# Abschlussarbeit


Zur Erlangung des Grades

eines

Bachelor of Engineering (B.Eng.)
Automatisierungstechnik

von Herrn Philipp Römer

Thema: Netzwerküberwachung mit SPS- bzw. SCADA Systemen
Beschreibung der Unterschiede anhand einer Beispielapplikation

Erstprüfer: Prof. Dr.-Ing. Peter Helm
Zweitprüfer: Dipl.-Phys. Erik Förster

Halle, den 14.05.2020

## Selbstständigkeitserklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Halle, den 14.05.2020

# Inhaltsverzeichnis

# Abkürzungsverzeichnis

| | |
|---|---|
| **AE** | Alarms & Events |
| **AWL** | Anweisungsliste |
| **COM** | Component Object Model |
| **DA** | Data Access |
| **DB** | Datenbaustein |
| **DCOM** | Distributed Component Object Model |
| **FB** | Funktionsbaustein |
| **FQDN** | Fully Qualified Domain Name |
| **FUP** | Funktionsplan |
| **GDS** | GlobalDiscoveryServer |
| **HDA** | Historical Data Access |
| **IT** | Informationstechnologie |
| **KOP** | Kontaktplan |
| **LDS** | LocalDiscoveryServer |
| **MAC** | Media Access Control |
| **ME** | Multicast Extension |
| **OB** | Organisationsbaustein |
| **OLE** | Object Linking and Embedding |
| **OPC** | Open Platform Communications |
| **OSI** | Open Systems Interconnection |
| **OT** | Operative Technologie |
| **PG** | Programmiergerät |
| **PubSub** | Publish/Subscribe |
| **SCADA** | Supervisory Control and Data Acquisition |
| **SFP** | Small Form-Factor Pluggable |
| **SCL** | Structured Control Language |
| **SPS** | Speicherprogrammierbare Steuerung |
| **TSN** | Time Sensitive Networking |
| **UA** | Unified Architecture |
| **XML** | Extensible Markup Language |

# Abbildungsverzeichnis

# 1 Einleitung und Vorstellung des Projektes

Für die Firma Delta Automation GmbH ist ein Programm zur Überwachung der IT-Komponenten, im Speziellen der Switches und Router, in einem Lagersystem, zu entwerfen und anschließend in eine bestehende Anlage zu implementieren.

Dieses Programm stellt den Abschluss eines Projektes in einem Lagersystem dar, welches seitens der IT von Delta Automation GmbH umgebaut wurde. Inhalt dieses Projektes war eine komplette Erneuerung der bestehenden IT. Die vorliegende Bachelorarbeit beinhaltet die Realisierung des Projektes der Netzwerküberwachung, sowie der Implementierung dieser in das laufende System.

Die Umsetzung über eine zentrale Speicherprogrammierbare Steuerung (SPS) zur Überwachung der IT ist sinnvoll, da hierdurch nur marginal in die bestehende Anlage eingegriffen werden muss. Es wird lediglich eine weitere SPS angebaut und nur ein zusätzliches Bild für die Überwachung in WinCC erstellt, inklusive der entsprechenden Alarmmeldungen. Die Kopplung der Geräte an die Überwachung muss permanent verfügbar sein und sowohl gegen Ausfall als auch gegen Angriffe von außen sicher sein. Aus diesem Grund bietet es sich an den OPC UA Kommunikationsstandard zu nutzen. Die Überwachung der IT-Komponenten schließt auch die Überwachung der Komponenten im Serverraum mit ein. So dürfen neben den Switches und Routern auch die Betriebsmittel, vor allem die Spannungsversorgung und die Kühlung der Geräte nicht vernachlässigt werden.

## 1.1 Zonenkonzept

In einem bereits bestehendem Lagersystem soll eine Überwachung des Serverraumes erfolgen. In dem Serverraum befinden sich sowohl diverse Rechner und Server als auch die verschiedenen Switches und Router, welche die Kommunikation der Server untereinander regeln. Die Verbindung zwischen den einzelnen Komponenten bezeichnet man als Netzwerkumgebung. Die Netzwerkumgebung dieses Lagersystems wurde komplett umgebaut. Das IT-System besteht seitdem aus 2 voneinander unabhängigen Netzen: dem Automatisierungsnetzwerk, der operativen Technik (OT) und dem Office-Netzwerk. In diesem Projekt wird das OT-Netz als Zone 4 und das Office-Netz als Zone 3 bezeichnet. Die Netzwerkgeräte werden in verschiedene Netze unterteilt, da zwischen dem OT-Netz und dem Office-Netz unterschiedliche Datenarten und Protokolle gesendet werden, um so optimale Datenübertragungsgeschwindigkeit bei höchstmöglicher Sicherheit zu gewährleisten.

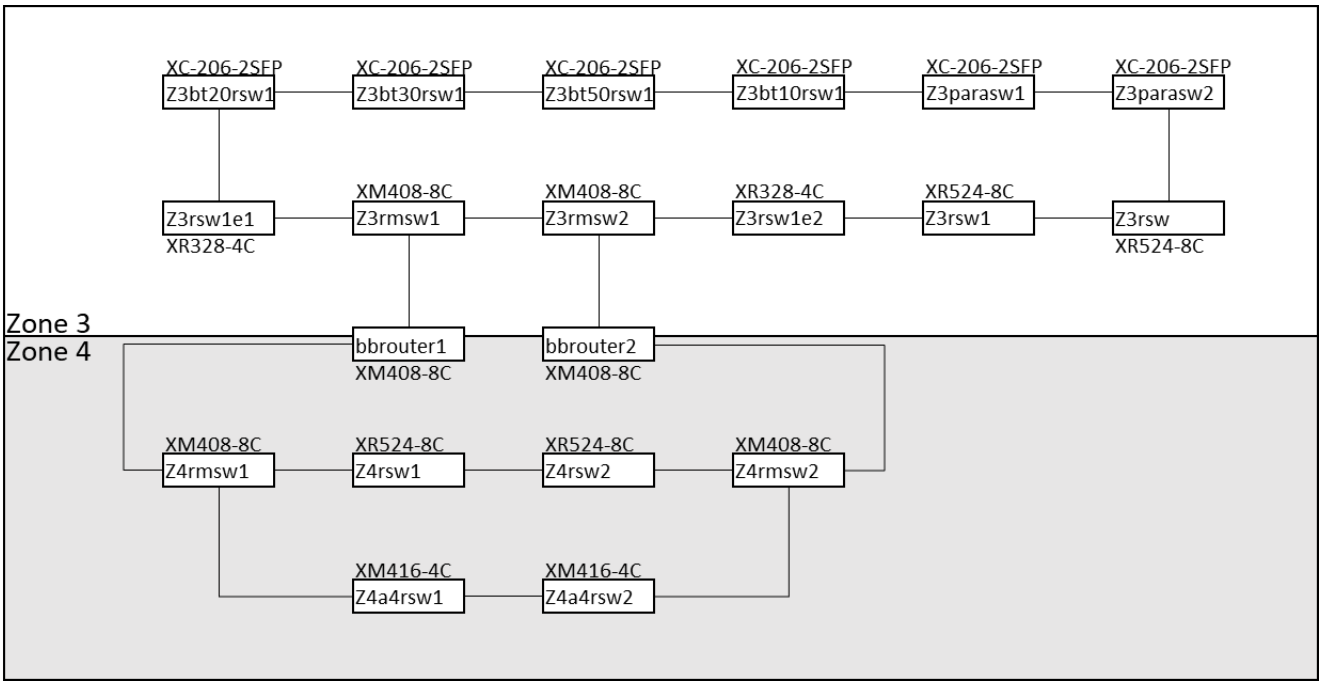


*Abbildung 1: Anordnung der Switches*

In Abbildung 1 ist das neue Zonenkonzept dargestellt. Es zeigt sowohl die Verteilung der Zonen als auch die Anordnung der Switches als Ringstruktur.

In Zone 3 ist das Office-Netz lokalisiert. Das Office-Netz involviert alle Geräte, die für den Bürogebrauch verwendet werden. Das beinhaltet beispielsweise Rechner für den E-Mail-Verkehr nach außen, Computer, die mit dem Internet verbunden sind oder auch Computer, auf denen spezielle Büroprogramme genutzt werden, zum Beispiel SQL-Server. Hier werden Protokolle, wie HTTP, FTP und POP3, verwendet. Das IT-Netz

ist vom Aspekt der Sicherheit kritischer zu betrachten und durch Software und Firewalls zu schützen, da es durch die Anbindung zum Internet wesentlich einfacher möglich ist von der Ferne her auf das Netz zuzugreifen und die Geräte zu manipulieren.

Das Automatisierungsnetz, also alle Komponenten, welche für die eigentliche Produktion benötigt werden, wie zum Beispiel die SPSen, die Messungen und das SCADA-System befinden sich in Zone 4. Typische Protokolle sind hier PROFIBUS DP, PROFIBUS PA, PROFINET, HART und Ethernet POWERLINK. Dieses Netz ist unabhängig von dem IT-Netz und kommuniziert ausschließlich innerhalb der eigenen Strukturen. Dadurch entsteht vor allem im Bereich der Netzsicherheit ein wesentlicher Vorteil. Da die Geräte nicht über das Internet nach außen kommunizieren sind sie kaum anfällig für Angriffe von Schadsoftware. Die einzige reale Möglichkeit im OT-Netz Schadsoftware zu implizieren besteht darin, vor Ort die kommunizierenden Geräte zu manipulieren. Dies wird bestmöglich durch betriebsinterne Richtlinien vermieden.

Diese Netzwerke bestehen aus mehreren Switches, die innerhalb der Zone ringförmig aufgebaut sind. Der Vorteil an dieser Struktur besteht nach /BRA07 S.34/ darin, dass sowohl die Übertragungsbandbreite garantiert werden kann als auch eine hohe Ausfall- und Übertragungssicherheit gegeben sind. Nachteile der Ringstruktur sind neben der Komplexität der Verbindungen, die hohen Installationskosten. Dadurch entstanden 2 Ringstrukturen, die durch die sogenannten Backbone-Router miteinander verknüpft sind. Weiterhin sind die meisten Switches innerhalb dieser Ringstruktur redundant ausgelegt, um eine stete Kommunikation auch im Fehlerfall gewährleisten zu können.

Vor diesem Umbau war das Netzwerk dieser Anlage nicht in verschiedene Zonen unterteilt. Speziell unter dem Aspekt der IT-Sicherheit hatte dieses Konzept einige Schwächen. Der Lagerbestand wird von SQL-Servern dokumentiert. SQL-Server sind Server, auf denen Datenbanken installiert sind, die unter anderem Wareneingänge und Warenausgänge buchen und den Warenbestand erfassen. SQL-Server werden nicht nur innerhalb des Lagersystems verwendet. Sie müssen die vorhandenen Daten auch zu anderen Rechnern außerhalb des Netzwerks weiterleiten, um zum Beispiel den Rechnungsverkehr anzustoßen oder Warenbestände zu bilanzieren.

Aus diesem Grund wurde das IT-System auf das oben genannte Zonenkonzept umgebaut. Der Datenaustausch zwischen den IT-Komponenten und den OT-Komponenten wurde voneinander separiert, damit eine maximale Effizienz der technischen Geräte gewährleistet werden kann. Durch diesen Umbau war eine Neubetrachtung der Sicherheit notwendig. Das gesamte System musste sowohl hochverfügbar als auch nah an der Produktion sein. Die Hochverfügbarkeit wird durch die Ringredundanz gewährleistet. Sollte einer der Switches ausfallen, bspw. z4rsw1,

siehe Abbildung 1, ist die Kommunikation zwischen den anderen Switches trotzdem noch gegeben, da der Ring zwar aufgelöst wird, aber eine funktionierende Kette entsteht. Weiterhin würde durch die Redundanz der einzelnen Switches in diesem Fall der Switch z4rsw2 einspringen und die Verteilungsarbeiten von z4rsw1 übernehmen. Folglich würde sich bei einem Ausfall von z4rsw1 oberflächlich betrachtet nichts ändern und die Anlage würde weiterhin problemlos funktionieren. In dieser Tatsache liegt das Problem. Ein Ausfall dieses Switches würde nicht auffallen. Trotzdem wäre sowohl die Ringstruktur als auch die Redundanz außer Kraft gesetzt und bei einem weiteren Ausfall von bspw. Z4rsw2 würde es unverzüglich zum Anlagenstop kommen, da die IT-Komponenten nicht mehr miteinander kommunizieren könnten. Aus diesem Grund ist es von immenser Bedeutung die Verfügbarkeit der Switches kontinuierlich zu überwachen. Die Nähe der Produktion ist durch die Backbone-Router gegeben. Backbone-Router sind das Bindeglied zwischen der IT und der OT und übermitteln zum Beispiel Lagerdaten zu den SQL-Servern

## 1.2 Diagnosemöglichkeiten an den Netzwerkkomponenten

Die Switches und Router der Zone 3 und 4 sind zu überwachen. Hierbei handelt es sich um folgende Switches der Firma Siemens:

- Scalance XC206-2SFP
- Scalance XR328-4C
- Scalance XM408-8C
- Scalance XM416-4C
- Scalance XR524-8C

Diese Geräte sind gemanagte Switches. Managebare Geräte bieten einen Zugriff für erweiterte Funktionen, wie beispielsweise Filterung von Media Access Control- (MAC-) Adressen, Routing und Bandbreitenkonfiguration. Der Zugriff auf diese Geräte erfolgt hier über eine Weboberfläche. Die verschiedenen Scalance Switches der Firma Siemens sind allesamt Switches, die für den Bereich industrielle Kommunikation optimiert sind. In Hinblick auf Anschlüssmöglichkeiten, Bauform, Leistung und Zusatzausstattung wird zwischen den Serien Scalance X-200, X-300, X-400 und X-500 unterschieden. Die Variationen in der Endbezeichnung stellen hierbei lediglich physische Anschlusskriterien dar.

