

Bachelorarbeit

Kooperation von Comexio-Systemen über das Ethernet zum Aufbau einer Leitechnik auf der Basis von IP-Symcon

Cooperation of COMEXIO Systems via Ethernet for the Creation of a Building Management System Based on IP-Symcon

Wintersemester 2015/16

Erstell von:

Marcel Schulte-Pelkum

MATRIKELNUMMER | 7088543

Betreut von:

Prof. Dr. Bernd Aschendorf

FH Dortmund

Marcel Schulte-Pelkum

Dohlenstr. 7

44357 Dortmund

01631749081

marcel.schulte-pelkum065@stud.fh-dortmund.de

Erstprüfer: Prof. Dr. Bernd Aschendorf

Zweitprüfer: Dipl.-Ing Sandra Stahlberg

Selbständigkeitserklärung

Hiermit erkläre ich, Marcel Schulte-Pelkum, dass die vorliegende wissenschaftliche Arbeit von mir selbst verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt worden ist und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt worden sind.

Geistiges Eigentum Anderer ist in jedem Einzelfall eindeutig am Ort des Zitats bzw. der Entlehnung als solches kenntlich gemacht und ins Quellenverzeichnis eingetragen. Mir ist bekannt, dass Plagiate als Täuschungsversuche gewertet werden und entsprechende Konsequenzen eintreten würden.

Dortmund, den 11. Februar 2016 _____

Zusammenfassung

Die vorliegende Arbeit thematisiert die Zusammenarbeit zwischen der Automatisierungssoftware, „IP-Symcon“ (IPS) und der Gebäudeautomation, „COMEXIO“.

In früheren Serverversionen war es den IO-Servern nicht möglich über das Netzwerk, bzw. Ethernet, zu kommunizieren. Größeren Projekten, mit mehreren Servern, stand es nicht zur Verfügung Informationen oder Daten auszutauschen. Erst seit der Firmware 2.7, besitzen die COMEXIO-Systeme eine offizielle API (zu deutsch: „Schnittstelle zur Anwendungsprogrammierung“), die es ihnen ermöglicht, mit anderen COMEXIO-Systemen oder übergeordneten Softwarelösungen zu kommunizieren.

Es wird in dieser Arbeit gezeigt, wie sich mit mehreren COMEXIO-Systemen eine Gebäudeleittechnik auf der Basis von IP-Symcon einrichten lässt. Um dies zu ermöglichen, werden die IO-Server via Ethernet mit dem Netzwerk verbunden. Durch diese Verbindung ist es möglich, die Gebäudeautomation von COMEXIO in ein zentrales Verwaltungssystem einbinden zu können. Dieses wird mit IP-Symcon aufgebaut. IPS wird hierbei verwendet, um Daten vom Bus oder aus dem Netzwerk zu erfassen. Diese Daten, wie z.B. Temperatur, Leistung, Stromverbrauch etc., sollen visuell dargestellt und gespeichert werden. Zudem ist es möglich über IP-Symcon Daten bzw. Werte an die Gebäudeautomation über visualisierte Endgeräte (z.B. Computer oder mobile Geräte) senden zu können. Durch die Kombination von IPS und COMEXIO wird unter anderem ein Smart-Meter eingerichtet, welcher dem Benutzer den tatsächlichen Energieverbrauch und die tatsächliche Nutzungsdauer anzeigt. Aus den erfassten Eingangs- bzw. Ausgangsgrößen lassen sich Benutzerprofile sowie Kostenberechnungen erstellen. Um die relevanten Daten erfassen und senden zu können, wird über sog. „http-Requests“ auf die API der COMEXIO-Server zugegriffen. Diese Programmierschnittstelle gibt der Software von IPS die Möglichkeit über das Netzwerk auf die Mastereinheiten der Gebäudeautomation zuzugreifen. Die Kommunikation zwischen IP-Symcon und COMEXIO muss über eigens geschriebene Scripte realisiert werden, da IP-Symcon das COMEXIO System noch nicht unterstützt. Die Parametrierung von IP-Symcon bzw. das Anlegen von Systemvariablen, Scripten und der „WebFront“ über die Verwaltungskonsole sowie die Programmierung der Gebäudeautomation über das Webinterface werden im Verlauf dieser Arbeit detailliert beschrieben.

Abstract

The present thesis deals with the interaction between the automation software IP-Symcon (IPS) and the building automation system COMEXIO.

In earlier server versions, the IO servers were unable to communicate via the network or the Ethernet. In larger projects involving multiple servers an exchange of information or data was not possible. It was only after the release of Firmware 2.7 that the COMEXIO systems possessed an official API (Application Programming Interface) enabling them to communicate with other COMEXIO systems or superordinate software solutions.

In this thesis it will be shown how to implement a building management system based on IP-Symcon with the help of multiple COMEXIO systems. To achieve this, the IO servers will be connected to the network via Ethernet. This allows an integration of the building automation system by COMEXIO into a central management system. This management system will be built using IP-Symcon. Here, IPS serves the purpose of gathering the data from the bus or from the network. These data, such as temperature, performance, power consumption etc. are to be visualized and saved. Furthermore, it is possible to send data and values by means of IP-Symcon to the building automation system via visualized devices (e.g. computers or mobile devices). Through the combination of IPS and COMEXIO a smart-meter (amongst other things) will be implemented, showing the user the actual energy consumption and the actual time of use. User profiles and cost calculations can be created from the gathered input and output values. To be able to gather and send the relevant data, the API of the COMEXIO servers will be accessed via so-called “http requests”. This programming interface enables the software by IPS to access the master units of the building automation system via the network. The communication between IP-Symcon and COMEXIO needs to be realized via specifically written scripts, as IP-Symcon does not support the COMEXIO system yet. The parameterisation of IP-Symcon and the creation of system variables, scripts and the “WebFront” via the management console, as well as the programming of the building automation via web interface will be described in detail in this thesis.

Inhaltsverzeichnis

1	Einleitung.....	1
2	Grundlagen	2
2.1	Gebäudeautomation	2
2.1.1	Feldebene	4
2.1.2	Automationsebene.....	4
2.1.3	Managementebene	4
2.2	Smart Meter	5
2.3	Gebäudeleittechnik	5
2.4	Comexio gmbh.....	6
2.4.1	Firmenporträt	6
2.4.2	Bussystem.....	6
2.4.3	Programmierung des Bussystems.....	7
2.4.4	Die WebIO	10
2.4.5	COMEXIO Monitoring	12
2.4.6	Der Visu-Editor	13
2.5	IP-Symcon	13
2.5.1	Scriptsprache: PHP	14
2.6	http:Request-Methoden	14
3	Aufgabenstellung.....	16
3.1	Bestehender Versuchsaufbau	16
3.2	Aufgabe des neuen Versuchsaufbaus	17
4	Realisierung des Projekts	19
4.1	Anpassung der Gebäudeautomation: COMEXIO.....	19
4.1.1	API-Einrichtung der IO-Server	19
4.1.2	Programmierung der WebIO	19
4.1.3	Verwendung der WebIO-Befehle	22
4.1.4	Visualisierung des COMEXIO-Systems.....	23
4.2	Konfigurierung von IP-Symcon	26
4.2.1	Daten senden (IPS -> COMEXIO)	27
4.2.2	Datenabrufen (IPS <- COMEXIO)	31
4.2.3	Daten empfangen (COMEXIO -> IPS).....	34
4.2.4	Variablen Visualisierung	37
5	Funktionsumfang des Projektes	39
5.1	SmartMetering	39

5.2	Anwesenheit.....	42
5.2.1	Automation durch Tasker.....	43
5.3	Zugriff auf die Gebäudeleittechnik.....	44
5.3.1	APPs.....	45
5.3.2	Browser.....	45
6	Ergebnis.....	47
7	Anhang.....	i
7.1	Programmierung.....	i
7.1.1	COMEXIO Funktionsplan & WebIO.....	i
7.1.2	IP-Symcon Quellcode.....	iv
7.1.3	Tasker Application.....	viii
7.2	Schaltpläne.....	ix
8	Abkürzungen.....	xi
9	Literaturverzeichnis.....	xii

1 EINLEITUNG

Das technische Facility Management (kurz: FM) bekommt in der heutigen Zeit eine immer größere Bedeutung. Dabei steht im Vordergrund, die Effizienz und Produktivität von Gebäuden und Prozessen zu verbessern und Kosten zu sparen. Das Betreiben, Instandhalten und technische Optimieren von Gebäuden oder Anlagen gehört zu den Aufgaben des FM. Durch das Gebäudemanagement soll zudem auf das Umfeld des Menschen Rücksicht genommen werden. Mittels einer Steigerung von Komfort und Sicherheit, werden somit Produktivität und Wohlbefinden des Menschen am Arbeitsplatz erhöht. Die Hauptaufgabe ist es, somit Kosten durch Optimierungsprozesse einzusparen, Abläufe zu managen, Qualität sowie Zufriedenheit zu verbessern und dabei den Komfort und die Sicherheit des Nutzers zu gewährleisten. Das Facility Management kann dabei von einer Gebäudeautomation, Brandmeldeanlage und Videotechnik unterstützt werden. Alle technischen Einrichtungen lassen sich durch die Gebäudeautomation über ein Gebäudeleitsystem (Gebäudeleittechnik) steuern, regeln und überwachen. Dadurch ist die Gebäudeautomation ein wichtiger Bestandteil des technischen Facility Managements.

Um ein umfangreiches Gebäudeleitsystem aufbauen zu können und somit den Benutzern Arbeit abzunehmen und dadurch automatisch einen höheren Komfort zu leisten, müssen alle verwendeten Einrichtungen miteinander vernetzt werden. Zudem muss die Kommunikation zwischen denselben oder unterschiedlichen Systemen gewährleistet sein. Im Rahmen dieser Bachelorarbeit soll die Kooperation zwischen zwei Servereinheiten der Gebäudeautomations-Firma COMEXIO GmbH im Zusammenhang mit der Gebäudeleittechnik-Software IP-Symcon untersucht werden.

2 GRUNDLAGEN

2.1 GEBÄUDEAUTOMATION

Unter Gebäudeautomation (kurz: GA) versteht man die Gesamtheit an technischen Einrichtungen in einem Gebäude, welche von Logikeinheiten gesteuert werden können. Dabei lassen sich Gebäudeautomations-Systeme in drei Ebenen unterteilen: die Management-, Automations- und Feldebene (siehe Abs. 2.1.1-2.1.3).

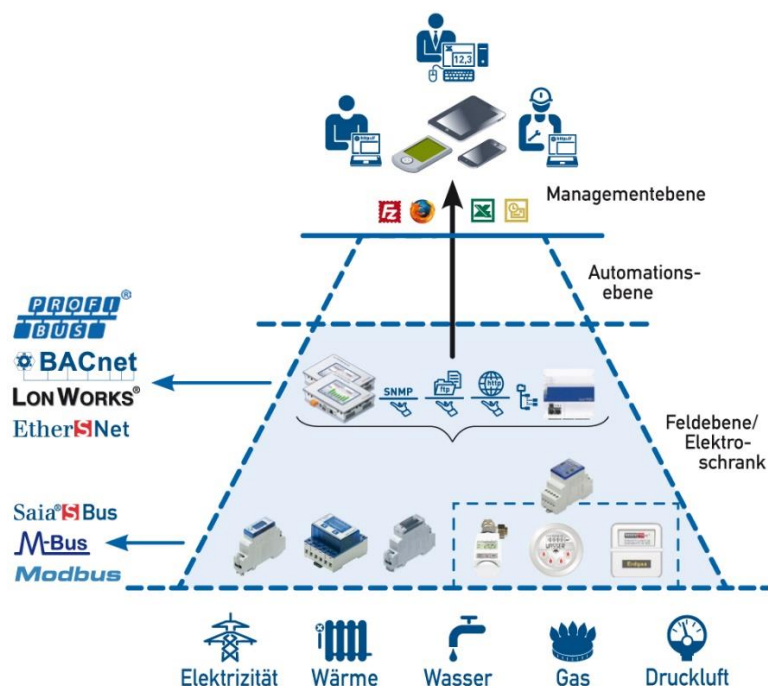


Abbildung 2.1: AUTOMATIONSPYRAMIDE

Durch die verschiedenen Schwerpunkte wird zwischen Industrie- und Heimautomation zwar unterschieden, allerdings werden in beiden Fällen dieselben Gebiete (Steuer-, Regel-, Überwach- und Optimierung) abgedeckt. Mit entsprechenden Überwachungseinrichtungen lassen sich Gebäudeteile, Zustandsinformationen und Zugänge durchgehend überwachen um die Gesamtsicherheit des Gebäudes zu erhöhen. Die Automation kann selbstständig und schnellstmöglich auf bestimmte Situationen reagieren um z.B. die Betriebssicherheit von technischen Anlagen zu gewährleisten. Das selbständige Erkennen von An- bzw. Abwesenheit, sowie das Erkennen von Feuer und Einbruch führen ebenfalls zu mehr Sicherheit.

Die aufgenommenen Informationen der Überwachungseinrichtungen können von der GA verwendet werden um auf bestimmte Szenarien entsprechend zu reagieren. So lässt sich u. a. die Klimatisierung oder Beleuchtung durch die Erfassung von Präsenz oder durch Kontakten in Fenstern und Türen, bedarfsgerecht optimieren. Dabei regelt und steuert das System selbständig, die technischen Einheiten und Verbraucher, unter Berücksichtigung von Sensoren und Parametern. Durch das Erfassen von Daten, wie Zeiten, Lasten, Eingangsströmen, Temperaturen, etc. lassen sich automatisch Verbrauchsprofile erstellen. Die Profile werden dann an das Gebäudeleitsystem übermittelt. Dort werden sie archiviert und/oder grafisch zugänglich gemacht um z.B. den Verlauf des Stromverbrauchs anhand von Diagrammen nachvollziehen zu können. Wird mithilfe der Gebäudeautomation ein „Smart Meter“ (siehe Abs.: 2.2) eingerichtet, lässt sich zu dem Verbrauchsprofil auch noch der tatsächliche Energieverbrauch und die dadurch entstehenden Kosten ermitteln. Diese Kosten können durch eine Gebäudeautomation deutlich reduziert werden, indem Witterungsbedingungen und Jahreszeiten dazu genutzt werden, um z.B. die Heizung oder Beleuchtung bedarfsgerecht zu regeln. Dies kann zu einer Steigerung der Energieeffizienzklasse eines Gebäudes führen.

Die Gesamtheit der Gebäudeautomation bietet nicht nur Sicherheit und Optimierungsmöglichkeiten, sondern auch ein hohes Maß an Flexibilität. Das Zusammenspiel zwischen den Erfassungs-, Steuer- und Regeleinrichtungen kann bedarfsorientiert in der Logikeinheit (Master bzw. Server) programmiert werden. Diese individuelle Anpassung des Systems kann jederzeit erweitert oder neu nach anderen Vorgaben realisiert werden. Durch diese Flexibilität ist es möglich, die Nutzung von Gebäuden, Räumen und Anlagen variabel zu gestalten. Ohne eine neue Verkabelung von Sensoren und Aktoren vornehmen zu müssen, lassen sich Bereiche auf neue Bedürfnisse einstellen. Lediglich die Zuordnung der benötigten Sensoren und Aktoren sind per Software zu ändern. Bereiche können ohne großen Kostenaufwand für verschiedenste Zwecke verändert werden.

2.1.1 FELDEBENE

Die unterste Ebene eines GA-Systems ist die Feldebene. Dort werden Informationen über Sensoren aufgenommen und über geeignete Bus-Systeme als Datentelegramme an die Aktoren übermittelt. Die Aktoren setzen daraufhin die empfangenen Daten in ein Schaltsignal um. Die Feldgeräte (Sensoren und Aktoren) verarbeiten die aufgenommen Informationen und stellen diese den übergeordneten Ebenen zur Verfügung.

Die Grenze zur nächst übergeordneten „Automationsebene“ lässt sich heutzutage nicht mehr eindeutig definieren, da mittlerweile auch Sensoren und Aktoren mit eigener Logik versehen werden.

2.1.2 AUTOMATIONSEBENE

In der Automationsebene findet die Steuer- und Regelung der technischen Einheiten statt. Dabei werden die Vorgaben der Managementebene und die Informationen der Feldebene berücksichtigt. Die Logikeinrichtungen übernehmen zudem noch die Überwachung von Werten und Zuständen um diese in die Automation einzubinden. Die Kommunikation der Master-Einheiten wird ebenfalls auf dieser Ebene beschrieben, sie kann allerdings auch über die Managementebene laufen. Dies ist abhängig von der Vernetzung der Systeme.

2.1.3 MANAGEMENTEBENE

Durch die Managementebene lassen sich Anlagen übergeordnet bedienen und beobachten. Alle Prozesse und Daten werden auf dieser Ebene überwacht um die Betriebsweise ggf. zu optimieren. Spezielle Software (Gebäudeleittechnik) wird hierbei genutzt um die gewünschten Informationen und Daten visuell darzustellen. Die Software läuft für gewöhnlich auf Computern oder kleinen Servern. Dort werden die Daten dann für die Visualisierung zusammengetragen und gespeichert. Die Managementebene hat zudem die Aufgabe eine anlagenübergreifende Kommunikation zu realisieren. So lassen sich z.B. durch Gateways unterschiedliche Systeme miteinander verbinden. Der Zugriff über das Internet oder Netzwerk wird ebenfalls durch diese Ebene ermöglicht. Dadurch kann eine zentrale Regel- und Steuerung ermöglicht werden.

2.2 SMART METER

Bei einem Smart Meter handelt es sich um einen intelligenten Zähler, welcher ähnlich wie ein elektronischer Zähler den Verbrauch von elektrischer Leistung in einem Gebäude bzw. Haushalt misst. Allerdings arbeitet dieser Zähler digital und speichert die Messungen detailliert ab. Dadurch lassen sich wiederum Verbrauchsprofile erstellen, in denen man erkennen kann, zu welcher Zeit wieviel Strom verbraucht worden ist.

Diese intelligenten Zähler verfügen zudem über eine Kommunikationsschnittstelle, die es ihnen ermöglicht, mit dem Internet verbunden zu werden. Über das Internet soll dann eine Kommunikation mit dem Netzbetreiber stattfinden, um ein intelligentes Stromnetz (Smart Grid) aufzubauen, so dass Energie bedarfsgerechter zur Verfügung gestellt werden kann. Ebenfalls soll der Smart Meter über Preisinformationen informiert werden. Herrscht eine Überproduktion von Energie im Netz, so kann der Preis z.B. gesenkt werden um die Nachfrage zu erhöhen. Durch ein Gebäudeleitsystem werden die elektrischen Verbraucher miteinander Vernetzt und lassen sich somit zu gewünschten Zeiten und Preisen aktivieren.

2.3 GEBÄUDELEITTECHNIK

Die Gebäudeleittechnik (kurz: GLT) ist ein Bestandteil der Gebäudeautomation und wird der Managementebene zugeordnet. Hierbei handelt es sich um übergeordnete Software zur Überwachung und Steuerung von Gebäuden oder Anlagen. Dabei stellt die GLT die Zustände und Informationen der Gebäudeautomation(en) dem Benutzer zur Verfügung. Alle Vorgänge, welche von der Automation erfasst oder getätigt werden, können in der Software visualisiert dargestellt werden. Das Überwachen bzw. Steuern kann dabei gebäude- sowie gebäudeautomationsübergreifend stattfinden. Die Software kann somit zentral die dezentralen Gebäudeautomationen sowie den Datenaustausch zwischen verschiedenen Bussystemen verwalten bzw. ermöglichen. Die Bussysteme müssen dabei geeignete Programmierschnittstellen (API) zur Verfügung stellen, damit der GLT-Software eine Anbindung an das System zur Verfügung steht. Durch eine herstellerunabhängige Software (z.B. IP-Symcon, Abs.: 2.5) können somit verschiedenste Gewerke miteinander kommunizieren. Die Software läuft dabei meist auf einer Servereinheit welche über das Netzwerk mit den anderen

Logikeinheiten der Gebäudeautomationen verbunden ist. Eine Anbindung an das Internet ermöglicht sogar eine Verwaltung aller integrierten Anlagen weltweit. Auf dem Server werden alle gewünschten und relevanten Daten gespeichert und sind somit jederzeit abrufbar. Dadurch lassen sich Verbraucherprofile bzw. Smart Meter einrichten, indem die Software mit den entsprechenden Daten (Zeit, Last, Strom, etc.) versorgt wird.

