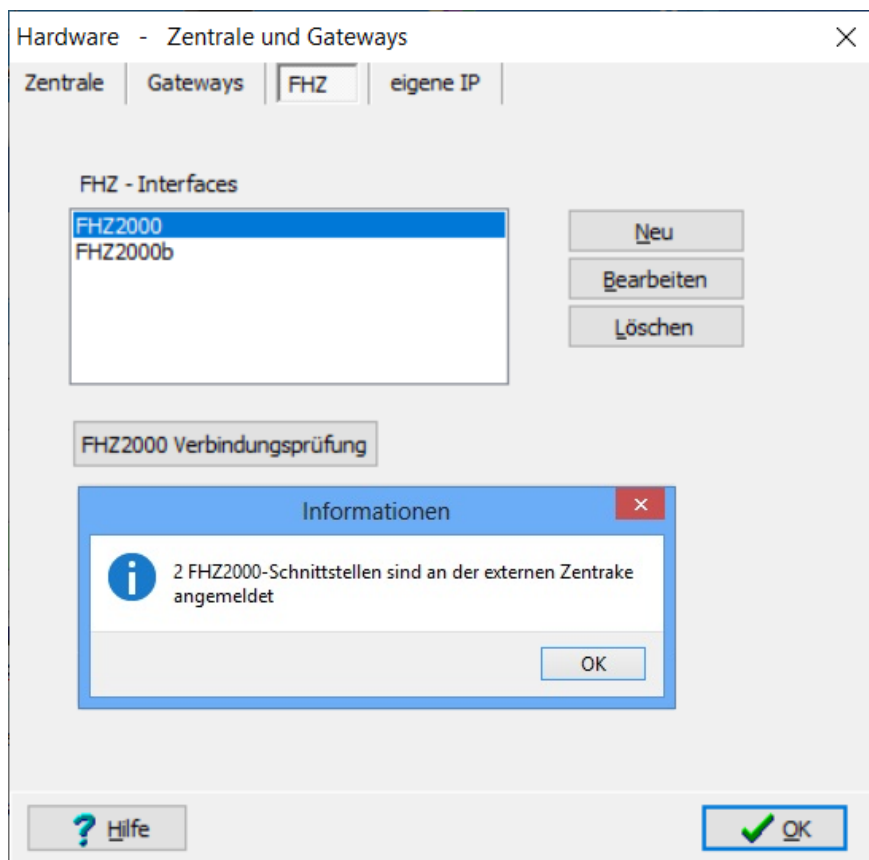
Normalerweise erscheint beim ersten Start einzelner Programm-Module eine Meldung des Firewall, in der die Netzwerk-Verbindung dauerhaft freigeschaltet werden kann. Form und Text dieser Meldung sind von der jeweiligen Windows-Version abhängig. Bitte achten Sie unbedingt darauf diese Meldung richtig zu beantworten, andernfalls wird die Verbindung blockiert und Sie müssen diese manuell im Firewall freigeben.

Sollte keine Verbindung hergestellt werden können, obwohl Firewall und Virenschutzprogramm die Verbindung nicht blockieren überprüfen Sie Ihre Netzwerkeinstellungen, insbesondere auch ob die benutzten IP-Adressen im selben Subnetzbereich liegen.

## **Benutzung einer CCU als Zentrale (ab Version CL-Studio für HomeMatic und FHZ)**

Wenn Sie eine CCU (oder CL-Box) als Zentrale benutzen müssen Sie als erstes die FHZ2000 anlegen wie im Abschnitt "Einstellungen im PC-Programm" beschrieben. Danach müssen Sie das Projekt mit der definierten FHZ2000 auf die Zentrale übertragen (Blitz-Button), damit das Schnittstellenprogramm für die FHZ2000 auf der Zentrale gestartet wird.

Es kann ca. eine Minute dauern bis eine FHZ2000 sich an der Zentrale angemeldet hat. Mit dem Button "FHZ2000 Verbindungsprüfung" können Sie überprüfen ob die FHZ2000 sich angemeldet hat.



Wenn sich die FHZ2000 nach zwei Minuten nicht an der Zentrale angemeldet hat überprüfen Sie bitte ob in der FHZ2000 die korrekte IP-Adresse der Zentrale angegeben wurde. Starten Sie die FHZ2000 ggfs. neu, indem Sie die Stromzufuhr für ca. 5 Sekunden unterbrechen. Sollte sich die FHZ2000 weiterhin nicht an der Zentrale anmelden prüfen Sie bitte ob Sie die aktuellsten Firmware- und Softwareversionen verwenden und machen ggfs. einen Update.

Bitte beachten Sie:

Das FHZ2000-Schnittstellenprogramm auf der Zentrale beendet sich automatisch nach ca. 10 Minuten wenn sich in dieser Zeit keine FHZ2000 an der Zentrale angemeldet hat. Durch einen Neustart der ExecEngine oder die neue Übertragung des Projekts können Sie dieses neu starten.



## 1.8. Einstellungen FHZ-Schnittstellen

### Einstellungen FHZ-Schnittstellen

Falls FHZ-Schnittstellen verwendet werden müssen nach der Installation der Software einige Einstellungen zur Verwendung der Schnittstellen gemacht werden.

#### FHZ2000

Die Konfiguration der FHZ2000 wird über einen Browser vorgenommen, nicht mit diesem Programm. Auf der Seite **Network Settings** muss die IP-Adresse der Zentrale eingestellt werden, wenn der PC als Zentrale benutzt wird also die IP-Adresse des PCs. Die aktuelle IP-Adresse des PCs können Sie im Reiter **eigene IP** nachsehen. Der Port muss auf 909 eingestellt werden.

Auf der Seite **FHT Options** wird ein FHT-Code angegeben, wenn FHT80-Raumregler verwendet werden. FHT80-Raumregler müssen an der FHZ2000 angemeldet werden, Infos dazu finden Sie im Kapitel [Anmelden von FHT80b-Raumreglern](#).

Weitere Informationen zur Konfiguration der FHZ2000 finden Sie in der Bedienungsanleitung der FHZ2000.

Bitte beachten Sie, dass die FHZ2000 sich erst ca. 10 Sekunden nach dem Start der Software verbindet.

Die Seriennummer der FHZ2000 muss korrekt angegeben werden damit Meldungen jeweils der richtigen Schnittstelle zugeordnet werden können.

Bei der Konfiguration der Aktoren muss angegeben werden, welche Schnittstelle jeweils an den Aktor senden soll.

Wenn Aktoren nicht geschaltet werden überprüfen Sie bitte, ob die Seriennummer des Interfaces korrekt angegeben wurde und ob in der Konfiguration des Aktors die richtige FHZ-Schnittstelle ausgewählt wurde.

Hardware - Zentrale und Gateways

Zentrale | Gateways | **FHZ** | eigene IP

FHZ - Interfaces

Bezeichnung  
FHZ2000Oben

Seriennummer  
JAA0001234

Interface - Typ  
☒ FHZ2000 (Port 909)  
☐ FHZ1xxxPC  
☐ FHZ1300WLAN (Port991)

Abbrechen OK

? Hilfe

✓ OK

## FHZ1xxxPC

### Hinweis:

Die Einbindung von Schnittstellen der Baureihe 13xxxPC ist nur mit der Version CL-Studio oder höher möglich.

Wenn der PC als Zentrale verwendet wird, muss bei FHZ-Schnittstellen des Typs FHZ1xxxPC der FTDI-Treiber für einen virtuellen COM-Port installiert werden.

Welcher virtuelle COM-Port für die FHZ verwendet wird können Sie im Gerätemanager der Systemsteuerung sehen.

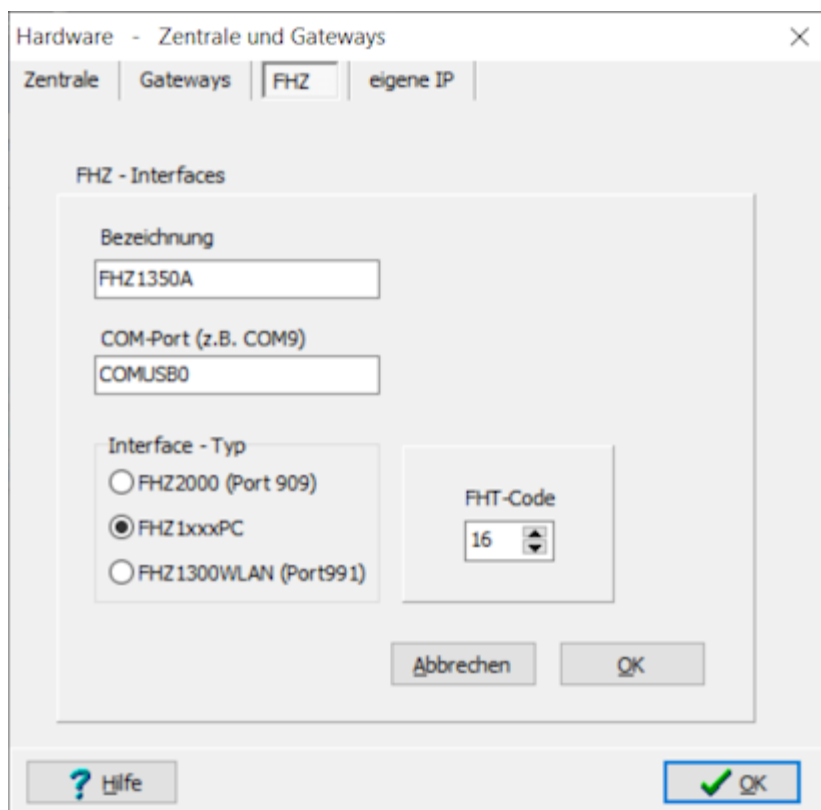
**Der normale mitgelieferte USB-Treiber der FHZ kann nicht verwendet werden !**

**Wenn eine externe Zentrale verwendet wird ist die Installation des Treibers nicht nötig !**

Falls der PC als zentrale verwendet wird und der Treiber unter WIN8 installiert werden muss wird es leider etwas komplizierter. Infos zur Installation eines nicht signierten Treibers finden Sie im Internet bei der Suche zur Treiberinstallation unter WIN8.

Wenn eine Linux-Zentrale (z.B. c-comatic oder CCU3) benutzt wird, muss bei der Linux Port-Bezeichnung die Kennung "tty" durch "COM" ersetzt werden, auf der c-comatic und der CCU3 ist die Schnittstelle normalerweise ttyUSB0, also wird als COM-Port COMUSB0 angegeben. Sollte ein weiterer USB-Port an der Zentrale benutzt werden, kann es sein, dass dieser den USB-Port schon belegt. In diesem Fall muss als Port COMUSB1 angegeben werden.

Mit einer CCU2 können USB-Schnittstellen nicht benutzt werden, da diese keinen USB-Treiber für eine FHZ hat.



Bei Verwendung einer FHZ1300(W)LAN muss die IP-Adresse der Schnittstelle angegeben werden. Als Port muss bei der Konfiguration der Schnittstelle 991 angegeben werden.

## 1.9. Benutzung des PCs als Zentrale

---

### Benutzung des PCs als Zentrale

Je nach Version können verschiedene Hardwareplattformen als Zentrale ausgewählt werden. Die Programmversionen ab CL-Studio haben im Fenster *Zentrale* (Menüpunkt *Konfigurieren->Hardware*) eine Auswahlbox, in der die benutzte Zentrale ausgewählt werden kann, wobei das je nach Version auch der PC sein kann.

Mit dieser Option ist es möglich eine Hausautomation mit einem PC ohne eine separate Zentrale zu realisieren, es wird nur ein Konfigurationsadapter oder eine FHZ-Schnittstelle (im Folgenden Interface genannt) benötigt um mit den Modulen zu kommunizieren.

In der Version CL-Studio für FHZ kann keine externe Zentrale benutzt werden, der PC ist als Zentrale eingestellt und diese Einstellung kann nicht verändert werden.

Der PC ist jedoch als Zentrale nicht in gleicher Weise geeignet wie die entsprechenden zur Verfügung stehenden Zentralen, insbesondere bezüglich der im Normalfall erforderlichen hohen Betriebssicherheit, die aufgrund der vielen unterschiedlichen Programme und Rahmenbedingungen auf einem PC nicht mit einer externen separaten Zentralen vergleichbar ist.

Auf dieser Seite ist unter anderem beschrieben wie die Interfaces (LAN- oder USB-Version) konfiguriert werden, wenn der PC als Zentrale benutzt wird. Falls Sie den PC als Zentrale benutzen, lesen Sie bitte unbedingt diese Hinweise und gehen bei der Konfiguration wie in den Hinweisen beschrieben vor.

Die ExecEngine ist die Software, die im laufenden Betrieb die Steuerung übernimmt und mit den einzelnen Modulen kommuniziert. Im Regelfall läuft die ExecEngine auf einer externen Linux-Zentrale (z.B. CCU, c-comatic oder RaspberryMatic). Sie kann jedoch auch auf dem PC laufen, so dass keine separate externe Zentrale erforderlich ist. Mit der Option die ExecEngine auf dem PC laufen zu lassen besteht auch die Möglichkeit Anwendungsprogramme zu testen ohne die laufende Anwendung einer externen Zentrale zu behindern. Falls eine Anwendung auf dem PC getestet werden soll während die ExecEngine auf einer externen Zentrale aktiv ist, muss auf jeden Fall die Option *Hardwareverbindung deaktivieren* ausgewählt werden, sonst kann es zu Konflikten und Kommunikationsproblemen mit der externen Zentrale kommen.

#### **Bitte beachten Sie Folgendes wenn Sie die Benutzerkontensteuerung von Windows eingeschaltet haben:**

Standardmässig wird das Programm nicht mit Administratorrechten gestartet um keine Sicherheitsabfrage bei eingeschalteter Benutzerkontensteuerung auszulösen. Je nach Windows-Version und Benutzerkonteneinstellung kann aber auf Dienste (also auf den BidCoS-Service) nur von einem Programm mit Administratorrechten zugegriffen werden. Bei eingeschalteter Benutzerkontensteuerung kann es je nach Windows-Version (insbesondere bei Windows 7) sein, dass der Status des BidCos-Services auf "nicht aktiv" steht, obwohl er tatsächlich aktiv ist. Der BidCos-Service kann dann auch nicht vom Programm gestartet oder beendet werden. Um dieses Problem zu beseitigen können Sie das Programm mit Administratorrechten starten lassen. Dazu klicken Sie die Programmdatei im Dateimanager mit der rechten Maustaste an und wählen im PopUp-Menü den Menüpunkt "Eigenschaften". Gehen Sie im Fenster das sich dann öffnet auf den Reiter "Kompatibilität" und aktivieren das Optionsfeld "Programm als ein Administrator ausführen" im unteren Teile des Fensters. Nach dieser Änderung kann es (wieder je nach Windows-Version) passieren, dass Sie die Programmausführung bei jedem Aufruf bestätigen müssen.

Normalerweise wird der BidCos-Service beim Start des Rechners automatisch gestartet wenn es eine korrekte BidCoS-Konfigurationsdatei gibt. Daher ist es nach der Erstellung der Konfigurationsdatei nicht unbedingt nötig den BidCosS-Service vom Programm steuern und anzeigen zu lassen. Allerdings wird ohne die Administratorrechte der Status immer als "nicht aktiv" angezeigt. Alternativ gibt es natürlich noch die Möglichkeit die Benutzerkontensteuerung einfach auszuschalten....

#### **Hinweise zur Konfiguration der BidCoS-Interfaces für HomeMatic**

Falls Sie HomeMatic-LAN-Interfaces verwenden starten Sie das Programm *HomeMatic->Hilfsprogramme->HomeMatic-Lan-Interface konfigurieren*.

In diesem Programm werden alle im Netzwerk befindlichen LAN-Interfaces angezeigt. Die Interfaces

sollten mindestens die Firmware-Version 0.961 haben, falls Ihr Interface eine kleinere Versionsnummer hat machen Sie bitte einen Firmwareupdate. Nehmen Sie die IP-Einstellungen vor. Sollte es Verbindungsprobleme geben deaktivieren Sie DHCP und vergeben feste IP-Adressen. Aktivieren Sie anfangs noch nicht die AES-Verschlüsselung. Aktivieren Sie diese erst wenn die Kommunikation mit allen Komponenten problemlos funktioniert.

Für den Betrieb der ExecEngine auf dem PC ist der BidCos-Service erforderlich. In den Installationsdateien zum LAN- bzw. USB-Interface ist dieser enthalten.

Die Downloads finden Sie auf folgender HomeMatic-Seite:

<http://www.homematic.com>

dann auf die Downloadseite aufrufen.

Der momentan aktuelle Pfad ist: Service->Downloads->Software.

Hier auch der momentan aktuelle direkte Link auf die Downloadseite:

<http://www.eq-3.de/software.html>

Im Abschnitt "Konfigurationsadapter LAN" die Setup-Dateien für das LAN-Interface downloaden, darin befindet sich auch eine Setup-Datei für den BidCos-Service.

Es ist möglich, dass die hier angegebenen aktuellen Links auf die Downloadseite sich ändern.

Vor dem nächsten Schritt muss der BidCos-Service installiert worden sein, sonst fehlt die erforderliche Konfigurationsdatei.

Bevor Sie Module anlernen müssen die verwendeten BidCoS-Interfaces in einer Konfigurationsdatei definiert werden. Öffnen Sie dazu das Fenster *Hardware* der homeputer CL-Software unter dem Menüpunkt *Konfigurieren->Hardware*. Wählen Sie in dem Fenster *Zentrale* den Reiter *Zentrale*. Aktivieren Sie die Option *Ausführung auf dem PC*. Klicken Sie auf den Knopf [BidCoS-Interfaces konfigurieren]. Das Programm *Konfiguration BidCoS-Interfaces* wird gestartet. Bei ersten Start befinden sich normalerweise Testdaten in der Konfigurationsdatei. Löschen bzw. ändern Sie diese entsprechend den von Ihnen benutzten Interfaces. Legen Sie in diesem Programm alle benutzten Interfaces an. Vergessen Sie nicht den korrekten Typ einzustellen und achten Sie darauf die Seriennummer korrekt einzugeben. Wählen Sie als Log-Level den normalen Status "Fehler" damit die Log-Dateien nicht zu gross werden.

Beenden Sie dieses Programm durch Anklicken des Knopfes oder Menüpunkts [Beenden und Konfigurationsdatei speichern].

Wenn der BidCos-Service sich nicht starten lässt, ist das meistens auf Fehler in der Konfigurationsdatei zurückzuführen. Das kann z.B. durch eine falsche Seriennummer für ein Interface verursacht werden. Wenn der BidCos-Service startet, die Kommunikation aber nicht funktioniert überprüfen Sie bitte die Eintragungen in der Konfiguration, insbesondere Seriennummer und Typ der Interfaces.

Wenn die Power-LED bei dem LAN-Interface blinkt hat der BidCos-Service keine Verbindung zum LAN-Interface weil der BidCos-Service nicht aktiv ist oder keine Netzwerkverbindung zum BidCos-Service aufgebaut werden kann. Ursache können fehlerhafte Einstellungen in der Konfigurationsdatei sein.

### **Hinweise zum Anlernen und Konfigurieren von HomeMatic-Komponenten**

Starten Sie den HomeMatic-Konfigurator unter dem Menüpunkt

HomeMatic->*HomeMatic-Komponenten konfigurieren*.

Der HomeMatic-Konfigurator zum Anlernen der HomeMatic-Komponenten kann über den BidCoS-Service mit dem oder den Interfaces kommunizieren oder falls Sie nur ein Interface benutzen auch direkt mit dem Interface verbunden werden. Die Auswahl erfolgt unter dem Menüpunkt *Datei->Einstellungen* des HomeMatic-Konfigurators.

Bitte beachten Sie dabei: Wenn Sie mit mehreren Interfaces arbeiten können Sie den Auswahlpunkt *Entfernter BidCoS-Service* benutzen, dann muss der BidCoS-Service auf dem PC vorher gestartet worden sein (ist also in Wirklichkeit in diesem Fall nicht "entfernt", als IP-Adresse geben Sie die IP des PC's an). Der BidCoS-Service wird mit dem CL-Programm im Fenster *Hardware* auf der Seite *Zentrale* gestartet oder gestoppt. Das Fenster *Hardware* wird mit dem Menüpunkt *Konfigurieren->Hardware* geöffnet.

Wichtig ist, dass die BidCoS-Konfigurationsdatei wie im vorigen Abschnitt beschrieben mit den Einstellungen für die verwendeten Interfaces erstellt wurde.

Wenn Sie nur ein Interface (egal ob LAN oder USB) benutzen können Sie die Konfiguration auch auf diese Weise konfigurieren oder jeweils die Option *HomeMatic-Lan-Interface* oder *HomeMatic-USB-Interface* auswählen. Wenn Sie die Interfaces nicht über den BidCoS-Service verbinden, sondern die Einzelverbindung benutzen, muss der vorher BidCoS-Service beendet werden, da es sonst zu Kommunikationskonflikten kommen kann.

Lernen Sie nun die HomeMatic-Module mit dem Menüpunkt *Geräte anlernen* wie in der Bedienungsanleitung der Geräte beschrieben an.

Nach dem Anlernen der Geräte schliessen Sie das Programm HomeMatic-Konfigurator und starten den BidCoS-Service falls dieser nicht aktiv ist.  
Dann öffnen Sie das Fenster Modulauswahl über den Menüpunkt *Konfigurieren->Modulauswahl* und klicken auf den Knopf [Module importieren] und die angelernten Module in die Software zu übernehmen. Hinweise zur weiteren Vorgehensweise finden Sie im Kapitel [Erste Schritte](#), wobei Sie die Teile, die sich auf die externen Zentralen beziehen überspringen können.

## 1.10. Installation Hardware

---

### Installation Hardware

Die Zentrale bzw. Funkschnittstellen dürfen nicht unmittelbar neben oder auf anderen elektronischen Geräten wie WLAN-Router, PCs und Geräten stehen, insbesondere wenn diese Funkfunktionen beinhalten.

Funkmodule sollten nicht in der Nähe grosser Metallflächen oder direkt neben oder auf elektronischen Geräte aufgestellt werden.

Die Reichweite von Funkmodulen im Gebäude ist abhängig von den individuellen Örtlichkeiten und wird durch Decken, Wände und elektromagnetische Strahlungen vermindert.

**Beachten Sie unbedingt die Sicherheitshinweise der jeweiligen Bedienungsanleitungen bei der Installation von elektrischen Komponenten.**

# 1.11. Die wichtigsten Begriffe des Systems

---

## Die wichtigsten Begriffe des Systems

Ein **Projekt** ist die komplette individuell erstellte Anwendung.

**Makro** ist die Bezeichnung für eine oder mehrere Anweisungen, die bestimmte Aktionen bewirken. Das System verfügt über leistungsfähige Anweisungen, mit denen auch sehr komplexe Funktionen realisiert werden können. In Makros können Aktionen über wenn-Anweisungen von bestimmten Bedingungen abhängig gemacht werden. Das können Zustände bzw. Werte von Objekten, aber auch z.B. Wochentag und Uhrzeit sein.

**Aktoren** sind Module, die Endgeräte wie Beleuchtungen, Jalousien usw. schalten und steuern.

**Sensoren** sind Eingabeelemente wie z.B. Taster, Fernbedienungen, Bewegungsmelder und Temperatursensoren. Sensoren wie Taster und Fernbedienungen, die einen Zustand senden (z.B. aus/an/kurz/lang) werden als Binärsensoren oder Schaltsensoren bezeichnet. Sensoren, die einen Zahlenwert senden, also z.B. Temperatur oder Luftfeuchtigkeit werden als Analogsensoren bezeichnet.

**Objekt** ist der Oberbegriff für Sensoren und Aktoren, wobei der Begriff „Objekt“ auch die Definitionen und Makros des Sensors bzw. Aktors mit einschließt. Wenn das Steuerungsprogramm Objektzustände von Aktor-Objekten verändert wird automatisch eine entsprechende Meldung an den Aktor gesendet. Wenn ein Sensor eine Meldung sendet wird das Makro des Sensor-Objekts ausgeführt.

In der Studio-Version und höheren Versionen gibt es die Möglichkeit Objekte ohne zugeordnete Hardware zu definieren, sogenannte "virtuelle Objekte".

**Ansichten** zeigen die Zustände und Werte von Objekten und ermöglichen je nach Objekttyp die Änderung von Zuständen und Werten am Bildschirm.

In der Standard-Version können bis zu drei, in der Studio-Version beliebig viele individuelle Ansichten erstellt werden. Diesen Ansichten kann optional ein Hintergrundbitmap zugewiesen werden.

Welche Darstellungsart verwendet werden soll, kann für jede Ansicht in der jeweiligen Definition eines Objekts definiert werden.

# 1.12. Makro-Programmierung

---

## Makro-Programmierung

Die Programmiersprache des Systems ermöglicht es jedem Anwender ohne spezielle Programmierkenntnisse komplexe Steuerungsfunktionen zu realisieren. Für jedes Objekt kann ein Makro erstellt werden, welches durch die Änderung von Zuständen bzw. Werten die mit den Objekten verbundenen Aktoren steuert. Zustände und Werte von Objekten und Variablen sowie Zeiten können mit vielfältigen Bedingungen abgefragt und durch Zuweisungen verändert werden.

Makros von Sensor-Objekten werden normalerweise immer ausgeführt, wenn eine Meldung von dem Sensor empfangen wird, beispielsweise also wenn eine Taste einer Fernbedienung gedrückt wird oder die aktuelle Temperatur eines Temperatursensors empfangen wird. Dazu wird beim Anlegen eines Sensor-Objekts standardmässig automatisch das Optionsfeld "Ausführen bei Empfang" aktiviert.

Makros von Aktoren werden üblicherweise immer in einem bestimmten Zeitintervall aktiviert, beispielsweise um den Aktor nach einer bestimmten Zeit auszuschalten oder zu bestimmten Situationen und Abhängigkeiten von den Zuständen mehrerer anderer Objekten oder Zeiten zu steuern. Nach Möglichkeit sollte das gewählte Zeitintervall für die Makroausführung nicht kleiner als nötig gewählt werden. Üblicherweise reicht eine Minute, wenn jedoch Uhrzeitabfragen im Makro vorkommen, bei denen keine ganzen Minuten sondern Sekunden abgefragt werden muss das Zeitintervall kleiner, normalerweise auf 5 Sekunden eingestellt werden. Bitte beachten Sie dabei, dass die normale Uhrzeit im 5-Sekundentakt aktualisiert wird, die Uhrzeitangabe in den Bedingungen also ebenfalls eine Uhrzeit im 5-Sekundenintervall sein muss.

Die Makros können bei der Studio-Version wahlweise durch Menüs automatisch erstellt werden oder über die Tastatur eingegeben werden. Anstelle der in der Beschreibung der einzelnen Anweisungen verwendeten Syntax kann auch die bei der automatischen Erstellung von Makro-Anweisungen verwendete Syntax über die Tastatur eingegeben werden verwendet werden. In jeder Zeile eines Programms darf nur eine Anweisung stehen, Kommentarzeilen müssen mit 2 Schrägstrichen (//) beginnen, diese werden dann bei der Generierung des Codes ignoriert. In den Beispielen sind Programmanweisungen eingerückt, dies ist nicht erforderlich, aber aufgrund besserer Übersichtlichkeit empfehlenswert.

Wenn während der Code-Erstellung Fehler auftreten, wird der fehlerhafte Quelltext mit einer entsprechenden Fehlermeldung angezeigt. Die fehlerhafte Zeile wird soweit möglich markiert, hier kann es eventuell zu Verschiebungen kommen, da Fehler im Quell-Code nicht immer einer Zeile zugeordnet werden können. Wenn in der markierten Zeile kein Fehler erkennbar ist, müssen die vorangegangenen Zeilen geprüft werden.

Anweisungsteile, die bei der Erklärung der Syntax in < spitzen Klammern > stehen sind optional und können weggelassen werden.

Die Uhrzeit wird nicht jede Sekunde, sondern alle 5 Sekunden aktualisiert. Bei Zeitvergleichen ist also zu beachten, dass die Sekunden immer durch 5 teilbar sein müssen.

Wenn Schaltmodule eingeschaltet werden, kann die Einschaltdauer angegeben werden.

### Die Elemente der Programmiersprache sind die

- **Objekte** mit ihren Zuständen, Werten und Variablen. Zustände und Werte von Objekten (und somit die zugehörige Hardware) können abgefragt und verändert werden.
- **Operatoren** für logische Verknüpfungen und Vergleiche
- **Schlüsselwörter**, die in der Programmiersprache definierte Bedeutungen haben
- **Anweisungen**, die in der Programmzeile beschreiben was getan werden soll
- **Funktionen**, die in der Programmzeile Werte liefern, z.B. für Vergleiche.

## Objekte

### Hardwareobjekte

sind Objekte, die bestimmter Hardware zugeordnet sind und zum Schalten, Messen, Regeln etc. benutzt werden. Man unterscheidet zwischen Sensoren und Aktoren.



**Sensoren** sind Eingabeelemente wie Taster von Fernbedienungen, Bewegungsmelder, Fensterkontakte, Temperatursensoren, Wassermelder, Rauchmelder usw. In den Makros der Sensoren stehen üblicherweise Anweisungen um Aktionen auszuführen wenn eine Meldung von der zugehörigen Hardware empfangen wird. Eine Makroanweisung zu einem Taster könnte also z.B. einen Aktor schalten wird, das sieht z.B. so aus:

```
Stehlampe umschalten
```

**Aktoren** sind Schaltmodule, an die Endgeräte wie Beleuchtung, Rollläden, Schalter, Markisen etc. angeschlossen werden. Wenn Makros von Aktoren benutzt werden, werden diese normalerweise in bestimmten Zeitintervallen ausgeführt. So kann z.B. ein Licht immer nach einer bestimmten Zeitdauer ausgeschaltet werden durch folgendes Makro:

```
wenn Badventilator eingeschaltet und
    Schaltdauer(Badventilator) groesser "00:10:00" dann
    Badventilator ausschalten
endewenn
```

### Virtuelle Objekte

sind Objekte, die keiner Hardware zugeordnet und z.B. nur zur Bildschirm-Ein- oder Ausgabe benutzt werden.

Im Fenster Einstellungen können über den Button "Neues Objekt" virtuelle Objekte erstellt werden. Dabei wird der Name, die Bezeichnung und der Typ festgelegt: z.B. Name: Anzeige, Bezeichnung: Anzeige, Typ: Zeichen.

In einem Makro kann dem virtuelle Objekt Anzeige dann ein Wert zugewiesen werden, z.B:

```
Anzeige:="aktuelle Temperatur ist " +
WandthermostatWohnzimmer.Temperatur+" Grad C"
```

### Variablen

Variablen dienen der Zwischenspeicherung von Werten. Variablen werden wie die Objektprogramme auf der Seite Programmierung der Objektdefinition definiert. Einer Variablen kann im Objektprogramm ein Wert zugewiesen werden. Dieser Wert kann der Inhalt einer anderen Variablen, ein Objektzustand oder eine Konstante sein. Die in einem Objektprogramm definierten Variablen können von allen anderen Objektprogrammen benutzt werden, wenn der Name des Objekts, in dem die Variable definiert wurde, der Variablen getrennt durch einen Punkt vorangestellt wird.

Beispiel:

```
Anzeige:=RaumthermostatWohnen.Temperatur
```

Für jede Variable muss ein Typ festgelegt werden. Es können die gleichen Typen wie für Objekte verwendet werden.

#### **Typen sind z.B.: Licht, Schalter, Zeichen, Zahl**

Zusätzlich zu diesen Typen können für Variablen auch Zeittypen (Uhrzeit, Datum) benutzt werden. Bei Zuweisungen zwischen nicht gleichen Typen erfolgt die Zuweisung entsprechend der Reihenfolge der für diesen Typ definierten Zustände, unabhängig von der Bezeichnung des Zustands. Intern werden Zustände als Zahl entsprechend der Reihenfolge in der Definition des Typs behandelt. Bei Zuweisungen wird diese Zahl zugewiesen.

Es gibt eine vordefinierte Variable des Typs Uhrzeit für jedes Objekt. Der Name der Variablen ist **CT** (für **ChangeTime**). In dieser Variablen wird die jeweils letzte Änderungszeit eines Objekts gespeichert. Mit der Funktion **STOPPUHR(Objektname.CT)** kann ermittelt werden wie lange ein Objekt sich in seinem aktuellen Zustand befindet.

Bitte beachten Sie:

Wenn Sie eine Variable vom Typ Zahl definieren, gibt es zwei unterschiedliche Typen von Zahlen: Zahlen ohne Komma (**Ganzzahl**) und Zahlen mit Komma (**Gleitkommazahl**). Von welchem Typ die Zahl ist, wird durch den Startwert festgelegt. Wird als Startwert eine Zahl mit Komma angegeben (z.B. 1,0) so wird die Variable als Gleitkommazahl behandelt, andernfalls als Ganzzahl.

Es macht bei Rechenoperationen und Zuweisungen einen großen Unterschied wie eine Zahl definiert ist.

Wenn einer Ganzzahl eine Gleitkommazahl zugewiesen wird, werden die Kommastellen einfach abgeschnitten.

Wenn eine Rechenoperation nicht das gewünschte Ergebnis liefert, prüfen Sie wie die in der Anweisung verwendeten Zahlen definiert sind.

Beispiel:

Die Minimal und Maximal-Temperatur sollen täglich auf einer Anzeige ausgegeben werden. Im Objektprogramm des Objekts TempAussen (das Thermometer im Garten) sind die Variablen MIN und MAX als Zahl definiert. Das Makro des Objekts TempAussen lautet:

```
wenn TempWG > MAX dann
  MAX:=TempWG
endewenn
wenn TempWG < MIN dann
  MIN:=TempWG
endewenn
```

Das Makro des Objekts Anzeige, das zu jedem Tageswechsel aktiviert wird:

```
AnzeigeMinMax:="Min: "+TempWG.MIN+" Max: "+TempWG.MAX
```

## Zuweisungen

Zustände bzw. Werte von Objekten oder Variablen können durch Zuweisungen geändert werden. Die grundsätzliche Form einer Zuweisung im Programm ist

**Ziel := Quelle**

wobei Ziel ein Objekt oder eine Variable sein kann. Quelle kann ein Objekt, eine Variable oder eine Konstante

sein kann. Bei Zuweisungen von Werten unterschiedlichen Typs wird soweit möglich automatisch eine

Konvertierung entsprechend des Typs des Ziels vorgenommen.

Bei Zuweisungen zwischen nicht gleichen selbst definierten Typen erfolgt die Zuweisung entsprechend der Reihenfolge der für diesen Typ definierten Zustände, unabhängig von der Bezeichnung des Zustands. Zum Beispiel muss bei der Definition neuer Typen mit den Zuständen AUS/AN darauf geachtet werden, dass der Zustand AUS immer zuerst definiert wird, damit bei Zuweisung an andere AUS/AN - Objekte der richtige Zustand übertragen wird.

Bei der Verwendung von Schlüsselwörtern wie **einschalten** bei Zuweisungen oder **ausgeschaltet** in Wenn-Bedingungen ist es erforderlich, dass AUS immer als erster Zustand definiert wird, da intern mit der numerischen Reihenfolge der Zustände gearbeitet wird. Durch diese Verfahrensweise ist es möglich, auch Zuweisungen zwischen Typen mit unterschiedlicher Bezeichnung für gleiche Zustände durchführen zu können. Bei der Definition der Zustände von Fenstern und Türen muss **offen** als erster Zustand und **zu** als zweiter Zustand definiert werden, damit die entsprechenden Schlüsselwörter bei Zuweisungen und Bedingungen benutzt werden können.

## Schlüsselwörter für Zuweisungen

Für Schalter: **an, aus, einschalten, ausschalten**

Bei Lichtern oder Geräten: **einschalten für**

Für Rollläden: **rauffahren, runterfahren**

Beispiele für Zuweisungen:

**LichtBad:=SchalterBad** hat die gleiche Wirkung wie

**LichtBad wie SchalterBad**

**Licht1Garten einschalten** hat die gleiche Wirkung wie

**Licht1Garten:=an** oder **Licht1Garten:=1**

Die besonderen Schlüsselwörter **AN** und **AUS** können bei einer Zuweisung mit dem Zuweisungsoperator **:=** ohne Hochkommas verwendet werden.

Wird einem Objekt ein anderer Zustand als Text zugewiesen, so muss dieser in Hochkommas gesetzt werden und genau der Schreibweise der Typdefinition entsprechen, auch Groß - / Kleinschreibung muss berücksichtigt werden.

Beispiel:

Anstatt

**Rolllade runterfahren**

Könnte man auch die Anweisung

**RollladeWohnen:= "unten"**

verwenden, das Wort **unten** muss genau wie in der Typdefinition geschrieben sein und in Hochkommas gesetzt werden.

Eine besondere Anweisung zum Einschalten von Lichtern oder Geräten ist die Anweisung *Objekt einschalten für Zeitdauer*

als Zeitdauer kann die Zeit in Hochkommas oder als Variable angegeben werden.

Beispiel:

**Stehlampe einschalten für "00:30:00"**

oder:

**Dauer1:="00:10:00"**

**Gartenlicht einschalten für Dauer1**

wobei Dauer1 eine Variable vom Typ *Uhr* oder *Zeichen* sein muss.

### Rechenfunktion bei Zuweisungen numerischer Werte

Bei Zuweisungen an ein Objekt bzw. eine Variable des Typs *Zahl* können die vier Grundrechenarten benutzt werden. Damit ist es z.B. möglich Eingabewerte von Sensoren für die weitere Verarbeitung zu verändern.

Beispiel:

**NeuTemp:=AltTemp-5**

### Rechenfunktion bei Zuweisungen von Uhrzeiten

Uhrzeiten können mit Hilfe der Operatoren + und - addiert bzw. subtrahiert werden.

Beispiel:

Sie wollen, dass ein Makro alle morgendlichen Aktivitäten um eine Viertelstunde verzögert. Dies könnte so

aussehen:

**RollRaufZeit:= RollRaufZeit + "00:15:00"**

**KaffeeEinZeit:= KaffeeEinZeit + "00:15:00"**

### Rechenfunktion bei Zuweisungen von Datumswerten

Es ist auch möglich ein Datum durch eine Rechenoperation zu ermitteln.

Das Datum 11 Tage nach dem aktuellen Tag kann man z.B. einfach ermitteln mit der Anweisung:

**Zukunft:=Datum+11**

wobei die Variable Zukunft natürlich vom Typ *Datum* sein muss.

### Verknüpfungsfunktion bei Zuweisungen von Texten

Mit Hilfe von Zuweisungen können zwei oder mehr Zeichenketten verbunden werden. Wenn in der Anweisung

Operanden eines anderen Typs als Zeichenkette benutzt wird, werden diese soweit möglich automatisch konvertiert.

Beispiel:

**AnzeigeKueche:="Garage ist "+Garagentor**

Wenn das Objekt Garagentor die Typen *auf* und *zuhat*, und das Garagentor auf ist erscheint in der Anzeige

*Garage ist auf*

**AnzeigeBad:=Uhrzeit+" Uhr"**

Der Text in der Anzeige sieht dann so aus: *22:30:00 Uhr*

## wenn - Bedingungen

In Wenn-Anweisungen wird aufgrund von Bedingungen entschieden, welche weiteren Anweisungen ausgeführt werden. Eine Bedingung ist entweder WAHR oder FALSCH.

### Vergleichsoperatoren

= oder **gleich**

<> oder **ungleich**

< oder **kleiner**

> oder **groesser**

<= (kleiner oder gleich)

>= (größer oder gleich)

## Logische Operatoren

**UND** (im Sinne von und zugleich), **ODER** (im Sinne von oder auch),

**NICHT** (zur Negation der nachfolgenden Bedingung)

Bedingungen sind folgendermaßen aufgebaut:

**WENN** <**NICHT**> *Bedingung* <**UND/ODER**> *Bedingung* **DANN**

*Anweisungen*

<**SONST**>

*Anweisungen*

**ENDEWENN**

Die Wenn-Anweisung ist eine Anweisung, die sich über mehrere Zeilen erstreckt.

Mit der WENN-Anweisung ist es möglich, den weiteren Programmablauf von einer oder mehreren Bedingungen abhängig zu machen. Wenn-Anweisungen können auch verschachtelt werden, d.h. zwischen dem **wenn** und dem **endewenn** (bzw. **SONST**) können weitere Wenn-Anweisungen stehen. Jede Wenn-Anweisung muss mit einer endewenn-Anweisung beendet werden, ansonsten wird bei der Code-Generierung eine entsprechende Fehlermeldung ausgegeben.

Beispiel:

```
WENN LichtBad eingeschaltet UND
    SCHALTDAUER(LichtBad) groesser "00:00:30" DANN
    Ventilator einschalten
ENDEWENN
```

Vor jeder Bedingung kann ein **NICHT** gesetzt werden, dann wird die Anweisung hinter dann ausgeführt wenn

die Bedingung **nicht** zutrifft. **Bitte beachten Sie:**

Das Wort **NICHT** muss vor der eigentlichen Bedingung stehen, es darf nicht in der Bedingung stehen.

Beispiel:

**Falsch** wäre die umgangssprachliche Formulierung:

wenn LichtBad **NICHT** ausgeschaltet oder Tag **NICHT**="Montag" dann  
**es würde ein Syntaxfehler angezeigt.**

**Richtig ist:**

wenn **NICHT** LichtBad ausgeschaltet oder **NICHT** Tag="Montag" dann

**Bitte beachten Sie:**

Häufige Syntaxfehler in wenn-Anweisungen sind, dass das Wort **dann** vergessen wird und dass die Anweisung nicht mit einem **endewenn** abgeschlossen wird.

Da wenn-Anweisungen sich immer über mehrere Zeilen erstrecken, kann die Zeile für einen Syntaxfehler oft

nicht bestimmt werden. Wenn in einem Makro mit einer wenn-Anweisung ein Syntaxfehler ohne Fehlerbeschreibung auftritt, prüfen Sie alle Elemente der wenn-Anweisung um den Fehler zu finden.

Beispiel:

```
WENN Temperatur < 21,5 DANN
```

Wenn die Operanden unterschiedlichen Typs sind, findet soweit möglich eine automatische Konvertierung statt.

Wenn einer der Operanden eine Konstante ist, muss diese als *Operand2* stehen, damit eine korrekte Konvertierung durchgeführt werden kann.

Beispiel:

**Richtig:** wenn Uhrzeit = "15:15:00" dann

**Falsch:** wenn "15:15:00" = Uhrzeit dann

## Besondere Vergleichsbedingungen

wenn Objekt **eingeschaltet** hat die gleiche Wirkung wie

wenn Objekt = 1 bzw. wenn Objekt = **"an"**

wenn Objekt **ausgeschaltet** hat die gleiche Wirkung wie

wenn Objekt = 0 bzw. wenn Objekt = **"aus"**

## Schlüsselwörter für Bedingungen

Für Schalter: **eingeschaltet, ausgeschaltet**

Für Türen und Fenster: **geoeffnet, geschlossen**

Beispiel:

wenn Fenster **geoeffnet** dann

...

### Zeit-Vergleiche

Bei Zeitvergleichen mit Uhrzeiten ist zu beachten, dass die Uhrzeit im 5-Sekunden-Takt aktualisiert wird, die Sekunden der Uhrzeit bei der Prüfung auf Zeitgleichheit also immer durch 5 teilbar sein müssen.

wenn **Uhrzeit** = "**HH:MM:SS**" dann

bei dieser Bedingung ist auch zu beachten, dass die Uhrzeit in Hochkommas gesetzt wird

wenn **Datum** = "**TT.MM.JJJJ**" dann

bei dieser Bedingung ist zu beachten, dass das Datum in Hochkommas gesetzt wird und die Jahreszahl 4-stellig angegeben wird.

Beispiel:

wenn **Datum** = "24.12.2023" dann

wenn **Wochentag** = "Sonntag" dann

Prüfung auf einen Wochentag. Der Wochentag wird in Hochkommas gesetzt und muss mit einem Großbuchstaben und nachfolgenden Kleinbuchstaben geschrieben werden.

wenn **Monatstag** = 29 dann

Prüfung auf den Tag eines Datums.

wenn **Monat** > 4 und **Monat** < 10 dann

bewirkt, dass die folgenden Anweisungen nur zwischen Mai und September ausgeführt werden.

### Vergleichsoperator =\* für Vergleiche mit Jokerzeichen

Beim Vergleich von Uhrzeit und Datum mit einer Konstanten können auch Jokerzeichen verwendet werden.

Als Vergleichsoperator muss =\* verwendet werden. Operand2 muss eine Konstante in Hochkommas sein.

Beispiele:

Um Anweisungen jede volle und halbe Stunde auszuführen:

wenn **Uhrzeit** =\* "\*\*\*:00:00" oder **Uhrzeit** =\* "\*\*\*:30:00" dann

.....

Um Anweisungen immer am Ersten eines Monats auszuführen:

wenn **Datum** =\* "01.\*\*.\*\*\*\*\*" dann

.....

### Bedingung für Zeiträume

Manchmal ist es erforderlich Aktionen nur innerhalb bestimmter Zeiträume auszuführen.

Dies ist möglich mit der Bedingungsanweisung :

wenn **Uhrzeit/Datum zwischen** "**Zeitkonstante**" und "**Zeitkonstante**" dann

.....

Beispiel:

wenn **Uhrzeit zwischen** "23:00:00" und "05:00:00" dann

.....

### Vergleichsoperator =+ für Vergleiche mit Wochenmaske

Mit diesem Vergleichsoperator ist es möglich, Anweisungen nur an bestimmten Wochentagen ausführen zu

lassen. Als Operand2 muss eine 7-stellige Konstante bestehend aus Nullen und Einsen in Hochkommas

verwendet werden. Jede Stelle der Konstanten steht für einen Wochentag, beginnend mit Sonntag. Die

Bedingung ist wahr wenn an der Stelle des aktuellen Wochentags eine Eins steht.

Beispiel: Es soll geprüft werden, ob der aktuelle Tag ein Freitag, Samstag oder Sonntag ist.

wenn **Wochentag** =+ "1000011" dann

## 1.13. Hauptfenster

---

### Hauptfenster

Das Hauptfenster des Programms hat neben der Toolbar nur eine Statuszeile, in der wichtige Systemmeldungen angezeigt werden.

Von diesem Hauptfenster aus werden alle anderen Fenster aufgerufen.

Auf der Seite [Allgemein](#) im Einstellungsfenster können Optionen zum Verhalten des Hauptfensters festgelegt werden.

Unter anderem ist es möglich, das Hauptfenster immer im Vordergrund vor allen anderen Fenstern anzeigen zu lassen.

Wenn grössere Fenster der Anwendung geöffnet werden und das Hauptfenster diese verdecken würde oder diese Fenster das Hauptfenster verdecken würden, wird das Hauptfenster automatisch am oberen oder unteren Rand dieser Fenster positioniert.

Mit dem "Blitz"-Button oder dem Menüpunkt *Compilieren und übertragen* wird der Ausführungscode für die Zentrale erstellt und zur Zentrale übertragen. Vor der ersten Übertragung auf die Zentrale muss die Verbindung zur Zentrale konfiguriert und eine Konfigurationsdatei zur Zentrale übertragen werden. Das geschieht mit dem Menüpunkt *Konfigurieren->Assistent Zentralenverbindung*.

In Programmfenster ExecEngineWin kann im Tab SSH-Übertragung eine SSH-Verbindung zur Zentrale für die Übertragung der Ausführungsdatei festgelegt werden.

Das ist die bevorzugte Übertragungsmethode, die sehr viel schneller ist und insbesondere bei grösseren Dateien von Vorteil ist.

Um die SSH-Übertragung benutzen zu können muss auf der CCU im Fenster Sicherheit ssh für die CCU erlaubt werden. Das Kennwort, was dort festgelegt wird muss im Tab SSH-Übertragung des Fensters EXECEngineWin angegeben werden.

## 1.14. Visualisierungsfenster

---

### Visualisierungsfenster

Die Visualisierung wird vom Programm [VisuWin](#) bereitgestellt. Jede definierte Ansicht hat ein eigenes Visualisierungsfenster, in dem die Zustände und Werte der einzelnen Objekte angezeigt werden.

Mit der rechten Maustaste kann ein Pop-Up-Menü aufgerufen werden, mit dem in den Vollbildschirm-Modus geschaltet werden kann, dabei wird die aktuelle Ansicht auf dem ganzen Bildschirm ohne Menüs und Fensterrahmen angezeigt.

**Dieser Modus kann mit der ESC-Taste beendet werden.**

# 1.15. Das Fenster Modulauswahl

---

## Das Fenster Modulauswahl

Im linken Teil des Fensters wird eine Liste aller verfügbaren Module angezeigt. Im Auswahlfeld links unter der Liste kann der Typ der anzuzeigenden Module ausgewählt werden.

### **Hinweise zum HomeMatic-System:**

Bevor Module aus der Zentrale bzw. vom BidCoS-Service übernommen werden können, muss die IP-Adresse der Zentrale im [Fenster HomeMatic-Zentrale](#) angegeben werden.

Falls ein Modul benutzt werden soll, das noch nicht in der Zentrale angelernt worden ist so kann es im linken Listfenster markiert werden und mit dem Pfeil-Knopf in das rechte Listfenster *verwendete Module* kopiert werden. In diesem Fall muss später die Adresse des Moduls später manuell eingesetzt werden.

Mit dem Knopf **[Module importieren]** werden alle angelernten Module aus einer CCU übernommen. Module, die schon verwendet werden, bleiben erhalten. Die Erkennung ob Module schon existieren wird über die Adresse der Module durchgeführt. Bei schon vorhandenen Modulen werden keine Änderungen vorgenommen.

Die Namen der Module werden aus der CCU übernommen, Sonderzeichen (ÄÖÜ usw.) werden leider von der CCU nicht korrekt übermittelt und daher durch ein grosses X ersetzt. Die Objektnamen und -bezeichnungen für die einzelnen Kanäle werden ebenfalls aus der CCU übernommen. Da in Objektnamen Sonderzeichen, Punkte, Kommas, Minuszeichen usw. nicht verwendet werden dürfen, werden die Namen ggfs. entsprechend konvertiert. Für Umlaute gilt das gleiche wie beim Modulnamen, diese werden durch ein grosses X ersetzt.

### **Bitte bei der CLX-Version unbedingt beachten:**

Die virtuellen Kanäle der CCUs dürfen nur von einer CCU importiert werden, da die Namen dieser Kanäle auf der CCU nicht geändert werden können und bei allen CCUs gleich sind und somit jeweils überschrieben werden.!

Im Normalfall spielt das keine Rolle, da die virtuellen Kanäle der CCUs fast ausschliesslich für Programmierungen innerhalb der Web-UI benutzt werden.

### **Hinweise bei Benutzung von FHZ-Schnittstellen:**

Die anzulegenden Module werden im linken Listfenster markiert und mit dem Button mit dem roten Pfeil in das rechte Listfenster der verwendeten Module übertragen. Die Adresse der Module muss dann noch im jeweiligen Verwaltungsfenster der Module eingestellt werden. Bei Sensoren wird das normalerweise durch Empfang von den Sensoren gemacht, bei Aktoren durch Senden der Adresse an da in den Lern-Modus versetzte Modul. Falls die Adressen bekannt sind, können diese im Verwaltungsfenster der Module auch direkt eingetragen werden.

**Es werden nicht alle verfügbaren FS20-Module im linken Listfenster aufgeführt.** Dort sind nur die am häufigsten verwendeten Module vorhanden. Falls FS20-Module verwendet werden, die nicht aufgeführt sind, müssen dafür die Standard-Module "FS20 Empfänger" und "FS20 Sender" benutzt werden. Diese Standard-Module stehen mit unterschiedlicher Kanalanzahl zur Verfügung. Es sollten jeweils die Standard-Module ausgewählt werden, die die gleiche Anzahl oder mehr Kanäle wie das verwendete Modul haben. Es ist auch möglich für ein Hardware-Modul mehrere Standard-Module der Liste zu verwenden, es muss dann allerdings darauf geachtet werden, dass einzelne Kanäle nicht doppelt angelegt werden.

Ein Modul kann mit dem Knopf **[Modul löschen]** gelöscht werden, das Löschen mit der Entfernen-Taste ist aus Sicherheitsgründen nicht möglich. Bevor ein Modul tatsächlich gelöscht wird, erscheint eine Bestätigungsabfrage.

Wenn ein Modul in Makros verwendet wird, kann es nicht einfach gelöscht werden, da das Makro dann fehlerhaft wäre. Es erscheint dann ein Fenster, in dem angezeigt wird in welchen Makros das Modul verwendet wird. In diesem Fenster erscheint auch ein Button, mit dem das Löschen des Moduls erzwungen werden kann obwohl es Bezüge darauf gibt.

Mit dem Knopf **[Bearbeiten]** oder durch Doppelklick auf ein Modul im rechten Listenfenster wird das Definitionsfenster für das markierte Modul geöffnet.

Die Art des Fenster ist abhängig vom Typ des Moduls:

[Aktoren](#) zur Bearbeitung von Aktoren (Empfängern)



[Schaltsensoren](#) zur Bearbeitung von Schaltsensoren (Sendern)  
[Sensoren](#) zur Bearbeitung von Sensoren, die Zahlenwerte senden.

## 1.15.1. Das Fenster Schaltsensoren

---

### Das Fenster Schaltsensoren

Im Fenster Schaltsensoren werden alle erforderlichen Informationen für einen Schaltsensor eingestellt.