### 1.2.1 OSI-Modell

Um die verschiedenen Funktionen der Switches näher darlegen zu können, soll an dieser Stelle kurz auf das Open Systems Interconnection (OSI-) Modell eingegangen werden.

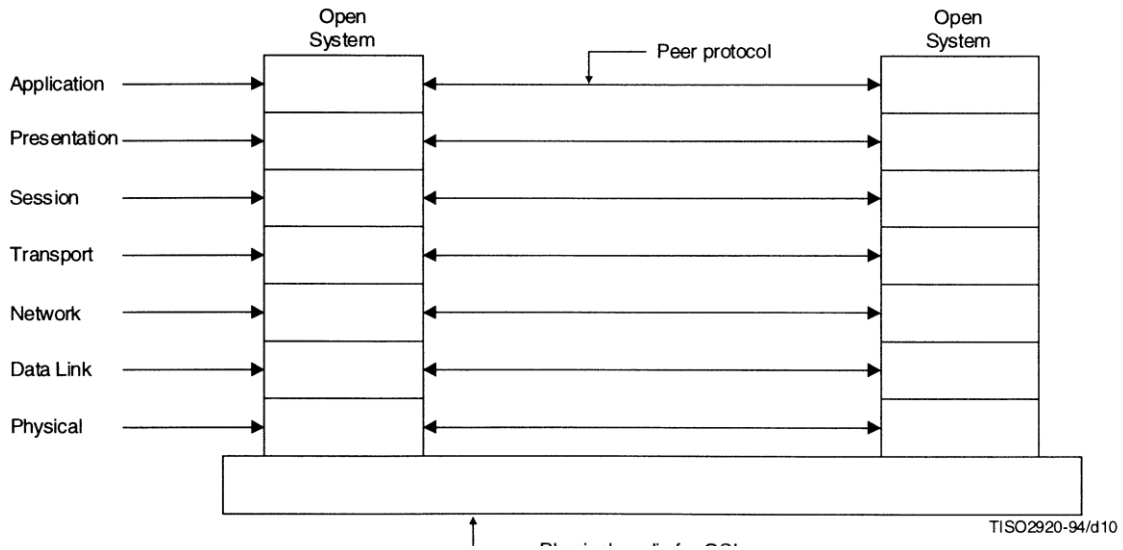


*Abbildung 2: OSI Referenzmodell /ISO96 S. 28/*

Das OSI-Modell definiert 7 verschieden Schichten, sogenannte Layer. Die 7 Schichten des OSI Modells sind in Abbildung 2 dargestellt. Durch dieses Modell wird die Kommunikation von Netzwerkkomponenten definiert und geordnet. Die Anwendung befindet sich in Layer 7, dem Application Layer. Sobald eine Anwendung mit einer anderen Komponente, zum Beispiel einem Programm, kommunizieren möchte, sendet es ein Datenpaket. Dieses Datenpaket wird in dem Presentation Layer, Layer 6, in ein unabhängiges Format, beispielsweise Unicode oder http übersetzt. Damit diese beiden Komponenten weiterhin miteinander kommunizieren können, wird in Layer 5, dem Session Layer, eine Sitzung aufgebaut und ständig synchronisiert. Dies ist vor allem hinsichtlich der Sicherheit und der Performance von Vorteil. Layer 5, 6 und 7 lassen sich unter dem Begriff Anwendungsschicht zusammenfassen. Die nachfolgenden 4 Schichten werden nur für die Übertragung der Daten benötigt. So werden die in Layer 7, 6 und 5 erstellten Nachrichten mit Hilfe von Layer 4 Komponenten über Ports verteilt. Geräte im Network Layer dienen anschließend dem Zweck des Routings. Daten werden IP-basiert weitergeleitet. Layer 1, der Physical Layer, stellt Mechanismen zu Verfügung, mit denen die Übertragung von Bits realisiert werden kann. Ein einfaches Beispiel hierfür wäre der RJ45 Stecker, welcher für Netzwerkkabel verwendet wird. /ISO96 S. 49/ Layer 2 gewährleistet die sichere Verteilung der verschiedenen Gerätedaten aus Layer 1 (vgl. /ECK14 S. 94ff./).

### 1.2.2 Switches


*Abbildung 3: Scalance XC206-2SFP*

Der Scalance XC206-2SFP (Abbildung 3) gehört zur X-200er Serie. Die Scalance X-200er Serie sind klassische Switches. Die Switches arbeiten auf der Data Link Schicht, Layer 2. Sie verbinden Geräte und Rechner über Netzwerkkabel. Der XC-200 besitzt 24 Anschlussmöglichkeiten für RJ45 Stecker und zeichnet sich vor allem durch seine vergleichsweise hohe Datenrate von bis zu 1000 Mbit/s aus. /SIE20/

*Abbildung 4: Siemens Scalance XR328-4C*

In Abbildung 4 ist ein Scalance XR328-4C abgebildet, welcher zur X-300er Serie gehört. Die Scalance X-300er Serie sind ebenfalls typische Layer 2 Switches. Der Unterschied zu der X-200er Serie besteht, neben der Bauform, hauptsächlich darin, dass sie durch Medienmodule erweitert werden können. Diese Medienmodule funktionieren wie eine Art Adapter. So ist es möglich neben RJ45 Steckern auch beispielsweise Lichtwellenleiter zu verwenden. In die XR-300 Switches können bis zu 12 Medienmodule eingebaut werden. /SIE20/

*Abbildung 5: Siemens Scalance XM408-8C und Siemens Scalance XM416-4C*

Die verbauten Switches der X-400er Serie sind in Abbildung 5 dargestellt. Die Scalance X-400er Serie werden wegen ihrer höheren Leistungsfähigkeit als Router verwendet. Somit agieren diese Komponenten nicht wie gewöhnlich auf OSI Layer 2 in der Paketverteilung, sondern zusätzlich in dem Network Layer, also Layer 3, im Routing zwischen den Netzen. Sie sind vor allem für den Einsatz in hochverfügbaren Umgebungen konzipiert. Der XM408-8C und der XM416-8C unterscheiden sich lediglich in Anzahl und Art der möglichen Anschlüsse. Der XM408-8C hat 8 RJ45 Ports, 8 Small Form-factor Pluggable (SFP) Ports und 8 Combo Ports. Der XM416-8C besitzt 16 RJ45 Ports, 4 SFP Ports und 4 Combo Ports. /SIE201/

*Abbildung 6: Scalance XR524-8C*

Die Scalance X-500er Serie sind sogenannte Core Switches. Diese sind, wie die X-400er Switches, für Routingaufgaben entworfen. Als Core Switches sind sie außerdem für die Kommunikation außerhalb des eigenen Netzes, zum Beispiel ins Internet, konzipiert. Weiterhin erreichen die X-500er Switches Bandbreiten von bis zu 10 Gbit/s. Die verbauten Switches dieser Serie sind Scalance XR524-8C, wie in Abbildung 6. /SIE201/

### 1.2.3 Optionen für die Überwachung

Die Überwachung der Switches kann entweder über die Meldekontakte der Switches oder netzwerkseitig erfolgen. Bei der Analyse über Meldekontakt gibt es mehrere Fehlerarten. Für diese Überwachung wird zwischen den folgenden unterschieden:

- Es wurde ein Fehler erkannt
- Die Spannungsversorgung fiel unter 9,6 V
- Ein Port hat keinen Datendurchsatz

Die netzwerkseitige Überwachung kann über die SINEMA Server erfolgen. SINEMA Server ist ein Server von Siemens zum Netzwerk-Monitoring. Für jede Zone existiert ein separater SINEMA Server. Dadurch kann eine optimale Trennung der beiden Netze voneinander gewährleistet werden. Nach Installation des SINEMA Servers können alle Netzwerkkomponenten gescannt und wahlweise in die Liste der zu überwachenden Geräte involviert werden. Nachdem die Switches und Router in diese Liste eingepflegt

wurden, werden sie überwacht. Laut der Betriebsanweisung des SINEMA Servers kann der Server nun folgende Status der Geräte anzeigen:

- Not Connected - Gerät nicht verbunden
- Ok - Kein Fehler anstehend
- Fault - Fehler
- Maintenance demanded - Wartung angefordert
- Maintenance required - Wartung erforderlich
- Not reachable - Gerät nicht erreichbar
- Not Monitored - Gerät nicht überwacht

Die meisten dieser Variablen sind selbsterklärend. Lediglich „Maintenance demanded“ und „Maintenance required“ bedürfen einer kurzen Erklärung. Da die Switches „managebar“ sind, also die Einstellungen mit Hilfe eines Webbrowsers bearbeitet werden können, besteht die Möglichkeit Alarme zu definieren, bei denen der Switch einen dieser Fehler anzeigt. Diese Alarme beziehen sich auf die Leistung der Switches. Sollte der Leistungsabfall eines Switches auf standardmäßig -50 dB ansteigen, so meldet der Switch Wartungsbedarf an (engl. Maintenance required). Sollte der Leistungsabfall eines Switches auf standardmäßig -55dB erhöhen, so meldet der Switch „Wartung erforderlich“ (engl. Maintenance demanded). Diese beiden Werte können in einem Webbrowser verändert werden. /SIE19 S. 92f./

Vergleicht man die beiden Möglichkeiten der verdrahteten und der netzwerkseitigen Überwachung, kristallisiert sich der Kostenfaktor als entscheidend heraus. Bei der verdrahteten Überwachung entstehen vor allem Kosten durch die Verdrahtung selbst, den Arbeitsaufwand diese Verdrahtung zu verlegen, jedes Gerät einzeln in einer SPS anzulegen und die Fehlerüberwachung dort zu programmieren. Bei der netzwerkseitigen Überwachung durch den SINEMA Server ist es, wie oben bereits erwähnt, recht simpel und zeitsparend die Switches zu implementieren und anschließend zuverlässig zu überwachen.

## 1.3 Hardwarestruktur SPS

Die SPS ist eine S7-1500 CPU 1513-1 PN mit einer 70W Spannungsversorgung, einem 32x24VDC HF Digitaleingabemodul, einem 16x24VDC/0,5A HF Digitalausgabemodul und einem AI 4xU/I/RTD/TC Analogeingabemodul. Laut Siemens Datenblatt besteht die Möglichkeit, die S7-1500er CPU als Open Platform Communications (OPC) Unified Architecture (UA) Client zu verwenden (/SIE182 S. 134/).

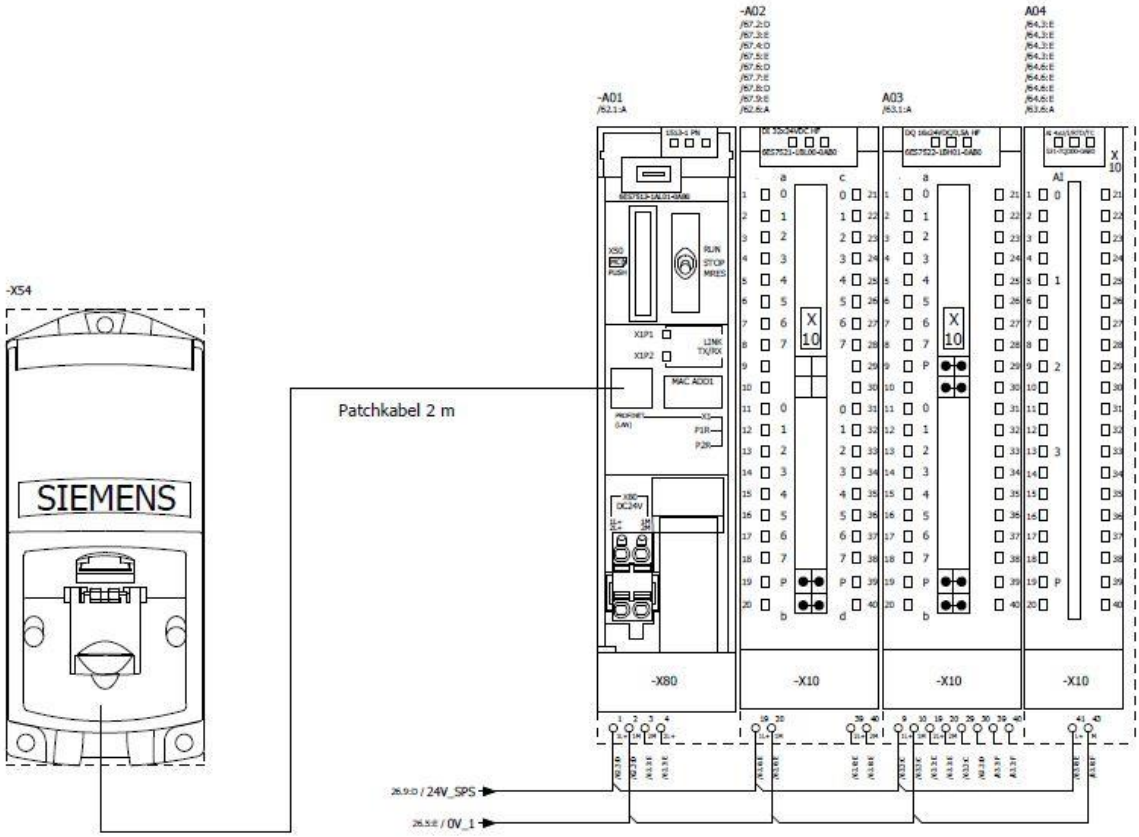


*Abbildung 7: Hardwarestruktur der SPS*

In Abbildung 7 ist die Hardwarestruktur der SPS abgebildet. Die Abbildung wurde dem Stromlaufplan entnommen.

