

## Visusystem v1.1 Softwaredoku

### Allgemein:

Das Visusystem 1.1 basiert auf einer völlig neuen Struktur der Datendarstellung sowie Funktionsbewältigung.

Im System 1.1 werden alle Bedienvorgänge über den Webbrowser erledigt, der Server selbst enthält nur eine Plattform auf der die einzelnen Tasks die aktuell laufen ersichtlich sind. Zentraler Bestandteil des Systems ist eine Datenbank in der alle Objekte, Parameter und die Gebäudetopologie gespeichert sind.

Als Interface zwischen dem EIB wird das Schlapps+Partner EIB-ASCII Gateway verwendet, da es mit dem OPC Server viele Probleme (Geschwindigkeit, Datenkorrektheit, und das Objektlesen) gab.

Als Interface zwischen dem ACS und dem System wird das ACS-RS232 Gateway mit einer Query Applikation verwendet.

### Leistungsmerkmale des Systems 1.1:

- Kompletter Zugriff auf alle Funktionen und Parameter über einen HTML Webserver
- Automatisches Kopieren von erforderlichen Funktionen insofern sie in der Topologie hinterlegt sind.
- Augenmerk auf automatisches Hochstarten, Datensichern und herunterfahren bei Spannungsausfall des Servers
- Der Server übernimmt alle Zentralen Steuerungsaufgaben, dies ist notwendig um einen reibungslosen zugriff auf die Parameter zu gewährleisten
- Die Funktion des Systems ist ohne lauffähigen Server nicht mehr gewährleistet
- Integration von Videokameras über integrierte Capture Karte und Image Save
- Integration von Audiowiedergabe über Soundkarte und Sound02 Lautstärkereglern

### Grundsätzliche Funktion:

Das System besteht aus zwei Datenbanken, der Topologie und der RecordDataBase.

Die Topologie Datenbank speichert alle Daten die das System kennzeichnen, also Gebäudestruktur, Komm Objekte, Ressourcen des Systems, und auch die Systemparameter welche jederzeit vom Kunden geändert werden können.

Die Record DataBase ist speziell für die Aufzeichnung von Daten (Logging) gedacht, die Datentabellen die dort angelegt sind benutzerspezifisch da jeder Benutzer seine eigenen Aufzeichnungen anlegen kann.

Zentraler Bestandteil des Systems ist die Funktion StartFunktionen.vi, diese Funktion ruft zuerst alle Fixen Funktionen und anschließend alle Temporär generierten Funktionen auf. Alles was zum Betrieb des Systems nötig ist wird in diesem Task vereinigt und gestartet. Wenn also diese Funktion gestartet wird, dann läuft nach ca. 30-60 Sekunden das System komplett (also in die Autostart damit).

### Aufsetzen des Servers:

Aktueller Stand als Server ist ein Shuttle PC der Fa. Shuttle.com, dieser leistungsfähige kompakt PC ist bestens geeignet, beachtet werden muss aber das nur eine PCI Einsteckkarte reinpasst, ansonsten siehe [ipc-markt.de](http://ipc-markt.de).

Als Betriebssystem sind die bestehen Erfahrungen mit Windows XP Professional gemacht worden.

Als erstes ist Also Windows XP Proffessional zu installieren.

Die Netzwerkkonfiguration auf eine fixe IP einrichten und das automatische anmelden aktivieren, alle externen zugriffe werden gesperrt (Remoteedesktop, Freigabe von Dateien).

Beim PC muss noch im BIOS das automatische einschalten nach Netzausfall und „Halt on none“ bezüglich der nichtvorhandenen Tastatur eingestellt werden.

Nachdem Windows XP installiert worden ist muss nun LabView 6.1 mit dem Database Toolkit und dem Internet Toolkit installiert werden, bezüglich der Lizenzierung muss noch abgeklärt werden wie das gehen soll, momentan sind wir nicht legal unterwegs.

Nun müssen noch die Registry Einträge und die Datenbank DNS eingerichtet werden (Topologie und RecordDataBase).

Registry enthält folgende Schlüssel:

Schlüsselname: HKEY\_LOCAL\_MACHINE\SOFTWARE\Kremmel & Schneider\Visualisierung

Name: OPCPfad	Typ: REG_SZ	Daten: EIBKom/
Name: VisuPfad	Typ: REG_SZ	Daten: c:\VisuSys1-1
Name: MailServer	Typ: REG_SZ	Daten: email.aon.at
Name: WebServerAdress	Typ: REG_SZ	Daten: localhost
Name: WebPfad	Typ: REG_SZ	Daten: c:\visusys1-1\WebServer\Visu
Name: EIBLokalAdress	Typ: REG_SZ	Daten: 1.1.33
Name: EIBComPort	Typ: REG_SZ	Daten: 0
Name: ACSComPort	Typ: REG_SZ	Daten: 1
Name: ProduktKey	Typ: REG_SZ	Daten: 42210-3000
Name: KryptedKey	Typ: REG_SZ	Daten: 10112004171
Name: ExpirationDate	Typ: REG_SZ	Daten: 01012004
Name: SMSService	Typ: REG_SZ	Daten: A1
Name: WDT	Typ: REG_SZ	Daten: 60
Name: TaskTime	Typ: REG_SZ	Daten: 100
Name: MailSender	Typ: REG_SZ	Daten: info@sontec.at
Name: MailSubject	Typ: REG_SZ	Daten: Visualisierung, Alarmmeldu

Konfiguration der automatischen Anmeldung unter Windows XP:

Als erstes bitte das Remote tool zur Fernwartung installieren, da diese Programme meistens diese Anmeldevorgänge verändern.

Dann einfach folgendes Kommando ausführen „control userpasswords2“ es erscheint ein Dialog in dem es eine entsprechende Einstellung gibt.

Installation des Fernwartungstools:

Am besten wäre der Windows XP Remotedesktop geeignet, da dieser aber bei Soundwiedergabe nicht funktioniert, wird NetSupport verwendet.

## **Funktionsbeschreibung der FunktionsTypen:**

### **Licht:**

Es wird ein Standard HTML Interface zur Lichtsteuerung geladen, es wird keine VI Funktion geladen.

Durch das HTML Interface kann man nur das Licht ein- und ausschalten bzw. den Status sehen.

Es werden folgende OPC Objekte eingebunden:

#### **Switching**

Der HTML Source Code ist in der Datei [**WebPfad**]\Interface\lichtStandard.html zu finden.

### **DimmLicht:**

Es wird ein Interface zur dimmbaren Beleuchtungssteuerung geladen, es wird keine VI Funktion geladen.

Es werden folgende OPC Objekte benötigt:

#### **Switching DimmingRelativ DimmingValue**

Der HTML Source Code ist in der Datei [**WebPfad**]\Interface\lichtdimmbar.html zu finden.

### **Jalousie:**

Es wird ein Interface zur Jalousiensteuerung geladen, es ist möglich eine absolute Position anzufahren und die Lamellen zu stellen.

Es werden folgende OPC Objekte benötigt:

#### **Step Move PositionStep PositionMove**

Der HTML Code ist in der Datei: [**WebPfad**]\Interface\JalousieStandard.html zu finden

### **JalousieOL:**

Es wird ein Interface zur Jalousiensteuerung geladen, es ist möglich eine absolute Position anzufahren und die Jalousie zu stoppen.

Es werden folgende OPC Objekte benötigt:

**Step**  
**Move**  
**PositionMove**

Der HTML Code ist in der Datei: [WebPfad]\Interface\JalousieOL.html zu finden

### **FussbodenHeizung:**

Es wird ein Interface für eine Fußbodenheizung geladen, außerdem wird eine entsprechende VI Funktion geladen.

Es ist möglich die aktuellen Temperaturen Soll/ Ist / Komfort zu sichten und die Komforttemperatur vorzugeben, außerdem kann die Betriebsart eingestellt werden.

Das Ventil wird angezeigt offen/zu und die Betriebsart heizen / kühlen.

Die Heizung verfügt über eine Uhr mit Wochenprogramm zur Nachtabsenkung.

Vor dem Absenkbetrieb wird der aktuelle Zustand Komfort oder Standby gesichert und dann bei Rückkehr aus dem Nachtbetrieb wiederhergestellt.

Beim einschalten wird die Komforttemperatur und der Status Komfort an den Schalter gesendet.

Beim ausschalten wird der Schalter in Komfortbetrieb versetzt.

Es werden folgende OPC Objekte benötigt:

**HeizungSoll**  
**HeizungIst**  
**HeizungVentilHeizen**  
**HeizungKom**  
**HeizungNacht**  
**HeizungStandby**  
**HeizungFrost**  
**HeizungEinstellungen**

Das Objekt Heizung Einstellungen ist ein Parameter Array und wie folgt zu dekodieren:

<b>Index</b>	<b>Funktion</b>
0-6	Nachtbetrieb Startzeiten von 0=Montag bis 6=Sonntag
7-13	Nachtbetrieb Endzeiten von 0=Montag bis 6= Sonntag
14	Temperatursturzerkennung aktiv
15	Temperatursturz in °C pro min
16	Heizung Komfort Grundtemperatur
17	Heizung automatisch per Außentemperatur auf kühlen
18	Heizung Außentemperatur Grenze
19	Betrieb 1=kühlen 0=heizen
20	Kühlbetrieb Startmonat 0=Jänner

21	Kühlbetrieb Endmonat 0=Jänner

Der HTML Code ist in folgender Datei zu finden: [WebPfad]\Interface\fbheizung.html  
Die VI Funktion ist in der Datei: [VisuPfad]\Funktionen\MainFunktionen\FBHeizung.vi

**Lüfter:**

Eine Lüfter wie er in WCs etc. verwendet wird, es stehen Funktionen wie Nachlauf, Periodisches Lüften und Aktivierung mit Raumlicht zu Verfügung.  
Es werden alle Switching Objekte miteinander verodert, es können also auch mehrere Beleuchtungen verknüpft werden.

Folgende OPC Objekte werden benötigt:

- LueftungParameter**
- LueftungSwitch**
- Switching**

Die Parameter sind wie folgt Codiert:

<b>Index</b>	<b>Funktion</b>
0	Nachlauf aktivieren
1	Nachlaufzeit in Minuten
2	Mit Raumlicht aktivieren
3	Zyklisch einschalten
4	Zykluszeit (der Lüfter wird zyklisch alle xx min für die Nachlaufzeit lang aktiviert)

Der HTML Code ist in der Datei: [WebPfad]\Interface\luefterStandard.html

Die VI Funktion ist in der Datei: [VisuPfad]\Funktionen\MainFunktionen\Lüftung.vi

### **Lichtszene:**

Es können flexibel gestaltete Szene angelegt werden.

Folgende OPC Objekte werden benötigt:

**LichtszeneAufrufen**  
**LichtszeneObjekte - Parameter**  
**LichtszeneWerte - Parameter**  
**LichtszeneWertAktiv - Parameter**

Für nähere Informationen zu Lichtszene System siehe Kapitel Lichtszenen Verwaltung und Management.

Der HTML Code ist in der Datei: [WebPfad]\Interface\Lichtszene.html

Die VI Funktion ist in der Datei: [VisuPfad]\Funktionen\MainFunktionen\Lichtszene.vi

### **Lautsprecher:**

Prinzipiell gleiche Funktion wie ein dimmbares Licht.

Es wird keine VI Funktion geladen

Folgende OPC Objekte werden benötigt:

**AudioSwitch**  
**AudioDimmingRelativ**  
**AudioDimmingValue**

Der HTML Code ist in der Datei: [WebPfad]\Interface\Lautsprecher.html

### **WeckerStandard:**

Es wird eine Wecker Funktion implementiert, welche zu einer eingestellten Zeit einen Lautsprecher einschaltet oder das Licht oder beides.

Es kann die Lichtstärke und Lautstärke eingestellt werden.

Außerdem kann über einen Schalter die Zeit verstellt und der Wecker aktiviert werden.

Die Weckzeit ist über ein OPC Objekt auf dem Bus verfügbar.

Folgende OPC Objekte werden benötigt:

- WeckerZeit** - EIB Objekt Zeit
- WeckerStellen** - EIB Objekt, der Wecker geht um Schrittweite +/-
- WeckerSwitch** - EIB Objekt, der Wecker ein/aus
- DimmingValue** - EIB Objekt das Licht wird eingeschaltet
- AudioDimmingValue**
- WeckerEinstellungen**

Die Einstellungen sind Codiert:

Index	Funktion
0	Wecker Weckzeit
1	Modus 0=NurSound 1=Nur Licht 2=Beides
2	Lautstärke in %
3	Lichtwert in %
4	Schrittweite für Schalter Verstellung in min
5	Audio Quelle 0=Tuner, 1=Stereo, 2=WAV datei
6	Audio File

Der HTML Code ist in der Datei: [WebPfad]\Interface\WeckerStandard.html

Die VI Funktion ist in der Datei: [VisuPfad]\Funktionen\MainFunktionen\Wecker.vi

### **Klingel:**

Eine Klingel die es ermöglicht eine WAV Datei wiederzugeben, und noch vieles mehr.

Folgende OPC Objekte werden benötigt:

- Klingeltaster**
- Tueroeffner**
- Klingelgeraet**
- Klingeinstellungen**
- AudioStatusObjekt**
- AudioDimmingValue**

Das Objekt Klingelgerät wird als Array geführt, das erste Element wird als Standard Klingel verwendet, das zweite wird verwendet falls die Option Klingel auf Handy weiterleiten aktiv ist.

Ebenso werden die Objekte AudioStatus und AudioValue als Array geführt, die Sounddatei wird auf allen hier angelegten Lautsprechern wiedergegeben.

Zuerst wird über das Objekt AudioStatusObjekt geprüft ob der Lautsprecher schon an ist, falls er an ist wird nichts gemacht, falls er aus ist wird er auf den eingestellten Wert gesetzt und am Ende Der Datei wieder ausgeschaltet.