Im rechten oberen Teil können Name und Bezeichnung für das [Objekt](#) verändert werden.

Der Name wird bei Definitionen und in Makros verwendet, die Bezeichnung wird z.B. in [Visualisierungen](#) angezeigt.

Mit dem Knopf **[Visualisierung]** wird die [Seite Visualisierung der Objekdefinition](#) aufgerufen.

Für jede Taste eines Senders (also jeden Kanal) kann ein separates Makro erstellt oder aus der Liste vorhandener Makros ausgewählt werden.

Durch Klick auf den Knopf **[Makro bearbeiten]** wird das [Fenster Makro bearbeiten](#) geöffnet, in dem Sie die einzelnen Makroanweisungen erstellen und editieren können.

Durch Klick auf den Knopf **[Makro auswählen]** wird ein Fenster geöffnet, in dem eine Liste aller vorhandenen Makros angezeigt wird. Es erscheinen auch Makros, für die noch keine Anweisungen hinterlegt wurden, damit können auch Makros ausgewählt, die später erst erstellt werden sollen.

Wenn sie hier ein Makro auswählen, so wird dieses ausgeführt, wenn während der [Ausführung](#) die zugehörige Taste betätigt wird. Diese Methode ermöglicht es vorhandene Makros mehrfach zu nutzen ohne die Anweisungen mehrmals eingeben zu müssen. Tatsächlich wird in diesem Fall in das Makro der Taste nur eine Anweisung mit dem Aufruf des ausgewählten Makros erstellt. Wenn Sie mit dem Knopf **[Makro bearbeiten]** in das Makro gehen, sehen Sie diese Anweisung und können bei Bedarf weitere Anweisungen zufügen.

Mit dem Knopf **[Makro löschen]** können Sie das Makro für die aktuelle Taste löschen, wenn das Makro mehrere Anweisungen hat, erscheint vor dem Löschen eine Bestätigungsabfrage.

Bitte beachten Sie die Besonderheiten bei [Zustandsmeldern](#).

#### Bitte beachten Sie bei HomeMatic-Modulen:

Mit dem Knopf **[Adresse empfangen]** können die Adressen einzelner Tasten von HomeMatic-Modulen überprüft werden.

Gehen Sie dazu auf die Zeile einer Taste und klicken dann auf den Knopf **[Adresse empfangen]**. Es erscheint ein Hinweisfenster. Betätigen Sie jetzt die entsprechende Taste der Fernbedienung bzw. des Schaltsensors. Wenn die empfangene Adresse identisch mit der aktuell ausgewählten Zeile der Taste ist, ertönt ein Bestätigungston und das "Warte-Fenster" verschwindet. Ist die empfangene Adresse nicht identisch erscheint eine Abfrage ob die empfangene Adresse in die aktuelle Zeile eingesetzt werden soll. Beantworten Sie die Abfrage nur mit Ja, wenn Sie sicher sind, dass die Adressen ausgetauscht werden sollen, normalerweise ist das nicht erforderlich oder sinnvoll. Es kann jedoch nötig werden, wenn eine z.B. eine Fernbedienung gegen eine andere getauscht werden soll um nicht alle Definitionen neu eingeben zu müssen.

#### Bitte bei HomeMatic-Funkmodulen unbedingt beachten:

Die meisten Fernbedienungen und Funk-Schaltsensoren müssen in der Zentrale registriert werden, bevor sie benutzt werden können. Dabei werden entweder einzelne Tasten oder das komplette Modul registriert, erst dann werden die Meldungen einzelner Tasten verarbeitet bzw. stehen im Display einer Fernbedienung zur Verfügung. Eine Registrierung ist immer dann erforderlich wenn der Registrierungsknopf **[Taste registrieren]** oder **[Modul registrieren]** im Fenster angezeigt wird.

Wenn dieser Knopf nicht aktiviert ist, schliessen Sie dieses Fenster und stellen die Verbindung zur Zentrale her. Infos zur Zentralenverbindung finden Sie im [Fenster Hardware](#) auf der Seite "Zentrale".

Klicken Sie dann auf den Registrierungsknopf und befolgen die Anweisungen am Bildschirm. Je nach Modul und Konfigurationsstatus muss vor der Registrierung einer Taste die Konfigurationstaste gedrückt werden.

Für leitungsgebundene Sensoren (HMW...-Module) ist eine Registrierung nicht erforderlich.

#### Bitte beachten Sie bei FS20-Modulen:

Mit dem Knopf **[Adresse empfangen]** wird der Empfangsmodus zum Einstellen der Adressen aktiviert. Wenn Sie nun eine Taste drücken, wird die empfangene Adresse in für die ausgewählte Taste eingetragen. Wenn aktuell die erste Taste ausgewählt ist und es sind noch keine Adressen

eingetragen wird automatisch immer eine Zeile weitergesprungen bis die letzte Taste erreicht ist.

## 1.15.2. Das Fenster Aktoren

---

### Das Fenster Aktoren

Im Fenster Aktoren werden alle erforderlichen Informationen zur Steuerung eines Aktors (Empfängers) hinterlegt.

Im rechten oberen Teil können Name und Bezeichnung für das [Objekt](#) verändert werden.

Der Name wird bei Definitionen und in Makros verwendet, die Bezeichnung wird z.B. in [Visualisierungen](#) angezeigt.

Mit dem Knopf **[Visualisierung]** wird die [Seite Visualisierung der Objekdefinition](#) aufgerufen.

**Bitte beachten Sie:**

Durch die Auswahl eines Bild-Symbols kann sich auch der „Typ“ des Objekts ändern. Dadurch werden auch die vorhandenen Visualisierungsdarstellung für alle Ansichten verändert. Ändern Sie das Bildsymbol also bitte nur wenn Sie wollen, dass sich alle vorhandenen Darstellungen für das Objekt in allen Ansichten ändern wollen. Wollen Sie nur einzelne Darstellungen ändern, so können Sie das über die Auswahl eines anderen Bitmaps tun.

In der Tabelle im rechten Teil des Fensters können Schaltzeiten für das Objekt festgelegt werden. Dabei ist es auch möglich den Zeitpunkt des Sonnenaufgangs oder Sonnenuntergangs für das aktuelle Datum auszuwählen. Dies geschieht mit einem Pop-Up-Menü, das durch Klick auf die rechte Maustaste geöffnet wird wenn der Cursor sich im Zeiteingabefeld befindet. Es kann auch direkt **SA** für Sonnenaufgang oder **SU** für Sonnenuntergang in die Stundenstellen des Zeiteingabefensters eingetragen werden.

**Bitte beachten Sie:**

*Alle Zeitbearbeitungen erfolgen im 5-Sekunden-Intervall, bei Zeitangaben müssen die Sekunden immer durch 5 teilbar sein (xx:xx:00, xx:xx:05, xx:xx:10 ... xx:xx:55).*

Bitte beachten Sie folgende Hinweise:

[Besonderheiten bei Dimmern](#)

## 1.15.3. Das Fenster Sensoren

---

### Das Fenster Sensoren

Im Fenster Sensoren werden alle Sensoren verwaltet, die Zahlenwerte wie Temperatur, Luftfeuchtigkeit, Helligkeit usw. schicken.

Im rechten oberen Teil können Name und Bezeichnung für das [Objekt](#) verändert werden. Der Name wird bei Definitionen und in Makros verwendet, die Bezeichnung wird z.B. in [Visualisierungen](#) am Bildschirm angezeigt.

Mit dem Knopf **[Visualisierung]** wird die [Seite Visualisierung der Objekdefinition](#) aufgerufen. Im rechten Teil des Fensters können Makros für verschiedene Werte-Bereiche des Sensors definiert werden. Geben Sie in den beiden Eingabefeldern oberhalb des Knopfes **[Neuer Bereich]** den gewünschten Wertebereich an und klicken dann auf den Knopf **[Neuer Bereich]**. Das Fenster Makro bearbeiten wird geöffnet und Sie können die Anweisungen eingeben, die ausgeführt werden sollen, wenn der Sensor einen Wert sendet, der innerhalb des Bereichs liegt. Die Bereiche können sich beliebig überschneiden, es werden dann alle Teil-Makros ausgeführt. Das bietet eine Vielzahl von Möglichkeiten, **beachten Sie aber dabei:**

*Wenn beispielsweise der Bereich von 10,0 bis 20,0 für ein Teilmakro und der Bereich von 20,0 bis 30,0 für ein anderes Teilmakro angegeben wird werden beide Teilmakros ausgeführt (weil der Wert 20,0 für beide gültig ist). Soll nur ein Makro ausgeführt werden, muss in diesem Fall z.B. der zweite Bereich bei 20,1 beginnen.*

Mit dem Knopf **[Bereich bearbeiten]** können vorhandene Teilmakros zu den verschiedenen Bereichen bearbeitet werden. Wählen Sie in der Tabelle die Zeile aus und klicken Sie auf den Knopf **[Bereich bearbeiten]**. Einfacher ist ein Doppelklick auf die Zeile, auch dann wird das Fenster zur Bearbeitung des Teilmakros geöffnet.

Mit dem Knopf **[Bereich löschen]** wird die ausgewählte Bereichszeile inklusive der Makroanweisungen gelöscht. Die ausgewählte Zeile ist die farblich hinterlegte Zeile.

Bitte beachten Sie die Besonderheiten bei [Zustandsmeldern](#).

## 1.16. Einstellungen

---

In diesem Fenster können auf mehreren Seiten Einstellungen vorgenommen werden.

**Bitte beachten Sie:**

Je nach Seite und Art der Daten werden die neuen Werte erst nach dem Schliessen des Fensters übernommen. Schliessen Sie also dieses Fenster, nachdem Sie Änderungen vorgenommen haben um diese zu übernehmen.

## 1.16.1. Seite Ansichten

---

### Seite Ansichten

Auf der Seite werden unterschiedliche Ansichten zur Visualisierung eingerichtet.

Für jede Ansicht kann optional ein [Bitmap](#) als Hintergrundbild festgelegt werden. Wenn für eine Ansicht ein Hintergrundbild festgelegt wurde, werden die [Objekte](#) als Bild-Symbol dargestellt – um welches Objekt es sich handelt, geht dann aus der Positionierung im Hintergrundbild hervor. Wenn kein Hintergrundbild ausgewählt wird, werden die Objekte je nach Typ und Definition als [Objektrahmen oder Textrahmen](#) dargestellt.

Für jede Ansicht kann mit dem Knopf **[Hintergrundfarbe]** eine Hintergrundfarbe ausgewählt werden. Diese Farbe füllt das ganze Fenster aus, mit Ausnahme des Bereichs, in dem das Hintergrundbild steht.

Es gibt Versionen bei denen die Anzahl der möglichen individuellen Ansichten beschränkt ist.

## 1.16.2. Seite Objekte

---

### Seite Objekte

Auf dieser Seite wird eine Übersicht über alle [Objekte](#) angezeigt.

Durch Anklicken der Schaltfläche **[Modulauswahl]** wird das Fenster zur Modulauswahl aufgerufen. Durch Betätigung des Knopfes **[Bezüge anzeigen]** wird angezeigt, in welchen Makros das ausgewählte Objekt verwendet wird. Diese Funktion ist in komplexen Projekten wichtig um Zuordnungen schnell erkennen zu können.

Mit den weiteren Schaltflächen werden die Objekte verwaltet. Das Löschen ist nur für Objekte möglich, die nicht mit einem Hardwaremodul verbunden sind.

Mit dem Button **[Typdefinitionen]** wird ein Fenster geöffnet, mit dem selbst definierte Typen und Zustände erstellt und verwaltet werden können.

Mit dem Button **[Spezielle Objekte]** wird ein Fenster geöffnet, in dem Objekte für die Bearbeitung von Batteriewarnungen und Kommunikationsstörungen festgelegt werden können.



## 1.16.3. Seite Makros

---

### Seite Makros

Auf dieser Seite können [Makros](#) unabhängig von Hardwaremodulen definiert werden. Diese können zu bestimmten Ausführungszeiten gestartet werden, aber auch durch andere Makros aufgerufen werden.

Makros beinhalten [Makroanweisungen](#) mit denen die jeweils gewünschten individuellen Anwendungsfunktionen erstellt werden.

Mit dem Button **[Ausführungszeiten]** wird ein Fenster geöffnet, in dem Makros zu bestimmten Zeiten aktiviert werden können.

**Bitte beachten Sie:**

*Alle Zeitbearbeitungen erfolgen im 5-Sekunden-Rhythmus, bei Zeitangaben müssen die Sekunden immer durch 5 teilbar sein (xx:xx:00, xx:xx:05, xx:xx:10 ... xx:xx:55).*

## 1.16.4. Seite Allgemein

---

### Seite Allgemein

Auf dieser Seite können einige allgemeine Einstellungen vorgenommen werden.

Es gibt folgenden Optionen, die über Checkboxes ausgewählt werden können:

#### *Hinweise am Mauszeiger anzeigen*

Es werden erläuternde Hinweise zu vielen Bildelementen angezeigt, wenn der Mauszeiger auf diesen Elementen steht.

#### *Hauptfenster immer im Vordergrund vor allen Fenstern*

Das Hauptfenster bleibt immer im Vordergrund, auch vor Fenstern anderer Anwendungen.

#### *Automatische Sicherungen ...*

Das aktuelle Projekt wird im angegebenen Zeitabstand gesichert, nachdem die Anzahl der angegebenen Sicherungen erreicht ist wird die erste Sicherung wieder überschrieben. Bei diesen Sicherungsdateien wird hinter den Namen des Projekts ein Unterstriche Doppelkreuz und laufende Nummer eingefügt, der Dateiname ist dann z.B. :*MeinProjekt\_#3.SPG*.

#### *Letztes Projekt bei Start automatisch öffnen*

Das zuletzt bearbeitete Projekt wird bei Programmstart automatisch geöffnet. Wenn mit mehreren Projekten und verschiedenen Netzwerklauferwerken gearbeitet wird kann diese Option deaktiviert werden um zu vermeiden, dass es beim Programmstart zu Timeout-Wartezeiten durch nicht vorhandene Netzwerkverbindungen kommt.

#### *Im PC-Modus Ausführung bei Programmstart aktivieren*

Wenn der PC als Zentrale benutzt wird, wird die Ausführung bei Programmstart automatisch aktiviert.

#### *Manuelle Änderungen in Menü-Makros zulassen*

Makros, die menügeführt erstellt wurden können über den Texteditor geändert werden wenn diese Option aktiviert ist.

#### *Historydaten schreiben*

Diese Option schaltet das Schreiben von Historydaten generell ein- oder aus. Die Historydaten beinhalten Datum, Uhrzeit und den neuen Zustand/Wert eines Objekts. In der Objektdefinition muss ebenfalls die Option *Historydaten schreiben* aktiviert sein. Die Historydatei heisst *history.hst*. In welches Verzeichnis die Datei geschrieben wird ist vom Typ der Zentrale abhängig. Das Verzeichnis wird im Reiter *Zentrale* des Hardwarefensters eingestellt. Das Standardverzeichnis wird jeweils bei Auswahl der Zentralentyps voreingestellt und kann geändert werden. Bei Benutzung einer externen Zentrale sollte dieses Verzeichnis nur geändert werden, wenn bei einer CCU2 auf einen USB-Stick statt auf die SD-Karte geschrieben werden soll.

#### *Systemlog schreiben*

Mit dieser Option wird das Schreiben von Systemmeldungen in eine Textdatei ein- oder ausgeschaltet. Diese Datei heisst *systemlog.txt*. In welches Verzeichnis die Datei geschrieben wird ist vom Typ der Zentrale abhängig. Beschreibungen dazu stehen im vorigen Abschnitt *Historydaten schreiben*. Bitte beachten Sie die dort beschriebenen Hinweise. Welche Systemmeldungen geschrieben werden kann über das in den [speziellen Objekten](#) unter *Einstellung für Systemlog* angegebene Objekt bestimmt werden.

#### *Allgemeine Einstellung in Projektdatei speichern*

Bei Aktivierung dieser Option werden generelle Einstellungen wie z.B. die IP-Adresse der Zentrale nicht in der Registry des PCs, sondern in der Projektdatei gespeichert. So ist es z.B. einfach möglich mit einem PC mehrere Installationen verwalten zu können ohne bei Wechsel auf eine andere Installation die individuelle Umgebung neu einrichten zu müssen.

#### *Aktuelle Werte der ExecEngine speichern/laden*

Bei Beendigung der ExecEngine werden alle aktuellen Werte der Objekte und Variablen (also auch

die Zustände/Werte von Hardwaremodulen) gespeichert und beim nächsten Start der ExecEngine wieder eingelesen. Ein Neustart wird bei dieser Option erheblich schneller ausgeführt, da vor dem Start des normalen Betriebsmodus nicht gewartet wird bis alle aktuellen Werte der Hardwaremodule abgefragt wurden. Zudem bleiben die Werte der Module die nicht abgefragt werden können (Batteriesensoren wie z.B. Tür/Fensterkontakte) erhalten. Es werden aber trotzdem im Hintergrund alle Module für die eine Zustandsabfrage möglich ist abgefragt während die ExecEngine schon im normalen Betrieb läuft - wenn die Hardwareabfrage nicht durch die folgenden Option deaktiviert wird. Unter welchen Voraussetzungen eine Abfrage der Hardware erfolgt, wird in der folgenden Option *Keine Hardwareabfrage bei Neustart* festgelegt.

#### *Keine Hardwareabfrage bei Neustart innerhalb...*

Die Abfrage der Hardware beim Start kann den [Duty-Cycle](#) stark belasten. Daher kann es sinnvoll sein, die Hardwareabfrage zu deaktivieren, insbesondere wenn die vorige Option *Aktuelle Werte der ExecEngine speichern/laden* aktiviert ist und die Zustände bei Beendigung in einer Datei gespeichert wurden.

Wenn die Option *Aktuelle Werte der ExecEngine speichern/laden* aktiviert ist erscheint ein zusätzliches Eingabefeld, in dem die Zeit in Sekunden angegeben werden kann, die nach der Erstellung der Datei mit den aktuellen Zuständen und Werten vergehen darf ohne dass eine Hardwareabfrage durchgeführt wird.

Ist diese Zeit überschritten oder gibt es keine Datei weil die ExecEngine nicht normal beendet wurde sondern es z.B. einen Stromausfall gab wird eine Hardwareabfrage durchgeführt.

Wenn keinesfalls eine Hardwareanfrage durchgeführt werden soll, auch nicht es keine aktuelle Datei gibt, kann als Zeit 0 oder 1 Sekunde angegeben werden.

Bitte beachten Sie, dass es zu Fehlfunktionen kommen kann wenn keine Hardwareabfrage durchgeführt wird und es Funktionen gibt, bei denen Aktionen vom aktuellen Zustand der Hardware abhängig sind.

Im Auswahlfenster *Geografischer Ort zur Berechnung Sonnenzeiten* wird je nach Land die Postleitzahl, nächstgelegene Stadt oder Region aus der zur Verfügung stehenden Liste ausgewählt. Aufgrund dieser Ortsangabe werden jeweils die tagesaktuellen Zeiten für [Sonnenaufgang und Sonnenuntergang](#) ermittelt.

Mit dem Button **[weitere Einstellungen]** wird das Fenster [Einstellungen zu Sonnenzeiten](#) geöffnet, in dem detaillierte Angaben zum geografischen Standort gemacht werden können. Dort können auch Grenzwerte und Veränderungen der Sonnenzeiten festgelegt werden.

Mit dem Button **[Anwesenheitssimulation]** wird das Fenster zur Verwaltung der Daten für die [Anwesenheitssimulation](#) geöffnet.

Mit dem Button **[Feiertage]** wird das Fenster zur Verwaltung der Feiertage und freien Tage geöffnet. Im Fenster zur Verwaltung der Feiertage kann ausgewählt werden welche Feiertage und welche frei definierbaren Tage und Zeiträume in der Zeittabelle wie Sonntage behandelt werden sollen.

Mit dem Button **[e-mail-Einstellungen]** wird das [Fenster der e-mail-Konfiguration](#) geöffnet. Weitere Infos zur Verwaltung eigener Dateien auf der Zentrale und zur Einbindung von USB-Stick und SD-Karte finden Sie unter dem Hilfeindex zu SCHREIBEDATEI.

## 1.16.4.1. Fenster Sonnenzeiten

---

### Fenster Sonnenzeiten

In diesem Fenster können individuelle Anpassungen bezüglich der Zeiten zum [Sonnenaufgang und Sonnenuntergang](#) gemacht werden.

Da je nach Jahreszeit die Zeiten für Sonnenaufgang und Sonnenuntergang sehr früh bzw sehr spät sein können, kann es sinnvoll sein diese Werte zu begrenzen. Es wird dann bei Benutzung dieser Funktionen der angegebene Grenzwert nicht unter- bzw. überschritten.

Weiterhin können individuelle Anpassungen bezüglich der Zeitpunkte festgelegt werden. Dabei werden die Minuten angegeben, um die der berechnete Zeitpunkt verändert wird.

## 1.16.4.2. Fenster Feiertage

---

### Fenster Feiertage

Dieses Fenster wird mit dem Button *Feiertage* im Reiter *Allgemein* des Einstellungsfensters geöffnet. Im ersten Reiter des Fensters kann ausgewählt werden welche der dort angezeigten Feiertage in der Zeittabelle wie Sonntage behandelt werden sollen, im zweiten Reiter können einzelne Tage und Zeiträume angegeben werden, die in der Zeittabelle wie Sonntage behandelt werden sollen. Im zweiten Reiter können auch regionale Feiertage angegeben werden, die im ersten Reiter nicht aufgeführt sind.

So ist es einfach möglich Tage, an denen man zuhause ist, bezüglich der Zeittabellen wie Sonntage zu behandeln.

## 1.16.4.3. Fenster e-mail Konfiguration

---

### Fenster e-mail Konfiguration

In diesem Fenster werden die Einstellungen zum e-mail-Versand festgelegt..

Im Eingabefeld "SMTP-Server" wird der Name des Posteingangsserver Ihres Providers eingetragen. Es muss sich dabei um einen SMTP-Server handeln.

Wenn Sie den Namen des SMTP-Servers Ihres Providers nicht kennen, können Sie diesen beim Provider erfragen, einige gebräuchliche Provider und zugehörige SMTP-Server sind am Ende dieser Seite aufgeführt und normalerweise auf den Internetseiten des Providers aufgeführt

Im Eingabefeld "Port" wird der benutzte Port des SMTP-Servers eingetragen, für SSL-Verschlüsselung ist das üblicherweise 465, für TLS üblicherweise 587. Die Ports können jedoch je nach Provider auch anders festgelegt sein.

Im Eingabefeld "e-mail-Adresse" wird die Adresse eingetragen, die als Absender der e-mail benutzt werden soll. Bei manchen Providern muss das zwingend exakt die Adresse sein, unter der man als Benutzer eingetragen ist - andernfalls wird die mail nicht versendet.

Im Eingabefeld "Benutzer" wird der Benutzername eingetragen, im Feld "Kennwort" das Kennwort. Bezüglich der Einstellungen können Sie alle Werte auch in Ihrem normalen e-mail-Programm nachsehen.

Im Zweifelsfall oder wenn eine Verbindung nicht funktioniert fragen Sie bei Ihrem Provider nach den erforderlichen Einstellungen und Zugangsdaten.

Gmail von Google kann seit Mai 2022 nicht mehr als Mailserver benutzt werden, da eine zweistufige Authentifizierung gefordert ist und nur wenige Apps ohne diese Authentifizierungsart von Google als sicher eingestuft werden.

Hier einige Provider mit Servernamen und Ports, ohne Gewähr für Richtigkeit, da diese Zugangsdaten vom Provider geändert werden können:

t-online

SSL : securesmtp.t-online.de / Port 465

TLS : securesmtp.t-online.de /Port 587

1und1

SSL : smtp.1und1.de / Port 465

TLS : smtp.1und1.de / Port 587

web.de

TLS : smtp.web.de / Port 587

gmx.net

SSL : mail.gmx.net / Port 465

TLS : mail.gmx.net / Port 587

## 1.16.5. Seite Verzeichnisse

---

### Seite Verzeichnisse

Auf dieser Seite werden die benutzten Standardverzeichnisse für Bilddateien und die eigenen Projektdateien eingestellt.

Diese Verzeichnisse werden vom Programm automatisch benutzt.

Es können zwar im Einzelfall auch Dateien aus anderen Verzeichnissen ausgewählt werden, das sollte aber vermieden werden, da diese Dateien dann in Apps und anderen

Visualisierungsprogrammen, die mit den Bilddateien arbeiten nicht zugegriffen werden kann wenn diese in anderen als dem Standardverzeichnis abgespeichert sind.

## 1.16.6. Seite Sicherheit

---

### Seite Sicherheit

In diesem Fenster werden verschiedene Zugangsdaten verwaltet.

#### **Code für den SMS-Versand**

Hier wird der Code eingetragen, den Sie für Ihr SMS-Konto erhalten haben.

Mit der Anweisung SENDESMS können SMS nur versendet werden, wenn hier ein Code eingetragen wurde. Für jede SMS wird ein Credit vom SMS-Konto abgebucht.

SMS-Credits können im Online-Shop von CL-control erworben werden..

#### **Zugangsdaten Apps**

Wenn eine App (Android/iPhone/iPad) verwendet wird muss hier ein Benutzername und ein Kennwort festgelegt werden.

Die App kann erst verwendet werden nachdem das Projekt neu auf die Zentrale übertragen wurde.

#### **Zugangsdaten für den VPN-Service**

Hier werden die Zugangsdaten für den VPN-Service von CL-control eingetragen.

Mit dem VPN-Service kann eine sichere Verbindung zur Zentrale über das Internet hergestellt werden, ohne DYNDNS-Adressen und unsichere Portfreigaben im Router benutzen zu müssen.

Die Android-App und die iPhone/iPad-Apps und der CL-Webserver unterstützen den Zugriff über den VPN-Service, er kann auch benutzt werden um über einen Browser über das Internet auf die WEB-UI der CCU zuzugreifen.

Die Datenverbindung über den VPN-Service ist eine sichere verschlüsselte und über Zertifikate geschützte Verbindung. Der VPN-Service stellt nur die Verbindung zur Zentrale her und speichert keinerlei Daten.

#### **Benutzer VisuWin**

Bei Anklicken dieses Knopfes wird ein Fenster geöffnet, in dem mehrere VisuWin-Benutzer angelegt und verwaltet werden können.

Dabei ist es möglich die Seiten auf die zugegriffen werden kann den Benutzern einzeln zuzuweisen.



## 1.16.7. Seite PC-Modus

---

### Seite PC-Modus

Auf dieser Seite werden spezielle Einstellungen vorgenommen, die nur benutzt werden wenn der PC als Zentrale benutzt wird.

# 1.17. Fenster Hardware

---

## Fenster Hardware

### Tab Zentrale

Auf dieser Seite werden Einstellungen zur Verbindung mit der Zentrale verwaltet. Je nach Programmversion können verschiedene Zentralentypen ausgewählt werden. Bei Auswahl eines Zentralentyps werden automatisch die Verzeichnisse eingetragen, die auf der Zentrale zur Speicherung von Systemlog und Historydaten, sowie eigenen Dateien verwendet werden. Die Verzeichnisse dürfen nur verändert werden, wenn andere Verzeichnisse benutzt werden sollen und auf der Zentrale eingerichtet wurden.

Infos zum PC als Zentrale sind [hier](#) beschrieben.

Auf der Seite *Zentrale* des Formuläreiters werden die IP-Adresse der Zentrale und der auf der Zentrale benutzte Port (2110) eingegeben. Dieser Port wird auch in den Verbindungsdaten des Programms [VisuWin](#) angegeben und dient der Verbindung mit der *ExecEngine* auf der Zentrale. **Bitte beachten Sie:** bei der Angabe von IP-Adressen dürfen grundsätzlich keine führenden Nullen eingegeben werden !  
( Beispiel: 192.168.2.25 ist korrekt, 192.168.002.025 ist falsch! )

Auf der Seite *eigene IP* werden im Listfenster die auf dem PC zur Verfügung stehenden IP-Adressen angezeigt. Hier muss die IP-Adresse ausgewählt werden, die zur Herstellung der Verbindung mit der Zentrale benutzt werden soll. Als Standard-Port hier ist 3101, falls dieser anderweitig benutzt wird kann ein anderer Port eingegeben werden.

Auf der Seite *Zentrale* kann mit dem Knopf **[Verbindungstest Zentrale]** getestet werden ob die Verbindungsdaten zur Zentrale korrekt eingegeben wurden.

Mit dem Knopf **[Kontrollprogramm für Zentrale]** wird das Programm ExecEngineWin gestartet, das zur Steuerung und zum Dateiaustausch mit der Exec-Engine auf der Zentrale dient.

### Tab BidCoS (nur bei PC als Zentrale für HomeMatic)

Auf dieser Seite kann mit dem Button "BidCoS-Interfaces konfigurieren" ein Fenster geöffnet werden, indem die Konfigurationsdaten für LAN-Konfigurationsadapter eingetragen werden können. Zudem kann hier der BidCoS-Service gestartet und gestoppt werden. Die für diesen Modus erforderliche BidCoS-Software für PCs und die LAN-Konfigurationsadapter werden von eQ-3 schon stehen schon seit mehreren Jahren nicht mehr zur Verfügung.gestellt bzw. hergestellt.

### Tab Gateways

Auf dieser Seite werden die als Schnittstellen verwendeten Gateways angelegt. Der Typ der Schnittstelle muss ausgewählt werden. Bei der Version CLX können auch CCUs als Gateways benutzt werden.

### Reiter FHZ (nur Version CLX und CL-Studio mit FHZ Option)

Auf dieser Seite werden die als Schnittstellen verwendeten Gateways (z.B. HomeMatic-CCUs) angelegt. Der Typ der Schnittstelle muss angegeben werden.

## **Tab "eigene IP"**

Da die meisten PCs mehrere IP-Adresse haben muss hier die IP-Adresse des Netzwerks angegeben werden, in dem die Zentrale angeschlossen ist.

## 1.17.1. PC als Zentrale

---

### PC als Zentrale

**Der BidCoS-Service für PCs wird von eQ-3 nicht mehr unterstützt und die erforderlichen HomeMatic-Konfigurationsadapter werden nicht mehr hergestellt und vertrieben. Die Beschreibung zum PC als Zentrale im Folgenden bezüglich HomeMatic beschreibt daher Hardware und Firmware-Software, die nicht nicht mehr als Neuware erhältlich ist bzw. nicht mehr im Downloadbereich des Herstellers zur Verfügung steht.**

**Mit der Version CLX kann unabhängig von einem BidCoS auf dem PC der PC als Zentrale mit CCUs als Schnittstellen benutzt werden.**

Die Option den Typ der Zentrale zu ändern steht nicht in allen Programmversionen zu Verfügung. Die Programmversionen, in denen der Zentralentyp ausgewählt werden kann haben im [Hardwarefenster](#) eine Auswahlbox für den Zentralentyp.

Mit dieser Option ist es möglich eine Hausautomation mit einem PC ohne eine separate Zentrale zu realisieren, es wird nur ein HomeMatic-Konfigurationsadapter oder eine FHZ-Schnittstelle (im Folgenden Interface genannt) benötigt um mit den Modulen zu kommunizieren.

Der PC ist jedoch als Zentrale nicht in gleicher Weise geeignet wie die entsprechenden zur Verfügung stehenden Zentralen, insbesondere bezüglich der im Normalfall erforderlichen hohen Betriebssicherheit, die aufgrund der vielen unterschiedlichen Programme und Rahmenbedingungen auf einem PC nicht mit einer der angebotenen Zentralen vergleichbar ist.

Auf dieser Seite ist unter anderem beschrieben wie die Interfaces (LAN- oder USB-Version) konfiguriert werden, wenn der PC als Zentrale benutzt wird. Falls Sie den PC als Zentrale benutzen, lesen Sie bitte unbedingt diese Hinweise und gehen bei der Konfiguration wie in den Hinweisen beschrieben vor.

Die ExecEngine ist die Software, die im laufenden Betrieb die Steuerung übernimmt und mit den einzelnen Modulen kommuniziert. Im Regelfall läuft die ExecEngine auf einer externen Linux-Zentrale (CCU von eQ-3, CL-Box oder einer anderen Zentrale von contronics). Sie kann jedoch auch auf dem PC laufen, so dass keine separate externe Zentrale erforderlich ist. Mit der Option die ExecEngine auf dem PC laufen zu lassen besteht auch die Möglichkeit Anwendungsprogramme zu testen ohne die laufende Anwendung einer externen Zentrale zu behindern. Falls eine Anwendung auf dem PC getestet werden soll während die ExecEngine auf einer externen Zentrale aktiv ist, muss auf jeden Fall die Option *Hardwareverbindung deaktivieren* ausgewählt werden, sonst kann es zu Konflikten und Kommunikationsproblemen mit der externen Zentrale kommen.

**Bitte beachten Sie Folgendes wenn Sie die Benutzerkontensteuerung von Windows eingeschaltet haben:**

Standardmässig wird das Programm nicht mit Administratorrechten gestartet um keine Sicherheitsabfrage bei eingeschalteter Benutzerkontensteuerung auszulösen. Je nach Windows-Version und Benutzerkonteneinstellung kann aber auf Dienste (also auf den BidCoS-Service) nur von einem Programm mit Administratorrechten zugegriffen werden. Bei eingeschalteter Benutzerkontensteuerung kann es je nach Windows-Version (insbesondere bei Windows 7) sein, dass der Status des BidCos-Services auf "nicht aktiv" steht, obwohl er tatsächlich aktiv ist. Der BidCos-Service kann dann auch nicht vom Programm gestartet oder beendet werden. Um dieses Problem zu beseitigen können Sie das Programm mit Administratorrechten starten lassen. Dazu klicken Sie die Programmdatei im Dateimanager mit der rechten Maustaste an und wählen im PopUp-Menü den Menüpunkt "Eigenschaften". Gehen Sie im Fenster das sich dann öffnet auf den Reiter "Kompatibilität" und aktivieren das Optionsfeld "Programm als ein Administrator ausführen" im unteren Teile des Fensters. Nach dieser Änderung kann es (wieder je nach Windows-Version) passieren, dass Sie die Programmausführung bei jedem Aufruf bestätigen müssen.

Normalerweise wird der BidCos-Service beim Start des Rechners automatisch gestartet wenn es eine korrekte BidCoS-Konfigurationsdatei gibt. Daher ist es nach der Erstellung der Konfigurationsdatei nicht unbedingt nötig den BidCosS-Service vom Programm steuern und anzeigen zu lassen. Allerdings wird ohne die Administratorrechte der Status immer als "nicht aktiv" angezeigt. Alternativ gibt es natürlich noch die Möglichkeit die Benutzerkontensteuerung einfach auszuschalten....

## Hinweise zur Konfiguration der BidCoS-Interfaces für HomeMatic

Falls Sie HomeMatic-Konfigurationsadapter verwenden, starten Sie das Programm

*HomeMatic->Hilfsprogramme->HomeMatic-Lan-Interface konfigurieren.*

In diesem Programm werden alle im Netzwerk befindlichen LAN-Interfaces angezeigt. Die Interfaces sollten mindestens die Firmware-Version 0.961 haben, falls Ihr Interface eine kleinere Versionsnummer hat machen Sie bitte einen Firmwareupdate. Nehmen Sie die IP-Einstellungen vor. Sollte es Verbindungsprobleme geben deaktivieren Sie DHCP und vergeben feste IP-Adressen. Aktivieren Sie anfangs noch nicht die AES-Verschlüsselung. Aktivieren Sie diese erst wenn die Kommunikation mit allen Komponenten problemlos funktioniert.

Für den Betrieb der ExecEngine auf dem PC ist der BidCos-Service erforderlich. In den Installationsdateien zum LAN- bzw. USB-Interface ist dieser enthalten.

Vor dem nächsten Schritt muss der BidCos-Service installiert worden sein, sonst fehlt die erforderliche Konfigurationsdatei.

Bevor Sie Module anlernen müssen die verwendeten BidCoS-Interfaces in einer Konfigurationsdatei definiert werden. Öffnen Sie dazu das Fenster *Hardware* der homeputer CL-Software unter dem Menüpunkt *Konfigurieren->Hardware*. Wählen Sie in dem Fenster *Zentrale* den Reiter *Zentrale*. Aktivieren Sie die Option *Ausführung auf dem PC*. Klicken Sie auf den Knopf [BidCoS-Interfaces konfigurieren]. Das Programm *Konfiguration BidCoS-Interfaces* wird gestartet. Bei ersten Start befinden sich normalerweise Testdaten in der Konfigurationsdatei. Löschen bzw. ändern Sie diese entsprechend den von Ihnen benutzten Interfaces. Legen Sie in diesem Programm alle benutzten Interfaces an. Vergessen Sie nicht den korrekten Typ einzustellen und achten Sie darauf die Seriennummer korrekt einzugeben. Wählen Sie als Log-Level den normalen Status "Fehler" damit die Log-Dateien nicht zu gross werden.

Beenden Sie dieses Programm durch Anklicken des Knopfes oder Menüpunkts [Beenden und Konfigurationsdatei speichern].

Wenn der BidCos-Service sich nicht starten lässt, ist das meistens auf Fehler in der Konfigurationsdatei zurückzuführen. Das kann z.B. durch eine falsche Seriennummer für ein Interface verursacht werden. Wenn der BidCos-Service startet, die Kommunikation aber nicht funktioniert überprüfen Sie bitte die Eintragungen in der Konfiguration, insbesondere Seriennummer und Typ der Interfaces.

Wenn die Power-LED bei dem LAN-Interface blinkt hat der BidCos-Service keine Verbindung zum LAN-Interface weil der BidCos-Service nicht aktiv ist oder keine Netzwerkverbindung zum BidCos-Service aufgebaut werden kann. Ursache können fehlerhafte Einstellungen in der Konfigurationsdatei sein.

## Hinweise zum Anlernen und Konfigurieren von HomeMatic-Komponenten

Starten Sie den HomeMatic-Konfigurator unter dem Menüpunkt

*HomeMatic->HomeMatic-Komponenten konfigurieren.*

Der HomeMatic-Konfigurator zum Anlernen der HomeMatic-Komponenten kann über den BidCoS-Service mit dem oder den Interfaces kommunizieren oder falls Sie nur ein Interface benutzen auch direkt mit dem Interface verbunden werden. Die Auswahl erfolgt unter dem Menüpunkt *Datei->Einstellungen* des HomeMatic-Konfigurators.

Bitte beachten Sie dabei: Wenn Sie mit mehreren Interfaces arbeiten können Sie den Auswahlpunkt *Entfernter BidCoS-Service* benutzen, dann muss der BidCoS-Service auf dem PC vorher gestartet worden sein (ist also in Wirklichkeit in diesem Fall nicht "entfernt", als IP-Adresse geben Sie die IP des PC's an). Der BidCoS-Service wird mit dem homeputer CL-Programm im Fenster *Hardware* auf der Seite *Zentrale* gestartet oder gestoppt. Das Fenster *Hardware* wird mit dem Menüpunkt *Konfigurieren->Hardware* geöffnet.

Wichtig ist, das die BidCoS-Konfigurationsdatei wie im vorigen Abschnitt beschrieben mit den Einstellungen für die verwendeten Interfaces erstellt wurde.

Wenn Sie nur ein Interface (egal ob LAN oder USB) benutzen können Sie die Konfiguration auch auf diese Weise konfigurieren oder jeweils die Option *HomeMatic-Lan-Interface* oder *HomeMatic-USB-Interface* auswählen. Wenn Sie die Interfaces nicht über den BidCoS-Service verbinden, sondern die Einzelverbindung benutzen, muss der vorher BidCoS-Service beendet werden, da es sonst zu Kommunikationskonflikten kommen kann.

Lernen Sie nun die HomeMatic-Module mit dem Menüpunkt *Geräte anlernen* wie in der Bedienungsanleitung der Geräte beschrieben an.

Nach dem Anlernen der Geräte schliessen Sie das Programm HomeMatic-Konfigurator und starten den BidCoS-Service falls dieser nicht aktiv ist.

Dann öffnen Sie das Fenster Modulauswahl über den Menüpunkt *Konfigurieren->Modulauswahl* und klicken auf den Knopf [**Module importieren**] und die angelernten Module in die Software zu übernehmen. Hinweise zur weiteren Vorgehensweise finden Sie im Kapitel [Erste Schritte](#), wobei Sie die Teile, die sich auf die externen Zentralen beziehen überspringen können.

## 1.18. Objektdefinitionen

---

### Objektdefinitionen

Das Fenster zur Definition der Objekte wird auf der [Seite Objekte](#) des Einstellungsfensters durch Doppelklick auf ein Objekt oder den Button **[Bearbeiten]** gestartet.

## 1.18.1. Seite Allgemein der Objektdefinition

---

### Seite Allgemein

Auf dieser Seite wird der Objekttyp und die Eigenschaften des Objekts definiert.

#### Auswahlbox Objekttyp

In der Auswahlbox Objekttyp oben links im Fenster kann der Typ des Objekts ausgewählt werden. Wenn es sich um ein Objekt zu einem Hardwaremodul handelt wurde der für dieses Modul korrekte Standard-Typ bei der Erstellung des Moduls eingetragen. Dieser kann verändert werden, z.B. von Rolllade zu Markise oder es kann einem Dimmer oder Rollladenaktor der Typ Zahl zugewiesen werden um die prozentuale Einstellung des Moduls zu steuern. Da auch benutzerdefinierte Objekttypen erstellt werden können, kann das Programm nicht prüfen ob ein Objekttyp kompatibel zum zugeordneten Hardwaremodul ist. Bei einer Veränderung des Typs eines Objekts zu einem Hardwaremodul liegt es in der Verantwortung des Benutzers darauf zu achten, dass der ausgewählte Typ kompatibel zum Hardwaremodul ist. Eine Veränderung des Objekttyps verändert nicht den Typ der zugeordneten Hardware!

#### Optionsfelder

##### Deaktivieren beim Start

Das Objekt ist beim Start des Steuerungsprogramms [deaktiviert](#). Es kann mit der [Programmanweisung AKTIVIEREN](#) aktiviert werden.

##### Unsichtbar beim Start

Diese Auswahl bewirkt, dass das Objekt bei der Ansicht nach dem Programmstart [unsichtbar](#) ist. Während der Programmausführung kann das Objekt [sichtbar](#) gemacht werden.

##### Ausführen bei Eingabe

Diese Option bewirkt, dass das Makro des Objekts ausgeführt wird wenn dieses Objekt mit der Maus angeklickt wird. Bei Tasten von Modulen ist diese Option standardmässig nicht aktiviert, weil das Tastenereignis über die Zentrale aktiviert wird und das Makro somit ausgeführt wird wenn "Ausführung bei Empfang" aktiviert ist. Wenn diese Option ausgewählt ist, wird nur das Makro des Objekts ausgeführt, aber keine Funkmeldung über die Zentrale versendet. Falls direkte Verknüpfungen mit der Taste bestehen wird der Verknüpfungspartner der Taste also nicht geschaltet.

##### Ton beim Umschalten

Diese Auswahl bewirkt einen Bestätigungston bei einer Änderung des Objektzustands.

##### nicht benutzen

Wenn diese Option aktiviert ist, steht dieses Objekt für die Programmierung nicht zur Verfügung und wird in Auswahllisten für Objekte nicht angezeigt und darf auch anderweitig nicht benutzt werden.

Bitte beachten Sie:

Wird dieses Objekt z.B. in einem manuell erstelltem Makro verwendet, so führt das zu einem Fehler bei der Erstellung des Ausführungscode.

Diese Option sollte nur ausnahmsweise mit Vorsicht benutzt werden.

##### Bezeichnung

Hier kann eine Bezeichnung für das Objekt angegeben werden. Diese Bezeichnung wird im [Ausführungsfenster](#) verwendet.

##### Startwert

Der hier angegebene Wert wird beim Start des Programms für das Objekt eingestellt.

Wenn der Objektwert eine Dezimalzahl ist, wird die Anzahl der Kommastellen, die in der Visualisierung angezeigt werden, durch den Startwert bestimmt. Wenn zwei Kommastellen angezeigt werden sollen würde also beispielsweise ein Startwert von 0,00 angegeben.

Weitere Eingabefelder in diesem Fenster können sein:



### **Resetzeit in Sekunden** (nur bei bestimmten Sensoren)

Bei Tasten von Fernbedienungen, bei Bewegungsmeldern und anderen Binärsensoren und virtuellen Objekten vom Typ Taster erscheint ein zusätzliches Eingabefeld *Resetzeit in Sekunden*. In dieses Feld kann die Zeit in Sekunden angegeben werden, nach der z.B. eine gedrückte Taste wieder in den Normalzustand (nicht gedrückt) versetzt wird oder ein Bewegungsmelder wieder in seinen Normalzustand gesetzt wird. Dieses automatische Rücksetzen dient dazu den Normalzustand eines Objektes wieder herzustellen, das Makro des Objektes wird dadurch nicht gestartet.

Bei Objekten vom Typ **Taster** kann hinter der Resetzeit eine Visualisierungsaktion angegeben werden, indem hinter der Anzahl der Sekunden die Zeichen "->" eingegeben werden, gefolgt vom Schlüsselwort *Vollbild* oder *Fenster* oder dem *Namen der Ansicht*. Weitere Infos dazu sind unter **Visualisierungs-Aktion** beschrieben.

Beispiel einer Eingabe bei Benutzung einer Taste zum Wechseln der Ansicht:

#### **3->Obergeschoss**

Diese Eingabe bewirkt, dass sofort die Ansicht "Obergeschoss" angezeigt wird und die Taste nach 3 Sekunden wieder in den Normalzustand gesetzt wird.

### **Visualisierungs-Aktion**

Bei Objekten vom Typ Knopf erscheint ein zusätzliches Eingabefeld *Visualisierungs-Aktion*. In diesem Feld kann eingetragen werden was bei Betätigung des Knopfes im Visualisierungsprogramm VisuWin passieren soll.

Es gibt folgende Eingabemöglichkeiten in diesem Feld:

#### **Vollbild**

Das Wort Vollbild bewirkt, dass die aktuelle Ansicht als Vollbild angezeigt wird und den ganzen Bildschirm ausfüllt. Mit der ESC-Taste kann zurück in den Fenstermodus geschaltet werden.

#### **Fenster**

Das Wort Fenster bewirkt, dass die aktuelle Ansicht als Fenster dargestellt wird. Diese Funktion dient dazu aus den Vollbild-Modus zu beenden ohne die Taste ESC benutzen zu müssen (z.B. für Touchscreen-Monitore).

#### **Name einer Ansicht**

Wenn Name einer Ansicht im Feld Visualisierungs-Aktion eingetragen wird, wird das Fenster für diese Ansicht bei Betätigung des Knopfes angezeigt.

Diese Visualisierungs-Aktionen werden ausschliesslich im Visualisierungsprogramm [VisuWin](#) durchgeführt und haben keinen weiteren Einfluss auf die Programmausführung.

### **Hardware-Modul**

In diesem Rahmen wird angezeigt, mit welcher Hardware dieses Objekt verbunden ist. Die Adressen werden nicht bei allen Systemtypen angezeigt.

Für das HomeMatic-System kann die Hardwarezuordnung des Objektes bearbeitet werden und ein Objekt ohne Hardwarezuordnung kann einem HomeMatic-Modul zugeordnet werden.

Ein entsprechender Dialog öffnet sich durch Anklicken des Buttons **[Hardwarezuordnung ändern]**.

Bei der Zuordnung können Kanalnummer und Datenpunktname angegeben werden. Der Typ des Objekts muss zum Typ des Kanals passen. Für boolsche Typen sollte ein Objekttyp mit zwei Zuständen benutzt werden, der erste Zustand entspricht *false*, der zweite Zustand *true*. Wenn ein Zahlenobjekt ohne Kommastellen für boolsche Werte benutzt wird ist 0=*false* und 1=*true*.

#### **Bitte unbedingt beachten:**

Diese Option ist für erfahrene Benutzer gedacht, die das Konzept zur Adressierung von HomeMatic-Hardware kennen und wissen welche Auswirkungen Änderungen haben.

Es erfolgt keine Plausibilitäts- oder Fehlerprüfung! Bei falschen Einstellungen kann es sein, dass das Objekt oder andere Objekte des Moduls nicht mehr funktionieren.

Sichern Sie Ihr Projekt unbedingt, bevor Sie Änderungen an der Hardwarezuordnung vornehmen.

## 1.18.2. Seite Makro

---

### Seite Makro

Für jedes Objekt kann ein eigenes Makro erstellt werden. In diesem Makro wird programmiert welche Funktionen ausgeführt werden sollen. Makros können durch Empfang vom [Sensor](#) eines Objekts, durch eine Änderung des Objektwerts oder in bestimmten Zeitintervallen aktiviert werden. Welche Option der Ausführungsaktivierung benutzt werden soll, wird auf dieser Seite festgelegt.

Das Zeitintervall wird in der Auswahlbox über dem Fenster mit dem Programmtext eingestellt. In den meisten Fällen ist das Zeitintervall "jede Minute" oder "alle 5 Sekunden" die richtige Wahl. Wenn in dem Objektprogramm Zeiten abgefragt werden, bei denen Sekunden angegeben sind, muss das Zeitintervall auf *alle 5 Sekunden* eingestellt werden, da die Funktion "Uhrzeit" die Zeit im 5-Sekunden-Raster zurückgibt.

Ausführungsintervalle kleiner 5 Sekunden sollte nur gewählt werden wenn diese kurzen Intervalle tatsächlich erforderlich sind. Wenn zu viele Makros unnötigerweise in sehr kurzen Intervallen ausgeführt werden, kann das dazu führen, dass die Anwendung insgesamt langsamer wird und einzelne Aktionen grössere Reaktionszeiten haben, da Makros die unnötigerweise zu oft ausgeführt werden die Gesamtperformance und die Kommunikation mit der Hardware behindern können.

#### Bitte beachten Sie:

Das Zeitintervall *permanent* ist nur in Sonderfällen sinnvoll und sollte mit Bedacht gewählt werden. Es sollte nicht benutzt werden wenn es sich um ein grösseres Makro handelt oder andere Makros in dem permanent aktivierten Makro gestartet oder aufgerufen werden.

Keinesfalls dürfen von einem permanenten Makro bei jedem Durchlauf Zustands- oder Werteänderungen ausgehen, die zu einer Hardwareaktion oder einer Änderung der Visualisierung führen würden. Diese könnten in der schnellen Taktfrequenz von den Netzwerkrouتين nicht bearbeitet werden, was letztlich zu einem Speicherüberlauf und einem Programmabsturz führen kann. Verwenden Sie also das Zeitintervall *permanent* nur mit Bedacht und wenn Sie die Auswirkungen übersehen können. Das gilt je nach verwendeter Hardware und Grösse des Projekts auch für Zeitintervalle von mehreren Ausführungen in der Sekunde.

**Sollten Sie permanente Ausführung oder ein Ausführungsintervall kleiner als 1 Sekunde verwenden und Ihr Projekt nicht fehlerfrei laufen erhöhen Sie das Zeitintervall.**

Für Sensoren gibt es den Auswahlpunkt **Ausführung bei Empfang**.

Wenn dieser aktiviert ist, wird das Objektprogramm ausgeführt, sobald eine Meldung vom zugeordneten Sensor empfangen wird. Die Ausführung erfolgt auch wenn sich der Sensorwert nicht geändert hat.

Für Aktoren gibt es den Auswahlpunkt **Ausführung bei Änderung**.

**Achtung : Die unsachgemässe Verwendung dieser Option kann schwerwiegende Fehler zur Folge haben.**

Wenn diese Option aktiviert ist, wird das Makro immer dann ausgeführt, wenn sich der Wert des Objekts ändert - egal an welcher Stelle diese Änderung erfolgt (also auch innerhalb des eigenen Objektmakros).

Die Benutzung dieser Option ist insbesondere sinnvoll in Makros von Tür/Fensterkontakten, da bei Verwendung von Ausführung bei Empfang das Makro auch beim Empfang von Statusmeldungen aktiviert wird, ohne dass sich der Zustand geändert hat.

Diese Option bietet viele Möglichkeiten, die Benutzung ist jedoch auch gefährlich - wenn mehrere Objekte diese Option verwenden und es gegenseitige Abhängigkeiten gibt, so dass Makros wechselseitig aktiviert werden können Endlosschleifen entstehen, die das ganze System blockieren.

**Diese Option sollte daher nur mit Vorsicht benutzt werden, wenn in dem Makro Zustände anderer Objekte verändert werden, die ebenfalls diese Option verwenden.**

**Keinesfalls darf es zu einem Zirkelbezug kommen!**

Makros können die Zustände des eigenen und aller anderen [Objekte](#) abfragen und ändern. Zustände von Objekten und den diesen zugeordneten Hardwaremodulen werden automatisch abgeglichen, d.h. wenn ein zugeordneter Sensor eine Zustandsänderung meldet, ändert sich der

Zustand des Objekts. Wenn das Programm einen Objektzustand ändert wird eine Meldung an den zugeordneten [Aktor](#) (Empfänger) geschickt.

Mit dem Button **[Menü-Makro bearbeiten]** kann das Fenster zur Menügeführten Erstellung von Makro-Anweisungen aufgerufen werden. Dieses Fenster ist nur aus Kompatibilitätsgründen zur Standardversion des Programms vorhanden und verfügt nur über eine eingeschränkte Auswahl an Makroanweisungen.