## 1.4 Kommunikationsstandard OPC UA

Wie in den Unterkapiteln 1.2 Diagnosemöglichkeiten an den Netzwerkkomponenten und 1.3 Hardwarestruktur SPS bereits nahegelegt wurde, soll ein Großteil der Daten über den Kommunikationsstandard OPC UA erfolgen. Dieser soll in diesem Unterkapitel näher beleuchtet werden, wobei zuerst Augenmerk auf die historische Entwicklung von OPC UA gelegt wird und im Anschluss näher auf die Struktur, den Aufbau des Standards und seine einzelnen Spezifikationen eingegangen wird.

### 1.4.1 Entwicklung von OPC UA

Da der Standard OPC UA historisch gewachsen ist, ist es sinnvoll, etwas näher auf dessen Entstehung einzugehen, um dessen Strukturen besser nachvollziehen zu können. Bereits Anfang der 90er Jahre war Windows von Microsoft das meistgenutzte Betriebssystem auf Computern. In diesem Zeitraum etablierten sich auch die ersten supervisory control and data acquisition (SCADA) Programme. Dementsprechend wurden diese Systeme vor allem für Windows ausgelegt. Zu dieser Zeit benötigte jedes Programm noch einen separaten Treiber, um auf dem Computer funktionieren zu können. Diese Systeme waren bereits objektorientiert, also realitätsnah. Der Standard zur Ausführung dieser Systeme wurde als Object Linking and Embedding (OLE) bezeichnet.

Der Vorteil von OLE lag darin, dass die Verbindungen zwischen Dateien und Programmen dynamisch waren. Nach dem Beispiel von /LAN10 S. 307/ konnte in einer WORD Anwendung eine EXCEL Tabelle eingepflegt werden. Ändert man eine Zelle in der Originaltabelle, so ändert sich der Wert in dem WORD Dokument ebenfalls. Die Weiterentwicklung dieser Technologie führte dazu, dass der Anwendungsbereich der OLE Technik erweitert wurde. Neben der Nutzung in Schreibprogrammen konnte OLE dann auch für andere Anwendungen genutzt werden. Dieses verbesserte OLE wird seit 1995 als Component Object Model (COM) bezeichnet. Der COM Standard konnte lediglich die Kommunikation von Programmen auf einem Rechner managen. Deshalb wurde diese Technologie zur Distributed Compoent Object Model (DCOM) weiterentwickelt. Mit DCOM können Prozessdaten von einem Rechner auf andere direkt übertragen werden. Bis zu diesem Zeitpunkt benötigte jedes Programm einen eigenen Treiber. In Anbetracht der Programmierung und Implementierung dieser auf den PCs war das sehr ineffizient. Aus diesem Grund musste die Software für die rechnerübergreifende Kommunikation standardisiert werden. Deswegen wurde 1995 die OPC Task Force von verschieden Herstellern, wie z.B. Rockwell und Siemens AG, gegründet.

OPC war zu diesem Zeitpunkt ein Akronym für OLE for Process Control. Ein Jahr später verabschiedete die OPC Task Force die erste Spezifikation „OPC Specification Version 1“. Aus dieser Spezifikation entwickelte sich nach kurzer Zeit der OPC Data Access (DA) Standard. Mit diesem Standard war es möglich Prozessdaten von einem Windows basierten Gerät auf ein anderes, in einer für Prozessleitsysteme verträglichen Geschwindigkeit, zu lesen und zu schreiben. Bereits OPC DA basierte auf dem Client-Server-Prinzip. Im Laufe der Jahre wurde OPC DA stetig weiterentwickelt und den Bedürfnissen der Hersteller und Kunden immer weiter angepasst.

So wurde 1999 mit OPC Alarms and Events (AE) eine weitere Spezifikation veröffentlicht (vgl. /LAN10 S.5/). Mit OPC AE konnten ab diesem Zeitpunkt Alarmmeldungen und Ereignisse von OPC Servern gelesen und beschrieben werden. Die letzte Weiterentwicklung vom ursprünglichen OPC wird als OPC Historical Data Access (HDA) bezeichnet. Durch OPC HDA kann auf historische Daten zugegriffen werden. Anfang der 2000er Jahre kristallisierte sich das große Problem der bisherigen OPC Spezifikationen heraus. DCOM ist eine Microsoft Technologie und kann ausschließlich auf Windows Systemen effektiv genutzt werden. Manche Hersteller von Prozessleitsystemen nutzten damals jedoch bereits andere Betriebssysteme. So arbeitet Software von B&R beispielsweise ausschließlich auf UNIX Betriebssystemen. Weiterhin entwickelte sich die Nutzung von Feldgeräten und Steuerungen insofern weiter, dass diese mit Rechenleistung belegt wurden und entsprechend auch mit eigenen Betriebssystemen und Software-Standards bestückt wurden. Abhilfe für dieses System führte Microsoft im Jahr 2002 mit dem .NET Framwork ein (vgl. /LAN10 S.96/).

Durch das .NET Framework war es nun möglich Internetanwendungen zu nutzen und Kommunikation auch außerhalb der eigenen Rechnernetze zu realisieren. Webanwendungen arbeiten bis heute mit der Extensible Markup Language (XML). Durch diese standardisierte XML-Sprache wurde die wesentliche Einschränkung der DCOM Schnittstelle annulliert. Ab diesem Zeitpunkt war es möglich OPC auch auf anderen Betriebssystemen zu nutzen. Daraus entstand der OPC XML DA Standard. Die Nachteile von OPC XML DA lagen in der Geschwindigkeit der Datenübertragung und den Kosten. Durch das SOAP Protokoll, welches XML nutzt, wurde die Geschwindigkeit der Datenübertragungen um ein Vielfaches im Vergleich zu OPC DA verringert. Weiterhin mussten für OPC DA, OPC AE und OPC HDA jeweils ein separater Server erstellt werden, da die 3 Spezifikationen unterschiedlich und mit verschiedenen Protokollen arbeiteten. Dadurch existierte sowohl Programmier-, als auch Wartungsaufwand. Diese Einschränkungen wurden Mitte der 2000er Jahre bearbeitet und es entstand OPC UA.
OPC UA fasst OPC DA, OPC AE und OPC HDA unter einer Semantik zusammen und stellt diese sowohl über ein Protokoll mit Binärwerten (UA Binary) als auch als

Webservice zur Verfügung. OPC DA, AE, HDA und XML DA werden heute als OPC Classic bezeichnet (vgl. /LAN10 S. 6ff./).

Mit OPC UA war es gelungen einen herstellerübergreifenden Kommunikationsstandard zu schaffen, der die grundlegenden Ansprüche der Industrie bedienen kann. Aus diesem Grund änderte sich 2011 der Name von OLE for Process Automation in Open Platform Communications. Perspektivisch entwickelt sich OPC UA stetig weiter. Einer der wichtigsten Kritikpunkte an OPC UA besteht darin, dass dieser Kommunikationsstandard nicht echtzeitfähig ist. Deswegen arbeitete die OPC Foundation mit einigen Herstellern an einer Alternative dazu. Die Lösung wird als OPC UA over TSN (Time Sensitive Networking) bezeichnet. Entgegen des typischen OPC UA Client-Server-Modells wird für die Umsetzung von OPC UA over TSN das Publish-Subscribe- (PubSub-) Modell verwendet. Beim Client-Server-Modell muss ein Client zu jedem Server eine einzelne Verbindung aufbauen. Dadurch ist die Anzahl der Verbindungsmöglichkeiten durch Prozessoren und Speicherkapazitäten begrenzt. Der Publisher stellt, wie der Server, Daten bereit, die von den Subscribern gelesen werden können. Jedoch muss zwischen Publisher und Subscriber kein separater Verbindungskanal freigegeben werden, da der Publisher seine Daten für alle zugänglich an einen Router übermittelt. Der Router verteilt daraufhin die Informationen an die Subscriber. Dadurch werden Rechen- und Speicherleistung gespart. Weiterhin kann die Kommunikation schneller erfolgen, da sich die Subscriber nicht ständig neu mit den Publishern verbinden müssen und es kann eine sogenannte many-to-many Kommunikation stattfinden. Das bedeutet, dass viele verschiedene Publisher gleichzeitig Daten von verschiedenen Subscribern empfangen können (vgl. /OPC14 4/)

### 1.4.2 OPC UA Spezifikationen

In Kapitel 1.4.1 wurde bereits verdeutlicht, dass OPC UA auf seinen fest definierten Spezifikationen aufbaut. Diese Spezifikationen werden in diesem Kapitel erklärt. Zum Zeitpunkt der Erstellung der Arbeit gibt es 16 Kernspezifikationen von OPC UA:

- Part 1: Overview and Concepts
- Part 2: Security Model
- Part 3: Address Space Model
- Part 4: Services
- Part 5: Information Model
- Part 6: Mappings
- Part 7: Profiles
- Part 8: Data Access
- Part 9: Alarms und Conditions
- Part 10: Programs
- Part 11: Historical Access
- Part 12: Discovery and Global Services
- Part 13: Aggregates
- Part 14: PubSub
- Part 19: Dictionary Reference
- Part 100: Device Information Model

Part 1: Overview and Concepts gibt einen Überblick über die Struktur der Spezifikationen. Weiterhin beschreibt Part 1 das Client-Server-Modell und die Interfaces, die OPC UA verwendet.

Part 2: Security Model erklärt das Sicherheitsmodell von OPC UA. Das Security Model definiert sich hauptsächlich durch 2 große Sicherheitsaspekte. Ein Aspekt ist die Anmeldung beim Server mit Hilfe von X.509-Zertifikaten und der andere Sicherheitsaspekt ist das Aufbauen von Sessions zwischen Clients und Servern durch einen sicheren Kanal /LAN10 S. 114/.

Part 3: Address Space Model erklärt, wie der Adressraum eines OPC Servers aufgebaut ist. Um eine einheitliche Plattform zur Kommunikation zu realisieren werden die Variablen vereinheitlicht und der Zugriff auf diese klar definiert. Die einzelnen Elemente werden unter OPC UA als Objekte bezeichnet. Jedes Objekt besteht aus verschiedenen „Nodeclasses“. Die „Nodeclasses“ werden weiterhin in verschiedene „Nodes“ unterteilt. Die „Nodes“ sind Datensätze, die von definierten Eigenschaften „Attributes“ beschrieben werden. Die „Nodes“ werden über „References“ miteinander verbunden. Durch diese

References können die Nodes im Adressraum eines OPC UA-Servers mit Hilfe des Browsing gefunden werden. /OPC03 4/

Mögliche „NodeClasses“ sind:

- Variable
- Object
- Method
- View
- Datatype
- VariableType
- ObjectType
- ReferenceType

Part 4: Services beschreibt die Dienste, die auf einem OPC UA Server verwendet werden können. Diese Services sind in Gruppen von Service Sets unterteilt. Ein typischer Service des „Discovery Service Set“ ist zum Beispiel „GetEndpoints“. Dieser Service definiert die Sicherheitseinstellungen für die Endpunktanfrage des Clients und die Parameter, wie die Nachricht aufgebaut wird. /OPC04/

Part 5: Information Model stellt eine Erweiterung von Part 3: Address Space Model dar. In Part 5 wird beschrieben, wie die „NodeIDs“ von Variablen im Server exakt bezeichnet werden und wie diese über den „BrowseName“ gefunden werden können. /OPC05/

Part 6: Mappings ordnet die abstrakten Datensätze den realen Anwendungen zu. Hierbei wird vor allem auf 3 wichtige Aspekte Augenmerk gelegt: der Datencodierung, den Sicherheitsprotokollen und den Transportprotokollen /OPC06 4/. Die Datencodierung unter OPC UA umfasst 25 verschiedene Datentypen, wie zum Beispiel Boolean (0 oder 1) oder auch Int32 (ganzzahliger Wert zwischen -32768 und 32767) /OPC6 5.1.2/. Die Datencodierung kann über die folgenden Protokolle stattfinden:

- OPC UA Binary
    - für hohe Leistung und schnelle Codierung und Decodierung
- OPC UA XML
    - für Online-Anwendungen
- OPC UA JSON
    - für Cloudbasierte Kommunikation über PubSub mit Clients auf Basis von JavaScript

Part 7: Profiles beschreibt die möglichen Profile von Servern und Clients und wie diese implementiert werden. OPC UA bietet zahlreiche unterschiedliche Funktionen. Um diese Funktionen gegenseitig einfacher zertifizieren zu können, werden die Funktionen definierten „ConformanceUnits“ zugeordnet. Mehrere „ConformanceUnits“ bilden zusammen ein „Profile“. Mehrere Profile können jedoch auch eine gleiche „ConformanceUnit“ involvieren. Diese „Profiles“ werden wiederum in verschiedene Profilkategorien unterteilt:

- Client
- Global Directory Service
- Security
- Server
- Transport

/OPC07 4.4/

Jeder OPC UA-Server und jeder OPC UA-Client benutzt je nach Programmierung unterschiedliche „Profiles“. Die Profile, die in die Clients programmiert werden müssen von den Servern unterstützt werden, da die Profile sonst nicht in dieser Konstellation funktionieren und ein anderer OPC UA-Server benötigt wird.