Mit der GLT lassen sich u.a. spezifische Nutzerinterfaces auf grafischen Endgeräten anzeigen, welche mit unterschiedlichen Rechten auf das System zugreifen können.

2.4 COMEXIO GMBH

2.4.1 FIRMENPORTRÄT

„Die Firma Comexio GmbH mit Sitz in Kerzenheim ist ein innovatives eigenständiges Unternehmen aus der Greiner Gruppe, welche sich auf die Entwicklung und Vermarktung von innovativen Produkten rund um Smart Home und Smart Metering für das Eigenheim und Gewerbe konzentriert. Zum Einsatz kommen hier sogenannte Automations-Server, die über den meistverbreiteten Bus Standard Ethernet gesteuert werden. Als wahrscheinlich erste Geräteserie ihrer Art sind diese voll webbasiert. Mit diversen Erweiterungen können die Server nahezu beliebig um Ein- und Ausgänge und Schnittstellen erweitert werden. Die Vision ist es, ein herstellerübergreifendes System zu etablieren, welches im Schwerpunkt benutzerfreundlich und gleichzeitig im höchsten Maße flexibel ist.“ (COMEXIO GmbH 2015)

2.4.2 BUSSYSTEM

Das COMEXIO Bussystem wird durch den IO-Server aufgebaut, dies ist die Logikeinheit der Automation und liefert schon einige Anschluss- und Visualisierungsmöglichkeiten. Er verfügt über Ein- und Ausgänge und unterstützt sogar durch eine EIB/KNX-Schnittstelle deren Standard. Des Weiteren bietet die Mastereinheit die Möglichkeit einen Eindraht-Bus (1-Wire) direkt zu betreiben. Dieser dient dem Aufbau einer günstigen Temperaturmessung durch entsprechende Sensoren. Auf den IO-Server wird über die Ethernet-Schnittstelle mit Internet Browsern zugegriffen und ebenfalls programmiert. Um die Mastereinheit zu erreichen, muss dessen IP-Adresse in der Adressleiste des Browsers aufgerufen werden. Nach dem Einloggen mit Benutzernamen und Kennwort ist es möglich, die vom Router vergebene Adresse durch eine manuelle zu ändern. Dieser Schritt ist zwingend notwendig, sobald

sich mehrere IO-Server in einem Netzwerk befinden. Es kann sonst zu IP-Konflikten führen, sofern sich die Server noch im Werkszustand befinden. Dafür muss vom zuerst angezeigten „Dashboardbereich“ auf den „Adminbereich“ gewechselt werden. Im Dashboardbereich lassen sich visuelle Grafiken und Taster anzeigen bzw. bedienen. Der Adminbereich dient der Konfiguration und Verwaltung des Systems. Dort wird die Einrichtung des Dashboards, die Programmierung der Funktionen, Verwaltung weiterer Module sowie zusätzlicher Einstellungen vorgenommen.

Seit der Firmware-Version 2.7, ist es dem System möglich über Ethernet und somit über das Netzwerk mit anderen netzwerkfähigen Systemen zu kommunizieren. Dazu wurde eine API zur Verfügung gestellt, welche es ermöglicht über sog. „http-Requests“ (siehe Abs.: 2.6 „http: Request Methoden“) auf die Master-Einheit (IO-Server) zuzugreifen. Seit dieser Version ist es den IO-Servern somit möglich, Daten miteinander auszutauschen.

Der Bus kann durch Erweiterungen, wie die IO-Extension ergänzt werden. Die Extension bietet dabei ähnlich dem IO-Server Schnittstellen für EIB/KNX und 1-Wire, sowie weitere Ein- und Ausgänge. Mit der Extension lässt sich ein intelligenter Stromzähler einrichten, da jeder Ausgang der COMEXIO-Module über eine Strommessung verfügt. Die Eingänge dieses Moduls müssen dafür auf Analog gestellt werden um die Phasenverschiebung (Φ) erkennen zu können. Diese wird dazu benötigt die Schein-, Blind- und somit die Wirkleistung zu bestimmen.

Durch die „EnOcean Unit“ kann der gleichnamige Standard, der batterielosen Funkkomponenten empfangen und direkt auf dem COEMXIO Bus verarbeitet werden. Es wird dazu keinerlei weitere Hard- bzw. Software benötigt. Konfiguriert wird dieses Modul ebenfalls mit der Server-Einheit über einen Browser.

Durch Universal- und LED-Dimmer lassen sich Halogen-, Glüh- Energie- oder LED-Leuchtmittel über die Automation steuern und regeln.

2.4.3 PROGRAMMIERUNG DES BUSSYSTEMS

Die Programmierung einer COMEXIO Automation wird über die vom IO-Server zur Verfügung gestellte Weboberfläche realisiert. Die Mastereinheit wird dafür per Ethernet mit einem browserfähigen Gerät verbunden. Einzige Voraussetzung ist, dass im Browser JavaScript aktiviert ist. Folgende Browser werden dabei von COMEXIO unterstützt:

Für die Visualisierung:

- Safari (ab Version 5)
- Firefox (ab Version 15)
- Opera (ab Version 12)
- Internet Explorer (ab Version 5)
- Chrome (ab Version 18)

Für die Programmierung:

- Firefox (Empfohlen) (ab Version 12)
- Chrome (ab Version 18)
- Safari (ab Version 5)“

(COMEXIO Inbetriebnahme Handbuch Version 3.0 V20)

Die COMEXIO OS ist in zwei Bereiche aufgeteilt. Im „COMEXIO Studio“ (Adminbereich) werden die Einstellungen und Konfigurationen für die Gebäudeautomation vorgenommen. Der „COMEXIO Grafik“-Bereich (Boardbereich) dient der Verwendung der im Studio angelegten Visualisierungen. Auf diesen Bereich wird unter Abschnitt 2.4.6 näher eingegangen.

Im Adminbereich unter dem Menüpunkt „Logik“, wird die Programmierung der Funktionen vorgenommen. Dabei stehen zwei Varianten zur Verfügung. Die sog. Anweisungsliste (kurz AWL) wird verwendet um in Scriptsprache (hier COMEXIO AWL) komplexere Algorithmen zu erstellen. Hierbei sind leichte Programmierkenntnisse empfehlenswert. Es sollten die zur Verfügung stehenden Befehle, sowie die Syntax der Sprache bekannt sein. Um die Übersicht zu behalten, können mehrere Anweisungslisten hinzugefügt werden. Es muss in diesem Fall darauf geachtet werden, dass Ausgänge wie auch Merker nicht in anderen Scripten verwendet worden sind. Beim Aktivieren der AWL, überprüft das System die Gültigkeit der Programmierung. Bei Syntaxfehlern, Schreiben auf einem Eingang und Verwendung von nicht definierten Merkern, kommt es zu einer Fehlermeldung und das ausgewählte Script wird gespeichert, aber inaktiv gesetzt.

Mit Hilfe des Funktionsplans (kurz: FUP) lassen sich die gewünschten Funktionen grafisch durch Logikbausteine zusammensetzen. Dabei stellt das System eine Vielzahl an Bausteinen zur Verfügung. Durch ein Infofenster können die verschiedenen Eigenschaften dieser Elemente angesehen bzw. nachvollzogen werden. Auch hier

lassen sich Zusatzbereiche hinzufügen um eine Aufteilung bzw. Gliederung zu ermöglichen. In jedem Bereich lassen sich die Funktionsbausteine aus der rechten Modulleiste per Drag & Drop in die Bearbeitungsfläche hineinziehen. Zur Übersicht und schnellerem Finden der gewünschten Bausteine, sind diese in entsprechende Kategorien aufgeteilt. Befinden sich die für die Funktion benötigten Bausteine in der Arbeitsfläche, können diese durch Anklicken der „Pfeile“ oder „Punkte“, am Anfang oder Ende eines Elements, mit Linien miteinander verbunden werden (vgl. Abb.:2). Kann über diese Verbindung ein Analogwert gesendet werden, so handelt es sich um einen Punkt. Handelt es sich wiederum um digitale Signale (0 oder 1), so wird ein Pfeilsymbol verwendet. Es muss darauf geachtet werden, dass Analog- und Digitalwerte nicht miteinander verbunden werden. Sind alle Komponenten wie gewünscht miteinander verbunden, entscheidet der Logikbaustein was mit eingehenden Signalen oder Werten geschehen soll. Im Anschluss muss nur noch der bzw. die gewünschten Funktionspläne aktiviert werden. Daraufhin kann keine Bearbeitung mehr stattfinden, erst nach Deaktivierung ist dies wieder möglich. In dem Menüreiter „Logik“ lassen sich u.a. eigene Bausteine (Szenenbausteine), Kalenderfunktionen, Zeitglieder, die WebIO, etc. nach eigenen Vorgaben definieren. Auch die Verwendung der verschiedenen Schnittstellen (z.B. KNX, 1-Wire) wird dort konfiguriert.

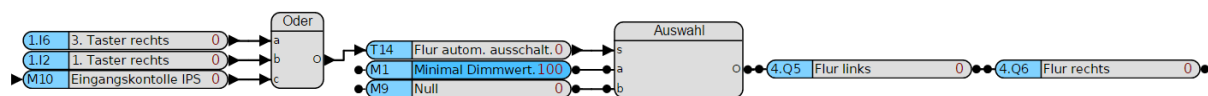


ABBILDUNG 2.2: COMEXIO FUNKTIONSPLAN

Die Grafische-Programmierung bietet im Zusammenhang dieser Arbeit alle benötigten Funktionen an. Sie verschafft eine leichtere Übersicht über das Programm und lässt sich einfacher verändern bzw. anpassen. Im weiteren Verlauf wird ausschließlich mit dieser Variante gearbeitet, alle durch Logikbausteine realisierten Funktionen lassen sich selbstverständlich auch durch eine AWL-Programmierung realisieren.

2.4.4 DIE WEBIO

Über die „WebIO“ wird die Kommunikation mit weiteren netzwerkfähigen Geräten verwaltet. Hier werden die gewünschten Geräte mit Hilfe ihrer IP-Adresse angelegt und die Webbefehle definiert bzw. programmiert. Die Schnittstellen dieser Geräte müssen Anfragen über „http-Requests“ unterstützen. Dadurch lassen sich z.B. Netzwerkkameras oder Multimediasysteme mit der Gebäudeautomation verbinden.

Die WebIO lässt sich im Adminbereich des IO-Servers unter dem Menüpunkt „Logik“ konfigurieren. Da jeder Teilnehmer im Netzwerk eine eigene IP-Adresse hat, muss für jedes Gerät ein extra Reiter angelegt werden. Beim Hinzufügen muss ein Name und die zum Gerät gehörige IP-Adresse vergeben werden. Unter dem Punkt „Typ“ wird der Gerätetyp festgelegt. COMEXIO bietet dort schon ein paar Systeme, wie z.B. das drahtlose Soundsystem „SONOS“, an. Die WebIO zeigt bei unterstützten Typen die zur Verfügung stehenden Funktionen direkt an. In diesen Befehlen sind alle benötigten Parameter und Programmierungen schon vordefiniert. Zusätzlich lassen sich auch neue Gerätetypen erstellen. Dieser Vorgang ist u.a. notwendig, wenn zwei COMEXIO Bussysteme miteinander kommunizieren sollen. Nach dem Anlegen des neuen Gerätes lassen sich die Funktionen bzw. Befehle einzeln hinzufügen. Das Bearbeitungsfenster sieht dabei wie folgt aus:

The screenshot shows a web-based configuration window titled "Den Webbefehl Ausgang 1 bearbeiten". The fields are as follows:

- Name:** Ausgang 1
- Variablen-Typ:** Digital (dropdown menu)
- ☐ Nur bei einer logischen 1 senden
- Minimaler Wert:** 0
- Maximaler Wert:** 1
- Parameter:**

```
function parameter (v)
return '/api/?action=set&text=COMEXIO-
Server&io=Q5&value='..v
end
```
- Header Modifizierung:** (empty text area)
- Html-Daten:** (empty text area)
- Typ:** Ausgang (dropdown menu)
- TCP-Port:** (empty text area)
- Aufrufmethode:** GET (dropdown menu)
- Authentifizierung:** Nein (dropdown menu)
- Wiederholfrequenz:** (empty text area)
- Antwortinterpretier:** (empty text area)

At the bottom, there are two buttons: "Testen" and "Speichern".

ABBILDUNG 2.3:WEBBEFEHL BEARBEITUNG

Der „Variablen-Typ“ bestimmt, ob durch diesen Befehl ein Analog- oder ein Digitalwert übertragen werden soll. Bei Ausführung eines Webbefehls werden somit http-Anfragen an die entsprechende Schnittstelle gesendet. Dabei muss der Befehl, wie unter „HTTP: Request-Methoden“ beschrieben, aufgebaut sein. Damit die Kommunikation stattfinden kann, müssen im Feld „Parameter“ die benötigten Konstanten bestimmt werden. Bei manchen Geräten sind „Header Modifizierung“ und/oder „Html-Daten“ notwendig, um mit diesen Gerät kommunizieren zu können. Diese Parameter lassen sich ebenfalls in den entsprechenden Feldern eintragen. In den Bereichen kann direkt in String geschrieben werden oder aber in der Scriptsprache „Lua“, falls mehr Logik benötigt wird. Das System erkennt ein Script in dem es wie folgt beginnt:

function parameter	bei Parameter
function header_modifier	bei Header Modifizierung
function data	bei Html-Daten
function reply_interpreter	bei Antwortinterpreter

(COMEXIO Inbetriebnahme Handbuch Version 3.0 V20)

Der Typ des Befehls (Eingang oder Ausgang) bezieht sich dabei auf das Bussystem wo dieser angelegt wird. So steht nach dem Anlegen eines Web-Ausgangs in den Logikbausteinen ein Ausgang zur Verfügung, welcher Signale oder Werte erhält und dadurch Aktionen ausführt. Im Gegensatz dazu rufen die Web-Eingänge Informationen von den anderen Systemen ab und geben diese dann an das eigene System weiter.

Falls das Gerät nicht den Standard-Port verwendet, muss unter „TCP-Port“ dieser angegeben werden. Bei den Aufrufmethoden stehen „GET“ oder „POST“ zur Verfügung, so dass ein GET- oder POST-Request verwendet werden kann. Bei der Verwendung von Web-Eingängen müssen die Abfragen regelmäßig durchgeführt werden. Dafür wird das Feld „Wiederholungsfrequenz“ genutzt. Dieses gibt an, in welchen Zeitintervallen eine Abfrage getätigt werden soll. Ohne eine Angabe würde die Abfrage der Information nicht stattfinden oder gar wiederholt werden. Der „Antwortinterpreter“ muss ebenfalls unbedingt definiert werden. Dieser übersetzt die Antwort des Servers (String) in Zahlenwerte, die vom System verstanden bzw. weiter verwendet werden können. So lassen sich u.a. auch Daten aus Textdateien (.txt) abrufen.

Sofern alle Einstellungen und Programmierungen vollständig sind besteht die Möglichkeit den Webbefehl vor dem Speichern zu testen. Dabei können beliebige Werte angegeben und übertragen werden. Der Aufbau und die Funktionen werden im weiteren Verlauf unter Abschnitt 2.6 genauer beschrieben. Nach dem Speichern des Webbefehls können weitere Befehle für dieses oder andere Geräte angelegt werden.

Jeder angelegte Befehl steht im Anschluss je nach Deklaration in der grafischen Programmierung als Ein- bzw. Ausgang zur Verfügung. Dieser kann ebenfalls per Drag & Drop in das Bearbeitungsfenster gezogen und mit anderen Bausteinen verknüpft werden.

2.4.5 COMEXIO MONITORING

Im Bereich Monitoring werden Graphen für den COMEXIO-Studiobereich erstellt. Diese können dazu verwendet werden um aktuelle oder zeitabhängige Werte darzustellen. So lassen sich Temperaturen, Stromverbräuche oder Schaltzeiten etc. in einem Diagramm anzeigen. Für jedes Monitoring steht ein Diagramm zur Verfügung in dem mehrere Datenpunkte angezeigt werden können. Diese können farblich unterschieden werden. Eine eingeblendete Legende ordnet den Farben die zugehörigen Namen zu und ermöglicht somit eine bessere Übersicht. Ebenfalls können die Speicherintervalle jedes Datenpunktes je nach Bedarf angepasst werden. Neben den Grundeinstellungen der jeweiligen Grafik, wie Bezeichnung und Rastergröße, stehen noch Anzeige- und Verbrauchsdatenoptionen zur Verfügung. Während unter den Anzeigeoptionen die Darstellung des Diagramms geändert werden kann, kann bei den Verbrauchsdaten die Verarbeitung der erfassten Daten eingestellt werden. Dort lässt sich ein Zähler aktivieren welcher z.B. Schaltvorgänge addiert. Des Weiteren kann ein Durchschnitt der erfassten Werte berechnet bzw. angezeigt werden. Beide Optionen lassen sich dabei zeitlich einschränken, so dass z.B. der Durchschnitt von einer Woche berechnet wird. Alle aufgezeichneten Werte lassen sich zudem durch ein Backup auch auf anderen Endgeräten verwendet werden. Das Backup kann dabei über einen FTP- oder einen E-Mail-Server durchgeführt werden. So lassen sich die Daten von den Servern durch externe Software, wie z.B. Excel, aufarbeiten oder analysieren.

Mit dem Anlegen weiterer Monitorings stehen somit weitere Diagramme zu Verfügung. Diese können daraufhin im „Visu-Editor“ verwendet werden um sie auf dem Dashboard anzuzeigen.

2.4.6 DER VISU-EDITOR

Der Visu-Editor dient einzig und allein der Einrichtung des Dashboard-Layouts. Hier werden u.a. die Monitoring-Grafen auf die gewünschte Seite bzw. Positionen gesetzt. Um den „Studiobereich“ eine gewisse Struktur zu geben und die Übersicht zu verbessern, können mehrere Dashboards angelegt werden. Nach der Vergabe des Namen und der Größe können in das neu angelegte Fenster die zur Verfügung stehenden Symbole bzw. Module per Drag & Drop hineingezogen werden. Nach einer Verknüpfung mit einem Ein- bzw. Ausgang oder Merker, dienen die Symbole als visuelle Anzeige- oder Schaltflächen. Dadurch lassen sich Aktionen durch Verwendung des Dashboard darstellen oder durchführen.

2.5 IP-SYMCON

IP-Symcon (kurz: IPS) ist ein herstellerunabhängiges Gebäudeleitsystem, welches die Anbindung verschiedenster Systeme und Geräte an eine flexible Benutzeroberfläche ermöglicht. Dabei trägt die Software alle Informationen der GA zusammen und stellt sie bei Bedarf visuell zur Verfügung. Über das „WebFront“ (für: Webfrontent) von IP-Symcon lassen sich alle verbundenen Geräte, wie die GA oder Multimediasysteme, an jedem Endgerät bedienen und anzeigen. Durch entsprechende Apps für Android und iOS, wird das WebFront auf mobilen Geräten in angepasster Form zur Verfügung gestellt (siehe Abs.: 5.3.1). Das WebFront kann individuell an die jeweiligen Bedürfnisse angepasst werden. IPS legt dafür einen Server an, auf dem die Dienste, wie das „WebFront“, Instanzen, Scripte, etc. laufen und gespeichert werden. Über die Verwaltungskonsole kann die Konfiguration der GLT vorgenommen werden. In der Konsole lässt sich die Struktur des Projektes durch den Objektbaum festlegen und übersichtlich gestalten. Dort werden die Instanzen, Systemvariablen, Funktionen, etc. an gewünschter Stelle hinzugefügt und konfiguriert.

Die Daten von verschiedenen Systemen lassen sich in IPS, durch die Unterstützung von vielen Herstellern besonders gut erfassen. Da viele Hersteller unterstützt werden, ist die Konfiguration mit diesen Systemen besonders komfortabel. Durch die passenden Konfiguratoren können die Module der entsprechenden GA selbstständig gefunden oder manuell ausgewählt werden. Bussysteme welche noch nicht in IPS integriert sind, lassen sich durch entsprechende APIs in die Gebäudeleittechnik einbinden. Durch selbstgeschriebene Programme in der Scriptsprache PHP, können

durch entsprechende Befehle/Funktionen diese Schnittstellen verwendet werden, um somit den Zugriff auf andere Systeme zu ermöglichen. Durch die Einbindung von „HTML“ oder „JavaScript“ in PHP lässt sich durch die Scripte u.a. das WebFront individuell gestalten oder Daten aus dem Internet abrufen und in das System einbringen. Zudem kann damit der Funktionskatalog von IPS erweitert werden. IPS stellt zum Schreiben von Scripten einige Systemfunktionen zur Verfügung, welche das Erstellen von komplexeren Programmen vereinfacht.