Mit der Schaltfläche **[Objekte/Var.]** kann ein Element einer Liste der Objekte und Variablen ausgewählt werden, das an der Cursorposition in das Makro eingesetzt wird.

Mit der Schaltfläche **[Anweisungen]** kann eine Anweisung aus der Liste der meistbenutzten Anweisungen ausgewählt werden. Diese wird direkt der Cursorposition in das Makro eingesetzt. Die Anweisungen sind nicht alphabetisch, sondern nach Anwendungshäufigkeit und Gruppen sortiert.

Mit der Schaltfläche **[Editor]** wird ein separater Editor aufgerufen, der eine bessere Übersichtlichkeit und Optionen zum Suchen/Ersetzen bietet.

Dieser Editor kann auch durch Doppelklick auf das Makrofenster gestartet werden.

Im unteren Teil der Seite können [Variablen](#) für das Objektprogramm definiert werden.

Variablen sind dem jeweiligen Objekt zugeordnet, können jedoch auch in anderen Objektprogrammen verwendet werden, wenn der Name der Variablen qualifiziert, d.h. mit vorangestelltem zugehörigem Objektnamen, von der Variablen durch einen Punkt getrennt, angegeben wird.

Wenn Beispielsweise im Programm **Gartenlicht** die Variable **Autoschalter** definiert ist, so kann diese in anderen Programmen unter dem Namen **Gartenlicht.Autoschalter** verwendet werden.

Da das Navigieren in der Variablentabelle ohne Maus für Anwender mit Sehbehinderung schlecht machbar ist wurde folgende spezielle Tastensteuerung implementiert:

Mit Alt-V wird der Cursor in die Variablen-Tabelle gesetzt.

Mit Alt-X wird der Cursor eine Spalte weiter gesetzt, wenn er sich in der letzten Spalte befindet in die nächste Zeile.

Mit Alt-Y wird der Cursor eine Spalte zurück gesetzt, wenn er sich in der ersten Spalte befindet in die Zeile über der aktuellen Zeile.

## 1.18.3. Seite Zeit-Tabelle

---

### Seite Zeit-Tabelle

Auf dieser Seite kann festgelegt werden wann ein Aktor-Objekt geschaltet wird bzw. wann ein Makro gestartet werden soll.

Bei Sensor-Objekten ist diese Seite nicht sichtbar.

Die Einschaltdauer steht nur bestimmten Objekten in Abhängigkeit des Modultyps zur Verfügung. Durch Eingabe von **SA** im Feld Uhrzeit kann die Sonnenaufgangszeit gewählt werden, mit **SU** die Sonnenuntergangszeit.

Das Eingabefenster zur Angabe der geographischen Lage Ihres Standorts wird mit dem Knopf [weitere Einstellungen] auf der Seite Allgemein geöffnet.

Im Normalfall ist diese Tabelle immer während der Ausführung aktiviert.

Die Aktivierung der Tabelle kann über Makros gesteuert werden, indem die Variable des Typs Schalter mit dem Namen *Zeittabelle* ein-/ bzw. ausgeschaltet wird. Diese Variable wird für Aktoren automatisch angelegt. Die in der Tabelle festgelegten Schaltungen werden nur vorgenommen, wenn die Variable *Zeittabelle* des Objekts eingeschaltet ist.

Die Aktivierung der Tabelle kann also z.B. mit den Makroanweisungen

`Objektname.Zeittabelle einschalten`

und

`Objektname.Zeittabelle ausschalten`

gesteuert werden.

## 1.18.4. Seite Visualisierung

---

### Seite Visualisierung

Auf dieser Seite wird definiert, wie ein Objekt in der Visualisierung im Programm [VisuWin](#) dargestellt werden soll.

Dazu können im Listefeld *Darstellung* eine oder mehrere unterschiedliche Darstellungsformen für das Objekt angelegt werden.

Im Listefeld Ansichten werden die Ansichten angezeigt, in denen das Objekt darstellt wird. Es können zusätzliche Ansichten ausgewählt werden, in diesen wird das Objekt dann in der linken oberen Ecke positioniert und kann später im [Entwurfsfenster](#) für Ansichten an die gewünschte Stelle positioniert werden.

Mit dem Knopf zwischen den beiden Listefeldern wird die aktuell ausgewählte Darstellungsform der aktuell ausgewählten Ansicht zugeordnet. So kann das Objekt in jeder Ansicht unterschiedlich dargestellt werden, es kann allerdings auch eine Darstellungsform für alle Ansichten benutzt werden.

Die Darstellungsform *Standard* wird automatisch erstellt, wenn ein Objekt in einer Ansicht positioniert wird und noch keine Darstellungsform definiert wurde.

Je nach Objekt und ausgewählter Darstellungsart erscheint der Knopf [Rahmendarstellung], mit dem vielfältige Einstellungen wie Farbe, Ränder, Schriftarten und -größen zur Darstellung in einem Rahmen vorgenommen werden können.

Wenn das Optionsfeld "Keine Eingabe" aktiviert wird, kann der Wert des Objekts bei der Visualisierung in der aktuell ausgewählten Darstellung nicht verändert werden.

## 1.18.4.1. Bildsymbole für Objekte

---

### Bildsymbole für Objekte

In diesem Fenster kann ein Bild-Symbol für das aktuelle Objekt ausgewählt werden.

Beim Öffnen des Fensters wird eine Auswahl von Bild-Symbolen angezeigt, die am ehesten auf das Objekt zutreffen. Mit der Schaltfläche [Mehr anzeigen] werden weitere im Programm zur Verfügung stehende Bild-Symbole angezeigt, unabhängig davon ob für das aktuelle Objekt zutreffen oder sinnvoll sind.

Mit der Schaltfläche [Bild ändern] können beliebige Bitmaps zur Darstellung des Objekts bzw. des Objektzustands eingebunden werden.

Solche Bitmaps können auch selbst erstellt werden. Das kann mit z.B. mit dem standardmässig in jeder Windows-Version vorhandenem Programm MS-Paint/Paintbrush geschehen (unter Zubehör).

Mit der Schaltfläche [Neuer Darstellungstyp] können neue Darstellungstypen erstellt werden. Dazu wird die aktuell ausgewählte Objektdarstellung inklusive der Zustandsarten, aber ohne die zugehörigen Bilder kopiert. Als Bild wird für jeden Zustand ein Testbild eingesetzt. Dieses kann dann durch eigene Bitmaps ersetzt werden.

Ein solcher selbst definierter Darstellungstyp steht auch für andere Objekte des Projekts zur Verfügung.

Mit der Schaltfläche [Darstellungstyp löschen] können selbst erstellte Darstellungstypen wieder gelöscht werden, jedoch nicht die standardmässig im Programm angebotenen Darstellungstypen.

# 1.19. Fenster Anwesenheitssimulation

---

## Fenster Anwesenheitssimulation

Das Fenster Anwesenheitssimulation wird aufgerufen aus der [Seite Allgemein](#) des Einstellungsfensters.

In diesem Fenster kann angegeben werden, innerhalb welcher Zeiten Empfänger geschaltet werden sollen. Der Empfänger wird dann an den ausgewählten Tagen zu einem zufälligen Zeitpunkt innerhalb des angegebenen Zeitraums geschaltet. Wenn keine obere Grenze des Zeitrahmens angegeben wird (Spalte *und*) wird innerhalb von 1-5 Sekunden nach dem in der Spalte *zwischen* festgelegten Zeitpunkt geschaltet.

Wenn keine Dauer angegeben wird bleibt der Schaltzustand erhalten.

Die Anwesenheitssimulation dient dazu, bei Abwesenheit der Bewohner Geräte so zu schalten, dass ein Beobachter den Eindruck bekommt das Haus oder die Wohnung sei nicht verlassen. Normalerweise ist es nicht gewünscht oder sinnvoll, dass die Anwesenheitssimulation dauernd aktiv ist, daher kann die Anwesenheitssimulation über einen Schalter ein- bzw. ausgeschaltet werden.

Sobald Zeiten in der Anwesenheitssimulationstabelle eingetragen sind, wird bei der Ausführung ein spezielles [Objekt](#) generiert, das Objekt **Anwesenheitssimulation**. Dieses Objekt wird mit dem Typ *EAGeraet* angelegt. Somit hat es eine Zeittabelle und kann über die diese wie auch über eine Visualisierung ein- und ausgeschaltet werden.

In den Uhrzeitfeldern können auch Sonnzeiten angegeben werden, SA für Sonnenaufgang, SU für Sonnenuntergang.

Bei der Verwendung von Sonnenzeiten wird nicht geschaltet, wenn die Sonnenzeiten am aktuellen Tag so sind, dass die Bis-Uhrzeit kleiner als die Von-Uhrzeit ist. In dem Fall wird diese entsprechende Zeitspanne komplett ignoriert.

Wenn es ein Objekt mit dem Namen *Anwesenheitssimulation* (z.B. als Schalter) schon gibt, so wird dieses benutzt um die Anwesenheitssimulation ein- bzw. auszuschalten. Es wird dann kein neues Objekt generiert.

**Die Anwesenheitssimulation kann nur über einen vorhandenen oder „virtuellen“ Schalter mit dem Namen „Anwesenheitssimulation“ ein- und ausgeschaltet werden.**

**Bitte beachten Sie:**

*Wenn Sie ein eigenes Objekt Anwesenheitssimulation erstellen, müssen Sie darauf achten, dass dieses ein Objekt ist, dass ein- und ausgeschaltet werden kann.*

# 1.20. Das Fenster Visualisierung

---

## Das Fenster Visualisierung

In diesem Fenster können individuelle eigene Ansichten erstellt werden, die dann in im Visualisierungsprogramm [VisuWin](#) angezeigt werden

Das Fenster wird aufgerufen durch Anklicken des Buttons **[Ansichten bearbeiten]** auf der Seite *Ansichten* des Einstellungsfensters.

Wenn ein Hintergrundbild ausgewählt wurde, werden die [Objekte](#), für die nicht [Objektrahmen oder Textrahmen](#) als Darstellungsart gewählt wurde als Bild-Symbol dargestellt.

Wenn kein Hintergrundbild gewählt wurde, werden alle Objekte mit einem Rechteck dargestellt.

Die Grösse dieser Rechtecke kann verändert werden, indem man mit der Maus auf den Rand des Rechtecks geht. Es erscheint ein Doppelpfeil für die Richtung und man kann die Grösse des Objektes mit gedrückter Maustaste verändern.

Die Mindestgrösse des Objekts richtet sich nach seiner Textlänge im Visualisierungsfenster (nicht in diesem Entwurfsfenster!). Es kann nicht so klein gemacht werden, dass die aktuelle Bezeichnung des Objekts abgeschnitten würde. (Ausser der Bezeichnungstext würde nach der Positionierung in diesem Fenster verändert).

Normalerweise entspricht eine Ansicht einem Geschoss des Gebäudes.

Zur Darstellung einer Ansicht kann auf der [Seite Ansichten](#) ein [Bitmap](#) ausgewählt werden. Vor dem Hintergrund dieses Bitmaps werden die dieser Ansicht zugeordneten Objekte dargestellt. Jedes Objekt kann in bis zu drei verschiedenen Ansichten mit unterschiedlicher Grösse und Position dargestellt werden.

Um ein Objekt auszuwählen, wählen Sie im Menü Objekte den Menüpunkt *Objektliste* aus. Es erscheint ein Fenster mit allen zur Verfügung stehenden Objekten. Wählen Sie das gewünschte Objekt aus diesem Fenster aus und ziehen es mit gedrückter linker Maustaste auf die gewünschte Position innerhalb des Entwurfsfensters und lassen die Maustaste los (Drag&Drop). Das Objekt lässt sich nur an eine freie Stelle positionieren, ob das Objekt an eine Stelle gesetzt werden kann, sehen Sie an der Form des Mauszeigers.

Eine weitere Möglichkeit ein Objekt auszuwählen ist ein Doppelklick auf das Objekt oder das Anklicken des Knopfes **[Übertragen]**. Auf diese Weise ausgewählte Objekte werden in der linken oberen Ecke des Entwurfsfensters platziert – von dort müssen Sie diese an die gewünschte Position bringen.

Um die Position eines Objektes zu verändern, gehen Sie mit der Maus auf das Objekt (als Mauszeiger erscheint eine Hand), drücken die linke Maustaste und ziehen das Objekt an die gewünschte Stelle.

Wenn der Mauszeiger sich nicht über einem Objekt befindet und die rechte Maustaste betätigt wird, wird ein Popup-Menü aufgerufen, mit dem markierte Objekte behandelt werden können.

Objekte werden markiert durch Drücken der STRG-Taste und gleichzeitigem Drücken der linken Maustaste, wenn der Mauszeiger sich in einem Objekt befindet, oder wenn die linke obere Ecke des Objekts im Markierungsrechteck liegt. Das Markierungsrechteck wird erzeugt, indem der Mauszeiger bei gedrückter linker Maustaste von links oben nach rechts unten gezogen wird.

Nach der Markierung der Objekte kann mit dem Popup-Menü die gewünschte Aktion für die markierten Objekte ausgewählt werden. Die Markierung bleibt erhalten, bis sie mit dem Menüpunkt **Markierung entfernen** zurückgesetzt wird. Dieser Menüpunkt setzt auch den aktuellen Positionierungs-Modus zurück.

Um Objekte gruppenweise auszurichten oder zu positionieren, markieren Sie diese indem Sie die STRG-Taste drücken und mit er Maus anklicken, oder die linke Maustaste ausserhalb eines Objektes drücken und mit gedrückter Maustaste ein Rechteck ziehen, das die linke obere Ecke der zu markierenden Objekte einschliesst.

**Alles markieren** markiert alle Objekte im Entwurfsfenster.

**Ausrichtung vertikal** bedeutet, dass die markierten Objekte bei Betätigung der linken Maustaste auf die aktuelle vertikale Position (x-Position) des Mauszeigers gesetzt werden.

**Ausrichtung horizontal** bedeutet, dass die markierten Objekte bei Betätigung der linken Maustaste auf die aktuelle horizontale Position (y-Position) des Mauszeigers gesetzt werden.

**Gleiche Breite** bedeutet, dass alle markierten Objekte auf die Breite des breitesten markierten Objekts gebracht werden.

**Gleiche Höhe** bedeutet, dass alle markierten Objekte auf die Höhe des höchsten markierten Objekts gebracht werden.

**Verschieben** verschiebt alle markierten Objekte synchron zum Mauszeiger, wenn dieser sich nicht



in einem Objekt befindet und der linke Mausknopf gedrückt wird.

Die jeweilig aktivierte Aktion wird durch den Mauszeiger dargestellt, wenn dieser sich nicht auf einem Objekt befindet.

Diese Möglichkeiten der Positions- und Grössenmanipulation sind hilfreich bei der Positionierung vieler [Objektrahmen oder Textrahmen](#) in einer Ansicht, wenn kein Hintergrundbild angegeben wurde.

## 1.21. Fenster Makro-Anweisungen erstellen

---

### Fenster Makro-Anweisungen erstellen

In diesem Fenster werden die Anweisungen für Makros menügeführt erstellt. Dieses Fenster ist in der Standard-Version und in der Studio-Version vorhanden. In diesem Fenster können die erweiterten Anweisungen und Funktionen der Studio-Version nicht erstellt oder bearbeitet werden.

Wenn Sie einen Aktor schalten wollen, wählen Sie diesen aus dem Auswahlfenster *Objekt schalten/steuern* aus. Je nach Typ öffnet sich ein Fenster in dem Sie den gewünschten Zustand für das Objekt angeben können.

Wenn Schaltmodule eingeschaltet werden, kann die Einschaltdauer angegeben werden.

Mit dem Auswahlfenster *Anweisungen* können Anweisungen erstellt werden, die nicht das direkte Schalten von Empfängern betreffen.

Mit dem Knopf **[wenn-Bedingung]** wird ein Fenster geöffnet, mit dem Sie einen wenn-Block erstellen können. Anweisungen in einem wenn-Block werden nur ausgeführt wenn die Bedingung in der wenn-Zeile wahr ist.

Diese Anweisung ermöglicht komplexe logische Verknüpfungen aller [Objekte](#).

**Bitte beachten Sie folgende Hinweise:**

#### [Besonderheiten bei Dimmern](#)

**Bedenken Sie bei allen Zeitangaben:**

*Zeitbearbeitungen erfolgen im 5-Sekunden-Rhythmus, bei Zeitangaben müssen die Sekunden immer durch 5 teilbar sein (xx:xx:00, xx:xx:05, xx:xx:10 ... xx:xx:55).*

## 1.21.1. Makro-Anweisungen

---

### Makro-Anweisungen

#### warten

Mit dieser Anweisung kann das Makro an der Stelle der Anweisung für die angegebene Zeit angehalten werden.

#### Audio-Datei abspielen

Mit dieser Anweisung kann eine beliebige Wave-Datei abgespielt werden. Die Datei kann in einem Datei-Auswahlfenster ausgewählt werden.

#### Makro aufrufen

Mit dieser Anweisung wird innerhalb der Ausführung des aktuellen Makros ein anderes Makro ausgeführt.

#### Makro starten

Mit dieser Anweisung wird innerhalb der Ausführung des aktuellen Makros ein anderes Makro gestartet. Die Ausführung beginnt erst **nach** der Beendigung des aktuellen Makros.

#### Windows-Programm starten

Mit dieser Anweisung können Sie aus einem Makro heraus ein beliebiges Windows-Programm starten. Das Programm kann mit einem Dateiauswahlfenster ausgewählt werden.

## 1.21.1.1. warten

---

### Anweisung warten

Mit dieser Anweisung kann das Makro an der Stelle der Anweisung für die angegebene Zeit angehalten werden.

Diese Anweisung hat keine Auswirkungen auf andere Makros oder Funktionen.

Wenn das Makro innerhalb der Wartezeit erneut aufgerufen wird (nach Zeit, durch Tasterbetätigung oder durch ein anderes Makro) wird die alte Ausführung (und damit auch die Warte-Anweisung) abgebrochen, die Ausführung des Makros wird mit der ersten Zeile des Makros neu gestartet.

## 1.21.1.2. Audio-Datei abspielen

---

### Anweisung Audio-Datei abspielen

Mit dieser Anweisung kann eine beliebige Wave-Datei im Visualisierungsprogramm [VisuWin](#) abgespielt werden. Die Datei kann in einem Datei-Auswahlfenster ausgewählt werden.

Diese Option dient dazu aktuelle Informationen und gegebenenfalls akustische Warnungen auszugeben.

Wenn eine solche Anweisung ausgeführt wird, so beginnt die Ausgabe sofort, eine eventuell noch aktive Ausgabe einer anderen Wave-Datei wird abgebrochen.

Damit ist sichergestellt, dass aktuelle Meldungen und eventuell wichtige Warnhinweise ohne Verzögerungen direkt ausgegeben werden!

**Bitte beachten Sie:**

*Wenn Sie diesen Befehl mehrmals direkt hintereinander verwenden, werden Sie nur die letzte Datei hören, da die vorigen durch die jeweils nächste Anweisung unterbrochen werden.*

*Wenn Sie mehrere Dateien hintereinander ausgeben wollen, müssen Sie zwischen den Dateien [warte-Anweisungen](#) mit entsprechender Wartezeit einfügen.*

### 1.21.1.3. Makro aufrufen

---

#### Anweisung Makro aufrufen

Diese Anweisung führt ein Makro direkt aus, achten Sie unbedingt darauf, dass dadurch mehrmaligen verschachtelten Aufruf von Makros ein Makro nicht nochmals aufgerufen wird, dadurch würde eine Endlosschleife entstehen, die alles blockiert und letztlich zum Abbruch des aktuellen Makros aufgrund einer Zeitüberschreitung führt.

Das aufrufende Makro setzt seine Ausführung in der Zeile hinter dem Aufruf fort, wenn das aufgerufene Makro beendet wird.

**Bitte beachten Sie dabei:**

*Wenn in dem aufgerufenen Makro eine warte-Anweisung steht, wird in dem Moment in das aufrufende Makro zurückgesprungen, in dem die warte-Anweisung beginnt, die Verarbeitung des aufrufenden Makros wird fortgesetzt. Das aufgerufene Makro bleibt auch aktiv, dort wird die Ausführung nach Ablauf der Wartezeit wieder fortgesetzt.*

#### 1.21.1.4. Makro starten

---

##### Anweisung Makro starten

Mit dieser Anweisung wird innerhalb der Ausführung des aktuellen Makros ein anderes Makro zur Ausführung vorbereitet, jedoch nicht direkt ausgeführt. Die Ausführung beginnt erst nach Beendigung des aktuellen Makros.

Die Anweisung bewirkt, dass das zu startende Makro in eine Ausführungswarteschlange gestellt wird. Es können mehrere Makros gleichzeitig in dieser Warteschlange stehen, es wird immer das Makro als nächstes ausgeführt, das am längsten in der Warteschlange steht (FIFO-Prinzip).

Diese Anweisung hat einige Vorteile gegenüber der Anweisung *Makro aufrufen* und sollte - wenn der gleiche Zweck erfüllt wird – anstelle der Anweisung *Makro aufrufen* benutzt werden.

## 1.21.1.5. Windows-Programm starten

---

### Anweisung Windows-Programm starten

Mit dieser Anweisung können beliebige Windows-Programme gestartet werden.  
Das Programm kann mit einem Dateiauswahlfenster ausgewählt werden.  
Das angegebene Windowsprogramm wird aus dem dem Visualisierungsprogramm [VisuWin](#) gestartet, dieses muss dazu aktiv sein.



# 1.22. Die Programmierung des Systems

---

## Die Programmierung des Systems

Die Programmiersprache des Systems ist leicht zu erlernen und zur effektiven Programmierung des speziellen Anwendungsbereichs dieses Systems entwickelt. Diese Sprache ermöglicht es jedem Anwender ohne spezielle Programmierkenntnisse komplexe Steuerungsfunktionen zu realisieren. Für jedes [Objekt](#) kann ein [Makro](#) erstellt werden, welches durch die Änderung von Zuständen bzw. Werten die mit den Objekten verbundenen [Aktoren](#) steuert. Zustände und Werte von Objekten und [Variablen](#) sowie Zeiten können mit vielfältigen [Bedingungen](#) abgefragt und durch [Zuweisungen](#) verändert werden.

Anstelle der in der Beschreibung der einzelnen Anweisungen verwendeten Syntax kann auch die bei der automatischen Erstellung von Makro-Anweisungen verwendete Syntax verwendet werden.

In jeder Zeile eines Programms darf nur eine Anweisung stehen, Kommentarzeilen müssen mit 2 Schrägstrichen(//) beginnen, diese werden dann bei der Generierung des Codes ignoriert. In den Beispielen sind Programmanweisungen teilweise eingerückt, dies ist nicht erforderlich, aber aufgrund besserer Übersichtlichkeit empfehlenswert.

### Ein wichtiger Hinweis:

Bitte denken Sie daran Ihre Programm-Dateien (Endung SPG) vor Änderungen zu sichern. Das kann mit dem Menüpunkt Projekt->Schnellsicherung passieren, oder indem Sie das Projekt unter einem anderen Namen speichern. Der Projektname sollte allerdings nach Änderungen im Projekt nicht geändert werden, da der Projektname auch in den Apps verwendet wird und ein geänderter Projektname dazu führt, dass die Definitionen und Konfigurationen in den Apps unter einem neuen Projektnamen nicht mehr benutzt werden. Änderungen können nicht rückgängig gemacht werden, wenn diese sich als fehlerhaft herausstellen, oder beim Speichern bzw. Beenden des Programms ein Fehler auftritt kann das vorhandene funktionierende Programm verloren sein. Bei jedem Speichern wird automatisch eine Sicherheitskopie der SPG-Datei mit der Endung "SPB" erstellt, diese wird bei jedem erneuten Speichern wieder überschrieben, wenn also ein fehlerhaftes Programm zweimal abgespeichert wird, ist das Original verloren. Grundsätzlich sollten unbedingt auch Sicherungskopien der SPG-Datei auf einem separaten Datenträger gespeichert werden, so dass diese im Falle eines Festplattendefekts nicht verloren geht.

Wenn während der Code-Erstellung Fehler auftreten, wird das fehlerhafte Objektprogramm mit einer entsprechenden Fehlermeldung angezeigt. Die fehlerhafte Zeile wird soweit möglich markiert, hier kann es jedoch zu Verschiebungen kommen, da Fehler im Quell-Code nicht immer einer Zeile zugeordnet werden können. Wenn in der markierten Zeile kein Fehler erkennbar ist, müssen die vorangegangenen Zeilen geprüft werden.

Anweisungsteile, die bei der Erklärung der Syntax in < spitzen Klammern > stehen sind optional und können weggelassen werden.

Die Uhrzeit wird nicht jede Sekunde, sondern alle 5 Sekunden aktualisiert. Bei Zeitvergleichen ist also zu beachten, dass die Sekunden immer durch 5 teilbar sein müssen.

Wenn Schaltmodule eingeschaltet werden, kann die Einschaltdauer angegeben werden.

Grundsätzlich kann man zwischen Anweisungen und Funktionen einer Programmiersprache unterscheiden.

Anweisungen beschreiben in der Programmzeile was getan werden soll.

Funktionen liefern z.B. für eine Zuweisung oder eine Bedingungsabfrage Werte zurück.

### Bitte beachten Sie folgende Hinweise:

[Besonderheiten bei Dimmern](#)

## 1.22.1. Makros

---

### Makros

Makros sind Programme, die der Benutzer erstellen kann um die individuellen Anwendungsfunktionen seines Systems zu programmieren.

Jedes Objekt eines Moduls kann ein Makro beinhalten. Es können auch Objekte ohne Hardwarebezug (virtuelle Objekte) erstellt werden, für die ein Makro geschrieben wird.

Makros können durch Empfang vom [Sensor](#) eines Objekts, durch eine Änderung des Objektwerts oder in bestimmten Zeitintervallen aktiviert werden.

Welche Option der Ausführungsaktivierung benutzt werden soll, wird auf der [Seite Makro der Objektdefinitionen](#) festgelegt.

Makros können in bestimmten Zeitintervallen, durch den Empfang vom zugeordneten [Hardwaresensor](#), durch Aufruf aus anderen Makros oder durch Bedienung über eine Visualisierung aktiviert werden.

In einem Makro kann die Funktion [SELBST](#) verwendet werden, um das aktuelle Objekt, das das Makro beinhaltet zu benutzen.

Beispiel um eine Beleuchtung immer nach maximal einer Stunde auszuschalten in einem Makro, das im Zeitintervall von 1 Minute aktiviert wird:

```
wenn SELBST eingeschaltet und Schaltdauer(SELBST) > "01:00:00" dann
    SELBST ausschalten
endewenn
```

Es gibt zwei spezielle Makrotypen, die jeweils beim Start und bei der Beendigung der ExecEngine ausgeführt werden.

Makros, deren Namen mit "INIT\_" beginnen werden beim Start der ExecEngine vor allen anderen Makros ausgeführt.

Makros deren Namen mit "END\_" beginnen werden beim Beenden der ExecEngine ausgeführt. Das geschieht allerdings nur wenn die ExecEngine "normal" beendet wird, d.h. durch Compilieren und Übertragen eines Projekts oder durch das Kontrollprogramm ExecEngineWin.

Da das Starten der END-Makros erst nach Beendigung der Makro-Ausführungssteuerung beginnt, dürfen in END-Makros keine WARTE-Anweisungen verwendet werden. Bei Ausführung einer WARTE-Anweisung wird das End-Makro endgültig beendet, da es nach Ablauf der Wartezeit nicht mehr durch die Ausführungssteuerung aktiviert werden kann. Das gilt nicht wenn das End-Makro im normalen Betrieb z.B. von einem anderen Makro aufgerufen wird.

Weiterhin dürfen in einem END-Makro keine Aktionen gestartet werden, die im Hintergrund ablaufen, wie das Versenden von Mails oder ein http-Request mit GETSITE. Diese Hintergrundaktivitäten können dann eventuell nicht mehr komplett ausgeführt werden weil das Programm sich vorher beendet.

Die END-Makros werden nicht ausgeführt wenn die ExecEngine nicht über ExecEngineWin oder die Übertragung einer Ausführungsdatei beendet wird, also z.B. durch einen Neustart der Zentrale, einen Betriebssystembefehl in der Konsole oder Webserverfunktionen.

Kommentare und Bemerkungen in Makros beginnen mit zwei Schrägstrichen //, diese können am Anfang einer Zeile oder auch hinter einer Anweisung benutzt werden.

Beispiel:

```
// jetzt folgt die Berechnung
DWert1:=(a + b) / 2 // Durchschnittswert
```

Üblicherweise sind Makros nicht länger als 0,1 bis 0,2 Sekunden aktiv. Bei sehr langen Makros, die mehr machen als nur einige Aktoren zu schalten und mehrere hundert Zeilen haben kann die Ausführungszeit aber auch deutlich länger sein. Das sollte möglichst vermieden werden um die Reaktionszeiten zum Schalten von Aktoren bei der Bedienung nicht zu hoch werden zu lassen.

Daher wird während der Ausführung überprüft, ob ein Makro aussergewöhnlich lange läuft und in dem Fall ein Eintrag in das Systemlog geschrieben. Das passiert standardmässig wenn ein Makro länger als 2 Sekunden aktiv ist. Dann sollte das Makro ggfs. entsprechend verändert werden bzw. die Funktionen auf mehrere Makros aufgeteilt werden. Falls das nicht gewünscht ist, kann die Standardzeit bei der eine solche Warnung geschrieben wird verändert werden. Dazu wird ein Objekt des Typs Zahl mit dem Namen SCRIPTTIMEOUT erstellt und mit einem Startwert versehen, der die maximale Zeit in Sekunden angibt, die ein Makro aktiv sein darf bevor eine Meldung in das Systemlog geschrieben wird.

## 1.22.2. Variablen

---

### Variablen

Variablen dienen der Zwischenspeicherung von Werten. Variablen werden wie die Objektprogramme auf der [Seite Programmierung](#) der [Objektdefinition](#) definiert. Einer Variablen kann im [Objektprogramm](#) ein Wert zugewiesen werden. Dieser Wert kann der Inhalt einer anderen Variablen, ein Objektzustand oder eine Konstante sein. Die in einem Objektprogramm definierten Variablen können von allen anderen Objektprogrammen benutzt werden, wenn der Name des Objekts, in dem die Variable definiert wurde der Variablen getrennt durch einen Punkt vorangestellt wird.

Der Name einer Variablen kann die Buchstaben A-z, Ziffern und den Unterstrich beinhalten. Er muss mit einem Buchstaben beginnen, landesspezifische Buchstaben und Sonderzeichen sind nicht erlaubt (Beispiel: Var\_99).

Für jede Variable muss ein Typ festgelegt werden. Es können die gleiche Typen wie für Objekte verwendet werden. Zusätzlich zu diesen Typen können für Variablen auch Zeittypen (Uhrzeit, Datum) benutzt werden.

Bei Zuweisungen zwischen nicht gleichen Typen erfolgt die [Zuweisung](#) entsprechend der Reihenfolge der für diesen Typ definierten Zustände, unabhängig von der Bezeichnung des Zustands. Intern werden Zustände als Zahl entsprechend der Reihenfolge in der Definition des Typs behandelt. Bei Zuweisungen wird diese Zahl zugewiesen.

Es gibt eine vordefinierte Variable des Typs Uhrzeit für jedes Objekt. Der Name der Variablen ist **CT** (für **C**hange**T**ime). In dieser Variablen wird die jeweils letzte Änderungszeit eines Objekts gespeichert. Mit der Funktion [STOPPUHR](#)(Objektname.CT) kann ermittelt werden wie lange ein Objekt sich in seinem aktuellen Zustand befindet. Die aktuelle Schaltdauer eines Objekts kann auch mit der Funktion [SCHALTDAUER](#)(Objektname) ermittelt werden.

Bitte beachten Sie:

Wenn Sie eine Variable vom Typ Zahl definieren, gibt es zwei unterschiedliche Typen von Zahlen: Zahlen mit Komma (Gleitkommazahlen) und Zahlen ohne Komma. Von welchem Typ die Zahl ist, wird durch den Startwert festgelegt. Wird als Startwert eine Zahl mit Komma angegeben (z.B. 1,0) so wird die Variable als Zahl mit Kommastellen behandelt, andernfalls als Zahl ohne Kommastelle. Als Dezimaltrennzeichen kann anstelle eines Kommas auch ein Punkt (amerikanische Schreibweise für Gleitkommazahlen) verwendet werden.

Es macht bei Rechenoperationen und Zuweisungen einen grossen Unterschied wie eine Zahl definiert ist. Wenn einer Zahl ohne Kommastellen eine Zahl mit Kommastellen zugewiesen wird, werden die Kommastellen einfach abgeschnitten.

Wenn eine Rechenoperation nicht das gewünschte Ergebnis liefert, prüfen Sie wie die in der Anweisung verwendeten Zahlen definiert sind.

Es gibt vordefinierte Variablen, die verwendet werden können. Jedes Objekt hat beispielsweise eine Variable vom Typ Zeit mit dem Namen CT, in der die letzte Schaltzeit des Objekt abgespeichert ist. Um also die aktuelle Schaltdauer eines Objekts zu ermitteln kann man folgende Anweisung verwenden:

Anzeige:=Schaltdauer(Lampe.CT)

Ein Beispiel zur Verwendung von Variablen :

Die minimal und maximal-Temperatur sollen jede Stunde auf einer Anzeige ausgegeben werden.

Im Objektprogramm des Objekts TempWG sind die Variablen MIN und MAX als Zahl definiert.

Das Makro des Objekts TempWG:

wenn TempWG > MAX dann

MAX:=TempWG

endewenn

wenn TempWG < MIN dann

MIN:=TempWG

endewenn

Das Makro des Objekts Anzeige, das zu jeder vollen Stunde aktiviert wird:

Anzeige:="Min: "+TempWG.MIN+" Max: "+TempWG.MAX

Die maximale Länge einer Variablen vom Typ Zeichen ist 16000 Zeichen, für andere Zentralen 32000 Zeichen, was darüber hinausgeht wird abgeschnitten.

## 1.22.3. Zuweisungen

---

### Zuweisungen

Zustände bzw. Werte von [Objekten](#) oder [Variablen](#) können durch Zuweisungen geändert werden. Die grundsätzliche Form einer Zuweisung im Programm ist

**Ziel := Quelle**

wobei Ziel ein Objekt oder eine Variable sein kann. Quelle kann ein Objekt, eine Variable oder eine Konstante sein kann.

**Bei Zuweisungen von Werten unterschiedlichen Typs wird soweit möglich automatisch eine Konvertierung entsprechend des Typs des Ziels vorgenommen, d.h. das Ergebnis hat den Typ der Zielvariable-**

Wenn die Zielvariable z.B. eine Uhrzeit ist und es werden zwei Zeichen-Variablen mit Werten im Zeitformat addiert ist das Ergebnis die Summe der Zeiten, ist die Zielvariable aber eine Zeichenvariable werden nur die Texte aneinandergehängt.

Beispiele

```
ZeichenVariable1 := "14:00:00"  
Zeitvariable1 := "03:10:00"
```

Bei einer Zeichenvariablen als Zielvariable:

```
ZeichenvariableX := ZeichenVariable1 + Zeitvariable1  ist das Ergebnis  
"14:00:0003:10:00"
```

Bei einer Variable vom Typ Uhrzeit als Zielvariable:

```
ZeitvariableX := ZeichenVariable1 + Zeitvariable1      ist das Ergebnis "17:10:00"
```

Bitte beachten Sie, dass Zahlen mit bzw. ohne Kommastellen intern in unterschiedlichen Formaten dargestellt werden und es daher zu Problemen kommen kann wenn diese unterschiedlichen Zahlentypen einander zugewiesen werden. Bei Zuweisungen von Konstanten an Zahlenvariablen mit Nachkommastellen sollte die Konstante immer ein Komma beinhalten damit diese als Gleitkommazahl gespeichert wird, dabei können hinter dem Komma natürlich auch eine Null stehen. Es wird zwar bei der Ausführung versucht solche Probleme automatisch auszuschalten, wenn bestimmte Grössen überschritten werden funktioniert das aber nicht mehr (wenn eine Integer-Zahl grösser als 2147483647 einer Gleitkommavariablen zugewiesen wird, wird diese Zahl als negative Zahl interpretiert).

Bei Zuweisungen zwischen nicht gleichen selbstdefinierten Typen erfolgt die Zuweisung entsprechend der Reihenfolge der für diesen Typ definierten Zustände, unabhängig von der Bezeichnung des Zustands. Zum Beispiel muss bei der Definition neuer Typen mit den Zuständen AUS/AN darauf geachtet werden, dass der Zustand AUS immer zuerst definiert wird, damit bei Zuweisung an andere AUS/AN-Objekte der richtige Zustand übertragen wird.

Bei der Zuweisung von Zuständen können die Zustandstexte in Hochkomma oder der Index eines Zustands verwendet werden, wobei der erste Zustand den Index 0 hat. Bei Verwendung von Zustandstexten muss das Wort exakt so geschrieben werden wie der definierte Zustand in der Typdefinition, auch Gross/Kleinschreibung muss berücksichtigt werden.

Wenn eine Lampe also den Objekttyp Licht mit den Zuständen "aus" und "an" hat kann folgende Anweisung zum Ausschalten verwendet werden:

```
Lampe := "aus"  
oder auch  
Lampe := 0
```

Für bestimmte Objekttypen können auch umgangssprachliche Begriffe als Schlüsselwörter verwendet werden.

Bei der Verwendung von Schlüsselwörtern wie *einschalten* bei Zuweisungen oder *ausgeschaltet* in wenn-Bedingungen ist es erforderlich, dass AUS immer als erster Zustand definiert wird, da intern mit der numerischen Reihenfolge der Zustände gearbeitet wird. Durch diese Verfahrensweise ist es möglich, auch Zuweisungen zwischen Typen mit unterschiedlicher Bezeichnung für gleiche Zustände durchführen zu können. Bei der Definition der Zustände von Fenstern und Türen muss *offen* als erster Zustand und *zu* als zweiter Zustand definiert werden, damit die entsprechenden

Schlüsselwörter bei Zuweisungen und Bedingungen benutzt werden können.

Beispiele für Zuweisungen:

**LichtBad:=SchalterBad**

hat die gleiche Wirkung wie

**LichtBad wie SchalterBad**

**Gartenlicht einschalten**

hat die gleiche Wirkung wie

**GartenLicht:=1**

Wird einem Objekt ein Zustand als Text zugewiesen, so muss dieser in Hochkommas gesetzt werden und genau der Schreibweise der Typdefinition entsprechen, auch Gross/Kleinschreibung muss berücksichtigt werden.

Beispiel:

Anstatt

**Gartenlicht einschalten**

könnte man auch die Anweisung

**Gartenlicht:="an"**

verwenden, das Wort **an** muss genau wie in der Typdefinition geschrieben sein und in Hochkommas gesetzt werden.

Eine besondere Anweisung zum Einschalten Schaltaktoren ist die Anweisung

*Objekt einschalten für Zeitdauer*

als Zeitdauer kann die Zeit in Hochkommas oder als Variable angegeben werden.

Beispiel:

**Stehlampe einschalten für "00:30:00"**

oder

**Dauer1:="00:10:00"**

**Gartenlicht einschalten für Dauer1**

wobei Dauer1 eine Variable vom Typ *Uhr* oder *Zeichen* sein muss.

Ein weiteres wichtiges Schlüsselwort ist **ausschalten** um einen Schaltaktor auszuschalten.

Die Art der Rechenoperation bei Zuweisungen wird durch den Typ der Zielvariablen bestimmt. Wenn der Zieltyp vom Typ *Zeichen* ist, werden die Zeichen in der Zuweisung als Zeichenkette aneinander gereiht.

Wenn die Zielvariable vom Typ *Zahl* ist, wird eine normale Rechenoperation vorgenommen falls dies mit den angegebenen Operanden und Operatoren möglich ist, dabei können auch Klammern gemäss den üblichen Rechenregeln verwendet werden.

### **Rechenfunktionen für Zahlen**

Es können die vier Grundrechenarten benutzt werden und die Modulo-Operation. Der Operator für die Modulo-Operation ist das Prozentzeichen %. Das Ergebnis der Modulo-Operation ist der ganzzahlige Rest einer Division, wenn einer der Operanden eine Zahl mit Nachkommastellen ist werden diese abgeschnitten.

Weiterhin werden folgende mathematischen Funktionen unterstützt:

LN = natürlicher Logarithmus

EXP = Exponentialfunktion (Umkehrfunktion von LN)

LOG = dekadischer Logarithmus (zur Basis 10)

SIN = Sinus

ASIN = Arcus Sinus

COS = Cosinus

ACOS = Arcus Cosinus

TAN = Tangens

ATAN = Arcus Tangens

SQRT = Quadratwurzel

ABS = Absolutwert

Beispiel:

**Durchschnitt:=Summe / Anzahl**  
**Var1:=Wert1 \* (Wert2 + Wert3)**  
**QW:=SQRT(Var1)**

**Bitte unbedingt beachten:**

Innerhalb der Klammer einer Funktion dürfen keine Rechenoperationen stehen, nur eine Konstante oder eine Variable.

**Rechenfunktion bei Zuweisungen von Uhrzeiten**

Uhrzeiten können mit Hilfe der Operatoren + und - addiert bzw. subtrahiert werden.

Dabei ist unbedingt zu beachten, dass die Zielvariable den Typ Uhrzeit haben muss, siehe auch Erläuterung oben im zweiten Abschnitt.

Beispiel:

**AktuelleWeckzeit:=Weckzeit + "00:30:00"**

Anstatt von Uhrzeiten in Hochkommas kann auch die Zeit in Sekunden angegeben werden.

Diese Anweisung addiert 2 Minuten zu einer Uhrzeit:

**ZeitLicht1:=ZeitLicht1 + 120**

**Rechenfunktion bei Zuweisungen von Datumswerten**

Es ist auch möglich ein Datum durch eine Rechenoperation zu ermitteln.

Dabei ist zu beachten, dass die Zielvariable den Typ Datum haben muss.

Das Datum 11 Tage nach dem aktuellen Tag kann man z.B. einfach ermitteln mit der Anweisung:

**Zukunft:=Datum+11**

wobei die Zielvariable **Zukunft** vom Typ Datum sein muss damit die Berechnung korrekt erfolgt.

**Verknüpfungsfunktion bei Zuweisungen von Texten**

Mit Hilfe von Zuweisungen können zwei oder mehr Zeichenketten verbunden werden. Wenn in der Anweisung Operanden eines anderen Typs als Zeichenkette benutzt wird, werden diese soweit möglich automatisch konvertiert.

Beispiel:

**AnzeigeKueche:="Garage ist "+Garagentor**

Wenn das Objekt Garagentor die Typen *auf* und *zuhat*, und das Garagentor auf ist erscheint in der Anzeige **"Garage ist auf"**.

**AnzeigeBad:=Uhrzeit+" Uhr"**

Der Text in der Anzeige sieht dann so aus: **22:30:00 Uhr**

**Bitte beachten:**

Wenn das Ziel der Wert oder Zustand eines Hardwaregeräts ist, muss der zugewiesene Wert im Wertebereich und Format sein, der für das Hardwaregerät gültig ist. Dies zu prüfen liegt in der Verantwortung des Benutzers, da eine solche Prüfung weder im Editor noch im Compiler vorgenommen werden kann weil die Werte vom jeweiligen Gerät und bei demselben Gerät teilweise von der im Gerät installierten Firmware abhängig sein kann.



## 1.22.4. Bedingungen

---

### Bedingungen

Im Folgenden werden die Formen der Bedingungen in wenn-Anweisungen beschrieben, nicht die Wenn-Anweisung selber.

Details zu dieser Anweisung finden Sie im Kapitel [wenn-Anweisungen](#).

In wenn-Anweisungen wird aufgrund von Bedingungen entschieden welche weiteren Anweisungen ausgeführt werden. Eine Bedingung ist entweder WAHR oder FALSCH. Bedingungen können mit UND bzw. ODER verknüpft werden.

Bedingungen sind folgendermaßen aufgebaut:

**wenn** *Operand1 Vergleichsoperator Operand2*

Beispiel:

wenn Temperatur kleiner 21,5

Wenn die Operanden unterschiedlichen Typs sind, findet soweit möglich eine automatische Konvertierung statt. Dazu ist es erforderlich, dass das in der Bedingungs abgefragte Objekt bzw. die Variable immer Operand1 in der Vergleichsabfrage ist, wie es der üblichen Logik einer Abfrage entspricht. So kann der Operand2 dann auf den passenden Typ konvertiert werden wenn unterschiedliche Typen verglichen werden sollen.

Wenn einer der Operanden eine Konstante ist muss diese immer Operand2 sein damit eine Konvertierung auf den Typ des abzufragenden Objekts/Variablen durchgeführt werden kann.

Beispiel einer Uhrzeitabfrage:

**Richtig:** wenn Uhrzeit = "15:15:00" dann

**Falsch:** wenn "15:15:00" = Uhrzeit dann

#### Standard-Vergleichsoperatoren

= oder **gleich**

<> oder **ungleich**

< oder **kleiner**

> oder **groesser**

<= (kleiner oder gleich)

>= (grösser oder gleich)

#### Vergleichsbedingungen

Bei Vergleichen von Zuständen können die Zustandstexte in Hochkomma oder der Index eines Zustands verwendet werden, wobei der erste Zustand den Index 0 hat.

Die Abfrage ob eine Lampe ausgeschaltet ist kann also mit der Anweisung

wenn *Lampe* = "**aus**" dann

oder auch

wenn *Lampe* = 0 dann

durchgeführt werden.

#### Schlüsselwörter in Vergleichsbedingungen

Bei Vergleichen können für bestimmte Objekttypen auch Schlüsselwörter benutzt werden.

wenn *Lampe* **eingeschaltet**

hat die gleiche Wirkung wie

wenn *Lampe* = "**an**"

bzw.

wenn *Lampe* = 1

wenn *Lampe* **ausgeschaltet**

hat die gleiche Wirkung wie

wenn *Lampe* = "**aus**"

bzw.

wenn *Lampe* = 0

weitere besondere Schlüsselwörter für Bedingungen sind  
**GEÖFFNET** und **GESCHLOSSEN** für Tür/Fensterkontakte (**nicht für Rollläden, Jalousien, KeyMatic, Winmatic !**),  
also z.B.:  
wenn Fenster **geoeffnet** dann

Um bei einer Fernbedienungstaste festzustellen ob diese kurz oder lang gedrückt wurde kann z.B. folgende Abfrage verwendet werden:

```
WENN FBWohnenTaste2 = "lang" dann
    Stehlampe einschalten
SONST
    Stehlampe ausschalten
ENDEWENN
```

### Zeit-Vergleiche

Bei Zeitvergleichen ist zu beachten, dass die Uhrzeit im 5-Sekunden-Takt aktualisiert wird, die Sekunden der Uhrzeit bei der Prüfung auf Zeitgleichheit also immer durch 5 teilbar sein müssen. Wenn Uhrzeitvergleiche verwendet werden, muss sichergestellt sein, dass das Makro zu dem Zeitpunkt zu dem die Bedingung zutrifft könnte auch ausgeführt wird. Wenn also die Vergleichszeit volle Minuten hat, muss das Zeitintervall für das Makro mindestens auf 1 x in der Minute gesetzt sein, wenn es eine durch 5 Sekunden teilbare Uhrzeit ist muss das Ausführungsintervall auf 5 Sekunden gesetzt werden.

wenn **Uhrzeit** = "**HH:MM:SS**" .....

bei dieser Bedingung ist zu beachten, dass die Uhrzeit in Hochkommas gesetzt wird. Weiterhin müssen die Sekunden durch 5 teilbar sein, da die Bedingung sonst nie zutrifft.

wenn **Datum** = "**TT.MM.JJJJ**"

bei dieser Bedingung ist zu beachten, dass das Datum in Hochkommas gesetzt wird und das Jahr vollständig, also 4-stellig mit Jahrtausend angegeben werden muss.

wenn **Wochentag** = "*Wochentag*"

Prüfung auf einen Wochentag. Der Wochentag wird in Hochkommas gesetzt und muss mit einem Großbuchstaben und nachfolgenden Kleinbuchstaben geschrieben werden.

Beispiel : wenn **Wochentag** = "**Freitag**" dann

wenn **Monat** Vergleichsoperator *Zahl*

Beispiel :

wenn **Monat** groesser 4 und Monat kleiner 10 dann

bewirkt, dass die folgenden Anweisungen nur zwischen Mai und September ausgeführt werden.

### Vergleichsoperator =\* für Vergleiche mit Jokerzeichen

Beim Vergleich von Uhrzeit und Datum mit einer Konstanten können auch Jokerzeichen verwendet werden. Als Vergleichsoperator muss dann =\* verwendet werden. Operand2 muss eine Konstante in Hochkommas sein.

Beispiele:

Um Anweisungen jede volle und halbe Stunde auszuführen:

wenn **Uhrzeit** =\* "\*\*\*:00:00" oder **Uhrzeit** =\* "\*\*\*:30:00" dann

.....

Um Anweisungen immer am Ersten eines Monats auszuführen:

wenn **Datum** =\* "01.\*\*.\*\*\*\*" dann

### Bedingung für Bereiche

Mit diesem besonderen Bedingungsstyp ist es möglich z.B. Aktionen nur innerhalb bestimmter

Bereiche und Zeiträume auszuführen.

**wenn Prüfwert zwischen Startwert und Endwert dann**

.....

Beispiele für Zeiträume:

**wenn Uhrzeit/Datum zwischen "Zeitkonstante" und "Zeitkonstante" dann**

**wenn Uhrzeit zwischen "23:00:00" und "05:00:00" dann**

**wenn Monat zwischen 5 und 9 dann**

ist wahr von Mai bis September

**wenn Monat zwischen 10 und 4 dann**

ist wahr von Oktober bis April

**wenn Monat zwischen 6 und 6 dann**

ist nur im Juni wahr

Dieser Bedingungstyp kann nicht nur für Zeiträume, sondern grundsätzlich für alle Werte benutzt werden.

**wenn Temperatur zwischen 18,0 und 21,0 dann**

ist von 18,0 bis 21,0 Grad wahr

**Bitte beachten - grundsätzlich gilt:**

wenn der *Startwert* kleiner ist als der *Endwert* ist die Bedingung **wahr** wenn der *Prüfwert* im zu prüfenden Bereich liegt, also auch wenn er gleich mit dem *Start-* oder *Endwert* ist.

wenn der *Startwert* grösser ist als der *Endwert* ist die Bedingung **wahr** wenn der *Prüfwert nicht* zwischen *Startwert* und *Endwert* liegt.

Diese Verfahrensweise bietet die grösste Flexibilität und ergibt auch bei Bedingungen für Zeiträume mit Tageswechsel wie z.B. "*wenn Uhrzeit zwischen "23:00:00" und "05:00:00" dann*" das richtige Ergebnis, aber es muss genau darauf geachtet werden in welcher Reihenfolge Start- und Endwert angegeben werden.

**Vergleichsoperator =+ für Vergleiche mit Wochenmaske**

Mit diesem Vergleichsoperator ist es möglich Anweisungen nur an bestimmten Wochentagen ausführen zu lassen. Als Operand2 muß eine 7-stellige Konstante oder Variable vom Typ Zeichen bestehend aus Nullen und Einsen in Hochkommas verwendet werden. Jede Stelle steht für einen Wochentag, beginnend mit Sonntag. Die Bedingung ist wahr wenn an der Stelle des aktuellen Wochentags eine Eins steht.

Beispiel : Es soll geprüft werden, ob der aktuelle Tag ein Freitag, Samstag oder Sonntag ist.

**wenn Wochentag =+ "1000011" dann**

## 1.22.5. Anweisungen

---

### Anweisungen

Auf den folgenden Seiten werden die Anweisungen und Funktionen aufgeführt, die für die Programmierung zur Verfügung stehen.

Da von Anwendern, die mit Programmierungen vertraut sind, oft die Frage kommt, ob die Programmierung nicht wie bei Programmiersprachen üblich mit englischen Begriffen durchgeführt werden kann, hier eine Liste der möglichen englischen Anweisungen.

Ein Hinweis dazu noch:

In deutschen Makros wird beim Schalten von Aktoren üblicherweise die Form z.B.

MeinLicht einschalten

benutzt. Da die Wortstellung im Englischen so nicht üblich ist, kann auch die Form

Switchon MyLight

benutzt werden.