Das Klingelgerät wird intern für die gewisse Zeit ein und ausgeschaltet, es wird also ein Standard Schaltkontakt benötigt.

Die Einstellungen sind wie folgt codiert:

Index	Funktion
0	Klingel aktiv
1	Klingel auf Klingelgerät 2 weiterleiten (Handy)
2	Klingel Dauer in Sekunden
3	Klingel auf Lautsprecher wiedergeben
4	Sounddatei (Grundpfad ist \UserData\MP3)
5	Lautstärke in %
6	Klingel öffnet automatisch die Türe
7	Videobild der Kammera speichern false/true
8	Kamera Funktions ID
9	Kamera Type (ftp für z.b. Convision, pic für FrameGraber Karte und http für GV)

Der HTML Code ist in der Datei: [WebPfad]\Interface\Klingel.html

Die Vi Funktion ist in der Datei: [VisuPfad]\Funktionen\MainFunktionen\Klingel.vi

### **BWM:**

Es wird ein Standard BWM implementiert, welcher außenlichtabhängig gesteuert ist.

Folgende OPC Objekte werden benötigt:

**Switching**  
**Switching**  
**Daemmerungssensor**  
**BWMEinstellungen**  
**BWMOut (es werden alle BWMOut verodert !)**  
**BWMOut**

Das erste Switching Objekt ist die Beleuchtung, das zweite ein Schalter welchem eine flexible Funktion zugeordnet werden kann.

Der Dämmerungssensor kann ein Innen oder Außensensor sein, im HTML heißt er aber Außenlicht.

Das Objekt BWM Out ist das Ausgangsobjekt des BWM, der BWM muss auf Dauerbetrieb gestellt werden und mit der kürzesten Zeit (10s) versehen werden.

Die BWM Telegramme werden automatisch für 12Sekunden nach einem Licht aus Kommando ignoriert um zu vermeiden das der BWM von Licht beeinflusst wird, siehe Parameter 5.

BWMEinstellungen:

Index	Funktion
0	BWM Zeit in Sekunden
1	Außenlichtabhängig aktivieren
2	Außenlichtwert %
3	Betriebsart 0=BWM, 1=Dauer ein, 2=Dauer aus



4	Schalter Funktion
	0= Dauer ein / BWM Zeit
	1= Dauer aus/ BWM Zeit
	2=Licht für BWM Zeit aktivieren und licht ignorieren
5	Optional Sperrzeit in Sekunden nach Aus Telegramm, wenn dieser Parameter Fehlt wird der Wert 12 Sekunden genommen

Der HTML Code ist in der Datei: [WebPfad]\Interface\Bewegungsmelder.html  
 Die VI Funktion ist in der Datei: [VisuPfad]\Funktionen\MainFunktionen\BWM.vi

**Zutritt:**

Es wird ein Zutrittsleser implementiert, welcher über ein ACS Gateway eine Transponder liest und dann eine Tür öffnet.

Für Detail Information siehe bitte Kapitel Zutrittssystem

Folgende OPC Objekte werden benötigt:

- Tueroeffner**
- TPCReader**
- TPCName**
- TPCEinstellungen**

Der Türöffner muss als Impulsausgang programmiert werden, wir geben nur eine 1 bit 1 Impuls raus.

TPCName ist eine Funktion und muss entsprechend in der FunktionSources DB angelegt sein, sie liefert anhand eines Codes den in der Datenbank gespeicherten Namen zurück.

Die Funktion heißt GetTPCName.vi, als Parameter wird lediglich das TPCReader Objekt übergeben.

Die TPC Einstellungen:

<b>Index</b>	<b>Funktion</b>
0	Zutritte Protokollieren
1	Betrieb 0=Zutritt, 1=Zutritt+Zeit, 2=Dauer auf
2	Nicht verwendet
3-9	Türe Auf Zeiten 0=Montag
10-16	Türe zu Zeiten 0=Montag

Sollte die Türe aufgehalten werden durch dauer auf oder die Zeituhr, so werden einfach zyklisch alle 4 Sekunden 1 Impulse gesendet, also entsprechend die Türzeit länger als 4 Sekunden stellen.

Der HTML Code ist in der Datei: [WebPfad]\Interface\Zutritt.html

Die VI Funktion ist in der Datei: [VisuPfad]\Funktionen\MainFunktionen\Zutritt.vi

**JalousieWetter:**

Dies Wetterstation kann einfach alles Sonnenachführung, Lammellenverstellung, temperaturabhängige Beschattung uvm.

Die Raumtemperaturabhängige Aktivierung der Sonnenautomatik besitzt eine Hysterese von 1,5 °C d.h. bei einem Wert von 25 °C muss die Temperatur > 25 °C sein, dann wird die Automatik aktiviert, bei <23,5°C dann wieder deaktiviert.

Über den Parameter 31 kann das zyklische Senden des Sicherheitsposition auf 0 deaktiviert werden normalerweise ist dieses Element Leer, oder false, dann wird bei Sicherheitsalarm alle 5 Sekunden die Position 0 auf den Bus gesendet, bei true wird dies nicht gemacht.

Diese Option kann aber vom Kunden selbst nicht verändert werden.

Die Einstellung „bei Dämmerung schliessen“ schliesst die Jalousien wenn es dunkel wird, geöffnet werden sie aber nicht mehr automatisch. Folgende OPC Objekte werden benötigt:

**WetterEinstellungen**  
**Dämmerungssensor**  
**SonneAzimuth**  
**SonneElevation**  
**Regensensor**  
**Windsensor**  
**PositionMove**  
**PositionStep**  
**HeizungIst**

Dämmerungssensor für die Automatische Steuerung mit Dämmerung

WetterEinstellungen:

Index	Funktion
0	Sonnenautomatik ein/aus
1	Windsicherheit aktiv
2	Windgrenze
3	Regen aktiv
4-10	Sonnenautomatik Aktivierungszeitfenster 0=Montag Beginn
11-17	Sonnenautomatik Aktivierung Ende 0=Montag
18	Azimuth Sonnenfenster Von
19	Azimuth Sonnenfenster bis
20	Elevation Sonnenfenster von
21	Elevation Sonnenfenster bis
22	Bei Dämmerung schließen
23	Dämmerung Grenzwert
24	Sonnennachführung der Lamellen aktivieren
25	Sonnennachführung Modus
	0 Maximum Licht
	1 Min Temperatur
	2 Immer schließen
	3 Immer öffnen
26	Korrekturwinkel in °
27	Temperatur aktiviert Sonnenautomatik
28	Temperatur Grenzwert
29	Wind+Sonne Verzögerungszeit zusätzlich
30	Schließwinkel bei Sonne in %
31	Zyklische senden des Windalarms deaktivieren mit true

Der HTML Code ist in der Datei: [WebPfad]\Interface\JalousieWetter.html

Die VI Funktion ist in der Datei: [VisuPfad]\Funktionen\MainFunktionen\JalousieWetter.vi

## **JalousieWetterOL:**

Diese Wetterstation bietet die selben Funktionalitäten wie die Standard Funktion jedoch ohne Lamellen also ohne ein PositionStep objekt und somit ohne Sonnennachführung.  
Bei einem Sicherheitsalarm wird hier die Position mit 0 zyklisch alle 5 Sekunden an die Aktoren gesendet, dies kann aber über den Parameter 31 ausgeschalten werden.  
Bei Der Temperaturabhängigen Aktivierung der Sonnenautomatik wird eine Hysterese von 1,5 °C verwendet.  
Bei der Option schließen bei Dämmerung wird die Jalousie nur geschlossen, nicht aber wieder geöffnet.

Zu beachten ist aber das der Parameter String genau gleich Codiert ist !

HTML Code: [WebPfad]\Interface\JalousieWetterOL.html  
Vi Funktion: [VisuPfad]\Funktionen\MainFunkzione\JalousieWetterOL.vi

Als besonders für beide Wetterstation gilt das sie noch jeweils ein Menü HTML Code haben, dieser lautet JalousieWetterOptions...

## **KuehlenHeizen:**

**Dieses Element muss noch fertig entwickelt werden, die VI Funktion fehlt noch !**

## **ZentralHeizung:**

Es wird benötigt um eine Zentrale Heizung/ Kühlungsanlage anzusteuern, es werden zwei Meldungen generiert einmal Heizung Energiebedarf und Betriebsart.  
Diese Meldungen werden mit den Einzelraumreglern verknüpft.  
Gedacht ist das Ganze um der Heizung einen Kontakt heizen und einen zweiten kühlen zu schalten, wenn keiner der beiden ein ist, dann wird die Heizung ausgeschaltet.  
Die Betriebsart Kontakte sind für die Schalter gedacht, einen Inversen gibt es für die Tritons da die 3er mit RTR verkehrt zu den 5ern sind.  
Der Energiebedarf ist zeitverzögert mit einer Ausschaltzeit von 5s gesteuert.

Es werden folgende OPC Objekte benötigt:

<b>HeizungEnergiebedarf</b>	<b>(heizen)</b>
<b>HeizungEnergiebedarf</b>	<b>(kühlen)</b>
<b>HeizungBetriebsart</b>	<b>(5er triton)</b>
<b>HeizungBetriebsartInvers</b>	<b>(3er triton)</b>
<b>HeizungEinstellungen</b>	
<b>HeizungVentilHeizen</b>	
<b>HeizungVentilHeizen</b>	
<b>Temperatursensor</b>	

Das 1.Energiebedarf Objekt wird für heizen genommen das zweite für kühlen

Die Objekte Heizung Energiebedarf und Betriebsart werden zyklisch auf den Bus gesendet 10Sekunden.

Die Objekte HeizungVentil werden alle miteinander Oder verknüpft, sobald eines 1 ist wird über das Objekt HeizungEnergiebedarf=1 Energie angefordert.

HeizungEinstellungen in diesem Fall:

Index	Funktion
0	Energiebedarf Methode
	0=Automatisch über RTR ermitteln
	1=Dauerbetrieb
	2=Keine Energie dauernd
1	Umschaltmethode heizen / kühlen
	0=Automatisch über Außentemperatur
	1=Dauernd heizen
	2=dauernd kühlen
2	Außentemperatur Grenze
3	Startmonat Kühlperiode 0=Jänner
4	Endmonat Kühlperiode 0= Jänner
5	Energiebedarf Mindestwert
6	Energiebedarf Aktueller Wert wird alle 30s neu berechnet
7	Energiebedarf aktuell true/false
8	Zeitverzögerung Energiebedarf aus in s
9	Frei
10-xx	Jedes Element wird einem OPC Objekt Ventil zugeordnet für Raumwertigkeit

Es muss die Außentemperatur größer als der eingestellte Wert sein und der Aktuelle Monat muss Innerhalb des Fensters liegen damit in den Kühlbetrieb gewechselt wird, ansonsten ist immer Heizbetrieb.

Der Energiebedarf aktuell wird anhand aller Ventile im Gebäude berechnet, dies wird alle 30 Sekunden gemacht und in % dargestellt, der Energiebedarf muss größer als der Grenzwert sein damit Energie bei der Heizung angefordert wird.

Die Berechnung erfolgt mit einer Fallverzögerung von 30Sekunden, das heisst wird der E-Bedarf wert höher, so erfolgt das sofort, fällt er so muss der Wert 30 Sekunden niedrig sein bevor er übernommen wird.

Wertigkeiten für die Energiebedarfsermittlung:

Jedem Raum (also jedem Ventil) kann eine Wertigkeit vergeben werden, einfach ins Array ab 10 eingeben, diese Werte müssen zwingend eingegeben werden, es können hier direkt die Flächenmaße eingegeben werden oder auch Punkte etc.

Der E-Bedarf wird dann nach Gesamtpunktesumme zu aktueller Punktesumme berechnet.

Der HTML Code ist in er Datei: [WebPfad]\Interface\ZentralHeizung.html

Die VI Funktion ist in der Datei: [VisuPfad]\Funktionen\MainFunktionen\ZentralHeizung.vi

### **ZentralLueftung:**

Die Zentral Lüftung ist gedacht um kontrollierte Wohnraum Be und Entlüftungsgeräte anzusteuern.

Es können 4 Stufen über potentialfreie Kontakte gewählt werden.

Die Funktion verfügt über ein Zeitprogramm in dem jeweils zu bestimmten Zeitpunkten bestimmte Stufen geschaltet werden können.

Folgende OPC Objekte werden benötigt:

**LueftungParameter**  
**LueftungSwitch**  
**LueftungSwitch**  
**LueftungSwitch**

Die 3 Switch Objekte werden gereiht und sind gemäss des Stiebel Eltrons Netzteiles gesteuert

Object 1 = Off

Object 2 = Day

Objekt 3 = Party

Kein Objekt = Stufe 1

LueftungParameter in diesem Fall:

<b>Index</b>	<b>Funktion</b>
0-6	Umschaltzeitpunkt 1 0=Montag
7-13	Umschaltzeitpunkt2 0=Montag
14	Betriebsart aktuell 0-3
15	Betriebsart Zeitpunkt1 0-3
16	Betriebsart Zeitpunkt2 0-3
17	Betriebsstundenzähler in Minuten
18	Schalter Funktion 0=Stufe Auf/Ab 1=Stufe0 / Lüftung Auf Fixe Stufe 2=Lüftung für Zeit auf Fixe Stufe danach wieder auf vorherigen Wert
19	Schalter Stufe
20	Schalter Zeit (min)
21-27	Umschaltzeitpunkte 3
28-34	Umschaltzeitpunkte 4
35	Betriebsart Zeitpunkt 3
36	Betriebsart Zeitpunkt 4

Der HTML Code ist in er Datei: [WebPfad]\Interface\ZentralLüftung.html

Die Vi Funktion ist in der Datei: [VisuPfad]\Funktionen\MainFunktionen\ZentralLüftung.vi

### **AudioSelect:**

Die Audio select Funktion dient dazu einen Audioumschalter anzusteuern, es gibt 3 Quellen Tuner, Stereoanlage und PC diese sind mit Namen fix eingegeben. Dazu gibt es 3 Objekte EIS1 mit dem Typ AudioSelectSource. Die Objekte werden über eine Funktion gesteuert welche ständig einen Parameter überwacht, der Parameter ist vom Typ AudioEinstellungen und kennt 4 Möglichkeiten:

- 0=Mute
- 1=Tuner
- 2=Stereoanlage
- 3=PC-Soundkarte

Folgende OPC Objekte werden benötigt:

- AudioSelectSource**
- AudioSelectSource**
- AudioSelectSource**
- AudioEinstellungen**

AudioEinstellungen kennt nur das Array Element 0.