2.5.1 SCRIPTSPRACHE: PHP

Hierbei handelt es sich um die Scriptsprache „hypertext Preprocessor“ (kurz: PHP). Diese wird bevorzugt dafür verwendet, um dynamische Webseiten zu programmieren, mit denen wiederum Web Applications (Webanwendungen) realisiert werden können. Die Programmierung der an „C“ und „Perl“ angelegten Programmiersprache ist recht einfach gehalten, da diese Sprache ausschließlich für die Web-Serverprogrammierung entwickelt worden ist und diesbezüglich auch nur die notwendigen Bestandteile enthält. Zudem werden von PHP viele Plattformen und Datenbanken (z.B. MySQL oder JavaScript) unterstützt. So lässt sich u.a. auch sehr einfach HTML in PHP einbetten, welches das Programmieren von Webseiten (des WebFront) wesentlich vereinfachen kann. Dies macht die Sprache flexibel und erhöht ihre Möglichkeiten. PHP ist vollkommen kostenlos zugänglich und so gut wie betriebssystemunabhängig. Es handelt sich hierbei um eine Serverprogrammiersprache, d.h. alle in PHP geschriebenen Programme werden serverseitig ausgeführt, der Client muss die gesendeten Daten nur umsetzen bzw. verstehen können.

2.6 HTTP:REQUEST-METHODEN

Die Programmierschnittstelle die von COMEXIO mit der Version 2.7 zur Verfügung gestellt wurde, wird durch sog. „Http-Request“ verwendet. Von der Vielzahl an Methoden für eine Http-Abfrage stehen zwei davon zur Verfügung. Die beiden Aufrufmethoden GET und POST werden von IO-Servern unterstützt und bieten somit die einfachste Art eines Requests an. Die GET- und POST-Methoden dienen dem Abrufen von Daten zwischen Client und Server, lediglich der Aufbau ist unterschiedlich. Dies muss bei der Konfiguration der WebIO-Befehle berücksichtigt werden. Bei beiden Methoden geht es darum Daten vom Server abzurufen oder ihm zu übergeben. Die benötigten Datenpunkte werden bei einem GET-Befehl als String

in der URL angegeben. Der Post-Befehl sendet diesen String dahingegen in einem sog. „message body“ ab. Die Struktur eines GET-Befehls ist wie folgt aufgebaut:

„*http://<Comexio-Ip>/api/?action=<Aktion>&<Datenpunkt>&value=<Wert>*“

Alle Begriffe innerhalb von „<...>“ sind als Variablen zu betrachten. Sie müssen dem gewünschten Vorhaben angepasst werden. Für „<Comexio-Ip>“ muss die IP-Adresse der Servereinheit eingetragen werden, an der die Informationen gesendet bzw. abgerufen werden sollen. Unter „<Aktion>“ wird festgelegt, ob der gewünschte Wert gelesen („get“) oder geschrieben („set“) werden soll. Das Feld für den Datenpunkt ist abhängig vom Datentyp, dabei wird zwischen Ein- und Ausgängen („ext=<Gerät>&io=<Ein/Ausgang>“), Merkern („marker=<id>) und 1-Wire („onewire=<id>) Daten unterschieden. Beim Zugriff auf die Ein- und Ausgänge, muss für „<Gerät>“ der in der Modulparametrierung verwendete Name eingesetzt werden. „<Ein/Ausgang>“ wird einfach durch die zugehörige Abkürzung wie in der Programmierung, z.B. I2 für Eingang 2 oder Q5 für Ausgang 5, ausgetauscht. Somit kann ein http-Request zum Setzen des Ausgangs 5, der Servereinheit „IO-Server_local“, auf „0“ z.B. lauten:

„*http://192.168.178.50/api/?action=set&ext=IO-Server_local&io=Q5&value=0*“

Diese Anfrage kann durch jeden beliebigen Browser ausgeführt werden. Durch die Eingabe der URL in die Adresszeile des Browsers kann die Servereinheit der Gebäudeautomation abgerufen werden.

Da die GET-Variante einfacher zu verwenden ist, wird im Zusammenhang der Kommunikation zwischen den Bussystemen nur diese Methode verwendet. Auf die POST-Methode wird im späteren Verlauf näher eingegangen, wenn es um die Kommunikation mit IPS geht. Es lässt sich noch erwähnen, dass das POST-Verfahren dann Sinn macht, wenn die URL des GET-Befehls zu lang werden würde (ab 1024) oder ganze Texte übermittelt werden sollen, welche über die max. Zeichenlänge verfügen. Die POST-Methode wird im WebIO verwendet, um z.B. Multimediasystemen mehrere Daten mit nur einem Befehl senden zu können.

3 AUFGABENSTELLUNG

3.1 BESTEHENDER VERSUCHSAUFBAU

Der bestehende Versuchsaufbau entstand 2015 im Rahmen einer Bachelorarbeit von Simon Tack an der Fachhochschule Dortmund. Im weiteren Verlauf dieser Arbeit wird dieses Bussystem auch als „System 1“ bezeichnet. Mit Hilfe von Sensoren, Aktoren und dem COMEXIO Bussystem wurde ein laborfähiger Testaufbau erstellt. Um die Funktionalität und die Programmierung des Systems zu testen, wurde eine LED-Platte mit insgesamt 20 Leuchtmitteln installiert. Diese Platte stellt zwei Büroraume dar und soll veranschaulichen, unter welchen Voraussetzungen etwas geschieht. Die LEDs sind mit den Ausgängen des LED-Dimmer, des IO-Servers und der IO-Extension verbunden. Des Weiteren wurden auf dem Aufbau noch zwei Temperatursensoren (PT100) montiert, um über die IO-Extension die Temperatur in die Automation einbeziehen zu können. Die EnOcean Unit ermöglicht die Verwendung von Funksensoren, die zu den vier Doppel-Tastern dem System weitere Schaltsignale bzw. Impulse übermitteln können. Der montierte Router dient dem IO-Server als Anschlussmöglichkeit an das Netzwerk und erlaubt zudem ein Zugriff über WLAN, sobald Router und Server via Ethernet miteinander verbunden sind. Dieser Aufbau diente allein zu Test- und Präsentationszwecken des COMEXIO Bussystems.

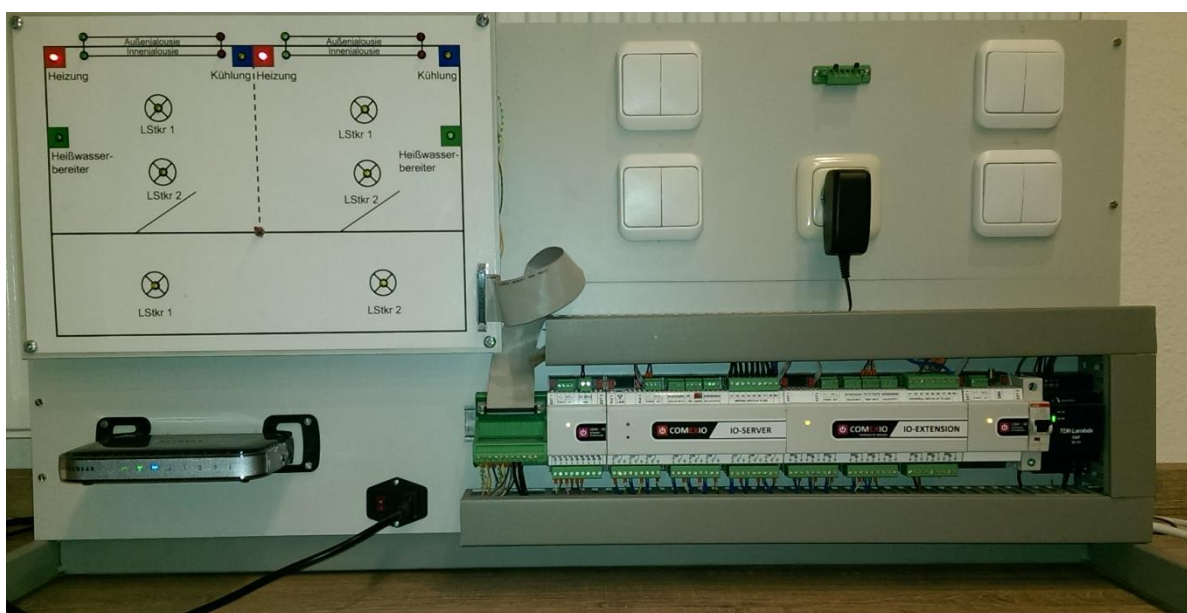


ABBILDUNG 3.1:ALTER VERSUCHSAUFBAU

Dieser Versuchsaufbau zeigt, dass die Verwendung von COMEXIO komfortabel ist. Die programmierten Funktionen wurden anhand dieser Platte getestet. Ein weiterer Punkt, der durch diesen Aufbau deutlich wird, ist das wenige Module verwendet werden müssen, um viele Aufgaben bewältigen zu können. Die COMEXIO Module beschränken sich nicht nur auf ein Gebiet, sondern bieten meistens mehrere Einsatzbereiche an. So kann die Mastereinheit z.B. Aktoren schalten, Sensoren verwalten, 1-Wire Temperatur erfassen und KNX anbinden. Allerdings wurde im Rahmen dieses Projekts erkannt, dass das Bussystem in sich beschränkt ist. Die Kommunikation mit anderen Servereinheiten oder Systemen war zur dieser Zeit noch nicht möglich.

3.2 AUFGABE DES NEUEN VERSUCHSAUFBAUS

Um die Funktionalität der COMEXIO Gebäudeautomation zu erweitern, soll in dieser Arbeit die Kooperation des bestehenden Testaufbaus mit einem weiteren COEMXIO Bussystem („System2“) realisiert werden. Zudem sollen beide Bussysteme über das Gebäudeleitsystem IP-Symcon verwaltet werden. Die Software wird auf das Bussystem zugreifen können, um Daten auf dem IPS-Server speichern und verarbeiten zu können. Über das WebFront von IPS soll zudem das Schalten und Regeln der Automationen möglich sein. Das wesentliche Ziel ist das Erreichen einer Gebäudeleittechnik, welche durch die Vernetzung mehrerer Systeme für den Einsatz in größeren Projekten, wie z.B. Bürogebäuden, gewachsen ist. Das Augenmerk liegt dabei nicht wie bei dem vorherigen Versuchsaufbau auf den Funktionen und Eigenschaften für eine Gebäudeautomation, sondern viel mehr in der Kommunikation mit anderen Systemen. Die Programmierung wurde diesbezüglich neu eingerichtet. Am Ende soll es jedem möglich sein, IO-Server über das Netzwerk bzw. Ethernet auf der Basis von IPS oder anderen übergeordneten Softwarelösungen, mit entsprechender Schnittstelle, miteinander verbinden zu können.

Um die Ergebnisse dieser Arbeit testen und präsentieren zu können, wird zu der vorhanden Automation ein weiterer Testaufbau erstellt. Da für die Kommunikation nur eine weitere Mastereinheit benötigt wird, besteht das neue Bussystem lediglich aus einem IO-Server. Durch die Ein- und Ausgänge ist dieser ausreichend, um die Kommunikation zwischen mehrerer IO-Server zu demonstrieren. Zudem wurden Taster und Aktoren auf dem neuen Testaufbau installiert um Aktionen durchführen

bzw. darstellen zu können. Des Weiteren wird ein externer Server (z.B. Computer) für den Betrieb von IP-Symcon verwendet.

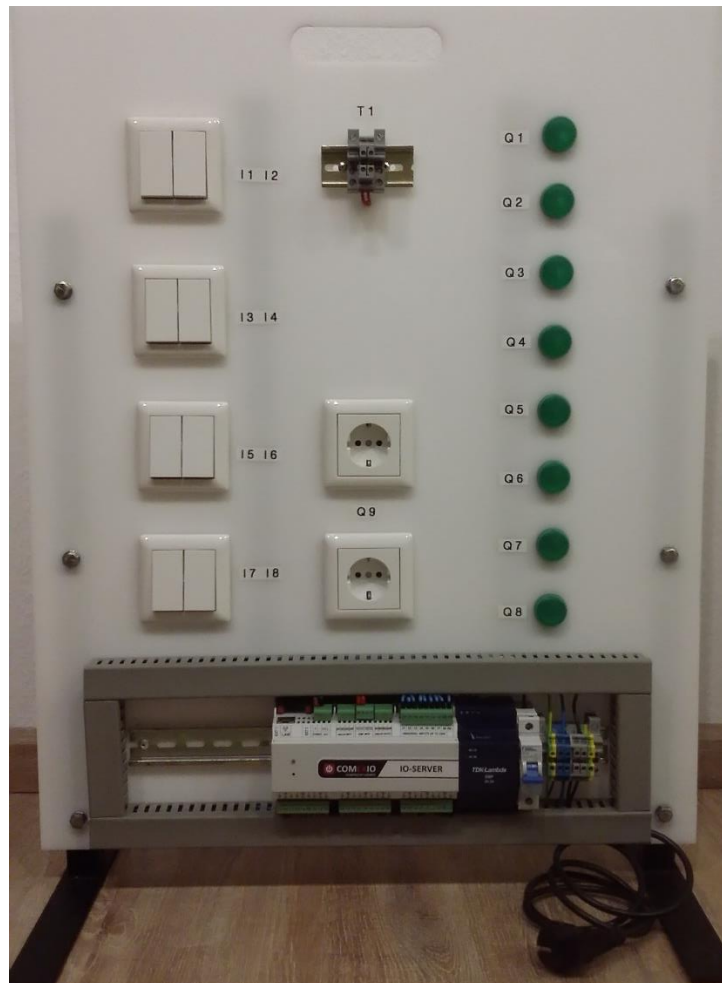


ABBILDUNG 3.2: NEUER VERSUCHSAUFBAU

4 REALISIERUNG DES PROJEKTS

Um die Kommunikation unter den Systemen zu ermöglichen, müssen zusätzliche Einstellungen aktiviert und spezifische Programmierungen getätigt werden. Diese werden nachfolgend genau erklärt und beschrieben. Dabei wird bei der Programmierung hauptsächlich auf die Kommunikation der Systeme Rücksicht genommen. Die vollständige Konfiguration des Bussystems oder die von IP-Symcon kann dem beiliegendem Datenträger entnommen werden.

4.1 ANPASSUNG DER GEBÄUDEAUTOMATION: COMEXIO

4.1.1 API-EINRICHTUNG DER IO-SERVER

Damit die IO-Server untereinander und von anderen Systemen erreichbar sind, so dass Daten gesendet bzw. abgerufen werden können, muss die API jeder Servereinheit konfiguriert werden. In der Weboberfläche des IO-Servers muss dafür im „Adminbereich“ unter „Einstellungen“ der Untermenüpunkt „API“ ausgewählt werden. Dort lässt sich der Zugriff auf die API aktivieren und verwalten. Es ist möglich, die Schnittstelle über Login-Daten vor nicht autorisiertem Zugriff zu schützen. Des Weiteren kann der jeweilige Umgang mit den Datentypen (Ein- und Ausgänge, Merker, 1-Wire) festgelegt werden, um z.B. den Lese- und Schreibzugriff zu managen oder ihn bei Bedarf zu verbieten. Sobald diese Schnittstelle eingerichtet ist, kann über sog. „http-Requests“ auf das Bussystem zugegriffen werden.

4.1.2 PROGRAMMIERUNG DER WEBIO

Wie in Abschnitt 2.4.4 beschrieben, werden zuerst die benötigten Web Ein- und Ausgänge angelegt. Dabei wird bzw. muss, für jeden Wert der gesendet oder empfangen wird, ein separater Web-Befehl angelegt werden. Bei der Übertragung von Bit-Werte wird der „Variablen-Typ“ auf Analog gesetzt. Durch die Angabe von min. und max. Wert wird der Analogwert eingegrenzt (z.B. bei Temperaturen „-20 bis +60“). Werden wiederum nur Zustände und Signale übertragen, wird der Typ Digital verwendet. Hierbei ist es möglich diesen Befehl nur auszuführen, wenn eine „logische 1“ aktiv wird. Bei der Kommunikation zwischen COMEXIO-Systemen ist, wie in Abbildung 6, nur eine Programmierung der „Parameter“ notwendig, die „Header Modifizierung“ und die „Html-Daten“ werden nicht benötigt. Kommuniziert das

Endgerät zudem über einen standardisierten TCP-Port, muss dieser ebenfalls nicht angegeben werden. Bei den Web-Eingängen muss zwingend darauf geachtet werden, dass „Wiederholungsfrequenz“ und „Antwortinterpret“ angegeben sind. Da sonst keine Abfrage des Wertes stattfindet und dieser zudem vom System nicht verstanden werden kann. Durch:

„function parameter (v)“

wird die Variable „v“ festgelegt und dem System mitgeteilt, dass ein Script ausgeführt werden muss. Diese Variable kann Werte annehmen die durch „Minimaler Wert“ und „Maximaler Wert“ begrenzt werden. Durch die Rückgabe („return“) wird der API-Befehl zusammen gesetzt und an die zuvor angegebene IP-Adresse angehängen.

„return'/api/?action=set&ext=IO-Server_local&io=Q5&value='..v'“

Innerhalb der einfachen Hochkommata (' ') steht der String, welcher die zu übertragende Aktion festlegt. Am Ende wird noch der Parameter „v“ an den Befehl angehängt um einen Wert (hier: 0 oder 1) senden zu können. Das Ende des Scriptes wird durch ein „end“ definiert. Beim Abrufen von Werten über die Schnittstellen kann der API-Befehl auch ohne Scriptsprache unter „Parameter“ eingetragen werden, wie z.B.:

„/api/?action=get=IO-Server_local&io=I1“.

Dieser Befehl ist ausreichend um den Status vom ersten Eingang des IO-Servers „IO-Server_local“ abzufragen. In „Wiederholungsfrequenz“ können normale Zahlenwerte verwendet werden. Dabei ist zu beachten, je kürzer die Frequenz, desto öfter wird der Bus belastet. Bei einer großen Menge an Abfragen mit häufiger Wiederholung kann es somit zu Verzögerungen auf dem Bussystem kommen. Beim Auslesen von Werten bzw. Ein- oder Ausgängen von anderen System (wie z.B. 1-Wire Temperaturen), können die Werte (Strings) nicht direkt vom System verstanden werden. In solchen Fällen wird der „Antwortinterpret“ programmiert um den Wert für das System umzuwandeln („+45“ zu „45“). Dieser ist wie folgt aufgebaut:

“function reply_interpreter (v)

return tonnumber (v)

end”

Durch diese Programmierung wird die erhaltene Antwort des Server von einem String, also eine Zeichenkette, in eine dem System verständliche Zahl umgewandelt (z.B. float

oder integer). In den meisten Fällen macht es wenig Sinn, direkt Ausgänge zu steuern oder zu schalten. Dem externen System ist ohne eine zusätzliche Web-Abfrage, der Zustand des Ausgangs nicht bekannt. So kann es dazu führen, dass ein Tastendruck nichts bewirkt, erst beim zweiten Mal eine Aktion ausgeführt wird oder der Ausgang durch Funktion dauerhaft ein bzw. ausgeschaltet bleibt und vom jeweils anderen System nicht mehr beeinflusst werden kann. Wenn Ausgänge von mehreren Systemen gesteuert werden sollen, werden in dieser Arbeit nur Web-Ausgänge verwendet, welche auf Merker zugreifen. Diese Merker werden dann in der Programmierung des entsprechenden Systems verwendet um die gewünschte Aktion auszuführen. Da auf einen Merker geschrieben werden soll, wird in diesem Fall ebenfalls ein Web-Ausgang angelegt. Die Syntax der Parametrierung ist dabei dieselbe wie bei den Ausgängen. Nur der Befehl muss anders aufgestellt werden:

***„function parameter (v)
return'/api/?action=set&marker=M20&value='..v
end“***

Bei Merkern muss kein Gerätenamen angegeben werden, da die Merker zum Gesamtsystem gehören und dieses wird durch die IP-Adresse festgelegt.