Hier eine Tabelle der Anweisungen und Funktionen mit den Begriffen, die für die Programmierung in englischer Sprache benutzt werden können:

#### Für Abfragen / Wenn- Bedingungen

WENN	IF
SONST	ELSE
ENDEWENN	ENDIF
DANN	THEN
GLEICH	EQUAL
UNGLEICH	NOTEQUAL
IST	IS
KLEINER	LESSER
GROESSER	GREATER
KLEINERGLEICH	LESSEREQUAL
GROESSERGLEICH	GREATEREQUAL
ZWISCHEN	BETWEEN
ODER	OR
UND	AND
NICHT	NOT
AUSGESCHALTET	SWITCHEDOFF
EINGESCHALTET	SWITCHEDON
GEOEFFNET	OPENED
GESCHLOSSEN	CLOSED
ENTRIEGELT	UNLOCKED
VERRIEGELT	LOCKED

#### Für Zuweisungen

EINSCHALTEN	SWITCHON
AUSSCHALTEN	SWITCHOFF
UMSCHALTEN	TOGGLE
RUNTERFAHREN	MOVEDOWN
RAUFFAHREN	MOVEUP
SCHLIESSEN	CLOSE
OEFFNEN	OPEN
ENTRIEGELN	UNLOCK
VERRIEGELN	LOCK
TUEROEFFNEN	OPENDOOR

#### Weitere Anweisungen

ABBRECHEN	CANCEL
ABFRAGE	REQUESTDEVICE
ABSPIELEN	PLAY
AKTIVIEREN	ACTIVATE
ALLEABFRAGEN	REQUESTALLDEVICES
ALLEWERTESICHERN	SAVEALLVALUES
ANZEIGEN	DISPLAY

AUFRUFEN	CALL
DEAKTIVIEREN	DEACTIVATE
DIMMEN	DIM
ERLEDIGT	DONE
GEHEZU	GOTO
GRUPPENZUWEISUNG	SETGROUP
LESEINTERNETSEITE	GETSITE
LESEWERTEDATEI	READVALUEFILE
LOESCHEDATEI	DELETEFILE
LÖSCHEANZEIGE	CLEARDISPLAY
RAUFDIMMEN	DIMUP
RUNTERDIMMEN	DIMDOWN
SCHLIESSEDATEI	CLOSEFILE
SCHLIESSEDATEIEN	CLOSEFILES
SCHREIBEDATEI	WRITEFILE
SENDE	SENDDEVICE
SENDEMAIL	SENDMAIL
SENDEMAILANHANG	SENDMAILATTACH
SENDESMS	SENDSMS
SETZEHISTORYDIFFERENZ	SETHISTORYSENSIBILITY
SETZEKONFIG	SETCONGIG
SETZEWERT	SETVALUE
SETZEZEITTABELLENWERT	SETTIMETABVAL
SICHTBAR	VISIBLE
SPRACHAUSGABE	SPEECHOUTPUT
STARTE	RUN
STARTEPROGRAMM	STARTWIN
STARTPROGRAMM	EXECUTE
STARTUHR	STARTWATCH
STOPPDIMMEN	STOPDIM
STOPPE	STOPDEVICE
TEMPERATURABSENKUNG	TEMPREDUCTION
UNSICHTBAR	INVISIBLE
VERLASSEN	RETURN
WARTE	WAIT

#### Funktionen

AKTIVIERT	ISACTIVE
BATTERIELEER	LOWBAT
DATEIVORHANDEN	FILEEXISTS
DATUM	DATE
FEIERTAG	HOLIDAY
GESCHALTET	SWITCHED
HOLEWERT	GETVALUE
JAHR	YEAR
MINUTE	MINUTE
MONAT	MONTH
MONATSTAG	DAYOFMONTH
SCHALTDAUER	SWITCHDURATION
SEKUNDE	SECOND
SONNENAUFANG	SUNRISE
SONNENUNTERGANG	SUNSET
STOPPUHR	STOPWATCH
STUNDE	HOURL
UHRZEIT	CLOCKTIME
UHRZEITSEKUNDE	CLOCKTIMESEC
VERBINDUNGSFEHLER	COMERROR
WOCHENTAG	DAYOFWEEK
ZEIT	TIME
ZUFALLSZEIT	RANDOMTIME

#### Funktionen zur Textbearbeitung

ERSETZEN	REPLACE
GROSSBUCHSTABEN	UPPERCASE
KLEINBUCHSTABEN	LOWERCASE
LESETEXTPAR	GETSTRPAR
LINKERTEIL	LEFTPART
OBJEKTBEZ	OBJECTDESCR
OBJEKTNAME	OBJECTNAME

RECHTERTEIL	RIGHTPART
TEXTLÄNGE	STRLEN
TEXTPOSITION	STRPOS
TEXTTEIL	COPYTEXT
TEXTZWISCHEN	TEXTBETWEEN

## 1.22.5.1. ABBRECHEN

---

### ABBRECHEN

Syntax :

**ABBRECHEN**(*Objektname*)

Mit dieser Anweisung wird die Ausführung des angegebenen Objektmakros abgebrochen, wenn es sich im WARTE-Status befindet (also aktuell eine WARTE-Anweisung aktiviert ist).

Beispiel:

ABBRECHEN(Lichtspiel)

## 1.22.5.2. ABFRAGE

---

### ABFRAGE

Syntax :

**ABFRAGE**(*Objektname*)

Mit dieser Anweisung kann der aktuelle Zustand/Wert der Hardware eines Objekts abgefragt werden. Normalerweise ist das nicht nötig, da Änderungen automatisch aktualisiert werden. In speziellen Anwendungen kann es sinnvoll sein diese Anweisung zu verwenden, sie sollte jedoch nicht zu häufig und möglichst nur bei sicherheitsrelevanten Objekten benutzt werden, da sie relativ zeitintensiv ist und es bei zu häufiger Verwendung zu Verzögerungen bei der normalen Kommunikation kommen kann.

Beispiel:

ABFRAGE(Tiefkuehltruhe)

#### **Bitte beachten Sie unbedingt:**

Diese Anweisung generiert bei jeder Ausführung eine Meldung an das Gerät.

Wenn es sich um ein Funkmodul handelt kann die hemmungslose und unsachgemäße Benutzung dieser Anweisung viele unnötige Funkmeldungen generieren, was dazu führen kann, dass das [Duty-Cycle-Konto](#) überläuft und alle Funkmeldungen an Aktoren zeitweise blockiert werden.

Bei batteriebetriebenen Geräten wird je nach Typ keine Meldung an das Gerät geschickt, wenn dieses nicht ständig empfangsbereit ist. In dem Fall wird der Wert gemeldet, den das Gerät in der WEB-UI hat. Dieser kann in seltenen Fällen vom tatsächlichen Wert des Geräts abweichen.

Diese Anweisung funktioniert nicht mit FHZ-Schnittstellen weil im Protokoll der Geräte eine solche Abfrage nicht vorgesehen ist.



### 1.22.5.3. AKTIVIEREN

---

#### AKTIVIEREN

Syntax :

**AKTIVIEREN(OBJEKT)**

Die Ausführung im Zeitintervall wird aktiviert. Natürlich muss ein Zeitintervall definiert worden sein, damit das Programm in diesem Intervall ausgeführt werden kann.

siehe auch [DEAKTIVIEREN](#)

## 1.22.5.4. ALLEABFRAGEN

---

### ALLEABFRAGEN

Syntax :

### ALLEABFRAGEN

Diese Anweisung startet eine Hintergrundabfrage bei der der aktuelle Zustand aller Hardwaremodule abgefragt wird.

#### **Bitte beachten Sie:**

Da diese Abfrage intern beim Start automatisch ausgeführt wird und danach die Synchronisation immer auch automatisch erfolgt wird diese Anweisung normalerweise nicht verwendet und ist nur in speziellen Fällen sinnvoll wenn der Verdacht besteht, dass die automatische Synchronisation gestört wurde.

Da diese Anweisung erheblichen Funkverkehr verursachen kann und Ressourcen benötigt sollte sie keinesfalls unnötigerweise verwendet werden.

#### **Bitte beachten Sie unbedingt:**

Diese Anweisung generiert bei jeder Ausführung eine Meldung an das Gerät.

Wenn es sich um ein Funkmodul handelt kann die hemmungslose und unsachgemäße Benutzung dieser Anweisung viele unnötige Funkmeldungen generieren, was dazu führen kann, dass das [Duty-Cycle-Konto](#) überläuft und alle Funkmeldungen an Aktoren zeitweise blockiert werden.

## 1.22.5.5. ALLEWERTESICHERN

---

### ALLEWERTESICHERN

Syntax :

#### **ALLEWERTESICHERN**

Mit dieser Anweisung werden alle Werte von Objekten und Variablen gesichert, so wie es beim normaler Beendigung der ExecEngine passiert wenn die Option "Aktuelle Werte der ExecEngine speichern/laden" aktiviert ist.

Die Datei wird auf einer Linux-Zentrale im Programmverzeichnis unter dem Namen <Projektname>### gespeichert.

Diese Datei wird bei einem Neustart der ExecEngine eingelesen und die in der Datei gespeicherten Werte werden als Startwerte für die Objekte gesetzt.

Ob beim Neustart eine Hardwareabfrage durchgeführt wird, kann im Reiter Allgemein des Hardwarefensters in Abhängigkeit vom Alter der Datei eingestellt werden.

## 1.22.5.6. ANZEIGEN

---

### ANZEIGEN

Syntax :

**ANZEIGEN**(OBJEKT)

Mit dieser Anweisung wird der Text und die über Variablen eingestellten Zusatzinformationen zu einem speziellen Anzeige-Hardwaremodul (z.B. [19-Tasten-Fernbedienung](#)) gesendet.

Für die 19-Tasten-Fernbedienung ist das Objekt, das als Parameter eingetragen werden muss das letzte Objekt der Fernbedienung,  
also zum Beispiel:

```
Anzeigen(FernbedienungW_18)
```

## 1.22.5.7. AUFRUFEN

---

### AUFRUFEN

Syntax :

**AUFRUFEN** (*OBJEKT*)

Diese Anweisung führt das angegebene Makro sofort aus. Nach Beendigung des aufgerufenen Makros wird die Ausführung mit der nächsten Anweisung fortgesetzt.

## 1.22.5.8. DEAKTIVIEREN

---

### DEAKTIVIEREN

Syntax :

**DEAKTIVIEREN**(OBJEKT)

Die Ausführung im Zeitintervall wird deaktiviert. Alle anderen Möglichkeiten das zur Aktivierung des Objektprogramms sind davon nicht betroffen. Das heisst das Programm kann weiterhin durch die Optionen "Ausführung bei Empfang", "Ausführen bei Änderung" und die Anweisungen [STARTE](#) oder [AUFRUFEN](#) aktiviert werden.  
siehe auch [AKTIVIEREN](#)

## 1.22.5.9. ERLEDIGT

---

### ERLEDIGT

Syntax :

**ERLEDIGT**(OBJEKT)

Das angegebene Objekt wird als erledigt gekennzeichnet, d.h. die nächste [GESCHALTET](#) - Abfrage gibt den Wert FALSCH zurück, falls sich der Zustand des Objekts nach der ERLEDIGT - Anweisung nicht wieder geändert hat.  
siehe auch GESCHALTET

## 1.22.5.10. GEHEZU

---

### GEHEZU

Syntax :

**GEHEZU** *ADRESSE*

.....  
*ADRESSE:*

.....

**Achtung: Bei falscher Anwendung dieser Anweisung können Endlosschleifen entstehen, durch die das aktuelle Makro ständig läuft ohne beendet zu werden.**

Eine gefährliche Anweisung, die im Normalfall nicht gebraucht wird und die man vermeiden sollte. Die GEHEZU-Anweisung bewirkt, dass die Programmausführung an der durch *ADRESSE* festgelegten Stelle fortgesetzt wird. Als Adresse kann ein beliebiger Name aus Buchstaben und Ziffern gewählt werden, das erste Zeichen muss ein Buchstabe sein. Die Adresse muss immer am Zeilenanfang beginnen und mit einem Doppelpunkt abgeschlossen werden. Innerhalb eines Objektprogramms muss eine Adresse eindeutig sein.

Diese Anweisung sollte möglichst vermieden werden, da bei falscher Anwendung Endlosschleifen entstehen können und die Ausführung des aktuellen [Objektprogramms](#) nicht beendet wird. Wenn die Sprungadresse vor der GEHEZU-Anweisung liegt, wird die Ausführung des Makros für einen Prozesstakt unterbrochen, damit andere Makros und Ein/Ausgaben ausgeführt werden können, und eine eventuelle Endlosschleife nicht das komplette System blockiert.

Beispiel:

WENN Uhrzeit > "22:00:00" DANN

**GEHEZU ABEND**

ENDEWENN

*Anweisungen*

.....

**GEHEZU ENDE**

**ABEND:**

*Anweisungen*

.....

**ENDE:**

*Anweisungen*



## 1.22.5.11. GETCCUSYSVAR

---

### GETCCUSYSVAR

Syntax :

**GETCCUSYSVAR**(Systemvariablenname,Zielvariable,IP\_der\_CCU)

Mit dieser Anweisung können Systemvariablen der CCU gelesen werden. Der Wert der Systemvariablen steht nach Ausführung der Anweisung in der angegebenen Variablen bzw. im Objekt zur Verfügung. Der Typ der als Ziel angegebenen Variablen oder des Objekts muss kompatibel mit dem Typ der Systemvariablen sein, soweit möglich wird der Typ konvertiert. Bei Systemvariablen des Typs "Logikwert" wird in einer Textvariablen "true" oder "false" zurückgegeben, in einer Zahlenvariablen 1 oder 0.

Mit dieser Anweisung können nicht nur Systemvariablen der CCU, sondern auch Systemwerte ausgelesen werden. Dazu wird als Variablenname der Systemwert mit einem führenden \* angegeben. Also z.B. "\*SunsetTime". Wichtig ist, dass Gross/Kleinschreibung des Systemwerts genau eingehalten werden.

Als IP der CCU wird für die aktuelle Zentrale normalerweise "127.0.0.1" angegeben, es kann allerdings auch die IP-Adresse einer beliebigen CCU im Netzwerk angegeben werden. Die Angabe "localhost" funktioniert bei der CCU nicht zuverlässig, daher sollte besser die localhost-IP "127.0.0.1" verwendet werden.

Die Ausführung dieser Anweisung wird im Hintergrund durchgeführt. Während das passiert werden andere Makros weiter ausgeführt. Nachdem die CCU-Logikschicht geantwortet hat wird die Ausführung des Makros fortgesetzt. Die Ausführung der Anweisung kann je nach Auslastung der CCU und individueller Umgebung einige Sekunden dauern.

Daher muss unbedingt darauf geachtet werden dass die Abfragen nicht ständig in einem kürzeren Zeitintervall durchgeführt werden. Wenn das Zeitintervall bei permanenten Abfragen unter Antwortzeit des Requests liegt, kommt es nach einiger Zeit zu einem Überlauf des Buffers, was zur Folge hat, dass Speicher blockiert wird und keine weiteren HTTP-Requests mehr möglich sind. Daher sollte das Abfrageintervall für ständige Abfragen keinesfalls unter 5 Sekunden gewählt werden.

Beispiele:

```
GetCCUSysVar( "Anwesenheit",CCU.Anwesenheit,"127.0.0.1")
```

```
GetCCUSysVar( "*SunriseTime",Text1,"192.168.0.81")
```

```
Varname:="Anwesenheit"
```

```
GetCCUSysVar(Varname,MeineVariable,"127.0.0.1")
```

Bei diesem Beispiel wird nicht der Name der CCU-Systemvariablen angegeben, sondern die CL-Variable **Varname**, in der der Name der CCU-Systemvariablen steht.

## 1.22.5.12. GETSITE

---

### GETSITE

Syntax :

**GETSITE**(*Webseite,Port,Zielvariable*)

Mit dieser Anweisung können beliebige Webseiten von einem Webserver aus dem lokalen Netzwerk oder aus dem Internet eingelesen werden indem ein http-Request erzeugt wird.

Als erster Parameter wird die aufzurufende Seite angegeben, falls ein anderer als der Standardport (80 bei http, 443 bei https) verwendet wird, kann dieser mit einem Doppelpunkt hinter der Adresse oder als zweiter Parameter der Anweisung angegeben werden. Falls der Port mit einem Doppelpunkt hinter Adresse angegeben wird, wird der zweite Parameter ignoriert.

Falls https-Seiten aufgerufen werden, muss der Seitenname mit den Zeichen "https://" beginnen.

Als zweiter Parameter wird der Port angegeben. Wenn die Standardports verwendet werden kann hier 0 eingesetzt werden, es wird dann automatisch der richtige Port benutzt.

Wenn der Port mit einem Doppelpunkt hinter dem Servernamen angegeben wird, wird der zweite Parameter ignoriert.

Als dritter Parameter wird die Variable oder das Objekt angegeben, in dem der Seiten Quelltext empfangen wird. Diese Variable bzw. das Objekt muss vom Typ Zeichen sein.

Bitte beachten Sie, dass die einzulesenden Daten der Webseite eventuell sehr gross sein können.

Die Ausführung des Befehls kann daher länger dauern und einiges an Ressourcen brauchen.

Achten Sie daher darauf, dass die Seiten, die aufgerufen werden möglichst nicht zu gross sind. Die maximale Grösse, die in der Zielvariablen gespeichert wird ist abhängig von der verwendeten Hardware.

Die Ausführung dieses Befehls kann je nach Grösse der Seite einige Sekunden dauern. Die Ausführung dieser Anweisung wird im Hintergrund durchgeführt. Während das passiert werden andere Makros weiter ausgeführt. Nachdem die Seite in der angegebenen Variablen gespeichert wurde wird die Ausführung des aktuellen Makros fortgesetzt.

Beispiele:

```
GetSite( "www.meine.seite.de/html/CCUSeite.html",0,Webseitentext )
```

```
GetSite( "www.meine.seite.de/html/CCUSeite.html",8081,Webseitentext )
```

```
GetSite( "www.meine.seite.de:8081/html/CCUSeite.html",0,Webseitentext )
```

```
GetSite( "https://meine.spezielleseite.de/html/Seite1.html",0,Webseitentext )
```

#### **Bitte beachten Sie:**

Die Anweisung GETSITE eröffnet einen separaten Thread im Hintergrund, damit das Warten auf eine Internetseite die übrige Ausführung nicht behindert und Zeitverzögerungen vermieden werden. Die Verwaltung der Threads geschieht automatisch durch das Betriebssystem und beansprucht einige Ressourcen. Daher sollte diese Anweisung nur sparsam und mit möglichst grossem Zeitabstand verwendet werden.

Je nach Zentralenhardware, Ressourcenauslastung und individueller Umgebung kann es eventuell sein, dass es bei ständiger Ausführung der Anweisung in kürzeren Zeitabstände zu Problemen kommt.

Wir empfehlen diese Anweisung nur sparsam einzusetzen und bei ständiger Verwendung mindestens 30 Sekunden Abstand zwischen den Aufrufen einzuhalten.

Auf vielfachen Wunsch gibt es noch eine spezielle Option der GETSITE-Anweisung zur Ansteuerung spezieller Geräte über http-Requests. Bei dieser Option wird nur die http-Anfrage gesendet und unmittelbar danach die Netzwerkverbindung zum Server geschlossen - ohne auf eine Antwort zu warten.

**Diese Option entspricht nicht den Spezifikationen zur Kommunikation mit einem Webserver und kann beim Server zu Fehlfunktionen führen !!**

Diese Option ist nur als Ausnahme zum Ansprechen spezieller Geräte gedacht, bei denen der Webserver keine korrekte Antwort generiert oder diese zu lange dauert, und sollte nur in speziellen

Ausnahmefällen verwendet werden. Im Normalfall führt die Verwendung dieser Option zu Fehlfunktionen beim Server.  
Um eine solche spezielle http-Anfrage abzusetzen wird einfach der Parameter "Zielvariable" weggelassen, da ja nicht auf eine Antwort gewartet wird.  
Bei dieser Option muss der Port als Parameter 2 angegeben werden, die Angabe des Ports mit Doppelpunkt hinter dem Servernamen funktioniert hier nicht.  
Die Option kann nur mit einfachen http://-Abfragen verwendet werden - https-Abfragen sind nicht möglich.

Beispiel:

```
GetSite("MeinSpeziellesGeraet/Pfad/Seite?Parameter",81)
```

## 1.22.5.13. GRUPPENZUWEISUNG

---

### GRUPPENZUWEISUNG

Syntax :

**GRUPPENZUWEISUNG**(*Typ, Gruppe, Wert*)

Mit dieser Anweisung kann mehreren Objekten gleichzeitig ein neuer Wert zugewiesen werden. Die Gruppe kann eine benutzerdefinierte [Gruppe von Objekten](#) ein oder ein Gerätecode. Bei Verwendung eines Gerätecodes wird allen Geräten mit diesem Code der angegebene Wert zugewiesen.

Die einzelnen Parameter haben folgende Bedeutung:

#### **Typ**

Typ der Gruppenzuweisung als Buchstabe. Möglich sind "G" für benutzerdefinierte Gruppen und "D" für Zuweisungen nach Gerätecode (Devicecode).

#### **Gruppe**

für den Typ "G" der Name der Gruppe als Zeichenkonstante oder Zeichenvariable.

für den Typ "D" der Gerätecode als Zahlenkonstante oder Variable.

Folgende Gerätecodes sind möglich:

1100 : Schaltaktoren

1200 : Dimmer des HomeMatic-Systems

1300 : Rollladenaktoren

10010 : Wandthermostate

#### **Wert**

Der Wert muss jeweils kompatibel zur angegebenen Gruppe sein. Das heisst für Binäraktoren muss der Wert 0 oder 1 sein oder der genaue Zustandstext der Aktoren.

#### **Beispiele:**

für Typ "G":

```
Gruppenzuweisung("G", "LichtEG", 1)
```

hat die gleiche Wirkung wie

```
Gruppenzuweisung("G", "LichtEG", "an")
```

```
Gruppenzuweisung("G", "AlleDimmer", 45)
```

```
Gruppenzuweisung("G", "RollladenEG", "halb")
```

für Typ "D":

```
Gruppenzuweisung("D", 1100, 0)
```

```
Gruppenzuweisung("D", 10010, 0)
```

kann benutzt werden um alle Wandthermostate auf "OFF" zu stellen

## 1.22.5.14. LESEWERTEDATEI

---

### LESEWERTEDATEI

Syntax :

**LESEWERTEDATEI**("Speicherort/Dateiname")

Mit dieser Anweisung können Objektwerte, die zuvor mit SCHREIBEDATEI in eine Datei geschrieben wurden wieder ausgelesen und den Objekten zugewiesen werden.

Eine Datei, die mit LESEWERTEDATEI gelesen wird muss im Linux-Format geschrieben worden sein, d.h. jede Zeile darf nur mit einem LF abgeschlossen werden. Dateien, die im Windows-Format vorliegen können nicht verwendet werden.

Eine Werte-Datei wird normalerweise mit der normalen SCHREIBEDATEI-Anweisung erstellt, es muss darauf geachtet werden als Zeilenende-Parameter 1 zu verwenden, damit die Datei im Linux-Format erstellt wird.

Der oder die Variablenwerte werden in die Datei geschrieben, wobei jeder Wert in einer separaten Zeile stehen muss.

Beispiel (die Variable WertText muss als Variable vom Typ Zeichen definiert sein):

```
Löschdatei("werte.txt")
WertText:="Schaltsteckdose=" + Schaltsteckdose
Schreibdatei("werte.txt",WertText,1)
WertText:="LichtDiele.Zeittabelle=" + LichtDiele.Zeittabelle
Schreibdatei("werte.txt",WertText,1)
```

in der Datei steht dann z.B.

Schaltsteckdose=an

LichtDiele.Zeittabelle=aus

Die Anweisung SCHREIBEDATEI hängt die neuen Zeilen normalerweise immer an die schon vorhandenen Zeilen an, es können beliebig viele Zeilen in die Datei geschrieben werden.

Wenn eine Datei nur einen Wert enthält sollte diese Datei auch nur eine Zeile enthalten, damit nicht unnötigerweise derselbe Wert mehrmals eingelesen wird. Um nicht vorher jedesmal die Anweisung LÖSCHEDATEI ausführen zu müssen gibt es die Option bei SCHREIBEDATEI ein \* hinter den Dateinamen zu setzen, um die Datei neu zu erstellen, so dass diese nur eine Zeile beinhaltet.

Ansonsten muss vor der Erstellung einer neuen Werte-Datei eine alte Datei mit gleichem Namen mit der Anweisung [LÖSCHEDATEI](#) gelöscht werden! Wenn das nicht getan wird, kommt es zwar nicht zu einer Fehlfunktion, da immer der letzte gelesene Wert aktuell ist, aber die Datei wird immer grösser und beim Lesen werden viele unnötige Zeilen verarbeitet.

Es können beliebig viele unterschiedliche Wertedateien erstellt werden, so ist es auch möglich für bestimmte Objekte/Variablen jeweils separate Dateien zu benutzen.

Mit der Anweisung LESEWERTEDATEI kann diese Datei gelesen werden, die gespeicherten Werte werden dann direkt in die Objekte bzw. Variablen eingesetzt. Die Zeile

Schaltsteckdose=an

in der Datei würde also bewirken, dass das Objekt "Schaltsteckdose" den Wert "an" bekommt.

Beispiel für die Anweisung:

```
Lesewertedatei("werte.txt")
```

## 1.22.5.15. LÖSCHEANZEIGE

---

### LÖSCHEANZEIGE

Syntax :

**LÖSCHEANZEIGE**(OBJEKT)

Mit dieser Anweisung werden die Zusatzvariablen eines speziellen Anzeigemoduls (z.B. [19-Tasten-Fernbedienung](#)) gelöscht.

Diese Anweisung bewirkt keine Aktualisierung des Hardwaremoduls, falls dieses aktualisiert werden soll, muss zusätzlich die Anweisung [ANZEIGEN](#) benutzt werden.

## 1.22.5.16. LÖSCHEDATEI

---

### LÖSCHEDATEI

Syntax :

**LÖSCHEDATEI**("Unterverzeichnis/Dateiname")

Mit dieser Anweisung können Dateien gelöscht werden. Das können selbst erstellte Textdateien, aber auch beliebige andere Dateien wie die Historydatei history.txt oder die Systemlog-Datei syslog.txt sein.

**Bitte beachten Sie unbedingt:**

Durch das beabsichtigte oder unbeabsichtigte Löschen von Dateien, die nicht selbst erstellt wurden, kann die Systemintegrität zerstört und die Funktionen der Zentrale beeinträchtigt werden.

Löschen Sie daher keinesfalls Dateien, von denen Sie nicht wissen wofür diese benutzt werden .

Achten Sie unbedingt darauf, dass das auch nicht versehentlich, etwa durch Angabe eines falschen Dateinamens, geschieht.

Aus Sicherheitsgründen ist das Löschen von Dateien nur innerhalb des festgelegten Standardverzeichnisses möglich, daher sollte aus Sicherheitsgründen immer ein Standardverzeichnis angegeben werden.

Beispiel:

LÖSCHEDATEI ( "BEWEGUNGEN.TXT" )

## 1.22.5.17. PLAY

---

### PLAY

Syntax :

**PLAY("Pfad\Datei")**

Die in Klammern angegebene Audio-Datei wird abgespielt.

Bitte beachten Sie:

Diese Funktion wird im Programm VisuWin ausgeführt, wenn kein Programm [VisuWin](#) aktiv ist wird diese Anweisung ignoriert.

Wenn die Datei nicht existiert oder es sich nicht um eine Wave-Datei handelt, so wird diese Anweisung ignoriert. Es wird keine Fehlermeldung ausgegeben, um die weitere Ausführung nicht zu behindern.

Beispiel:

wenn Klingelknopf gedrueckt dann

    wenn Uhrzeit zwischen "10:00:00" und "18:00:00" dann

**PLAY ( "C:\PROGRAMME\HOMEPUTER\WAV\TROMPETE.WAV" )**

    sonst

**PLAY ( "C:\PROGRAMME\HOMEPUTER\WAV\FLOETE.WAV" )**

    endewenn

endewenn



## 1.22.5.18. SCHLIESSEDATEIEN

---

### SCHLIESSEDATEIEN

Syntax :

### **SCHLIESSEDATEIEN**

Mit dieser Anweisung werden alle geöffneten Dateien, die mit der Anweisung SCHREIBEDATEI erstellt worden sind geschlossen.

Diese Anweisung muss verwendet werden bevor auf die Datei zugegriffen wird (z.B. zum email-Versand) und bevor die Zentrale ausgeschaltet wird oder der USB-Stick entfernt wird, andernfalls könnten Daten verloren gehen.

Beispiel:

SCHLIESSEDATEIEN

## 1.22.5.19. SCHREIBEDATEI

---

### SCHREIBEDATEI

Syntax :

**SCHREIBEDATEI**("Pfad/Dateiname","Text",Zeilenvorschubcode)

Mit dieser Anweisung können Textdateien erstellt werden.

Je nach Typ der verwendeten Zentrale werden unterschiedliche Verzeichnisse verwendet.

Als erster Parameter in der Klammer wird der Dateiname angegeben, Dieser Parameter ist vom Typ Zeichen und kann eine Konstanten oder eine Variable sein. Der Dateiname kann auch einen gültigen Pfad beinhalten, dieser muss dann aber schon vorhandenen sein, es wird kein neues Verzeichnis angelegt falls ein nicht existierendes Verzeichnis angegeben wird.

Die neue Zeile in der Datei wird an den vorhandenen Inhalt angehängt. Oftmals ist es erforderlich, dass die neue Zeile nicht angehängt wird, sondern als einziger Wert in der Datei stehen soll. Das ist z.B. der Fall wenn die Datei später mit [LESEWERTEDATEI](#) benutzt wird.

Um die Datei neu zu schreiben und den vorherigen Inhalt zu überschreiben, kann ein \* als letztes Zeichen an den Dateinamen angehängt, beispielsweise:

**SCHREIBEDATEI**("Alterwert.txt\*", Zeile, 1)

Als zweiter Parameter die Zeile, die in die Datei geschrieben werden soll, dieser Parameter ist vom Typ Zeichen und kann eine Konstante in Hochkomma oder eine Variable sein,

Als dritter Parameter wird angegeben welcher Zeilenvorschub für die ausgegebene Zeile verwendet werden soll. Dieser Parameter ist vom Typ Zahl und kann als Konstante oder Variable angegeben werden. Es gibt folgende Codes für den Zeilenvorschub:

0 = kein Zeilenvorschub, die nächste Ausgabe beginnt unmittelbar hinter dieser Zeile

1 = Linux Zeilenvorschub (LF), es wird nur ein Linefeed ausgegeben, mit einem Editor unter Windows wird normalerweise kein Zeilenvorschub dargestellt. Wenn die Datei mit [LESEWERTEDATEI](#) gelesen werden soll, muss dieser Code für den Zeilenvorschub verwendet werden.

2 = Windows-Zeilenvorschub (CR/LF), es werden die für Windows üblichen zwei Codes für den Zeilenvorschub ausgegeben, so dass die Datei nach einem Download unter Windows mit einem Texteditor betrachtet werden kann. Im Normalfall sollte diese Option benutzt werden.

Bei der CCU1 werden die Textdateien auf einem USB-Stick, der in der Zentrale eingesteckt sein muss, geschrieben.

Bei der CCU2 kann sowohl auf die SD-Karte wie auch auf einen USB-Stick geschrieben werden. Wenn auf die SD-Karte der CCU2 geschrieben werden soll muss diese über die WEB-UI initialisiert und eingebunden worden sein. Das passiert im Menüpunkt

**Einstellungen->Systemsteuerung->Speichereinstellungen.**

Auf welches Medium bzw. in welches Verzeichnis geschrieben wird, kann im Reiter Zentrale des Hardwarefensters festgelegt werden. Bei der CCU1 kann das Verzeichnis nicht verändert werden, da nur das vorgegebene Verzeichnis verfügbar ist. Bei der CCU2 kann ein beliebiges Verzeichnis als Standardverzeichnis angegeben werden. Voreingestellt ist der Standardpfad für die SD-Karte. Falls standardmässig auf einen USB-Stick geschrieben werden soll, muss der Pfad **/var/datadisk/** angegeben und der USB-Stick eingebunden werden.

Bei der CCU3 wird auf einen eingesteckten USB-Stick geschrieben, das Standardverzeichnis ist hier  
**/media/usb0/** .

Wenn in der Anweisung nur der Dateiname ohne Pfadangabe benutzt wird wird in das festgelegte Standardverzeichnis geschrieben. Wenn vor dem Dateinamen ein Pfad angegeben ist, der mit / beginnt, ist das eine absolute Pfadangabe und die Datei wird an den angegebenen Ort geschrieben. Wenn in ein Unterverzeichnis des Standardverzeichnisses geschrieben werden soll darf die Pfadangabe nicht mit / sondern muss mit dem Namen des Unterverzeichnisses beginnen. Das Unterverzeichnis muss existieren, also vorher angelegt worden sein.

Als erster Parameter in der Klammer wird der Dateiname (ggfs. mit Pfad) angegeben, als zweiter

Parameter die Zeile, die in die Datei geschrieben werden soll. Beide Parameter sind vom Typ Zeichen und können Konstanten oder Variablen sein. Der Dateiname kann auch einen gültigen Pfad beinhalten, dieser muss dann aber auf dem USB-Stick vorhandenen sein erstellt werden. Als dritter Parameter wird angegeben welcher Zeilenvorschub für die ausgegebene Zeile verwendet werden soll. Dieser Parameter ist vom Typ Zahl und kann als Konstante oder Variable angegeben werden. Es gibt folgende Codes für den Zeilenvorschub:

0 = kein Zeilenvorschub, die nächste Ausgabe beginnt unmittelbar hinter dieser Zeile

1 = Linux Zeilenvorschub (LF), es wird nur ein Linefeed ausgegeben, mit einem Editor unter Windows wird normalerweise kein Zeilenvorschub dargestellt.

2 = Windows-Zeilenvorschub (CR/LF), es werden die für Windows üblichen zwei Codes für den Zeilenvorschub ausgegeben, so dass die Datei nach einem Download unter Windows mit einem Texteditor betrachtet werden kann. Im Normalfall sollte diese Option benutzt werden.

Bevor ein USB-Stick entfernt wird muss die Anwendung SCHLIESSEDATEIEN ausgeführt werden, sonst gehen die zuletzt geschriebenen Daten bzw. Dateien verloren.

#### **Bitte beachten Sie bei der CCU2 unbedingt:**

Wenn ein USB-Stick verwendet wird muss dieser formatiert sein und **vor** dem Start der Zentrale eingesteckt werden. Nach dem Einstecken eines USB-Sticks muss die Zentrale ggfs. neu gebootet werden oder der USB-Stick muss über den entsprechenden Punkt des Webservers in der WEB-UI eingebunden werden.

Menüpunkt Einstellungen->Systemsteuerung->CL-Studio->Dateien bearbeiten.

Über diese WEB-Server-Seite können die Dateien bearbeitet werden und es kann ein Download der Dateien durchgeführt werden.

Bevor Dateien bearbeitet oder downgeloadet werden muss die Anweisung SCHLIESSEDATEIEN ausgeführt werden, damit alle geschriebenen Daten in den Dateien vorhanden sind. Ein USB-Stick sollte nicht entfernt werden bevor die Dateien durch Download gesichert wurden.

Bei der Angabe eines Pfads, bei dem in andere Verzeichnisse als die Standardverzeichnisse für SD-Karte oder USB-Stick geschrieben wird kann das Betriebssystem der Zentrale und die Systemstabilität beschädigt werden. Auf andere Verzeichnisse als die angegebenen Standardverzeichnisse kann über das Webserverprogramm nicht zugegriffen werden, d.h. Dateien an anderen Orten können nicht über das Browserprogramm gewartet werden. Benutzen Sie andere Verzeichnisse als die Standardverzeichnisse nur wenn Sie genau wissen was Sie tun und über entsprechende Fachkenntnisse verfügen.

Wenn das Standardverzeichnis für USB-Stick der SD-Karte angegeben wird und diese nicht eingebunden worden sind, wird in den RAM der Zentrale geschrieben.

#### **Wenn die Dateien dort zu viel Platz beanspruchen wird die Funktionalität der Zentrale beeinträchtigt und es kommt zu Fehlfunktionen oder Absturz !**

Die Anweisung SCHREIBEDATEI sollte also möglichst immer nur Dateien auf SD-Karte oder USB-Stick schreiben. Andernfalls müssen die Dateien regelmässig kontrolliert und ggfs. gelöscht werden. Andernfalls kann es passieren, dass zu viel Speicher der Zentrale belegt wird und ein normaler Betrieb nicht mehr möglich ist.

Beispiel:

(Zeile ist im Makro als Variable vom Typ Zeichen definiert)

```
Zeile:="Uhrzeit " + Uhrzeit + ": Bewegungsmelder Haustür"  
SCHREIBEDATEI("BEWEGUNGEN.TXT",Zeile,2)
```

#### **Beschreibung der Verwendung dieser Anweisung wenn der PC als Zentrale benutzt wird:**

In welches Verzeichnis standardmässig geschrieben wird, kann im Reiter Zentrale des Hardwarefensters festgelegt werden.

Es kann auch Dateiname angegeben werden, der einen Pfad beinhaltet.

Beispiel:

```
SCHREIBEDATEI("c:\daten\CL\MeineDatei.txt","Das ist der Text",2)
```

## 1.22.5.20. SENDE

---

### SENDE

Syntax :

**SENDE**(*Objektname*)

Mit dieser Anweisung kann der aktuelle Zustand eines Objekts an den zugeordneten Empfänger gesendet werden, ohne dass der Zustand des Objekts sich geändert hat.

Gesendet wird nicht bei Ausführung des Befehls sondern immer erst nach Beendigung des aktuellen Makros oder bei einer WARTE-Anweisung.

Diese Anweisung sollte auf keinen Fall in Makros verwendet werden, die in einem Zeitintervall aufgerufen werden.

Die Anweisung SENDE funktioniert nicht bei bei Geräten, bei denen die Einstellung nicht aufgrund des aktuellen Objektzustands vom Geräte selbst vorgenommen werden kann.

Die Anweisung kann z.B. nicht für FS20-Rollallden/Markisten-Aktoren benutzt werden, da hierbei die Einstellung über eine Prozedur in der Zentrale mit mehreren Funkmeldungen vorgenommen wird.

#### **Bitte beachten Sie unbedingt:**

Diese Anweisung generiert bei jeder Ausführung eine Meldung an das Gerät.

Wenn es sich um ein Funkmodul handelt kann die hemmungslose und unsachgemässe Benutzung dieser Anweisung viele unnötige Funkmeldungen generieren, was dazu führen kann, dass das [Duty-Cycle-Konto](#) überläuft und alle Funkmeldungen an Aktoren zeitweise blockiert werden.

## 1.22.5.21. SENDEMAIL

---

### SENDEMAIL

Syntax :

**SENDEMAIL**("Zieladresse", "Betrefftext", "Erste Zeile<<zweite Zeile<<dritte Zeile"<,"Anhang1;Anhang2">)

Mit dieser Anweisung können Mails versendet werden. Voraussetzung ist, dass im Netzwerk, in dem die Zentrale angeschlossen ist, eine Verbindung zum Internet besteht oder automatisch aufgebaut wird sobald die Zentrale eine Verbindungsanforderung zu einem Mail-Server schickt. Als erster Parameter der Anweisung wird die Zieladresse angegeben, als zweiter Parameter folgt der Betrefftext, als dritter Parameter der Text der Mail. Wenn mehrere Zeilen Text gesendet werden sollen, so können die <<-Zeichen als Zeilenvorschub-Zeichen verwendet werden, der Text darf maximal 8000 Zeichen haben.

Diese ersten drei Parameter müssen angegeben werden, jeder Parameter muss gültige Daten enthalten und darf nicht leer sein.

Der vierte Parameter ist optional, es können mehrere Dateien getrennt durch Semikolon angegeben werden, die als Anhang mit der mail geschickt werden.

Wenn zu einer Datei kein Pfad angegeben wird, wird der Standardpfad benutzt, der im Hardwarefenster im Reiter Zentrale als Pfad für eigene Dateien auf der Zentrale angegeben ist. Mit der Funktion [MAILLOG](#), kann der Status der jeweils letzten fünf versendeten Mails überprüft werden.

Wenn Mail-Server, Absenderadresse oder einer der ersten drei Parameter nicht angegeben oder leer ist wird die Mail nicht versendet. Im Maillog erscheint dann der Fehler 999.

Im Rahmen neuer Sicherheitsmaßnahmen müssen bei vielen Providern Absenderadressen und Kontonamen gleich sein um Täuschungen durch falschen Absender zu verhindern. Wenn Mails nicht versendet werden kann das eine Ursache sein, die geprüft werden sollte.

Bitte beachten Sie, dass jedesmal wenn die Anweisung ausgeführt wird eine Mail gesendet wird. Die Programmierung muss so aufgebaut sein, dass nicht ungewollt viele Mails versendet werden. Werden Mails z.B. als SMS auf das Handy gesendet, können Fehler in der Programmierung, die ungewollt viele Mails erzeugen, zu erheblichen Kosten führen. Es müssen also Vorkehrungen getroffen werden, um ein zu häufiges Versenden von Mails zu verhindern.

Im folgenden Beispiel verhindert die Abfrage nach MaxTemp, dass bei konstanter Temperatur z.B. über einem Grenzwert dauernd gesendet würde.

Beispiel:

```
sendemail("meineAdresse@Mein-Provider.com", "Temperaturprotokoll", "Temperaturen von heute", "Temperaturen.txt")
```

```
wenn TempASH550 > MaxTemp dann  
  Mailtext:="Info Treibhaus:<<Temperatur ist " + TempASH550 + " Grad"
```

```
sendemail("Meine.BueroAdresse@Mein-Provider.com", "Temperaturinfo", Mailtext)  
  MaxTemp:=TempASH550 + 0.5  
endewenn
```

Es wird jetzt immer nur dann gesendet, wenn die Temperatur um mehr als 0,5 Grad gestiegen ist. Würde die Abfrage dagegen z.B. lauten

```
wenn TempASH550 > 25 dann
```

so würde bei jeder Makroausführung, bei der eine höhere Temperatur als 25 Grad erreicht ist eine Mail gesendet, also z.B. alle paar Minuten wenn ein Temperaturwert empfangen wird, oder gar alle 5 Sekunden, wenn das Makro in diesem Zeitintervall ausgeführt würde.

Eine andere Möglichkeit ist, das Versenden von Mails z.B. über "virtuelle" Objekte zu steuern. Im folgenden Beispiel würde z.B. nur jede halbe Stunde eine Mail verschickt:

```
wenn mailSchalter eingeschaltet und Schaltdauer(mailSchalter) > "00:30:00"
```

```

dann
    mailSchalter ausschalten
endewenn
wenn mailSchalter ausgeschaltet dann
    sendemail("Meine.BueroAdresse@Mein-Provider.com", "Wasseralarm", "Wasser
im Keller !")
    mailSchalter einschalten
endewenn

```

Beachten Sie bitte, dass die Funktion SCHALTDauer nur für Objekte, nicht aber für Variablen verwendet werden kann.

Eine weitere Möglichkeit ist z.B. die Zeiten zwischen zwei Mails über STARTUHR und STOPPUHR zu begrenzen.

```

wenn STOPPUHR(LetzteMail) > "00:30:00" dann
    sendemail("Meine.BueroAdresse@Mein-Provider.com", "Wasseralarm", "Wasser
im Keller !")
    STARTUHR(LetzteMail)
endewenn

```

**Bitte unbedingt beachten:**

Testen Sie Ihre Programmierung und überprüfen ob nicht unbeabsichtigt viele Mails versendet werden. Beachten Sie, dass Mails Kosten verursachen können, z.B. durch Verbindungsentgelte oder SMS auf ein Handy.

## 1.22.5.22. SENDESMS

---

### SENDESMS

Syntax :

**SENDESMS**("TelefonnummerEmpfänger", "Text der SMS", "Absendername", Statusvariable)

Mit dieser Anweisung können SMS versendet werden.

Zum Versand von SMS ist ein SMS-Konto mit Guthaben erforderlich.

Zu diesem Konto gibt es einen SMS-Code, der im Reiter *Sicherheit* des Einstellungsfensters angegeben wird.

Eine SMS kann maximal 160 Zeichen haben.

Das SMS-Guthaben kann im CL-Shop unter diesem Link bestellt werden:

[SMS für CL-Software](#)

### **Beschreibung der Anweisung SENDESMS:**

Als erster Parameter der Anweisung wird die Mobilfunknummer des Handys angegeben, an das die SMS gesendet werden soll.

Wenn diese Nummer eine Festnetznummer im Inland ist wird dort angerufen und die SMS vorgelesen.

Je nach Telefonanbieter des Festnetzanschlusses kann es sein, dass dieses Feature nicht zur Verfügung steht.

Als zweiter Parameter wird der Text der SMS angegeben. Der Text muss aus Buchstaben, Ziffern und Leerezeichen bestehen, zulässige Sonderzeichen sind ! ? \*, Komma, Punkt, Doppelpunkt, Semikolon, Klammern und einfaches Anführungszeichen.

Die Zeichen % & + # \$ werden in Leerzeichen umgewandelt.

Umlaute werden in eine andere Schreibweise umgewandelt ä zu ae, Ö zu Oe usw.

Bei Verwendung anderer als der aufgeführten Sonderzeichen werden diese in ? umgewandelt, können jedoch auch dazu führen, dass die SMS nicht versendet werden kann.

Dritter Parameter ist der Absendername, unter diesem Namen erscheint die SMS auf dem Ziel-Handy.

Vierter Parameter der Anweisung ist eine Variable oder ein Objekt vom Typ Zeichen, welches selbst erstellt werden muss. In diese Variablen wird eine Statusmeldung geschrieben sobald die SMS an den SMS-Server übergeben wurde.

Es werden 3 Parameter, getrennt durch Komma in die Statusvariable geschrieben.

Erster Parameter ist eine dreistellige Zahl.

100 bedeutet, dass die SMS ohne Fehler SMS-Server übertragen wurde, eine andere Zahl oder Text zeigen an, dass ein Fehler aufgetreten ist.

Der zweite Parameter gibt an wieviele SMS-Credits verbraucht wurden und ist abhängig von der Länge der gesendete SMS.

Wenn die Nachricht kleiner als 160 Zeichen ist wird nur 1 SMS-Credit berechnet, über 160 Zeichen 2 Credits.

Eine SMS kann maximal 320 Zeichen haben, wenn der Text mehr Zeichen hat werden diese abgeschnitten.

Wenn aufgrund eines Fehlers in den Parametern eine SMS nicht versendet werden konnte, erfolgt keine Belastung der Credits, das gilt jedoch nicht wenn ungültige Mobilfunknummern verwendet werden, da dann trotzdem ein (erfolgloser) Versand erfolgt.

Den aktuellen Kontostand Ihrer SMS-Credits können Sie sich auf folgender Seite anzeigen lassen:

<https://cl-control.de/sms/smskonto.php>

### **Allgemeine Hinweise:**

Eine SMS wird nicht bei Ausführung des Befehls versendet um Verzögerungen der Ausführung durch Internetzugriffe zu vermeiden. Das Versenden passiert nach Ausführung des Befehls im Hintergrund, daher steht der Status des SMS-Versands immer erst einige Sekunden nach Ausführung des Befehls in der Variablen zur Verfügung.

Wann die SMS empfangen wird ist vom Mobilfunknetz bzw. der Mobilfunkverbindung des Empfängers abhängig.

**Bitte beachten Sie:**

Wenn versucht wird eine SMS an eine ungültige Mobilfunknummer versendet wird erscheint keine Fehlermeldung, weil die Unzustellbarkeit der Mail erst bei der späteren Verarbeitung im Mobilfunknetz festgestellt wird. Da es bei einem solchen Fehler keine Rückantwort gibt ist eine entsprechende Gutschrift auf dem SMS-Konto nicht möglich.

```
wenn TempTiefkuehl > -10 und SMSTiefkuehlMerker ausgeschaltet dann
    SMSText:="Achtung Temperatur Tiefkühlschrank ist " + TempTiefkuehl + "
    Grad"
    sendesms("01712345678", SMSText, "MeinHaus", SMSStatus)
    SMSTiefkuehlMerker einschalten
endewenn
```

**Bitte unbedingt beachten:**

Sie müssen durch geeignete Programmierung sicherstellen, dass nicht ungewollt zu viele SMS verschickt werden, also z.B. einen Zähler oder Merker wie im zweiten Beispiel verwenden. Ein solcher Merker kann dann manuell oder zeitgesteuert wieder ausgeschaltet bzw. auf 0 gesetzt werden.

Sie sollten Ihre Programmierung testen und überprüfen ob nicht unbeabsichtigt viele SMS versendet werden.



## 1.22.5.23. SETCCUSYSVAR

---

### SETCCUSYSVAR

Syntax :

**SETCCUSYSVAR**(Systemvariablenname, Wert, IP\_der\_CCU)

Mit dieser Anweisung können Systemvariablen der CCU verändert werden. Der Type des angegebenen Werts muss dem Typ der Systemvariablen entsprechen.

Bei Systemvariablen des Typs "Logikwert" kann eine Variable des Typs Zahl ohne Komma verwendet werden, der Wert 0 entspricht "falsch" der Wert 1 entspricht "wahr".

Der Wert kann eine Konstante oder eine Variable sein.

Als IP der CCU wird für die aktuelle Zentrale normalerweise "127.0.0.1" angegeben, es kann allerdings auch die IP-Adresse einer beliebigen CCU im Netzwerk angegeben werden. Die Angabe "localhost" funktioniert bei der CCU nicht zuverlässig, daher besser die "localhost"-IP "127.0.0.1" verwenden.

Die Ausführung dieser Anweisung wird im Hintergrund durchgeführt. Während das passiert werden andere Makros weiter ausgeführt. Nachdem die CCU-Logikschicht geantwortet hat wird die Ausführung des Makros fortgesetzt. Die Ausführung der Anweisung kann je nach Auslastung der CCU und individueller Umgebung einige Sekunden dauern.

Daher muss unbedingt darauf geachtet werden, dass die Abfragen nicht ständig in einem kürzeren Zeitintervall durchgeführt werden. Wenn das Zeitintervall bei permanenten Abfragen unter Antwortzeit des Requests liegt kommt es nach einiger Zeit zu einem Überlauf des Buffers, was zur Folge hat, dass Speicher blockiert wird und keine weiteren HTTP-Requests mehr möglich sind. Daher sollte nicht mehr als eine Abfrage innerhalb von 5 Sekunden gemacht werden.

Beispiele:

```
SetCCUSysVar( "Anwesenheit",1,"127.0.0.1" )
```

```
SetCCUSysVar( "MeinText", "Das ist der Text", "192.168.0.55" )
```

## 1.22.5.24. SETZEHISTORYDIFFERENZ

---

### SETZEHISTORYDIFFERENZ

Syntax :

**SETZEHISTORYDIFFERENZ**("Variablenname",Differenzwert)

Wenn für ein Objekt oder eine Variable die Historyfunktion aktiviert ist, wird jeder neue Wert, der sich vom vorigen Wert unterscheidet, in die History-Datei geschrieben, Bei Temperaturen z.B. auch wenn diese sich nur um 0,1 Grad schwanken. Dadurch entstehen sehr viele im Normalfall nicht benötigte Einträge, die die Historydatei unnötig aufblähen. Oftmals sind es gerade bei Temperaturen über einen längeren Zeitraum nur minimale Schwankungen von 0,1 Grad nach oben und unten, die viele einzelne Einträge verursachen.

Daher werden bei Werten mit Gleitkommazahlen (z.B. Temperaturen) nur neue Einträge in die History geschrieben, wenn ein Wert sich um 0,2 oder mehr vom letzten Historywert unterscheidet. Bei ganzen Zahlen (z.B. Luftfeuchtigkeit) wird nur ein neuer Historyeintrag geschrieben, wenn er sich um mehr als 1 vom vorigen Wert unterscheidet. Dadurch wird die Grösse der Historydatei erheblich vermindert.

Mit der Anweisung SETZEHISTORYDIFFERENZ kann in einem Makro (vorzugsweise in einem INIT\_ Makro) der für das Schreiben von Historydaten relevante Differenzwert individuell festgelegt werden. Wenn ein Wert 0 gesetzt wird, wird jede Veränderung in die Historydatei geschrieben, bei grösseren Werten nur wenn der mit der Anweisung festgelegte Wert erreicht oder überschritten wird. Bei Temperaturwerten kann es durchaus sinnvoll sein den Differenzwert auf 0,5 zu setzen, bei der Luftfeuchtigkeit auf einen Wert grösser 2.

#### **Bitte unbedingt beachten:**

Der erste Parameter ist eine Zeichenkonstante, diese muss in Hochkomma gesetzt werden!

Beispiel:

```
SetzeHistoryDifferenz("Raumthermostat1.Temperatur", 0.5)
SetzeHistoryDifferenz("Raumthermostat1.Luftfeuchtigkeit", 3)
SetzeHistoryDifferenz("TemperaturAquarium", 0.1)
```

## 1.22.5.25. SETZEKONFIG

---

### SETZEKONFIG

Syntax :

**SETZEKONFIG**(*Parameter*)

Mit dieser Anweisung können spezielle Konfigurationseinstellungen und Aktionen während der Laufzeit vorgenommen werden.

Diese Anweisung funktioniert nicht auf einer CCU1.

Dazu gibt es folgende Parameter:

SETZEDT zur Einstellung welches Dezimaltrennzeichen verwendet werden soll bei der Zuweisung von Fließkommazahlen an Textvariablen bzw. bei der Ausgabe von Zahlenwerten.

Die Zeichenfolge

SETZEDT=.

setzt den Punkt als Dezimaltrennzeichen, bei

SETZEDT=,

wird das Komma als Dezimaltrennzeichen festgelegt.

Manchmal kann es sinnvoll sein das Dezimaltrennzeichen nur kurz für bestimmte Zuweisungen innerhalb eines Makros umzustellen.

Beispiel:

```
SETZEKONFIG( "SETZEDT=." )
```

```
MeinTextPar:=NeueTemperatur + ", XY, 3"
```

```
SETZEKONFIG( "SETZEDT=," )
```

Die Parameter BEENDEN oder ENDPROGRAM bewirken eine Beendigung der ExecEngine nach Ausführung der Anweisung.