Der HTML Code ist in der Datei: [WebPfad]\Interface\AudioSelect.html  
Die VI Funktion: [VisuPfad]\Funktionen\MainFunktionen\AudioSelectSource.vi

### **SonnenPosition:**

#### **Windsensor**

#### **Sonnensensor**

#### **Regensensor**

#### **Daemmerungssensor**

#### **Temperaturssensor:**

Diese Funktionen dienen dazu um die benötigten Werte mit Einheit anzuzeigen, außerdem ist es wichtig damit diese Werte auch aufgezeichnet werden können!

Als Objekte benötigt jede Funktion sein gleichnamiges OPC Item vom Typ und Sonnen Position benötigt SonneAzimuth und SonneElevation.

HTML Codes:

<b>HTMLInterface</b>
[WebPfad]\Interface\Daemmerungssensor.html
[WebPfad]\Interface\Temperaturssensor.html
[WebPfad]\Interface\Windsensor.html
[WebPfad]\Interface\Sonnensensor.html
[WebPfad]\Interface\SonnenPosition.html
[WebPfad]\Interface\Regensensor.html

Die Funktionen besitzen keine Vi Funktion.



### **VideoCap:**

Diese Funktion lädt alle 800ms ein Picture neu vom Server und zeigt es an, der Komplette Pfad des Pictures muss in dem Parameter vom Type VideoEinstellungen enthalten sein, Achtung kein Array.

Beispiel für die Verwendung mit GrandTec: ../Video/DCAMImage.bmp

Die Funktion fügt dem Pfad automatisch eine id hinzu um bei IE4 zu verhindern dass das Bild gecacht wird.

HTML Code: [WebPfad]\Interface\VideoCam.html

### **VideoConvision:**

Diese Funktion benutzt einen Convision Video Web Server zur Bildanzeige, sie verwendet ein JavaApplet welches im Browser geladen wird.

Bei der Convision muss nur der FTP Server für anonym aktiviert werden, sonst sind keine Einstellungen nötig.

Es muss ein VideoEinstellungen Parameter enthalten, welches folgende Einstellungen ermöglicht:

VideoEinstellungen:

Index	Funktion
0	IP-Adresse des Servers 192.168.0.15
1	Kamera Nummer Wert von 1 bis 6
2	Farbsättigung von 0 bis 255
3	Kontrast von 0 bis 255
4	Helligkeit von 0 bis 255
5	FTP-Server Username (ftp)
6	FTP-Server Userpasswort (1234)
7	WAN Adresse des Anschlusses für Videobild von extern

HTML Code: [WebPfad]\Interface\VideoConvision.html

Es wird außerdem eine Options- HTML Seite geöffnet in dem die Einstellungen für Helligkeit, Kontrast und Farbsättigung geändert werden können

HTML Code: [WebPfad]\Interface\VideoConvisionOptions.html

## AlarmSensor:

Es wird ein Alarmsensor implementiert, welcher zyklisch einen Wert empfangen muss, empfängt er keinen Wert, so geht der Sensor auf Störung.

Das Sabotage Objekt meldet 0 bei Schleife geschlossen und eins bei offen, wenn dieses Objekt vorhanden ist wird es mit in den Parameter übernommen, ansonsten nicht.

Bei Systemstart werden die Parameter Sabotage und Alarm auf false gesetzt, so das nicht durch einen Systemstopp bei scharfem Sensor dieser danach automatisch wieder aktiv ist.

Folgende OPC Objekte werden benötigt:

**AlarmSensor**  
**AlarmEinstellungen**  
**AlarmSabotage (muss aber nicht sein)**

## **Der Parameter Alarm Einstellungen wird wie folgt aufgeschlüsselt:**

Index	Funktion
0	Sabotage true/false true=Sabotiert
1	Überwachungszeit 0=aus, in Sekunden
2	Alarm geben true=Alarm, false=Kein Alarm

HTML Code: [WebPfad]\Visu\Interface\AlarmSensor.html

Vi Funktion: [VisuPfad]\Funktionen\MainFunktionen\AlarmSensor.html

## AlarmAnlage1:

Die Alamanlage dient zur Überwachung eines Gebäudes, ihr werden Objekte Voralarm, Hauptalarm, aktivierung, usw übergeben und ein ParameterArray in dem sämtliche Alarmsensoren per FunktionsID aufgelistet sind.

Es werden folgende OPC Objekte benötigt:

<b>AlarmScharfSchalten</b>	<b>1=Scharf, 0=unscharf</b>
<b>AlarmToggeln</b>	<b>jedes Telegramm toggelt</b>
<b>AlarmMeldung</b>	<b>1=Scharf, 0=unscharf</b>
<b>AlarmEinstellungen</b>	
<b>AlarmVoralarm</b>	<b>Voralarm Signal</b>
<b>AlarmHauptalarm</b>	<b>Hauptalarm</b>
<b>Switching</b>	<b>Beleuchtungsobjekt 0</b>
<b>Switching</b>	<b>Beleuchtungsobjekt 1</b>

Die AlarmEinstellungen sind wie folgt aufgeschlüsselt:

Index	Funktion
0	Voralarm Dauer ( Sekunden )
1	Voralarm nach Voralarm Dauer deaktivieren
2	Licht bei Alarm einschalten
3	Licht bei Alarm Ende ausschalten
4	Anlage automatisch aktivieren
5	Automatik Aktivierungszeit 22:00
6	Anlage automatisch deaktivieren
7	Automatik Deaktivierungszeit 08:00
8	Audiodatei abspielen
9	Audio Datei Filename
10	Audio Lautstärke
11	
12-30	Reserviert
31	Sensor Funktions ID 1 (65)
32	Sensor Funktions ID 1 ist aktiv im Alarmschema (true)

Es können ab dem Index 31 beliebig viele Sensor Elemente Eingefügt werden, sie werden immer in der Form als Zahl und im Nachfolgenden Element die aktivierung true oder false gespeichert.

Also jeder Gerade Index ist die Aktivierung und jeder Ungerade Index ist die Funktions ID.

Die Index 11-30 sind für Zukünftige Erweiterungen der Alarmanlage reserviert.

Die Alarmanlage protokolliert folgende Aktivitäten:

- Sabotage immer auch wenn unscharf
- Sensoren auslösung wenn Anlage scharf
- Anlage aktivierung/deaktivierung
- Anlage Vor- Hauptalarm

Der Sabotagealarm lässt sich nicht deaktivieren, sobald etwas sabotiert wird, gibt es einen Alarm, egal ob die Anlage scharf oder unscharf ist.

Bei Systemstart werden der Melde , Hauptalarm und Voralarm Ausgang auf false gesetzt.

Der Alarm bleibt solange bestehen bis die Anlage deaktiviert wird, die Beseitigung des Alarms reicht nicht.

Die Anlage kann über ein Toggel oder direkt Objekt aktiviert bzw. deaktiviert werden, ob die Anlage auch wirklich scharf ist muss Anhand des melde Objektes überprüft werden.

Die Anlage lässt sich bei ausgelösten Sensoren nicht scharfschalten.

Die Audio Wiedergabe wurde zwar in der Konfiguration vorgesehen, im VI fehlt aber noch alles erforderliche.

HTML Code: [WebPfad]\Visu\Interface\AlarmAnlage1.html

Vi Funktion: [VisuPfad]\Funktionen\MainFunktionen\AlarmAnlage1.vi

**ZeitUhrDaemmerung:**

Diese Funktion wird verwendet um eine Beleuchtung etc. über eine Zeitprogramm in Kombination mit einem Dämmerungsschalter zu schalten.

Es benötigt eine ZeitEinstellungen OPC Objekt für die Einstellungen und ein Dämmerungsobjekt für den Dämmerungsschalter.

Als Schaltobjekt wird ein Switching Typ benötigt.

Es wird mit den Schaltpunkten in true oder false gesetzt.

Folgende OPC Objekte werden benötigt:

ZeitEinstellungen  
 Daemmerungssensor  
 Switching

ZeitEinstellungen:

Index	Funktion
0	Dämmerungs Grenzwert
1	Betriebsart (0=Auto, 1=Dauer ein, 2=Dauer aus)
2-8	Zeiten 1 ein
9-15	Zeiten 1 aus
16-22	Zeiten 2 ein
23-29	Zeiten 2 aus

HTML Code: [WebPfad]\Visu\Interface\ZeitUhrDaemmerung.html

Vi Funktion: [VisuPfad]\Funktionen\MainFunktionen\ZeitUhrDaemmerung.html

## HeizungPIDRegler:

Diese Funktion bildet die Komplette Heizungsregelung im PC nach, es wird ein Standard Heizung RTR verwendet, aber das Ventil wird nicht auf dem Thermostat programmiert. Es stehen zahlreiche Parameter zu Verfügung.

Die Frostschutz bzw. Hitzeschutz Temperatur kann nicht verändert werden, sowie die Anhebung beim Kühlbetrieb.

Der Regler kann sowohl kühlen wie auch heizen, allerdings sind die Parameter immer für beide Funktionen gleich.

Der Regler setzt folgende Einstellungen Am RTR voraus:

- Absenkung Standby Betrieb 2K
- Absenkung Nachtbetrieb 4K
- Anhebung Totzone Kühlen 2K
- Frostschutz 7°C
- Hitzeschutz 44°C
- Standard Objekte für Komfort/Standby/Nacht/Frost
- Standard Objekte für Basis/Soll/Ist wert
- Bei Gira die Sollwertverschiebung dauerhaft übernehmen

Der Regler ist ein PID Regler, welcher mit entsprechenden Einstellungen auch als P, PI, PD Regler verwendet werden kann.

P-Anteil	D-Anteil	I-Anteil	Funktion	
P	0	INF	P-Regler	
P	D	INF	PD-Regler	
P	0	I	PI-Regler	
P	D	I	PID-Regler	
1	10s	150min	Standardeinstellung für Fussbodenheizung	

Der Regler selbst besitzt keinen Ausgang, er speichert den Wert der Stellgröße (0-100%) im Parameter Array, die Abtastzeit beträgt hierbei 30 Sekunden.

Wenn der Par 27 auf true ist dann wird die Stellgröße direkt an ein Proportional Ventil ausgegeben.

Sollten Schaltantriebe zum Einsatz kommen so muss noch eine PWMSteuerung hinzugefügt werden.

Folgende OPC Objekte werden benötigt:

- HeizungIst**
- HeizungSoll**
- HeizungKom**
- HeizungStandby**
- HeizungNacht**
- HeizungFrost**
- HeizungEinstellungen**
- HeizungBetriebsart**
- HeizungVentilHeizen (eventuell, falls Proportional Ventil)**

## HeizungEinstellungen:

Index	Funktion	
0-6	Nachtbetrieb Startzeit, wobei 0=Mo	
7-13	Nachtbetrieb Ende, wobei 0=Mo	
14	(nicht verwendet)	
15	Temperaturanhebung Kühlbetrieb	
16	Komforttemperatur	
17	Absenkung Standbybetrieb	
18	Absenkung Nachtbetrieb	
19	Betriebsart aktuell 0=Komfort/Standby/Nacht/Frost	
20	Stellgröße min	
21	Stellgröße max	
22	Reglerbereich, für die Normierung (N=Soll-Ist/Regelbereich)	
23	Betriebsart 0=heizen/1=kühlen	
24	P-Anteil	
25	I Anteil in min	
26	Stellgröße aktuell	
27	Stetige Ventile verwenden und Sollwert direkt auf VentilHeizen ausgeben	
28	D-Anteil in sec	

Beim Programmstart werden folgende Aktionen durchgeführt:

- Komforttemperatur senden
- Betrieb auf heizen
- Betriebsart auf Komfort

Die Stellgröße kann Skaliert werden mit Stellgröße min und Stellgröße max, die Stellgröße bewegt sich immer innerhalb dieser zwei Werte.

Die Nachtabsenkung wird nur aktiviert wenn der Schalter nicht schon im Frost Betrieb ist.

Bei Änderung der Komforttemperatur, Standby Absenkung, Nach Absenkung oder Betriebsart werden die Werte automatisch sofort neu an den Schalter gesendet.

Auf die OPC Objekte HeizungBetriebsart und HeizungVentilHeizen kann verzichtet werden, dann gibt es aber keinen Kühlbetrieb !

Die Anhebung Kühlbetrieb kann über Parameter 15 eingegeben werden, im Kühlbetrieb wird der Sollwert um diesen Wert angehoben.

HTML Code: [WebPfad]\Visu\Interface\HeizungPIRegler.html

Vi Funktion: [VisuPfad]\Funktionen\MainFunktionen\HeizungPIRegler.vi

### HeizungPWMSteuerung:

Diese Funktion übernimmt die Ansteuerung eines Schaltenden Ventils mit einem PWM Signal, so wird eine quasi stetiges Ventil realisiert.

Gedacht ist diese Funktion zur Verwendung in Kombination mit dem HeizungPIRegler.

Diese Funktion übernimmt direkt aus dem Parameter 26 denn Sollwert der Stellgröße und regelt anhand dessen sein PWM Signal.