Durch die programmierten Scripte wird somit der http-Request zusammengesetzt. Wie in Abs.: 2.6 wird der Befehl an die IP-Adresse angehängt und somit am gewünschten Server ausgeführt.

The image shows two side-by-side web configuration windows. The left window is titled 'Den Webbefehl Onboard Temp. bearbeiten' and the right window is titled 'Den Webbefehl Ausgang 1 bearbeiten'. Both windows have a similar layout with various input fields and buttons.

Left Window: Den Webbefehl Onboard Temp. bearbeiten

- Name: Onboard Temp.
- Variablen-Typ: Analog
- Minimaler Wert: -10
- Maximaler Wert: 100
- Parameter: /api/?action=get&ext=COMEXIO-Server&io=TL1
- Header Modifizierung: (empty)
- Html-Daten: (empty)
- Typ: Eingang
- TCP-Port: (empty)
- Aufrufmethode: GET
- Authentifizierung: Nein
- Wiederholfrequenz: 10
- Antwortinterpret: function reply_interpreter (v)
return tonumber (v)
- Buttons: Testen, Speichern

Right Window: Den Webbefehl Ausgang 1 bearbeiten

- Name: Ausgang 1
- Variablen-Typ: Digital
- ☐ Nur bei einer logischen 1 senden
- Minimaler Wert: 0
- Maximaler Wert: 1
- Parameter: function parameter (v)
return'/api/?
action=set&marker=M20&value='..v
end
- Header Modifizierung: (empty)
- Html-Daten: (empty)
- Typ: Ausgang
- TCP-Port: (empty)
- Aufrufmethode: GET
- Authentifizierung: Nein
- Wiederholfrequenz: (empty)
- Antwortinterpret: (empty)
- Buttons: Testen, Speichern

ABBILDUNG 4.1: PROGRAMMAUSZUG WEBIOS

4.1.3 VERWENDUNG DER WEBIO-BEFEHLE

Damit die durch die Webbefehle erhaltenen Werte visuell verwendet werden können, müssen diese zunächst auf Merker geschrieben werden. Auch für das Monitoring lassen sich diese Werte nicht direkt abrufen, sondern müssen zwischengespeichert werden. Da alle angelegten Webbefehle als Logikbausteine zur Verfügung stehen, lassen sie sich einfach in einem Funktionsplan mit Merkern verbinden. Diese lassen sich im Anschluss für die Visualisierung oder im Monitoring des IO-Servers verwenden. Unter den Eigenschaften der im „Visu-Editor“ angelegten Module, wird der Ein- oder Ausgang festgelegt, der mit diesem Modul verbunden sein soll. Dort lassen sich alle Parameter des Bussystems auswählen. Um einen visuellen Taster zu konfigurieren, der eine Aktion auf einem anderen System ausführen soll, wird das entsprechende Modul in die Arbeitsfläche gezogen. Dieses Modul wird daraufhin mit einem Merker verknüpft. Dieser spiegelt somit den Wert oder Zustand des visuellen Elements, da in diesem Zusammenhang das Modul als Taster verwendet wird und durch dessen Betätigung eine Aktion durchgeführt werden soll. Wird dieser als „Visualisierungseingang“ unter dem Eigenschaftspunkt „Funktion“ festgelegt, steht dieser als genau solches im Logikbereich zur Verfügung (z.B. VISU7). Da der Taster eine Aktion auf dem anderen Bussystem ausführen soll, wird dieser mit dem gewünschten WebIO-Befehl im Funktionsplan verbunden. Weil das Betätigen des Tasters ein zu kurzen Impuls weitergibt, muss dieser visuelle Eingang mit Hilfe eines Zeitgliedes entprellt werden (vgl. Abb.: 4.2). Das Zeitglied dient als Ausschaltverzögerung des eingehenden Signals durch den Taster. Dabei reicht eine Verzögerung von 0,1 Sekunden.

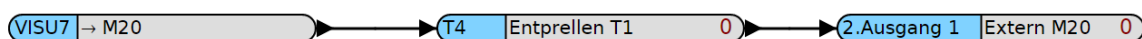


ABBILDUNG 4.2: PROGRAMMAUSZUG SYSTEM 1

Durch die Betätigung des Tasters, auf dem Dashboard, erhält somit der Visualisierungseingang einen Impuls welcher das Zeitglied aktiviert. Dieser überträgt über den WebIO-Ausgang („Ausgang 1“) für 0,1s eine logische „1“. Der Befehl greift auf eine externe API zu und schreibt auf einem Merker (hier: M20).

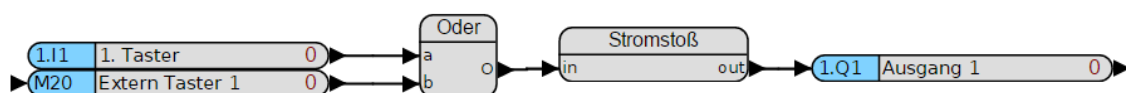


ABBILDUNG 4.3: PROGRAMMAUSZUG SYSTEM 2

Der Merker kann wie ein normaler Taster in der Programmierung der Gebäudeautomation verwendet werden. Dabei liefert er ebenso wie ein herkömmlicher Taster ein Impuls, welcher in diesem Fall die „Stromstoß“-Funktion aktiviert und dadurch den Ausgang ein- oder ausschaltet.

Web-Eingänge werden generell auf Merkern gespeichert, um die Werte überhaupt im Bussystem verwenden zu können. Um den Status bzw. Zustand eines Ausgangs an ein anderes System weiterzuleiten, wird am Ausgang ein Web-Befehl angehängt. Dieser Befehl wird nur dann ausgeführt, wenn eine Änderung stattfindet. Der Bus wird somit nur in diesem Fall belastet.

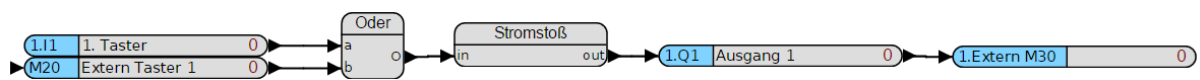


ABBILDUNG 4.4: PROGRAMMAUSZUG SYSTEM 2

Der vollständige Funktionsplan um einen Ausgang vom eigenen und externen Bussystem zu steuern, kann wie in Abb. 4.4 realisiert werden. Der Merker M30 kann dabei vom externen System verwendet werden, um weitere Aktionen bzw. Funktionen durchzuführen (z.B. Statusanzeige auf Dashboard). Bei Werten die sich häufig ändern können, wird eine Abfrage durch einen WebIO-Eingang durchgeführt. Dieser wird einfach mit einem Merker verknüpft, auf dem die Werte übertragen werden. Dadurch können die Daten in der Programmierung weiterverwendet werden.

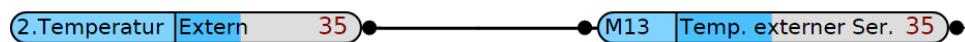


ABBILDUNG 4.5: WEB-EINGANG

In Abb. 4.5 wird die Temperatur von einem externen Bussystem alle zehn Sekunden abgefragt und auf den Merker M13 gespeichert. Dieser wird nachfolgend für das Monitoring weiterverwendet.

4.1.4 VISUALISIERUNG DES COMEXIO-SYSTEMS

Um die von der Gebäudeautomation erkannten Messgrößen im Nachhinein mit den vom Bus erfassten Daten durch IP-Symcon vergleichen zu können, werden diese Größen durch Segmentanzeigen auf dem Dashboard wiedergegeben. Da es sich bei den Segmentanzeigen nur um aktuelle Werte handelt, werden dazu noch Monitoring-Graphen erstellt, welche die Temperatur- und Verbrauchsdaten zeitlich erfassen. Ebenfalls werden visuelle Module verwendet, um auf den Bussystemen Aktionen auszuführen oder Zustände abzurufen.

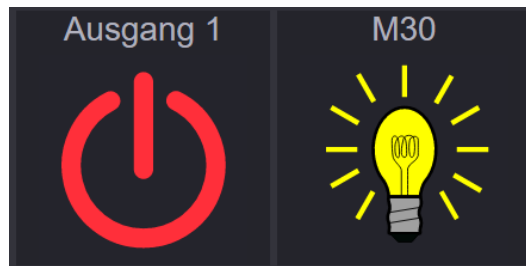


ABBILDUNG 4.6: DASHBOARDMODULE

Der Dashboard-Taster (links) aus Abb. 4.6 hat die Funktion eines Visualisierungseingangs. Durch Drücken bzw. Klicken dieses Elements wird ein Impuls auf den zugehörigen Eingang (hier: VISU7 vgl. Abs.: 4.1.3) gesendet. Durch das rechte Element wird der Zustand von Merker 30 des eigenen Systems abgerufen. Da dieser Merker allerdings mit einem Web-Befehl verbunden ist, (vgl. ebenfalls Abs.: 4.1.3) wird über dieses Symbol der Zustand des ersten Ausgangs vom externen System angezeigt. Durch Analog-Module lassen sich sogar ganze Zahlenwerte darstellen. So kann z.B. durch eine Siebensegmentanzeige ein Analogwert, wie u.a. die momentane Temperatur, auf der Weboberfläche wiedergegeben werden.

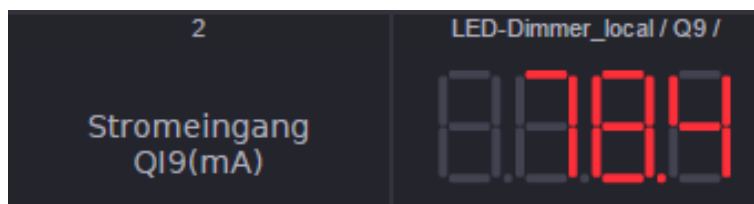


ABBILDUNG 4.7: DARSTELLUNG DURCH SIEBENSEGMENTANZEIGE

Damit zeitabhängige Werte nachvollzogen werden können, werden Monitoring-Graphen verwendet. Im Dashboard kann dann der Verlauf des Graphen in beliebigen Zeitabständen betrachtet werden. Dadurch kann der Bereich von Minuten bis hin zu Monaten definiert werden.

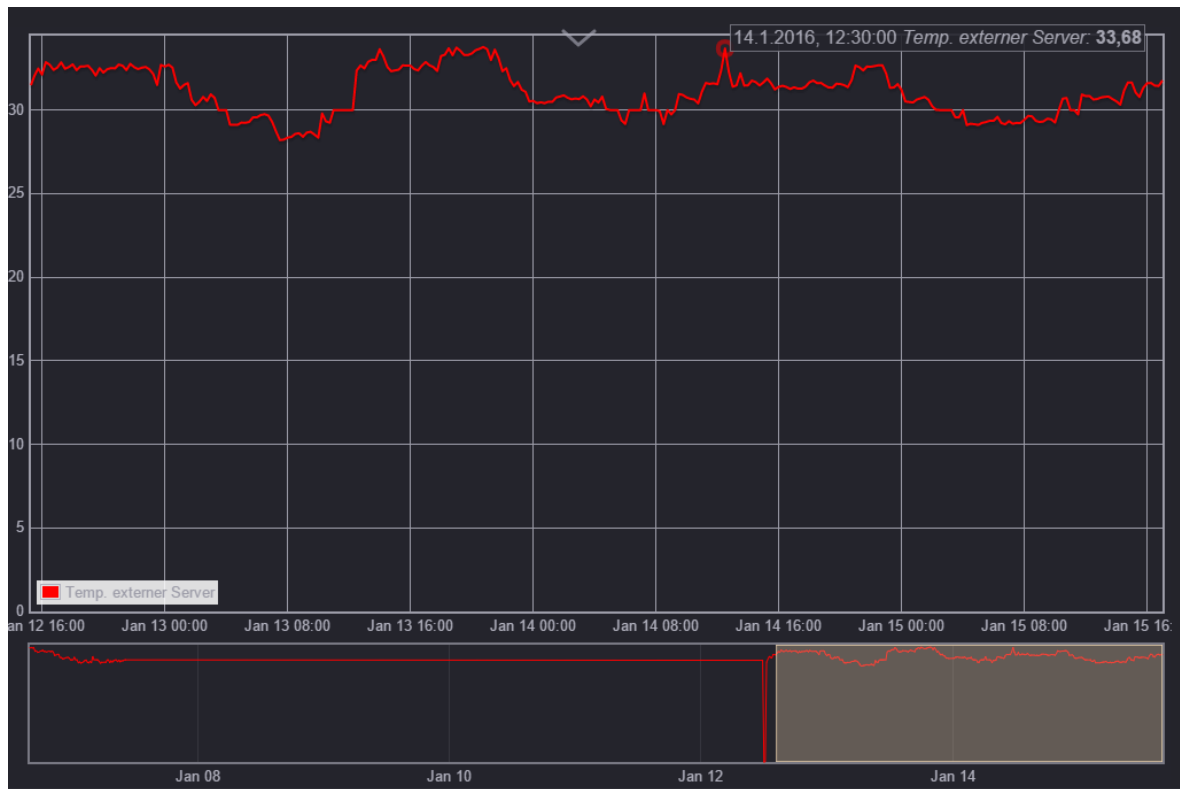


ABBILDUNG 4.8: DASHBOARD-MONITORING-TEMPERATURVERLAUF

4.2 KONFIGURIERUNG VON IP-SYMCON

Beim Starten der IP-Symcon Verwaltungskonsolle befindet man sich im Hauptbereich. Von dort aus lassen sich alle Unterkategorien und Ansichten erreichen bzw. abrufen. Der Umfang der Gebäudeleitsoftware kann durch die Vielzahl an Möglichkeiten, Funktionen und Schnittstellen schnell sehr komplex werden. Um eine bessere Übersicht und Dokumentation des Projekts zu ermöglichen, wird der Objektbaum verwendet. Dabei stehen insgesamt drei Varianten zur Verfügung, unter „Physikalischer Baumansicht“ werden alle verwendeten Instanzen aufgelistet. In der Listenansicht werden alle verwendeten Objekte (Instanzen, Scripte, Variablen, etc.) untereinander aufgezählt. Die „logische Baumansicht“ liefert gegenüber den anderen beiden Varianten eine Übersicht über das gesamte Projekt. Dort lässt sich die Struktur der GLT durch das Anlegen von Kategorien darstellen. In jeder Kategorie lassen sich beliebig viele weiter hinzufügen, sowie beliebige andere Objekte. So kann z.B. ein Gebäude in Abteilungen, Etagen, Räume oder Funktionen unterteilt werden.

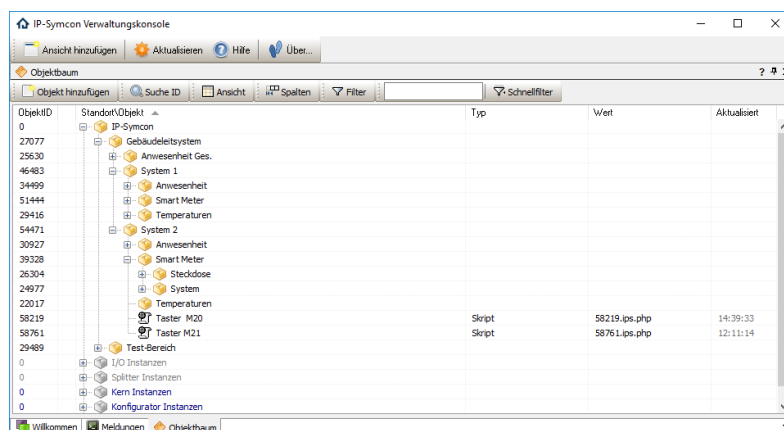


ABBILDUNG 4.9:IP-SYMCON OBJEKTBAUM

COMEXIO wird als Gebäudesystem von IP-Symcon nicht unterstützt. Die direkte Anbindung an die Gebäudeleitsoftware ist nur durch eigens geschriebene Scripte oder manuell konfigurierbarer Instanzen möglich. Zusätzlich müssen Variablen und dessen Profile angelegt werden, damit diese von den Scripten verwendet werden können, um Werte speichern oder verarbeiten zu können. Die Profile dienen der visuellen Darstellung, sie fügen den Werten Kontextdaten bei (z.B. € oder °C). Im „Profil-Manager“ werden neben der Bezeichnung des Variablenprofils auch der Datentyp und dessen Wertebereich angegeben. Für jeden Wert kann eine Assoziation erstellt werden. Diese legt bei der Visualisierung fest, welcher Text, welche Farbe, sowie

welches Symbol beim Erreichen eines Wertes angezeigt werden soll. Im „Variablen-Editor“ werden bei Bedarf Aktionen hinzugefügt. Dadurch kann realisiert werden, dass durch eine Änderung des Variablenwertes eine Instanz bzw. ein Script ausgeführt wird. Diese Variable steht dem Script somit automatisch zur Verfügung und kann durch sog. Systemfunktionen abgefragt werden. Mit Hilfe der Archiveinstellungen werden alle Wertänderungen aufgezeichnet und lassen sich Graphisch abrufen und auswerten.

4.2.1 DATEN SENDEN (IPS -> COMEXIO)

Die einzige Möglichkeit auf die API der Bussysteme zuzugreifen besteht darin, diese über http-Request anzusprechen. Damit IPS eine solche Anfrage sendet, muss ein Script in PHP programmiert werden. Dabei können diese Befehle auf unterschiedliche Weisen getätigt werden.

IPS bietet bei der Programmierung von Scripten eine Menge an Befehlen und Funktion an, welche das Schreiben von Programmen wesentlich vereinfacht. Die Funktionen gehören nicht zur PHP-Bibliothek, können aber trotzdem in das PHP-Script ohne weitere Angaben, eingebaut werden. Um nun eine Anfrage an die Gebäudeautomation senden zu können, muss wie unter Abschnitt 2.6 erwähnt, nur die Ressource über die zugehörige URL abgerufen werden. Damit dieser Vorgang (das Aufrufen der URL) im Hintergrund stattfindet, wird die Systemfunktion:

`„string Sys_GetURLContent (string $URL)“`

benutzt. Diese Funktion ruft eine URL am Server auf und gibt den Inhalt als Rückgabewert zurück. Durch das Aufrufen lassen sich somit http-Requests an die Servereinheit senden. Mit dieser Variante ist das Script mit wenigen Zeilen programmiert. Wobei immer auf die gewünschte Funktion geachtet werden muss. Auch hier macht es, wie auch bei der Kommunikation von COMEXIO zu COMEXIO sinn, die Werte auf Merkern zu speichern. Um einen kurzen Impuls an das System zu senden, genügt es den Datenpunkt (Ein- bzw. Ausgang oder Merker) durch die obengenannte Funktion auf eine logische „1“ zu setzen und im selben Ablauf kurz darauf wieder auf eine Null. Ein Script welches einen Tastendruck simuliert, wird wie folgt aufgebaut:

```

<?
$url1 = "http://192.168.178.51/api/?action=set&marker=M20&value=1";
$url0 = "http://192.168.178.51/api/?action=set&marker=M20&value=0";
Sys_GetURLContent ($url1);
//sleep (0.1);
Sys_GetURLContent ($url0);
?>

```

QUELLTEXT 1: IP-SYMCON TASTER

Der „sleep“-Befehl ist auskommentiert, da dieser zum Abschicken eines Impulses nicht benötigt wird. Das Script wird von oben nach unten abgearbeitet. Es wird somit zuerst „\$url1“ ausgeführt und der Merker „M20“ auf 1 gesetzt. Erst danach wird die zweite Systemfunktion aufgerufen und „\$url2“ setzt den Merker nach kurzer Zeit wieder auf 0. Der „sleep“-Befehl kann verwendet werden um den Impuls länger zu erhalten, z.B. beim Steuern von Jalousien. Damit nicht für jede Aktion separate Programme geschrieben werden müssen, lassen sich „if/else“- oder „switch/case“- Anweisungen verwenden. Dabei wird eine Variable so editiert, dass diese über mehrere Assoziationen verfügt. Dadurch wird pro angelegter Assoziation ein Feld auf dem WebFront hinterlegt. Als Datentyp wird hierbei Integer verwendet, damit mehr als nur zwei Eigenschaften der Variable angelegt werden können. Jede Assoziation erhält dabei einen Wert. Zudem können noch Namen, Anzeige-Icon und eine Farbe ausgewählt werden. Es entsteht ein WebFront Element wie es in Abb. 15 dargestellt wird.