Bei einer Linux-Zentrale wird die ExecEngine durch einen Watchdog automatisch neu gestartet.

Bei einem PC als Zentrale erfolgt kein automatischer Neustart.

Beispiel:

```
SETZEKONFIG( "BEENDEN" )
```

## 1.22.5.26. SETZEWERT

---

### SETZEWERT

Syntax :

**SETZEWERT**(*Objektname, Wertecode, Wert*)

Mit dieser Anweisung können verschiedene Einstellungen in den Hardwaremodulen des HomeMatic-Systems vorgenommen werden.

Für das FHZ-System gibt es auch einige Module, für die SETZEWERT verwendet werden kann, weitere Infos dazu finden Sie unter [FHZ-System](#).

Diese Anweisung erzeugt immer einen Sendebefehl und sollte nur sparsam verwendet werden, Sie darf keinesfalls in Makros erwendet werden, die in kurzen Zeitintervallen ausgeführt werden, das könnte den DUTY-CYCLE an seine Grenzen bringen und so das ganze System blockieren.

Es muss auch unbedingt beachtet werden, dass bei Einstellungen über diese Anweisung nur ein Sendebefehl erzeugt wird. Die Werte und Zustände von Objekten und Variablen werden nicht direkt beeinflusst, sondern erst durch eventuelle Rückmeldungen von den Hardwaremodulen aktualisiert. Das kann zur Folge haben, dass Hardware und Visualisierung nicht oder nicht zeitgleich synchron sind und Zustandsabfragen in wenn-Bedingungen nicht aktuell sind. Daher sollte diese Möglichkeit den Zustand von Hardwaremodulen zu ändern nur in Ausnahmefällen benutzt werden.

Die Anweisung sollte nur von erfahrenen Benutzern verwendet werden, die unsachgemässe Verwendung dieser Anweisung kann zu Fehlfunktionen führen.

Die Wertecodes können Konstanten oder auch Variablen vom Typ Zeichen sein.

Die Wertecodes müssen in Grossbuchstaben geschrieben sein, es kann auch die englische Bezeichnung (in Klammern) verwendet werden.

Mit dieser Anweisung wird die Methode setValue aus dem XMLRPC-Protokoll des BidCoS-Services aufgerufen. Das funktioniert nicht nur für die im unteren Teil angegebenen Wertecodes, sondern auch für andere, es kann aber setValue-Kombinationen geben, die nicht funktionieren z.B. weil ein anderer Kanal erforderlich ist als der im Objekt hinterlegte.

Es liegt in der Verantwortung des Benutzers zu prüfen ob die verwendeten Wertecodes und Werte zu dem Gerät passen, an das gesendet wird.

Der Compiler kann nicht prüfen ob die benutzten Wertecodes und Werte plausibel und richtig sind!

Beispiel zum einschalten den BOOST-Modus bei Thermostaten (wird von Thermostaten der ersten Serie nicht unterstützt):

```
SETZEWERT(ThermostatName, "BOOST_MODE", 1)
```

#### **Bitte unbedingt beachten:**

Bis auf den ersten Parameter müssen alle in Hochkomma stehen und Grossbuchstaben haben. Nur bei ganzen Zahlen muss der letzte Parameter nicht in Hochkomma stehen.

Diese Option funktioniert nur mit dem HMIP-Rauchmelder, nicht mit dem normalen HomeMatic--Rauchmelder.

#### **Für folgende Wertecodes können momentan deutsche Schlüsselwörter verwendet werden:**

EINSCHALTZEIT (ON\_TIME), Einheit ist Sekunden

Beispiel:

```
Setzewert(Stehlampe, "EINSCHALTZEIT", 30)
Stehlampe einschalten
```

Mit diesem Schlüsselwert kann z.B. für Schaltaktoren und Dimmer die Einschaltzeit eingestellt werden. Es ist keine Dauer-Einstellung, sie ist nur gültig für den nächsten Einschaltbefehl an das Modul. Bei dieser Möglichkeit ein Gerät für eine bestimmte Zeit einzuschalten, wird der interne Timer des Moduls benutzt, d.h. das Gerät schaltet nach der angegebenen Anzahl von Sekunden aus, auch wenn die Zentrale nicht mehr aktiv ist. Die Verwendung dieser Methode kann bei sicherheitskritischen Funktionen vorteilhaft sein.

VERZÖGERUNG (RAMP\_TIME), Einheit ist Sekunden

Beispiel:

```
Setzewert(Dimmer, "VERZÖGERUNG", 8)
Dimmer:=70
```

Mit diesem Schlüsselwert kann z.B. die Dimmgeschwindigkeit von Dimmern gesteuert werden.  
ALTE STUFE (OLD\_LEVEL)

Beispiel:

```
Setzewert(Dimmer, "ALTE STUFE", 1)
```

Mit diesem Schlüsselwert kann ein Dimmer auf seine letzte Dimmstufe vor dem Ausschalten gesetzt werden. Die 1 ist als fester Wert erforderlich.

STUFE (LEVEL)

Beispiel:

```
Setzewert(Dimmer, "STUFE", "0.6")
```

Mit diesem Schlüsselwert kann die Helligkeit eines Dimmers eingestellt werden. Der Wert ist ein Dezimalwert zwischen 0 und 1, als Parameter in Form einer Konstante oder Variable vom Typ Zeichen.

Diese Möglichkeit der Dimmersteuerung sollte nur in Ausnahmefällen benutzt werden, besser ist es den Dimmer über eine Zuweisung in einem Makro (z.B. Dimmer:=60) einzustellen.

ZUSTAND (STATE)

Beispiel:

```
Setzewert(Stehlampe, "ZUSTAND", 1)
```

Mit diesem Schlüsselwert kann ein Binäraktor geschaltet werden. Bei Wert 1 wird eingeschaltet, bei Wert 0 wird ausgeschaltet. Diese Steuerungsmöglichkeit sollte nur in Ausnahmefällen verwendet werden, besser ist es Binäraktoren über die Anweisungen "einschalten" und "ausschalten" zu steuern.

Wichtiger Hinweis zur besonderen Behandlung von Werten des Typs "bool"

Es gibt viele Werte vom Typ "bool". Damit diese korrekt gesetzt werden muss als Wert "TRUE" oder "FALSE" (jeweils in doppeltem Hochkomma) angegeben werden. Durch diese Wertangabe wird erkannt, dass es sich um einen Wert vom Typ "bool" handelt und für die Übertragung an die Zentrale der korrekte Werttyp eingestellt. So kann dann z.B. auch ein Schaltaktor gesteuert werden.

Beispiel für einen Schataktor:

```
Setzewert(Stehlampe, "STATE", "TRUE")
```

Beispiel zum Ausschalten der Bewegungserkennung eines Bewegungsmelders:

```
Setzewert(Bewegungsmelder, "MOTION_DETECTION_ACTIVE", "FALSE")
```

Mit SETZEWERT ist es auch möglich die Sirene von HMIP-Rauchmeldern zu schalten.

Zum Einschalten:

```
SETZEWERT(MeinHMIPRauchmelder, "SMOKE_DETECTOR_COMMAND", "INTRUSION_ALARM")
```

Zum Ausschalten:

```
SETZEWERT(MeinHMIPRauchmelder, "SMOKE_DETECTOR_COMMAND",  
"INTRUSION_ALARM_OFF")
```

Für neuere Thermostate können über diese Anweisung mehrere Einstellungen vorgenommen werden (gilt nicht für ältere Baureihen).

Der BOOST-Modus wird eingeschaltet mit

```
SETZEWERT(ThermostatName, "BOOST_MODE", 1)
```

Der AUTO-Modus wird eingeschaltet mit

```
SETZEWERT(ThermostatName, "AUTO_MODE", 1)
```

Der MANU-Modus wird eingeschaltet mit

```
SETZEWERT(ThermostatName, "MANU_MODE", Temperatur)
```

wobei die Temperatur eine Konstante mit einer Kommastelle oder Variable vom Typ Zahl mit einer Kommastelle sein kann.

Die Angabe der Kommastelle ist zwingend erforderlich, also z.B.:

```
SETZEWERT(Wohnzimmer, "MANU_MODE", 22.0)
```

Um die Absenk- bzw. Komforttemperatur einzustellen können folgende Anweisungen benutzt werden:

```
SETZEWERT(ThermostatName,"LOWERING_MODE",1)  
bzw.  
SETZEWERT(ThermostatName,"COMFORT_MODE",1)
```

**Bitte beachten Sie unbedingt:**

Diese Anweisung generiert bei jeder Ausführung eine Meldung an das Gerät.

Wenn es sich um ein Funkmodul handelt kann die hemmungslose und unsachgemässe Benutzung dieser Anweisung viele unnötige Funkmeldungen generieren, was dazu führen kann, dass das [Duty-Cycle-Konto](#) überläuft und alle Funkmeldungen an Aktoren zeitweise blockiert werden.

## 1.22.5.27. SETZEZEITTABELLENWERT

---

### SETZEZEITTABELLENWERT

Syntax :

**SETZEZEITTABELLENWERT**(*Objektname*)

Mit dieser Anweisung kann für ein Objekt der Wert aus der Zeittabelle des Objekts eingestellt werden, der nach der Zeittabelle zum letzten Zeitpunkt der Zeittabelle vor der aktuellen Zeit eingestellt worden ist bzw. eingestellt worden wäre falls die Zeittabelle aktiv gewesen wäre.

Diese Anweisung ist z.B. sinnvoll wenn die Zeittabelle eines Objekt ausgeschaltet war und zu einem beliebigen Zeitpunkt wieder eingeschaltet wird. Bei Ausführung dieser Anweisung wird dann der Wert der Zeittabelle im Objekt eingestellt, der eingestellt worden wäre wenn die Zeittabelle eingeschaltet gewesen wäre.

Beispiel:

```
ThermostatWohnraum.Zeittabelle einschalten  
SETZEZEITTABELLENWERT(ThermostatWohnraum)
```

## 1.22.5.28. SICHTBAR

---

### SICHTBAR

Syntax :

**SICHTBAR**(*Objekt*)

Das angegebene Objekt wird auf dem Bildschirm sichtbar gemacht. Falls es schon sichtbar war hat diese Anweisung keine Auswirkung. Ein Objekt kann mit der Anweisung [UNSICHTBAR](#) unsichtbar gemacht werden.

siehe auch [UNSICHTBAR](#)



## 1.22.5.29. SPRACHAUSGABE

---

### SPRACHAUSGABE

Syntax :

**SPRACHAUSGABE**(*Variable*)

Diese Funktion gibt den Textinhalt der Variablen als Sprache auf die Soundkarte des PC's aus. Dazu muss auf einem PC das Programm [VisuWin](#) aktiv sein. Ist kein Programm VisuWin aktiv, so wird diese Anweisung ignoriert.

Anstelle einer Variablen kann auch ein Text in Hochkommas verwendet werden.

Die Qualität der Sprachausgabe ist abhängig von der auf dem PC installierten Stimmen.

Weitere Informationen zur [Sprachausgabe](#)

Beispiel:

```
Sprachausgabe("Das Licht wurde eingeschaltet")
```

Wenn zusammengesetzter Text ausgegeben werden soll, muss dieser vorher in einer Variablen erstellt werden:

```
Text:="Die Temperatur im Terrarium ist mit "+TempTerrarium+" zu niedrig,  
bitte prüfen"  
Sprachausgabe(Text)
```

## 1.22.5.30. STARTE

---

### STARTE

Syntax :

**STARTE**(OBJEKT)

Diese Anweisung stellt das angegebene Makro in die Ausführungwarteschlange. Die Programme der Objekte in der Ausführungwarteschlange werden nach dem FIFO (First In First Out) -Prinzip abgearbeitet. Nach jeder Ausführung eines Programms innerhalb des normalen Ausführungsintervalls werden alle Programme in der Ausführungwarteschlange ausgeführt. Die Ausführung des Makros wird also erst gestartet nachdem das aktuelle Makro beendet worden ist.

## 1.22.5.31. STARTPROGRAMM

---

### STARTPROGRAMM

Syntax :

**STARTPROGRAMM**("Programmname Parameter")

Mit dieser Anweisung können Programme oder Betriebssystembefehle der jeweiligen Hardwareplattform der Zentrale aufgerufen werden.

Die Parameter für das Programm müssen dabei je nach Hardwareplattform und Betriebssystem in einfache Hochkomma gesetzt werden.

Beispiel Programm mit Parameter aufrufen:

```
Startprogramm("/usr/local/myprograms/audiostart 'liste=2 mode=7'")
```

## 1.22.5.32. STARTUHR

---

### **STARTUHR**

Syntax :

**STARTUHR**(*Zeitvariable*)

Die aktuelle Zeit wird in der Zeitvariablen gespeichert. Dies ermöglicht eine Stoppuhrfunktion indem mit der Funktion [STOPPUHR](#)(*Zeitvariable*) die Differenz zwischen gespeicherter und aktueller Zeit ermittelt wird.

Beispiel:

Die Variable Uhr1 ist als Variable vom Typ Uhrzeit definiert.

**STARTUHR**(Uhr1)

Die seit der Ausführung der obigen Anweisung vergangene Zeit kann mit der Anweisung **STOPPUHR**(Uhr1) ermittelt werden.

siehe auch [STOPPUHR](#)

## 1.22.5.33. STARTWIN

---

### STARTWIN

Syntax :

**STARTWIN**("Pfad\Programm,<Parameter>,<Arbeitsverzeichnis>")

Startet ein Windows-Programm auf allen PC's auf denen VisuWin zum Zeitpunkt der Ausführung der Anweisung aktiv ist.

Die Parameter <Parameter> und <Arbeitsverzeichnis> sind optional. Bitte beachten Sie, dass das Arbeitsverzeichnis nicht automatisch als Pfad für das Programm benutzt wird, der Pfad für das Programm muss im ersten Parameter angegeben werden.

Beispiel:

Wenn eine bestimmte Temperatur unterschritten wird, soll ein vordefiniertes Fax geschickt werden.

WENN Temp1 KLEINER 15 DANN

**STARTWIN**( "C:\FAX\SENDEFAX.EXE,HEIZUNGSSTOERUNG" )

ENDEWENN

Um ein Programm auf der aktuellen Hardwareplattform der Zentrale zu starten wird die Anweisung [STARTPROGRAMM](#) benutzt.

## 1.22.5.34. STOPPE

---

### STOPPE

Syntax :

**STOPPE**(*Objektname*)

Mit dieser Anweisung kann der aktive Verstellvorgang von Rollladen- und Markisensteuerungen beendet werden.

Beispiel:

STOPPE(RolloWohnen)

## 1.22.5.35. TEMPERATURABSENKUNG

---

### TEMPERATURABSENKUNG

Syntax :

**TEMPERATURABSENKUNG**(*Wandthermostatname,Absenktemperatur*)

Diese Anweisung kann nur im Makro eines Tür/Fensterkontakts oder Drehgriffkontakts verwendet werden und funktioniert nur korrekt wenn es sich beim Raumthermostat und den Fensterkontakten um HomeMatic-Geräte handelt.

Damit kann ein Wandthermostat direkt in Abhängigkeit von Tür/Fensterkontakten (oder Drehgriffkontakten) über Makros sehr einfach gesteuert werden, ohne dass man bei der Erstellung der Anwendung in den Makros die unterschiedlichen Zustände von mehreren Fensterkontakten für einen Wandthermostat berücksichtigen muss. Mit dieser Anweisung können bis zu 8 Fensterkontakte für einen Wandthermostat verwendet werden. Beim Öffnen eines Fensters wird die in der Anweisung angegebene Absenktemperatur für den Wandthermostat eingestellt. Erst wenn alle Fenster wieder geschlossen sind, wird die zuletzt am Thermostat eingestellte Temperatur wieder aktiviert.

Wenn diese Anweisung benutzt wird, darf der Fensterkontakt nicht durch eine direkte Verknüpfung mit dem Wandthermostat verbunden sein und der Modus des Wandthermostats muss auf "manuell" stehen. Andernfalls ist eine Steuerung des Thermostats über die Makros der Fensterkontakte nicht möglich.

Es ist zu beachten, dass der Abgleich von Temperaturen einige Minuten dauern kann.

Wenn die Absenktemperatur eine Kommastelle hat, muss die Temperatur in Hochkommas gesetzt werden.

Diese Anweisung wird einfach als einzelne Zeile im Makro des des Sensors geschrieben ( ohne wenn-Bedingung bezüglich des Zustands des Tür/Fensterkontakts ! )

Beispiel:

```
TEMPERATURABSENKUNG(Wandthermostat1,"17,5")
```

#### **Bitte beachten Sie:**

Die kleinste Temperatur, die eingestellt werden kann ist 7 Grad, wenn ein kleinerer Wert angegeben wird, wird dieser bei der Ausführung auf 7 Grad geändert. Das passiert weil die Thermostate unterschiedliche Mindestwerte haben, die aber alle unter 7 Grad liegen. Bei einem Wert unter dem Mindestwert des jeweiligen Thermostats wird dieser nicht akzeptiert und das Thermostat sendet eine Wertänderung durch Hardware mit dem Minimalwert. Das würde dann dazu führen, dass der empfangene Wert berichtigt und erneut gesendet wird, was dann eine Endlos-Schleife mit unzähligen unnötigen Funkmeldungen verursacht.

Bei Fensterkontakten ist meistens die Option "Statusmeldungen senden" in den Einstellungen des Geräts aktiviert.

Wenn "Ausführen bei Empfang" für das Makro des Kontakts aktiviert ist, würde das Makro auch ausgeführt wenn der Zustand des Fensterkontakts sich nicht verändert hat. Das führt dann zu vielen unnötigen Temperaturmeldungen an das Thermostat.

Aus diesem Grunde sollte für das Makro eines Fensterkontakts in dem diese Anweisung verwendet wird die Option "Ausführen bei Änderung" statt "Ausführen bei Empfang" verwendet werden.

## 1.22.5.36. UNSICHTBAR

---

### UNSICHTBAR

Syntax :

**UNSICHTBAR**(*Objekt*)

Das angegebene Objekt wird auf dem Bildschirm nicht mehr angezeigt. Es kann mit der Anweisung [SICHTBAR](#) wieder zur Anzeige gebracht werden.

siehe auch [SICHTBAR](#)



## 1.22.5.37. VERLASSEN

---

### VERLASSEN

Syntax :  
**VERLASSEN**

Das aktuelle Makro wird sofort beendet. Die Ausführung wird mit dem nächsten anstehenden Makro fortgesetzt.

Beispiel:

```
WENN Uhrzeit groesser "23:00:00" DANN  
VERLASSEN  
ENDEWENN
```

## 1.22.5.38. WARTE

---

### WARTE

Syntax :

**WARTE** ("00:00:10")

Als Parameter für diese Anweisung kann die Zeit in Hochkommas oder eine Zeichen- oder Uhrzeit-Variable verwendet werden.

Beispiel :

Wartezeit:="01:30:00"

**WARTE** (Wartezeit)

Wartezeit ist im Beispiel eine Variable vom Typ Zahl.

Es kann auch ein konstanter Wert in ganzen Sekunden oder Minuten verwendet werden, dieser Wert kann aber nicht in einer Variablen stehen.

Beispiel:

warte 2 Minuten

Es kann auch eine Variable vom Typ Zahl als Parameter hinter der **WARTE**-Anweisung angegeben werden. In dem Fall muss diese die Sekunden beibehalten die gewartet werden soll und es darf keine Zeiteinheit angegeben werden.

Beispiel:

MeineSekundenZahl:=5

warte MeineSekundenZahl

MeineSekundenZahl ist im Beispiel eine Variable vom Typ Zahl

Mit der Anweisung **WARTE** kann die Ausführung eines Makros für die angegebene Zeit unterbrochen werden, das Makro wird dabei temporär beendet bis zum Ablauf der Wartezeit. Diese Anweisung hat keine Auswirkung auf andere Makros und Funktionen, diese laufen weiter. Nach der angegebenen Zeit wird die Ausführung mit der nächsten Anweisung hinter der **WARTE**-Anweisung fortgesetzt.

Wenn ein Makro mit der Anweisung **AUFRUFEN** als Unterprogramm aus einem anderen Makro aufgerufen wurde, wird das aufrufende Makro fortgesetzt wenn die **WARTE**-Anweisung ausgeführt wird.

### **Bitte unbedingt beachten:**

Eine **WARTE**-Anweisung und damit die aktuelle Ausführung des Makros wird abgebrochen, wenn das Makro vor Ablauf der Wartezeit erneut aufgerufen wird.

## 1.22.5.39. WENN

---

### WENN

Syntax :

**WENN** <NICHT> Bedingung <UND/ODER> Bedingung **DANN**

Anweisungen

<**SONST**>

Anweisungen

**ENDEWENN**

Die wenn-Anweisung ist die einzige Anweisung, die sich über mehrere Zeilen erstrecken kann. Der **wenn**-Teil kann in mehreren Zeilen stehen während **sonst** und **endewenn** –Zeilen jeweils einzelne Zeilen sein müssen.

Mit der WENN-Anweisung ist es möglich den weiteren Programmablauf von einer oder mehreren [Bedingungen](#) abhängig zu machen. Wenn-Anweisungen können auch verschachtelt werden, d.h. zwischen dem **wenn** und dem **endewenn** (bzw. sonst) können weitere wenn-Anweisungen stehen. Jede wenn-Anweisung muss mit einer endewenn-Anweisung beendet werden, ansonsten wird bei der Code-Generierung eine entsprechende Fehlermeldung ausgegeben.

Beispiele:

```
WENN LichtBad ausgeschaltet UND
    SCHALTDauer(LichtBad) groesser "00:05:00" DANN
    Ventilator ausschalten
ENDEWENN
```

```
WENN Temperatur < 20.5 dann
    Heizluefter einschalten
SONST
    Heizluefter ausschalten
ENDEWENN
```

```
WENN Taste2 = "lang" dann
    Stehlampe ausschalten
SONST
    Stehlampe einschalten
ENDEWENN
```

Vor jeder Bedingung kann ein **NICHT** gesetzt werden, dann wird die Anweisung hinter dann ausgeführt wenn die Bedingung **nicht** zutrifft.

**Bitte beachten Sie:**

Das Wort NICHT muss **vor** der eigentlichen Bedingung stehen, es darf **nicht in der Bedingung** stehen.

Beispiel:

**Falsch** wäre die umgangssprachliche Formulierung:

wenn LichtBad **NICHT** ausgeschaltet oder Wochentag **NICHT** = "Montag" dann  
**es würde ein Syntaxfehler angezeigt.**

**Richtig ist:**

wenn **NICHT** LichtBad ausgeschaltet oder **NICHT** Wochentag="Montag" dann

**Bitte beachten Sie:**

Häufige Syntaxfehler in wenn-Anweisungen sind, dass das Wort **dann** vergessen wird und dass die Anweisung nicht mit einem **endewenn** abgeschlossen wird.

Da wenn-Anweisungen sich immer über mehrere Zeilen erstrecken, kann die Zeile für einen Syntaxfehler oft nicht bestimmt werden. Wenn in einem Makro mit einer wenn-Anweisung ein Syntaxfehler ohne Fehlerbeschreibung auftritt, prüfen Sie alle Elemente der wenn-Anweisung um den Fehler zu finden.

Zur besseren Übersichtlichkeit sollten SONST und ENDEWENN immer in separaten Zeilen stehen

## 1.22.6. Funktionen

---

## 1.22.6.1. AKTIVIERT

---

### AKTIVIERT

Syntax :

**AKTIVIERT**(*Objekt*)

Mit dieser Funktion kann festgestellt werden, ob ein Objekt aktiviert ist. Diese Funktion ist z.B. nützlich wenn ein Programm für besondere Aktionen nur zeitweise aktiviert wird.

Beispiel:

```
wenn AKTIVIERT(Alarmanlage) dann  
    Starte(AlleLichterAn)  
endewenn
```

## 1.22.6.2. BATTERIELEER

---

### **BATTERIELEER**

Syntax :

**BATTERIELEER**(*Objekt*)

Mit dieser Funktion kann in Bedingungen geprüft werden, ob ein Modul eine schwache Batterie gemeldet hat.

Die Funktion ist WAHR, wenn eine Batteriewarnung von dem Modul empfangen wurde.

Beispiel:

```
wenn BATTERIELEER(TEMPERATURSENSOR) dann
  Anzeige:="Batterie Temperatursensor austauschen"
endewenn
```

### **Hinweis:**

Um auf Batteriewarnungen zu reagieren kann ein spezielles "Fehler"-Objekt vom Typ Zeichen definiert werden, nähere Infos dazu unter [Spezielle Objekte](#). Jedesmal wenn eine Batteriewarnung empfangen wird, enthält dieses Fehler-Objekt den Namen des Moduls/Objekts, bei dem die Batteriewarnung aufgetreten ist und das Makro des Fehler-Objekts wird ausgeführt.

### 1.22.6.3. COMERROR

---

#### COMERROR

Syntax :

**COMERROR**(*Objekt*)

Mit dieser Funktion kann in Bedingungen geprüft werden, ob für ein Modul aktuell eine Kommunikationsstörung gemeldet wurde.

Die Funktion ist WAHR, wenn aktuell eine Kommunikationsstörung existiert, d.h. wenn eine Meldung über eine Kommunikationstörung von der Zentrale empfangen wurde und danach keine weitere Meldung von dem Modul gekommen ist. Sobald eine normale Meldung von dem Modul empfangen wird, wird der Fehlerstatus zurückgesetzt.

Beispiel:

```
wenn COMERROR(Aussensteckdose) dann
  Anzeige:="Kommunikation mit Aussensteckdose immer noch gestört"
endewenn
```

#### **Hinweis:**

Um auf Kommunikationsstörungen direkt zu reagieren kann ein spezielles "Fehler"-Objekt vom Typ Zeichen definiert werden, nähere Infos dazu unter [Spezielle Objekte](#). Jedesmal wenn eine Meldung über eine Kommunikationsstörung empfangen wird, enthält dieses Fehler-Objekt den Namen des Moduls (Objekts), für das die Kommunikation gestört ist und das Makro des Fehler-Objekts wird ausgeführt.

## 1.22.6.4. DATEIVORHANDEN

---

### DATEIVORHANDEN

Syntax :

### **DATEIVORHANDEN**

Mit dieser Funktion kann geprüft werden ob eine Datei existiert.

Beispiel:

```
wenn DATEIVORHANDEN( "BEWEGUNGEN.TXT" ) dann
  Anzeige:="Meldungen vom Bewegungsmelder vorhanden"
endewenn
```



## 1.22.6.5. DATUM

---

### DATUM

Syntax :  
**DATUM**

Diese Funktion gibt das aktuelle Datum im Format TT.MM.JJJJ zurück.

Beispiel:

```
WENN DATUM =* "15.02.****" UND UHRZEIT="08:00:00" DANN  
PLAY("c:\AudioDaten\homeputer\HappyBirthday.wav")  
ENDEWENN
```

## 1.22.6.6. FEIERTAG

---

### FEIERTAG

Syntax :  
**FEIERTAG**

Diese Funktion hat keine weiteren Parameter und ist in WENN-Bedingungen wahr wenn der aktuelle Tag in der Konfiguration als Feiertag oder als freier Tag definiert ist.

Beispiel:

```
WENN FEIERTAG ODER WOCHENTAG = "Sonntag" DANN  
    WECKZEIT:="09:00:00"  
ENDEWENN
```

## 1.22.6.7. GESCHALTET

---

### GESCHALTET

Syntax :

**GESCHALTET**(*Objekt*)

Diese Funktion wird ausschließlich in Bedingungen verwendet, um zu prüfen ob ein Objektzustand sich geändert hat.

Beispiel:

```
WENN Alarmsensor eingeschaltet und GESCHALTET(Alarmsensor) DANN  
    SENDEMAIL(.....)  
    ERLEDIGT(Alarmsensor)  
ENDEWENN
```

Sobald sie die Anweisung ERLEDIGT für ein Objekt ausgeführt wurde ergeben die weiteren GESCHALTET-Abfragen das Ergebnis unwahr (also falsch) bis der Zustand des angegebenen Objekts sich erneut verändert. Die Bedingungsabfrage im obigen Makro würde beispielsweise verhindern, dass eine Alarm-mail mehrmals gesendet wird, da die GESCHALTET-Bedingung nur bei der ersten Ausführung des Makros zutrifft.

siehe auch [ERLEDIGT](#)

## 1.22.6.8. JAHR

---

JAHR

Syntax :  
**JAHR**

Diese Funktion gibt das aktuelle Jahr als Zahl zurück.

## 1.22.6.9. MAILLOG

---

### MAILLOG

Syntax :

**MAILLOG(Index)**

Mit dieser Funktion können die Logs der letzten 5 e-mail-Sendungen abgerufen werden. Der Index 1 der Tabelle enthält immer den Status der zuletzt gesendeten e-mail, der Index 5 den Status der ältesten der letzten fünf mails.

Die Status-Infos für eine mail werden als Text mit drei Zeilen gespeichert. Die Zeilen werden jeweils durch << getrennt und können daher direkt auf ein Anzeigeobjekt vom Typ Zeichen ausgegeben werden.

In der ersten Zeile stehen Datum, Uhrzeit und der Status der Sendung OK=korrekt versandt, bei einem Fehler eine Fehlerkennung und/oder ein Fehlertext.

In der zweiten Zeile steht die Zieladresse der mail.

In der dritten Zeile steht der Betrefftext der mail.

Die Fehlerkennungen haben folgende Bedeutungen:

999 : Einer der folgenden Werte ist ungültig oder leer: Mail-Server, Absenderadresse, Zieladresse, Betreff, Mailtext.

andere : Ursache im Text angegeben

Beispiel:

```
Anzeige1 := MAILLOG(1)
```

Anzeige1 hätte danach folgenden z.B. Textinhalt:

**02.06.2008 12:23:20 OK<<Meine.BueroAdresse@Mein-Provider.com<<Temperaturinfo**

Auf dem Anzeigepanel des Objekt Anzeige1 würde dann Meldung erscheinen:

**02.06.2008 12:23:20 OK**

**Meine.Bueroadresse@Mein-Provider.com**

**Temperaturinfo**

## 1.22.6.10. MONAT

---

### MONAT

Syntax :  
**MONAT**

Diese Funktion gibt den aktuellen Monat als Zahl zurück.

Beispiel:

```
WENN Monat = 5 DANN  
    Anzeigel:="Der Mai ist gekommen.."  
ENDEWENN
```

## 1.22.6.11. MONATSTAG

---

### MONATSTAG

Syntax :  
**MONATSTAG**

Diese Funktion gibt den aktuellen Tag des Monats als Zahl zurück.

Beispiel:

```
WENN MONATSTAG=1 DANN  
    Anzeigel:="Ein neuer Monat hat begonnen"  
ENDEWENN
```

## 1.22.6.12. OBJEKTBEZ

---

### OBJEKTBEZ

Syntax :

**OBJEKTBEZ**(*Objekt*)

Diese Funktion gibt die Bezeichnung des angegebenen Objekts als Text zurück und dient dazu diese anzeigen zu können oder z.B. als Mail verschicken zu können.

Beispiel:

```
Sendezeile:="Das ist eine mail von Objekt  " + OBJEKTBEZ(selbst)
SENDEMAIL("Status", Sendezeile, 2)
```



## 1.22.6.13. OBJEKTNAME

---

### OBJEKTNAME

Syntax :

**OBJEKTNAME**(*Objekt*)

Diese Funktion gibt den Namen des angegebenen Objekts als Text zurück und dient dazu diesen anzeigen zu können oder als Mail verschicken zu können.

Beispiel:

Sendezeile:="Das ist eine mail von Objekt " + OBJEKTNAME(selbst)

SENDEMAIL("Status", Sendezeile, 2)

## 1.22.6.14. LÖSCHEDATEI

---

### LÖSCHEDATEI

Syntax :

**LÖSCHEDATEI**("Dateiname")

Mit dieser Anweisung können Dateien gelöscht werden. Das können selbst erstellte Textdateien, aber auch beliebige andere Dateien wie die Historydatei **history.txt** oder die Systemlog-Datei **syslog.txt** sein.

**Bitte beachten Sie unbedingt:**

Durch das beabsichtigte oder unbeabsichtigte Löschen von Dateien, die nicht selbst erstellt wurden, kann die Systemintegrität zerstört und die Funktionen der Zentrale beeinträchtigt werden.

**Löschen Sie daher keinesfalls Dateien, von denen Sie nicht wissen wofür diese benutzt werden.**

**Achten Sie unbedingt darauf, dass das auch nicht versehentlich, etwa durch Angabe eines falschen Dateinamens, geschieht.**

Beispiel:

`LÖSCHEDATEI ( "BEWEGUNGEN.TXT" )`

## 1.22.6.15. SCHALTDauer

---

### SCHALTDauer

Syntax :

**SCHALTDauer**(*Objekt*)

Diese Funktion gibt die Zeit zurück, in der das Objekt sich in seinem aktuellen Zustand befindet. Es wird die reine Uhrzeitdifferenz zwischen letzter Schaltzeit und aktueller Uhrzeit ermittelt. Tage werden nicht berücksichtigt, d.h. als Ergebnis kann es keinen Wert grösser als 23:59:59 geben.

## 1.22.6.16. SELBST

---

### SELBST

Syntax :  
**SELBST**

Diese Funktion gibt das Objekt des Makros zurück, in dem sie verwendet wird.  
Damit ist es möglich das aktuelle Objekt im Makro zu verwenden ohne den Objektnamen schreiben zu müssen.

Einer der Vorteile der Funktion SELBST ist, dass das Makro einfach in ein anderes Objekt kopiert werden kann, ohne das Änderungen bezüglich des Objektnamens im Makro erforderlich sind.

Beispiel:

```
SELBST umschalten  
Anzeige:=Objektbez(SELBST) + " ist jetzt " + SELBST
```

Im Objekt Anzeige vom Typ Zeichen würde dann z.B. Folgendes angezeigt:  
**Stehlampe Wohnzimmer ist jetzt an**

## 1.22.6.17. SONNENAUFGANG

---

### SONNENAUFGANG

Syntax :  
**SONNENAUFGANG**

Mit dieser Funktion kann die Uhrzeit des Sonnenaufgangs für den aktuellen Tag ermittelt werden. Da diese Zeit vom geografischen Standort abhängig ist, kann auf der Seite *Einstellungen* das Postleitzahlgebiet angegeben werden. Durch Mausklick auf die Schaltfläche [Eingabe Koordinaten] können auch die exakten Koordinaten des Standorts eingegeben werden. Die Berechnung erfolgt durch astronomische Näherungsformeln, es wird die sogenannte „bürgerliche Dämmerung“ berechnet. Je nach Jahreszeit und geographischem Standort kann es zu einer Abweichung von einigen Minuten kommen.

Beispiel:

```
wenn Uhrzeit = SONNENAUFGANG dann  
    RollladenSchlafzimmer hochfahren  
endewenn
```

#### **Hinweis:**

Das Makro im Beispiel muss mindestens 1 x in der Minute ausgeführt werden, da die Zeit des Sonnenuntergangs immer volle Minuten hat. Das muss bei der Einstellung des Ausführungsintervalls beachtet werden für Makros in denen wenn-Bedingungen mit Sonnenzeiten benutzt werden.

siehe auch [SONNENUNTERGANG](#)

## 1.22.6.18. SONNENUNTERGANG

---

### SONNENUNTERGANG

Syntax :

### **SONNENUNTERGANG**

Mit dieser Funktion kann die Uhrzeit des Sonnenuntergangs für den aktuellen Tag ermittelt werden. Da diese Zeit vom geografischen Standort abhängig ist, kann auf der Seite *Einstellungen* das Postleitzahlgebiet angegeben werden. Durch Mausklick auf die Schaltfläche [Eingabe Koordinaten] können auch die exakten Koordinaten des Standorts eingegeben werden. Die Berechnung erfolgt durch astronomische Näherungsformeln, es wird die sogenannte „bürgerliche Dämmerung“ berechnet. Je nach Jahreszeit und geographischem Standort kann es zu einer Abweichung von einigen Minuten kommen.

Beispiel:

```
wenn Uhrzeit = SONNENUNTERGANG dann  
  RollladenSchlafzimmer runterfahren  
endewenn
```

### **Hinweis:**

Das Makro im Beispiel muss mindestens 1 x in der Minute ausgeführt werden, da die Zeit des Sonnenuntergangs immer volle Minuten hat. Das muss bei der Einstellung des Ausführungsintervalls beachtet werden für Makros in denen wenn-Bedingungen mit Sonnenzeiten benutzt werden.

siehe auch [SONNENAUFGANG](#)

## 1.22.6.19. STOPPUHR

---

### STOPPUHR

Syntax:

**STOPPUHR**(Zeitvariable)

Diese Funktion gibt die Zeitdauerdifferenz zwischen der aktuellen Uhrzeit und der in der Zeitvariablen hinterlegten Zeit im Format HH:MM:SS zurück. Es wird die reine Uhrzeitdifferenz ermittelt, Tage werden nicht berücksichtigt, d.h. als Ergebnis kann es keinen Wert grösser als 23:59:59 geben.

Beispiel:

Das Treppenhauslicht soll ausgeschaltet werden, wenn es länger als 2 Minuten an ist und der Taster zum Einschalten länger als 1 Minute nicht mehr betätigt wurde. Beachten Sie, daß als Parameter bei der Ermittlung der Stoppzeiten die Variable CT des Objekts und nicht das Objekt selbst verwendet wird.

```
wenn LiTreppenh eingeschaltet dann
  wenn Stoppuhr(LiTreppenh.ct) groesser "00:02:00" und
    Stoppuhr(TastTreppenh.ct) groesser "00:01:00" dann
    LiTreppenh ausssschalten
  endewenn
endewenn
```

#### **Bitte beachten Sie:**

Als Parameter darf nicht der Objektname, sondern es muss eine Zeitvariable oder die für jedes Objekt vorhandenen Zeitvariable CT (also *Objektname.CT*) angegeben werden. Es gibt auch noch eine andere Funktion um festzustellen wie lange ein Objekt sich in seinem aktuellen Zustand befindet, das ist die Funktion SCHALTDAUER(Objekt), bei dieser Funktion wird das Objekt (ohne den Zusatz.CT) als Parameter angegeben.

## 1.22.6.20. TAUPUNKT

---

### TAUPUNKT

Syntax:

**TAUPUNKT**(Temperatur, Luftfeuchte)

Mit dieser Funktion kann die Taupunkttemperatur ermittelt werden.

Wenn die Taupunkttemperatur unterschritten wird, verflüssigt sich der in der Luft enthaltene Wasserdampf.

Es kommt an Wänden, Fenstern usw. zur Kondenswasserbildung, wenn dort die Taupunkt-Temperatur unterschritten wird.

Die entstehende Feuchtigkeit kann dann z.B. zu Schimmelbildung führen. Um das zu vermeiden kann durch Änderung der Temperatur bzw. der Luftfeuchtigkeit (z.B. durch Lüften) der Taupunkt verändert werden, damit er unter der Temperatur z.B. der Wände eines Raums liegt.

Beispiel zur Ermittlung des Taupunkt mit den Temperatur- und Luftfeuchtwerten eines Wandthermostats:

```
TaupunktKeller1:=Taupunkt(WTKeller1.Temperatur, WTKeller1.Luftfeuchtigkeit)
```



## 1.22.6.21. TRIGGER

---

### TRIGGER

Syntax:

**TRIGGER**(Objekt)

Mit dieser Funktion kann die Ursache der letzten Wertänderung des Objekts bestimmt werden. Der Rückgabewert ist ein Zahlencode mit folgender Bedeutung:

1=Makro-Programm

2=Hardware

3=Hardware-Status nach Abfrage

4=Zeittabelle

5=TimeControl (nur c-comatic mit Zeitsteuerungsmodul)

6=Fensterkontakt über Anweisung TEMPERATURABSENKUNG

7,8=extern über App, Webserver, Web-Schnittstelle, XMLRPC-Schnittstelle usw.

9=LESEWERTEDATEI

10=binäre Temperaturregelung (für beim Thermostat eingetragene Schaltaktoren)

11..20 VisuWin

Mit dieser Funktion ist es möglich in einem Makro unterschiedliche Aktionen in Abhängigkeit von der Ursache der aktuellen Wertänderung eines Objekts auszuführen.

Wenn die Option "Ausführen bei Änderung" aktiviert ist, kann so ermittelt werden, was die aktuelle Makroausführung verursacht hat um ggfs. unterschiedliche Programmierungen je nach Ursache der Wertänderung vorzunehmen.

Beispiel:

```
wenn TRIGGER(selbst) = 4 dann
  Anzeige:="Wert-Änderung durch Zeittabelle"
endewenn
```

### **Bitte beachten Sie:**

Wenn der Objektwert eines Hardware-Objekts geändert wird, melden die meisten Hardwaremodule den aktuellen Status zurück, nachdem sie auf den neuen Wert umgeschaltet haben. Da eine solche Hardwaremeldung nicht von einer Meldung durch tatsächliche Bedienung der Hardware zu unterscheiden ist, wird der aktuelle Triggercode je nach Hardware eventuell nur Sekundenbruchteile nachdem eine Änderung z.B. durch eine Makroanweisung erfolgte auf "Hardware" (also 2) geändert.

Bis der Triggercode durch diese Rückmeldung von der Hardware wieder geändert wird, gibt die Funktion den Triggercode zurück, der die Hardwareschaltung verursacht hat, kann also z.B. im Makro des geschalteten Objekts benutzt werden um die Ursache der aktuellen Wertänderung zu ermitteln.

## 1.22.6.22. UHRZEIT

---

### UHRZEIT

Syntax :  
**UHRZEIT**

Diese Funktion gibt die aktuelle Uhrzeit im Format HH:MM:SS zurück. Die Uhrzeit wird im 5-Sekunden-Takt aktualisiert, die Sekunden der Uhrzeit sind also immer durch 5 teilbar.

Beispiel:

```
WENN UHRZEIT groesser "23:00:00" DANN  
AKTZEIT:=UHRZEIT  
ENDEWENN
```

## 1.22.6.23. UHRZEITSEKUNDE

---

### UHRZEITSEKUNDE

Syntax :

### **UHRZEITSEKUNDE**

Diese Funktion gibt die aktuelle Uhrzeit im Format HH:MM:SS zurück. Die Sekunden sind aktuell und nicht wie in der Funktion UHRZEIT im 5-Sekundenintervall.

## 1.22.6.24. WOCHENTAG

---

### WOCHENTAG

Syntax :  
**WOCHENTAG**

Diese Funktion gibt den aktuellen Wochentag als Text zurück. Bei Vergleichsbedingungen mit dieser Funktion muss die Groß/Kleinschreibung beachtet werden.

Beispiel:

```
WENN WOCHENTAG = "Montag" DANN  
AKTTAG:=WOCHENTAG  
ENDEWENN
```

## 1.22.6.25. ZEIT

---

### ZEIT

Syntax :  
**ZEIT**

Diese Funktion ergibt die Zeit in einem internen Format. Dieses Format wird z.B. verwendet um die Stoppzeit zu ermitteln.

Beispiel:

Eine Zeitvariable kann mit der Funktion Zeit gefüllt werden. Die [Variable](#) UHR1 muss vom Typ *Zeit* sein.

Nachdem die Anweisung

```
UHR1 := ZEIT
```

ausgeführt wurde kann später mit der Funktion

```
STOPPUHR ( UHR1 )
```

die seit der Ausführung der ersten Anweisung vergangene Zeit ermittelt werden.

Die Anweisung UHR1:=ZEIT hat die gleiche Wirkung

wie die Anweisung STARTUHR(UHR1).

Weiterhin ist der Typ Zeit nützlich um die komplette Zeit (also Datum und Uhrzeit zu merken und auszugeben. Wenn eine Variable dieses Typs einem Zeichen-Objekt zugewiesen wird, werden Datum und Uhrzeit als Text ausgegeben.

## 1.22.6.26. ZUFALLSZEIT

---

### ZUFALLSZEIT

Syntax :

**ZUFALLSZEIT**(Zeitvariable)

Generiert mit einem Zufallsgenerator eine Zeit bis maximal zum in der Zeitvariablen angegebenen Wert. Die Zeitvariable muss den Typ Uhrzeit haben (nicht den Typ Zeit, da dieser auch das Datum beinhaltet). Der maximal zulässige Wert ist "23:59:59". Es wird immer eine durch 5 teilbare Sekundenzahl zurückgegeben.

Beispiel:

Ein Programm zur Anwesenheitssimulation während des Urlaub soll das Licht zwischen 20 und 21 Uhr einschalten und zwischen 22 und 23:30 Uhr ausschalten.

Die Ein- und Ausschaltzeiten werden am besten von einem Programm, welches bei Tageswechsel läuft festgelegt.

Dies könnte so aussehen:

```
Einschaltzeit:="20:00:00"+Zufallszeit("01:00:00")
```

```
** (generiert eine Zeit zwischen 20 und 21 Uhr)
```

```
Ausschaltzeit:= "22:00:00"+Zufallszeit("01:30:00")
```

```
** (generiert eine Zeit zwischen 22 und 23:30 Uhr)
```

Verwendet werden können diese Zeiten dann in einem Makro um z.B. Beleuchtungen zu schalten.

Ein solches Makro könnte so aussehen:

```
wenn Uhrzeit=Einschaltzeit dann
    Licht einschalten
endewenn
wenn Uhrzeit=Ausschaltzeit dann
    Licht ausschalten
endewenn
```

## **1.22.6.27. Funktionen zur Textbearbeitung**

---

## 1.22.6.27.1. ERSETZEN

---

### ERSETZEN

Syntax :

**ERSETZEN**(*Zeile,Alt,Neu*)

Diese Funktion ersetzt den Text des Parameters "Alt" in der Zeile durch den Text des Parameters "Neu".

"Zeile" ist ein Objekt/eine Variable vom Typ Zeichen, "Alt" und "Neu" sind Variablen/Objekte vom Typ Zeichen oder Konstanten in Hochkommas.

Beispiel:

```
ZEILE:="Es ist jetzt dunkel im Zimmer"
```

```
ZEILE:=ERSETZEN(ZEILE,"dunkel","hell")
```

Nach Ausführung hat sich die Zeile geändert auf : "Es ist jetzt hell im Zimmer".



## 1.22.6.27.2. GROSSBUCHSTABEN

---

### GROSSBUCHSTABEN

Syntax :

**GROSSBUCHSTABEN**(*Text*)

Diese Funktion wandelt alle Kleinbuchstaben des Parameters Text in Grossbuchstaben um. Text kann eine Variable oder Konstante vom Typ Zeichen sein.

Beispiel:

**GROSSETEXT:=GROSSBUCHSTABEN( ZEILE )**

### 1.22.6.27.3. KLEINBUCHSTABEN

---

#### KLEINBUCHSTABEN

Syntax :

**KLEINBUCHSTABEN**(*Text*)

Diese Funktion wandelt alle Grossbuchstaben des Parameters Text in Kleinbuchstaben um. Text kann eine Variable oder Konstante vom Typ Zeichen sein.

Beispiel:

**KLEINERTEXT:=KLEINBUCHSTABEN(ZEILE)**

## 1.22.6.27.4. LESETEXTPAR

---

### LESETEXTPAR

Syntax :

**LESETEXTPAR**(*Text*,*Index*,*Trennzeichen*)

Diese Funktion gibt den Textteil zurück, der im durch Index angegebenen Teil des Textes steht wobei die einzelnen Textteile durch das im Parameter *Trennzeichen* angegebene einzelne Zeichen getrennt werden.

Beispiel:

```
ZEILE:="Teil1/Teil2/Teil3/Teil4"
```

```
PAR3:=LESETEXTPAR(ZEILE,3,"/ ")
```

In der Variablen PAR3 würde nach der Ausführung "TEIL3" stehen.

## 1.22.6.27.5. LINKERTEIL

---

### LINKERTEIL

Syntax :

**LINKERTEIL**(*Text*,*Anzahl*)

Diese Funktion gibt die angegebene Anzahl Buchstaben der linken Seite des Textes zurück.  
Text ist eine Variable des Typs Zeichen, Anzahl ist eine Variable des Typs Zahl ohne Kommastellen  
oder eine Konstante.

Beispiel:

```
ZEILE:="Das ist eine Zeile"
```

```
LINKS:=LINKERTEIL(ZEILE,3)
```

In der Textvariablen LINKS würde nach der Ausführung nur das Wort "Das" stehen.

## 1.22.6.27.6. RECHTERTEIL

---

### RECHTERTEIL

Syntax :

**RECHTERTEIL**(*Text*,*Anzahl*)

Diese Funktion gibt die angegebene Anzahl Buchstaben der rechten Seite des Textes zurück.  
Text ist eine Variable des Typs Zeichen, Anzahl ist eine Variable des Typs Zahl ohne Kommastellen  
oder eine Konstante.

Beispiel:

```
ZEILE:="Das ist eine Zeile"  
RECHTS:=RECHTERTEIL(ZEILE,5)
```

In der Textvariablen RECHTS würde nach der Ausführung nur das Wort "Zeile" stehen.

## 1.22.6.27.7. TEXTLÄNGE

---

### TEXTLÄNGE

Syntax :  
**TEXTLÄNGE**(*Text*)

Diese Funktion gibt die Anzahl der Zeichen zurück, die in der angegebenen Textzeile enthalten sind.

Beispiel:

ZEILE:="abcde"  
ANZAHL:=**TEXTLÄNGE**(ZEILE)

In der Variablen ANZAHL vom Typ Zahl würde nach der Ausführung die Zahl 5 stehen weil der Text 5 Zeichen lang ist.

## 1.22.6.27.8. TEXTPOSITION

---

### TEXTPOSITION

Syntax :

**TEXTPOSITION**(*Textteil*, *Text*)

Diese Funktion gibt die Position eines Textteils innerhalb einer Textzeile zurück. Das Ergebnis ist vom Typ Zahl ohne Kommastellen.

Textteil ist ein Objekt/eine Variable vom Typ Zeichen oder eine Konstante in Hochkommas, Text eine Variable oder ein Objekt vom Typ Zeichen.

Beispiel:

```
ZEILE:="Wo ist denn das XY in dieser Zeile?"  
XYPOS:=TEXTPOSITION(ZEILE, "XY")
```

In der Variablen XYPOS würde nach der Ausführung die Zahl 16 stehen, weil das "X" von "XY" an der 16. Stelle der Zeile steht.

## 1.22.6.27.9. TEXTTEIL

---

### TEXTTEIL

Syntax :

**TEXTTEIL**(*Zeile,Pos,Anzahl*)

Diese Funktion ermittelt einen Textabschnitt aus der Zeile, beginnend an der Stelle "Pos" mit der Anzahl im Parameter "Anzahl" angegebenen Buchstaben.

Zeile ein Objekt/eine Variable vom Typ Zeichen, "Pos" und "Anzahl" sind Variablen/Objekte vom Typ Zahl ohne Kommastellen oder Konstanten.

Beispiel:

```
ZEILE:="Das xyz wir herausgeholt"  
NEUERTEXT:=TEXTTEIL(ZEILE,5,3)
```

In der Variablen NEUERTEXT würde nach der Ausführung "xyz" stehen.



## 1.22.6.27.10. TEXTZWISCHEN

---

### TEXTZWISCHEN

Syntax :

**TEXTZWISCHEN**(*Text*, *Textteil1*, *Textteil2*)

Diese Funktion gibt den Textteil zurück, der zwischen den in den Parametern Textteil1 und Textteil2 angegebenen Zeichen steht.

Beispiel:

```
ZEILE:="Breite=200cm,Farbe=rosarot;Gewicht=53kg;"  
Farbe:=TEXTZWISCHEN(ZEILE,"Farbe=",";")
```

In der Variablen Farbe würde nach der Ausführung "rosarot" stehen. Mit dieser Anweisung ist es z.B. möglich auf einfache Art bestimmte Werte einer zuvor mit [GETSITE](#) eingelesenen Internetseite auszuwerten.

## 1.23. Hinweise zur Hardware

---

### Hinweise zur Hardware

In diesem Abschnitt werden Besonderheiten zur Behandlung von System und speziellen Modulen beschrieben.

Es werden nicht alle Module aufgeführt, sondern nur die, bei denen die Ansteuerung und bzw. das Verhalten vom Standard abweicht.

## 1.23.1. Datenpunkte von Geräten ändern

### Datenpunkte von Geräten anlegen / ändern

#### Ändern der Eingabeart Taster/Schalter

Bei den meisten Geräten erfolgt die Umstellung der Hardware-Datenpunkte automatisch bei Änderung des Objekttyps. Das ist aber nicht bei allen Geräten - insbesondere bei Platinen/Bausätzen - so.

Bei Geräten, bei denen die Anpassung nicht automatisch bei Änderung des Objekttyps erfolgt, muss die Umstellung manuell gemacht werden.

Standardmäßig steht ein Gerät mit änderbarem Eingangstyp normalerweise auf Tastereingang. Wenn ein Kanal dann als Schaltereingang benutzt werden soll, reicht es nicht, diesen in der WEB-UI einzustellen, es muss dann auch in der CL-Software eine entsprechende Änderung durchgeführt werden.

Dazu wird die Seite *Objekt bearbeiten* des Geräts aufgerufen. Im Reiter *Allgemein* dieser Seite dann auf den Button *Hardwarezuordnung ändern* klicken.