Part 8: Data Access umschreibt die Verarbeitung von Daten für Prozessleitsysteme. Diese Prozessdaten können von E/A-Karten, die direkt mit dem Server verbunden sind, gelesen, beschrieben und überwacht werden, genauso, wie über Feldbussysteme zu SPSen oder anderen verarbeitenden Geräten geleitet werden. Die Verknüpfungen der Prozessdaten unter OPC UA heißen „DataItems“. In Part 8 ist auch definiert, wie OPC UA-Server mit OPC DA Systemen kommunizieren können, damit alte bereits bestehende Systeme weiter genutzt werden können. /OPC08 4/

Part 9: Alarms und Conditions definiert das Anlegen, Benennen und die Verarbeitung von Alarmen und Bedingungen. Bedingungen können quittierbar sein. Ist eine Bedingung quittierbar, wird sie als Alarm bezeichnet. Alarme können entweder aktiv oder inaktiv sein. Weiterhin können die Alarme Dialoge auslösen, wie bei einem Drucker, welcher anzeigt, dass er kein Papier mehr hat. /OPC09/

Part 10: Programs beschreibt die Erweiterung der in Part 3 und Part 4 bereits definierten Methoden. Ein Programm auf einem OPC UA Server ist eine sehr komplizierte Methode, die zum Beispiel Produktionskomplexe steuern kann, wie dem Ablaufschema eines Fließbandes. Part 10 definiert die Programmzustände (Stopp, Programm bereit, Programm läuft, Programm angehalten) und die Übergänge von einem Zustand in einen

anderen. Desweitern ist definiert, wie Clients laufende Programme überwachen und auf diese zu- und eingreifen können. /OPC10/

Part 11: Historical Access beschreibt den Umgang mit historischen Daten, also Daten, von einem vergangenen Zeitpunkt. Daten, die zum Beispiel in einer Datenbank gespeichert sind, können über einen OPC-Server dargestellt werden. Alle Daten, die vom Server bereitgestellt werden, sind mit einem „Timestamp“, also einem Zeitstempel, versehen, um die Historie eindeutig nachvollziehen zu können. Auf historische Daten kann mit dem „HistoryRead“ Dienst zugegriffen werden. /OPC11/

Part 12: Discovery and Global Services definiert, wie Clients Server in einem Netzwerkfinden können und auf diese zugreifen. In OPC UA-Servern ist im Normalfall ein Discovery Server installiert. Es gibt 3 Arten von Discovery Servern:

- LocalDiscoveryServer (LDS)
    - für alle Clients, die auf demselben Host kommunizieren,

- LocalDiscoveryServer mit MulticastExtension (LDS ME)
    - für alle Clients, die im selben Multicast Subnetz kommunizieren

- GlobalDiscoveryServer (GDS)
    - für alle Clients innerhalb einer Verwaltungsdomäne.

Wenn ein Client einen Server sucht, so findet er zuerst den Discovery Server. Der Client meldet sich bei diesem mit „FindServers“ an. Daraufhin gibt der Discovery Server die „DiscoveryUrls“, also die Urls der mit ihm verknüpften OPC UA-Server, an. Weiterhin definiert Part 12 die Zertifikatsverwaltung, die Key-Verwaltung und die Zugriffsrechte bei der Anmeldung des Clients beim Server. /OPC12 4/

Part 13: Aggregates beschreibt die Verfahren wie Daten zusammengefasst werden können. Mit OPC UA können Werte bereits verarbeitet weitergereicht werden. Beispiele für diese verarbeiteten Werte sind der Mittelwert, Minimum und Maximum eines Wertes aber auch Interpolationen von Wertereihen. Part 13 legt ebenso fest, dass die Herkunft dieser Daten unerheblich ist. Das heißt alle Daten, egal ob aus einem historischen Datenspeicher oder Echtzeitdaten, können so verarbeitet und anschließend im Serverraum angezeigt werden. /LAN10 S.209ff./

Part 14: PubSub stellt die Kommunikation über das Publish/Subscribe Model via OPC UA dar. Meist agiert ein OPC UA-Server als Publisher und ein OPC UA-Client als Subscriber. Diese Struktur ist aber nicht zwingend notwendig. /OPC14 4.4/

Part 19: Dictionary Reference definiert den Zugriff von OPC UA über andere „Wörterbücher“. Durch die Definitionen in Part 19, ist es möglich aus externen Quellen OPC UA Variablen zu erstellen, die genormt sind und dementsprechend einfach auch innerhalb der OPC UA Strukturen verwendet werden können. /OPC19 4/

Part 100: Device Information Model fasst die Inhalte der Spezifikationen zusammen. In Part 100 werden zum Beispiel die Grundlagen von OPC UA vermittelt. Desweiteren werden unter anderem die Client-Server-Struktur, die Schnittstellen, das Kommunikationsmodell, die Profile und die Namensräume erklärt. /OPC100/

# 2 Realisiertes Diagnosesystem

Im folgenden Kapitel werden die Entstehung und die Umsetzung des Projektes näher beschrieben, nachdem die vorhandenen Ressourcen in Kapitel 1 dargestellt wurden. Dafür wird zuerst die Überwachung der Infrastruktur näher beleuchtet. Das bedeutet, die Geräte und Konfigurationen der einzelnen verdrahteten Geräte, welche die optimale Arbeitsweise der Netzwerkkomponenten garantieren, werden untersucht und in das Gesamtkonzept eingebunden. Anschließend liegt das Augenmerk auf der OPC UA Kommunikation. Die OPC UA Kommunikation findet zwischen den Switches und Routern auf der einen Seite und den SINEMA Servern auf der anderen Seite statt. Daraufhin wird näher auf das WinCC eingegangen und die Möglichkeiten der Implementierung von Variablen in dieses. Im letzten Teilkapitel wird die Anbindung der Software an die SPS, das SCADA-System und die Kommunikation zwischen allen Komponenten dargelegt.

## 2.1 Überwachung Infrastruktur (Spannungsversorgung, Klimatisierung)

Die SPS wird mit dem TIA Portal V15.1 programmiert. Zu Beginn wird ein neues Projekt im TIA Portal angelegt. Dieses heißt WHS_Management_System. WHS steht für Warehouse (dt. Lager). Im Anschluss an die Projekterstellung wird die SPS als neues Gerät angelegt.

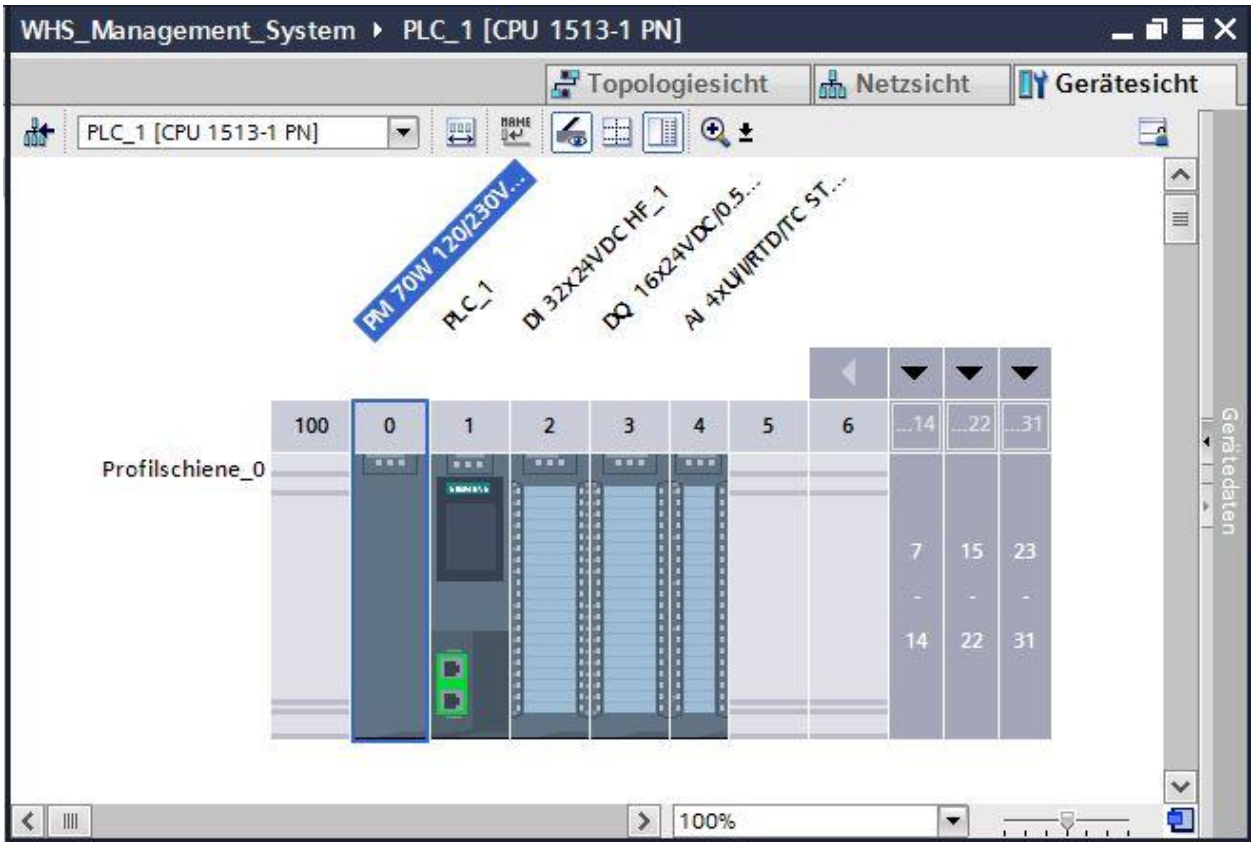


*Abbildung 8: Virtuelles Rack im TIA Portal*

Neben der SPS werden auch die Spannungsversorgung und die Ein- und Ausgabemodule zu dem virtuellen Rack hinzugefügt. Die komplette eingefügte Hardwarestruktur im TIA Portal ist in Abbildung 8 dargestellt. Nach Einfügen der CPU ist es möglich, die Programmierung dieser zu beginnen. Das TIA Portal legt automatisch den Organisationsbaustein OB1 an. OB1 ist der Hauptbaustein, welcher das eigentliche Programm abspielt. Das Abspielen des Programms von Anfang bis Ende nennt man 1 Zyklus. Nach Beendigung des Zyklus beginnt OB1 von vorn und das Programm läuft erneut ab. In OB1 ist es möglich entweder Funktionsbausteine (FBs) oder Funktionen (FCs) einzufügen. Der Unterschied zwischen FBs und FCs besteht darin, dass FBs Werte zwischenspeichern können und FCs lediglich Funktionen durchlaufen lassen. Bei dem hinzufügen eines FBs in ein Programm wird parallel dazu ein Instanz Datenbaustein (DB) erstellt. Instanz DBs gehören allgemein zur Gruppe der DBs. DBs sind Bausteine, in

welchen Daten vorhanden sind. Es wird zwischen Globalen DBs und Instanz DBs unterschieden. Globale Datenbausteine sind DBs, in welchen Werte händisch angelegt werden und jede Funktion, jeder FB und jeder Organisationsbaustein (OB) auf diesen Baustein lesend und schreibend zugreifen kann. Im Gegensatz dazu stehen die Instanz DBs. Diese werden zum Beispiel gebildet, sobald ein FB erstellt wird und beinhaltet, in dem Fall, die Daten, welche in dem FB gesichert werden sollen. Dementsprechend ist es nur für den FB möglich auf den entsprechenden Instanz DB lesend und schreibend zuzugreifen. Bei der Initialisierung eines FB oder eines FC ist es wichtig eine der möglichen Programmiersprachen zu wählen. Diese sind:

- Kontaktplan (KOP)
- Funktionsplan (FUP)
- Anweisungsliste (AWL)
- Structured Control Language (SCL)
- S7 Graph

Das vorliegende Projekt wurde mit FUP programmiert, da in dieser Programmiersprache bereits einige Bausteine vorhanden sind, die die Kommunikation über OPC UA realisieren können.

Die Überwachung der Fehlerkontakte der Klimaanlagen, der Sicherungen und der Phasen des Netzes, werden mit dem digitalen Eingangsmodul der SPS verdrahtet.