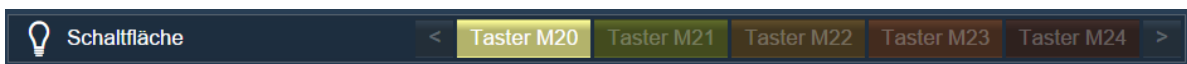


ABBILDUNG 4.10: WEBFRONT AUSZUG

Jeder dieser Flächen ist eine Assoziation dieser Variable und besitzt einen spezifischen Wert. Wird in diesem Fall z.B. das erste Feld betätigt, erhält die Variable den Wert „0“. Durch eine „if“- oder „switch/case“-Anweisung, kann durch die mit dem Script verknüpfte Variable eine gewünschte Funktion ausgeführt werden. Wenn für eine Aktion schon ein Script vorhanden ist, kann durch den Befehl

```
„Boolean IPS_RunScript (integer $ScriptID)“
```

dieses Script ausgeführt werden. Um die Anzahl an Scripts im Objektbaum gering zu halten, können die gewünschten Funktionen auch direkt unter der Bedingung programmiert werden.

```

<?
SetValue($_IPS['VARIABLE'], $_IPS['VALUE']);
$var = GetValue (53611);

switch($var)
{
    case "0":
        IPS_RunScript(58219);
        break;

    case "1":
        $url1 = "http://192.168.178.51/api/?action=set&marker=M21&value=1";
        $url0 = "http://192.168.178.51/api/?action=set&marker=M21&value=0";
        Sys_GetURLContent ($url1);
        Sys_GetURLContent ($url0);
        break;
}
?>

```

QUELLTEXT 2: SWITCH / CASE-ANWEISUNG

In diesem Programmauszug wird unter case "0" die Funktion durch das Aufrufen eines weiteren Scriptes ausgeführt. Bei case "1" wird ein Impuls direkt ohne weitere Instanzen als URL-Content gesendet. Welcher Fall ausgeführt werden soll, wird durch die Variable \$var bestimmt, welche wiederum von der Systemvariable durch die WebFront beeinflusst wird.

Des Weiteren können mit dem if-Befehl Schaltvorgänge leicht realisiert werden. Dabei wird durch eine Boole-Variable zwischen Ein- und Ausschalten unterschieden. Der Zustand des zu schaltenden Elements wird zuvor abgefragt um zu verhindern, dass derselbe Zustand ein weiteres Mal gesendet wird. Somit folgt auf eine *true* (ein) ein *false* (aus) und umgekehrt.

```

<?
$geturl= "http://192.168.178.51/api/?action=get&marker=M20";
$wert = Sys_GetURLContent ($geturl);
SetValue (34699, $wert);
$zustand = GetValue (34699);

IF ($zustand == false)
{
    Sys_GetURLContent ($url1) ;
}
else //oder IF ($zustand == true)
{
    Sys_GetURLContent ($url0);
}
?>

```

QUELLTEXT 3: IF-ANWEISUNG

Durch die if-Anweisungen wird der Zustand überprüft und entschieden, welcher Befehl ausgeführt werden soll. Ist der Zustand nicht *false* werden die else-Funktionen abgerufen. else kann hier auch durch eine weitere if-Anweisung ersetzt werden, die prüft, ob der Zustand *true* ist. In jedem Fall wird der entgegengesetzte Zustand übergeben.

Neben den digitalen Werten lassen sich durch IP-Symcon ebenfalls Analogwerte an die Gebäudeautomation übermitteln, um z.B. den Sollwert für Temperaturen zu übermitteln. Die Werte lassen sich dabei über das WebFront einstellen bzw. festlegen. Um dies zu ermöglichen müssen die Variablen, wie schon erwähnt, editiert werden. Als Datentyp wird hierbei ein Integer (oder Float, wenn Gleitkommazahl gewünscht) verwendet, um ganze Zahlen senden zu können. Es stehen verschiedene Möglichkeiten der visuellen Darstellung zur Verfügung (siehe Abs.: 4.2.4). Sobald diese Variable geändert wird, soll das Script, welches den Analogwert übermittelt ausgeführt werden. Dafür muss unter dem Punkt „Aktion“ dieses ausgewählt werden, beide Instanzen sind somit miteinander verknüpft. Durch die Verwendung von Systemvariablen („\$_IPS [‘ ’]“) lassen sich die verknüpften Variablen direkt in der PHP-Programmierung ohne weitere Befehle abrufen.

```
<?
SetValue($_IPS['VARIABLE'], $_IPS['VALUE']);
$soll = GetValue (13150 /*[Test-Bereich\Sollwert]*/);
$url =
"http://192.168.178.51/api/?action=set&marker=M40&value=$soll";

Sys_GetURLContent ($url);
?>
```

QUELLTEXT 4: ANALOGWERT SENDEN

Durch den Variablenzugriff („SetValue“) wird der neue Wert („\$_IPS[‘VALUE’]“) auf die verknüpfte Variable ((\$_IPS[‘VARIABLE’])) geschrieben. Diese wird im Anschluss über die „ObjektID“ abgefragt und unter „\$soll“ zwischengespeichert. Der neue Sollwert wird, wie auch die digitalen Impulse, durch einen http-Request an die GA gesendet, indem dieser an die URL angehängt wird. Bei der Ausführung des Request, wird „\$soll“ durch die darunter gespeicherte Zahl ersetzt.

4.2.2 DATENABRUFEN (IPS <- COMEXIO)

Um Informationen bzw. Daten vom Bussystem abrufen zu können, stehen mehrere Varianten zur Verfügung. Da die Kommunikation der Bussysteme über http-Anfragen durchzuführen ist, muss eine solche Anfrage an die Servereinheit gesendet werden. Der Server liefert daraufhin die gewünschte Antwort zurück. Somit muss dem System lediglich eine Anfrage gesendet und IPS die Antwort verständlich gemacht werden. Wird z.B. folgende URL,

„http://192.168.178.50/api?action=get&ext=IO-Extension_local&io=TI1“,

in die Adresszeile des Browser eingegeben, so erhält man auf diese Anfrage die Antwort des Servers in Form eines Strings (z.B. 22). Dieser gibt in diesem Zusammenhang an, welche Temperatur durch das System an „TI1“ erkannt wird. Der vom COMEXIO-Server angezeigt Wert entspricht in diesem Fall genau dem Quelltext der aufgerufenen Website. Der Quelltext einer Internetseite kann im Browser angezeigt werden. Dieser Vorgang ist wichtig, wenn erkannt werden, muss wie die Antwort des Servers aussieht und an welcher Stelle sich die gewünschten Informationen befinden. Dieser Schritt ist z.B. notwendig, wenn durch IPS Wetter-Daten von Webseiten, wie „www.wetter.com“ abgerufen werden sollen. Auf solchen Seiten ist der Quelltext selbstverständlich wesentlich komplexer.

Um über IP-Symcon eine Webseite und dessen Quelltext aufzurufen, wird der „WWW Reader“ verwendet. Dabei handelt es sich um eine Instanz (Schnittstellen-Modul), welche in der Lage ist, über http-GET-Anfragen Webseiten auszulesen. Innerhalb dieser Instanz werden die Authentifizierung, IP-Adresse sowie das Wiederholungs-Intervall der Abfrage angegeben. Die Antwort (der Quelltext der Webseite) wird dann durch eine untergeordnete Instanz abgerufen und für IPS verständlich gemacht bzw. abgespeichert. Für diesen Schritt wird die sog. „TextParser“-Instanz verwendet. Durch diesen können Quelltexte auf die gewünschten Informationen reduziert werden. Dabei schneidet diese Instanz durch angelegte Regeln Quelltext weg und speichert den benötigten Teil unter einer Variablen ab. Damit er weiß, welchen Teil er abschneiden (vor oder hinter), holen (bis) oder ausschneiden (zwischen) soll, werden bei jeder dieser Regeln sog. „Tag´s“ angegeben. Sobald eine bestimmte Zeichenfolge (Tag) auftritt, wird die ausgewählte Funktion durchgeführt. Damit die gewünschte Information auch herausgeschnitten bzw. geholt wird, muss auf jeden Fall ein Tag angegeben werden, damit Anfang und Ende des Wertes definiert sind. Innerhalb des Text Parser

wird die übergeordnete Instanz (hier: der „WWW Reader“) angegeben. Jedes Mal, wenn der Reader die Abfrage der Webseite durchführt, wird der erhaltene Quelltext automatisch an den Text Parser übermittelt. Dieser schneidet dann den definierten Teil heraus und stellt ihn dem System über eine Variable zur Verfügung. Ein großer Nachteil dieser Methode ist in diesem Zusammenhang, dass der Parser unbedingt eine Angabe von Tag's benötigt. Da die Gebäudeautomation von COMEXIO die Werte direkt als Quelltext angibt, ist es ohne weitere Operationen nicht möglich, mit dem Parser zu arbeiten. Erkennt das System z.B. eine Temperatur von 22°C, wird diese durch die Abfrage mit der oben genannten URL als „22“ (als Quelltext) angegeben. Der Text Parser benötigt nun eine Regel, von bzw. bis wo er etwas ausschneiden oder holen soll. In diesem Fall ist es nicht möglich eine Regel aufzustellen, ohne einen falschen Wert zu holen. Verwendet man die Regel „Hole Text bis zum Tag 2“, so würde der erhaltene Wert 0 sein. Auch das Ausschneiden des Wertes ist nicht durchführbar, da mit dieser Regel auch mindestens ein Wert wegfällt (in diesem Fall wäre der Wert ebenfalls 0). Nur durch eine Manipulation des Wertes innerhalb des COMEXIO-Systems kann helfen den Wert für die Verwendung durch „WWW READER“ und „TextParser“ IPS zugänglich zu machen. Durch das Anlegen einer Rechenoperation, kann der Wert mit 1,1 multipliziert werden. Dadurch kann im Parser eine Regel angelegt werden, welche den Wert bis zum ersten Komma ausschneidet. Diese Methode ist allerdings auch nur dann möglich, wenn es sich bei dem abzufragenden Wert nicht schon um eine Kommazahl handelt.

Eine wesentlich einfachere und unkompliziertere Variante bietet die Verwendung eines Scriptes. In diesem Fall ist es von Vorteil, dass der gesamte Quelltext ohne Ausschneiden oder dergleichen verwendet werden kann. Um die Abfrage am Server durchzuführen wird wie beim Senden (vgl. Abs.: 4.2.1) von Daten der „Sys_GetURLContent“-Befehl verwendet. Denn dieser Befehl dient im eigentlichen Sinne dem Auslesen von Webseiten. Er ruft im Hintergrund eine URL (z.B. eine Webseite) auf und holt den Inhalt, also den Quelltext dieser Seite. Der folgende Programmauszug zeigt wie ein Script zum regelmäßigen Abrufen programmiert werden kann.

```

<?
//Variablen + ID anlegen:
$Variable_Temp1 = 19507;
$Variable_Temp2 = 44560;
//URL unter Variable "$URL" speichern:
$URL1 = "http://192.168.178.50/api/?action=get&ext=IO-
Extension_local&io=TI1";
$URL2 = "http://192.168.178.50/api/?action=get&ext=IO-
Extension_local&io=TI2";
//Intervall Zeit in Sekunde
$Intervall = 30;
//Inhalt von URL:
$cont1 = Sys_GetURLContent($URL1);
$cont2 = Sys_GetURLContent($URL2);
//Inhalt in Variable setzen:
SetValue($Variable_Temp1, $cont1);
SetValue($Variable_Temp2, $cont2);
//Wiederholungsintervall:
if ($_IPS['SENDER'] == "Execute")
{
    IPS_SetScriptTimer($_IPS['SELF'], $Intervall);
}
?>

```

QUELLTEXT 5: TEMPERATURABFRAGE

Mit diesem Script werden zwei Temperaturen von dem Bussystem abgefragt und auf zwei verschiedenen Variablen gespeichert. Die erhaltenen Quelltexte werden hier auf die Variablen \$cont“...” geschrieben, welche im Anschluss auf globale Variablen gespeichert werden, damit diese in dem WebFront angezeigt werden können. Beim Variableneditieren muss in diesem Zusammenhang darauf geachtet werden, dass die Archiveinstellungen aktiviert sind, damit der Verlauf der Werte aufgezeichnet wird und z.B. über Graphen visuell abgerufen werden kann.

Um ein Intervall zu erstellen, welches das Script nach einer vorgegebenen Zeit wieder ausführen lässt, wird der Befehl,

boolean IPS_SetScriptTimer (integer \$ScriptID, integer \$TimerWert)

verwendet. Durch die Angabe der ID des Programms wird festgelegt, für welches Script ein Timer erstellt werden soll. Der „TimerWert“ gibt dabei das Wiederholungsintervall in Sekunden an. Die if-Anweisung dient dabei nur dem Anlegen bzw. des Starten des Timers. Ist das Script einmal aktiv wird diese Anweisung nicht mehr benötigt da sich das Script nach jedem Intervall erneut ausführt. Mit Hilfe der if-Anweisung und der Systemvariablen „\$_IPS[‘SENDER’]“ wird nur sichergestellt, dass der Timer automatisch durch die Konsole ausgeführt wird. „Execute“ ist somit ein Systemelement, welches von der Konsole aktiviert wird. Dadurch wird das Script sofort

ausgeführt, wenn der IPS-Server hochgefahren ist. Eine manuelle Aktivierung ist in diesem Fall nicht notwendig.

4.2.3 DATEN EMPFANGEN (COEMXIO -> IPS)

Damit der Bus der Gebäudeautomation nicht unnötig durch das Abfragen von IPS belastet wird, ist es möglich Befehle vom IO-Server direkt an die Gebäudeleitsoftware zu senden. Ein Zustand kann direkt bei der Änderung übertragen werden ohne, dass ein permanenter Abruf stattfinden muss. IP-Symcon bietet zwei Schnittstellen an, über die die Daten an die Software gesendet werden können. In dieser Arbeit wird ausschließlich die JSON RPC API verwendet. Diese bietet eine wesentlich höhere Geschwindigkeit gegenüber der älteren SOAP API an. Zudem besteht eine Sicherheit durch die Verwendung einer Authentifizierung.

Dieses Datenformat dient dem Austausch von Informationen zu entfernten Systemen, in einfacher Textform. Es können dafür verschiedene Kommunikationsprotokolle verwendet werden um das JSON-Objekt zu übertragen. In diesem Fall wird der Aufruftyp von COMEXIO vorgegeben, da dieser nur Requests senden kann. Die Parameter werden als JSON-String übergeben und werden somit im „Message Body“ der Abfrage angegeben. Um einen sog. Message Body zu übertragen, muss POST als Aufrufmethode verwendet werden. Die Bestandteile der Anfrage können aus der folgenden Tabelle verwendet werden:

Bestandteile	Beschreibung	Bemerkung
jsonrpc	Ein String mit dem Namen der JSON-RPC-Version ("2.0")	Ab Version 2.0
result	Das Rückgabeobjekt der Funktion, falls kein RPC-Fehler auftrat. Andernfalls: • Version 2.0: wird dieses Feld weggelassen. • Version 1.0: hat dieses Feld den Wert <code>null</code>.	
error	Das Fehlerobjekt, falls ein RPC-Fehler auftrat. Andernfalls: • Version 2.0: wird dieses Feld weggelassen. • Version 1.0: hat dieses Feld den Wert <code>null</code>.	
id	Enthält den gleichen Wert wie <code>id</code> des Requests, der diese Response verursacht hat.	

ABBILDUNG 4.11: TABELL:JSON BESTANDTEILE

Um auf die JSON API von IP-Symcon zugreifen zu können, muss der Fernzugriff in der Tray-Anwendung aktiviert werden. Das dort vergebene Passwort und die angezeigte Lizenz, dienen später für die Authentifizierung. Im Anschluss wird in der WebIO des COMEXIO Servers ein neues Gerät eingerichtet. Die IP-Adresse entspricht der des Computers bzw. Servers, auf dem IPS läuft. Benutzernamen und Passwort

sind, wie schon erwähnt, die Lizenz unter der IPS aktiviert ist und das selbst angelegte Fernzugriffspasswort.

Um auf die API von IP-Symcon zuzugreifen, muss der entsprechende Pfad angegeben werden, hinter dem sich die Schnittstelle verbirgt. Die Software ist standardmäßig über den Port 82 erreichbar, der Aufbau der URL sieht wie folgt aus:

„http://IP-Adresse:Port/api/“

Soll ein Zugriff von einem anderen Gerät erfolgen, muss der Anfrage noch die Authentifizierung hinzugefügt werden. Die URL würde in diesem Fall folgenden Aufbau haben:

„http://Lizenz:Passwort@IP-Adresse:Port/api/“

Somit wird beim Anlegen des Webbefehls bei Parameter der Pfad „/api/“ und unter TCP-Port die „82“ angegeben. Da die Aufrufmethode „POST“ ausgewählt ist, werden die Befehle aus den angegebenen HTML-Daten ausgeführt. Im Header (Header Modifizierung) wird zudem die Klassifizierung der übertragenen Daten durch den Content-Type angegeben. Dem Client wird dadurch mitgeteilt um welche Art von Inhalt es sich bei der Anfrage handelt. Durch die Auszeichnungssprache XML (maschinenlesbare Sprache) werden die Befehle in der Form von Textdateien übertragen. Der Inhalt der Anfrage wird somit durch den Typ: „application“ klassifiziert. Als Header Modifizierung muss somit „Content-Type:application/xml;“ angegeben werden.

Den Webbefehl Test1 bearbeiten

Name	Test1
Variablen-Typ	Digital
	☒ Nur bei einer logischen 1 senden
Minimaler Wert	0
Maximaler Wert	1
Parameter	/api/
Header Modifizierung	Content-Type:applicatio/xml;
Html-Daten	<pre>{ "jsonrpc": "2.0", "method": "IPS_RunScript", "params": [28073], "id": 1 }</pre>
Typ	Ausgang
TCP-Port	82
Aufrufmethode	POST
Authentifizierung	Ja
Wiederholfrequenz	
Antwortinterpretier	

Testen Speichern

ABBILDUNG 4.12:COMEXIO PROGRAMMAUSZUG

Unter dem Datenpunkt „methode“ werden die Systemfunktionen von IPS angegeben, welche durch „params“ definiert werden. Sie geben somit ID und Werte der Systemfunktion an.

Damit über einen Webbefehl auch Analogwerte geschickt werden können, wird wie eine Funktion benötigt. Diese wird bei der POST-Methode in den Body (unter HTML-Daten) des Request geschrieben. Das Script soll dabei den übergebenen Wert an den richtigen Datenpunkt der JSON Schnittstelle weiterleiten. Die Punkte und dessen Werte werden in diesem Fall als String weitergeben, müssen allerdings durch eine Funktion zusammengesetzt werden. Damit das System erkennt, dass eine Funktion verwendet wird, muss der Body mit „function data (Variable)“ beginnen. Der Wert innerhalb der Klammern ist der zu übergebene Analogwert. Dieser wird durch „..Variable..“ an die gewünschte Position gesetzt.

function data (v)

```
return '{"jsonrpc":"2.0","method":"SetValue","params":[53197,'..v..'],'id': 1}'  
end
```

Durch den „return“-Befehl werden alle benötigten Daten an die IPS API übermittelt. Wichtig ist, dass bei dem String innerhalb der geschweiften Klammern keine Absätze oder Leerzeichen gesetzt werden.

4.2.4 VARIABLEN VISUALISIERUNG

Wie schon erwähnt, lassen sich Variablen durch das Anlegen von Profilen für die WebFront bzw. für die visuelle Darstellung vorbereiten. So können eine Variable und dessen Werte u.a. Präfix sowie Suffix hinzugefügt werden. Das Präfix erscheint bei der Ansicht vor dem Wert und kann z.B. lauten: „Es ist:“. Mit dem Suffix kann der Variable z.B. eine Einheit wie: „°C“ angehängen werden (vgl. Abb.: 4.13).



ABBILDUNG 4.13:WEBFRONT AUSZUG-TEMPERATUR

Zwischen Prä- und Suffix, wird der Text der Variablen angezeigt. Dieser wird durch die Assoziationen definiert und kann sich dabei abhängig vom Wert ändern. Wird ein Text manuell definiert, so wird dieser beim Erreichen des zugehörigen Wertes angezeigt. Soll allerdings der Variablenwert angegeben werden, so muss unter Text „%.1f“ hinterlegt sein (vgl. Abb.: 4.14). Durch diesen Befehl wird zugleich auch die Anzahl an Nachkommastellen festgelegt.