The screenshot shows the 'Objekt bearbeiten' window for 'Schalter1'. The 'Allgemein' tab is active. On the left, there is a dropdown menu set to 'Taster2' and a list of checkboxes: 'Deaktivieren beim Start', 'Unsichtbar beim Start', 'Ausführen bei Eingabe', 'Ton beim Umschalten', 'nicht benutzen', and 'Historydaten schreiben'. Below these is a 'Gruppen des Objekts' section with a diagram placeholder and a 'Neu' dropdown. On the right, the 'Bezeichnung' is 'Schalter1', 'Startwert' is 'aus', 'Eingabeoptionen' is 'Standard', and 'Reset-Zeit in Sekunden' is '3'. At the bottom, 'Hardware-Modul' is 'Modulplatine' and 'Adresse' is '00251A499B30A4:13'. A red rectangle highlights the 'Hardwarezuordnung ändern' button. At the bottom right are buttons for 'Hilfe', 'Abbrechen', and 'OK'.

Es erscheint ein Warnhinweis, dieser wird mit *Ja* beantwortet.

Es erscheint das Fenster *Gerätedefinition*, in dem die Parameter für die Kommunikation mit dem Geräte geändert werden können.

Gerätedefinition

Modul: Modulplatine

Adresse: 00251A499B30A4

Kanal: 13

Datenpunkt: PRESS

Kanal-Typ: Sensor

Bus: HmIP-RF

Abbrechen OK

Bei Tastereingängen steht im Feld Datenpunkt das Schlüsselwort **PRESS**. Um den Kanal als Schalteringang oder Tür/Fensterkontakt zu benutzen muss dieses Schlüsselwort auf **STATE** geändert werden.

Gerätedefinition

Modul: Modulplatine

Adresse: 00251A499B30A4

Kanal: 13

Datenpunkt: STATE

Kanal-Typ: Sensor

Bus: HmIP-RF

Abbrechen OK

Dann wird die Eingabe mit OK bestätigt.

Nun muss noch der Objekttyp auf *Schalter* geändert werden und der Eingang kann als Schalter benutzt werden.

### Benutzung von Datenpunkten, die von der CL-Software standardmässig beim Import nicht angelegt werden

Beim Import von Geräten aus der CCU werden Schnittstellen für die normalerweise benutzten Werte erstellt.

Für jeden Wert der von der CCU kommt bzw. an die CCU geschickt wird gibt es einen festgelegten Kanal, der durch einen Doppelpunkt getrennt hinter der Adresse des Moduls steht und ein Schlüsselwort, das den Wert identifiziert (diese Kombination wird Datenpunkt genannt). Jeder Kanal hat normalerweise mindestens einen Schlüsselwert, oftmals sind es aber auch mehrere. Um Werte von Modulen zu verarbeiten, für die von der CL-Software kein Objekt angelegt wird (wenn das für jeden Datenpunkt gemacht würde gäbe es eine Unzahl unnötiger und meist unbenutzter Objekte, was viel zu unübersichtlich wäre).

Um aber im Bedarfsfall auch selten benutzte spezielle Werten benutzen zu können gibt es die Möglichkeit zu virtuellen (also selbst angelegten Objekten ohne Hardwarebezug) eine Geräteschnittstelle zuzufügen.

Diese Option ist für erfahrene Anwender mit entsprechendem Fachwissen gedacht.

Wenn einem Objekt keine Hardware zugeordnet ist kommt als erstes ein Auswahlfenster mit den importierten Hardwaremodulen. Hier muss ein Modul ausgewählt werden.

Dann können für dieses Modul Datenpunkte angelegt werden, die von der Software beim Import nicht angelegt worden sind.

Dazu muss die Kanalnummer und der Datenpunktname in den Eingabefeldern angegeben werden.

## 1.23.2. HomeMatic

---

### HomeMatic

HomeMatic-Geräte erkennt man daran, dass die Typbezeichnung mit HM-.. beginnt. Hier werden Hinweise zu speziellen HomeMatic-Geräten beschrieben.

Bei den HomeMatic und HomeMatic IP - Geräten, bei denen die Eingänge umkonfigurierbar sind ist folgendes zu beachten:

Die aktuelle Konfiguration des Eingangstyps wird beim Import nicht automatisch eingestellt, es muss im Programm jeweils der passende Objekttyp ausgewählt werden.

Der Eingangstyp *Schalter* des HomeMatic-Geräts entspricht nicht wirklich einem Schalter, sondern einem Taster. Bei jeder Aktion, egal ob der Kontakt geschlossen oder geöffnet wird, wird jeweils die Meldung *PRESS\_SHORT* gesendet. In der Software muss daher in dem Fall der Objekttyp *Taster* ausgewählt werden.

Wenn an den Eingang des Geräts ein Schalter angeschlossen wird, bei dem der aktuelle Zustand gesendet werden soll, muss am Gerät in der WEB-UI unter *Kanalverhalten* bzw. *Eingangstyp* **Tür/Fensterkontakt** oder **Schliesserkontakt** ausgewählt werden.

## 1.23.2.1. Rollladen/Jalousiesteuerungen

---

### Rollladen/Jalousiesteuerungen

Die Laufzeiten für Rollladen-, Jalousie- und Markisensteuerungen werden in der Konfiguration über die WEB-Oberfläche der Zentrale angegeben. Es sollte niemals eine kleinere Motorumschaltzeit als 0,5 Sekunden (besser 1 Sekunde) angegeben werden, da sonst die Gefahr besteht, dass die Relais des Aktors beschädigt werden.

Im Folgenden wird für Rollladen- Jalousie und Markisensteuerungen nur der Begriff Rolllade verwendet.

Beim Import bzw. der Erstellung wird für diese Steuerungen der Objekttyp *Jalousie* verwendet. Die Stellungen der Rollladen und Jalousien (mit Lamellen falls vorhanden) wird dann in Prozenten von 0 bis 100 angegeben, wobei der Wert 0 für komplett geschlossen und der Wert 100 für komplett geöffnet verwendet wird.

Der Objekttyp Jalousie ist der am besten geeignete Objekttyp für diese Steuerungen.

Um eine Rolllade mit dem Objekttyp **Jalousie** zu öffnen würde im Makro die beispielsweise folgende Anweisung verwendet:

```
RollladeKueche:=100
```

Um eine Rolllade zur Hälfte zu öffnen würde beispielsweise folgende Anweisung verwendet:

```
RollladeKueche:=50
```

Um den Zustand einer Rolllade abzufragen wird die Prozentzahl verwendet, also z. B.:

```
wenn RollladeDiele > 90 dann  
  Anzeige:="Rolllade Diele ist offen"  
endewenn
```

Wenn es sich um einen Jalousieaktor handelt, der auch Lamellen steuern kann, gibt es in dem Objekt eine zusätzliche Variable *Lamellen*, die zur Steuerung der Lamellenstellung benutzt wird. Auch hier werden Prozentzahlen von 0 - 100 benutzt.

Alternativ zum Objekttyp *Jalousie* kann für Rollladen auch der Objekttyp Rolllade oder Rolllade1 benutzt werden.

Bei diesem Objekttyp werden nicht Zahlen, sondern Zustände verwendet. Im Folgenden weitere Erklärungen dazu:

Das Verstellen von Rollladen kann im Makro mit den Schlüsselwörtern **runterfahren** und **rauffahren** geschehen.

also z.B.

```
RollladeWohnen runterfahren
```

Um in Zwischenstellungen zu fahren, wird die jeweilige Zwischenstellung direkt als Wert zugewiesen.

also z.B.

```
RollladeWohnen:="halb"
```

Beachten Sie dabei bitte, dass die gewünschte Zwischenstellung in Hochkomma gesetzt werden und exakt dem Wort des Zustands entsprechen muss (auch Gross-/Kleinschreibung beachten). Die genauen Wörter für die jeweiligen Positionen werden auf der Seite "Visualisierung" der Objektdefinitionen angezeigt.

Der Zustand einer Rolllade kann mit den Zustandstexten abgefragt werden. Schlüsselwörter (z.B. geöffnet, geschlossen) können für Rollladen **nicht** verwendet werden.

Richtig wäre also z.B.

```
wenn RollladeKueche = "oben" dann  
  AnzeigeRK:="offen"  
endewenn  
oder
```

```
wenn RollladeKueche = "unten" dann
  AnzeigeRK:="geschlossen"
endewenn
```

oder

```
wenn RollladeKueche = "dreiviertel" dann
  AnzeigeRK:="3/4 geschlossen"
endewenn
```

Für Rollladen gibt es eine zusätzliche spezielle Anweisung zum Stoppen eines Verstellvorgangs.  
Dies passiert mit der Anweisung  
STOPPE(*Rolldexx*).

**Bitte beachten Sie:**

Nach einem Fahrbefehl an eine Rolllade wird der neue Zustand in der Visualisierung für einen Sekundenbruchteil angezeigt, danach wird jeweils der Zustand angezeigt, der von dem Steuerungsmodul aktuell gesendet wird. Das ist normalerweise erst einmal der ursprüngliche Zustand, erst nach Abschluss der Verstellvorgangs wird der neue Zustand empfangen und angezeigt.



## 1.23.2.2. HM-Thermostate

### HM-Thermostate

Über eine kurze Betätigung der Menü-Taste des Thermostats kann der Modus eingestellt werden. Empfehlenswert ist der Modus "Manu". In diesem Modus kann das Thermostat über die Zentrale eingestellt werden, aber es sind auch manuelle Veränderungen der Temperatur am Stellrad möglich.

Damit ein Thermostat ausschliesslich über die Zentrale gesteuert wird, kann der Modus auf "Cent" gesetzt werden.

Weitere Infos dazu finden Sie in der Bedienungsanleitung des Thermostats.

Zur Visualisierung des Thermostats wird ein spezielles Panel benutzt.

Die Temperatur auf der linken Seite des Panels ist die Solltemperatur, die über die Visualisierung verändert werden kann. Die Temperatur auf der rechten Seite des Panels ist die aktuelle Ist-Temperatur, die vom Thermostat empfangen wurde. In der Mitte wird die aktuelle gemessene Luftfeuchtigkeit angezeigt, bei Heizkörperthermostaten die aktuelle Ventilstellung.

Die Darstellung von Luftfeuchtigkeit und Ist-Temperatur kann in der Definition des Thermostat-Panels ausgeschaltet werden.

Die Solltemperatur wird in den Makros direkt über den Namen des Thermostats eingestellt bzw. abgefragt.

Beispiel:

```
WandthermostatWohnen:=21,5
```

oder

```
WandthermostatWohnen setzen auf 21,5
```

bzw.

```
wenn WandthermostatWohnen > 23 dann
```

Bei Zuweisungen der Soll-Temperatur ist zu beachten, dass diese grundsätzlich in 0,5 Grad Schritten anzugeben ist und im min/max-Bereich des jeweiligen Geräts liegt, dieser kann je nach Thermostat unterschiedlich sein. Wenn Format oder Wertebereich nicht stimmen wird der Wert vom Gerät nicht akzeptiert.

Es können auch die Werte Off und On über Makros im Raumthermostat gesetzt werden, indem für Off der Wert 0, für On der Wert 100 zugewiesen wird.

also z.B.

```
WandthermostatWohnen:=100
```

Der Wert **Off** schliesst das Ventil vollständig, der Wert **On** öffnet das Ventil vollständig. Bei beiden Modi wird die interne Zeittabelle des Thermostats deaktiviert.

Die Ist-Temperatur steht in Makros als Variable *Temperatur* des Raumthermostats zur Verfügung, die Luftfeuchtigkeit als Variable *Luftfeuchtigkeit*,

also z.B.

```
wenn Wandthermostat.Temperatur > 22.5 dann
```

....

Wenn das Wandthermostat-Panel angeklickt wird, öffnet sich ein Fenster in dem die Solltemperatur verändert werden kann.

Bitte beachten Sie:

Es gibt Thermostate älterer Baureihen, die nur im Abstand von einigen Minuten kommunikationsbereit sind, bei diesen Thermostaten kann es Verzögerungen von einigen Minuten geben bis geänderte Werte übertragen und angezeigt werden.

#### Variablen eines Wandthermostats

Ein Wandthermostat hat einige Variablen, diese können je nach Typ des Wandthermostat unterschiedlich sein, bei manchen Typen stehen nicht alle Variablen zur Verfügung.

Hier eine Liste der Variablen:

Variablenname	Verwendung	Typ
Temperatur	Ist-Temperatur	Zahl

Luftfeuchtigkeit	Luftfeuchtigkeit	Zahl
Batteriestatus	aktuelle Batteriespannung	Zahl
Modus	Betriebsmodus	Zahl 0=AUTO, 1=MANUELL
Der Modus kann nicht verändert werden, dazu muss (s. unten)		SETZEWERT benutzt werden
Zeittabelle	Zeittabelle (de)aktivieren	Schalter ein/aus
Heizung	Steuerung Heizkaktor ein/aus	Schalter
Kuehlung	Steuerung Kuehlaktor ein/aus	Schalter
TCTemperatur	nur zur internen Verwendung	

Verwendung der Anweisung **SETZEWERT** bei HM-Thermostaten (gilt nicht für HmIP-Thermostate):  
Für das Heizkörperthermostat HM-CC-RT-DN und das Wandthermostat HMTc-IT-WM-W-EU können über die Anweisung SETZEWERT mehrere Einstellungen vorgenommen werden.  
Der BOOST-Modus wird eingeschaltet mit  
SETZEWERT(ThermostatName, "BOOST\_MODE", 1)

Der AUTO-Modus wird eingeschaltet mit  
SETZEWERT(ThermostatName, "AUTO\_MODE", 1)

Der MANU-Modus wird bei HM-Thermostaten mit  
SETZEWERT(RaumthermostatName, "MANU\_MODE", 17.0)  
gesetzt, wobei die Temperatur eine Konstante mit Kommastelle oder Variable vom Typ Zahl mit Kommastelle sein kann.  
(Gilt nicht für HMIP, dazu siehe Hinweise unter HmIP-Thermostaten!)  
Um die Absenk- bzw. Komforttemperatur einzustellen können folgende Anweisungen benutzt werden:  
SETZEWERT(ThermostatName, "LOWERING\_MODE", 1)  
bzw.  
SETZEWERT(ThermostatName "COMFORT\_MODE", 1)

#### Temperatursteuerung mit Wandthermostaten über Schaltaktoren:

Im Verwaltungsprogramm für Wandthermostate können binäre Aktoren zur Steuerung von Heiz- und Kühl-Geräten angegeben werden. Die über die Aktoren gesteuerten Geräte werden in Abhängigkeit der Soll- und Ist-Temperaturen direkt geschaltet. Dabei können über spezielle Profile die Temperatur-Grenzwerte zum Schalten der Aktoren den individuellen Gegebenheiten angepasst werden. Die Einstellungen der Profile kann mit dem Button unterhalb der Auswahlfelder für die Aktoren aufgerufen werden.

Mit diese Methode der Steuerung und der Möglichkeit eigene Heizprofile zu erstellen können auch träge reagierende Fussbodenheizungen individuell gesteuert werden.

Bitte beachten Sie:

Die Steuerung von Fussbodenheizungen über die direkte Verknüpfung von Wandthermostat und Schaltaktor ist normalerweise nicht möglich, da der hierbei verwendete Regelalgorithmus für Fussbodenheizungen nicht geeignet ist.

Es können unterschiedliche Aktoren zur Heizung und Kühlung verwendet werden, je nach System kann für Heizen und Kühlen aber auch derselbe Schaltaktor benutzt werden.

Dabei ist folgendes zu beachten:

Die Funktionen zur Heizung und Kühlung können standardmässig nicht gleichzeitig aktiv sein.

Die Steuerung welche Funktion aktiv sein soll erfolgt über die Variablen Heizung und Kühlung des Thermostats oder über die "Hauptschalter" für Heizen und Kühlen im Fenster Spezielle Objekte.

Um die Kühlung über Variable zu aktivieren muss die Variable Heizung ausgeschaltet werden und die Variable Kühlung des Thermostats eingeschaltet werden.

Das wir über Makroanweisungen realisiert, z.B.:

WTWohnen.Heizung ausschalten

WTWohnen.Kuehlung einschalten

Die Variable Heizung hat Vorrang, wenn diese eingeschaltet ist, wird die Kühlung unabhängig vom Stand der Variablen Kuehlung deaktiviert.

Eine andere Möglichkeit ist die jeweiligen „Hauptschalter“ zu benutzen. Diese Schalter können Hardwareschalter, aber auch virtuelle Schalter sein.

Die Schalter müssen jeweils im Fenster Spezielle Objekte (Menüpunkt

Konfigurieren->Einstellungen->Reiter Objekte->Button [Spezielle Objekte]) unter "Hauptschalter für Heizen" bzw. "Hauptschalter für Kühlen" ausgewählt werden sein.

Heizung und Kühlung sind dann jeweils nur aktiv wenn der jeweilige Schalter eingeschaltet ist.

Im Gegensatz zur Steuerung über Variablen können mit dieser Methode Heizung und Kühlung auch

gleichzeitig aktiviert werden, das ist jedoch keinesfalls zu empfehlen, da je nach Einstellung der Hysterese in den Heizprofilen beide Aktoren wechselseitig aktiviert werden können. Die Software überprüft ob gemäss der vorgenommen Einstellungen beide Aktoren gleichzeitig eingeschaltet würden, wenn das der Fall ist wird keiner der Aktoren eingeschaltet.

### 1.23.2.3. 19-Tasten-Fernbedienung

---

#### 19-Tasten-Fernbedienung

Die zwei oberen Tasten des Dreier-Blocks auf der rechten Seite können nicht durch das Programm benutzt werden, diese sind für interne Funktionen der Fernbedienung vorgesehen (siehe Bedienungsanleitung der Fernbedienung).

Das letzte Objekt der Fernbedienung (Standardname nach Modulimport ..\_18) ist ein Objekt vom Typ Zeichen, über das Text auf der Fernbedienung ausgegeben werden kann. Zur Ausgabe von Text wird dieser einfach dem Objekt zugewiesen, also z.B.:

```
AnzeigeRC19:="Alarm"
```

Weiterhin können die einzelnen Icons der Fernbedienung über Variablen dieses Objekts aktiviert werden.

Folgende Variablen für die einzelnen Icons stehen zur Verfügung:

**PfeilUnten, PfeilOben, Glocke, Rollo, Licht, Uhr, Tür, Telefon, Szene, Schalter, Fenster.**

Um die entsprechenden Icons anzuzeigen werden die Variablen "eingeschaltet". Also z.B.:

```
AnzeigeRC19.Glocke einschalten
```

```
AnzeigeRC19.Licht einschalten
```

bewirken, dass die Icons "Glocke" und "Glühbirne" angezeigt werden.

Ausser den Variablen für die Icons gibt es noch weitere Variablen um Informationen auf der Fernbedienung auszugeben.

Die Variable **Einheit** kann mit Werten von 0 bis 4 belegt werden und bewirkt die Anzeige einer der folgenden Einheiten:

0 = Keine Einheit

1 = Prozent

2 = Watt

3 = Celsius

4 = Fahrenheit

Mit der Variablen **Beleuchtung** kann die Hintergrundbeleuchtung gesteuert werden.

0 = Keine Hintergrundbeleuchtung

1 = Hintergrundbeleuchtung einschalten

2 = Langsames Blinken

3 = Schnelles Blinken

Mit der Variablen **Beep** kann die Ausgabe eines akustischen Signals aktiviert werden.

0 = Keine Ton

1 = Ton 1

2 = Ton 2

3 = Ton 3

Die Ausgabe auf die Fernbedienung erfolgt nicht automatisch nach Zuweisung von Werten sondern mit einem speziellen Ausgabebefehl nachdem alle gewünschten Werte gesetzt wurden. Dieser Befehl lautet:

**Anzeigen(Objektname)**

Die Ausgabe wird dann einige Zeit auf der Fernbedienung angezeigt. Soll die Ausgabe nochmals angezeigt werden, so muss die Ausgabeanweisung erneut ausgeführt werden.

Alle Variablen für die Icon-Darstellung auf der Fernbedienung können gelöscht werden mit der Anweisung

**LöscheAnzeige(Objektname)**

Dies erspart das Löschen der einzelnen Variablen, wenn die Ausgabe an die Fernbedienung geändert werden soll. Diese Anweisung bewirkt keine Aktualisierung der Anzeige des Moduls, die Aktualisierung muss jeweils mit der Anweisung **Anzeigen(...)** vorgenommen werden.

Beispiel:

Das Wort "LICHT" soll als Text erscheinen und das Glockensymbol der Fernbedienung soll eingeschaltet werden:

```
FB19Tasten_18.Glocke einschalten  
FB19Tasten_18:="WORT"  
Anzeigen(FB19Tasten_18)
```

## 1.23.2.4. MP3 Funkgong mit Signalleuchte HM-OU-CFM-...

---

### MP3 Funkgong mit Signalleuchte HM-OU-CFM-...

Das erste Objekt (der erste Kanal) dieses Moduls ist die Leuchte des MP3-Players. Der Typ des Objekts ist "Zeichen" (also Text). Zur Ansteuerung des MP3-Players werden mehrere Parameter benutzt, die durch Komma getrennt werden.

Parameter sind ein Start/Stopp-Code, die Anzahl der Wiederholungen, die Dauer der Ausgabe und bis zu 10 Codes zur Ausgabe von bis zu 10 unterschiedlichen Lichtsignalen.

Diese Parameter werden in einer Textzeile übergeben und jeweils durch ein Komma getrennt.

Im Folgenden die Beschreibung der einzelnen Parameter, in der Reihenfolge in der sie in der Zeile stehen müssen:

Aktivierungskennung: 1=Starten (normal), 0=Ausgabe beenden

Wiederholungen : Anzahl der Wiederholungen als ganze Zahl zwischen 1 und 255

maximale Gesamtdauer der Ausgabe in Sekunden : ganze Zahl zwischen 1 und 108000,

108000=keine Begrenzung, Ausgabe bis alle Licht-Parameter abgearbeitet sind.

Die nächsten 10 Parameter legen fest, was die Leuchte anzeigt:

	kurz	lang
Rot	17	18
Grün	33	34
Orange	49	50
Blau	65	66
Violett	81	82
Cyan	97	98
Weiss	113	114
Pause	2	
Aus	0	

Beispiel:

Die Leuchte soll dreimal folgende Sequenz anzeigen:

Grün kurz, Rot lang, Pause, Orange kurz

Dazu wird in einem Makro folgende Zuweisung gemacht:

```
MP3Leuchte:="1,3,108000,33,18,2,49"
```

Das zweite Objekt (der zweite Kanal) ist der MP3-Player. Der Typ des Objekts ist "Zeichen" (also Text). Zur Ansteuerung des MP3-Players werden mehrere Parameter benutzt, die durch Komma getrennt werden.

Parameter sind die Lautstärke, die Anzahl der Wiederholungen, die Gesamtspieldauer und der Index der abzuspielenden MP3-Datei bzw. eine Playlist von bis zu 10 MP3-Dateien.

Diese Parameter werden in einer Textzeile übergeben und jeweils durch ein Komma getrennt.

Im Folgenden die Beschreibung der einzelnen Parameter, in der Reihenfolge in der sie in der Zeile stehen müssen:

Lautstärke : Zahl zwischen 0 und 1 ( 0.1 bis 1.0), Dezimaltrennzeichen ist ein Punkt.

Wiederholungen : Anzahl der Wiederholungen als ganze Zahl zwischen 1 und 255

maximale Gesamtdauer der Soundausgabe in Sekunden : ganze Zahl zwischen 1 und 108000, 108000=Dateilänge der MP3-Datei

MP3-Datei bzw. Playlist : der Index der abzuspielenden MP3-Datei(en) jeweils getrennt durch Komma, maximal 10 Dateien.

MP3-Datei bzw. Playlist : der Index der abzuspielenden MP3-Datei(en) jeweils getrennt durch Komma, maximal 10 Dateien.

Beispiel:

Die Dateien 001\_Sound.mp3, 003\_Song und 006\_Ansage2.mp3 sollen mit halber Lautstärke einmal abgespielt werden.

Dazu wird in einem Makro folgende Zuweisung gemacht:

```
MP3Player:="0.5,1,108000,1,3,6"
```

Als Dezimaltrenner für die Lautstärke muss ein Punkt verwendet werden, da das Komma zur Trennung der Parameter verwendet wird.

**Bitte beachten Sie** unbedingt folgende Besonderheit bei diesem Modul:

Es wird immer nach einer Zuweisung gesendet und nicht wie bei anderen Modulen üblich nur wenn der Wert der Objekte sich ändert. Das wurde so eingerichtet, da die abzuspielende Sequenz oftmals gleich ist und bei erneuter Zuweisung auch erneut abgespielt werden soll. Es ist daher wichtig bei der Programmierung zu beachten, dass jede Zuweisung zu einer Ausgabe an das Modul führt auch wenn der Wert sich nicht ändert.

### 1.23.2.5. MP3-Player HM-OU-CM-PCB

---

#### MP3-Player HM-OU-CM-PCB

Der Typ des Objekts ist "Zeichen" (also Text). Zur Ansteuerung des MP3-Players werden mehrere Parameter benutzt, die durch Komma getrennt werden.

Parameter sind die Lautstärke, die Anzahl der Wiederholungen, die Gesamtspieldauer und der Index der abzuspielenden MP3-Datei bzw. eine Playlist von bis zu 10 MP3-Dateien.

Diese Parameter werden in einer Textzeile übergeben und jeweils durch ein Komma getrennt. Im Folgenden die Beschreibung der einzelnen Parameter, in der Reihenfolge in der sie in der Zeile stehen müssen:

Lautstärke : Zahl zwischen 0 und 1 ( 0.1 bis 1.0), Dezimaltrennzeichen ist ein Punkt.

Wiederholungen : Anzahl der Wiederholungen als ganze Zahl zwischen 1 und 255

maximale Gesamtdauer der Soundausgabe in Sekunden : ganze Zahl zwischen 1 und 108000, 108000=Dateilänge der MP3-Datei

MP3-Datei bzw. Playlist : der Index der abzuspielenden MP3-Datei(en) jeweils getrennt durch Komma, maximal 10 Dateien.

MP3-Datei bzw. Playlist : der Index der abzuspielenden MP3-Datei(en) jeweils getrennt durch Komma, maximal 10 Dateien.

Beispiel:

Die Dateien 001\_Sound.mp3, 003\_Song und 006\_Ansage2.mp3 sollen mit halber Lautstärke einmal abgespielt werden.

Dazu wird in einem Makro folgende Zuweisung gemacht:

```
MP3Player:="0.5,1,108000,1,3,6"
```

Als Dezimaltrenner für die Lautstärke muss ein Punkt verwendet werden, da das Komma zur Trennung der Parameter verwendet wird.

**Bitte beachten Sie** unbedingt folgende Besonderheit bei diesem Modul:

Es wird immer nach einer Zuweisung gesendet und nicht wie bei anderen Modulen üblich nur wenn der Wert der Objekte sich ändert. Das wurde so eingerichtet, da die abzuspielende Sequenz oftmals gleich ist und bei erneuter Zuweisung auch erneut abgespielt werden soll. Es ist daher wichtig bei der Programmierung zu beachten, dass jede Zuweisung zu einer Ausgabe an das Modul führt auch wenn der Wert sich nicht ändert.



#### **1.23.2.6. Statusanzeige HM-Dis-WM55**

---

## Statusanzeige HM-Dis-WM55

Mit der Statusanzeige können über das Drücken auf den oberen oder unteren Rand aktuelle Informationen abgerufen werden.

Die Statusanzeige hat zwei Taster, durch die Betätigung wird jeweils das Makro eines Tasters aufgerufen. In diesem Makro kann dann mit der Anweisung SETZEWERT der Text erstellt werden, der für einige Sekunden auf dem Display der Statusanzeige angezeigt wird. Die Ausgabe an die Statusanzeige kann nur im Makro eines Tasters der Statusanzeige mit der Anweisung **Setzewert** mit Wertecode "StatusDisplay" (siehe Beispiel) erzeugt werden. Die Statusanzeige erwartet die Zuweisung eines Textes Sekundenbruchteile nach Betätigung der Taste, daher dürfen in dem Makro keine Verzögerungen auftreten, also keine WARTE-Anweisungen oder Schleifen benutzt werden.

Das Schlüsselwort "selbst" kann innerhalb eines Makros immer benutzt werden um das "eigene" Objekt des Makros zu bezeichnen, das ist oftmals einfacher und schneller als jedes mal im Makro den "eigenen" Objektnamen zu schreiben.

Die Anweisungen des folgenden Beispiels müssen also im Makro eines Tasters der Statusanzeige stehen.

```
wenn selbst = "kurz" dann
  Setzewert(selbst,"StatusDisplay","Kurz<<gedrückt")
sonst
  Setzewert(selbst,"StatusDisplay","Lang<<gedrückt")
endewenn
```

Ein Zeilenvorschub wird mit zwei kleiner-Zeichen << hintereinander erzeugt

Der Text kann farbig dargestellt werden, am Ende einer Zeile kann ein Icon angezeigt werden.

Um die Textfarbe zu setzen wird ein Backslash ("\") gefolgt vom Buchstaben "F" (für Farbe) und einer Zahl oder einem Schlüsselwort, abgeschlossen wird der Code immer mit einem Semikolon (unbedingt erforderlich!).

Es stehen folgende Farben zur Verfügung:

Weiss -> Schlüsselwort WEISS oder Zahl 1

Rot -> Schlüsselwort ROT oder Zahl 2

Orange -> Schlüsselwort ORANGE oder Zahl 3

Gelb -> Schlüsselwort GELB oder Zahl 4

Grün -> Schlüsselwort GRÜN oder Zahl 5

Blau -> Schlüsselwort BLAU oder Zahl 6

Um ein Icon zu setzen wird ein Backslash ("\") gefolgt vom Buchstaben "I" (für Icon) und einer Zahl oder einem Schlüsselwort, abgeschlossen wird der Code immer mit einem Semikolon (unbedingt erforderlich!).

Es stehen folgende Icons zur Verfügung:

Lampe aus -> Schlüsselwort AUS oder Zahl 1

Lampe an -> Schlüsselwort AN oder Zahl 2

Schloss offen -> Schlüsselwort OFFEN oder Zahl 3

Schloss geschlossen -> Schlüsselwort GESCHLOSSEN oder Zahl 4

Fehler -> Schlüsselwort FEHLER oder Zahl 5

OK -> Schlüsselwort OK oder Zahl 6

Info -> Schlüsselwort INFO oder Zahl 7

Nachricht -> Schlüsselwort NACHRICHT oder Zahl 8

Servicemeldung -> Schlüsselwort SERVICE oder Zahl 9

Signal grün -> Schlüsselwort GRÜN oder Zahl 10

Signal gelb -> Schlüsselwort GELB oder Zahl 11

Signal rot -> Schlüsselwort ROT oder Zahl 12

Als erstes muss der Text stehen, Icon- oder Farbcodes dürfen nicht am Anfang stehen.

Der Text um zwei Zeilen zu erzeugen, wobei die erste Zeile gelb und die zweite Zeile blau dargestellt wird, am Ende der ersten Zeile ein offenes Schloss, am Ende der zweiten Zeile ein Info Zeichen sieht beispielsweise folgendermassen aus:

"Erste Zeile\FGELB;\IOPEN;<<zweite Zeile\F6;\I7;"



### 1.23.2.7. EPaper-Display HM-Dis-EP-WM55

---

#### EPaper-Display HM-Dis-EP-WM55

Dieses Gerät hat eine E-Paper Displayanzeige und zwei Tasten. Es kann jederzeit eine Ausgabe auf das Display erfolgen, wobei es auch möglich ist eine Tonfolge auszugeben und die LED aufblinken zu lassen. Im Display können bis zu drei Zeilen angezeigt werden.

Wenn eine der beiden Tasten betätigt wird, wird die Tastermeldung empfangen und im Display erscheint der Standardtext, der in der WEB-UI für die beiden Tasten hinterlegt ist. Eine neuer beliebiger Anzeigetext kann erst nach einer Wartezeit von mindestens 5 Sekunden an das Display geschickt werden.

Beispiel für ein Makro:

```
wenn Terrassentuer geschlossen dann
  AktuellerStatus:="Die Türe ist<<geschlossen\IGESCHLOSSEN;"
sonst
  AktuellerStatus:="Die Türe ist<<geöffnet\IOFFEN;"
EDisplay:=AktuellerStatus
```

wobei AktuellerStatus hier eine Variable Typ Zeichen ist.

Ein Zeilenvorschub wird mit zwei kleiner-Zeichen << hintereinander erzeugt

Am Ende einer Zeile kann ein Icon angezeigt werden.

Um ein Icon zu setzen wird ein Backslash ("\") gefolgt vom Buchstaben "I" (für Icon) und einem Schlüsselwort, abgeschlossen wird der Code immer mit einem Semikolon (unbedingt erforderlich!).

Es stehen folgende Icons zur Verfügung:

Lampe aus -> Schlüsselwort AUS oder Zahl 1  
Lampe an -> Schlüsselwort AN oder Zahl 2  
Schloss offen -> Schlüsselwort OFFEN oder Zahl 3  
Schloss geschlossen -> Schlüsselwort GESCHLOSSEN oder Zahl 4  
Fehler -> Schlüsselwort FEHLER oder Zahl 5  
OK -> Schlüsselwort OK oder Zahl 6  
Info -> Schlüsselwort INFO oder Zahl 7  
Nachricht -> Schlüsselwort NACHRICHT oder Zahl 8  
Servicemeldung -> Schlüsselwort SERVICE oder Zahl 9

Mit der Ausgabe auf das Display kann zusätzlich eine Tonfolge bestimmt werden.

Um eine Tonfolge auszugeben wird ein Backslash ("\") gefolgt vom Buchstaben "S" (für Sound) und einer Ziffer für die Tonfolge, abgeschlossen wird der Code immer mit einem Semikolon.

Folgende Tonfolgen sind möglich:

1 -> lang lang  
2 -> lang kurz  
3 -> lang kurz kurz  
4 -> kurz  
5 -> kurz kurz  
6 -> lang

Weiterhin kann die Anzahl der Wiederholungen für die Tonfolge bestimmt werden, das ist der zweite Wert innerhalb des Sound-Parameters.

Als dritter Wert des Sound-Parameters kann der Abstand der zu wiederholenden Tonfolge angegeben werden, wobei die Einheit ein 10-Sekunden-Intervall ist, der Wert 3 würde also einen Abstand von 30 Sekunden zwischen den Tonfolgen bedeuten.

Für die Tonfolge gibt es also drei Werte, die durch Komma getrennt werden;

\S<Tonfolge>,<Wiederholungen>,<Zeitabstand in 10 Sekundenintervall>;

Wenn der Wert für Wiederholungen und Zeitabstand weggelassen wird, wird der Ton nur einmal

ausgegeben.

Am Ende des Codes ist ein Semikolon erforderlich!

Weiterhin ist es möglich bei Ausgabe auf das Display die LED kurz aufblinken zu lassen, wobei unterschiedliche Farben benutzt werden können.

Um die LED aufblinken zu lassen wird ein Backslash ("\") gefolgt vom Buchstaben "L" (für LED) und einer Ziffer für die Farbe, abgeschlossen wird der Code immer mit einem Semikolon.

Folgende Farben können durch die Ziffer im Code bestimmt werden:

1 -> rot

2 -> grün

3 -> orange

Als erstes muss der Text stehen, Codes für Icons, Sound und LED dürfen nicht am Anfang stehen.

Hier ein Beispiel mit folgenden Eigenschaften:

Es werden drei Zeilen angezeigt, wobei die erste Zeile als Icon eine Glühbirne bekommt, die zweite Zeile ein geschlossenes Schloss und die dritte Zeile das Infozeichen bekommt. Als Soundausgabe wird ein langes Tonsignal mit einer Wiederholung im Abstand von 10 Sekunden festgelegt und ein Blinken der LED in Orange.

Dazu muss die Ausgabe wie folgt aufgebaut werden:

```
EDisplay:="Erste Zeile\IAN;<<Zweite Zeile\IGESCHLOSSEN;<<Dritte Zeile\I7;\S6,1,1;\L3;"
```

## 1.23.2.8. WinMatic

---

### WinMatic

Die Steuerung der WinMatic ist etwas speziell, intern wird die Position mit Kommazahlen zwischen 0 und 1 gesteuert, verriegelt wird mit -0,005.

Da die Steuerung des Geräts prinzipiell über Zahlen mit nur einem Datenpunkt erfolgt, lässt sich die Steuerung nicht wie bei anderen Geräten auch über Zustände realisieren.

Um dieses spezielle Gerät möglichst einfach abbilden zu können wird der interne Wert umgerechnet auf Prozentzahlen zwischen 0 und 100, als "Verriegelungswert" wird -1 benutzt.

Eine weitere Besonderheit dieses Geräts ist, dass man nicht einfach den Wert setzen kann, wenn der aktuelle Zustand "verriegelt", also -1 ist. Zuerst muss "geöffnet" werden, indem der Wert auf 0 gesetzt wird. Erst wenn dieser Verstellvorgang am Gerät abgeschlossen ist, kann der Wert für den Öffnungswinkel zum Gerät übertragen werden.

Hier ein Beispielmakro zur Ansteuerung einer WinMatic, in dem ein Fenster zum Lüften für 90 Minuten geöffnet wird:

```
wenn WAKtor < 0 dann // wenn verriegelt
  WAKtor:=0          // Fenstergriff öffnen
  warte 20 Sekunden  // warten bis der Öffnungsvorgang abgeschlossen ist
endwenn
WAKtor:=50           // Fenster kippen auf 50 Prozent
warte 90 Minuten     // 90 Minuten geöffnet lassen
WAKtor:=-1           // und zum Schluss schliessen und verriegeln
```

## 1.23.3. HomeMatic IP

---

### HomeMatic IP

Hier werden Hinweise zu speziellen HomeMatic-IP-Geräten beschrieben.  
HomeMatic-Geräte erkennt man daran, dass die Typbezeichnung mit HmIP-... beginnt.

Bei den HomeMatic und HomeMatic IP - Geräten, bei denen die Eingänge umkonfigurierbar sind ist folgendes zu beachten:

Die aktuelle Konfiguration des Eingangstyps wird beim Import nicht automatisch eingestellt, es muss im Programm jeweils der passende Objekttyp ausgewählt werden.

Das Verhalten bei Tastern hat sich gegenüber HomeMatic-Geräten leider geändert. Bei Geräten mit Tastern, die auch einen langen Tastendruck erlauben, wird dieser mit den Standardeinstellungen nicht mehr nur einmal, sondern ununterbrochen gesendet. Um das zu vermeiden muss in der Konfiguration des Tasters in der WEB-UI der Wert ***Timeout für langen Tastendruck*** auf 100ms gesetzt werden.

### **1.23.3.1. HmIP-Thermostate**

---



## HmIP-Thermostate

Zur Visualisierung des Thermostats wird ein spezielles Panel benutzt. Die Temperatur auf der linken Seite des Panels ist die Solltemperatur, die über die Visualisierung verändert werden kann. Die Temperatur auf der rechten Seite des Panels ist die aktuelle Ist-Temperatur, die vom Thermostat empfangen wurde. In der Mitte wird die aktuelle gemessene Luftfeuchtigkeit angezeigt, bei Heizkörperthermostaten die aktuelle Ventilstellung. Die Darstellung von Luftfeuchtigkeit und Ist-Temperatur kann in der Definition des Thermostat-Panels ausgeschaltet werden.

Die Solltemperatur wird in den Makros direkt über den Namen des Thermostats eingestellt bzw. abgefragt.

Beispiel:

```
WandthermostatWohnen:=21,5
```

oder

```
WandthermostatWohnen setzen auf 21,5
```

bzw.

```
wenn WandthermostatWohnen > 23 dann
```

Es können auch die Werte Off und On über Makros im Raumthermostat gesetzt werden, indem für Off der Wert 0, für On der Wert 100 zugewiesen wird.

also z.B.

```
WandthermostatWohnen:=100
```

Der Wert **Off** schliesst das Ventil vollständig, der Wert **On** öffnet das Ventil vollständig. Bei beiden Modi wird die interne Zeittabelle des Thermostats deaktiviert.

Bei Zuweisungen der Soll-Temperatur ist zu beachten, dass diese grundsätzlich in 0,5 Grad Schritten anzugeben ist und im min/max-Bereich des jeweiligen Geräts liegt, dieser kann je nach Thermostat unterschiedlich sein. Wenn Format oder Wertebereich nicht stimmen wird der Wert vom Gerät nicht akzeptiert.

Die Ist-Temperatur steht in Makros als Variable *Temperatur* des Raumthermostats zur Verfügung, die Luftfeuchtigkeit als Variable *Luftfeuchtigkeit*,

also z.B.

```
wenn Wandthermostat.Temperatur > 22.5 dann
```

....

Wenn das Wandthermostat-Panel angeklickt wird, öffnet sich ein Fenster in dem die Soll-Temperatur verändert werden kann.

### Variablen eines Wandthermostats

Ein Wandthermostat hat einige Variablen, diese können je nach Typ des Wandthermostat unterschiedlich sein, bei manchen Typen stehen nicht alle Variablen zur Verfügung.

Hier eine Liste der Variablen:

Variablenname	Verwendung	Typ
Temperatur	Ist-Temperatur	Zahl
Luftfeuchtigkeit	Luftfeuchtigkeit	Zahl
Batteriestatus	aktuelle Batteriespannung	Zahl (nicht bei der
Wired-Version)		
Modus	Betriebsmodus	Zahl 0=AUTO, 1=MANUELL
Der Modus kann nicht verändert werden, dazu muss	SETZEWERT benutzt werden	
(s. unten)		
Zeittabelle	Zeittabelle (de)aktivieren	Schalter ein/aus (nur wenn
für das Thermostat	eine Zeittabelle angelegt wird)	
Heizung	Steuerung Heizfaktor ein/aus	Schalter
Kuehlung	Steuerung Kuehlfaktor ein/aus	Schalter
TCTemperatur	nur zur internen Verwendung	(nicht bei allen
Thermostaten)		

Verwendung der Anweisung **SETZEWERT** bei HmIP-Thermostaten (gilt nicht für HM-Thermostate):



## 1.23.3.2. HmIP-Rauchmelder

[< Vorherige Seite](#)

[Nächste Seite >](#)

### HmIP-Rauchmelder

Bei den HmIP-Rauchmeldern kann die Sirene über die Zentrale geschaltet werden.

Das passiert mit der Anweisung [SETZEWERT](#).

Zum Einschalten wird folgende Anweisung benutzt:

```
SETZEWERT(MeinHMIPRauchmelder, "SMOKE_DETECTOR_COMMAND", "INTRUSION_ALARM")
```

Zum Ausschalten wird folgende Anweisung benutzt:

```
SETZEWERT(MeinHMIPRauchmelder, "SMOKE_DETECTOR_COMMAND",  
"INTRUSION_ALARM_OFF")
```

[< Vorherige Seite](#)

[Nächste Seite >](#)

### **1.23.3.3. HmIP-ASIR - Alarmsirenen**

---

## HmIP-ASIR - Alarmsirenen

Diese Alarmsirenen erlaubt die Ausgabe optischer und akustischer Alarmsignale. Die Aktivierung des Alarms erfolgt durch Ausgabe eines codierten Textes mit drei Parametern an die Alarmsirene.

Der erste Parameter bestimmt welches akustische Signal ausgegeben werden soll, der zweite Parameter bestimmt das optische Signal, der dritte Parameter die Zeit wie lange die Alarmsignale ausgegeben werden sollen.

Die Angabe der Signale erfolgt als Index einer Tabelle von Auswahlmöglichkeiten.

Folgende akustischen Signale stehen zur Verfügung (Parameter 1):

- 0 = kein akustisches Signal
- 1 = Frequenz steigend
- 2 = Frequenz fallend
- 3 = Frequenz steigend/fallend
- 4 = Frequenz tief/hoch
- 5 = Frequenz tief/mittel/hoch
- 6 = Frequenz hoch ein/aus
- 7 = Frequenz hoch ein, lang aus
- 8 = Frequenz tief ein/aus, hoch ein/aus
- 9 = Frequenz tief ein - lang aus, hoch ein - lang aus

Folgende optischen Signale stehen zur Verfügung (Parameter 2):

- 0 = kein optisches Signal
- 1 = Abwechselndes langsames Blinken
- 2 = Gleichzeitiges langsames Blinken
- 3 = Gleichzeitiges schnelles Blinken
- 4 = Gleichzeitiges kurzes Blinken
- 5 = Bestätigungssignal 0 - lang lang
- 6 = Bestätigungssignal 1 - lang kurz
- 7 = Bestätigungssignal 2 - lang kurz kurz

Die Zeit (Parameter 3) wird wie folgt angegeben:

Zeitwert direkt gefolgt von der Zeiteinheit.

Als Zeitwert kann eine Zahl zwischen 1 und 60 angegeben werden, als Zeiteinheit S für Sekunden oder M für Minuten.

Beispiel für eine Ausgabe innerhalb eines Makros:

```
Alarmsirene:="3,1,15S"
```

Diese Sequenz bewirkt, dass für 15 Sekunden ein akustisches Signal mit steigender und fallender Frequenz ausgegeben wird und die LEDs dabei abwechselnd langsam blinken.

**Bitte beachten Sie** unbedingt folgende Besonderheiten bei diesem Modul:

Zur Ansteuerung der Alarmsirene wird vom BidCoS ein anderes Verfahren benutzt als für Aktoren normalerweise üblich. Dadurch bedingt können die Reaktionszeiten schwanken und es kann über 5 Sekunden dauern bis die Alarmsirene reagiert.

Es wird immer nach einer Zuweisung gesendet und nicht wie bei anderen Modulen üblich nur wenn der Wert der Objekte sich ändert. Das wurde so eingerichtet, da die abzuspielende Sequenz oftmals gleich ist und bei erneuter Zuweisung auch erneut abgespielt werden soll. Es ist daher wichtig bei der Programmierung zu beachten, dass jede Zuweisung zu einer Ausgabe an das Modul führt auch wenn der Wert sich nicht ändert.

## 1.23.3.4. HmIP-BSL - Schaltaktor mit Signalleuchte

[< Vorherige Seite](#)

[Nächste Seite >](#)

### HmIP-BSL - Schaltaktor mit Signalleuchte

Dieser Schaltaktor verfügt neben dem Schaltausgang über zwei Signalleuchten die jeweils in unterschiedlicher Farbe und einer Helligkeit zwischen 0 und 100 aktiviert werden können.

Beim Import haben diese Signalleuchten jeweils den Namen *Farbsignal...* .

Der Name kann natürlich wie bei anderen Objekten geändert werden.

Jedes Objekt zu den Signalleuchten hat zusätzlich eine Variable mit Namen *Farbe*, die mit Zahlen zwischen 0 und 7 belegt werden kann.

Folgenden Zahlen und zugehörige Farben können verwendet werden:

0 = aus  
1 = blau  
2 = grün  
3 = türkis  
4 = rot  
5 = violett  
6 = gelb  
7 = weiss

Zu diesem Gerät werden bei Import 5 Objekte erzeugt:

Zwei Taster (Kanäle 1 und 2) ,

der eigentliche Schaltaktor (Index 1, Kanal 4) zum Schalten des Schaltaktors

und

Signalleuchte 1 (Index 2, Kanal 8)

Signalleuchte 2 (Index 3, Kanal 12)

jeweils als Dimmer mit Variable *Farbe* zum Setzen der Farbe

Beispiel:

```
AktorSig einschalten  
Signalleuchte1.Farbe:=4  
Signalleuchte1:=80
```

Schaltet den Aktor 1 und setzt die Farbe einer Leuchte auf rot und die Helligkeit auf 80%.

[< Vorherige Seite](#)

[Nächste Seite >](#)

### 1.23.3.5. HmIP-DLD - Türschlossantrieb

---

#### HmIP-DLD - Türschlossantrieb

Dieser HmIP-Türschlossantrieb hat leider ein vollkommen anderes Protokoll als die KeyMatic der HomeMatic-Serie.

Der Türschlossantrieb hat zwei separate Objekte mit unterschiedlichen Zuständen für Status und Aktionen, die Zustände des Aktionen-Objekts werden zur Steuerung des Antriebs benutzt, das Status-Objekt wird nur zur Anzeige des aktuellen Zustands benutzt, eine Zustandsänderung im Status-Objekt hat keinen Einfluss auf den Antrieb und macht daher keinen Sinn.

Zur Steuerung werden die Zustände des Aktionen-Objekts benutzt, die Makroanweisungen sehen also z.B. so aus:

Türaktion:="verschliessen"

Türaktion:="entriegeln"

Türaktion:="öffnen"

Normalerweise wird immer nur an die Hardware gesendet, wenn der Zustand eines Objekts sich geändert hat. Das ist bei dem Aktinen-Objekt dieses Moduls anders. Da der Zustand des Objekts sich durch die Rückmeldung des Geräts nicht ändert (weil diese an das Status-Objekt geht), wird bei diesem Objekt immer eine Meldung an die Hardware generiert wenn eine Zuweisung erfolgt. Daher ist bei der Programmierung darauf zu achten, dass nicht zu viele unnötige Zuweisungen gemacht werden.

## 1.23.3.6. HmIP-MP3P - Kombisignalgeber

[< Vorherige Seite](#)

[Nächste Seite >](#)

### HmIP-MP3P - Kombisignalgeber

Mit diesem Gerät können auf einer SD-Karte gespeicherte Sound-Dateien abgespielt werden und die integrierte Signalleuchte kann in verschiedenen Farben und Helligkeitsstufen gesteuert werden.

Steuerung des Sounds (Index 1, Kanal 2).

Die Lautstärke wird durch Zuweisung einer Zahl zwischen 0 und 100 an den ersten Aktor des Geräts (Index 1, Kanal 2) definiert. Über die Variable *Soundfile* des Aktors wird mit einer Zahl zwischen 1 und 252 angegeben welche Sounddatei abgespielt werden soll.

Sobald die Lautstärke des Geräts verändert wird, wird die Datei abgespielt.

Wenn die Variable den Wert 0 hat wird der interne Sound abgespielt.

Beispiel für Makroanweisungen um Datei 7 mit 50% Lautärke auszugeben:

```
Sound.Soundfile:=7
```

```
Sound:=50
```

Steuerung der Signalleuchte (Index 2, Kanal 6)

Die Helligkeit wird durch Zuweisung einer Zahl zwischen 0 und 100 bestimmt. Über die Variable *Farbe* des Aktors wird mit einer Zahl zwischen 0 und 7 die Farbe der Signalleuchte festgelegt.

Sobald die Helligkeit verändert wird, wird die Lampe eingeschaltet.

Folgenden Zahlen und zugehörige Farben können verwendet werden:

0 = aus

1 = blau

2 = grün

3 = türkis

4 = rot

5 = violett

6 = gelb

7 = weiss

Beispiel für Makroanweisungen um die LED rot leuchten zu lassen mit 80% Helligkeit:

```
Farbleuchte.Farbe:=4
```

```
Farbleuchte:=80
```

[< Vorherige Seite](#)

[Nächste Seite >](#)



### 1.23.3.7. HmIP-SWD Wassersensor

[< Vorherige Seite](#)

[Nächste Seite >](#)

#### HmIP-SWD Wassersensor

Der HmIP-Wassersensor hat leider eine etwas seltsame Logik.  
Üblicherweise gab es bei Wassersensoren immer einen Datenpunkt mit drei möglichen Zuständen OK, Feuchtigkeit, Wasser.  
Das ist bei diesem HmIP-Wassersensor leider anders.

Es gibt insgesamt 3 Datenpunkte, einmal den "Hauptdatenpunkt" ALARMSTATE mit den Zuständen OK und Alarm und für Feuchtigkeit und Wasser jeweils einen separaten Datenpunkt, ebenfalls mit den Zuständen OK und Alarm.

Für den Wassersensor wird daher ein Objekt für den Datenpunkt ALARMSTATE erstellt. Um nicht drei unterschiedliche Objekt für das Modul zu haben und es zu unübersichtlich werden zu lassen sind die Werte für Feuchtigkeit und Wasser in zwei Variablen des Objekt untergebracht. Um diese in einer Visualisierung anzuzeigen können virtuelle Objekte erstellt werden, denen im Makro des Objekts zum Wassersensor die Variablenwerte zugewiesen werden.

Dazu wird die Option *Ausführung bei Empfang* aktiviert und z.B. folgende Anweisungen im Makro geschrieben:

```
ObjWasser:=Wassersensor.Wasser  
ObjFeuchtigkeit:=Wassersensor.Feuchtigkeit
```

wobei ObjWasser und ObjFeuchtigkeit in diesem Beispiel virtuelle Objekt mit dem Objekttyp *ASchalter* sind.

Wenn das Objekt des Wassersensors den Zustand Alarm bekommt, hat immer auch die Variable *Feuchtigkeit* den Zustand Alarm.

Wenn Wasser detektiert wird hat auch die Variable *Wasser* den Zustand Alarm.

[< Vorherige Seite](#)

[Nächste Seite >](#)

### 1.23.3.8. HmIP-WHS2 - Heizungsaktor

[< Vorherige Seite](#)

[Nächste Seite >](#)

#### HmIP-WHS2 - Heizungsaktor

Dieses Gerät ist für spezielle Anwendungsfunktionen vorgesehen und arbeitet etwas anders als normale Schaltaktoren, bei denen der Ausgang jeweils über einen Kanal, in der Software also ein Objekt, gesteuert wird.

Dieser Heizungsaktor hat 2 Aktoren (Relais), die über jeweils 3 unterschiedliche Kanäle angesteuert werden.