Außerdem stehen Funktionen wie Ventilschutz (um 1:00 Nachts für 10 min auf) und Ventil dauer auf/zu zu Verfügung.

OPC Objekte:

- HeizungEinstellungen
- HeizungVentilHeizen

Die Parameter vor dem Index 30 werden in der Heizungsregelung verwendet !

Index	Funktion
.....	
30	Zykluszeit (min)
31	Ventil Öffnungsdauer bei 1% (min)
32	Ventil Öffnungsdauer bei 100% (min)
33	Modus 0=Auto/1=Dauer Auf/2=Dauer Zu
34	Ventilschutz aktiv ja/nein

Das Signal setzt sich folgendermassen zusammen:

- Das Ventil wird alle Zykluszeit für die berechnete Öffnungsdauer geöffnet
- Sollte der Modus auf Dauer aus sein, so wird auch der Ventilschutz abgeschaltet
- Beim Programmende wird das Ventil geöffnet
- Beim Programmstart geschieht nichts

HTML Code: [WebPfad]\Visu\Interface\HeizungPWMSteuerung.html

Vi Funktion: [VisuPfad]\Funktionen\MainFunktionen\HeizungPWMSteuerung.html

### HeizungStetigSteuerung:

Selbe Funktion wie HeizungPWMSteuerung, jedoch wird kein Schalt Ventil angesteuert, sondern ein direktes Proportionalventil, die Parameter liegen auch etwas anders, nämlich nur 30,31:

Index	Funktion
30	Ventilschutz aktiv ja/nein



31	Modus 0=Auto/1=Dauer Auf/2=Dauer Zu	
----	-------------------------------------	--

### FussbodenHeizungGira:

Selbe Funktion wie die Standard FBHeizung, jedoch wird die Betriebsart mit 4 Objekten, für jede Betriebsart eines, umgeschaltet.

Deshalb gibt es einen neuen OPC Type HeizungKomfort ist das Komfort Objekt.

Außerdem wird der Parameter 15 der Heizung Einstellungen nicht mehr für die Temperatursturzerkennung, sondern für die Betriebsart verwendet.

Diese Funktion bietet außerdem die Möglichkeit die Temperaturanhebung im Kühlbetrieb flexibel einzustellen, d.h. wenn der Befehl zum umschalten kommt wird die Komforttemp.

Automatisch um den Eingestellten Wert angehoben.

Vorausgesetzt ist allerdings dass die Anhebung im Schalter auf 2K eingestellt ist .

OPC Objekte:

**HeizungSoll**

**HeizungIst**

**HeizungVentilHeizen**

**HeizungKom**

**HeizungNacht**

**HeizungStandby**

**HeizungFrost**

**HeizungEinstellungen**

**HeizungBetriebsart** Wobei Heizung Betriebsart 0=kühlen, 1=heizen

Index	Funktion
0-6	Nachtbetrieb Startzeiten von 0=Montag bis 6=Sonntag
7-13	Nachtbetrieb Endzeiten von 0=Montag bis 6= Sonntag
14	Anhebung des Kühlbetriebes in °C
15	Betriebsart Komfort / Standby / Nacht / Frost
16	Heizung Komfort Grundtemperatur

VI-Funktion: FBHeizungGira.vi

HTML Funktion: FBHeizungGira.html

### ZeitUhrWoche:

Eigentlich genau gleich wie die ZeitUhrDämmerung, aber eben ohne Dämmerung, die Parameter sind genau gleich aufgebaut.

Sie schaltet ein und aus und ein..., es ist ein Problem bekannt wenn man als Start und Endzeit einfach 0:0 eingibt dann schaltet das Programm den Verbraucher dauerhaft ein.

OPC Objekte:

- Switching
- ZeitEinstellungen

Beide Objekte werden zwingend benötigt !

ZeitEinstellungen:

Index	Funktion
0	Nicht verwendet (frei)
1	Betriebsart (0=Auto, 1=Dauer ein, 2=Dauer aus)
2-8	Zeiten 1 ein
9-15	Zeiten 1 aus
16-22	Zeiten 2 ein
23-29	Zeiten 2 aus

HTML Code: [WebPfad]\Interface\ZeitUhrWoche.html

Vi Funktion: [VisuPfad]\Funktionen\MainFunktionen\ZeitUhrWoche.vi

### **CSL1A:**

Codeschlossfunktion für ACS Codeschloss v1.0, diese Funktion benutzt direkt ACS Programmierbefehle um den Speicher im CSL umzuprogrammieren.  
Es werden intern in einer Datenbank alle Codes gespeichert, sollte sich durch Benutzereingabe ein Code ändern, dann wird er neu ins EEPROM programmiert.  
Dazu ist das CGI etwa 1-5 Sekunden beschäftigt und legt solange komplett den Webserver lahm, da eine Schleife im selben Task durchlaufen wird

OPC Objekte:

- CSLEinstellungen

CSL Einstellungen für das Codeschloss:

Index	Funktion
0	Code Nummer 1 mit Raute am Schluss ! (1#)
1	Code Nummer 2 mit Raute am Schluss ! (2#)
2	Code Nummer 3 mit Raute am Schluss ! (3#)
3	Beschreibung Code 1 als Text für den User
4	Beschreibung Code 2 als Text für den User
5	Beschreibung Code 3 als Text für den User
6	DID Adresse des CSL Modules

HTML Interface: [WebPfad]\Interface\Codeschloss.html

**UserWebInterface:**

Das User Web Interface stellt eine Möglichkeit dar Kunden spezifische Web Interfaces zu erstellen.

Die Interfaces müssen im Ordner [UserData]\Web platziert sein, und müssen genau gleich wie der Funktions- Name lauten.

Es können alle OPC Items replaziert werden, da zuerst der HTML Code geladen und dann sämtlich OPC Objekte usw. ersetzt werden.

Hier ein Beispiel Code:

(einfach eine Anzeige eines LueftungSwitching Objektes als 0-100%)

```
<table cellpadding="0" cellspacing="0" >
<tr>
<td class="BlindElement"></td>
<td class="NameTableStyle">
<p class="ElementName">L&uuml;ftung Zentralger&auml;t Leistung (0-100):</p>
</td>
<td class="StatusTableStyle">
<input type="text" size="3" name="LueftungSwitch" class="StatusBox" readonly>
</td>
</tr>
</table>

<script language="JavaScript">
<!--

//Im Kommentar die OPC Objekte die automatisch zu updaten sind definieren
//OPCReloadItem=LueftungSwitch%2Ddigits%2F0;

//-->
</script>
```

Voraussetzung für die Korrekte Funktionsweise ist natürlich das mit der Funktion Das OPC Item vom Typ LueftungSwitch verknüpft ist.

Momentan kann dies mit der Steup Oberfläche noch nicht gemacht werden, sondern muss per Access DB erledigt werden.

HTML-Interface: [WebPfad]\Interface\UserWeb.html

## **JalPositioner:**

Diese Funktion wird benötigt wenn Aktoren ohne Positionierung eingesetzt werden. Über Timer usw. wird intern immer die Position der Jalousie berechnet, dazu müssen alle Move und Step Objekte die Der Aktor empfängt auch mit dieser Funktion verknüpft bzw. die EIB Objekte korrekt verlinkt werden.

Die Laufzeit der Jalousie wird dann in einen Parametersatz in Sekunden eingegeben, außerdem wird sobald sich die Position der Jalousie um 2% ändert der Wert in die Parameter DB gesichert.

Dies dient dazu dass die Position wenn möglich auch nach neustart etc. immer erhalten bleibt. Sollte die Position verloren gehen so ist einfach eine Fahrt nach 100 oder 0 % nötig um die Funktion wieder in den Rhythmus zu bringen.

Generell entspricht 100% ganz geschlossen bzw. der unteren Endlage und 0 % ganz offen bzw. der oberen Endlage.

OPC-Objekte:

JalousieEinstellungen  
 PositionMove  
 Move 1-10 Objekte  
 Step 1-10 Objekte

Es werden zwar alle 10 Move bzw. Step Objekte überwacht, aber gesendet wird auf Objekt 0, also auf das jeweils erste.

Jalousie Einstellungen:

Index	Funktion
0	Laufzeit der Jalousie gesamt in Sekunden
1	Sicherung der Aktuellen Position in %

HTML Interface: [WebPfad]\Interface\JalPosSystemPositioner.html

VI-Funktion: [VisuPfad]\Funktionen\MainFunktionen\JalPosSystemPositioner.vi

## VideoGVSystem:

Diese Funktion wird in Zusammenhang mit einer GV250 Video Karte, welche im Server oder separat auf einem externen Server platziert ist kommuniziert.

Die GV250 kann auch Videobilder bei Bewegung Aufzeichnen im avi Format, derzeit gibt es noch keine Oberfläche um diese AVI Files anzusehen, aber bald wird es eine geben.

Um diese GV250 im Zusammenhang mit der Klingel verwenden zu können muss lediglich der Typ http in den Klingeleinstellungen angegeben werden, dann kontaktiert die Video Funktion den Webserver mit <http://IP/camx.jpg>, und lädt das Bild so per http herunter, allerdings wird das 2mal gemacht, da das erste Bild gecachter müll ist.

Die Vi Funktion überwacht das Verzeichnis in dem die Aufzeichnungen gespeichert werden, es werden zu kleine Files gelöscht und der gesamte Speicher des Verzeichnisses begrenzt.

Wenn der Ordner zu groß wird, dann werden die ältesten Aufzeichnungen gelöscht.

Außerdem wird automatisch jede Minute ein File Namens CamList.txt im Ordner erzeugt, in diesem File wird tabellarisch die LV Time + Komplettem Dateipfad aller Aufzeichnungen gelistet.

Es muss ein VideoEinstellungen Parameter enthalten, welches folgende Einstellungen ermöglicht:

VideoEinstellungen:

Index	Funktion
0	IP-Adresse des Servers 192.168.0.15
1	Kammera Nummer Wert von 0 bis 15
2	Nicht verwendet
3	Nicht verwendet
4	Nicht verwendet
5	Nicht verwendet
6	Nicht verwendet
7	WAN Adresse des Anschlusses für Videobild von extern
8	Record Pfad C:\temp\cam01 ist default
9	Maximale Record Größe des Pfades in MB (500MB)
10	Auto Deletee File, Dateien die kleiner als die Angegeben Größe in KB sind, werden Automatisch gelöscht (100KB)

Wichtig zu beachten ist auch das die Aufzeichnungen direkt vom Apache Server an den Client gesendet werden, dazu wird ein Alias verwendet:

Alias /GV250Records c:\temp

Dieser Alias sorgt dafür das das Aufzeichnungsverzeichnis für den Client sichtbar wird, es muss allerdings auf die Einstellungen angepasst werden.

Außerdem sorgt dies dafür das die Kammernummer mit cam0x immer stimmen muss, also Dieser Teil des Verzeichnisses muss so heißen, sonst funktioniertes nicht.

HTML Code: [WebPfad]\Interface\VideoGVSystem.html

VI Funktion: [VisuPFad]\Funktionen\MainFunktionen\VideoGV250.vi

## FensterAutomatik:

Die Fenster Automatik ist für Fenstersteuerung konzipiert worden.

Es kann eine Temperaturautoantik aktiviert werden, welche Zeitgesteuert für eine Gewisse Zeit das Fenster Öffnet, diese Automatik ist mit einer fixen Hysterese von 1,5°C versehen.

Außerdem kann noch eine Absolute Position vorgegeben werden auf die Das Fenster Maximal öffnet.

Auserdem kann ein Schalter gesteuert werden.

Die Funktion verfügt über Sicherheitsalarme Wind, Regen und Sperren über Zeit.

Es besteht die Möglichkeit mehrere PositionMove Objekte mit der Funktion zu Verknüpfen, die Funktion sendet dann an Alle Objekte.

Das PositionMove Objekt ist folgendermassen definiert:

0% = Fenster ganz Offen  
100% = Fenster ganz geschlossen

Prinzipiell vom Gedankengang gleich wie bei einer Jalousie.

Die Eingaben die Der Benutzer Tätigt sind aber genau invertiert zu sehen.

Für den Benutzer spricht man immer vom Öffnungsgrad also sind 100% Offen ganz auf und 0% Offen ganz geschlossen !

Die Stufenöffnung ermöglich das kontinuierliche Öffnen des Fensters, es wird das Fenster zuerst auf den Öffnungswinkel gefahren, und dann alle Zeiteinheiten um die % mehr geöffnet, bis der Maximale Öffnungsgrad erreicht ist.

OPC-Objekte:

- Regensensor
- Windsensor
- Aussentemperatur
- WetterEinstellungen
- Raumtemperatur
- PositionMove
- Switching

Wetter Einstellungen Parameter:

Index	Default	Funktion
0	false	Sicherheit Windsensor aktiv
1	60	Sicherheit Grenzwert Windsensor
2	false	Sicherheit Regen Aktiv
3	false	Sicherheit Außentemperatur aktiv
4	10	Sicherheit Außentemperatur Grenzwert
5	0:0	Zeitpunkt Fenster Sperren start
6	0:0	Zeitpunkt Fenster Sperren Ende

7	1	Verzögerung Sicherheit in Minuten
8-14		Automatik Zeitpunkte Ein (0=Montag)
15-21		Automatik Zeitpunkte Aus (0=Montag)
22	false	Automatik aktiv
23	22	Innentemperatur Grenzwert
24	180	Maximale Öffnungsdauer in Minuten
25	0	Schalter Funktion 0=Fenster Auf/Zu 1=Fenster Auf Pos und für Zeit Öffnen 2=Notöffnung bei Sicherheit auf Pos für Zeit
26	10	Schalter Zeit in Minuten
27	100	Schalter Position in Minuten 100% = Ganz auf, 0% =Ganz Zu
28	false	Zyklisches Senden der Sicherheitsalarme mit true disable
29	100	Öffnung des Fensters bei Sonnenautomatik 0-100%
30	False	Stufenöffnung aktivieren
31	30	Öffnungszeitspanne in Min
32	10	Öffnungswinkel in %

HTML-Dokument: FensterAutomatik.html  
FensterAutomatikOptions.html

Vi-Funktion: FensterAutomatik.vi

### AudioSound03:

Diese Funktion ist speziell für den Sound03 entwickelt worden, sie ermöglicht die Komplette Bedienung des Sound03 über die Visu, also Lautstärke, Quellenwahl, Höhen und Tiefen. Ein Problem stellt noch die Nicht Speicherung der Quelle, Höhen und Tiefen dar, Sobald dass System neu gestartet ist und der Sound03 nicht sind im System wieder die Default Werte und im Sound03 noch die Alten Werte.