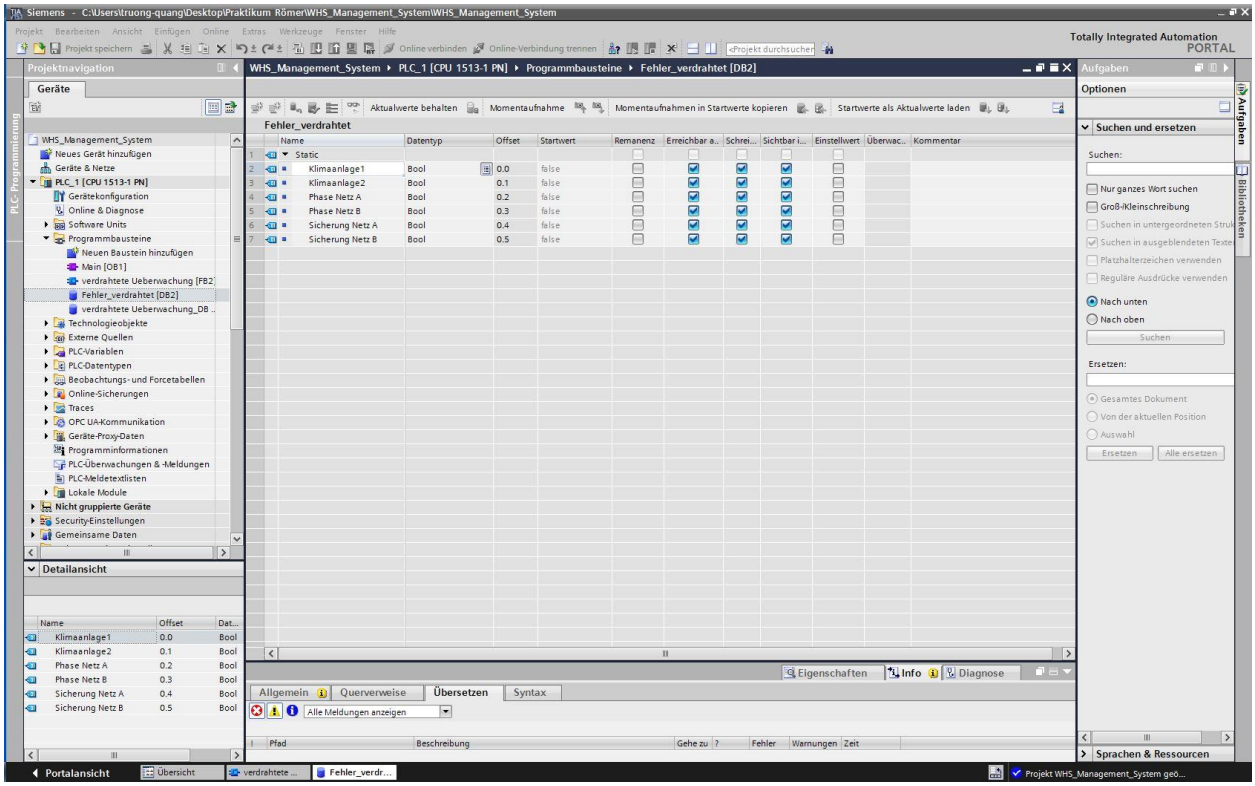


*Abbildung 9: Datenbaustein "Fehler_verdrahtet“*

Für die Weiterverarbeitung der Daten wurde ein Datenbaustein (DB) angelegt. Dieser Datenbaustein heißt „Fehler_verdrahtet“ und ist der DB2. Die Variablen sind in Abbildung 9 einsehbar. Die Variablen sind alle vom Typ Bool, also Binärwerte. Das bedeutet, die Variablen können entweder den Wert 0 oder den Wert 1 annehmen. Der DB2 ist ein globaler DB. Weiterhin wurde ein FB erstellt, der die Binärwerte erfasst und zu den Variablen in den dafür vorgesehenen DB, DB2, weiterleitet. Durch Erstellung des FB2 wurde automatisch der Instanz DB DB1 erstellt. DB1 dient als Datenpuffer und als Aufrufbaustein für FB2. Dieser FB wurde in OB1 eingefügt. OB1 hat in diesem Fall die Funktion das Programm von FB2 abzuspielen. Die Programmstruktur innerhalb FB2 ist sehr simpel gehalten. Jede Variable wird negiert einem Bit in einem Datenbaustein zugeordnet. Die Negation ist deswegen nötig, damit die Fehlersicherheit gewährleistet wird. Fehlersicherheit bedeutet in dem Fall, dass auch bei Spannungsausfall ein Fehler dargestellt wird, obwohl der Meldekontakt weiterhin geschlossen bleibt.

Selbstverständlich wäre es auch möglich gewesen die Eingänge der SPS ohne Programm zu entnehmen und im WinCC entsprechend zu verarbeiten. Das wäre jedoch entgegen der Aufgabenstellung. Diese besagt, dass die gesamte Überwachung soweit möglich über die SPS zu gestalten ist und so wenig wie möglich im bereits bestehenden SCADA System verändert werden soll.

## 2.2 OPC UA Kommunikation Netzwerkkomponenten

OPC UA Kommunikation kann über verschiedene bekannte Protokolle arbeiten. Da in dem Projekt keine Cloudanbindung oder andere Verbindungen über das Internet nötig sind, erfolgt die Kommunikation nur per TCP/IP über UA Binary. Eine signifikante Eigenschaft von TCP/IP ist, dass zu Beginn der Kommunikation immer zuerst der sogenannte Drei-Wege-Handshake (engl. Three-Way-Handshake) stattfindet. Diese Prozedur wird nach /EIG12, S.141/ in drei Schritten definiert.

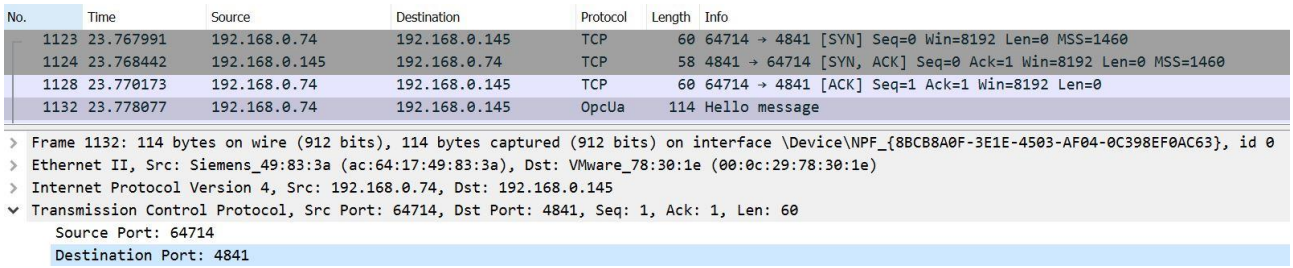


*Abbildung 10: Three-Way-Handshake*

Wie in Abbildung 10 ersichtlich, sendet der Client zu Beginn ein Paket mit den Verbindungsanforderungen an den Server. In diesem Paket existieren lediglich Steuerinformationen und keine Anwendungsdaten. Die wichtigsten Informationen davon sind das erste Bit, SYN, welches den Wert 1 hat und die Startsequenznummer SEQ, welche je nach Anzahl der Verbindungen unterschiedlich sein kann. Dieses Paket wird von dem Server im Anschluss mit einer Bestätigung beantwortet. Diese Bestätigung enthält neben dem SYN, welches wieder den Wert 1 hat, zusätzlich noch ein ACK Bit mit dem Wert 1 und der ACK-NR. Die ACK-NR hat den Wert der Startsequenznummer +1. Das bedeutet, dass er das Paket, welches zu der entsprechenden SEQ gehört, empfangen hat und ein Paket mit der nächsthöheren SEQ erwartet. Weiterhin enthält das Paket eine eigene SEQ vom Server. Als Antwort auf das Paket vom Server sendet der Client ein Paket, in welchem das SYN 0, ACK 1 und SEQ vom Server um 1 erhöht wurde. Damit ist der Drei-Wege-Handshake beendet und die Kommunikation kann über das OPC UA Protokoll fortgesetzt werden. Diesen Three-Way-Handshake und die anschließende OPC UA Kommunikation kann man mit Hilfe von Netzwerkanalysetools, wie zum Beispiel der Software Wireshark, aufzeichnen und analysieren.

Die eigentliche OPC UA Verbindung wird erst zu diesem Zeitpunkt aufgebaut und läuft nach folgendem Schema ab:

- Hello message: Client meldet sich beim Server an
- Acknowledge message: Server antwortet auf die Anmeldung
- OpenSecureChannelRequest: Client fragt an eine sichere Verbindung zu eröffnen
- OpenSecureChannelResponse: Server antwortet, dass die sichere Verbindung hergestellt wurde
- GetEndpointsrequest: Client erfragt die Endpunkte des Servers
- GetEndpointsresponse: Server teilt seine Endpunkte mit

Nun ist die Verbindung zwischen dem Client und dem Server aufgebaut und je nach Bedarf kann zum Beispiel der Server durchsucht werden, Daten können aus diesem gelesen oder beschrieben werden oder Methoden aufgerufen werden.

Mit diesem Hintergrundwissen ist es theoretisch möglich das Projekt wie angedacht zu realisieren. Da die bestehende Anlage sich nicht für solche Versuche eignet, ist es sinnvoll, eine Testumgebung aufzubauen, welche die grundlegend vorhandenen Strukturen der Realität gut nachstellt.

### 2.2.1 Testumgebung

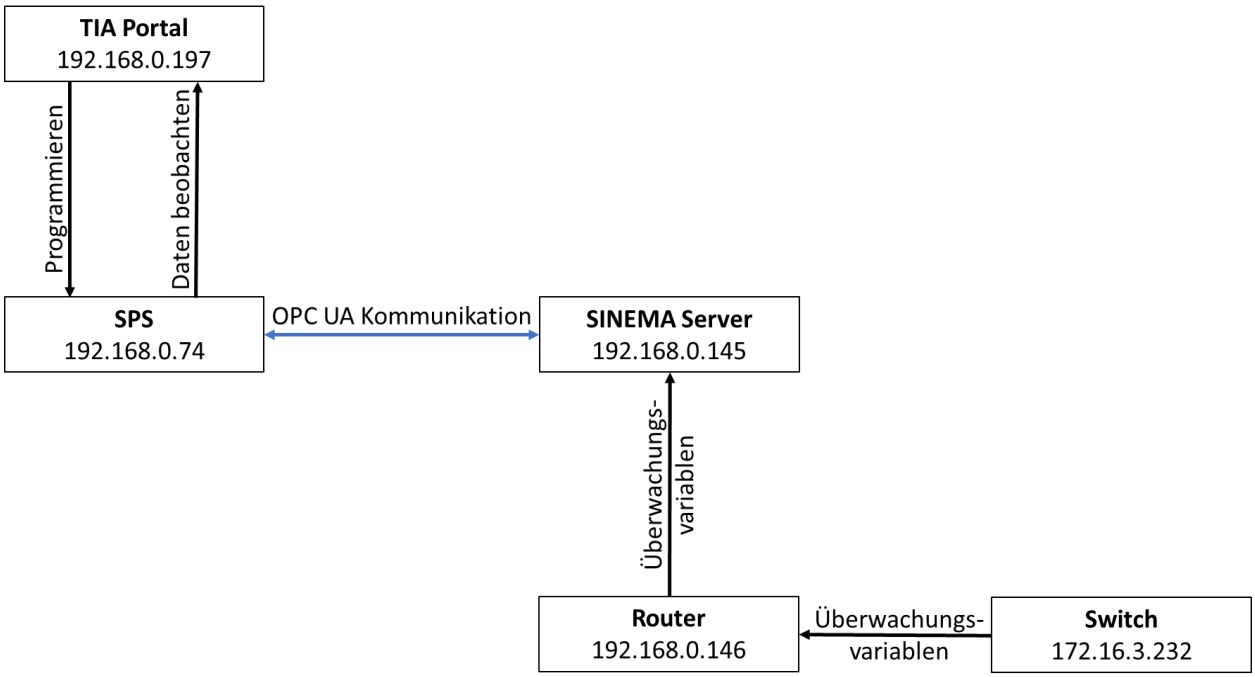


*Abbildung 11: Testumgebung OPC UA Kommunikation*

Die Testumgebung ist schematisch in Abbildung 11 dargestellt. Der Switch ist mit einem Router verbunden. Dieser Router ist auf Windows Server 2016 installiert. Der Router Kommuniziert mit dem Sinema Server, der auf einem weiteren Windows Server 2016 installiert. Das TIA Portal läuft unter dem Betriebssystem Windows 7. Alle Betriebssysteme sind virtuelle Maschinen (VMs), welche auf einem IOS System ausgeführt werden.

Der PC, auf dem das TIA Portal existent ist, dient als Programmiergerät (PG) der SPS. Weiterhin ist es möglich, die aktuellen Werte der SPS mit Hilfe des TIA Portals zu beobachten, um zum Beispiel Fehlercodes auszulesen. Die SPS soll mit dem SINEMA Server über OPC UA kommunizieren. Der SINEMA Server soll den Switch überwachen. Dieser Switch liegt in einem anderen Netzwerk, weswegen ein Router die Kommunikation zwischen dem 192.168.0.xxx und dem 172.16.3.xxx Netz herstellt und aufrechterhält. Der Kommunikationsaufbau zwischen der SPS als OPC UA-Client und dem SINEMA als OPC UA-Server schlägt fehl. Während die OPC UA Kommunikation von PC Software als Client tadellos funktioniert, treten Fehler beim Kommunikationsaufbau der SPS zum Server auf. Die Ursache dafür findet sich in dem Mitschnitt der Kommunikation mit Hilfe der Software Wireshark.

Der Aufbau der Kommunikation zwischen der SPS und dem SINEMA Server funktioniert zu Beginn. Das bedeutet der Three-Way-Handshake läuft problemlos ab, die SPS meldet sich als Client beim Server an und eine sichere Verbindung wird aufgebaut. Die

Endpunktanfrage der SPS wird auch beantwortet und die Sitzung wird beendet. Anschließend müsste ein erneuter Three-Way-Handshake erfolgen, welcher ausbleibt. Nach genauer Untersuchung der Kommunikation fällt auf, dass die SPS eine Endpunktanfrage mit der angegeben IP-Adresse sendet.