Dabei wird der erste Ausgang eingeschaltet, sobald einer der Kanäle 2-4 (erkennbar an der Ziffer hinter dem Doppelpunkt der Adresse) eingeschaltet wird.

Ausgeschaltet wird der Ausgang nur wenn alle Kanäle ausgeschaltet sind, solange ein zugehöriges Objekt noch eingeschaltet bleibt der Ausgang eingeschaltet.

Der zweite Ausgang funktioniert genauso und wird über die Kanäle 6-8 gesteuert.

Es ist also kein Fehler wenn der Ausgang nicht ausgeschaltet wird wenn nur eines der zugehörigen Objekte ausgeschaltet wird, es müssen alle Objekte eines Ausgangs ausgeschaltet werden, damit das Relais abschaltet.

[< Vorherige Seite](#)

[Nächste Seite >](#)

### **1.23.3.9. HmIP-WRCD EPaper-Display**

---

## HmIP-WRCD EPaper-Display

Dieses Gerät hat eine E-Paper Displayanzeige und zwei Tasten. Es kann eine Ausgabe in das Display geschrieben werden und es können verschiedene Tonsignale ausgegeben werden.

Im Display können 5 einzelne Zeilen angezeigt werden, Zeile 1 ist die oberste Zeile, in der normalerweise der Text für Taster 1 steht, Zeile 5 ist die unterste Zeile, in der normalerweise der Text für Taster 2 steht.

Taster 1 und Taster 2 können als normale Taster behandelt werden, es ist eine Registrierung im Fenster [Schaltsensoren](#) erforderlich damit Tastendrucke empfangen werden können.

Zur Ausgabe in das Display können mehrere Parameter gesetzt werden, alle Parameter ausser dem Text sind optional.

Die Parameter werden durch Komma getrennt, die Reihenfolge der Parameter sieht folgendermassen aus:

Text, Zeile, Ausrichtung, Hintergrundfarbe, Textfarbe, Icon, Tonsignal, Zeitabstand, Wieder

### Beschreibung der Parameter:

Text : Der auszugebende Text, maximal 14 Zeichen, wenn zusätzlich ein Icon ausgegeben werden soll darf der Text nur 11 Zeichen haben damit das Icon vollständig zu sehen ist.

Alle weiteren Parameter sind optional, wenn keine Zeile und Ausrichtung angegeben wird, wird mittig in Zeile 3 geschrieben.

Zeile : Es können die Zeilen 1-5 benutzt werden, wobei 1 und 5 jeweils die oberste bzw. unterste Zeile ist, in der der Text für den Taster steht.

Ausrichtung : 0=links, 1=mittig, 2=rechts

Hintergrundfarbe : 0 = WEISS  
1 = SCHWARZ

Textfarbe : 0 = WEISS  
1 = SCHWARZ

Icon : 0 = kein Icon  
1 = LAMP\_OFF  
2 = LAMP\_ON  
3 = PADLOCK\_OPEN  
4 = PADLOCK\_CLOSED  
5 = ERROR  
6 = EVERYTHING\_OKAY  
7 = INFORMATION  
8 = NEW\_MESSAGE  
9 = SERVICE\_MESSAGE  
10 = SUN  
11 = MOON  
12 = WIND  
13 = CLOUD  
14 = THUNDERSTORM  
15 = DRIZZLE  
16 = CLOUD\_AND\_MOON  
17 = RAIN  
18 = SNOW  
19 = CLOUD\_AND\_SUN  
20 = CLOUD\_SUN\_AND\_RAIN  
21 = SNOWFLAKE  
22 = RAINDROP



## 1.23.3.10. HmIP-WRCR - Drehtaster

[< Vorherige Seite](#)

[Nächste Seite >](#)

### HmIP-WRCR - Drehtaster

Der HmIP-WRCR-Drehtaster ist in erster Linie zur Verwendung in direkten Verknüpfungen zusammen mit Dimmern geeignet.

Beim Drehen des Knopfes werden keine Zahlenwerte zur Drehbewegung gemeldet, sondern je nach Dauer der Drehbewegung mehrere Ereignisse "PRESS\_LONG" hintereinander. Diese Ereignisse werden dann an die Logikschichten, also in diesem Fall an die ExecEngine, gemeldet.

Beim Import des Geräts werden 3 Taster-Objekte erstellt.

Das erste Taster-Objekt bildet die Druckfunktion des Drehrads ab, funktioniert wie ein normaler Taster und kann die Zustände "kurz" und "lang" melden.

Das zweite Taster-Objekt des Geräts wechselt in den Zustand "lang" wenn der Drehknopf mit dem Uhrzeigersinns gedreht wird.

Das dritte Taster-Objekt des Geräts wechselt in den Zustand "lang" wenn der Drehknopf entgegen des Uhrzeigersinns gedreht wird.

Beim Import wird standardmässig für die Dreh-Taster eine Reset-Zeit in den Ruhezustand von 2 Sekunden eingetragen, zwei Sekunden nach der letzten "lang"-Meldung einer Drehbewegung wird also der Zustand des Taster-Objekts auf "aus" gesetzt.

Wenn eine Reset-Zeit von nur 1 Sekunde benutzt wird, kann es passieren, dass während der Drehbewegung zwischendurch kurz der Zustand "aus" gesetzt wird.

Für die Nutzung in der Programmierung ist das Gerät aufgrund der beschriebenen Funktionsweise nur bedingt geeignet, es wird vorzugsweise für direkte Verknüpfungen verwendet.

[< Vorherige Seite](#)

[Nächste Seite >](#)

### 1.23.3.11. HmIP-USBSM

---

#### HmIP-USMSM

Bei diesen USB-Aktor ist das Verhalten zum Zeitpunkt der Implementierung nicht entsprechend der Geräte- und Protokoll-Beschreibung.

Beim Drücken der Taste auf Kanal 1 wird der Aktor umgeschaltet, aber es werden keine Tastermeldungen generiert.

Eventuell wird das mit einem zukünftigen Firmwareupdate für das Gerät behoben.

### 1.23.3.12. HmIP-MIOB

---

#### HmIP-MIOB

Bei diesen Gerät ist das Verhalten zum Zeitpunkt der Implementierung nicht entsprechend der Geräte- und Protokoll-Beschreibung.

Bei Änderung des Status der Eingänge werden keine Meldungen an die Zentrale geschickt.



### 1.23.3.13. HmIP-WUA/ELV-SH-WUA

---

#### HmIP-WUA / ELV-SH-WUA

Bei diesem Gerät wurde zusätzlich zum in der Beschreibung und WEB-UI vorhandenen Ausgang 1 (OUT1) auf Kanal 2 auch die Ansteuerung des Ausgangs 2 (OUT2) auf Kanal 4 implementiert. weiterführende Infos zum zweiten Ausgang können der Bedienungsanleitung des Geräts entnommen werden.

### 1.23.3.14. ELV-SH-WSC

---

#### ELV-SH-WSC - Servosteuerung

Bei der Servosteuerung wird die Position des Servomotors in Prozentzahlen von 0-100 angegeben. Diese Position wird dann sofort ohne Verzögerung der Fahrzeit dauerhaft eingestellt.

Optional ist es auch möglich neben der Position die Fahrzeit zur Verstellung und die Zeit in der der Motor in der neuen Position verbleibt bis er sich wieder zurückstellt anzugeben.

Das geschieht über die Anweisung SETZEWERT mit entsprechenden Parameter unter dem Datenpunkt "COMBINED\_PARAMETER".

Eine solche Anweisung sieht dann z.B. so aus:

```
Setzewert ( Servo1 , "COMBINED_PARAMETER" , "L=60 , OT=20 , RT=5 " )
```

Der erste Parameter der Anweisung SETZEWERT ist das Objekt welches angesteuert werden soll, der zweite Parameter ist immer "COMBINED\_PARAMETER". Der dritte Parameter ist eine Variable vom Typ Zeichen oder eine Konstante in Hochkomma, die drei weitere Parameter zur Einstellung des Motors beinhalten.

- Parameter L ist der Level für den Servomotor zwischen 0 und 100%.
- Parameter OT steht für "ON\_TIME" und ist die Zeit in Sekunde, in der die neue Position verbleiben soll wobei der höchste Wert 8580000 für unendliche Verweildauer steht.
- Parameter RT steht für "RAMP\_TIME" und ist die Fahrzeit in Sekunden für die Verstellung zur aktuellen Position.

Im oben angegebenen Beispiel fährt der Motor also auf die Position 60% innerhalb von 5 Sekunden und bleibt dort für 20 Sekunden.

### 1.23.3.15. HmIP-WKP Keypad

---

#### HmIP-WKP Keypad

Das Keypad wurde anscheinend in erster Linie zur Verwendung mit Direktverknüpfungen konzipiert und ist zur Verwendung an einer Zentrale leider nur eingeschränkt geeignet.

Wenn ein korrekter Code eingegeben wird und eine der die Schlosstasten "Verriegeln" oder "Entriegeln" gedrückt wird, werden Meldungen nur auf Kanal 0 an die Zentrale geschickt. Der Datenpunkt CODE\_ID enthält die Benutzernummer des eingegebenen Codes. Wenn ein ungültiger Code eingegeben wurde wird als CODE-ID 32 empfangen.

Es kommen aber leider keinerlei Infos über die gedrückte Schlosstaste , so dass unterschiedliche gewünschte Funktionen nur über unterschiedliche Codes realisiert werden können.

In der Software wird beim Import ein Objekt vom Typ Zahl angelegt, das bei Empfang vom Keypad die empfangene Benutzernummer zugewiesen bekommt. 1 bis 8 sind gültige Werte, bei 32 wurde ein falscher Code eingegeben.

### 1.23.3.16. HmIP-RGBW

---

#### HmIP-RGBW

Der Betriebsmodus des Geräts muss in der WEB-UI im Modul unter "Einstellen" im Abschnitt "Modus" aus einer Auswahlbox ausgewählt eingestellt werden. Der Modus ist aber leider nicht wie üblich in den beim Import übertragenen Daten enthalten. Daher muss wenn nicht der RGBW- bzw. RGB-Modus benutzt wird, der Modus "Tunable White" bzw. "Einzelkanäle" durch manuelles Anlegen des Moduls eingestellt werden.

Dazu wird das durch den Import generierte Gerät HmIP-RGBW gelöscht und eines der in der Geräteliste aufgeführten Module HmIP-RGBW-TW für "Tunable White" bzw. HmIP-RGBW-DIM für vier einzelne Dimmer (also Auswahl "Einzelkanäle) aus der Modulliste angelegt.

Nach dem Anlegen muss die Adresse des Moduls noch manuell eingetragen werden. Die Kanalnummer ist schon vorbelegt, die Platzhalter xxx im Adressfeld müssen durch die Adresse des Geräts ersetzt werden.

### 1.23.3.17. HmIPW-WRC6

---

#### HmIPW-WRC6

Dieses Gerät hat für jede der 6 Tasten eine LED, die mit verschiedenen Optionen gesteuert werden kann.

Die Objekte für die LEDs haben den Objekttyp *Dimmer* und können mit Helligkeiten zwischen 0 (=aus) und 100 (=volle Helligkeit) gesteuert werden. Die erste LED ist dem ersten Taster zugeordnet, die 6. LED dem 6. Taster, das letzte LED-Objekt wird zur Steuerung aller LEDs gemeinsam benutzt. Die Optionen zur Darstellung werden über Variablen gesteuert, dazu werden folgende Variablen benutzt:

**Farbe:** diese Variable kann die Zahlen 0 bis 7 zugewiesen bekommen, die Zahlen haben folgende Bedeutung:

0 = aus (bei 0 bleibt die LED dunkel, es muss also immer ein Wert zugewiesen werden damit die LED leuchtet.)

1 = Blau

2 = Grün

3 = Türkis

4 = Rot

5 = Violet

6 = Gelb

7 = Weiss

8 = alter wert

**Modus:** diese Variable bestimmt das Verhalten der LED und kann die Zahlen 0 bis 11 zugewiesen bekommen. Die Zahlen haben folgende Bedeutung:

0 = aus (bei 0 bleibt die LED dunkel, es muss also immer ein Wert zugewiesen werden damit die LED leuchtet.)

1 = ununterbrochen an

2 = langsames Blinken

3 = normales Blinken

4 = schnelles Blinken

5 = langsames Blitzen

6 = mittelmässiges Blitzen

7 = schnelles Blitzen

8 = langsames auf/abdimmen

9 = mittleres auf/abdimmen

10 = schnelles auf/abdimmen

11 = alter Wert

**Dauer:** Ein Wert zwischen 0 und 16343, der die Dauer in Sekunden angibt, für die die LED eingeschaltet bleiben soll, nach Ablauf der Dauer wird die LED ausgeschaltet indem die Helligkeit auf 0 gesetzt wird. Die Dauer muss als letzter Parameter zugewiesen werden.

#### Beispiel von Anweisungen in einem Makro:

```
Taste2LED:=70          // Helligkeit auf 70%
Taste2LED.Farbe:=4      // Farbe auf rot setzen
Taste2LED.Modus:=3      // normales Blinken
warte 1 Sekunde         // damit die gesetzten Werte sofort und damit auf jeden
Fall vor der Dauer gesendet werden
Taste2LED.Dauer:=4      // nach 4 Sekunden Helligkeit auf 0 setzen
```

## 1.23.3.18. HmIP-Wired

---

[< Vorherige Seite](#)

[Nächste Seite >](#)

### HmIP-Wired

Bei den HmIP-Wired-Geräten ist zu beachten, dass hier im Gegensatz zu den HM-Wired-Geräten bei den meisten Tastern/Schaltern eine Registrierung der einzelnen Kanäle (wie bei HM-Funk-Modulen) erforderlich ist.

[< Vorherige Seite](#)

[Nächste Seite >](#)

## 1.23.4. Shelly-System

---

### Shelly-System

Die Shelly-Geräte kommunizieren über WLAN direkt mit der Zentrale, ohne dass ein zusätzliches Gateway erforderlich ist.

Hersteller der auf dem bekannten ESP8266-Prozessor basierenden Geräte ist Allterco Robotics aus Bulgarien.

Erhältlich sind die Geräte in vielen deutschen Online-Shops wie <https://www.technikhaus.de>

Zur Konfiguration stellen die Geräte jeweils ein eigenes WLAN zu Verfügung und sind über die jeweils in der Anleitung angegebene IP (normalerweise 192.168.33.1) erreichbar.

Bei der Konfiguration über einen Browser kann dann die Integration in das eigene lokale WLAN vorgenommen werden.

Momentan werden folgende Shelly-Geräte unterstützt:

**Shelly1** und **Shelly1PM** inkl. der verschiedenen Addons zu den Geräten

**Shelly2.5** als 2-fach-Aktor und als Rollladenaktor

**ShellyPlug** - Schaltsteckdose

**ShellyDuo** - Lampe

**ShellyVintage** - Vintage-Glühbirne

**ShellyDimmer** - Dimmer UP

**ShellyDW2** - Tür/Fensterkontakt

**Shellyi3** - Eingabe f. Taster, Schalter, Kontakte

**ShellyHT** - Temperatur/Luftfeuchtigkeitssensor

**ShellyMotion** - Bewegungsmelder

**ShellyFlood** - Wassermelder

**ShellyButton1** - Button

## 1.23.4.1. Allgemeine Gerätekonfiguration

---

### Shelly Gerätekonfiguration

Das zu installierende Shelly-Gerät wird in der linken Listbox im Fenster Modulauswahl ausgewählt und mit dem Roten-Pfeil-Button in die Listbox der verwendeten Geräte übertragen.

Shelly-Aktoren werden über ihre IP-Adresse angesprochen, diese Eingabe wird für den jeweiligen Aktor auf der Shelly-Konfigurationsseite im Browser aktiviert (5) und auf der der Geräteseite der CL-Software eingegeben (1). Es sollte eine feste IP-Adresse verwendet werden, damit diese nicht über DHCP durch den Router geändert wird, was zur Folge hätte, dass das Gerät nicht mehr gesteuert werden kann.

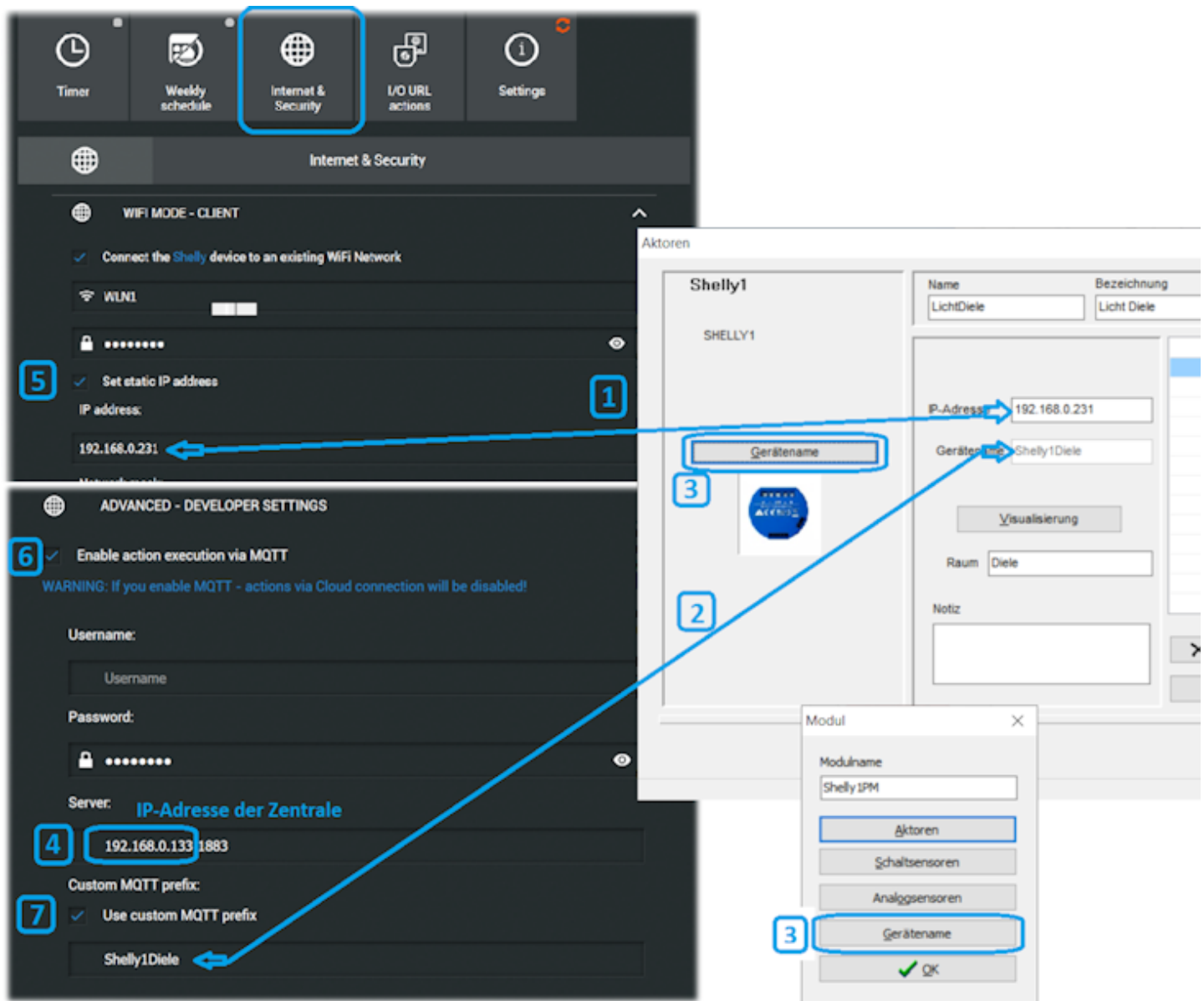
Die Werte und Ereignisse von Shelly-Sensoren werden an den MQTT-Server der CL-Software geschickt, der diese aufbereitet und an die ExecEngine weiterleitet. Im Shelly-Gerät werden die MQTT-Einstellungen (6) und die Eingabe eines Gerätenamens aktiviert (7), es darf kein Benutzername (Username) und Kennwort (Password) vergeben werden. In der Konfigurationsseite des Browsers wird ein Geräteiname (2) festgelegt, der in der CL-Software auf der Geräteseite der Sensoren angegeben wird (2). Der Geräteiname entspricht dabei der Geräte-Adresse in anderen Systemen.

Zur Eingabe des Gerätenamens wird der Knopf "Geräteiname" (3) angeklickt, dieser wird dann in alle Geräte eines Moduls eingesetzt. Je nach Gerät kann der Knopf zur Eingabe des Gerätenamens an verschiedenen Stellen sein (3). Die IP-Adresse der Zentrale wird unter "Advanced-Developer Settings" im Eingabefeld "Server" der Shelly-Konfigurationsseite im Browser angegeben (4). Der Port, der hinter der IP-Adresse angegeben wird, ist immer 1883.

**Nach den Änderungen der Einstellungen des Shelly-Geräts muss dieses neu gestartet werden damit die neuen Einstellungen benutzt werden!**

**Nach dem Start der Zentrale bzw. der ExecEngine kann es je nach Geräten eventuell mehrere Minuten dauern bis diese sich am MQTT-Server angemeldet haben und Daten übermitteln.**





Beim Anlegen der Geräte werden jeweils die passenden Objekte automatisch generiert.  
Bei Sensoreingängen muss der Objekttyp entsprechend der im Shelly-Gerät eingestellten Funktion (Taster/Schalter/Kontakt) geändert werden.

## 1.23.4.2. Shelly1(PM) mit Addons

---

### Shelly1 und Shelly1PM mit Addons

Diese beiden Aktoren haben eine Schnittstelle, an die externe Sensoren angeschlossen werden können. Daher gibt es bei der in der Liste der zur Verfügung stehenden Geräte für diese Aktoren unterschiedliche erweiterte Gerätebezeichnungen..Dabei wird an die normale Gerätebezeichnung ein Zusatz angehängt, aus dem die jeweils verwendeten Sensoren hervorgehen.

Es gibt folgende Zusätze:

**Add-ExtSwitch** für einen externen Kontakt oder Taster

**Add-T3** für bis zu 3 Temperatursensoren

**Add-TH** für einen Temperatur/Luftfeuchtigkeitssensor

Je nach ausgewähltem Gerät werden bei der Erstellung des Geräts unterschiedliche Objekte für das Gerät erstellt.

### 1.23.4.3. Shelly2.5 als Rollladenaktor

---

#### Shelly2.5

Dieser Aktor beinhaltet zwei Relais, die für zwei unterschiedliche Geräte verwendet werden können. Alternativ kann dieser Aktor auch als Rollladenaktor konfiguriert werden.

Es gibt insgesamt drei unterschiedliche Auswahlmöglichkeiten in der Liste der Geräte.

**Shelly2** bei Verwendung der Ausgänge für zwei unterschiedliche Verbraucher.

**ShellyRollOhnePos** bei Verwendung als Rollladenaktor ohne Kalibrierung.

Dabei wird ein Objekt mit den Zuständen oben, stop, unten erstellt.

Beim Zustand stop wird in der Visualisierung eine halb geschlossene Rolllade dargestellt. Damit die Software ermitteln kann ob die Rolllade komplett geöffnet bzw. geschlossen ist oder in einer Zwischenstellung steht, muss die Gesamtlaufzeit der Rolllade im Konfigurationsfenster des Aktors in der CL-Software angegeben werden.

**ShellyRollMitPos** bei Verwendung als Rollladenaktor mit Kalibrierung.

Diese Option kann nur verwendet werden nachdem die Kalibrierungsfunktion des Aktors erfolgreich durchgeführt wurde, sonst kann der Aktor nicht über die Prozentzahlen zur Angabe der Position gesteuert werden.

Nachdem die Kalibrierung durchgeführt wurde wird auf der Webseite des Aktors ein Schieber zur Positionssteuerung angezeigt und der Aktor kann über einen Zahlenwert für die prozentuale Öffnung gesteuert werden und meldet seine aktuelle Stellung als Prozentzahl.

## 1.23.5. Spezielle Gateways

---

### Spezielle Gateways

Hier werden Gateways beschrieben, mit denen andere als HomeMatic- oder HomeMatic-IP-Geräte verwendet werden können.

[Philips Hue-Bridge](#)

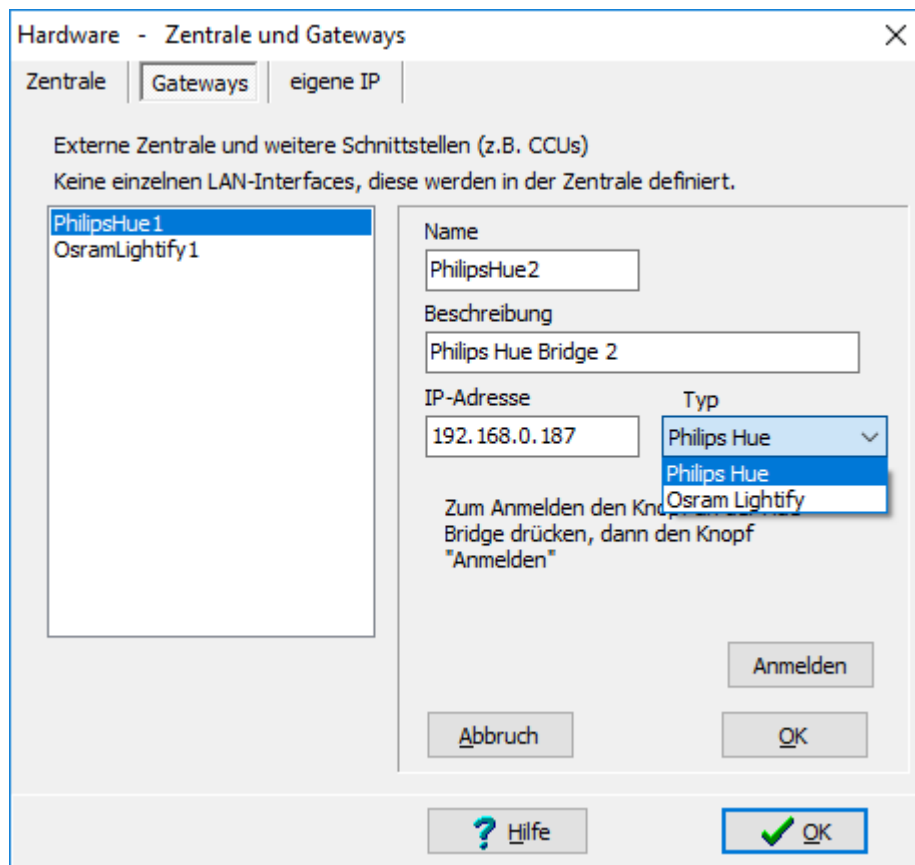
[Osram Lightify-Gateway](#)

## 1.23.5.1. Philips Hue Bridge

### Philips Hue Bridge

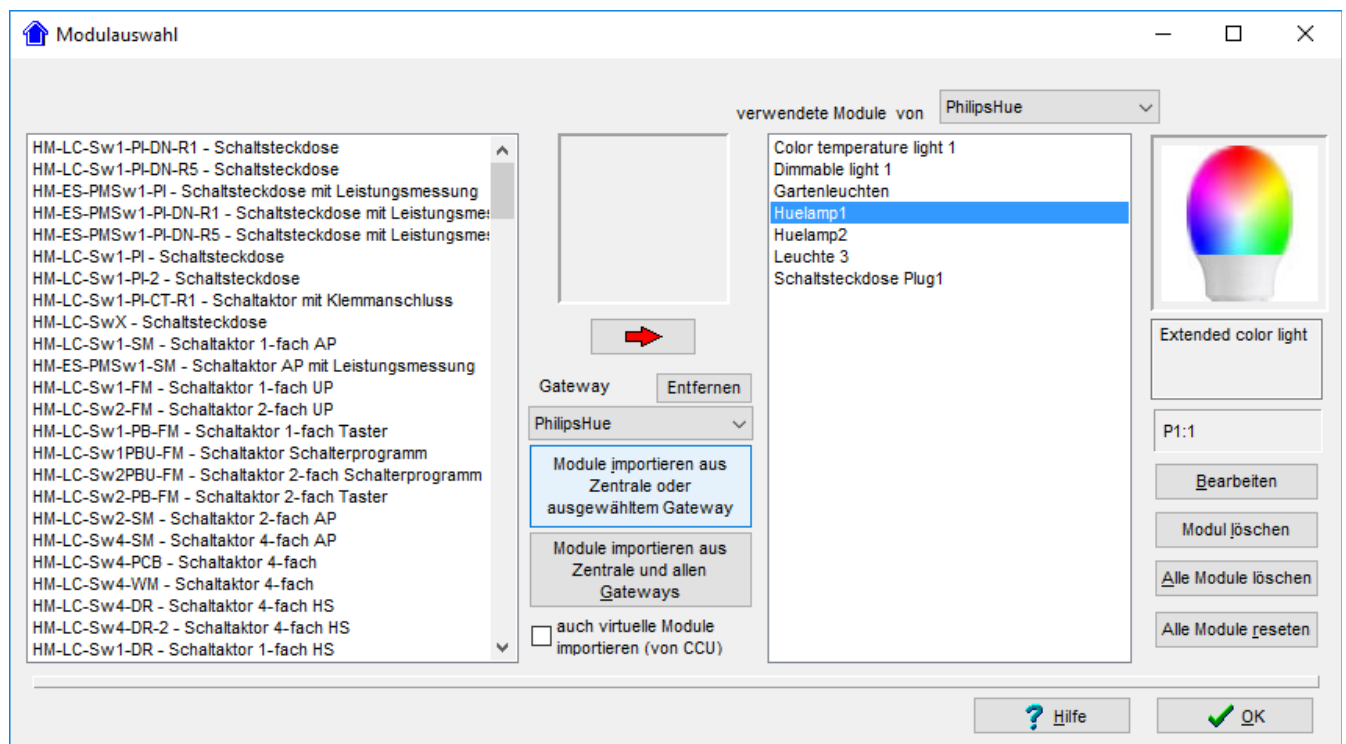
Um eine Philips Hue-Bridge verwenden zu können muss diese Im Reiter *Gateways* des Hardwarefensters angelegt werden. Klicken Sie dazu auf den Button *Neu* im Reiter *Gateways* des Hardwarefensters.

Dieses Programm ist kein Konfigurationsprogramm für die Philips Hue-Bridge, das Hinzufügen von Geräte und Konfigurationseinstellungen wird mit den Apps des Herstellers vorgenommen. Die Geräte der Bridge können über die Importfunktion im Modulfenster importiert werden.



Als erstes muss die IP-Adresse der Bridge ermittelt und im Feld IP-Adresse eingetragen werden. Welche IP-Adresse die Bridge hat kann im Router nachgesehen werden. Um auf die Bridge zugreifen zu können muss als nächstes ein Zugriffsschlüssel angefordert werden. Drücken Sie dazu den Knopf auf der Bridge und danach den Knopf *Anmelden* im Fenster. Es muss dann eine Meldung erscheinen, in der die erfolgreiche Anmeldung bestätigt wird.

Wenn die Bridge schon angemeldet ist erscheint im Panel ein Text "Es existiert schon eine Anmeldung ..."



Die an der Bridge angelernten Geräte werden im Modulfenster importiert. Wählen Sie dazu im Auswahlfenster *Gateway* die Bridge aus und klicken auf den Button *Module importieren aus Zentrale oder ausgewähltem Gateway*.

Im Listfenster der verwendeten Module erscheinen nun die an der Bridge angelernten Geräte. Immer wenn Änderungen an der Bridge vorgenommen werden muss ein neuer Import durchgeführt werden. Dabei müssen Geräte in der Software gelöscht werden wenn Sie von der Bridge abgelernt wurden, auch wenn danach wieder angelernt wurden. Das ist notwendig weil sich in so einem Fall die Adresse des Geräts ändern kann.

Für Geräteeigenschaften wie Farbe, Helligkeit, Lichttemperatur usw. werden Variablen generiert, die ausgelesen und gesetzt werden können. Welche Variablen generiert werden ist vom jeweiligen Gerät abhängig.

Folgende Gerätetypen der Philips Hue-Bridge werden unterstützt (ZigBee-Typbezeichnung):

*Extended color light*

*Color temperature light*

*Dimmable light*

*On/Off plug-in unit*

Es gibt folgende möglichen Variablen und Wertebereiche:

Bright = Helligkeit 0 bis 255

Hue = Farbe 0 bis 65535 gemäß des RGB-Farbkreises

Sat = Farbsättigung : 0 bis 255

Temp = Farbtemperatur 2000 bis 6500 Kelvin

Modus : 0 und 1, 0 = Farbmodus, 1 = Farbtemperaturmodus

Beim Import werden die Geräte entsprechend ihres Typs generiert, je nach Eigenschaften der Geräte werden auch Variablen zum Gerät angelegt, Anzahl und Typ der Variablen ist vom Gerät anhängig. Der Gerätezustand bzw. die Variablenwerte können über Makros verändert werden, geänderte Werte von Variablen werden an die Geräte übertragen.

Nicht in allen Visualisierungsoptionen stehen Objekte zur Verfügung, die die direkte Steuerung der ZigBee-Geräte erlauben. In dem Fall kann das Setzen der Variablen und somit die Steuerung des Geräts über virtuelle Objekte erfolgen.

Weitere Infos dazu sind hier beschrieben:

[Variablen über virtuelle Objekte setzen](#)

Wenn der Farbmodus verwendet wird ist die Farbtemperatur deaktiviert, wenn die Farbtemperatur verwendet wird kann nicht gleichzeitig eine Farbe eingestellt werden.

Die tatsächliche Farbe am Gerät entspricht der in Visualisierungen dargestellten Farbe nur näherungsweise, die Unterschiede in der Darstellung sind insbesondere abhängig vom verwendeten Leuchtmittel und Hersteller.



## 1.23.5.2. Osram Lightify Gateway

### Osram Lightify Gateway

Um ein Osram Lightify-Gateway verwenden zu können muss diese Im Reiter *Gateways* des Hardwarefensters angelegt werden. Klicken Sie dazu auf den Button *Neu* im Reiter *Gateways* des Hardwarefensters.

Dieses Programm ist kein Konfigurationsprogramm für das Osram Lightify-Gateway, das Hinzufügen von Geräte und Konfigurationseinstellungen des Gateways muss mit den Apps des Herstellers vorgenommen werden. Die Geräte des Gateways können über die Importfunktion im Modulfenster importiert werden.

Hardware - Zentrale und Gateways

Zentrale | Gateways | eigene IP

Externe Zentrale und weitere Schnittstellen (z.B. CCUs)  
Keine einzelnen LAN-Interfaces, diese werden in der Zentrale definiert.

PhilipsHue  
**OsramLightify**  
PilipsHue2

Name  
OsramLightify

Beschreibung  
Osram Lightify

IP-Adresse  
192.168.0.112

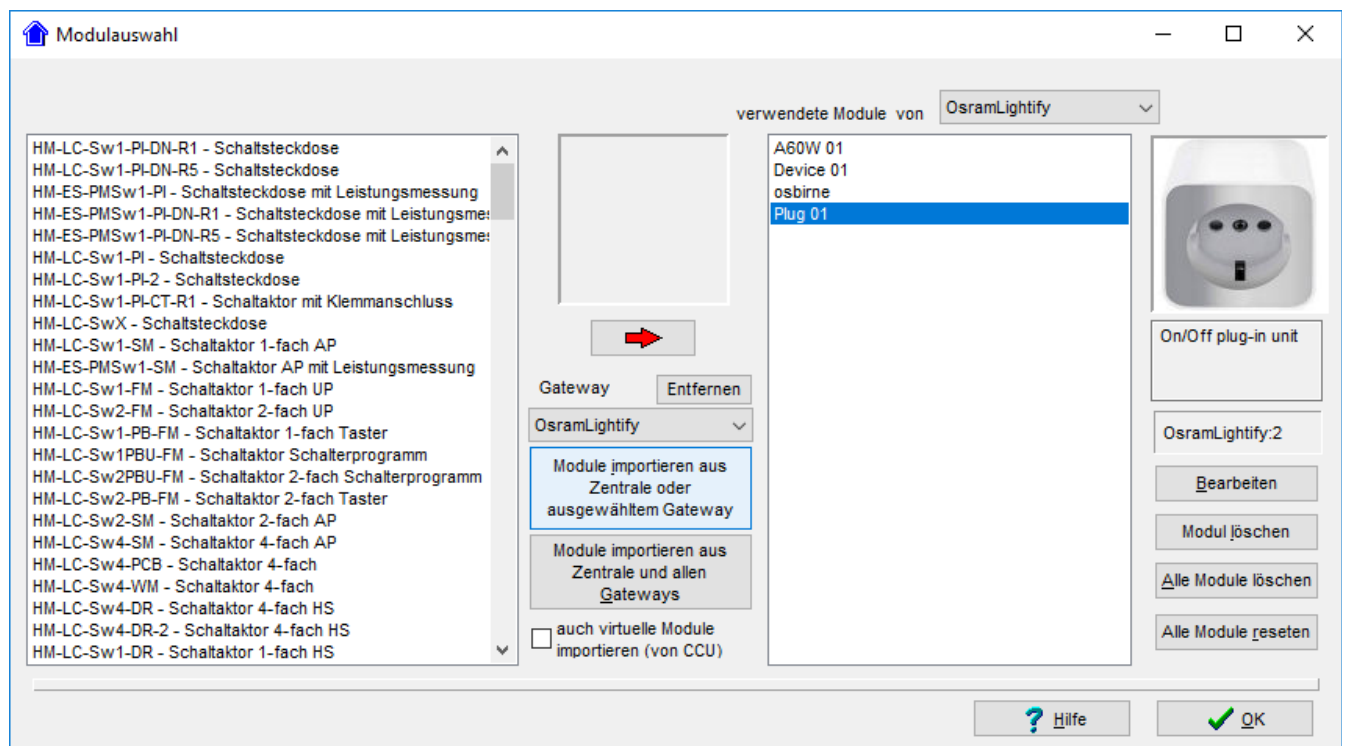
Typ  
Osram Lightify  
Philips Hue  
Osram Lightify

Abbruch OK

? Hilfe OK

Als erstes muss die IP-Adresse des Gateways ermittelt und im Feld IP-Adresse eingetragen werden. Welche IP-Adresse das Gateway hat kann im Router nachgesehen werden.





Die am Gateway angelernten Geräte werden im Modulfenster importiert. Wählen Sie dazu im Auswahlfenster *Gateway* das *Osram Lightify Gateway* aus und klicken auf den Button *Module importieren aus Zentrale oder ausgewähltem Gateway*.

Im Listfenster der verwendeten Module erscheinen nun die am Gateway angelernten Geräte. Immer wenn Änderungen an der Bridge vorgenommen werden muss ein neuer Import durchgeführt werden. Für Geräteeigenschaften wie Farbe, Helligkeit, Lichttemperatur usw. werden Variablen generiert, die ausgelesen und gesetzt werden können. Welche Variablen generiert werden ist vom jeweiligen Gerät abhängig.

Folgende Gerätetypen des Osram Lightify-Systems werden unterstützt (ZigBee-Typbezeichnung):

*Extended color light*  
*Color temperature light*  
*Dimmable light*  
*On/Off plug-in unit*

Beim Import werden die Geräte entsprechend ihres Typs generiert, je nach Eigenschaften der Geräte werden auch Variablen zum Gerät angelegt, Anzahl und Typ der Variablen ist vom Gerät abhängig. Der Gerätezustand bzw. die Variablenwerte können über Makros verändert werden, geänderte Werte von Variablen werden an die Geräte übertragen.

Es gibt folgende möglichen Variablen und Wertebereiche:

Bright = Helligkeit 0 bis 100  
 Hue = Farbe 0 bis 65535 gemäss des RGB-Farbkreises  
 Temp = Farbtemperatur 2000 bis 6500 Kelvin  
 Modus : 0 und 1, 0 = Farbmodus, 1 = Farbtemperaturmodus

Nicht in allen Visualisierungsoptionen stehen Objekte zur Verfügung, die die direkte Steuerung der ZigBee-Geräte erlauben. In dem Fall kann das Setzen der Variablen und somit die Steuerung des Geräts über virtuelle Objekte erfolgen.

Weitere Infos dazu sind hier beschrieben: [Variablen über virtuelle Objekte setzen](#)

Wenn der Farbmodus verwendet wird ist die Farbtemperatur deaktiviert, wenn die Farbtemperatur verwendet wird kann nicht gleichzeitig eine Farbe eingestellt werden.

Die tatsächliche Farbe am Gerät entspricht der in Visualisierungen dargestellten Farbe nur näherungsweise, die Unterschiede in der Darstellung sind insbesondere abhängig vom verwendeten Leuchtmittel und Hersteller.

## 1.23.6. FHZ-System

---

### FHZ-System

**Es werden nicht alle verfügbaren FS20-Module im linken Listfenster der Modulauswahl aufgeführt.**

Dort sind nur die am häufigsten verwendeten Module und spezielle Module vorhanden. Falls FS20-Module verwendet werden, die nicht aufgeführt sind, müssen dafür die Standard-Module "FS20 Empfänger" und "FS20 Sender" benutzt werden. Diese Standard-Module stehen mit unterschiedlicher Kanalanzahl zur Verfügung. Es sollten jeweils die Standard-Module ausgewählt werden, die die gleiche Anzahl oder mehr Kanäle wie das verwendete Modul haben. Es ist auch möglich für ein Hardware-Modul mehrere Standard-Module der Liste zu verwenden, es muss dann allerdings darauf geachtet werden, dass einzelne Kanäle nicht doppelt angelegt werden.

Alle Module, die nicht zum HomeMatic-System gehören können nicht aus einer Zentrale importiert werden, sondern werden im linken Listfenster der Modulauswahl ausgewählt und mit dem Pfeil-Button in das Listfenster der verwendeten Module kopiert.

Weitere Informationen dazu finden Sie im Kapitel [Modulauswahl](#).

**Der folgende Abschnitt gilt nur für FHZ2000 und FHZ1xxxPC mit PC als Zentrale ( nicht für FHZ13xxxPC mit CCU1 als Zentrale):**

Wenn eine Anweisung zum Einschalten für eine Zeitspanne für einen FS20-Aktor ausgeführt wird (z.B. AktorXX einschalten für 3 Minuten), wird auch die Zeitdauer an den Aktor gesendet, so dass er nach Ablauf dieser Dauer ausschaltet, auch wenn er keine Funkmeldung mit einem Ausschaltbefehl erhält.

Allerdings ist die maximal Zeitspanne für diese automatische Abschaltfunktion möglich ist 4 Stunden, da grössere Zeiten im FS20-Protokoll nicht möglich sind. Da die Zeitdauer in nur einem Byte kodiert wird ist diese nicht sekundengenau, es wird jeweils die nächstmögliche grössere Zeit hinter der Einschaltzeit benutzt - je grösser die Zeitspanne ist, desto höher ist die Differenz zur angegebenen Zeitspanne.

Durch diese Funktion soll sichergestellt werden, dass ein Aktor wieder ausschaltet, auch wenn der Ausschaltbefehl der Zentrale nicht empfangen wird. Im Normalfall wird der Aktor exakt nach Ablauf der Zeitspanne durch die Zentrale ausgeschaltet, nur wenn dieser Funkbefehl nicht empfangen wird schaltet der Aktor nach der übermittelten Zeitspanne aus, wenn diese nicht grösser als 4 Stunden ist.

Für FS20-Module gibt es eine spezielle Variante der Anweisung SETZEWERT, damit ist es möglich FS20-Aktoren direkt anzusprechen und FS20-1-Byte und FS20-2-Byte Befehle an FS20-Aktoren zu senden (**nicht mit FHZ1xxxPC und CCU1 als Zentrale**).

Beispiel zum Einschalten eines Schalt-Aktors:

```
SETZEWERT(EinFS20Aktor, "FS20", 1)
```

und zum Ausschalten:

```
SETZEWERT(EinFS20Aktor, "FS20", 0)
```

**Bitte beachten:**

Der Zustand des Objekts in der Software ändert sich durch eine Anweisung SETZEWERT nicht, daher sollte die Anweisung SENDEWERT nur für spezielle Funktionen benutzt werden, in denen die Steuerung von Aktoren über Zuweisungen möglich ist oder nicht ausreicht.

Hier eine Liste der FS20-Codes:

0 aus

1 an, 6,25% Einschalten auf Helligkeitsstufe 1 (min.)

2 an, 12,5%

...

15 an, 93,75%

16 an, 100% Einschalten auf Helligkeitsstufe 16 (max.)

17 an, alter Wert Auf letztem Helligkeitswert einschalten

18 toggle Wechsel zwischen "aus" und "an, alter Wert"

19 dim up Eine Helligkeitsstufe heller

20 dim down Eine Helligkeitsstufe dunkler

21 dim up and down ..,+ bis max, kurz warten,- bis min, kurz warten,..

## 1.23.6.1. Anmelden von FHT80b-Raumreglern

---

### Anmelden von FHT80b-Raumreglern

Damit die Raumregler mit einer FHZ-Schnittstelle kommunizieren können, müssen diese an der Schnittstelle angemeldet werden.

Das geschieht wie im Folgenden beschrieben:

In der verwendeten Schnittstelle muss der FHT-Zentralencode angelegt werden.

Bei den FHZ2000-Schnittstelle geschieht das über den WEB-Server der Schnittstelle, bei den FHZ1xxx-Schnittstellen wird diese Konfiguration in der Software unter dem Menüpunkt

*Konfigurieren->Hardware* auf der Unterseite *FHZ* vorgenommen.

Der Zentralencode muss zwischen 1 und 99 liegen.

Wenn als Zentralencode für die FHTs (auch mit FHT-Code bezeichnet) die Zahlen 0 oder 100 stehen wurde noch kein Zentralencode vergeben. Diesen müssen Sie dann vergeben (Zahl zwischen 1 und 99) bevor FHTs angemeldet werden.

**Falls mehrere FHZ verwendet werden beachten Sie bitte unbedingt folgende Hinweise: Alle verwendeten FHZs müssen unterschiedliche FHT-Zentralencodes haben !**

**Beim Anmeldevorgang darf jeweils nur die FHZ aktiv sein, an die der FHT angemeldet werden soll, alle anderen müssen stromlos sein !**

Der Raumregler FHT80b muss im Fenster *Modulauswahl* ausgewählt und angelegt werden.

Als Adresse wird der Code des Raumreglers eingegeben.

Den Code des Raumreglers können Sie am Raumregler wie folgt eingeben bzw. nachsehen:

- Taste PROG gedrückt halten bis *Sond* in der LCD-Anzeige erscheint.
- Das Stellrad drehen, bis *Code* in der Anzeige erscheint und die Taste PROG drücken.
- Es erscheint der Hinweis *Code 1* und eine Zahl zwischen 00 und 99 (die führende 0 wird ignoriert), die durch Drehen am Stellrad verändert werden kann.
- Diese Zahl bildet die beiden vorderen Ziffern der Adresse. **Beispiel: 021 = 21**
- Drücken Sie die Taste PROG noch einmal.
- Es erscheint der Hinweis *Code 2* und eine Zahl zwischen 00 und 99 (die führende 0 wird ignoriert), die durch Drehen am Stellrad verändert werden kann.
- Diese Zahl bildet die beiden hinteren Ziffern der Adresse. **Beispiel: 043 = 43**
- Verlassen Sie den Sondermodus des Raumreglers durch drücken der PROG-Taste, der Raumregler führt nun eine Synchronisation mit den Stellantrieben durch.

Geben Sie die Zahl im Adressfeld des Fensters *FS20 Empfänger* ein. **In diesem Beispiel 2143** Starten Sie die Ausführung des erstellten Programms.

Der Raumregler muss an einer FHZ angemeldet werden, da die Kommunikation mit den FHTs bidirektional ist und ein FHT jeweils immer nur mit einer Zentrale kommunizieren kann. Bei diesem Anlernprozess darf nur eine FHZ im Funkbereich aktiv sein, damit sichergestellt ist, dass die Anmeldung an dieser FHZ erfolgt.

Diese Anmeldung geschieht wie folgt:

- Taste PROG gedrückt halten bis *Sond* in der LCD-Anzeige erscheint.
- Das Stellrad drehen, bis *CEnt* in der Anzeige erscheint und die Taste PROG drücken.

- Das Stellrad drehen bis *nA* in der Anzeige erscheint, dann die Taste PROG drücken
- Nun in der Visualisierung der Ausführung für den Raumregler eine andere Solltemperatur einstellen.

Damit wird innerhalb der nächsten Minuten eine Verbindung zum Raumregler hergestellt. Im Rahmen dieser Verbindung wird dann auch der FHT-Code zum Raumregler hergestellt.

Bitte beachten Sie, dass der Ausführungsmodus aktiv sein muss, der Raumregler mit seiner korrekten Adresse im Programm angelegt sein muss und die Solltemperatur des Raumreglers in der Visualisierung verändert werden muss. Diese Veränderung wird nach 2-3 Minuten am Raumregler angezeigt. Dann sollte auch die Anmeldung erfolgt sein und einige Minuten nach der erfolgreichen Anmeldung muss die Visualisierung des Raumreglers am Bildschirm die Ist-Temperatur des Raumreglers anzeigen.

Geschieht das nicht überprüfen Sie bitte Folgendes:

Steht die Anzeige im Punkt *CEnt* auf *On* ?

Wenn ja ist die Anmeldung erfolgreich durchgeführt worden.

Wenn nicht überprüfen Sie bitte Folgendes:

Ist der Ausführungsmodus des PC-Programms aktiviert und wurde die Solltemperatur in der Visualisierung verändert ?

Ist ein Zentralencode zwischen 1 und 99 in der FHZ eingestellt ?

Stimmt die Adresse im PC-Programm mit der Adresse ( Code ) des Reglers überein ?

Steht unter dem Punkt *CEnt* im Raumregler der Wert *nA* ?

## 1.23.6.2. FS20-Rollladensteuerungen

---

### FS20-Rollladensteuerungen

Die Parameter zur Rollladensteuerung und Markisensteuerung werden im linken unteren Teil des Konfigurationsfensters angegeben.

Die Gesamtlaufzeit von unten nach oben (bei Markisen ist damit der Verstellvorgang von komplett ausgefahren bis eingefahren gemeint) wird im obersten Feld angegeben. Wichtig ist, dass die längste Laufzeit angegeben wird – das ist normalerweise immer die Laufzeit von unten nach oben. Zum jeweiligen Zustand der eingestellt werden soll, muss die Laufzeit angegeben werden, die benötigt wird diesen Zustand von der obersten Stellung her (Rolllade komplett geöffnet, Markise komplett eingefahren) zu erreichen.

**Bitte beachten Sie:**

*Bei einem Verstellvorgang, bei dem eine Zwischenstellung eingestellt werden soll, wird immer erst der obere Zustand (Rolllade komplett geöffnet, Markise komplett eingefahren) eingestellt. Dazu wird die Rolllade/Markise für die Gesamtlaufzeit heraufgefahren.*

*Erst nach der Gesamtlaufzeit wird in die gewünschte Zwischenstellung gefahren.*

*Dieses Verfahren zur Justierung der Rolllade/Markise ist nötig um die Zwischenstellung sicher einstellen zu können, ansonsten könnte die gewünschte Zwischenstellung aufgrund manueller Verstellungen oder mehrmaliger Verstellvorgänge nicht richtig eingestellt werden.*

*Durch die Auswahl eines anderen Bild-Symbols kann sich auch der Typ des Objekts ändern. Es liegt in der Verantwortung des Benutzers einen jeweils zum Hardwaremodul passenden Typ auszuwählen.*

### 1.23.6.3. FS20-Zustandsmelder

---

#### FS20-Zustandsmelder

Bitte beachten Sie, dass bei FS20-Zustandsmeldern normalerweise nur eine Meldung kommt, wenn der besondere Zustand dieses Melders erreicht ist, aber keine Meldung wenn dieser Zustand wieder in den Normalzustand wechselt.

Ein Regensensor sendet z.B. nur eine Meldung wenn es beginnt zu regnen, nicht aber wenn es aufhört. Ein Bewegungsmelder sendet nur eine Meldung in dem Moment wo er eine Bewegung erkennt.

Eine eventuell in dem Zustandsmelder einzustellende Einschaltzeit wird nicht berücksichtigt, da die Funktionen von der Programmierung bestimmt werden sollen.

Auch das Zurücksetzen des Sensorzustands auf den Normalwert muss vom Programm vorgenommen werden, damit die Visualisierung korrekt ist.

Am sinnvollsten geschieht das in dem Makro des Zustandsmelders. Dabei kann die [Anweisung WARTEN](#) benutzt werden, damit der Zustand einige Zeit am Bildschirm angezeigt wird.

#### **Beispiel bei einem Bewegungsmelder:**

*Anweisungen....*

warte 10 Sekunden

Bewegungsmelder ausschalten

## 1.23.7. CL-control

---

### CL-control

In diesem Abschnitt finden Sie Infos zu den Geräten von CL-control.

## 1.23.7.1. WLAN-Schaltsteckdose CL-SDAx

---

### WLAN-Schaltsteckdose CL-SDAx

Diese Schaltsteckdose kann als separates Gerät betrieben werden oder über die CL-Software von CL-control in ein Hausautomationssystem eingebunden werden.