Falls dies in der Praxis zum Problem werden sollte, müsste man eine Funktion Programmieren Welche die Werte speichert.

OPCObjekte:

- AudioDimmingRelative
- AudioDimmingValue
- AudioSwitch
- AudioBass
- AudioTreble
- AudioSelectSource
- AudioEinstellungen

Im Parameter Audio Einstellungen werden die Quellennamen gespeichert, Parameter0 Quelle 1 usw.

Das Objekt AudioSelectSource ist ein 8bit Objekt, wobei 0=Mute, 1=Quelle1 , usw., hier muss der Type aber EIS14 sein, da bei EIS2 eine Skalierung durchgeführt wird.



HTML Dokument: AudioSound03.html

### ZeitUhrWoche2:

Diese Funktion bietet die Möglichkeit über ein Switch Objekt eine Zeitschaltuhr zu realisieren.

Diese Neue Wochenuhr ist von der Bedienung her einfacher gestaltet als die Vorgänger Version.

Es ist nun flexibel möglich Wochenprogramme zu generieren, außerdem wurden Befehle wie Auf, und Ab für Jalousien mit eingebaut.

OPC-Objekte:

- ZeitEinstellungen
- Switching

ZeitEinstellungen:

Index	Funktion	default
0	Uhrzeit des Schaltpunktes 1	00:00
1-7	Wochentage ( Schaltpunkt 1) an dem Das Programm gültig ist 1=Mo, 7=So	False, alle
8	Funktion des Schaltpunkt 1 0..keine Funktion 1..Einschalten 2..Ausschalten 3..Abfahren 4..Auffahren 5..Umschalten	1
9	Uhrzeit Schaltpunkt 2	
10-16	Wochentage Schaltpunkt 2	
17	Funktion Schaltpunkt 2	
18	Uhrzeit Schaltpunkt 3	
19-25	Wochentage Schaltpunkt 3	
26	Funktion Schaltpunkt 3	
27	Uhrzeit Schaltpunkt 4	
28-34	Wochentage Schaltpunkt 4	
35	Funktion Schaltpunkt 4	
36	Modus 0..Automatisch 1..Dauer Ein (Ab) 2..Dauer Aus (Auf)	0

Vi-Funktion:           ZeitUhrWoche2.vi  
HTML Document:       ZeitUhrWoche2.html

## **JalousieWetter2:**

Dies Wetterstation kann einfach alles Sonnenachführung, Lammellenverstellung, temperaturabhängige Beschattung uvm.

Die Raumtemperaturabhängige Aktivierung der Sonnenautomatik besitzt eine Hysterese von 1,5 °C d.h. bei einem Wert von 25 °C muss die Temperatur > 25 °C sein, dann wird die Automatik aktiviert, bei <23,5°C dann wieder deaktiviert.

Über den Parameter 31 kann das zyklische Senden des Sicherheitsposition auf 0 deaktiviert werden normalerweise ist dieses Element Leer, oder false, dann wird bei Sicherheitsalarm alle 5 Sekunden die Position 0 auf den Bus gesendet, bei true wird dies nicht gemacht.

Diese Option kann aber vom Kunden selbst nicht verändert werden.

Die Einstellung „bei Dämmerung schliessen“ schliesst die Jalousien wenn es dunkel wird, geöffnet werden sie aber nicht mehr automatisch.

Neu gegenüber der Version 1 ist dass jetzt die Automatik auch abhängig von der Aussentemperatur aktiviert werden kann, es muss also innen einen gewissen mindestwert erreichen und die Aussentemperatur muss auch einen Gewissen Wert erreichen.

Folgende OPC Objekte werden benötigt:

**WetterEinstellungen**  
**Daemmerungssensor**  
**SonneAzimuth**  
**SonneElevation**  
**Regensensor**  
**Windsensor**  
**PositionMove**  
**PositionStep**  
**HeizungIst**  
**Temperatursensor (Aussentemperatur)**

Dämmerungssensor für die Automatische Steuerung mit Dämmerung

WetterEinstellungen:

<b>Index</b>	<b>Funktion</b>
0	Sonnenautomatik ein/aus
1	Windsicherheit aktiv
2	Windgrenze
3	Regen aktiv
4-10	Sonnenautomatik Aktivierungszeitfenster 0=Montag Beginn
11-17	Sonnenautomatik Aktivierung Ende 0=Montag
18	Azimuth Sonnenfenster Von
19	Azimuth Sonnenfenster bis
20	Elevation Sonnenfenster von
21	Elevation Sonnenfenster bis
22	Bei Dämmerung schließen
23	Dämmerung Grenzwert
24	Sonnennachführung der Lamellen aktivieren
25	Sonnennachführung Modus
	0 Maximum Licht

	1 Min Temperatur
	2 Immer schließen
	3 Immer öffnen
26	Korrekturwinkel in °
27	Temperatur aktiviert Sonnenautomatik
28	Temperatur Grenzwert
29	Wind+Sonne Verzögerungszeit zusätzlich
30	Schließwinkel bei Sonne in %
31	Zyklische senden des Windalarmes deaktivieren mit true
32	Aussentemperatur aktiv true/false
33	Aussentemperatur Grenze in °C

Der HTML Code ist in er Datei: [WebPfad]\Interface\JalousieWetter2.html

HTML-Options Dialog: JalousieWetter2Options.html

Die VI Funktion ist in der Datei: [VisuPfad]\Funktionen\MainFunktionen\JalousieWetter2.vi

### **AlarmSensorOLK:**

Dieser Sensor dient nur zur Anzeige der Status Informationen Sabotage und Alarm, nicht wie der AlarmSensor werden hier die Werte Zyklisch überwacht.

Für Die Anlage wird nur der Parameter herangezogen.

Um die Funktionalität vom empfangene Telegramm auf das Parameter Array zu Übertragen muss eine entsprechende Route im OPCLinker eingetragen werden vom Sensor auf das Einstellungen Array Nummer 2, welches nur bei Änderung gelinkt wird.

Diese Funktion wurde angelegt um die Funktionen etwas zu minimieren (bei großen Systemen)

Folgende OPC Objekte werden benötigt:

### **AlarmEinstellungen**

**Der Parameter Alarm Einstellungen wird wie folgt aufgeschlüsselt:**

Index	Funktion
0	Sabotage true/false true=Sabotiert
1	Nicht verwendet
2	Alarm geben true=Alarm, false=Kein Alarm

HTML Code: [WebPfad]\Visu\Interface\AlarmSensorOLK.html

### **FussbodenHeizungKühlen:**

Selbe Funktion wie die Standard FBHeizung, aber mit heizen und kühlen, außerdem wird die Anhebung kühlen durch den Kunden beeinflusst.

Außerdem wird der Parameter 15 der Heizung Einstellungen nicht mehr für die Temperatursturzerkennung, sondern für die Betriebsart verwendet.  
 Diese Funktion bietet außerdem die Möglichkeit die Temperaturanhebung im Kühlbetrieb flexibel einzustellen, d.h. wenn der Befehl zum umschalten kommt wird die Komforttemp. Automatisch um den Eingestellten Wert angehoben.  
 Vorausgesetzt ist allerdings dass die Anhebung im Schalter auf 2K eingestellt ist .

OPC Objekte:

- HeizungSoll**
- HeizungIst**
- HeizungVentilHeizen**
- HeizungKom**
- HeizungNacht**
- HeizungStandby**
- HeizungFrost**
- HeizungEinstellungen**
- HeizungBetriebsart** Wobei Heizung Betriebsart 0=kühlen, 1=heizen

<b>Index</b>	<b>Funktion</b>
0-6	Nachtbetrieb Startzeiten von 0=Montag bis 6=Sonntag
7-13	Nachtbetrieb Endzeiten von 0=Montag bis 6= Sonntag
14	Anhebung des Kühlbetriebes in °C
15	Betriebsart Komfort / Standby / Nacht / Frost
16	Heizung Komfort Grundtemperatur

VI-Funktion: FBHeizungKühlen.vi  
 HTML Funktion: FBHeizungGira.html

**DALI Device:**

Diese Funktion zeigt als HTML genau das Selbe wie ein DimmLicht, jedoch wird ein Dali Licht angesteuert.  
 Es wird eine VI Funktion geladen welche die Objekte DaliEVGError und DaliLampError überwacht, sobald eines der beiden auf 1 geht wird eine entsprechende Logging Meldung abgesetzt.

Es werden folgende OPC Objekte benötigt:

- Switching**
- DimmingRelativ**

**DimmingValue**  
**DaliEVGError**  
**DaliLampError**

HTML: [WebPfad]\Interface\lichtdimmbar.html

Vi: MainFunktionen\DaliDevice1.vi

**Jalousie Wetter 3:**

Diese Funktion ist exakt gleich wie die Jalousie Wetter2, aber der Regensensor erlaubt jetzt schließen und öffnen.

Index	Funktion
3	Regensensor Funktion 0-Keine Funktion 1-bei Regen öffnen (Sicherheitsalarm) 2-bei regen schliessen (wie Dämmerung)

Funktion: JalousieWetter3.vi

HTML: JalousieWetter3.html, JalousieWetterOptions3.html

**ZentralLueftungSL :**

Diese Funktion erlaubt die Steuerung einer Zentralen Lüftungsanlage mittels Zeitprogramm, es stehen 8 Schaltpunkte zu Verfügung.

Jedem Schaltpunkt kann eine Felxible Drehzahl von 0-100% zugeordnet werden.

Die Steuerung verfügt über die Betriebsarten Automatik und Dann Dauerbetrieb von 0, 10, 20, 30, 40, 50, 60, 70, 80 ,90 und 100%.

OPC-Objekte:

LueftungParameter  
 LueftungValue

Index	Funktion
0	Uhrzeit Schaltpunkt 0
1	Montag aktiv (als 0 oder 1 gesichert)
2	Dienstag aktiv
3	Mittwoch aktiv
4	Donnerstag aktiv
5	Freitag aktiv
6	Samstag aktiv
7	Sonntag aktiv
8	Geschwindigkeit 0-100
9-17	Schaltpunkt 1
18-26	Schaltpunkt 2
27-35	Schaltpunkt 3
36-44	Schaltpunkt 4
45-53	Schaltpunkt 5

54-62	Schaltpunkt 6	
63-71	Schaltpunkt 7	
72	Funktion 0=Auto, 1=0%, 2=10% usw	
73	Betriebsminuten	

HTML Funktion: ZentralLueftungSL.html  
 Vi-Funktion: ZentralLüftungSL.vi

**ZeitUhrWoche3, ZeitUhrWocheRD:**

Diese Funktion bietet die Möglichkeit über ein Switch Objekt eine Zeitschaltuhr zu realisieren.  
 Diese Neue Wochenuhr ist von der Bedienung her einfacher gestaltet als die Vorgänger Version.  
 Es ist nun flexibel möglich Wochenprogramme zu generieren, außerdem wurden Befehle wie Auf, und Ab für Jalousien mit eingebaut.  
 Es stehen 8 Schaltpunkte zu Verfügung.  
 Bei der RD Variante werden die Schaltpunkte per Zufallsgenerator variiert, es muss darauf geachtet werden dass die Abstände von den Schaltpunkten + des Maximal möglichen Variationsbereiches sich nicht überschneiden

OPC-Objekte:

- ZeitEinstellungen
- Switching

ZeitEinstellungen:

Index	Funktion	
0	Uhrzeit Schaltpunkt 0	
1	Montag aktiv (als 0 oder 1 gesichert)	
2	Dienstag aktiv	
3	Mittwoch aktiv	
4	Donnerstag aktiv	
5	Freitag aktiv	
6	Samstag aktiv	
7	Sonntag aktiv	
8	Funktion des Schaltpunkt 0: 0..keine Funktion 1..Einschalten 2..Ausschalten 3..Abfahren 4..Auffahren 5..Umschalten	
9-17	Schaltpunkt 1	
18-26	Schaltpunkt 2	

27-35	Schaltpunkt 3	
36-44	Schaltpunkt 4	
45-53	Schaltpunkt 5	
54-62	Schaltpunkt 6	
63-71	Schaltpunkt 7	
72	Betriebsart 0=Auto, 1=Dauer ein, 2=Dauer aus	
73	(nur bei RD) Zufallgenerator ein	
74	(nur bei RD) Variationszeit in min	

Vi-Funktion: ZeitUhrWoche3.vi  
 HTML Document: ZeitUhrWoche3.html

**AWSimulation:**

Eine Anwesenheitssimulation, es stehen verschiedenen Strategien zu Verfügung, prinzipiell werden Schaltvorgänge aufgezeichnet und dann entsprechende Nachgeführt  
 Es können beliebig viele Switching Objekte verbunden werden, diese werden dann aufgezeichnet und entsprechend simuliert.