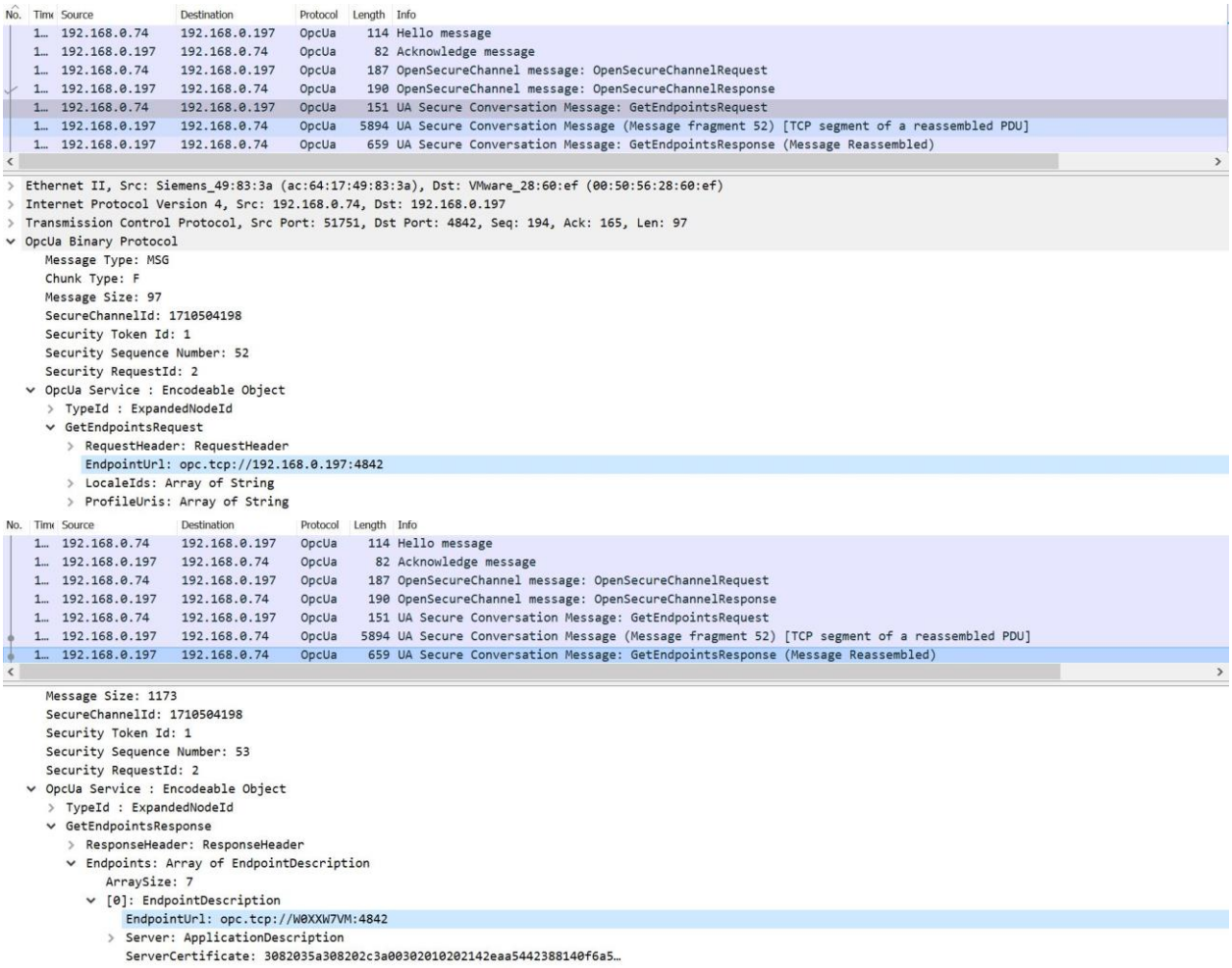


*Abbildung 12: Vergleich Endpointanfrage und Antwort*

Abbildung 12 verdeutlicht, dass sich der antwortende Server jedoch mit Host Namen, anstatt der IP-Adresse zurückmeldet. Die SPS bekommt einen anderen Wert. Statt dem Format einer IP-Adresse xxx.xxx.xxx.xxx, wird ein Name zurückgegeben. Das führt dazu, dass die SPS die Antwort zur Endpunktanfrage als falsch betrachtet, die weitere Kommunikation abbricht und beim Verbindungsaufbau den Fehlercode 8005 $0000_h$ ausgibt. Der Fehlercode 8005 $0000_h$ steht für „BadCommunicationError“, also einen allgemeinen Fehler in der Kommunikation.

### 2.2.2 Verwendung DNS-Server

Die Lösung für dieses Problem liegt in einem sogenannten Workaround, also einer Behelfslösung. Für die Auflösung des Namens muss ein DNS-Server installiert werden. Normalerweise sind in den Einstellungen dieses DNS-Servers, den DNS-Records, folgende Daten eingetragen:

- Domainname
- Forward Lookup Zone
- Reverse Lookup Zone

Der Domainname fasst die in den entsprechenden Zonen getätigten Zuweisungen unter einer Domain zusammen. So z.B. funktioniert der Datenverkehr bezüglich der Hochschule Merseburg unter der Domain hs-merseburg.de. Die Forward Lookup Zone weist einer eingegebenen IP-Adresse einen Namen zu, der an dieser Stelle definiert wird. Somit wird bei der Abfrage der Name in eine IP-Adresse aufgelöst. Die Reverse Lookup Zone ordnet dem Namen eine IP zu, wodurch bei der Abfrage der IP der Name ausgegeben wird (vgl. /ECK14, S.132/). So wird zum Beispiel bei Eingabe der Adresse www.hs-merseburg.de in einen Internet Browser von diesem zu der IP-Adresse 149.205.4.30 weitergeleitet und die Website der Hochschule öffnet sich. Damit können Geräte innerhalb eines Netzwerkes untereinander also sowohl über die IP-Adresse als auch über die Host Namen der Geräte miteinander kommunizieren. Das Problem besteht lediglich darin, das IP-Anfragen zwar umgewandelt, aber nur mit der Domain im Anhang des Namens wiedergegeben werden. Das nennt man den Fully Qualified Domain Name (FQDN).

Im Falle der Verbindung der SPS zum Server via OPC UA bedeutet das, dass die SPS in der Testumgebung nach dem PC-Namen fragt, nachdem die manuellen Einstellungen vorgenommen wurden. In diesem Fall nach „WIN-2MMG7BQAGEO“. Durch die Auflösung des DNS-Server kann diese Namensanfrage zu der IP-Adresse des SINEMA Servers weitergeleitet werden. Doch bei korrekten DNS-Records tritt im Anschluss das Problem auf, dass die Antwort des SINEMA Servers vom DNS-Server auch umgewandelt wird und nun den FQDN wiedergibt, also „win-2mmg7bqageo.automation.local“. Dieser Fehler wird auch in der OPC UA Spezifikation 12 im Anhang „C.3 DiscoveryUrlMapping“ beschrieben:

„*If the hostname is an* IPAddress *then it shall be converted to a domain name.If this cannot be done then LDS shall report an error.*“/OPC12 C3/

Auf Deutsch übersetzt bedeutet dieses Zitat, dass der Hostname einem Domainnamen zugeordnet werden muss, falls er nur aus einer IP-Adresse besteht, da der LDS des OPC UA-Servers einen Fehler anzeigt.
Dieses Problem geht auf die verschiedenen Arbeitsweisen zwischen PC als IT-Komponente und SPS als OT-Komponente zurück. Während ein PC die komplexen Nachrichtenstrukturen, die durch einen DNS-Server entstehen, umwandeln und nutzen kann, ist die SPS in ihrem Kommunikationsstandard als reine OT-Komponente eingeschränkt. Die SPS kann nicht mit Domainnamen arbeiten, da es keine Namensauflösung im Bereich der Automatisierungseben gibt und die Geräte nur IP-basiert miteinander kommunizieren. An dieser Stelle greift der eingangs erwähnte Workaround.

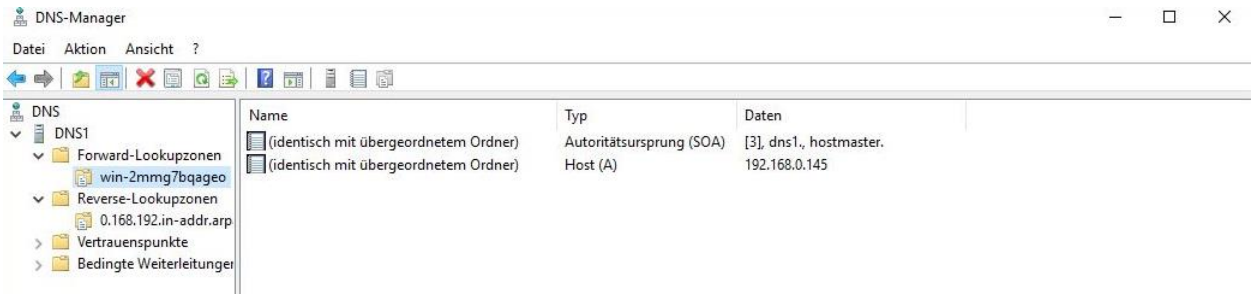


*Abbildung 13: DNS Records*

Wie in Abbildung 13 ersichtlich, kann dieses Problem damit umgangen werden, dass der Hostname manuell der Domainbezeichnung des Servers zugeordnet wird. Der Name, der anschließend dessen IP-Adresse zugeordnet wird, bleibt zeichenlos. Auf diese Weise wandelt der DNS-Server immer Anfragen auf die IP-Adresse des Servers und dessen Namen um und umgekehrt. Dadurch kann die SPS mit dem SINEMA Server über OPC UA kommunizieren, sogar, wenn die SPS die IP-Adresse als Endpunkt hat, so wie vorgesehen.

| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 1132 | 23.778077 | 192.168.0.74 | 192.168.0.145 | OpcUa | 114 | Hello message |
| 1134 | 23.778421 | 192.168.0.145 | 192.168.0.74 | OpcUa | 82 | Acknowledge message |
| 1135 | 23.782901 | 192.168.0.74 | 192.168.0.145 | OpcUa | 187 | OpenSecureChannel message: OpenSecureChannelRequest |
| 1136 | 23.783376 | 192.168.0.145 | 192.168.0.74 | OpcUa | 190 | OpenSecureChannel message: OpenSecureChannelResponse |
| 1137 | 23.790742 | 192.168.0.74 | 192.168.0.145 | OpcUa | 151 | UA Secure Conversation Message: GetEndpointsRequest |
| 1138 | 23.791901 | 192.168.0.145 | 192.168.0.74 | OpcUa | 8246 | [TCP Window Full] UA Secure Conversation Message (Message fragment 52) |
| 1144 | 23.796815 | 192.168.0.145 | 192.168.0.74 | OpcUa | 166 | UA Secure Conversation Message: GetEndpointsResponse (Message Reassembled) |
| 1146 | 23.803248 | 192.168.0.74 | 192.168.0.145 | OpcUa | 111 | CloseSecureChannel message: CloseSecureChannelRequest |
| 1165 | 24.008953 | 192.168.0.74 | 192.168.0.145 | OpcUa | 116 | Hello message |
| 1166 | 24.009287 | 192.168.0.145 | 192.168.0.74 | OpcUa | 82 | Acknowledge message |
| 1167 | 24.011506 | 192.168.0.74 | 192.168.0.145 | OpcUa | 187 | OpenSecureChannel message: OpenSecureChannelRequest |
| 1168 | 24.011693 | 192.168.0.145 | 192.168.0.74 | OpcUa | 190 | OpenSecureChannel message: OpenSecureChannelResponse |
| 1169 | 24.015604 | 192.168.0.74 | 192.168.0.145 | OpcUa | 368 | UA Secure Conversation Message: CreateSessionRequest |
| 1170 | 24.016082 | 192.168.0.145 | 192.168.0.74 | OpcUa | 8246 | [TCP Window Full] UA Secure Conversation Message (Message fragment 52) |
| 1176 | 24.019732 | 192.168.0.145 | 192.168.0.74 | OpcUa | 1125 | UA Secure Conversation Message: CreateSessionResponse (Message Reassembled) |
| 1177 | 24.026885 | 192.168.0.74 | 192.168.0.145 | OpcUa | 171 | UA Secure Conversation Message: ActivateSessionRequest |
| 1178 | 24.048321 | 192.168.0.145 | 192.168.0.74 | OpcUa | 150 | UA Secure Conversation Message: ActivateSessionResponse |
| 1245 | 25.752678 | 192.168.0.197 | 192.168.0.145 | OpcUa | 150 | UA Secure Conversation Message: ReadRequest |
| 1246 | 25.753112 | 192.168.0.145 | 192.168.0.197 | OpcUa | 120 | UA Secure Conversation Message: ReadResponse |

*Abbildung 14: Funktionierende OPC UA Verbindung*

Die weitere Kommunikation funktioniert, wie vorgesehen und könnte so umgesetzt werden. Dies ist in Abbildung 14 erkennbar, da nach dem Öffnen des sicheren Kanals und der erfolgreichen Endpunktanfrage ein neuer sicherer Kanal erstellt wird, in dem eine „Session“ aktiviert wurde und Daten aus dem Server herausgelesen werden können.
Der Nachteil dieses Workarounds liegt darin, dass in Automatisierungsanlagen nur sehr selten DNS-Server eingesetzt werden, da Automatisierungseinheiten ausschließlich auf IP-Basis kommunizieren. Sollte trotzdem ein DNS-Server bestehen, so ist dieser zwangsläufig mit Domain und Eintragungen in Forward- und Backup-Look-Zone einzustellen, da sonst eine erhebliche Sicherheitslücke im IT-System entstehen würde. Das bedeutet, dass die OPC UA Kommunikation zwischen der SPS und dem SINEMA Server grundlegend nicht möglich ist und andere Möglichkeiten das Netzwerk zu überwachen geprüft werden müssen.

## 2.3 SCADA-System mit Alarmmanagement

Ein SCADA-System ist ein System zur Steuerung und Überwachung von Prozessanlagen. Das vorhandene SCADA-System ist WinCC V7.4 der Firma Siemens. WinCC V7.4 kann sowohl als OPC UA-Server, als auch als OPC UA-Client fungieren. Um den OPC UA-Client nutzen zu können, wird der Treiber „OPC UA WinCC Channel“ benötigt. Sobald dieser in das Projekt migriert wurde, kann WinCC über OPC UA kommunizieren. Im „WinCC Configuration Studio“ ist es möglich, den Variablenhaushalt zu deklarieren. Innerhalb des Variablenhaushaltes ist es möglich den OPC UA WinCC Channel hinzuzufügen. Daraufhin kann sich der OPC UA-Client von WinCC mit einem OPC UA-Server verbinden. Dafür wird pro Server eine neue Verbindung angelegt, in welcher der Discovery Server und die Sicherheitseinstellungen der einzelnen Server angegeben sind. /SIE161 S.235/ Sobald WinCC mit einem OPC UA-Server verbunden ist, besteht die Möglichkeit, innerhalb des Servers zu browsen und einzelne Nodes in WinCC einzufügen. Dies geschieht mit Hilfe des „WinCC OPC UA Configurator“. Die Variablen der Serverraumüberwachung können bereits angelegt werden. Dies geschieht ebenfalls im „WinCC Configuration Studio“.