#### Inbetriebnahme

Als erstes muss die WLAN-Steckdose in das eigene WLAN eingebunden werden. Um diese Konfiguration vornehmen zu können stellt die WLAN-Steckdose ein eigenes WLAN zu Verfügung. Stecken Sie die Schaltsteckdose in eine Steckdose, die blaue LED auf der Vorderseite blinkt kurz auf. Solange Sie noch keine Verbindung zu Ihrem lokalen Netzwerk konfiguriert haben fungiert die Schaltsteckdose als Accesspoint mit eigenem WLAN-Netzwerk. In diesem Modus blinkt die blaue LED auf der Vorderseite nach ca. 10 Sekunden in Abständen von 2 Sekunden kurz auf.

Der Name des Accesspoints der Schaltsteckdose beginnt mit „ESP\_“ gefolgt von mehreren Ziffern. Stellen Sie eine Verbindung zu diesem Netzwerk her.

Sollte es keinen mit ESP\_ ..beginnenden Netzwerknamen in der Auswahl verfügbarer Netzwerke geben, suchen Sie bitte nach HomeMate\_AP und wählen dieses Netzwerk aus.

Stellen Sie eine Verbindung zu diesem Netzwerk her.

Geben Sie nun in der Adresszeile des Browsers die IP-Adresse **192.168.4.1** ein und drücken auf Eingabe.

Es erscheint die Startseite des Webservers der Schaltsteckdose. Es wird noch keine richtige Uhrzeit angezeigt, da noch jetzt noch keine Verbindung zu einem NTP-Server im Internet besteht und auch noch keine Uhrzeit eingegeben wurde.

Nun wird die Schaltsteckdose in das eigene lokale Netzwerk eingebunden. Klicken Sie dazu auf *Netzwerk* und im nächsten Fenster auf die Auswahlbox *Netzwerke*.

Es werden nun alle Netzwerke in Reichweite angezeigt. Je mehr Sternchen hinter einem Netzwerk stehen, desto besser ist die Verbindung. Das Doppelkreuz am Ende einer Zeile bedeutet, dass es sich um ein Netzwerk mit Verschlüsselung handelt. Wählen Sie nun Ihr Netzwerk aus und klicken auf den Knopf *Netzwerk auswählen*. Es erscheint ein Eingabefeld für den Netzwerkschlüssel. Geben Sie diesen ein und klicken den Knopf *Speichern*.

Stellen Sie dann die Verbindung zu Ihrem lokalen Netzwerk her und sehen in Ihrem Router nach welche IP-Adresse der Schaltsteckdose zugewiesen wurde. Der Netzwerkname der Schaltsteckdose beginnt mit *ESP-...* Die folgenden Buchstaben und Ziffern sind identisch mit denen im Namen des Anfangs benutzten Netzwerks, wenn Sie noch keinen anderen Namen vergeben haben.

Die IP-Adresse der Schaltsteckdose im Netzwerk muss jetzt als Adresse des Geräts in der CL-Software eingegeben werden.

Wenn eine PIN für die Steckdose festgelegt wurde muss diese beginnend mit einem #-Zeichen hinter der IP-Adresse angegeben werden.

z.B.:

192.168.0.167#222





## 1.24. Allgemeine Hinweise

---

## 1.24.1. Aktoren

---

### Aktoren

**Aktoren** sind Schaltmodule (also Empfänger), an die Endgeräte (Rollläden, Beleuchtung und andere beliebige Geräte) angeschlossen werden.

## 1.24.2. Ausführungscode erstellen

---

### Ausführungscode erstellen

Der Compiler wird vom Konfigurationsprogramm mit dem Zahnrad-Knopf gestartet und erstellt den Ausführungscode für die [Exec-Engine](#), die auf der Zentrale ausgeführt wird.

## 1.24.3. Balkendarstellung

---

### Balkendarstellung

Die Balkendarstellung eignet sich für Objekte des Typs Zahl, also z.B. Temperatur, Luftfeuchtigkeit, Windgeschwindigkeit usw.

Die Balkendarstellung kann horizontal (Standard-Einstellung) oder vertikal erfolgen.

Bei vertikaler Darstellung wird die Bezeichnung und der Wert nur dann mit ausgegeben wenn er nicht breiter ist als die Balkendarstellung, ansonsten werden Wert und Bezeichnung nicht angezeigt.

## 1.24.4. Bild-Symbole

---

### Bild-Symbole

In diesem Fenster kann ein Bild-Symbol für das aktuelle Objekt ausgewählt werden.

Beim Öffnen des Fensters wird eine Auswahl von Bild-Symbolen angezeigt, die am ehesten auf das Objekt zutreffen. Mit der Schaltfläche **[Mehr anzeigen]** werden weitere im Programm zur Verfügung stehende Bild-Symbole angezeigt, unabhängig davon ob für das aktuelle Objekt zutreffen oder sinnvoll sind.

Mit der Schaltfläche **[Bild ändern]** können beliebige [Bitmaps](#) zur Darstellung des Objekts bzw. des Objektzustands eingebunden werden.

Solche Bitmaps können auch selbst erstellt werden. Das kann mit z.B. mit dem standardmässig in jeder Windows-Version vorhandenem Programm MS-Paint/Paintbrush geschehen (unter Zubehör).

Mit der Schaltfläche **[Neuer Darstellungstyp]** können neue Darstellungstypen erstellt werden.

Dazu wird die aktuell ausgewählte Objektdarstellung inklusive der Zustandsarten, aber ohne die zugehörigen Bilder kopiert. Als Bild wird für jeden Zustand ein Testbild eingesetzt. Dieses kann dann durch eigene Bitmaps ersetzt werden.

Ein solcher selbst definierter Darstellungstyp steht auch für andere Objekte des Projekts zur Verfügung.

Mit der Schaltfläche **[Darstellungstyp löschen]** können selbst erstellte Darstellungstypen wieder gelöscht werden, jedoch nicht die standardmässig im Programm angebotenen Darstellungstypen.

## 1.24.5. Bitmaps

---

### Bitmaps

Ein Bitmap ist eine Bild-Datei. Bitmaps können mit beliebigen Zeichen- und Malprogrammen erstellt werden. Ein solches Programm finden Sie normalerweise auch bei Windows im Menüpunkt Zubehör. Ein Bitmap kann als Hintergrundbild für Ansichten und auch als Bild zur Darstellung von Objektzuständen verwendet werden.

Als Bildformat sollte vorzugsweise PNG oder JPG benutzt werden, damit die Bilder auch in anderen Anwendungen und Apps benutzt werden können.

## 1.24.6. Dateiverwaltung auf der Zentrale

---

### Dateiverwaltung auf der Zentrale

Dateien, die von der CL-Software erzeugt werden, können über den Webserver der Software verwaltet werden.

Das gilt für die History-Datei, die Syslog-Datei und alle mit der Anweisung SCHREIBEDATEI erzeugten Dateien.

Bei der c-comatic wird die Seite zur Dateiverwaltung über den Konfigurations-Webserver aufgerufen.

Bei der CCU2 und CCU3 gibt es in der WEB-UI für die CL-Zusatzsoftware eine spezielle Seite um Infos zu sehen und Einstellungen vorzunehmen.

Diese Seite kann aufgerufen werden durch Anklicken des Buttons **[CL-Studio]** unter *Einstellungen->Systemsteuerung* oder unter *Einstellungen->Systemsteuerung->Zusatzsoftware* mit dem Button **[Einstellen]** im Rahmen der Zusatzsoftware.

Auf der Seite die dann erscheint, stehen einige Infos zur aktuellen Programmausführung. Im unteren Bereich gibt es den Button **[Dateien bearbeiten]** mit dem eine neue Seite zur Dateiverwaltung geöffnet wird.

Auf dieser Seite kann der Zugriff nur auf die vorgegebenen Verzeichnisse der jeweiligen Zentrale erfolgen. Die Verwaltung von anderen Verzeichnissen oder Unterverzeichnissen ist nicht möglich.



## 1.24.7. Dimmersteuerung

---

### Dimmersteuerung

#### Besonderheiten bei Dimmern

Dimmer werden vorzugsweise als Schieberegler mit ein/aus Tasten oder nur als Schieberegler dargestellt. Der Wert dieser Schieberegler kann z.B. mit der Maus am Bildschirm oder über eine Makroanweisung verändert werden.

Um eine bestimmte Helligkeit an einem Dimmer einzustellen wird einfach dem Dimmerobjekt eine Zahl zwischen 0 und 100 zugewiesen. 0 schaltet den Dimmer aus, 100 auf volle Helligkeit.

Beispiel:

DimmerDiele:=50

#### **Bitte beachten Sie:**

Nachdem am Bildschirm ein neuer Wert für einen Dimmer eingestellt wurde, zeigt dieser die Werte an, die während des Einstellvorgangs gemeldet werden. Er springt dabei meistens auf den alten Wert oder eine geringfügig davon abweichende Einstellung zurück. Erst nachdem der Verstellvorgang abgeschlossen ist, wird der neue aktuelle Wert von der Hardware gesendet und eingestellt.

## 1.24.8. Duty-Cycle

---

### Duty-Cycle

Zu Funkmodulen im Bereich 868 MHz gibt es gesetzliche Vorschriften, nach denen ein Gerät maximal 1% seiner Betriebszeit senden darf. Im täglichen Betrieb führt das normalerweise zu keiner Einschränkung, aber man muss bei der Programmierung beachten, dass die CCU nicht unnötigerweise zu häufig sendet. Das würde im Extremfall dazu führen, dass das Duty-Cycle-Zeitkonto überläuft und das Senden für ein Zeitlang nicht mehr möglich ist.

Die CL-Software vermeidet unnötige Funksendungen automatisch, indem nur gesendet wird wenn der Zustand eines Aktors sich ändert.

Es gibt aber einige Anweisungen, mit denen der Benutzer selbst Funkmeldungen generieren und dadurch den Duty-Cycle belasten kann.

Das sind folgende Anweisungen:

[SENDE](#)

[SETZEWERT](#)

[ABFRAGE](#)

[ALLEABFRAGEN](#)

Diese Anweisungen werden nur in Ausnahmefällen benötigt, die Benutzung sollte dann nur sparsam und mit Bedacht erfolgen um das Duty-Cycle-Konto nicht unnötig zu belasten. Detaillierte Informationen zu den Anweisungen sind in der Hilfe beschrieben.

## 1.24.9. ExecEngine

---

### ExecEngine

Die Exec-Engine ist das Programm, das auf der Zentrale läuft und für die komplette Logik und Zeitsteuerung der Anwendung zuständig ist. Die Informationen zur programmierten Anwendung erhält die Exec-Engine über die Ausführungsdatei. Die Ausführungsdatei beinhaltet die in einen speziellen optimierten Binärcode übersetzten Makros, Zeittabellen und Definitionen, die mit dem Konfigurationsprogramm erstellt worden sind. Auch die Daten zur Visualisierung für das Programm **VisuWin** werden von der Exec-Engine bereitgestellt, wobei die Anzahl der möglichen PC's, mit denen die Exec-Engine gleichzeitig kommuniziert von der Version abhängig ist.

## 1.24.10. ExecEngineWin

---

### ExecEngineWin

Mit dem Programm **ExecEngineWin** wird die [Exec-Engine](#) ( das Ausführungsprogramm auf der Zentrale für die Logikfunktionen und Visualisierungen ) gesteuert. Dieses Programm dient auch zur Anzeige des aktuellen Status und zur Übertragung neuer oder veränderter Ausführungsdateien an die Exec-Engine.

Zum Aufbau einer Verbindung mit der Exec-Engine sind die Angaben der IP-Adresse und des Port des Gerätes auf dem die Exec-Engine läuft (also im Normalfall die Zentrale) erforderlich. Diese Daten werden aus dem Konfigurationsprogramm übernommen und können bei Bedarf im Programm ExecEngineWin verändert werden.

Bei der ersten Inbetriebnahme und bei Änderungen der Verbindungsdaten muss die Konfigurationsdatei mit dem Knopf [Konfigurationsdatei übertragen] auf der Seite **Einstellungen** zur Exec-Engine übertragen werden, damit die interne Kommunikation der Programme korrekt funktioniert.

Wenn keine Verbindung hergestellt werden kann, kann es erforderlich werden die Zentrale neu zu starten (zu booten).

Bitte beachten Sie, dass die im Folgenden beschriebenen Schritte nur im Ausnahmefall durchgeführt werden müssen wenn die Aktualisierung der Ausführungsdatei auf der Zentrale mit dem Blitz-Button nicht funktioniert !

Die einzelnen Schritte müssen in der angegebenen Reihenfolge ausgeführt werden.

1. Auf der Seite **Einstellungen** muss die Konfigurationsdatei durch Mausklick auf den Knopf [Konfigurationsdatei übertragen] zur Exec-Engine übertragen werden.
2. Auf der Seite **Dateien** wird der Pfad für die Ausführungsdatei und die Ausführungsdatei eingetragen und durch Mausklick auf den Knopf [Übertragen auf Zentrale] zur Exec-Engine übertragen.
3. Auf der Seite **Steuerung/Status** die Exec-Engine mit dem Knopf [Exec-Engine starten] starten bzw. mit dem Knopf [Reset Exec-Engine] neu starten wenn sie schon aktiv ist.

Nachdem diese Schritte erledigt sind, sind alle erforderlichen Daten auf der Zentrale gespeichert. Bei einem Neustart der Zentrale startet die Exec-Engine automatisch mit der aktuell eingestellten Ausführungsdatei.

## 1.24.11. Freigabe des Programms

---

### Freigabe des Programms

Die Freigabe zur zeitlich unbegrenzten Nutzung des Programms erfolgt durch Eingabe des Freigabecodes oder der PLN (Persönlichen Lizenznummer) im Freigabefenster.

Das Freigabefenster wird mit dem Menüpunkt "Projekt->Freigabe" geöffnet.

Wenn Sie die PLN eingeben wird diese benutzt um einen Freigabecode im Internet berechnen zu lassen. Wenn das Programm keinen Internetzugriff hat, kann der Freigabecode mit der PLN manuell mit einem Browser auf der entsprechenden Internetseite, die im Freigabefenster angezeigt wird, abgeholt werden. Dieser Freigabecode wird dann im Eingabefeld "Freigabecode" angegeben.

Bitte beachten Sie, dass bei der Freigabe im Reiter *Zentrale* des Hardwarefensters (Menüpunkt *Konfigurieren-->Hardware*) der Zentralentyp ausgewählt sein muss, für den die Freigabe erfolgen soll. Bei der Freigabe muss eine Verbindung zur Zentrale bestehen, da die Freigabe in der Zentrale gespeichert wird und nicht PC-abhängig ist.

Wenn Sie keine Zentrale benutzen und mit einem LAN-Adapter im PC-Modus arbeiten muss als Zentralentyp PC ausgewählt werden. In diesem Fall muss mindestens ein Konfigurationsadapter in der BidCos-Konfigurationsdatei angelegt worden sein. Zum Editieren der BidCos-Konfigurationsdatei klicken Sie auf der Seite *BidCos* im Hardwarefenster auf den entsprechend beschrifteten Button.

Die Freigabe wird nur einmal durchgeführt und für die weitere Verwendung gespeichert.

**Bitte bewahren Sie Ihre PLN unbedingt gut auf, es kann sein, dass Sie diese benötigen um künftige Updates zu erhalten oder nach einem Austausch der Zentrale eine erneute Freigabe zu machen.**

## 1.24.12. Gruppen

---

### Gruppen

Über das Anlegen von Gruppen ist es möglich mehreren Objekten gleichzeitig neue Werte zuzuweisen.

Zudem gibt es in Apps die Option Ansichten in Gruppen zu organisieren.

Die Gruppen, die Objekten zugeordnet werden können, werden auf der Seite *Allgemein* des Einstellungsfensters angelegt.

Auf der Seite *Allgemein* der Objekte können diese Gruppen den Objekten zugeordnet werden, dabei können den Objekten mehrere Gruppen gleichzeitig zugeordnet werden.

Einem Licht im Erdgeschoss kann also z.B. die Gruppe *Erdgeschoss* und die Gruppe *Licht* zugeordnet werden.

Die Zuweisung von Werten an Gruppen wird durch die Anweisung [GRUPPENZUWEISUNG](#) vorgenommen.

## 1.24.13. History und Log-Dateien

---

### History und Log-Dateien

**Bitte beachten Sie:**

Diese Features stehen nicht in allen Versionen zur Verfügung.

Mit den Versionen ab CL-Studio können Sie das Schreiben von Historydaten aktivieren. Dazu muss die Option *Historydaten schreiben* im Reiter *Allgemein* des Einstellungsfensters (Menüpunkt Konfigurieren->Einstellungen) aktiviert werden. Zusätzlich muss für die einzelnen Module für die Historydaten geschrieben werden sollen die Option *Historydaten schreiben* im Reiter *Allgemein* des Objekts aktiviert werden.

Das Schreiben einer Log-Datei mit allgemeinen Informationen zur Ausführung wird wie die Option zum Schreiben von Historydaten im Reiter *Allgemein* des Einstellungsfensters (Menüpunkt Konfigurieren->Einstellungen) aktiviert.

Je nach Art der Zentrale werden die Dateien in unterschiedliche Verzeichnisse geschrieben. Das jeweilige Verzeichnis kann im Reiter *Zentrale* des Hardwarefensters festgelegt werden. Bei der CCU1 kann das Verzeichnis nicht verändert werden, da nur das vorgegebene Verzeichnis verfügbar ist.

**Bitte unbedingt beachten:**

Wenn eine CCU1 oder CCU2 verwendet wird muss ein USB-Stick bzw. eine SD-Karte eingebunden werden, wenn in den internen Speicher der CCU geschrieben wird kommt es nach einiger Zeit zu Fehlfunktionen weil der interne Speicher überläuft.

Hinweise zum Einbinden eines USB-Sticks bei der CCU1 bzw. einer SD-Karte bei der CCU2 finden Sie im Kapitel [SCHREIBEDATEI](#).

History und Log-Datei können mit einem Browser angezeigt, downgeloadet und gelöscht werden.

Bei der CCU1 und CCU2 muss dazu folgende Eingabe in der Adresszeile des Browser gemacht werden:

**http://<IP der Zentrale>/addons/contronics/files/workfiles.cgi**

also zum Beispiel

**http://192.168.0.136/addons/contronics/files/workfiles.cgi**

Bei der CCU3 wird ein anders Verzeichnis benutzt:

**http://<IP der Zentrale>/addons/cl-control/workfiles.cgi**

also zum Beispiel

**http://192.168.0.53/addons/cl-control/workfiles.cgi**

## 1.24.14. Objekte

---

### Objekte

Die Objekte sind die zentralen Elemente für alle Steuerungsfunktionen.

Zunächst die Erklärung zu einigen Begriffen, die verwendet werden um ein Objekt zu beschreiben:

**Aktoren** sind Schaltmodule, also Empfänger. Wenn ein Modul mehrere Ausgänge hat, so ist jeder Ausgang ein separater Aktor.

**Sensoren** sind Sendemodule, wobei jede Taste einer Fernbedienung bzw. jeder Kanal eines Sensors ein separater Sensor ist.

Der Begriff **Objekt** umfasst einen Sensor bzw. einen Aktor inklusive der gesamten Definition, die für diesen im PC-Programm gemacht wurde (Darstellungsart, Makros Schaltzeiten usw.)

Wenn der Objektzustand eines Aktor- Objektes durch Mausklick oder Makros geändert wird, so wird automatisch an den zugehörigen Aktor eine Meldung gesendet und dieser schaltet auf den Zustand des Objekts.

Wenn ein Sensor-Objekt eine Meldung seines Sensors empfängt, ändert sich der Zustand des Objektes entsprechend und das Makro des Objekts wird ausgeführt.

Objekte können während der Ausführung eines Projektes in verschiedenen Ansichten am Bildschirm angezeigt werden. Die Zustände von Objekten können normalerweise durch Mausklick verändert werden. Für die [Darstellung eines Objektes](#) gibt es je nach Objekttyp mehrere Optionen. Es kann auch Objekte geben, denen kein Sensor oder Aktor zugeordnet ist, diese werden *virtuelle Objekte* genannt. Virtuelle Objekte können auf der Seite Objekte des Einstellungsfenster angelegt werden.

Es gibt auch ein automatisch erzeugtes virtuelles Objekt, das Objekt [Anwesenheitssimulation](#), das automatisch immer dann erzeugt wird, wenn im Fenster Anwesenheitssimulation etwas eingetragen wurde. Dieses Objekt ist vom Typ Schalter, hat aber kein zugeordnetes Hardware-Modul, sondern dient nur dazu die Anwesenheitssimulation zu aktivieren bzw. zu deaktivieren.

Ein weiteres virtuelles Objekt ist das zu jedem Projekt automatisch erzeugte Objekt *Uhr*. Das Objekt *Uhr* ist ein Anzeigefenster, in dem immer die aktuelle Uhrzeit (im 5-Sekunden-Rhythmus angezeigt wird).

In der Standard-Version wird automatisch ein Objekt vom Typ Zeichen mit dem Namen Anzeige erzeugt. Dieses Objekt kann zur Ausgabe von beliebiger werte benutzt werden.

In der Studio-Version können beliebig viele "virtuelle" Objekte mit beliebigem Typ angelegt werden, also auch beliebig viele Anzeige-Objekte, daher wird hier kein Anzeige-Objekt automatisch erstellt.



## 1.24.15. Objektdarstellung

---

### Objektdarstellung

Für ein [Objekt](#) gibt es verschiedene Darstellungsmöglichkeiten. Es kann als [Objektrahmen](#), [Textrahmen](#) oder als kleines Bild-Symbol ([Bitmap](#)) dargestellt werden. Welche Darstellungsart aktuell verwendet wird kann auf der Seite Visualisierung der Objektdefinitionen festgelegt werden.

Für Objekt mit mehreren möglichen Zuständen kann für die unterschiedlichen Zustände jeweils ein anderes Bild-Symbol definiert werden. Wenn nur ein Bild zur Visualisierung für ein Objekt ausgewählt wird, kann man aufgrund der Position des Objekts im Hintergrundbild erkennen um welches Objekt es sich handelt.

## 1.24.16. Objektrahmen und Textrahmen

---

### Objektrahmen und Textrahmen

Objektrahmen und Textrahmen dienen dazu ein Objekt und seinen Zustand oder Wert auf dem Bildschirm darzustellen. In einem Objektrahmen wird neben der Ausgabe der Objektbezeichnung und dem aktuellen Zustand als Text im rechten Teil auch ein Symbol angezeigt, wenn für dieses Objekt ein Symbol ausgewählt wurde. Durch Anklicken eines Objektrahmens oder Textrahmens kann der Zustand des Objekts verändert werden.

Die Grösse und Position für Objektrahmen und Textrahmen der einzelnen [Objekte](#) kann im [Entwurfswindow](#) festgelegt werden. Dieses Entwurfswindow wird mit dem Knopf **[Ansicht bearbeiten]** auf der [Seite Ansichten](#) des [Einstellungsfensters](#) geöffnet.

## 1.24.17. Programmierung

---

### Programmierung

Die Programmierung des Systems erfolgt über eine eigene [Makrosprache](#) mit vielen leistungsfähigen Anweisungen.

Die Anweisungen werden auf der Seite [Makro](#) der [Objektdefinitionen](#) eingegeben.

Alternativ können Anweisungen auch im Fenster [Makro-Anweisungen über Menü erstellen](#) eingegeben werden.

## 1.24.18. Projekt

---

### Projekt

Ein Projekt ist eine individuell erstellte Anwendung mit den ausgewählten Modulen, den Makros, programmierten Funktionen und erstellten Definitionen.

Das Projekt wird in einer Datei mit der Endung SPG gespeichert.

**Bitte beachte Sie unbedingt:**

Die Projektdatei muss regelmässig gesichert werden, da bei einem Festplattenfehler alle Konfigurationen und Programmierungen verloren gehen.

Die Konfigurationen und Programmierungen werden nicht in der Zentrale gespeichert!

Keinesfalls sollte der Projektname öfters geändert werden, weil sowohl auf der Zentrale, in Apps und auch auf dem PC Dateien und Daten in Abhängigkeit vom Projektnamen erzeugt und gespeichert werden.

Für die Sicherungen kann/muss natürlich ein andere Name vergeben werden, aber der Projektname, der auf die Zentrale übertragen wird sollte nicht bzw. möglichst selten verändert werden. Bei Änderungen des Namens kommt es dann z.B. bei Apps die Einstellungen unter dem Projektnamen speichern zu scheinbaren Datenverlusten und es entstehen unnötige Dateien auf der Zentrale..

Mit dem Menüpunkt Projekt->Schnellsicherung wird eine Sicherungsdatei erzeugt, bei der Datum und Uhrzeit an den Projektnamen angehängt werden. Speichern Sie diese Dateien nicht nur auf der Festplatte, sondern auch auf einem Datenträger, z.B. einem USB-Stick.

## 1.24.19. Schieberegler

---

### Schieberegler

Schieberegler werden benutzt um Zahlenwerte einzustellen. Ein Anwendungsbeispiel ist die Einstellung der Helligkeit eines Dimmers. Der aktuelle Wert des Moduls wird in einem Schieberegler oben rechts angezeigt. Der Wert des Schiebereglers kann manuell am Bildschirm mit der Maus oder durch ein Makro verändert werden.

## 1.24.20. Sensoren

---

### Sensoren

**Sensoren** sind Eingabeelemente wie z.B. Taster von Fernbedienungen, Bewegungsmelder, Temperaturfühler, Wassermelder, Rauchmelder usw.

## 1.24.21. Sonnenzeiten

---

### Sonnenzeiten

In einigen Zeitfenstern kann als Zeitpunkt zur Ausführung einer Aktion die Zeit des Sonnenaufgangs bzw. des Sonnenuntergangs für das aktuelle Datum angegeben werden. In wenn-Bedingungen können diese Zeiten zum Vergleich mit der aktuellen Uhrzeit verwendet werden. Zur Eingabe des Zeitpunkt *Sonnenaufgang* geben Sie anstelle der Uhrzeit die Buchstaben **SA** ein, für den *Sonnenuntergang* **SU**.

Es wird der Zeitpunkt der sogenannten „bürgerlichen Dämmerung“ berechnet. Es gibt noch andere Definitionen zu Sonnenaufgangs- und –untergangszeiten (z.B. astronomische Sonnenzeiten, nautische Sonnenzeiten), die teilweise in Abhängigkeit der Jahreszeiten stark voneinander abweichen. Daher sind diese Zeiten eventuell nicht identisch mit Angaben in Tabellen, in denen andere Sonnenzeiten verwendet werden. Die berechneten Zeiten sind bis auf wenige Minuten genau – Voraussetzung ist die Angabe des geografischen Standorts nach Postleitzahl oder Längen- und Breitengrad.

Die Zeiten für Sonnenaufgang und Sonnenuntergang können im Einstellungsfenster im Reiter **Allgemein** mit dem Button "weitere Einstellungen" individuell angepasst werden.

## 1.24.22. Spezielle Objekte

---

### Spezielle Objekte

Es können spezielle Objekte angelegt werden, die dazu dienen besondere Situationen, Warnungen oder Fehlermeldungen zu bearbeiten oder bestimmte Systemwerte einzustellen.

Spezielle Objekte sind normalerweise virtuelle Objekte, die je nach Art der Verwendung als Objekte vom Typ Zeichen, Schalter oder Zahl angelegt werden.

Existieren keine passenden Objekte für eine Funktion, so bleibt die Auswahlliste für diese Objektart leer.

Details zur Verwendung der einzelnen speziellen Objekte sind jeweils im zugehörigen Abschnitt beschrieben.

#### Level für History

Hier kann ein vorher definiertes Objekt vom Typ Zahl angegeben werden. Dieser Wert kann über Makros während der Programmausführung geändert werden.

Die Zahl kann folgende Werte haben:

0 : Es werden keine Historydaten geschrieben, auch wenn diese Option für ein Objekt eingestellt ist.

1 : Es werden Historydaten geschrieben für alle Objekte, für die diese Option aktiviert ist.

2 : Es werden Historydaten für alle Hardware-Objekte und für alle Objekte, für die die Historyoption aktiviert ist, geschrieben.

3 : Es werden Historydaten für alle Objekte geschrieben, auch für Objekte für die die Historyoption nicht aktiviert ist.

Bitte beachten Sie unbedingt: Bei Level 2 und 3 können grosse Datenmengen anfallen.

Insbesondere bei Verwendung einer CCU muss darauf geachtet werden, dass die Historydatei nicht in den normalen Speicher, sondern auf die SD-Karte oder einen USB-Stick geschrieben wird, um Fehlfunktionen aufgrund eines Speicherüberlaufs zu vermeiden.

#### Einstellung für Systemlog

Hier kann ein vorher definiertes Objekt vom Schalter oder Zahl angegeben werden. Beim Schalterzustand "ein" werden die Historydaten der Stufe 1 geschrieben, ist das Objekt vom Typ Zahl werden Systemmeldungen der in der Zahl angegebenen Stufe geschrieben. 0=kein Systemlog, 1=nur wichtige Systemmeldungen wie Ausführungsstart, Ausführungsende und Fehler, 2=zusätzlich werden Warnungen in die Log-Datei geschrieben, 3=es werden Fehler, Warnmeldungen und Informationen geschrieben. Die Stufe 3 kann grössere Datenmengen erzeugen und sollte im Normalfall nicht verwendet werden. Beachten Sie, dass es bei einem Speicherüberlauf zu gravierenden Fehlfunktionen und einem Systemabsturz kommen kann. Das gilt insbesondere bei Verwendung einer CCU ohne DS-Karte oder USB-Stick.

#### Objekt für Verbindungsfehler

Hier kann ein vorher definiertes Objekt vom Typ Zeichen angegeben werden. Jedesmal wenn eine Meldung über eine Kommunikationsstörung empfangen wird, wird der Name des Objekts, bei dem die Kommunikationsstörung aufgetreten ist als Wert für das *Objekt für Verbindungsfehler* gesetzt und das Makro wird ausgeführt.

#### Allgemeines Fehlerobjekt

Hier kann ein vorher definiertes Objekt vom Typ Zeichen angegeben werden. Wenn im System ein Fehler erkannt wird, werden Informationen dazu in dieses Objekt geschrieben und das Makro des Objekts wird ausgeführt.

#### Objekt für Sabotagemeldungen

Hier kann ein vorher definiertes Objekt vom Typ Zeichen angegeben werden. Jedesmal wenn eine Sabotagemeldung empfangen wird, wird der Name des Objekts das die Sabotagemeldung gesendet hat als Wert für das *Objekt für Sabotagemeldungen* gesetzt und sein Makro wird



ausgeführt.

### **Objekt für Batteriemeldungen**

Hier kann ein vorher definiertes Objekt vom Typ Zeichen angegeben werden. Jedesmal wenn eine Batteriewarnung empfangen wird, wird der Name des Objekts, das die Batteriewarnung gesendet hat als Wert für das *Objekt für Batteriemeldungen* gesetzt und das Makro wird ausgeführt.

### **Hauptschalter für Heizen**

Hier kann ein vorher definiertes virtuelles Objekt vom Typ Schalter angegeben werden. Beim Schalterzustand "aus" wird die Temperatursteuerung der Thermostate für die Binäraktoren zum Heizen deaktiviert. Mit diesem Schalter ist es möglich in Abhängigkeit von der Jahreszeit oder der aktuell gemessenen Aussentemperatur die Heizfunktion über den Binäraktor komplett zu deaktivieren oder zu aktivieren.

### **Hauptschalter für Kühlen**

Hier kann ein vorher definiertes virtuelles Objekt vom Typ Schalter angegeben werden. Beim Schalterzustand "aus" wird die Temperatursteuerung für den Binäraktoren zum Kühlen deaktiviert. Mit diesem Schalter ist es möglich in Abhängigkeit von der Jahreszeit oder der aktuell gemessenen Aussentemperatur die Kühlfunktion über den Binäraktor komplett zu deaktivieren oder zu aktivieren.

## 1.24.22.1. Typdefinitionen

---

### Typdefinitionen

Im Fenster Typdefinitionen können eigene Typen und Zustände definiert werden. Diese erscheinen dann in den entsprechenden Listen zur Typauswahl eines Objekts und können dort ausgewählt werden.

Es gibt einige vom System benutzte Standard-Typen. Diese erscheinen mit dem Hinweis – *reserviert* in dem Listfenster und können nicht bearbeitet oder gelöscht werden.

Zu jedem Typ können in der rechten Listbox beliebig viele Zustände für den jeweils markierten Typ angelegt werden.

Wenn ein Typ die Zustände AN/EIN und AUS verwendet, muss der Zustand AUS als erster, der Zustand AN/EIN als zweiter Zustand definiert werden. Dies ist erforderlich, damit dieser Typ kompatibel ist mit den vordefinierten AN/AUS-Typen und mit den Anweisungen wie EINSCHALTEN/AUSSCHALTEN behandelt werden kann.

Bei Typen für Jalousien und Rollläden müssen unbedingt die beiden Zustände "oben" und "unten" als erste definiert werden.

Der Name eines Typs kann die Buchstaben A-z, Ziffern und den Unterstrich beinhalten. Er muss mit einem Buchstaben beginnen, landesspezifische Buchstaben und Sonderzeichen sind nicht erlaubt (Beispiel: Typ\_Licht\_2).

In der rechten Listbox werden die Zustände verwaltet. Mit dem Knopf **[Bildsymbol]** kann ein Standard-Bildsymbol für den markierten Zustand festgelegt werden. Wenn der definierte Typ einem Objekt zugewiesen wird, wird automatisch dieses Bild im Programm [VisuWin](#) zur Anzeige des Zustands benutzt.

Falls ein anderes Bildsymbol benutzt werden soll, kann für das einzelne Objekt optional ein anderes Bildsymbol ausgewählt werden.

## 1.24.23. Sprachausgabe

---

### Sprachausgabe

Mit der Sprachausgabe-Funktion kann Text als Sprache ausgegeben werden. Es ist damit möglich in einem Makro eine Zeichen-Variable oder einen Text in Hochkommas als Sprache auszugeben.

Die Ausgabe erfolgt über das Visualisierungsprogramm [VisuWin](#), dieses muss auf einem PC aktiv sein, damit die Sprache auf diesem PC ausgegeben werden kann.

Voraussetzungen sind eine Soundkarte im PC und entsprechende Einstellungen in Windows.

Sollte eine englische Stimme auf dem PC verwendet werden kann ein Sternchen vor den auszugebenden Text gesetzt werden um die Qualität der Sprachausgabe zu verbessern. Die beste Qualität erreicht man aber wenn landesspezifische Stimmen in Windows installiert sind.

## 1.24.24. Visualisierung

---

### Visualisierung

Als Visualisierung wird die Darstellung der einzelnen Objekten und deren aktueller Zustände bzw. Werte in einem Fenster bezeichnet. Die Visualisierung auf dem PC erfolgt mit dem Programm [VisuWin](#). Dieses Programm kann separat oder auch durch Anklicken des Visualisierungsknopfes im Konfigurationsprogramm aufgerufen werden.

Es können individuelle eigene Ansichten erstellt werden, dazu wird das [Fenster Visualisierung](#) benutzt.

Bitte unbedingt beachten:

Das Visualisierungsprogramm [VisuWin](#) kann nur bei aktiver Ausführung benutzt werden.

## 1.24.25. Visualisierung mit VisuWin

---

### Visualisierung mit VisuWin

Das Programm **VisuWin** ist das Visualisierungsprogramm für Windows zur CL-Software. Das Programm kann auf einem oder mehreren PC's im Netzwerk installiert werden und kommuniziert mit der [Exec-Engine](#). Das Programm zeigt die mit dem Konfigurationsprogramm erstellten Visualisierungen an. Dabei ist es auch möglich Zustände und Werte von Objekten über den Bildschirm zu verändern.

Vor der ersten Anzeige von Visualisierungsansichten muss die Verbindung zur Exec-Engine hergestellt werden. Die dazu erforderlichen Informationen (IP-Adresse und Port der Exec-Engine, im Normalfall immer die Zentrale) werden unter dem Menüpunkt Konfiguration->Verbindung eingestellt.

## 1.24.26. Zeilenvorschub

---

### Zeilenvorschub

Wenn bei der Ausgabe von Texten über ein Objekt vom Typ Zeichen kann ein Zeilenvorschub mit zwei<<-Zeichen eingefügt werden.

Beispiel:

```
Textausgabe:="Das ist die erste Zeile<<und das ist die zweite Zeile."
```

## 1.24.27. Zustandsmelder

---

### Zustandsmelder

Bitte beachten Sie, dass bei Zustandsmeldern normalerweise nur eine Meldung kommt wenn der besondere Zustand dieses Melders erreicht ist, aber keine Meldung wenn dieser Zustand wieder in den Normalzustand wechselt.

Ein Regensensor sendet z.B. nur eine Meldung wenn es beginnt zu regnen, nicht aber wenn es aufhört, ein Bewegungsmelder sendet nur eine Meldung in dem Moment wo er eine Bewegung erkennt.

Eine eventuell in dem Zustandsmelder einzustellende Einschaltzeit wird nicht berücksichtigt, da die Aktionen auf die Meldung von der Programmierung bestimmt werden.

Auch das Zurücksetzen des Sensorzustands auf den Normalwert muss vom Programm vorgenommen werden, damit die Visualisierung korrekt ist.

Im Reiter Allgemein der Objektdefinitionen kann im Eingabefeld "Resetzeit in Sekunden" eine Zeit angegeben werden nach der der Zustand des Objekts automatisch wieder auf den AUS-Zustand gebracht wird.

Eine andere Option ist, dieses Zurücksetzen in den Normalzustand im Makro des Objekts zu machen.

Dabei kann die [Anweisung warten](#) oder [WARTE](#) benutzt werden, damit der Zustand einige Zeit am Bildschirm angezeigt wird.

#### **Beispiel bei einem Bewegungsmelder:**

*Anweisungen...*

warte 10 Sekunden

Bewegungsmelder ausschalten

## 1.24.28. Variablen über virtuelle Objekte setzen

### Variablen über virtuelle Objekte setzen

Bei einigen Geräten (z.B. ZigBee-Geräten) werden spezielle Eigenschaften über Variablen gesteuert. Nicht in allen Visualisierungsoptionen stehen spezielle Funktionen zur Steuerung solcher Geräte zur Verfügung. In dem Fall können einfach virtuelle Objekte erstellt werden, deren Werte dann die Variablen des Geräts setzen.

Der Objektwert einer ZigBee-Farbleuchte ist der Ein/Aus-Zustand des Objektes. Um die Farbe über eine Visualisierung zu setzen, in der kein spezielles Objekt für die Farbleuchte zur Verfügung steht wird dann einfach ein virtuelles Objekt von Typ Zahl mit dem Wertebereich 0-65535 angelegt und im Makro des Objekts die Farb-Variable des Geräts gesetzt, wie im Screenshot gezeigt.

Objekt bearbeiten

Farbe

Namen ändern

Allgemein | **Makro** | Visualisierung

Ausführungsintervall:  ☒ Ausführen bei Änderung

Farbleuchte.Hue:=selbst

Menü-Makro bearbeiten

Objekte/Variablen

Anweisung

Editor

Variablen:

Name	Typ	Startwert

Löschen

? Hilfe

X Abbrechen

OK



## 1.25. FAQs

---

### Allgemein

Wenn scheinbar grundlos plötzlich diffuse Probleme auftreten oder normale Funktionen nicht mehr funktionieren hilft es oftmals die Zentrale (CCU) neu zu starten.

Wenn die Kommunikation mit der Zentrale nicht funktioniert lässt sich das in den meisten Fällen beheben, indem die CCU länger als 20 Sekunden stromlos gemacht wird (nicht kürzer damit auch der BidCoS komplett neu startet).

Wenn das nicht hilft wäre der nächste Schritt den PC einmal zu booten.

---

### Problem:

Das PC-Programm hängt beim Start, am unteren Rand des Fensters wird die Meldung "Initialisierung und Verbindungsaufbau" angezeigt.

Das passiert wenn keine Verbindung zur Zentrale mit der angegebenen gespeicherten IP-Adresse hergestellt werden kann.

### Lösung:

Drücken Sie unmittelbar nach dem Programmstart irgendeine Taste auf der Tastatur (solange "Verbindungsaufbau deaktivieren mit Tastendruck..." angezeigt wird).

Dann wird nicht versucht eine Verbindung zur Zentrale aufzubauen und Sie können die Verbindungseinstellungen prüfen.

Oftmals ist die Ursache, dass die IP-Adresse der Zentrale durch DHCP geändert wurde.

Wenn es nicht an einer falschen IP-Adresse liegt kann das Problem meistens durch einen Neustart der Zentrale behoben werden, hilft ein einfacher Neustart nicht kann es sein, dass die Zentrale länger als 20 Sekunden stromlos gemacht werden muss.

Eine andere Ursache kann sein, dass die Verbindung nach einem Update des Virenschanners blockiert wird, in dem Fall muss die Freigabe der Verbindung im Virenschanner freigegeben werden.

---

### Problem:

In der Visualisierung VisuWin werden Zustände nicht automatisch aktualisiert, man erkennt das z.B. daran, dass die Uhrzeit im Uhr-Objekt nicht alle 5 Sekunden aktualisiert wird.

Das passiert wenn die Aktualisierungsmeldungen von der Zentrale nicht empfangen werden.

### Lösung:

Unter dem Menüpunkt "Konfiguration->Verbindung" in VisuWin ist keine oder eine falsche eigene IP-Adresse angegeben worden. Das muss dann berichtigt werden.

Die Meldungen der Zentrale (CCU) werden von der Windows-Firewall blockiert (kann z.B. nach einem Windows-Update passieren). Dann muss VisuWin.exe in der Windows-Firewall für den Empfang freigeschaltet werden.

Ein Virenschann blockiert die Verbindung, dann muss diese dort freigeschaltet werden.

Die Netzwerkroutrinen der Zentrale oder des PCs sind durcheinander. Dann muss die CCU mindestens 20 Sekunden stromlos gemacht werden bzw. der PC neu gebootet werden.

---

**Problem:**

Es wird eine neue Freigabe mit der Lizenznummer verlangt, aber diese funktioniert nicht.

**Lösung:**

Meistens liegt es daran, dass nicht die reguläre PLN benutzt wird, sondern eine temporäre PLN für einen Testzeitraum.

Eine reguläre PLN hat ein "W" an 5. Stelle, eine temporäre PLN hat ein "T" an 5. Stelle.

Unter diesem Link finden Sie eine Anleitung zur Freigabe:

<http://www.cl-control.de/downloads/pdf/FreigabeCLStudio.pdf>

Sollte die Freigabe trotz erfolgreicher neuer Freigabe erneut angefordert werden, so wurde diese wahrscheinlich nicht

korrekt in der Zentrale gespeichert. In diesem Fall hilft es normalerweise die Zentrale neu zu starten und die Freigabe dann zu wiederholen.

## 1.26. Online-Informationen

---

### Online-Informationen

Aktuelle Informationen und Updates zum Programm finden Sie im Internet unter:

<http://www.cl-control.de/software.html>



# Index

## A

ABBRECHEN, 98  
ABFRAGE, 99  
AKTIVIEREN, 100  
AKTIVIERT, 144  
Aktoren, 47, 246  
Alarmsirene HmIP-ASIR, 207  
ALLEABFRAGEN, 101  
ALLEWERTESICHERN, 102  
Allgemein, 7  
Ansichten, 50  
Ansichten bearbeiten, 50  
Anweisung ABFRAGE, 99  
Anweisung AKTIVIEREN, 100  
Anweisung ALLEABFRAGEN, 101  
Anweisung ALLEWERTESICHERN, 102  
Anweisung ANZEIGEN, 103  
Anweisung Audio-Datei abspielen, 80, 115  
Anweisung AUFRUFEN, 104  
Anweisung DEAKTIVIEREN, 105  
Anweisung ERLEDIGT, 106  
Anweisung GEHEZU, 107  
Anweisung GETCCUSYSVAR, 108  
Anweisung GETSITE, 109  
Anweisung GRUPPENZUWEISUNG, 111  
Anweisung LESEWERTEDATEI, 112  
Anweisung LÖSCHEANZEIGE, 113  
Anweisung LÖSCHEDATEI, 114  
Anweisung Makro ausführen, 81  
Anweisung Makro starten, 82  
Anweisung PLAY, 115  
Anweisung SCHLIESSEDATEIEN, 116  
Anweisung SCHREIBEDATEI, 117  
Anweisung SENDE, 119  
Anweisung SENDEMAIL, 120  
Anweisung SENDESMS, 122  
Anweisung SETCCUSYSVAR, 124  
Anweisung SETZEHISTORYDIFFERENZ, 125  
Anweisung SETZEKONFIG, 126  
Anweisung SETZEWERT, 127  
Anweisung SETZEZEITTABELLENWERT, 130  
Anweisung SICHTBAR, 131  
Anweisung SPRACHAUSGABE, 132  
Anweisung STARTE, 133  
Anweisung STARTPROGRAMM, 134  
Anweisung STARTUHR, 135  
Anweisung STARTWIN, 136  
Anweisung STOPPE, 137  
Anweisung TEMPERATURABSENKUNG, 138  
Anweisung UNSICHTBAR, 139

- Anweisung VERLASSEN, 140
- Anweisung WARTE, 141
- Anweisung warten, 79
- Anweisung WENN, 142
- Anweisung Windows Programm starten, 83
- Anweisungen, 95
- Anwesenheitssimulation, 74
- ANZEIGEN, 103
- Astrozeiten, 55
- Audio-Datei abspielen, 80, 115
- AUFRUFEN, 104
- Ausführung auf PC, 63

## B

- Batteriewarnung, 145, 267
- Bedingungen, 92
- Bedingungen für Zeiträume, 92
- Bewegungsmelder, 241
- Bildsymbole, 249
- Bild-Symbole auswählen, 249
- Bild-Symbole bearbeiten, 249
- Bildsymbole Objekt, 73
- Bitmap, 250

## C

- CCU, 20, 61
- COMERROR, 146
- CT, 87

## D

- Das Fenster Typdefinition, 269
- Dateiverwaltung auf der Zentrale, 251
- DATEIVORHANDEN, 147
- DATUM, 148
- DEAKTIVIEREN, 105
- Dimmer, 252, 264
- Duty-Cycle, 253

## E

- Einstellen der Anwesenheitssimulation, 74
- Einstellungen, 49
- Einstellungen nach der Installation, 20
- Einstellungen Sonnenzeiten, 266
- email Konfiguration, 57
- e-mail Konfiguration, 57
- Empfänger, 246
- End-Makro, 85
- Entwurfsfenster, 75, 272
- E-Paper-Display, 199, 214
- ERLEDIGT, 106
- ERSETZEN, 171
- Erste Schritte, 10

## F

Feiertage, 56, 149  
Fenster Aktoren, 47  
Fenster Anwesenheitssimulation, 74  
Fenster Einstellungen Sonnenzeiten, 55  
Fenster e-mail-Konfiguration, 57  
Fenster Feiertage, 56  
Fenster Hardware, 61  
Fenster Makro-Anweisungen, 77  
Fenster Menü-Makro bearbeiten, 77  
Fenster Schaltsensoren, 45  
Fenster Sensoren, 48  
Fenster Sonnenzeiten, 55  
Fenster Spezielle Objekte, 267  
Fenster Visualisierung, 75  
Fernbedienung mit 19 Tasten, 191  
FHZ, 28  
FHZ2000, 22  
Freigabe, 256  
Funktion AKTIVIERT, 144  
Funktion BATTERIELEER, 145  
Funktion COMERROR, 146  
Funktion DATEIVORHANDEN, 147  
Funktion DATUM, 148  
Funktion ERSETZEN, 171  
Funktion Feiertag, 149  
Funktion GESCHALTET, 150  
Funktion GROSSBUCHSTABEN, 172  
Funktion JAHR, 151  
Funktion KLEINBUCHSTABEN, 173  
Funktion LESETEXTPAR, 174  
Funktion LINKERTEIL, 175  
Funktion MAILLOG, 152  
Funktion MONAT, 153  
Funktion MONATSTAG, 154  
Funktion OBJEKTBEZ, 155  
Funktion OBJEKTNAME, 156  
Funktion RECHTERTEIL, 176  
Funktion SCHALTDAUER, 158  
Funktion SELBST, 159  
Funktion SONNENAUFANG, 160  
Funktion SONNENUNTERGANG, 161  
Funktion STOPPUHR, 162, 164  
Funktion TEXTLÄNGE, 177  
Funktion TEXTPOSITION, 178  
Funktion TEXTTEIL, 179  
Funktion TEXTZWISCHEN, 180  
Funktion TRIGGER, 164  
Funktion UHRZEIT, 165  
Funktion UHRZEITSEKUNDE, 166  
Funktion WOCHENTAG, 167  
Funktion ZEIT, 168  
Funktion ZUFALLSZEIT, 169  
Funktionen, 143

## G

GEHEZU, 107  
GESCHALTET, 150  
GETCCUSYSVAR, 108  
GETSITE, 109  
GROSSBUCHSTABEN, 172  
GRUPPENZUWEISUNG, 111

## H

Hauptfenster, 41  
Hinweise zur Dimmersteuerung, 252  
History, 258  
Historydaten, 258  
HM-Dis-EP-WM55, 199, 214  
HomeMatic CCU, 20

## I

Init-Makro, 85  
Installation, 17, 20, 22, 28, 33  
Installation Software, 17  
Installationshinweise, 17, 20, 22, 33

## J

JAHR, 151

## K

KLEINBUCHSTABEN, 173  
Konfigurationsadapter, 63

## L

LAN-Interface, 30  
LESETEXTPAR, 174  
LESEWERTEDATEI, 112  
LINKERTEIL, 175  
Logdatei, 258  
LÖSCHEANZEIGE, 113  
LÖSCHEDATEI, 114

## M

MAILLOG, 152  
Makro, 35, 69, 85  
Makro ausführen, 81  
Makro starten, 82  
Makroanweisungen, 35, 84  
Makroanweisungen über Menü, 77  
Makromenue, 77  
Makros, 85  
Makros erstellen, 69, 77, 85  
Makros über Menü erstellen, 77  
Makrosprache, 35, 85  
Markisensteuerung, 186, 206, 240  
Modulauswahl, 43  
MONAT, 153  
MONATSTAG, 154



MP3 Funkgong mit Signalleuchte, 193, 195

## O

OBJEKTBEZ, 155  
Objektdarstellung, 260, 261  
Objektdefinition, 66  
Objekte, 51, 259  
OBJEKTNAME, 156  
Objektrahmen, 261

## P

PC als HomeMatic Zentrale, 30  
PC-Modus, 28  
PLAY, 115  
Programmiersprache, 35, 84, 262  
Programmierung, 35, 84, 262  
Projekt, 263

## R

Raumthermostat, 188, 203, 238  
Rechenfunktion für Uhrzeiten, 89  
Rechenfunktionen, 89  
Rechenfunktionen für Datum, 89  
Rechenoperationen, 89  
RECHTERTEIL, 176  
Rollladensteuerung, 186, 206

## S

SCHALTDAUER, 158  
Schieberegler, 264  
SCHLIESSEDATEIEN, 116  
SCHREIBEDATEI, 117  
Seite Allgemein, 53  
Seite Allgemein Objekte, 67  
Seite Ansichten, 50  
Seite Makro, 69  
Seite Makros, 52  
Seite Objekte, 51  
Seite PC-Modus, 60  
Seite Sicherheit, 59  
Seite Verzeichnisse, 58  
Seite Visualisierung, 72  
Seite Zeit-Tabelle, 71  
SELBST, 159  
SENDE, 119  
SENDEMAIL, 120  
SENDESMS, 122  
Sensoren, 45, 48  
SETCCUSYSVAR, 124  
SETZEHISTORYDIFFERENZ, 125  
SETZEKONFIG, 126  
SETZEWERT, 127  
SETZEZEITTABELLENWERT, 130  
Sicherheit, 59

Sicherheitshinweise, 33  
SICHTBAR, 131  
Sonnenaufgang, 33, 266  
Sonnenuntergang, 266  
Sonnenzeiten, 266  
Sound-Datei abspielen, 80, 115  
Spezielle Objekte, 267  
Sprachausgabe, 67, 132  
Sprachsteuerung, 67, 132  
STARTE, 133  
STARTPROGRAMM, 134  
STARTUHR, 135  
STARTWIN, 136  
Statusanzeige HM-Dis-WM55, 196  
Steuerungsprogramm, 35, 84  
STOPPE, 137  
STOPPUHR, 162, 164  
Systemlog, 258

## T

TEMPERATURABSENKUNG, 138  
TEXTLANGE, 177  
TEXTPOSITION, 178  
Textrahmen, 260  
TEXTTEIL, 179  
TEXTZWISCHEN, 180  
TRIGGER, 164  
Typdefinitionen, 269

## U

UHRZEIT, 165  
UHRZEITSEKUNDE, 166  
UNSICHTBAR, 139  
USB-Interface, 30

## V

Variable CT, 87  
Variablen, 87  
Vergleichsoperator =\*, 92  
Vergleichsoperator =+, 92  
Vergleichsoperatoren, 92  
Verknüpfungsfunktion für Zeichenketten, 89  
VERLASSEN, 140  
Verzeichnisse, 58  
Visualisierung, 42, 272  
Visualisierungsfenster, 42

## W

WARTE, 141  
warten, 79, 141  
WENN, 142  
Wenn-Bedingungen, 142  
Windows Programm starten, 83, 136  
WOCHENTAG, 167

## Z

Zeilenvorschub, 273

ZEIT, 168

Zeittabelle, 47, 71

Zeit-Tabelle, 47, 71

Zeit-Tabelle für Objekte, 71

Zeit-Vergleiche, 92

ZUFALLSZEIT, 169

Zuweisung, 89

Zuweisungen, 89