Wert	Text	Icon	Farbe
-30	%.1f	Snowflake	Blue
10	%.1f	Snowflake	Cyan
17	%.1f	Temperature	Light Blue
20	%.1f	Temperature	Green
26	%.1f	Temperature	Orange
33	%.1f	Sun	Red

ABBILDUNG 4.14:VARIABLENEDITOR/PROFIL-MANAGER

Wie in dieser Abbildung zu sehen, können für jedes Assoziation eigene Icons und Farben verwendet werden. Erreicht die Variable einen bestimmten Wert, so wird in

dem WebFront das Icon und die Farbe des Darstellungsfeldes (wie oben zu sehen) geändert, bis wieder ein nächster Schwellenwert erreicht wird.

Wird bei Variablen zusätzlich die Archiveinstellung aktiviert, so steht einem die Diagrammanzeige in der Visualisierung unter dem Grafiksymbol neben des Anzeigefeldes zur Verfügung. Es wird dabei zwischen zwei Aggregationen unterschieden. Zum einen den Werteverlauf („Standard“) und zum anderen dem Aufzählen („Zählen“) der erfassten Daten.

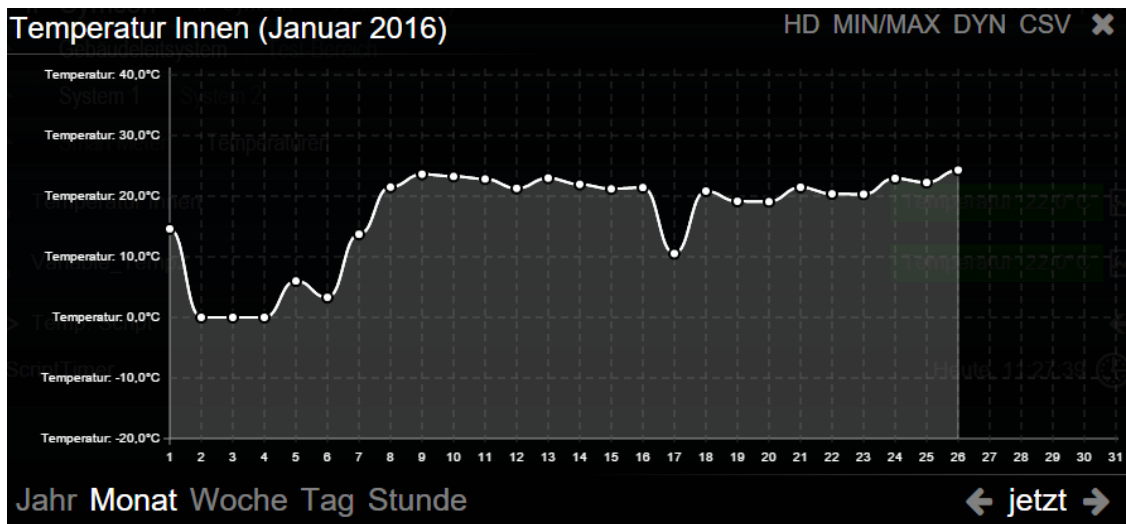


ABBILDUNG 4.15: WEBFRONT/TEMPERATURVERLAUF

Wird eine Variable als Eingabeelement (z.B. als Taster) verwendet, werden in dem WebFront je nach Profil entsprechend viele Flächen erstellt. Ohne Angaben von Assoziationen wird die Anzahl der Flächen durch das Definieren von Min-, Maxwert und der Schrittweite bestimmt. Die Schrittweite gibt an in welchen Abständen die Werte dargestellt werden sollen (z.B. 1 = 1,2,3; 2 = 2,4,6). Eine Anzeige der Flächen ist nur dann möglich, wenn der Wert der Weite größer als 0 ist. Ist dies nicht der Fall, werden die Variablenwerte wie bei dem Temperaturbeispiel nur angezeigt und können somit nicht betätigt werden. Werden zu diesen Einstellungen noch Assoziationen angelegt, kann die Darstellung in dem WebFront geändert werden (vgl. Abb.: 4.16 & 4.17).

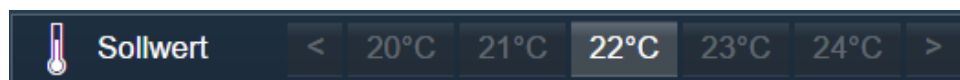


ABBILDUNG 4.16: WEBFRONT / VARIABLE MIT SCHRITTWEITE, OHNE ASSOZIATION

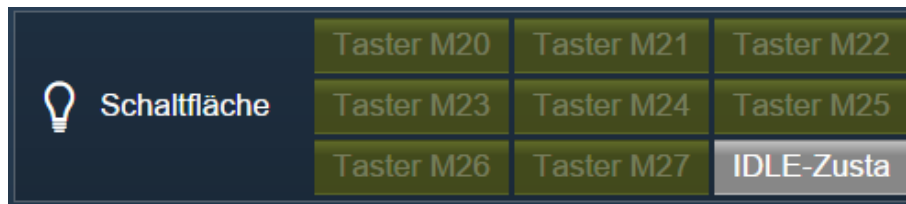


ABBILDUNG 4.17: WEBFRONT / VARIABLE MIT SCHRITTWEITE, MIT ASSOZIATIONEN

5 FUNKTIONSUMFANG DES PROJEKTES

5.1 SMARTMETERING

Wie schon erwähnt wurde, sollen die „intelligenten Zähler“ den aktuellen Energieverbrauch in digitaler Form erfassen. Die durch die Gebäudeautomation erfassten Daten werden im Anschluss von IP-Symcon zusammengetragen und lassen sich dort grafisch wiedergeben.

Damit das System die genaue Wirkleistung berechnen kann, muss die Phasenverschiebung zwischen Strom und Spannung ermittelt werden. Damit COMEXIO in der Lage ist den Winkel dieser Verschiebung angeben zu können, wird entweder der zum Bus passende „COMEXIO Smart-Meter“ oder eine „COMEXIO IO-Extension“ verwendet. Im Rahmen dieses Projektes steht eine Extension zur Verfügung um die benötigten Information zu erhalten. Da die IO-Extension im eigentlichen als Erweiterung der Ein- und Ausgänge verwendet werden soll, muss dieses Modul zuvor umkonfiguriert werden. Das System kann nur dann eine sinusförmige Phasenverschiebung erkennen, wenn die Phasen (L1, L2 und L3) auf die Eingänge I1, I2 und I3 gelegt werden. Zudem müssen diese drei Eingänge im System auf Analog gestellt werden. Jeweils drei Ausgänge teilen sich einen Leiter, ist z.B. auf I1 der Leiter L1 gelegt so müssen die Ausgänge Q1-Q3 ebenfalls mit L1 versorgt werden. In diesem Zusammenhang wird dann eine Phasenverschiebung unter dem zugehörigen Phaseneingang erkannt.

Da die Variablen dem COMEXIO-System nur temporär zur Verfügung stehen, sollten die Werte über IP-Symcon verwaltet werden. Die Daten lassen sich entweder vom Bus abrufen oder können direkt von dort an IPS gesendet werden. Die Berechnung der Werte kann dabei sowohl durch die Gebäudeautomation, als auch durch die Gebäudeleittechnik übernommen werden.

Durch Verwendung der Funktionsbausteine wird die Berechnung der gewünschten Werte mit Hilfe der GA vorgenommen. Mit dem Logikbaustein „Leistung II“ lassen sich die Wirk-, Blind- und Scheinleistung automatisch berechnen. Damit das System die Blind- sowie die Scheinleistung berechnen kann muss der Phasenwinkel in der Programmierung benutzt werden. Um die Werte im Anschluss von außerhalb abrufbar zu können, werden sie auf Merkern zwischen gespeichert. Über Zeitglieder und Zählerbausteine, wird die Zeit bestimmt, in der Energie verbraucht wurde. Das Zeitglied gibt den Takt für den Zähler an. Dieser zählt für jeden eingehenden Impuls einen Schritt hoch. Gibt das Zeitglied somit im Sekundentakt ein Impuls weiter so kann die Dauer des Verbrauchs festgehalten werden. Dieser Wert wird in Stunden umgerechnet und mit der Wirkleistung multipliziert. Dadurch kann der Energieverbrauch in kWh berechnet werden. Von IPS braucht eigentlich nur dieser Wert abgerufen werden. Durch ein Script kann dieser Wert zusätzlich mit dem Strompreis multipliziert werden um die verursachten Kosten zu ermitteln.

Diese Berechnung des Energieverbrauchs kann auch durch IPS übernommen werden. Dabei müssen lediglich zwei Werte vom Bus-System abgerufen werden, der Strom und der Phasenwinkel. Diese werden wie üblich durch eine URL Abfrage an das IPS Script übergeben. Die Antwort des Servers auf die URL Anfrage wird von IP-Symcon als String erkannt. Damit diese Zeichenkette in weiteren Berechnungen verwendet werden kann, wird mit Hilfen von

`„float floatval (mixed $var)“`

aus dem String eine float-Variable gemacht und wird somit als Dezimalzahl gespeichert. Für die Berechnung der Blind-, Schein- und somit der Wirkleistung, mit Hilfe der Phasenverschiebung, werden die Kosinus- und Sinusfunktionen in PHP verwendet. In dieser Sprache wird mit dem Bogenmaß eines Winkels gerechnet. Um diese Umrechnung durchzuführen wird,

`„float deg2rad ( float $number)“`

verwendet. Diese Funktion rechnet somit den Wert „\$number“ von Grad in Bogenmaß um und speichert diesen unter einer float-Variablen ab (vgl. Anhang: 7.1.2.). Um die exakte Zeit zu erhalten, in der die Verbrauchte Energie ermittelt werden soll, wird im folgenden Script der Unix Timestamp verwendet.

```

<?
if (GetValue(44071) == true)
{
$Since = time() - IPS_GetVariable(44071) ['VariableChanged'];
SetValue (28401,$Since);
}
else if (GetValue(44071) == false)
{
SetValue (40362,GetValue(28401)+GetValue(40362));
SetValue(28401,0);
}
?>

```

Quelltext 6: UNIX-ZEITSTEMPEL

Bei diesem Zeitstempel handelt es sich so gesehen um einen Counter, welcher im Sekundentakt aufwärts zählt. Die sog. Unixzeit der Version 6 ist eine einfache Zeitdefinition, die seit dem 1. Januar 1970 aktiv ist. Mit „time()“ wird dieser Stempel im Programm abgerufen. Damit die verstrichene Zeit genau bestimmt werden kann, muss das System erkennen ab wann die Zeit berechnet werden muss. Dafür bietet IPS die Systemfunktion „IPS_GetVariable“ an. Durch diese Funktion lassen sich sämtliche Informationen von angelegten Variablen erhalten. Wie in Quelltext 6 wird mit „['VariableChanged']“ der Zeitstempel bei Variablenänderung übergeben. Die Differenz zwischen aktueller Unixzeit und der bei der Änderung gestoppten Zeit ergeben somit die genaue Betriebszeit. Die gesamte Betriebszeit wird im Anschluss, wenn die Statusvariable auf *false* gesetzt wird, zusammen gerechnet und gespeichert. Die Gesamtzeit und die Zeitdifferenz müssen auf unterschiedlichen Variablen gespeichert werden, da bei einer erneuten Änderung des Status der Stempel erneut festgehalten wird und die Differenz somit wieder von null an gezählt wird. Die erfassten Zeiten werden dann zur Berechnung des Energieverbrauchs benutzt.

Das System 2 ist nicht in der Lage eine Phasenverschiebung zu erkennen, da dieses System nur aus einem IO-Server aufgebaut wurde. Die Leistung wird nach dem Netzteil berechnet. Es handelt sich hierbei um 24V DC, somit gibt es auch keine Verschiebung der Phasen, welche angegeben werden könnte. Der Energieverbrauch über die Steckdose wird separat betrachtet, da dort 230V AC anliegen. Allerdings kann auch dort kein Winkel berücksichtigt werden. Dieses System soll eine andere Methode der Berechnung und Übertragung zeigen.

Da die Leistung des Systems sich durchgehend ändern kann, ist es nicht möglich den Stromverbrauch durch Multiplikation von Spannung, Strom und der Zeit, zu bestimmen. Der aktuelle Verbrauch wird somit in regelmäßigen Intervallen abgespeichert. Im Anhang wird eine mögliche Variante gezeigt wie mit

Funktionsbausteinen diese Problematik gelöst werden kann. Die momentane Leistung wird mit Hilfe eines Auswahl-Moduls immer dann mit der bereits verbrauchten Energie addiert, wenn das Zeitglied ein *true* übergibt. Da das Glied ebenfalls in Sekundentakt arbeitet erhält man die verbrauchte Energie in Wattsekunde [Ws]. Durch dieses Verfahren wird der Merker auf dem die verbrauchte Energie gespeichert ist immer wieder mit dem aktuellen Energieverbrauch überschrieben. IP-Symcon greift diesen, wie üblich über ein Script mit beliebiger Wiederholungsfrequenz ab und speichert ihn im System.

Das Metering der Steckdosen funktioniert nach dem gleichen Prinzip. Die Berechnung übernimmt die Gebäudeautomation selbst. Die Werte wie Betriebsdauer, Energieverbrauch, Strom und aktuell anliegende Leistung werden allerdings zudem auf die gewünschten Größen umgewandelt (h, kWh, A, W). Für jeden zu übertragenden Wert ist ein Webbefehl angelegt worden. Dieser sendet die Werte bei jeder Änderung direkt an IPS. Lediglich Variablen müssen in der Leittechnik angelegt sein. In diesem Fall bedarf es keinerlei Programmierung in IPS. Alle verwendeten Variablen sind vom Typ „float“, damit diese auch in der Lage sind Dezimalzahlen anzeigen zu können. Nur Zustand und Betriebsdauer in Sekunden werden als Integer wiedergeben.

5.2 ANWESENHEIT

Die Anwesenheit kann durch viele verschiedene Faktoren erkannt werden (Bewegungsmelder, Türkontakte, Taster, etc). Diese Informationen können z.B. dafür verwendet werden die Beleuchtung ein bzw. auszuschalten oder andere Funktionen von der Gebäudeautomation auszuführen. Mit Hilfe einer Gebäudeleittechnik können zudem die Daten für alle Teilnehmer (wenn gewünscht) sichtbar gemacht werden. In diesem Fall werden die erkannten Daten an IPS gesendet und dort weiter verarbeitet.

In dieser Arbeit werden die Signale für die Anwesenheit durch Taster, Funk-Kartenschalter oder externer Software übergeben. Jedes dieser Signale könnte natürlich ebenfalls durch einen Bewegungsmelder oder anderen Kontakten ausgelöst werden. Mit Hilfe der Funk-Kartenschalter der Firma „EnOcean“, wird gezeigt wie sich z.B. eine Eingangskontrolle mit Zeiterfassung realisieren lässt. In der Programmierung des Bus-Systems wird der Eingang der EnOcean Unit über die API mit IPS verbunden. Durch diesen Vorgang erkennt IPS bei Verwendung des Kartenschalters die

Anwesenheit. Ist die Statusvariable somit auf *true* gesetzt, so wird der Unixstempel verwendet um die Dauer der Anwesenheit zu erfassen. Befinden sich der Kartenschalter z.B. am Eingang eines Gebäudes so kann durch dessen Aktivierung u.a. die Beleuchtung des Flurs oder andere Aktionen eingeschaltet werden. Beim Verlassen des Gebäudes wird durch erneuter Verwendung des Kartenschalters der Zustand wieder auf *false* gesetzt und die Zeiterfassung gestoppt sowie gespeichert. Weitere Funktionen können sein, dass den anderen Gebäudeautomationen dieser Status ebenfalls übermittelt wird. Durch die Darstellung der Variablen in dem WebFront kann jeder Teilnehmer der Gebäudeleittechnik abrufen, wann und wo sich jemand befindet. Durch die Verwendung folgender Funktion,

`„boolean Sys_Ping (string $Host, integer $Timeout)“`

Tätigt IP-Symcon eine Ping-Abfrage an den angegebenen Host. Kann diese erfolgreich durchgeführt werden, so ist dieser Wert *true* und bedeutet, dass sich das angepingte Gerät im Netzwerk befindet und eingeschaltet ist. Durch dieses Diagnose-Werkzeug kann jedes IP-Netzwerkfähige Gerät überprüft werden. Es lassen sich so die Zustände von Desktop PCs bzw. die Anwesenheit von mobilen Geräten ermitteln.

Die Anwesenheitskontrolle kann zudem dazu genutzt werden, unerlaubten Aufenthalt zu erfassen und dem entsprechend zu reagieren. Durch SMTP-Instanzen ist IPS in der Lage Nachrichten über E-Mailserver zu versenden. Wird somit eine Anwesenheit erkannt erfolgt eine Benachrichtigung und es kann entschieden werden was als nächstes getan werden soll. Diese Funktion kann u.a. zur Einbrucherkennung genutzt werden.

5.2.1 AUTOMATION DURCH TASKER

Eine weitere Variante um die aktuelle oder baldige Anwesenheit dem System, von mobilen Teilnehmern zu übermitteln, ist der Einsatz von „Tasker“. Hierbei handelt es sich um eine App, mit dessen Hilfe sich Android Geräte automatisieren lassen. Mit dieser App lassen sich alle Eigenschaften des Handys oder Tablets verwenden um diesen Zuständen, Aufgaben (Task) zuzuordnen. Auf den gesamten Umfang dieser App wird in dieser Arbeit nicht weiter eingegangen. Es wird nur anhand eines Beispiels erläutert wie diese Anwendung Einfluss auf die Gebäudeleittechnik haben kann.

Bezüglich des Themas dieser Arbeit, steht die Kommunikation mit der Gebäudeleitsoftware im Vordergrund. Wie schon bekannt, steht einem die JSON Schnittstelle von IPS zur Verfügung. Genau wie COMEXIO ist es mit Tasker möglich

einen http-POST-Request an einen Server zu senden. Dazu wird in der App ein neuer Task angelegt und die gewünschten Aktionskategorien hinzugefügt. Um IPS zu erreichen wird unter „Netzwerk“ die „http POST“-Aktion ausgewählt. Wie auch bei den Webbefehlen müssen hier bei der Angabe des Servers und Ports die Fernzugriffsdaten angegeben werden. Der Pfad an den die Abfrage enden soll muss zur API von IPS führen („/api/“). Die zu übergebenen Daten werden wie gewohnt als String unter „Data/File“ eingetragen. Der Content-Type (Inhalts-Typ) wird dem anderen System mit „application/json“ mitgeteilt. Dadurch ist IPS in der Lage die Daten auch richtig zu verarbeiten. Im Anschluss werden Profile angelegt, welche die Bedingungen für die Task sind.

Als Bedingung lassen sich sämtlich Einschalten bzw. Zustände verwenden, die das Mobilgerät erkennen kann. Für die Anwesenheit kann die App den Namen und die MAC-Adresse des verbundenen WLANs überprüfen. Stimmt dieses mit dem eigenen Heim-Netz überein sendet die Anwendung den Request an IPS und setzt so z.B. eine Statusvariable auf *true*.

Um die baldige Anwesenheit IPS mitzuteilen, ist die App u.a. in der Lage GPS Daten oder den verwendeten Funkmast zu überprüfen und kann somit dem System frühzeitig eine Ankunft mitteilen. Ebenfalls ist es möglich die Kamera des Gerätes als Bewegungsmelder umzufunktionieren.

5.3 ZUGRIFF AUF DIE GEBÄUDELEITTECHNIK

Damit die Gebäudeleittechnik durch andere Geräte erweitert werden kann, müssen alle Teilnehmer dementsprechend konfiguriert werden. Für den Zugriff auf die Weboberflächen von IP-Symcon, sowie die der Gebäudeautomation von COMEXIO, stehen neben den Standard-Browsern auch noch eigene Apps zur Verfügung. Über diese ist es möglich mit mobilen Geräten (z.B. Handy oder Tablet) auf die zugehörige Visualisierung zuzugreifen. Von dort ist somit eine Steuerung der Leuchttechnik möglich.

Die Einrichtung dieser Applikationen wird hier nur am Rande erläutert, da diese nicht explizit notwendig sind um eine Gebäudeleittechnik aufzubauen.