Nachdem die Variablen hinzugefügt wurden ist es nötig, Alarme zu erstellen, falls ein Switch ausfällt. Da bereits ein Meldesystem besteht müssen diese Alarme in das System integriert werden. Im „Alarm Logging“ in WinCC können Variablen aus dem Variablenhaushalt migriert werden. Diese Variablen werden mit einem Text versehen, welcher in der Alarmleiste bei einer Meldung angezeigt wird, zum Beispiel „Störung z3sa“. Weiterhin gehören Meldungen unter WinCC immer einer Meldeklasse und einer Meldeart an. In dem vorhandenen WinCC ist die Meldeklasse für die Netzwerküberwachung „Kommunikation“. Diese Meldeklasse wurde betriebsintern angelegt und involviert zum Beispiel Ausfälle von Frequenzumrichtern oder anderen Geräten. Die Meldeart wurde ebenfalls betriebsintern festgelegt. Hier wird zwischen den Kommunikationsplattformen unterschieden. Beispielsweise können die Daten von AS-I oder Profibus übertragen werden. Die Netzwerküberwachung bekommt dementsprechend die Meldeart Ethernet, da die OPC UA Kommunikation über OPC UA Binary stattfindet. Die Meldeart ist insofern wichtig, dass über diese die farbliche Darstellung der Fehlermeldungen eingestellt werden kann.

Sollte ein Netzwerkfehler auftreten, so ist zusätzlich eine Tonaufnahme abzuspielen. Dies geschieht ebenfalls durch das SCADA System, wo ein Script hinterlegt ist.

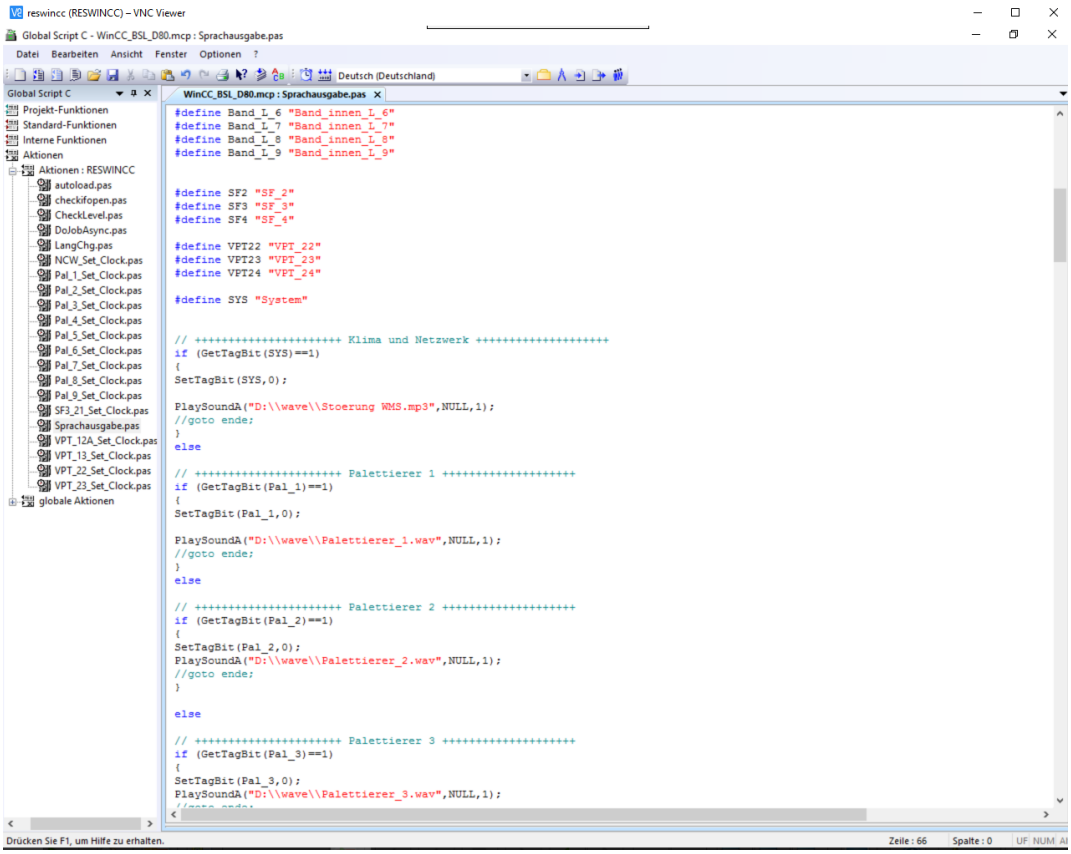


*Abbildung 15: Script Audio*

Das VB Script, also der niedergeschriebene Programmcode, ist in Abbildung 15 dargestellt. Das Script besagt, dass, wenn ein Fehler auf der Netzwerkseite (SYS) auftritt, die Audiodatei StoerungWMS.mp3 abgespielt werden soll.

## 2.4 Anbindung an SPS und Software

Das erstellte SPS Programm wurde in die bereits eingebaute SPS implementiert. Anschließend wurden die Daten aus dem DB in WinCC eingefügt und den dort vorhandenen angelegten Variablen zugeordnet. Wie in Kapitel 2.2 deutlich wurde, ist es nicht möglich, die Überwachung der Netzwerkkomponenten über die SPS zu realisieren. Um die Kosten trotzdem möglichst gering zu halten werden die Statusinformationen der Switches weiterhin über OPC UA übertragen. Dies ist möglich, da WinCC auch über diesen Standard kommunizieren kann.

Die OPC UA-Server arbeiten derzeit noch ohne Verschlüsselung, ohne Benutzeranmeldung und ohne Zertifikatszuweisungen. Aus diesem Grund wurden diese Einstellungen alle auf „None“, also ohne Zuweisung, gesetzt.

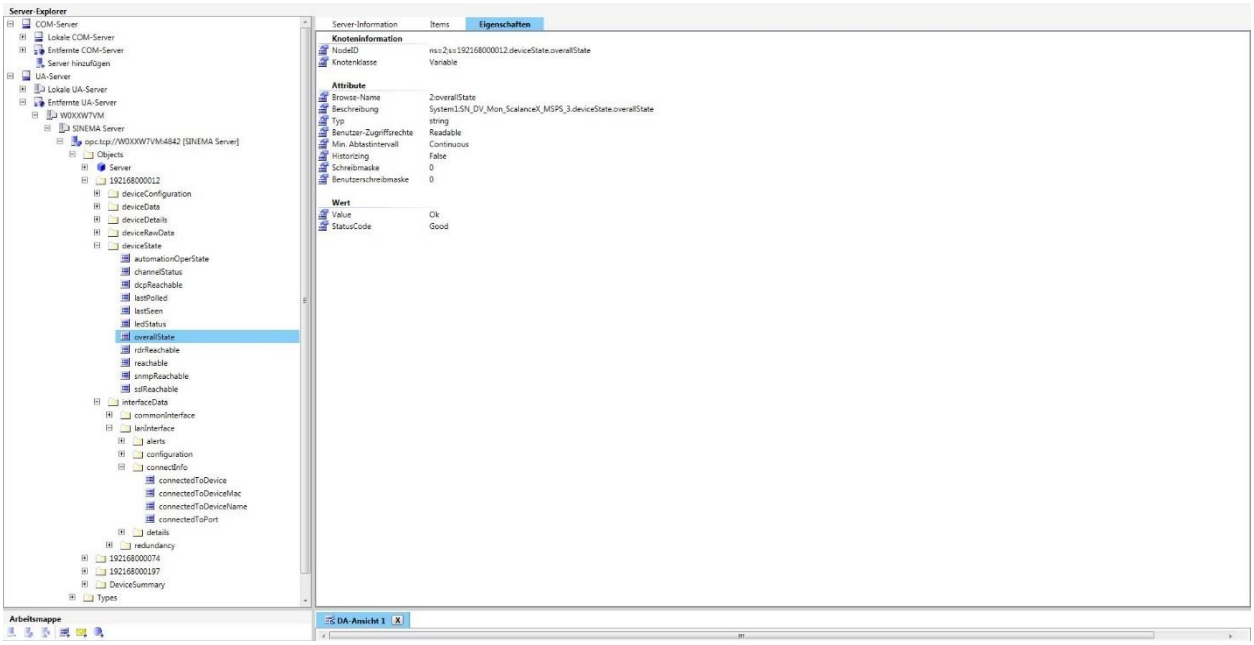


*Abbildung 16: Statusbit des Switches*

In Abbildung 16 ist die Struktur des zu überwachenden Nodes beispielhaft dargestellt. Das Object 192168000012 ist ein Scalance XC206-2SFP. Der Name des Objects setzt sich aus seiner IP-Adresse zusammen. Also hat dieser Switch die IP-Adresse 192.168.0.12. Wie in Kapitel 1.2 dargestellt ist der OverallState der Node, den es zu überwachen gilt. Diese „OverallState“ Nodes der verschiedenen Switches können so über den OPC UA Configurator in WinCC eingefügt werden. Weiterhin besteht die Möglichkeit, diese erstellten Variablen noch umzubenennen und kurze Beschreibungen hinzuzufügen. /SIEopc winCC S.221ff./ Um diese Variablen im Anschluss weiter verwenden zu können muss noch ein Script erstellt werden, da die benötigten Informationen nur als Strings ausgegeben werden.

Das Script ist für die Variablen immer gleich aufgebaut: Ein Fehler wird gemeldet, sobald der Switch den Wert „Fault“ oder den Wert „Not reachable“ anzeigt. Das Script ist, wie bereits das Script zum Abspielen der Tonaufnahme in Kapitel 2.3, mit VB geschrieben. Für jeden eingehenden Fehler wurde im Anschluss eine separate Alarmmeldung erstellt, aus welcher hervorgeht, welcher Switch ein Problem hat.

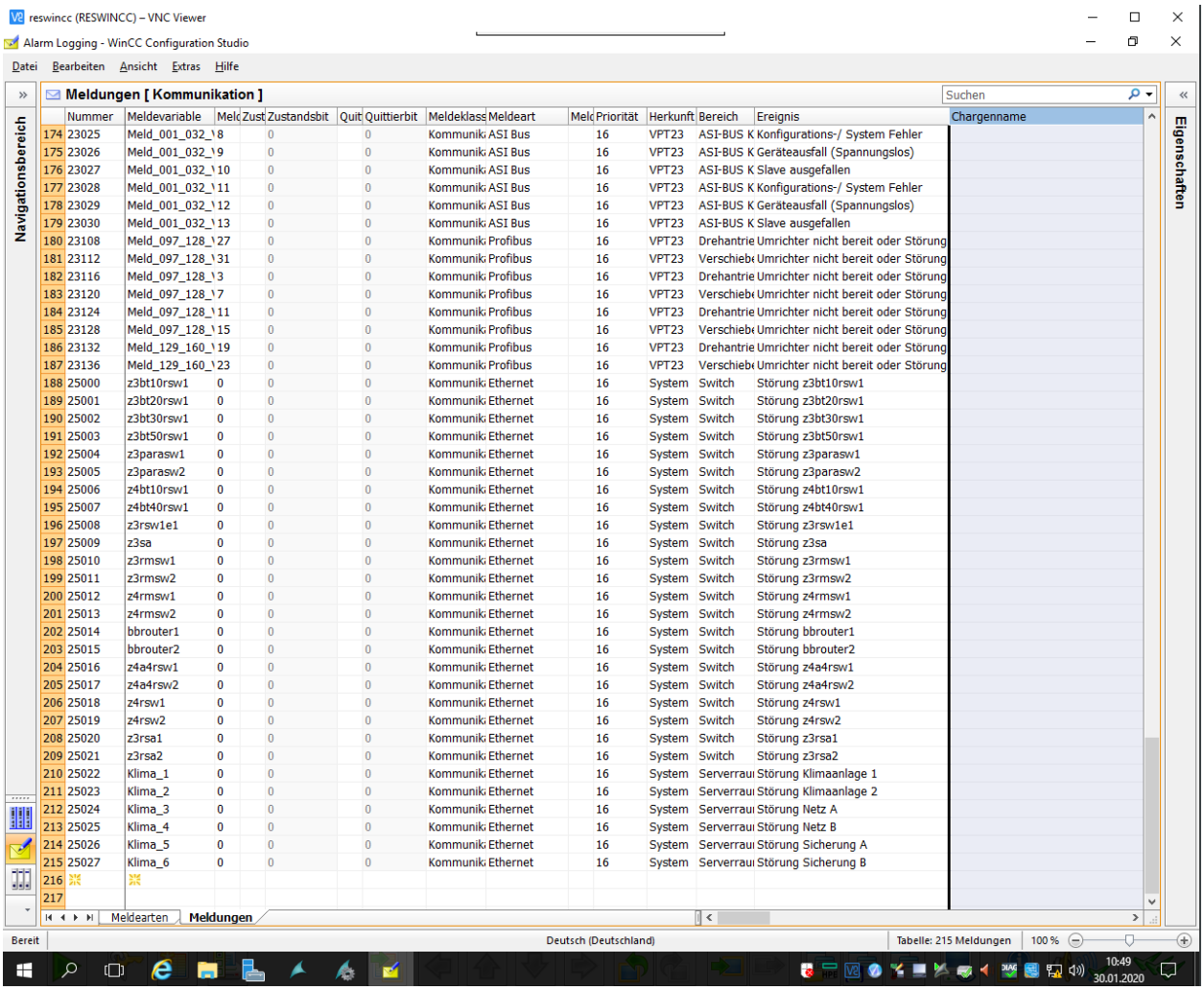


*Abbildung 17: Alarmmeldungen*

In Abbildung 17 finden sich die angelegten Alarmmeldungen mit der Herkunft System. Für jeden Switch und jeden Fehlerkontakt der Geräte des Serverraums wurde ein separater Alarm erstellt, damit aus der Alarmzeile leicht herauszulesen ist, welcher der Komponenten einen Fehler hat. Nachdem alle Variablen in WinCC integriert wurden, die Alarmmeldungen erstellt wurden und das Abspielen der Audiodatei mit den Alarmmeldungen verknüpft wurde, muss wegen der Übersichtlichkeit der Informationen noch ein Prozessbild erstellt werden.