Parameter:

Index	Funktion	default
0	Simulation an/aus	False
1	Modus der Simulation: 0-Zeiten exakt schalten 1-Zeiten per Zufallsgenerator variieren	0
2	Zufallsgenerator Zeitraum +/- xx min	20
3	Beobachtungszeitraum in Tagen	21
4	Totraum-Start innerhalb dieser Zeit finden keine Aktionen statt	23:00
5	Totraum-Ende und es werden jedenfalls alle Lichter abgeschaltet Zu Beginn des Totraumes	04:00

Es ist nötig die Tabelle AWSimulation in der TopologieDB anzulegen, in diese DB werden dann die Schaltvorgänge aufgezeichnet.  
 Aufbau der AWSimulation DB:

Feldname	Datentyp	Funktion	Sonstiges
ID	AutoWert	ID Primärschlüssel	
Kanal	int	Kanal Index	



Zeit	double	Zeit im LabView Format	
Wert	String	Wert des Befehls 0, 1 etc.	Leere Folge Ja

**Funktion:**

- Es wird jedes Telegramm in die DB eingetragen, auch während der Totzeit
- Um Mitternacht wird die DB gesäubert, also die ältesten Werte (älter als die Tage der Aufzeichnung) gelöscht
- Zu Beginn der Totzeit wird sicherheitshalber auf alle Teilnehmer ein aus gesendet
- Um 00:00 werden alle Schaltpunkte aus der DB gelesen, oder wenn man die Simulation einschaltet dann auch oder bei neustart des Systems
- Dann wird der Tag abgearbeitet, jedes Telegramm gesendet
- Wenn die Simulation ein ist wird simuliert, wenn die Simulation aus ist wird aufgezeichnet, sonst würde die Simulation ihre eigenen Schaltvorgänge aufzeichnen.
- Sollte die DB an dem gesuchten Tag keine Daten enthalten so wird der Tag verkleinert bis auf gestern um die Daten zu suchen !

**Farblicht:**

Diese Funktion ermöglicht die Ansteuerung von Farblichtern mit RGB Ausführung (rote, grüne und blaue leuchten).

Der Benutzer hat die Möglichkeit zum eine direkte Farben zu setzen und zu testen, dann können 6 Ablaufszenen verwaltet werden und über einen Dimmer Schalter kann das Licht ein/aus geschaltet und über die Dimmfunktion können die Farben gewechselt werden. Es ist nötig eine neu Tabelle in der Topologie DB zu erstellen Farblicht.

**Tabellenaufbau Farblicht:**

Spaltennahme	Funktio	Optionen
ID	Auto ID Wert	Counter
FID	Funktions Nummer	Long
SID	Sequenz Nummer 1-6	Int
Index	Index des Befehls 1,2,3,4-...	Int
Modus	Modus: 0-keine Aktion 1-farbe setzen 2-Farbe zufällig generieren 3-Warten (min) 4-Warten mit Farbübergang 5-Springe zu Anfang 6-Seuqenz beenden	
Parameter	Parameter in der Form: Color1/Color2/WaitTime	String, Leerezeichen erlaubt

## Parameteraufbau

Es wird ein Parameter mit der Funktion verknüpft, dessen Parameter Elemente haben folgende Funktion:

ID	Funktion	default
0	Rot Offset	0
1	Rot Faktor	1
2	Grün Offset	0
3	Grün Faktor	1
4	Blau Offset	0
5	Blau Faktor	1
6	Beleuchtungstype: 0-Leuchtstofflampen 1-LED 2-Glühlampen	1
7	Simulationsmodus der Automatik, wenn hier true steht wird die Sequenz im Simulationsmodus betrieben, d.h die Zeiten werden in 100 mal schneller abgespielt, es kann zu Problemen beim Farbübergang kommen	false
8	Startfarbe im Hexwert RGB, diese Farbe wird nach dem Einschalten aktiviert	
9	Dimmgeschwindigkeit in ms, zwischen zwei Schritten, zu kleine Werte funktionieren nicht da immer alle drei EIB Werte für RGB neu gesendet werden müssen also > 100ms	
10	Dimmschrittweite Standard ist 6	

## OPCItems:

FarblichtRotWert  
 FarblichtGruenWert  
 FarblichtBlauWert  
 FarblichtEinstellungen  
 FarblichtRGBValue  
 Switching  
 DimmingRelative  
 LichtszeneAufrufen

Über die OPCs Farblicht Rot Gruen und Blau werden die drei Werte direkt über EIS2 Values an die Beleuchtung gesendet, sie werden nur bei Änderung gesendet.

Das Objekt FarblichtRGBValue ermöglicht das direkte setzen des Lichtwertes über 0xHHHHHHH Werte.

Über switching und dimming relativ kann das licht mittels einer Dimmwippe angesteuert werden, also ein/aus schalten und dir Farben Dimmen.

Es können 6 Lichtszene Aufrufen Objekte verbunden werden, sobald einer 1 sendet wird diese Szene aufgerufen und abgearbeitet.

Es wird auch dann an alle anderen Szene Tasten eine 0 gesendet, damit eventuelle LEDs wieder abgeschaltet werden.



## JalPositionerLam:

Diese Funktion wird benötigt wenn Aktoren ohne Positionierung eingesetzt werden. Über Timer usw. wird intern immer die Position der Jalousie berechnet, dazu müssen alle Move und Step Objekte die Der Aktor empfängt auch mit dieser Funktion verknüpft bzw. die EIB Objekte korrekt verlinkt werden.

Die Laufzeit der Jalousie wird dann in einen Parametersatz in Sekunden eingegeben, außerdem wird sobald sich die Position der Jalousie um 2% ändert der Wert in die Parameter DB gesichert.

Dies dient dazu dass die Position wenn möglich auch nach Neustart etc. immer erhalten bleibt.

Sollte die Position verloren gehen so ist einfach eine Fahrt nach 100 oder 0 % nötig um die Funktion wieder in den Rhythmus zu bringen.

Generell entspricht 100% ganz geschlossen bzw. der unteren Endlage und 0 % ganz offen bzw. der oberen Endlage.

Zusätzlich kann diese Funktion noch die Lamellen in vier Schritten nachführen.

Dazu ist es aber notwendig das eine Komplette Wendung seitens der Aktor Zeiten auch in vier Schritte unterteilt ist.

Die Funktion führt dann nachdem die Absolute Position erreicht ist die Lamellen entsprechend nach.

Es ist nötig das eine auffahrende Jalousie zu 0% die Lamellen schließt und eine abfahrende zu 100% die Lamellen schließt, der Wirksinn ist 100%-Jalousie komplett zu, 0% Jalousien komplett offen.

Eine Nachführung der Lamellen ist nicht möglich, sollte ein Telegramm PositionStep empfangen werden so wird dies erst bei der Nächsten Absoluten Positionsänderung nachgeführt und nicht gleich.

OPC-Objekte:

- JalousieEinstellungen
- PositionMove
- PositionStep
- Move 1-10 Objekte
- Step 1-10 Objekte

Es werden zwar alle 10 Move bzw. Step Objekte überwacht, aber gesendet wird auf Objekt 0, also auf das jeweils erste.

Jalousie Einstellungen:

Index	Funktion
0	Laufzeit der Jalousie gesamt in Sekunden
1	Sicherung der Aktuellen Position in %

HTML Interface: [WebPfad]\Interface\JalPosSystemPositioner.html

VI-Funktion: [VisuPfad]\Funktionen\MainFunktionen\JalPosSystemPositioner.vi

## HeizungMertenKNX:

Diese Funktion ist speziell für Merten Schalter.

Sie ermöglicht erstmalig eine ganz neue Representation der Heizung gegenüber dem Kunden:

- Der Kunde kann im System alle 6 Temperaturen selbst einstellen, und die Regelung bleibt im Schalter
- Es giebt nur noch Tag, Nacht und Schutz Betriebsart, kein Standby
- Es wird die Betriebsart über KNX Byte umgeschaltet 0/Komfort, 1/Standby, 2/Nacht, 3/Frostschutz
- Die Solltemperatur kann am Schalter eingestellt werden, bleibt aber nur bis zur Nächsten Tag Nacht Umschaltung aktiv, dann wird sie wieder mit dem Wert in der Visu überschrieben
- Die Funktion ermöglicht ein Standard Nachtabsenkungs Programm

OPC-Objekte:

- HeizungIst
- HeizungSoll
- HeizungEinstellungen
- HeizungBetriebsart (KNX)
- HeizungVentilHeizen
- Switching

Das Switching Objekt wird für heizen/kühlen Umschaltung verwendet, 0/kühlen, 1/heizen. Betriebsart als KNX Byte Objekt

Parameter HeizungEinstellungen:

Index	Funktion	default
0-6	Nachtabsenkung von	22:00
7-13	Nachtabsenkung bis	04:00
14	Nicht verwendet	
15	Betriebsart 0/Tag, 1/Nacht, 2/Frost	0
16	Heizen Tagtemperatur	22
17	Heizen Nachttemperatur	20
18	Heizen Frostschutztemperatur	7
19	Kühlen Tagtemperatur	25
20	Kühlen Nachttemperatur	28
21	Kühlen Frostschutztemperatur	35

HTML Interface: HeizungMertenKNX.html

VI Funktion: HeizungMertenKNX.vi

## FensterEasy

Eine Fensterfunktion die nicht so komplexe Funktion bietet, aber Wind, Regen und Temperaturverriegelung sowie das automatisch schließen nach einer bestimmten Öffnungsdauer.

OPC-Objekte:

- Move (Fenster auf/zu)
- WetterEinstellungen
- Windsensor
- Regensensor
- Temperatursensor

Die Funktion überwacht direkt das Move Objekt, bei einem Sicherheitsalarm wird hier zyklisch der schliesbefehl gesendet (alle 600sekunden).

WetterEinstellungen:

Index	Funktion	Default
0	Windsicherheit ein/aus	False
1	Windgrenze km/h	70
2	Regensicherheit ein/aus	False
3	Außentemperaturverriegelung ein/aus	False
4	Außentemperatur Grenze	5
5	Spezial Funktion 0-keine 1-Automatisch schließen nach 15min 2-Automatisch schließen nach 30min 3-Automatisch schließen nach 1h 4-Automatisch schließen nach 2h 4-Automatisch schließen nach 4h	0
6	Auf/Ab invertieren	False

Der Parameter 6 biete die Möglichkeit das auf/ab zu invertieren, einfach mit true oder false arbeiten.

Die Anzeige im Web muss stimmen dann stimmt auch die Funktion !

ViFunktion: FensterEasy.vi  
HTML Interface: FensterEasy.html

### **ZeitUhrWocheFT:**

Exakt die selbe Funktion wie die ZeitUhrWoch3, jedoch wird die Möglichkeit geboten Feiertag im System, wie ein Sonntag auszuführen.

Wenn also der heutige Tag als Feiertag erkannt wird, so werden die Schaltzeiten vom Sonntag abgearbeitet insofern aktiv.

Demzufolge sind OPCItems exakt wie bei ZeitUhrWoche3.

ZeitEinstellungen:

Exakt wie bei ZeitUhrWoche3, jedoch der Parameter Nummer 73 signalisiert ob die Feiertage aktiv sind oder nicht (default false)

ViFunktion:           ZeitUhrWocheFT.vi  
HTMLInterface:       ZeitUhrWocheFT.html

### **JalousieWetter3SI:**

Eine Exakte Nachbildung der Funktion JalousieWetter3, jedoch mit der Möglichkeit ein Sicherheitsobjekt zu verbinden.

Das Sicherheitsobjekt ist vom Typ Switching und wird zyklisch alle 30s gesendet.

Zusätzlich werden PositionMove und PositionLam auch noch zyklisch alle 5s gesendet, dies muss bitte manuell über Parameter Nummer 31 auf true setzen.

VIFunktion:           JalousieWetter3SI.vi  
HTMLInterface:       JalousieWetter3.html

### **StatusanzeigeEinAus:**

Eine einfache Statusanzeige im Webbrowser, es wird als roter Text ein oder aus angezeigt eines Switching OPC Items.

OPC-Objekte:

-Switching

HTMLInterface:       StatusEinAus.html

### **LichtAutoStatusRM:**

Praktische Funktion zur Kontrolle von Zentralobjekten bei Lichtern.

Es können beliebig viele Switching Items verbunden werden und ein SwitchingRM Objekt zur Signalisierung.

Also wenn ich beispielsweise wissen möchte ob noch irgendwo im Gebäude ein Licht brennt, verbinde ich alle OPCItems der Lichter mit dieser Funktion und lege ein Rückmeldeobjekt als Dummy an.

Es werden dann alle Switching Werte verodert und auf das Rückmeldeobjekt geschrieben.

Wenn alle aus sind -> Rückmeldeobjekt auch aus, wenn mindestens einer ein ist -> Rückmeldeobjekt auch ein.

OPC-Objekte:

- Switching
- SwitchingRM

HTML- Interface: LichtRM.html  
Vi-Funktion: LichtAutoStatus.vi

### **BWAutomatik:**

Funktion zur automatischen Bewässerung von Pflanzen, es können bestimmte grenzen eingegeben werden, dann wird die Bewässerung ausgesetzt (Außentemperatur unter einem Wert, Regen länger als xx Minuten...).

Außerdem kann die Bewässerungszeit anhand von Sonnenscheindauer, Temperatur abgeleitet werden.

OPC-Items:

- Switching
- Magnetventil
- WetterEinstellungen
- Regensensor
- Temperatursensor
- SonneAzimuth

Über das Switching Objekt wird der Befehl zum Bewässern starten für die automatisch berechnete Zeit (z.b. von einer Zeitschaltuhr oder einem Taster gegeben).