5.3.1 APPs

COMEXIO:

Da das COMEXIO-System vollständig visuell eingerichtet bzw. dargestellt werden kann, ist es durch die App „COMEXIO“ (Android/iOS) möglich, auf die Weboberfläche zu gelangen. So lassen sich über diese App die angelegten Dashboards im Studiobereich abrufen und steuern. Der Admin-Bereich ist über diesen Weg nicht zugänglich, es können somit keine Veränderungen an der Systemkonfiguration vorgenommen werden. Damit man auf dem richtigen Dashboard landet, können die IO-Server in der App bei Bedarf einzeln hinzugefügt werden. Dies kann manuell durch Eingabe der Lokalen IP-Adresse oder durch die automatische „WLAN-Suche“ durchgeführt werden. Die App erkennt dabei alle COMEXIO-Server im Netzwerk. Um einen Zugriff von außerhalb zu ermöglichen, muss manuell die „Globale-Adresse“ eingegeben werden. Die Gebäudeautomation ist somit auch außerhalb des Netzwerkes erreichbar.

IP-Symcon:

Um auf die Gebäudeleittechnik IP-Symcon zugreifen zu können, kann ebenfalls eine Application für Android oder iOS Geräte genutzt werden. Hierbei ist zu beachten, dass der Zugriff im WebFront-Konfigurator vorab aktiviert werden muss. Dafür muss in der Verwaltungskonsole der Visualisierungskonfigurator aufgerufen werden. Dort wird das gewünschte WebFront ausgewählt und unter dem Reiter „Mobile/Retro“ der mobile Zugang aktiviert. Ebenfalls kann dort der Bereich angegeben werden, der für den mobilen Zugriff erlaubt ist. Da diese App das WebFront von IP-Symcon abrufen, folgt es derselben Struktur wie es durch die Konsole vorgegeben wird. Die App („IP-Symcon Mobile“) dient ähnlich wie der von COMEXIO nur als Erweiterung des Systems, es sind auch in dieser Anwendung nicht alle Funktionen in vollem Umfang vorhanden. Das System kann zwar gesteuert werden, aber die Darstellung von Scripten, Variablen, Anzeigen usw. unterscheidet sich von der Ansicht in einem Browser deutlich. Der Datenverlauf, in Form von Graphen kann über diese App z.B. nicht angezeigt werden.

5.3.2 BROWSER

COMEXIO:

Wie auch bei der App ist es möglich, die Weboberfläche des IO-Servers durch dessen IP-Adresse zu erreichen. Dabei wird diese Adresse in die Adresszeile eines Browser eingegeben. Über den Browser lassen sich genau wie auf den mobilen Geräten die

visuellen Komponenten in vollem Umfang nutzen. Wie schon erwähnt wurde, ist über den einen Browser auch der Zugang zum Admin-Bereich möglich. Es kann somit die Konfiguration des Systems verändert werden. Es spielt dabei keine Rolle, ob der verwendete Browser sich auf einem Computer oder einem Mobilgerät befindet. Einen Fernzugriff erhält man bei dieser Variante über die „COMEXIO Remote“-Funktion, welche über einen SynDNS Service den Zugang zu der Gebäudeautomation weltweit ermöglicht. Als Alternative kann auch ein VPN-Tunnel eingerichtet werden um in das entsprechende Netzwerk zu gelangen.

IP-Symcon:

Durch die Verwendung eines Browsers, kann das Webfrontend von IP-Symcon abgerufen werden. Der volle Umfang dieser Anwendung steht bei diesem Aufruf zur Verfügung. Die Darstellung aller Komponenten bzw. Elementen wird genauso angezeigt, wie sie in der Verwaltungskonsole oder im Webeditor programmiert wurde. Eine Programmierung des Gebäudeleitsystems ist über die Weboberfläche nicht möglich. Nur die Konfiguration der WebFront selbst kann bearbeitet werden. Auf der Startseite der Oberfläche muss lediglich das Bearbeitungssymbol gedrückt werden. Dort lässt sich dann wie in der Konsole, die Struktur anlegen und die Elemente die angezeigt werden sollen definieren.

Das WebFront steht unter der Standard Adresse „http://127.0.0.1/“ mit dem Port „82“ zur Verfügung. Die Oberfläche kann allerdings auch durch die IP-Adresse des Servers bzw. Computers, auf dem IPS läuft, erreicht werden. Es muss nur der Port an die Adresse angehängt werden (z.B. „http://192.168.179.2:82/“). Befindet man sich im selben Netzwerk kann auch der Servername (Computername) verwendet werden, z.B. http://Heimserver:82/. Um von außerhalb des Netzwerkes Zugriff zu bekommen, muss diese Einstellung in der Tray-Anwendung zuerst aktiviert werden. Der Zugang ist dann nur mit dem angelegten Passwort und dem Lizenznamen möglich.

6 ERGEBNIS

Durch die Einführung der Programmierschnittstelle war die Frage zu Beginn dieser Arbeit, ob es mit dem COMEXIO-Bus möglich ist eine Gebäudeleittechnik aufzubauen. Vor der Einführung der API stand es dem System nicht zur Verfügung sich über das Ethernet mit anderen System auszutauschen. Der Bus und auch der Funktionsumfang waren somit in sich beschränkt und konnten nicht wirklich erweitert werden. Es war schwierig Großprojekte zu realisieren.

Die Aufgabe dieser Arbeit bestand darin, die Funktionsfähigkeit der neuen API zu testen und zu erkennen in welchem Umfang sich diese verwenden lässt. Dabei sollte die Kommunikation zwischen COMEXIO-Systemen und einer übergeordneten Gebäudeleittechnik ermöglicht werden. Der bestehende Versuchsaufbau (System 1) wurde im Rahmen dieses Projekts durch ein weiteres COMEXIO-System (System 2) erweitert. Der neue Aufbau soll Informationen über das Ethernet bzw. über die API erhalten und senden können. Die erhaltenen Zustände werden über die LEDs auf der Präsentationsplatte dargestellt. Die Leitebene wurde durch die Software IP-Symcon aufgebaut und hat somit den Funktionsumfang der beiden Gebäudeautomationen ergänzt. Es wurde gezeigt wie sich Daten zwischen all den Teilnehmern dieses Projektes senden bzw. abrufen lassen. Zur Verfügung stand hierbei eines der bekanntesten Übertragungsprotokolle, http. Es wurde gezeigt wie sich mit Hilfe der http-GET und http-POST Methode digitale sowie auch analoge Daten zwischen den einzelnen Systemen austauschen lassen. Dieser Austausch an Informationen wurde an Hand von Anwesenheits- und Metering-Funktionen genauer dargestellt und erläutert. Durch die Möglichkeit der Kommunikation über Ethernet mit anderen Systemen ist es mit COMEXIO möglich Großprojekte zu realisieren.

Eine Gebäudeautomation mit COMEXIO bietet meines Erachtens einige Vorteile mit sich. Das System ist so aufgebaut, dass dieses von den Bauherren zur Not oder bei Belieben auch selber programmiert werden kann. Durch die Verwendung der Funktionsbausteine sind keine großen Programmierkenntnisse notwendig. Das System ist zudem im Vergleich zu anderen recht günstig und bietet alleine mit der Server-Einheit schon die Möglichkeit Aktoren und Sensoren zu verwenden. COMEXIO verfolgt mit ihrem System, die Idee die Standards der anderen Gebäudeautomations-systeme zusammen zu führen. Kein System bietet in allen Bereichen die optimalste

Lösung an, erst die Kombination mehrerer Systeme würde eine perfekte Automation ermöglichen. Durch die Schnittstellen, welche COMEXIO zur Verfügung stellt lassen sich andere Busse in das System ohne Zwischeninstanzen integrieren. So können alleine durch den IO-Server der KNX- und der 1-Wire-Bus an das System angeschlossen werden. Durch das EnOcean Modul, lässt sich ebenfalls dessen Standard dem Bus zur Verfügung stellen. Der Funktionsumfang und dessen Flexibilität werden durch diese Schnittstellen deutlich erhöht.

Ein weiterer positiver Aspekt ist gleichzeitig auch ein Nachteil, der webbasierte Zugriff. Die Programmierung wird über die Weboberfläche des IO-Servers getätigt. Es ist somit keine weitere Software notwendig. Allerdings wird dadurch auch die Geschwindigkeit der Programmierung beeinträchtigt. Bei jedem hin und herschalten zwischen den Menüs und Reitern benötigt das System Zeit um die Seite zur Verfügung zu stellen. Teilweise waren die angezeigten Browserseiten nicht mehr aktuell und die Programmierung wurde nicht übernommen, sodass die Verbindungen der Funktionsbausteine zwar angezeigt wurden, aber nach der Aktualisierung nicht mehr vorhanden waren. Hier wäre zu der Browserprogrammierung noch eine kleine Software für den PC zu wünschen. Dort sollte man mit Split Screen und einem Debugger schneller arbeiten können.

IP-Symcon wurde in dieser Arbeit dazu verwendet die Systeme miteinander in eine Gebäudeleittechnik einzubinden. Wie auch beim COMEXIO-System, soll es mit IPS möglich sein alle Einzelsysteme in ein Gesamtsystem zusammen zuführen. Auf dem Server der Leittechnik können somit alle wichtigen Daten gespeichert und verarbeitet werden ohne dass sie bei einem Systemausfall verloren gehen würden. Der Funktionsumfang von IP-Symcon kann zusätzlich dazu genutzt werden um die Gebäudeautomationen um diesen zu erweitern. Die Software kann zudem genutzt werden, um verschiedene Systeme über ihre APIs miteinander kommunizieren zu lassen. IPS dient dabei als Gateway und macht dem anderen System somit die gewünschten Aktionen verständlich.

Das Hauptproblem um eine komplexe Gebäudeautomation bzw. Leittechnik aufzubauen, sind die unterschiedlichen Standards jedes einzelnen Herstellers. Mit Hilfe von IPS und den Schnittstellen von COMEXIO kann diese Problematik zwar etwas umgehen werden. Sinnvoller wäre allerdings ein gemeinsamer Standard für alle Systeme zu finden.

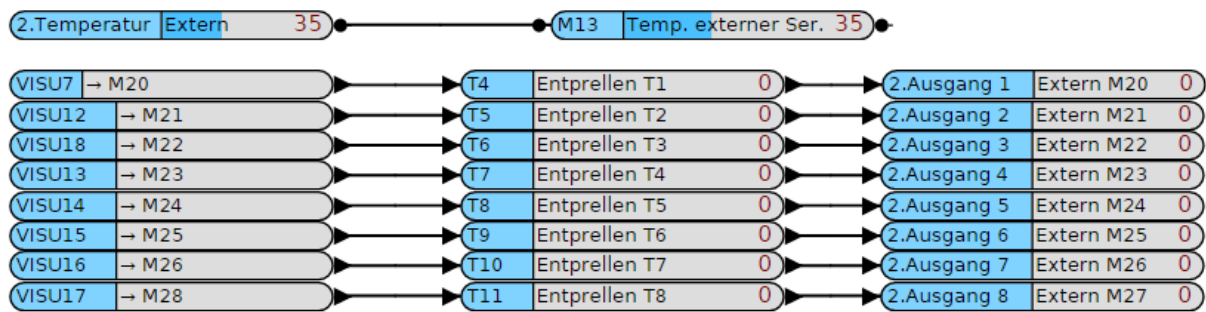


ABBILDUNG 7.3: FUNKTIONSPLAN SYSTEM 1 / VIUS EINGÄNGE AUF API ZUGRIFF

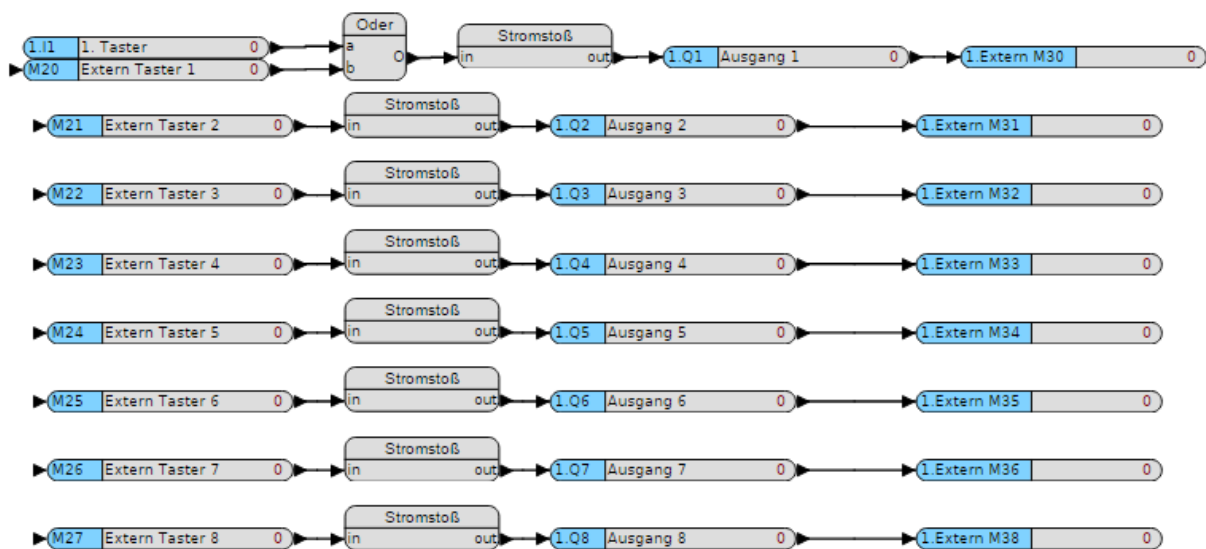


ABBILDUNG 7.4: FUNKTIONSPLAN SYSTEM 1 / SCHALTEN DER LEDS

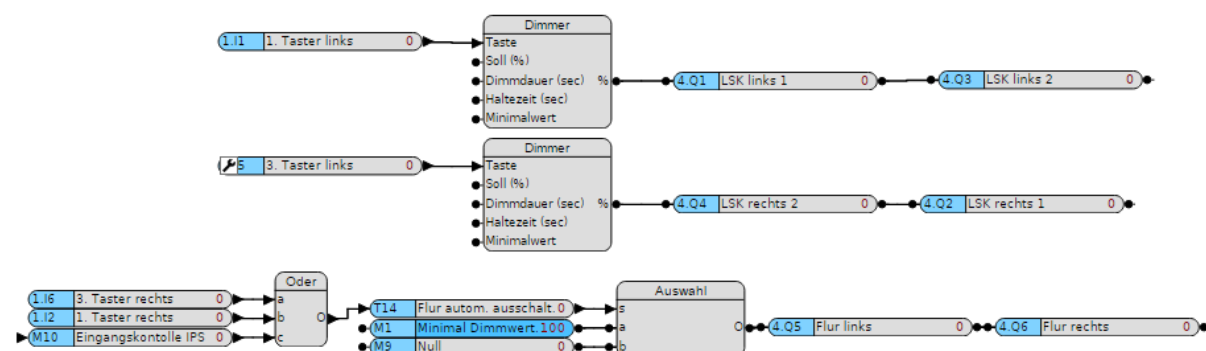


ABBILDUNG 7.5: FUNKTIONSPLAN SYSTEM 1 / BELEUCHTUNG

Den Webbefehl Energieverbrauch zu IPS bearbeiten

Name	Energieverbrauch zu IPS
Variablen-Typ	Analog
Minimaler Wert	0
Maximaler Wert	100000
Parameter	/api/
Header Modifizierung	Content-Type:
Html-Daten	<pre>function data (v) return '{"jsonrpc":"2.0","method":"SetValue","params": [22769,'..v..'],'id":1}' end</pre>
Typ	Ausgang
TCP-Port	82
Aufrufmethode	POST
Authentifizierung	Ja
Wiederholfrequenz	
Antwortinterpretier	

ABBILDUNG 7.6: WEBIO SYSTEM 2 / POST AN JSON RPC

7.1.2 IP-SYMCON QUELLCODE

```
<?
$url_strom      = "http://192.168.178.50/api/?action=get&ext=IO-
Server_local&io=QI7";
$url_winkel     = "http://192.168.178.50/api/?action=get&marker=M41";

if (GetValue(44071)==True)
{
    $winkel = floatval (Sys_GetURLContent ($url_winkel));
    SetValue(37510,$winkel);

    $strom = floatval (Sys_GetURLContent ($url_strom));
    SetValue(21085,$strom);

    $wirk = 230*$strom*cos(deg2rad($winkel));
    SetValue(51900,$wirk);

    $blind = 230*$strom*sin(deg2rad($winkel));
    SetValue(10982,$blind);

    $schein = 230*$strom;
    SetValue(53890,$schein);

    $zeit_s = GetValue (28401)+GetValue (40362);
    $zeit_h = $zeit_s / 3600;
    $kwh = ($wirk/1000)*$zeit_h;
    SetValue (43551,$kwh);
    $kosten = $kwh*GetValue(19162);
    SetValue(24381,$kosten);
}
?>
```

ABBILDUNG 7.7: IPS QUELLCODE / METERING SYSTEM 1

```
<?
if (GetValue(44071) == true)
{
    $trueSince = time() - IPS_GetVariable(44071) ['VariableChanged'];
    SetValue (28401,$trueSince);
}
else if (GetValue(44071) == false)
{
    SetValue (40362,GetValue(28401)+GetValue(40362));
    SetValue(28401,0);
}
?>
```

ABBILDUNG 7.8: IPS QUELLCODE / ZEITGEBER (UNIXSTAMP)

```

<?
//Variablen + ID anlegen:
$Variable_Temp1 = 19507;
$Variable_Temp2 = 44560;
//URL unter Variable "$URL" speichern:
$URL1 = "http://192.168.178.50/api/?action=get&ext=IO-
Extension_local&io=TI1";
$URL2 = "http://192.168.178.50/api/?action=get&ext=IO-
Extension_local&io=TI2";
//Intervall Zeit in Sekunde
$Intervall = 30;
//Inhalt von URL:
$cont1 = Sys_GetURLContent($URL1);
$cont2 = Sys_GetURLContent($URL2);
//Inhalt in Variable setzen:
SetValue($Variable_Temp1, $cont1);
SetValue($Variable_Temp2, $cont2);
//Wiederholungsintervall:
if ($_IPS['SENDER'] == "Execute")
{
    IPS_SetScriptTimer($_IPS['SELF'], $Intervall);
}
?>

```

ABBILDUNG 7.9: IPS QUELLCODE / TEMPERATURABFRAGE SYSTEM 1

```

<?
if (GetValue(27478)==true)
{
    IPS_SetScriptTimer (37174,1);

    $zeit = time() - IPS_GetVariable(27478) ['VariableChanged'];
    SetValue (33133,$zeit/3600);
    if($zeit <= 1 )
    {
        $an =
"http://192.168.178.50/api/?action=set&marker=M10&value=1";
        $aus =
"http://192.168.178.50/api/?action=set&marker=M10&value=0";
        Sys_GetURLContent($an);
        Sys_GetURLContent($aus);
    }
}
else if(GetValue(27478)==false)
{
    SetValue (15620,GetValue(33133)+GetValue(15620));
    IPS_SetScriptTimer (37174,0);
}
?>

```

ABBILDUNG 7.10: IPS QUELLCODE / ANWESENHEIT SYSTEM 1

```

<?
$ip='192.168.178.22';
$ping = Sys_Ping($ip,1000);
SetValue(39522,$ping);

if($ping==1)
{
    $m27_on = "http://192.168.178.51/api/?action=set&ext=COMEXIO-
Server&io=Q8&value=1";
    Sys_GetURLContent($m27_on);
}
else if ($ping==0)
{
    $m27_off = "http://192.168.178.51/api/?action=set&ext=COMEXIO-
Server&io=Q8&value=0";
    Sys_GetURLContent($m27_off);
}
?>

```

ABBILDUNG 7.11: IPS QUELLCODE / ANWESENHEIT DURCH PINGABFRAGE

```

<?
$url_strom      = "http://192.168.178.51/api/?action=get&ext=COMEXIO-
Server&io=QI8";
$url_energie    = "http://192.168.178.51/api/?action=get&marker=M42";
$url_leistung   = "http://192.168.178.51/api/?action=get&marker=M40";

$strom = floatval (Sys_GetURLContent($url_strom));
$energie = floatval (Sys_GetURLContent($url_energie));
$leistung = floatval (Sys_GetURLContent($url_leistung));

    SetValue (23455,$strom);
    SetValue (34884,$energie);
    SetValue (30961,$leistung);
?>

```

ABBILDUNG 7.12: IPS QUELLCODE / METERING SYSTEM 2

```

<?
SetValue($_IPS['VARIABLE'], $_IPS['VALUE']);
$soll = GetValue (13150);
$url = "http://192.168.178.51/api/?action=set&marker=M40&value=$soll";

Sys_GetURLContent ($url);
?>

```

ABBILDUNG 7.13: IPS QUELLCODE / SENDEN EINES ANALOGWERTES

```

<?
$url1 = "http://192.168.178.51/api/?action=set&marker=M20&value=1";
$url0 = "http://192.168.178.51/api/?action=set&marker=M20&value=0";
$geturl= "http://192.168.178.51/api/?action=get&marker=M20";
$wert = Sys_GetURLContent ($geturl);
SetValue (34699, $wert);
$zustand = GetValue (34699);

IF ($zustand == false)
{
    Sys_GetURLContent ($url1);
}
else //oder IF ($zustand == true)
{
    Sys_GetURLContent ($url0);
}
?>

```

ABBILDUNG 7.14: IPS QUELLCODE / TASTER DURCH IF-ANWEISUNG

```

<?
SetValue($_IPS['VARIABLE'], $_IPS['VALUE']);
$var1 = GetValue (53611);

switch($var1)
{
    case "0":
        IPS_RunScript(58219);
        break;
    case "1":
        IPS_RunScript(58761);
        break;
}
?>

```

ABBILDUNG 7.15: IPS QUELLCODE / TESTER DURCH SWITCH/CASE –ANWEISUNG

IP-Symcon Verwaltungskontrolle

Ansicht hinzufügen Aktualisieren Hilfe Über...