WetterEinstellungen Parameter:

Index	Funktion	Default
0	Modus: 0-Bewässerung aus 1-Bewässerung Automatik 2-Bewässerung Dauerbetrieb	
1	Außentemperatur Grenze	
2	Regendauer Grenze in Minuten	
3	Minimum Bewässerungsdauer in Minuten	
4	Maximum Bewässerungsdauer in Minuten	
5	Berechnung Bewässerungsdauer 0-immer Maximum 1-abhängig von der Temperatur 2-abhängig von Sonnenscheindauer 4-abhängig von Temperatur + Sonnenscheindauer	
6	Berechnungsgrundlage Außentemperatur von °C	



7	Berechnungsgrundlage Außentemperatur bis °C	
8	Berechnungsgrundlage Sonnenscheindauer von (Minuten)	
9	Berechnungsgrundlage Sonnenscheindauer bis (Minuten)	
10	Datum + Uhrzeit der letzten Bewässerung	
11	Jetzt Bewässern Signalisierung 1-auto 2-minimum 3-maximum 4-Stopp	

Die Berechnung der Bewässerungszeit erfolgt immer durch Lineare Interpolation zwischen den Punkten min und max.

Über den Parameter 11 kann durch setzen auf einen der Werte 1-4 entsprechend die Bewässerung gestartet werden.

HTML Funktion: Bewaesserung.html

Vi Funktion: BewässerungAutomatik.vi

- **Die Tabelle FunktionType in der TopologieDB:**

Spalten beschreibung:

**FunktionType:**

Name der Funktion nur Buchstaben und Zahlen, keine Sonderzeichen erlaubt

**HTML Interface:**

HTML Code der Fuktion, normalerweise im WebServer\Visu\Interface Verzeichnis, es muss Immer ein HTML Code eingetragen werden, eine Funktion welche kein HTML besitzt muss eine Leere Datei als Quelle angeben da es sonst eine Fehlermeldung gibt !

**VI Funktion:**

Vi Interface Pfad, gemäss den Standardfunktionen wird dieses VI ins Temporäry Verzeichnis vervielfältigt und ausgeführt, dieser Eintrag ist Optional, wenn dieses Feld leer bleibt dann wird keine Funktion ausgeführt.

**Caching:**

Je Nach (1=Ja, 0=Nein) wird für diese Funktion eine Caching Datei angelegt d.h. dass nicht jedes Mal, sondern nur beim ersten Aufruf die Webseite aus der DB erstellt wird. Beispielseise kann dies Funktion nicht verwendet werden wenn Parameter beim Laden der Seite aus der DB gelesen und angezeigt werden sollen (also das Value Attribut funktioniert nicht).

**WAP-Interface:**

Gleiche Funktion wie beim THML Interface, jedoch als WAP Code im Verzeichnis [WebPfad]\Interface\wap. Diese Angabe ist Optional, sollte keine WAP Seite erstellt worden sein so bitte einfach das Feld leer lassen.

**Priority :**

Ausführungspriorität, 0 ist die höchste (100ms) dann wird einfach das vi jede 100ms ausgeführt, bei 10 wird das vi alle 1,1s ausgeführt. Dies gillt natürlich nur wenn TaskTime auf 100ms !

## **Verzeichnisstruktur**

Das Visusystem hat folgende Verzeichnisse:

### **\ACSInterface**

Sämtliche Komponenten für ein ACS Bus Interface welches mit einem ACS-RS232 Gateway mit aktiver Telegrammwiederholung realisiert wird.

### **\AMX**

enthält das AMX Interface zu Commtex AMX Systemen, ein TCP/IP Interface welches vor allem mit Kreil und Lechner zum Einsatz kommt (kosten 500Eur)

### **\Application Control**

Alles für die Generelle Programmsteuerung, Logging System Registry, System aktivation und sämtliche Funktionen die allgemein Verwendung finden, Arrays usw..

### **\AudioControle**

Dieses Verzeichnis enthält eine Funktion und deren Komponenten um Audio Files auf einem Lautsprechersystem wiederzugeben.

### **\BusSystems**

Zukünftiges Verzeichnis für alle neu implemmtierbaren Bussysteme wie Mbus, Luxmate und Sauter MSR.

### **\DataBase**

Enthält sehr viele Datenbankfunktionen vor allem aber die SQL Query Topologie und SQL Query Record, die Unterste SQL Schnittstelle für den Zugriff von LabView auf diese Datenbanken.

Außerdem befindet sich eine Leere Topologie und Record Tabelle im Verzeichnis System

### **\DataLogger1-2**

Record Engine

### **\EIBInterface**

Enthält das EIB Interface mittels Schlaps Partner Weiche

### **\EventExplorer**

Enthält alles was mit Alarm und Weiterleitung zu tun hat (Email, SMS usw.)

### **\Funktionen**

Enthält alle Standard Systemfunktionen die über das Web Interface angelegt werden können

### **\Logik Funktionen**

Enthält Logische Grundfunktionen (Flip Flops, Timer, Flanken, etc.)

### **\LV-Apache**

CGI Interface zu Apache 2

### **\OPC Server Comm**

Alles was mit Datenaustausch zu tun hat EIB, ACS, Mbus etc wird dort abgewickelt. Das Grundlegende Interface ist dort enthalten.

### **\OPCSubFunktions**

Spezielle Form der OPC Items Werte über Funktionen erhalten, derzeit können hier nur Funktionen die nicht wieder OPCItems lesen oder schreiben aufgerufen werden ( Rekursionsproblematik )

### **\PHP**

PHP Extension Version 4.3.7.7

### **\System**

Systemfunktionalitäten wie der TCPAdmin Server oder ähnliches

### **\Time Functions**

Zeitfunktionen

### **\UserData**

Alle Dateien die ein System dem Nutzer anpassen sind hier abgelegt, beide Datenbanken, die FixFunkions.txt Grundrissbilder und die MainFrame.html Datei, also das Home der Visu, da diese immer benutzerspezifisch geladen wird.

### **\Video**

Ein paar spezielle Funktionen um Bilder per FTP von Webcams zu lesen oder um Bildformate zu konvertieren

### **\WebServer**

in diesem Verzeichnis sind alle Files für den Webserver enthalten, auserdem ist das Root Verzeichnis auf dieses Verzeichnis gelegt.

Auch Bilder usw. für den WebServer sind hier abgelegt.

### **\Wetter**

Wetter Funktionen, vor allem Sonnenstandsberechnung.

### **\Zutritt**

Zutrittskontrolle ACS Implementierung

## **Globale Funktionselemente Konfigurieren:**

Globale Funktionselemente sind Funktionen die über die FixFunkions.txt Datei geladen werden, das besondere ist das sie nicht dynamisch anhand des Funktionsindex kopiert werden sondern in ihrem Originalverzeichnis ausgeführt werden.

### **Element ACSInterfaceTask:**

Übernimmt die Komm mit dem ACS Bus es muss in der Registry der COM Port eingetragen werden.

Außerdem kann der ACS Bus im 8-bit Adressmodus betrieben werden, dies ist nötig wenn alte Geräte die nicht der ACS2.0 Spec. Unterliegen angeschlossen werden.

Dazu muss die Funktion geöffnet werden und per Datenoperation->Standardwert diese Funktion eingestellt werden.

### **Element RecordEngine:**

Übernimmt das Management der Aufzeichnungen, hier muss die Datenbank entsprechend konfiguriert werden.

**Element RecordClearUp:**

Übernimmt das löschen der alten Datensätze um Mitternacht.

**Element SendSystemzeit:**

Sendet alle 10s die Systemzeit auf den Bus, das EIBObjekt **muss** Systemzeit heißen !

**Element SunCalculatePosition:**

Berechnet alle 5 Sekunden die Position der Sonne, als Parameter werden Absolute Position, Verzögerungszeit usw. übergeben.

Das VI wird auch als Graphische Anzeige für den Explorer verwendet dazu lädt das VI beim Start ein Picture aus dem Verzeichnis \Userdata\Image, das bild muss SonnenPos.bmp heißen.

Wenn das Bild nicht vorhanden ist, dann wird nichts geladen.

Es müssen folgende OPCItems mit dem Namen angelegt sein:

- SonneAzimuth - Parameter
- SonneElevation - Parameter
- WetterEinstellungen - Parameter
- Sonnenfuehler - EIB Objekt (0-100%) Sonnenschein

Sollten diese OPC Items nicht existieren oder so benahmt sein, Funktioniert die Sonnenberechnung nicht.

Die Berechnung gibt bei einem Sonnenstand der unter dem Grenzwert liegt und schon eine Verzögerungszeit lange weg ist für SonneAzimuth und SonneElevation 0 zurück !

**Element SystemStartStopp:**

Diese Funktion ist momentan nur zuständig die System Start und Systemstopp Meldungen zu protokollieren.

Gedacht wäre ein automatischer Shutdown des Systems bei Spannungsausfall in Verbindung mit einer USV.

**Element SystemDataBackupTask:**

Diese Funktion ist für den Automatischen Backup zuständig, außerdem sorgt sie dafür das Backups nach einer bestimmten zeit automatisch vom Datenträger gelöscht werden.

Beide Funktionen SystemStartStopp und SystemDataBackupTask sind immer vorhanden und benötigen ein OPC Item Namens Systemeinstellungen welches folgendermaßen Aufgesplitet wird:

Index	Funktion
0	System Start und Stopp loggen
1	Bei Spannungsausfall automatisch Shutdown (mit USV)
2	Sicherungszeitpunkt (Uhrzeit)
3	Auto Backup aktiv
4	Backup Pfad
5	Backup Tiefe in Tagen
6	Backups automatisch löschen
7	Backup Intervall in tagen

8	Backup Modus 0= Nur Parameter / 1= Nur Aufzeichnungen / 2 =beides

## **Konfiguration des Derdack Message Master:**

Der Derdack Message Master wird verwendet um SMS über ein analoges Modem zu versenden.

Als erstes muss die entsprechende VCL35.bpl und noch eine Borland Library in das Windows System32 Verzeichnis kopiert werden.

Dann kann das Setup Programm im Verzeichnis \EventExplorer ausgeführt werden.

Im Message Master Konfigurationsprogramm wird nun das Modem und der SMS dienst konfiguriert.

Modem: Initialisierung bei analogen TK Nebenstellen ATX3L1

A1 SMS dienst, einfach über die Datei Services-Modem-ISDN.inf im Verzeichnis \EventExplorer den A1 Service importieren.

Die Provider Nummer von A1 ist 0900664914 / TAP Protokoll

Nach der Installation des MessageMaster ist ein Neustart des Rechners unumgänglich, das Toolkit funktioniert zwar, lässt sich aber aus LabView nicht starten

Ab der Version 1.3 ist es möglich die Service Provider zu wählen, d.h. dass die DLL neu kompiliert wurde und nun als Parameter auch den Service Namen übermittelt, hierzu sollte ein Eintrag in die Registry Namens SMSService gemacht werden und der Service Name muss dem im Message Master entsprechen.

SMSService=A1

Sollte dieser Eintrag in der Registry fehlen oder leer sein dann wird defaultmässig der Service „A1 SMS (A1)“ genommen“.

## **SystemDatenbank:**

Aus Gründen der Kompatibilität bei Systemerweiterungen, wird eine Neue Datenbank im Verzeichnis DataBase\System\System.mdb angelegt.

Diese Datenbank enthält alle Tabellen die nicht vom Kunden abhängig sind, sondern von der Version des Systems.

Diese Tabellen werden in der TopologieDB mit einem Link eingefügt (Datei, ExterneDaten, Tabellen Verknüpfen).

Folgende Tabellen sind gelinkt:

- OPCTypes, enthält alle OPC Typen
- RecordModes, enthält alle Aufzeichnungstypen
- Lichtszenen, enthält alle Lichtszenen
- FunktionTypes, enthält alle Funktionen mit Pfad und VI Funktion

Dies hat den Vorteil das bei Systemupdates die Datenbanken nicht (bis auf Parametersätze) geändert werden müssen.

Zu Beachten ist jedoch das diese gelinkten Datenbanken nicht mehr über das Netzwerk auf dem Client Rechner zu öffnen sind, da dieser dann die System.mdb nicht findet (Access Problem)

### **OPCLinker:**

Der OPC Linker kann alle internen OPC items miteinander verbinden und Linken, es stehen möglichkeiten zur Richtungssteuerung und zur Updatewahl zu Verfügung.  
Konfiguriert wird der OPC linker über die OPCLinker Tabelle in der Topologie DB.

Die Option Direction Beide Richtungen funktioniert aber noch nicht, hier muss noch die Source Adresse miteinbezogen werden, damit nicht unendlich gelinkt werden kann.

Zu Beachten ist das bei OPC Items vom Type Parameter der Modus Jedes Telegramm nicht zu Verfügung steht !



## **Web Server Programmierung:**

Der Web Server benötigt HTML Dateien die bekannten Namen werden durch entsprechende Daten ersetzt.

Das Interface nimmt die HTML Datei sucht alle in der Funktion enthaltenen OPC Typen und ersetzt sie durch den realen Namen.

Es können auch Index verwendet werden.

Außerdem kann durch Angabe der Erweiterung Value nach dem OPC Namen oder dem Index, der momentane Wert des Objektes eingesetzt werden.

Weiteres wird durch Angabe des Strings //OPCReloadItem=xx; in einem JavaScript Abschnitt die Angabe des xx alle 2 Sekunden aktualisiert.

## **Text Plug In:**

Oft ist es erforderlich das Elemente speziell formatiert werden, dazu wird der WebServer um eine PlugIn Funktionalität erweitert.

Das Plug In wird al aller erstes in den Vorhandenen Vorlagen Text eingefügt und danach werden die üblichen Strings ersetzt.

Durch Die Angabe LoadWebServerPlugIn=PlugIn.vi(Par1/Par2/Par3/Par4/....); wird an genau dieser Stelle das VI PlugIn.vi welches im Verzeichnis \Interface\PlugIn platziert sein muss ausgeführt.

Das PlugIn besitzt 4 Eingänge und einen Ausgang es muss exakt diesem Anschluss entsprechen da es sich hierbei um eine Strikt typisierte VI Referenz handelt.

Alle sind vom Type String.

Verwendung sollte das EmptyWebPlugIn.vi im Verzeichnis finden.

Beim 1.Eingang handelt es sich um die aktuelle FunktionsID sollten Daten wie FunktionsName usw. benötigt werden, so kann dies von der Datenbank abgefragt werden.

Der Eingang zwei spiegelt die Parameter in übergebener Reihenfolge wieder, es handelt sich um einen mit / getrennten Par String welcher mit der OPC Funktion ParameterStringToArray in eine Array gewandelt werden kann.

Die Eingänge 3, 4 sind noch nicht definiert.

Das Plug In liefert einen HTML oder http Text zurück welcher anstelle des PlugIn Aufrufes platziert wird.

## Logging:

### Allgemein:

Das Logging ist die Grundlage für alle Alarmmeldungen, es können keine Alarmmeldungen ohne Logging weitergeleitet werden.

Das Logging besteht aus 3 Tabellen:

Die UserLoggMessage Tabelle in der Topologie Datenbank, in ihr werden ab Index 1000 alle Kundenspezifischen Meldungen abgegeben.

Die UserLoggMessageTable in der RecordDB (ist mit einer Tabelle in der System DB verlinkt) in dieser Tabelle werden autoindizierte Systemmeldungen gespeichert.

Die LoggBase in der RecordDB speichert alle Loggings, unendlich lange.

Bei der Darstellungen eines Logging werden die 4 String Parameter direkt als HTML Code geladen so ist es beispielsweise möglich Bilder miteinzubinden.

Bilder werden im Verzeichnis Userdata\Images\Logg gesichert, ebenfalls unendlich lange.

Wird ein Logging gesichert so wird überprüft ob das Logging einen String der Form <> enthält, sollte der String diese Zeichen enthalten so wird der enthaltenen Index überprüft und sollte dieser Index in der AlarmTable hinterlegt sein so wird eine entsprechende Meldung abgesetzt.

Momentan sind die Meldungen SMS, Email, Sound MessageWindow und None verfügbar, eine Hinzufügung ist mit Aufwand verbunden da diese Möglichkeiten mehrmals fix ins System programmiert sind.

Der Type Sound ist aber noch nicht funktionsfähig.

Usermeldungen werden ab Index 1000 abgelegt, sie sind in der Topologie des Kunden gesichert, so dass die Portabilität des Systems erhalten bleibt.

Der Aufruf von Usermeldungen muss mit speziell programmierten Überwachungsfunktionen realisiert werden, hierfür steht momentan keine standardisierte Lösung zur Verfügung. Es muss also eine VI Funktion im Verzeichnis Userdata programmiert und in der FixFunkions.txt aufgerufen werden.

## **Alarmmeldungen:**

Es stehen mehrere Services zur Verfügung, generell kann jedes Logging mehrmals mit einem Service verknüpft werden, dann werden alle aufgerufen.

Die Services sind in der Datei AlarmServices.txt definiert.

In dieser Datei sind Abschnittsweise alle Services aufgelistet, mit dem Namen einem Index und einer VI Funktion welche den Service Managet.

Außerdem enthält jeder Service ab v1.5 den Eintrag Hypertext= indem ein php Script aufgerufen wird, welches die Parameter bearbeitet.

Die AlarmService Vis werden über strikt typisierte VI Refnums aufgerufen deshalb muss das Anschlusspanel genau dem der anderen AlarmService Vis entsprechen.

Generell werden die Parameter mit einem / getrennt übergeben und in ein Array konvertiert.

### **Service None:**

Standard es wird nichts gemacht.

### **Service Email:**

Es wird eine Email versendet, es werden der Original Message Text und der User Text miteinander verknüpft, als Parameter dienen mit / getrennte Email Adressen.

Das Vi benötigt einen SMTP Server, die Adresse des Servers ist in der Registry gespeichert (MailServer).

Der Absender steht im Eintrag MailSender und die Betreff Zeile kann über den Eintrag MailSubject eingegeben werden.

### **Service SMS:**

Es wird eine SMS an eine oder mehrere Nummern gesendet, Voraussetzung ist ein installiertes MessageMaster Toolkit+ Borland C++ Builder Runtime dlls.

Es wird der Original Meldungstext und die Benutzermeldung verknüpft übergeben.

Der Service muss über das SMS Toolkit konfiguriert werden, jedenfalls wird der verwendete Service über den Registry Eintrag SMSService festgelegt, sollte der Eintrag leer sein wird standardmäßig A1 (SMS) A verwendet (dies wurde gemacht um abwärtskompatibel zu bleiben)..

### **Service Sound abspielen:**

In Zusammenhang mit ACS Sound03 Modulen kann hier eine Modulare Raumalarmanlage aufgebaut werden.

Jeder Sound03 besitzt 3 Eingänge und je ein Umschaltobjekt (0=Mute, 1=Kanal 1 usw.).

Die Benutzeroberfläche erlaubt es die Lautstärke individuell für jeden Raum den Wiedergabemodus das Audiofile und die Anzahl der Wiederholungen einzugeben.

Die Modie sollten eigentlich selbsterklärend sein, sie werden mit Index wie sie gelistet sind abgespeichert (0= nicht wiedergeben, 1=immer wiedergeben , usw.).

Im Parameter Array werden Pro Raum eine 3er Kombination an Parametern gespeichert immer: FunktionID/Modus/Lautstärke.

Als Alarm text wird zuerst der Filename (ohne Verzeichnis) und dann mit / getrennt die Anzahl der Wiederholungen gesichert.

Als Verzeichnis ist das Verzeichnis fix auf c:\VisuSys1-1\UserData\Audio eingestellt.

Die Funktion Sound Wiedergabe stoppen beendet alle Wiedergaben und löscht auch die komplette Alarm Que !

Dies ist nötig wenn man z.b. eine Sirene 1000mal wiedergibt und dann bei abschalten des Alarmes möchte das auch die Sirene ausgeht, dann muss dies entsprechend verknüpft werden.

Ablaufdiagramm:

- 1.) sobald ein Soundfile in der Que landet sucht der Player alle OPC Objekte anhand der Funktions Ids raus und schaltet die Quellen um.
- 2.) Dann werden die Lautstärken angepasst
- 3.) Nach einer Wartezeit von 1s wird dann der Player gestartet
- 4.) Nach Ende Aller Wiedergaben schaltet zuerst die Lautstärke und dann die Quelle wieder zurück.

Probleme:

1. Der Sound03 Die Rückmeldung funktioniert per Firmware nicht korrekt
2. Es sollte eine Wartezeit von 1-2s nach ende des abspielen zwischen Quelle und Lautstärkenumschaltung eingefügt werden
3. Es sollte noch die Quelle vor der Umschaltung gesichert werden

Voraussetzungen:

1. Es muss ein Sound 03 sein welcher die AudioSource, die AudioStatus und die AudioDimming Value Objekte korrekt verknüpft hat.
2. Beim Sound03 sollte der Dimmwert angesprungen werden um schnell umschalten zu können.
3. Der Tuner ist auf Eingang 1 und der PC auf Eingang 3

### **Service TCP/IP Client:**

Der TCP/IP Client ist ein Programm welches auf dem Client Rechner des Kunden Installiert ist und dort bestimmte Befehle ausführen kann.

Als Parameter steht lediglich die IP-Adresse des Client PCs.

Im Feld Meldungstext kann dann eine bestimmte Meldung an den Client gesendet werden

Es stehen Funktionen wie open, execute, playsound und messagebox zu Verfügung.

Als Port wird der Port 4567 geöffnet.

Clientseitig ist keine Konfiguration nötig, der Client kann unter Systemeinstellungen / Software downgeloadet werden.

### **Service OPC Wert setzen:**

Hier besteht die Möglichkeit jedes OPCItem im System mit einem wert zu setzen sobald die Meldung ausgeführt wird.

Als Parameter muss der OPCName gefolgt vom Wert übergeben werden.

OPCItem/1

## **Web-Caching:**

Um die Geschwindigkeit des Webservers zu erhöhen wurde eine Caching Funktionalität hinzugefügt, dazu ist ein Verzeichnis \UserData\WebCache nötig.

Die Vorgehensweise ist wie folgt:

Der Webserver prüft ob ein Cache File vorhanden ist  
Falls Ja übermittelt er direkt den Inhalt des Cache File  
Falls nein generiert er die Daten neu und speichert sie aber in einem Cache File

Es stehen Funktionen im Verzeichnis WebServer\GeneralUse\Caching zu Verfügung.  
Die Umsetzung im Detail hängt von der Funktion ab.  
Das webCaching Verzeichnis wird standardmäßig bei jedem Systemstart komplett gelöscht (alle Dateien).

## WDT, Watchdog Timer:

Weil es immer wieder vorgekommen ist das das System (LabView) abgestürzt ist, haben wir einen Watchdog Timer programmiert.

Dieser Task muss in den FixFunktionen ausgeführt werden.

Beim Systemstart wird dann ein zweiter Prozess namens wdt.exe ausgeführt dieser muss immer vom VisuSys zurückgesetzt werden, wird er dies nicht innerhalb eines bestimmten Zeitrahmens, so wird das Kommando „shutdown -r -t 10“ in der Dos Shell ausgeführt und der Server komplett neu gestartet.

Die Kommunikation von LabView und dem WDT Prozess erfolgt zum einen über ein Event Objekt und zum anderen über eine Datei namens wdt im c:\temp Verzeichnis.

Die WDT Time wird in der Registry als wdt eingetragen (60s), standardmäßig sollten 60 Sekunden eingetragen werden, da das System insbesondere beim Ansehen von Langen Aufzeichnungen sehr lange braucht um diese zu laden und teilweise den Ganzen LabView Prozess blockiert.

Es wird eine Logg Datei namens C:\RestartLogg.txt geführt in welcher alle Systemstarts und WDTs aufgelistet werden.

## Der Setup Editor:

Über den Setup Editor können sämtliche Einstellungen vorgenommen werden, er wird über Internet IP-Adress/Setup/ gestartet.

In der Ersten Version können folgende Dinge noch nicht gemacht werden (dies muss dann direkt über Access eingegeben werden).

- OPCLinker
- einzelne OPC Items anlegen
- UserLogging Meldungen anlegen
- Registrierung bearbeiten
- UserWebInterface es können nicht beliebige OPC Types eingefügt werden

## PHP-Update:

Seit der Version 1.4 wird PHP eingefügt.

Alle PHP Source Files sind im Verzeichnis VisuSys1-1\PHP enthalten, zur Installation muss lediglich die PHP.ini ins c:\Windows kopiert werden.

Der Script Alias für den Apache ist im Konfig File bereits enthalten.

PHP wird momentan für die Grafischen Funktionen wie GeSunPosPicture.php und Aufzeichnungen verwendet (GD2 Libraray), außerdem für GetLittleLogg.php.

Alle PHP greifen direkt per ODBC auf die Datenbanken zu, um OPC Werte etc. von LabView zu bekommen wird einfach eine Anfrage an localhost per PHP durchgeführt, deshalb ist es unbedingt nötig das der localhost auch alle Rechte im Apache Konfigfile bekommt.



## EIB Interface / ACSInterface:

Das EIB Interface dient um den EIB Bus direkt anzusprechen.

Alle Adressen müssen in der EIBObjekts Tabelle korrekt angelegt sein mit Type, Adresse und Link Adressen.

Die Beschreibung gilt auch für das ACSInterface allerdings kann beim ACS die option readable nicht verwendet werden, da nicht unterstützt.

Tabellenspalten:

GroupAdress:

Gruppenadresse im Dezimahl Format.

Description:

Beschreibung der Adresse, für den Elektriker

EISType:

EISType der Adresse wie folgt:

ID	Name	Funktion
0	uncertain	Unbekannter Datentyp
1	EIS1	1bit Datentyp ein/aus
2	EIS2V	1Byte Wert von 0-100% (wird als 0-255 skaliert)
3	EIS2RD	4bit Telegramm zum dimmen
4	EIS3	Zeitlegramm
5	EIS4	Datumstelegramm
6	EIS5	2Byte Float Zahl nach IEEE
7	EIS9	4Byte Float Zahl nach IEEE
8	EIS10	16 bit Counter unskaliert
9	EIS11	32 bit Counter unskaliert
10	EIS15	Charakter String mit max. 14 Zeichen
11	EIS14	8 bit Counter unskaliert

Priority:

Priorität des Telegrammes wie folgt:

ID	Priority	Funktion
0	System	Höchste Priorität
1	Alarm	
2	High	
3	Low	Kleinste Priorität, ist normal Standard

**LinkAddress:**

Die Link Adressen werden mit / getrennt eingegeben, wenn z.b. die Adresse 1 als Link Adresse 10 und 11 eingegeben hat.

Dann bedeutet das das bei Empfang von Daten auf die Adresse 10 oder 11 auch die Daten von Adresse 1 aktualisiert werden.

**InitMode:**

Der InitMode kann verschiedene Anweisungen enthalten, derzeit sind folgende möglich.

Sie können auch mehrere dieser Anweisungen eingeben diese mit / getrennt eingeben.

Die Anweisungen sind nicht Case sensitiv.

Anweisung		
readable	Bedeutet das bei einer Leseanforderung auf diese Adresse das System antwortet	
setdata	Das OPC Objekt wird mit diesem Wert initialisiert und es wird ein Telegramm auf den Bus gesendet. Im Parameter danach muss der Wert als Text stehen (true, false etc.)	
remanent	Die Daten werden in der DB bei jedem Empfang gesichert So dass sie nach Systemstart sofort wieder mit aktuellem Wert und Zeitstempel verfügbar sind	



Blumenastr. 4  
A-6890 Lustenau

**Tel:** +43(5577)84279-0  
**Fax:** +43(5577)84279-15

[www.sontec.at](http://www.sontec.at)  
[info@sontec.at](mailto:info@sontec.at)