Objektbaum

Objekt hinzufügen Suche ID Ansicht Spalten Filter Schnellfilter

ObjektID	Standort/Objekt	Typ	Wert	Aktualisiert
0	IP-Symcon			
27077	Gebäudeleitsystem			
25630	Anwesenheit Ges.			
46403	System 1			
34999	Anwesenheit			
33133	Aktuelle Zeit	Float	0,008 h	19:02:57
59248	Anwesenheit	Boolean	False	Nie
27478	Eingangskontrolle	Boolean	False	19:02:57
15620	Gesamte Zeit	Float	1002,008 h	19:02:57
37174	Anwesenheit	Script	37174.ps.php	19:02:57
51444	SmartMeter			
10982	Blindleistung	Float	-11,127Var	18:53:53
26401	Dauer seit Aktualisierung	Integer	0	16:55:33
43551	Energieverbrauch	Float	0,0275kWh	18:53:53
40362	Gesamtdauer	Integer	3425s	16:55:33
37510	Phasenverschiebung	Float	-21,0°	18:53:53
19162	Preis / kWh	Float	0,15	17:53:23
53890	Scheinleistung	Float	31,05	18:53:53
44071	Status Q7 AH/AUS	Boolean	False	18:53:53
21085	Strom	Float	0,135A	18:53:53
24381	Verbrauchskosten	Float	0,004€	18:53:53
51900	Wirkleistung	Float	28,988W	18:53:53
28257	Metering System 1	Script	28257.ps.php	16:54:58
49005	Ereignis: Täglich alle 1 Sekunden	Ereignis		16:55:33
37955	Timer	Script	37955.ps.php	16:55:33
10319	Ereignis: Täglich alle 1 Sekunden	Ereignis		16:55:33
20416	Temperaturen			
10507	Temperatur Innen	Integer	Temperatur: 27.0°C	19:58:10
44560	Variable_Temp2	Integer	Temperatur: 28.0°C	19:58:10
17658	Temp. Script	Script	17658.ps.php	19:58:10
13161	ScriptTimer (Ereignis: Täglich alle 30 Sekunden)	Script		19:58:09
54471	System 2	Ereignis		

Willkommen Meldungen Objektbaum

ABBILDUNG 7.16: IPS OBJEKTBAUM

7.1.3 TASKER APPLICATION

Aktion Bearbeiten
1. HTTP Post

AKTION WECHSELN ? HILFE X ABBR.

Server:Port

sandra.stahlberg@fh-dortmund.de:Marcel24!@192.168.178.32:82

Pfad

/api/

Data / File

{ "jsonrpc": "2.0", "method": "SetValue", "params": [12243, 1], "id": 1 }

Cookies

Optional

User Agent

Optional

Timeout

30

Inhalts Typ

application/json

Ausgabe Datei

Optional

Trust Any Certificate

If

ABBILDUNG 7.17: TASKER / ANWESENEHIT DURCH HTTP-POST

7.2 SCHALTPLÄNE

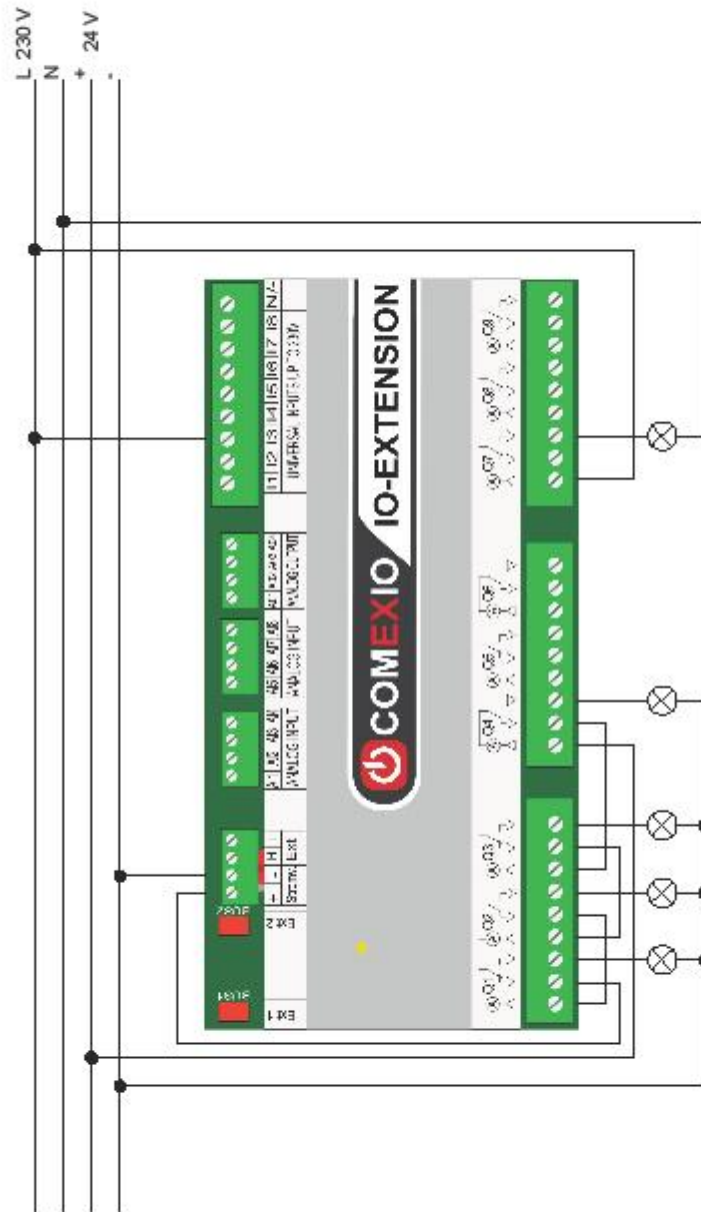


ABBILDUNG 7.18: SCHALTPLAN FÜR IO-SERVER SYSTEM 1

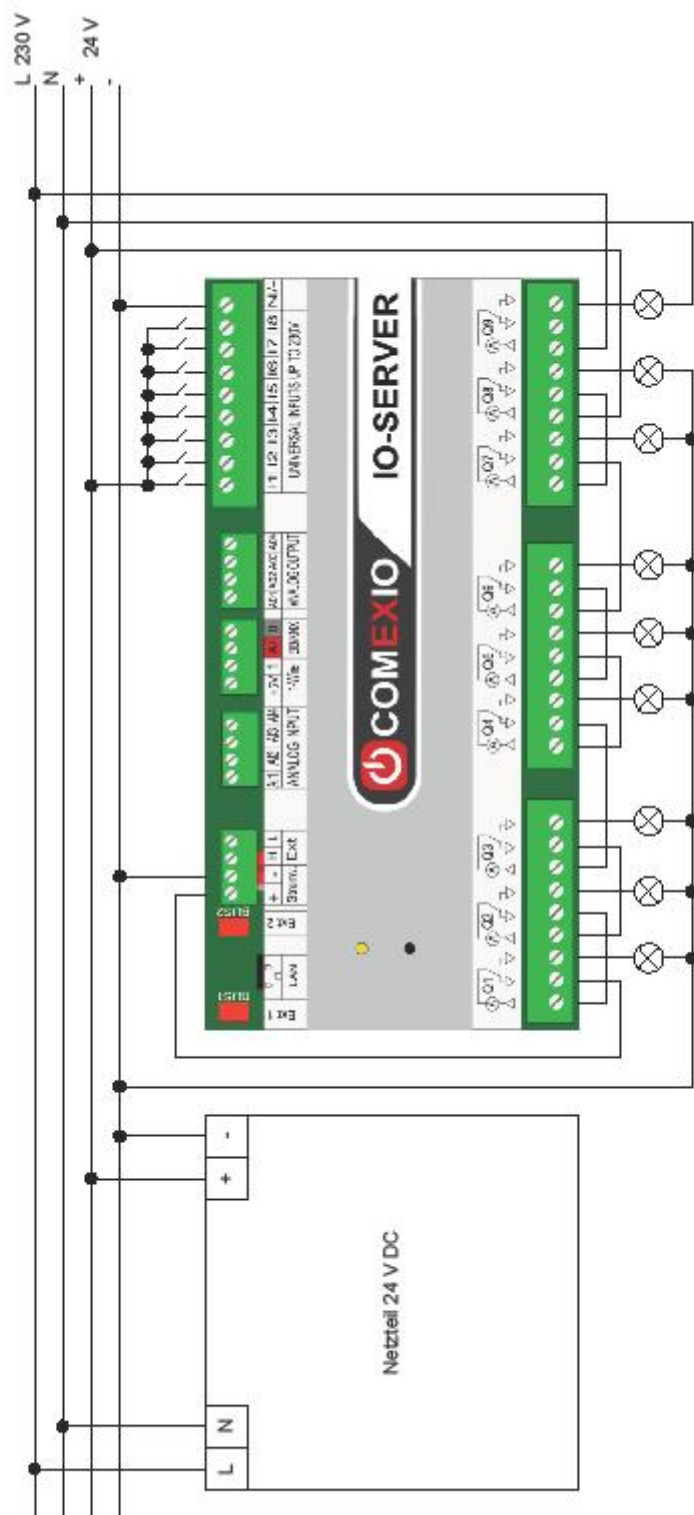


ABBILDUNG 7.19: SCHALTPLAN SYSTEM 2

8 ABKÜRZUNGEN

A	Ampere
API	application programming interface - Schnittstelle zur Anwendungsprogrammierung
AWL	Anweisungsliste
Bus	Binary Unit System
FTP	File Transfer Protocol
FUP	Funktionsplan
GA	Gebäudeautomation
GLT	Gebäudeleittechnik
h	Stunde
HTML	Hypertext Markup Language
IPS	IP-Symcon
kWh	Kilowattstunde
PHP	Hypertext Preprocessor (Scriptsprache)
s	Sekunde
URL	Uniform Resource Locator
V	Volt
VA	Volt Ampere, Einheit der Scheinleistung
var	Einheit der Blindleistung
W	Watt
WebFront	WebFrontend

9 LITERATURVERZEICHNIS

- [1] T. Theis, Einstieg in PHP 5.6 und MySQL 5.6, Bonn: Galileo Computing, 2014.
- [2] J. Wolf, HTML5 und CSS3: Das umfassende Handbuch, Bonn: Galileo Press, 2015.
- [3] Symcon GmbH, „Dokumentation,“ 18 Januar 2016. [Online]. Available: <https://www.symcon.de/service/dokumentation/befehlsreferenz/variablenzugriff/getvalue/>.
- [4] Symcon GmbH, „Dokumentation,“ 17 Januar 2016. [Online]. Available: <https://www.symcon.de/service/dokumentation/befehlsreferenz/variablenverwaltung/variablenprofile/ips-setvariableprofileassociation/>.
- [5] Symcon GmbH, „Dokumentation,“ 8 Januar 2016. [Online]. Available: <https://www.symcon.de/service/dokumentation/modulreferenz/system-informationen/sys-geturlcontent/>.
- [6] Symcon GmbH, „Dokumentation,“ 1 Januar 2016. [Online]. Available: <https://www.symcon.de/service/dokumentation/befehlsreferenz/>.
- [7] Symcon GmbH, „Dokumentation,“ 18 Januar 2016. [Online]. Available: <https://www.symcon.de/service/dokumentation/konzepte/variablen/variablenprofile/>.
- [8] Symcon GmbH, „Dokumentation,“ 18 Januar 2016. [Online]. Available: <https://www.symcon.de/service/dokumentation/befehlsreferenz/ablaufsteuerung/ips-runsript/>.
- [9] Symcon GmbH, „Dokumentation,“ 19 Januar 2016. [Online]. Available: <https://www.symcon.de/service/dokumentation/befehlsreferenz/variablenzugriff/SetValue/>.
- [10] A. Pelzeter, „HWR Berlin,“ 2 Januar 2016. [Online]. Available: <http://www.hwr-berlin.de/fachbereich-duales-studium/studiengaenge/technisches-facility-management/>.
- [11] GIGA, „Industrie- und Gebäudeautomation,“ 2 Januar 2016. [Online]. Available: http://www.giga-automation.de/uploads/pics/SAIA_EM_Pyramide_02.jpg.
- [12] Baunetzwissen, „Das Online Fachlexikon,“ 2 Januar 2016. [Online]. Available: http://www.baunetzwissen.de/index/Elektro-Bussysteme_33387.html.
- [13] COMEXIO GmbH, „COMEXIO,“ 3 Januar 2016. [Online]. Available: <http://www.comexio.com/ueber-uns/>.
- [14] COMEXIO GmbH, „COMEXIO,“ 3 Januar 2016. [Online]. Available: <http://www.comexio.com/blog/>.
- [15] Baunetzwissen, „Planungsgrundlagen,“ 3 Januar 2016. [Online]. Available: http://www.baunetzwissen.de/standardartikel/Elektro-Smart-Grids_1636653.html.

- [16] COMEXIO GmbH, „COMEXIO,“ 5 Januar 2016. [Online]. Available: <http://www.comexio.com/firmware-download>.
- [17] COMEXIO GmbH, „COMEXIO,“ 5 Januar 2016. [Online]. Available: <http://www.comexio.com/handbuch-download/>.
- [18] „PHP.lernenhoch2,“ 6 Januar 2016. [Online]. Available: http://php.lernenhoch2.de/lernen/php-anfaenger/formulare-erstellen-und-auswerten/unterschied-zwischen-get-und-post/#weiter_links.
- [19] A-Coding-Projekt.de, „HTTP: Request-Methoden,“ 6 Januar 2016. [Online]. Available: <http://www.a-coding-project.de/209/request-methoden/>.
- [20] FH Wedel, „fh-wedel.de,“ 9 Januar 2016. [Online]. Available: <http://www.fh-wedel.de/~si/seminare/ws09/Ausarbeitung/09.lua/lua3.htm>.
- [21] R. Ierusalimschy, „lua.org,“ 9 Januar 2016. [Online]. Available: <http://www.lua.org/pil/4.2.html>.
- [22] PHP Group, „PHP Manual,“ 16 Januar 2016. [Online]. Available: <http://php.net/manual/de/function.sleep.php>.
- [23] Symcon GmbH, „Dokumentation,“ 22 Januar 2016. [Online]. Available: <https://www.symcon.de/service/dokumentation/entwicklerbereich/datenaustausch/>.
- [24] Apfeltalk-Community, „apfeltalk.de,“ 25 Januar 2016. [Online]. Available: <http://www.apfeltalk.de/community/threads/http-post-request-json.465125/>.
- [25] PHP Group, „php.net,“ 3 Februar 2016. [Online]. Available: <http://php.net/manual/de/function.time.php>.
- [26] PHP Group, „php.net,“ 27 Januar 2016. [Online]. Available: <http://php.net/manual/fr/function.header.php>.
- [27] PHP Group, „php.net,“ 2 Februar 2016. [Online]. Available: <http://php.net/manual/de/function.sin.php>.
- [28] PHP Group, „php.net,“ 2 Februar 2016. [Online]. Available: <http://php.net/manual/de/function.number-format.php>.
- [29] handcoding web services, „unixtimestamp.de,“ 5 Februar 2016. [Online]. Available: <http://www.unixtimestamp.de/>.
- [30] Google Play ; Tasker, „play.google.com,“ 5 Februar 2016. [Online]. Available: <https://play.google.com/store/apps/details?id=net.dinglich.android.taskerm&hl=de>.
- [31] PHP Group, „php.net,“ 6 Februar 2016. [Online]. Available: <http://php.net/manual/de/function.deg2rad.php>.

Abbildungsverzeichnis

Abbildung 2.1: Automationspyramide	2
Abbildung 2.2: COMEXIO Funktionsplan	9
Abbildung 2.3: Webbefehl Bearbeitung	10
Abbildung 3.1: Alter Versuchsaufbau	16
Abbildung 3.2: Neuer Versuchsaufbau	18
Abbildung 4.1: Programmauszug WebIOs	21
Abbildung 4.2: Programmauszug System 1	22
Abbildung 4.3: Programmauszug System 2	22
Abbildung 4.4: Programmauszug System 2	23
Abbildung 4.5: Web-Eingang	23
Abbildung 4.6: Dashboardmodule	24
Abbildung 4.7: Darstellung durch Siebensegmentanzeige	24
Abbildung 4.8: Dashboard-Monitoring-Temperaturverlauf	25
Abbildung 4.9: IP-Symcon Objektbaum	26
Quelltext 1: IP-Symcon Taster	28
Abbildung 4.10: WebFront auszug	28
Quelltext 2: switch / case-Anweisung	29
Quelltext 3: IF-Anweisung	29
Quelltext 4: Analogwert senden	30
Quelltext 5: Temperaturabfrage	33
Abbildung 4.11: Tabell: JSON Bestandteile	34
Abbildung 4.12: COMEXIO PProgrammauszug	36
Abbildung 4.13: WebFront Auszug-Temperatur	37
Abbildung 4.14: Variableneditor/Profil-Manager	37
Abbildung 4.15: WebFront/Temperaturverlauf	38
Abbildung 4.16: WebFront / Variable mit Schrittweite, ohne Assoziation	38
Abbildung 4.17: WebFront / Variable mit Schrittweite, mit Assoziationen	39
Quelltext 6: Unix-Zeitstempel	41
Abbildung 7.1: Funktionsplan System 1 / Metering	i
Abbildung 7.2: Funktionsplan System 2 / Metering	i
Abbildung 7.3: Funktionsplan System 1 / Vius Eingänge auf API Zugriff	ii
Abbildung 7.4: Funktionsplan System 1 / Schalten der LEDs	ii
Abbildung 7.5: Funktionsplan System 1 / Beleuchtung	ii
Abbildung 7.6: WebIO System 2 / POST an JSON RPC	iii
Abbildung 7.7: IPS Quellcode / Metering System 1	iv
Abbildung 7.8: IPS Quellcode / Zeitgeber (Unixstamp)	iv
Abbildung 7.9: IPS Quellcode / Temperaturabfrage System 1	v
Abbildung 7.10: IPS Quellcode / Anwesenheit System 1	v
Abbildung 7.11: IPS Quellcode / Anwesenheit durch Pingabfrage	vi
Abbildung 7.12: IPS Quellcode / Metering System 2	vi
Abbildung 7.13: IPS Quellcode / Senden eines Analogwertes	vi
Abbildung 7.14: IPS Quellcode / Taster durch IF-Anweisung	vii
Abbildung 7.15: IPS Quellcode / Tester durch switch/case –Anweisung	vii
Abbildung 7.16: IPS Objektbaum	viii
Abbildung 7.17: Tasker / Anwesenheit durch http-POST	viii
Abbildung 7.18: Schaltplan für IO-Server System 1	ix
Abbildung 7.19: Schaltplan System 2	x