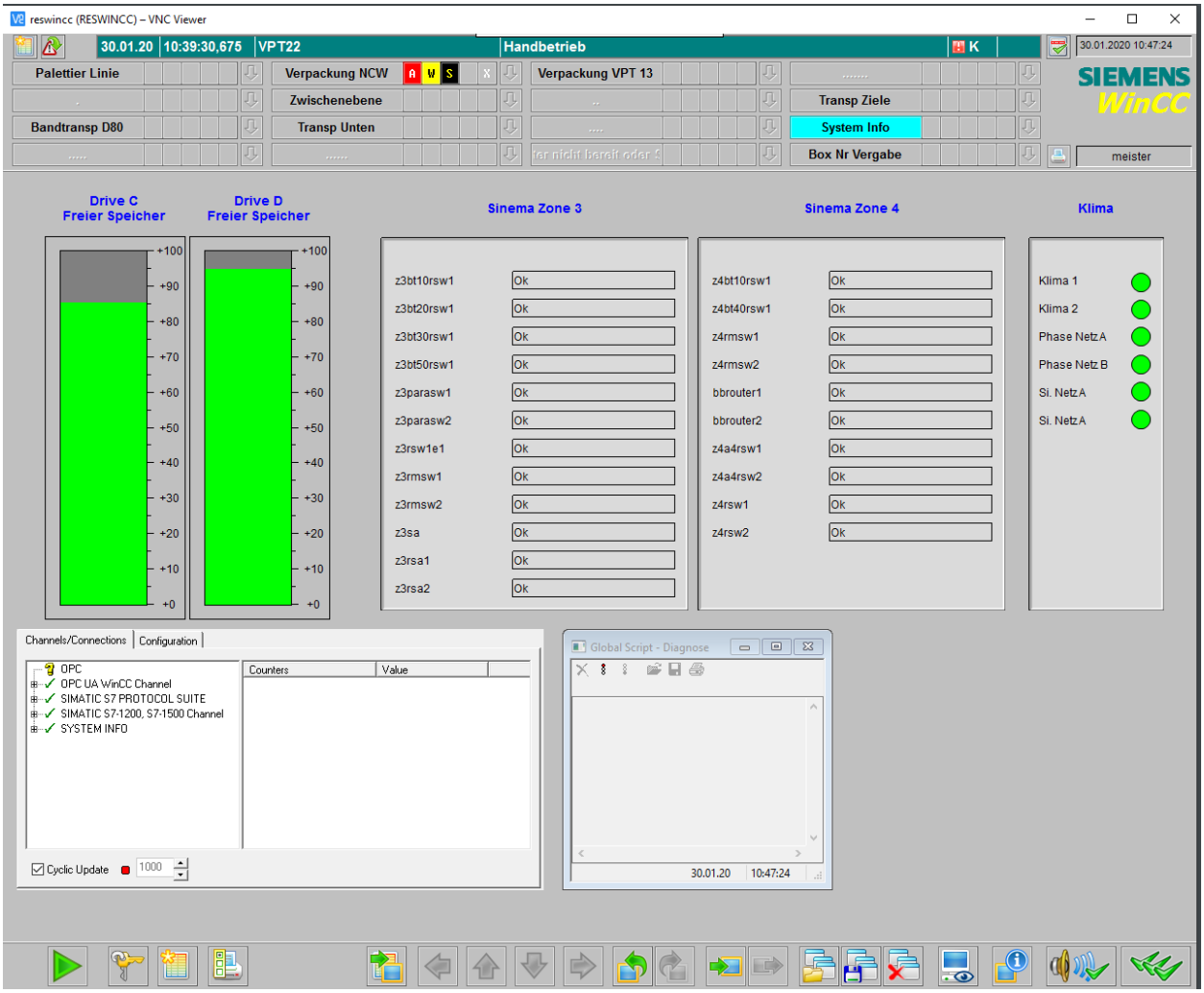


*Abbildung 18: Prozessbild*

Dieses Prozessbild ist in Abbildung 18 dargestellt. Es trägt den Titel System Info. Die eingehenden Statusinformationen der Switches werden, genauso wie sie über OPC UA vermittelt werden, als Strings dargestellt. Sie sind nach den Zonen sortiert, in denen die Switches liegen, also in Zone 3 und Zone 4. Neben den Informationen der Switches sind die Strom- und die Klimaanlagenüberwachung. Da diese nur binäre Informationen enthalten, bietet es sich an, farbige Buttons zu nutzen. Diese Buttons zeigen die Farbe Grün solange kein Fehler anliegt. Sollte eine Fehler auftreten, wechselt die Farbe des entsprechenden Buttons auf Rot. Weiterhin befindet sich in dem Prozessbild ein Fenster, welches die aktuellen Verbindungen anzeigt. Dieses Fenster dient dazu die OPC UA Verbindung zwischen WinCC und den beiden SINEMA Servern zu überwachen.

# 3 Ausblick und Fazit

Durch die 4. Industrielle Revolution verschwimmen die Grenzen zwischen der Automatisierungstechnik und der IT zunehmend. Dies sieht man am Beispiel des OPC UA Standard. Wie in dieser Arbeit hervorgegangen ist, birgt dieser Standard durch seine Vielseitigkeit und seine sehr vielfältigen Anwendungsgebiete ein hohes Potential, um der führende Standard in Europa in den nächsten Jahren zu werden und zu bleiben. Durch die stetige Weiterentwicklung, gerade im Bereich OPC UA over TSN, um mit OPC UA auch echtzeitfähig arbeiten zu können, wird dieser Standard für die Industrie zunehmend flexibler einsetzbar und damit auch attraktiver. Der Nachteil von OPC UA besteht jedoch darin, dass trotz der Standardisierung durch die Spezifikationen, manche Komponenten nicht miteinander kommunizieren können. Die Ursache hierfür findet sich in einer, der bereits bekannten, größten Herausforderungen der Umsetzung von Industrie 4.0, nämlich der zunehmend verschwimmenden Grenzen zwischen IT und Automatisierung. So ist es in Zukunft für den klassischen Automatisierer unerlässlich nicht nur IT-Grundlagenwissen, sondern auch fortgeschrittene Fähigkeiten in den Bereichen Netzwerke, Software und Anwendungsentwicklung zu beherrschen, um die vielfältigen Möglichkeiten, wie OPC UA Kommunikation, optimal nutzen zu können.

Das Ziel dieser Arbeit war an ein Projekt gekoppelt, ein Programm zur Überwachung der IT-Komponenten in einer bestehenden Anlage zu implementieren. Der Versuch, diese Überwachung ausschließlich über eine zentrale Überwachungs-SPS laufen zu lassen, scheiterte, da es derzeit praktisch nicht möglich ist, eine OPC UA Kommunikation zwischen dem vorhandenen SINEMA Server und der Siemens SPS als OPC UA Client herzustellen. Dementsprechend wurde die SPS nur für die Raumüberwachung, also die Klimaanlagen und die Spannungsversorgungen, eingesetzt. Die Variablen, welche vom SINEMA Server via OPC UA übertragen werden, wurden in das SCADA System implementiert und arbeiten dort auch problemlos, sobald die Verbindung zum OPC UA Server hergestellt wurde.

# Literaturverzeichnis


AUS19 Ausberger , T., & Štetina, M. (2019). General methodology for buildung of OPC UA gateways. IFAC (International Federation of Automation Control) (S. 317 - 322). Elsevier Ltd.

BRA07 Brandt, F., & Otten, L. (2007). NET IT Fachqualifikationen Netzwerktechnologien. Verlag Handwerk und Technik.

ECK14 Eckert, C. (2014). IT-Sicherheit : Konzepte -Verfahren - Protokolle. Oldenbourg Wissenschaftsverlag.

EIG12 Eigner , M., Gerhardt, F., Gilz, T., & Mogo Nem, F. (2012). Informationstechnologie für Ingenieure. Springer Vieweg.

ISO96 ISO/IEC 7498-1 : Second Edition : 994-11-15 : Corrected and reprinted : Information Technology - Open Systems Interconnection - Basic Reference Model: The Basic Model. (15. Juni 1996).

LAN10 Lange, J., Iwanitz , F., & Burke, T. (2010). OPC Von Data Access bis UNIFIED Architecture ., völlig neu bearbeitete und erweiterte Auflage. Berlin, Offenbach: VDE Verlag GmbH.

MEI12 Meinel, C., & Sack, H. (2012). Internetworking Technische Grundlagen und Anwendungen. Springer-Verlag.

OPC01 OPC Foundation. (22. November 2017). OPC UA Specification Part 1: Overview and Concepts. 1.04. Abgerufen am 15. März 2020 von https://reference.opcfoundation.org/v104/Core/docs/Part1/

OPC02 OPC Foundation. (03. August 2018). OPC UA Specification Part 2: Security. (1.04). Abgerufen am 02. März 2020 von https://reference.opcfoundation.org/v104/Core/docs/Part2/

OPC03 OPC Foundation. (22. November 2017). OPC UA Specification Part 3: Address Space Model. 1.04. Abgerufen am 12. März 2020 von https://reference.opcfoundation.org/v104/Core/docs/Part3/

OPC04 OPC Foundation. (22. November 2017). OPC UA Specification Part 4: Services. 1.04. Abgerufen am 29. Februar 2020 von https://reference.opcfoundation.org/v104/Core/docs/Part4/

OPC05 OPC Foundation. (22. November 2017). OPC UA Specification Part 5: Information Model. 1.04. Abgerufen am 10. März 2020 von https://reference.opcfoundation.org/v104/Core/docs/Part5/

OPC06 OPC Foundation. (22. November 2017). OPC UA Specification Part 6: Mappings. 1.04. Abgerufen am 29. Februar 2020 von https://reference.opcfoundation.org/v104/Core/docs/Part6/

OPC07 OPC Foundation. (22. November 2017). OPC UA Specification Part 7: Profiles. 1.04. Abgerufen am 08. März 2020 von https://reference.opcfoundation.org/v104/Core/docs/Part7/

OPC08 OPC Foundation. (22. November 2017). OPC UA Specification Part 8: Data Access. 1.04. Abgerufen am 16. März 2020 von https://reference.opcfoundation.org/v104/Core/docs/Part8/

OPC09 OPC Foundation. (22. November 2017). OPC UA Specification Part 9: Alarms and Conditions. 1.04. Abgerufen am 11. März 2020 von https://reference.opcfoundation.org/v104/Core/docs/Part9/

OPC10 OPC Foundation. (22. November 2017). OPC UA Specification Part 10: Programs. 1.04. Abgerufen am 12. März 2020 von https://reference.opcfoundation.org/v104/Core/docs/Part10/

OPC11 OPC Foundation. (09. Januar 2018). OPC UA Specification Part 11: Historical Access. 1.04. Abgerufen am 12. März 2020 von https://reference.opcfoundation.org/v104/Core/docs/Part11/

OPC12 OPC Foundation. (07. Februar 2018). OPC UA Specification Part 12: Discovery and Global Services. 1.04. Abgerufen am 16. März 2020 von https://reference.opcfoundation.org/v104/GDS/docs/

OPC13 OPC Foundation. (22. November 2017). OPC UA Specification Part 13: Aggregates. 1.04. Abgerufen am 19. März 2020 von https://reference.opcfoundation.org/v104/Core/docs/Part13/

OPC14 OPC Foundation. (06. Februar 2018). OPC UA Specification Part 14: PubSub. 1.04. Abgerufen am 02. März 2020 von https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-14-pubsub/

OPC19 OPC Foundation. (05. März 2020). OPC UA Specification Part 19: Dictionary Reference. 1.04. Abgerufen am 20. März 2020 von https://reference.opcfoundation.org/v104/Core/docs/Part19/

OPC100 OPC Foundation. (19. April 2019). OPC UA Specification Part 100: Device Information Model. 1.02. Abgerufen am 20. März 2020 von https://reference.opcfoundation.org/v104/DI/docs/

PLE19 Plenk, V. (2019). Angewandte Netzwerktechnik Kompakt Dateiformate, Übertragungsprotokolle und ihre Nutzung in Java-Applikationen. Springer Vieweg.

SIE161 Siemens AG. (Februar 2016). Siemens : SIMATIC HMI : WinCC V.4 : WinCC: Kommunikation : Systemhandbuch. A5E37536514-AA.

SIE162 Siemens AG. (Februar 2016). Siemens : SIMATIC HMI : WinCC V7.4 : WinCC: Arbeiten mit WinCC : Systemhandbuch. A5E37536340-AA.

SIE181 Siemens AG. (Dezember 2018). SIEMENS : SIMATIC NET : Industrial Ethernet Switches : Scalance XB-200/XC-200/XF-200BA/XP-200XR-300WG Web Based Management : Projektierungshandbuch. C79000-G8900-C360-08. Nürnberg.

SIE182 Siemens AG. (Oktober 2018). SIEMENS : SIMATIC:S7-1500, ET 200MP, ET 200SP, ET200AL, ET 200pro: Kommunikation : Funkrionshandbuch . A5E03735814-AG. Nürnberg.

SIE19 Siemens AG. (April 2019). Siemens : SIMATIC NET : Netzwerkmanagement : SINEMA Server : Betriebsanleitung. C79000-G8900-C241-10.

SIE20 Siemens AG. (12. März 2020). Siemens Produktkatalog Layer 2 Switches. Abgerufen am 16.03.2020 von https://mall.industry.siemens.com/mall/de/de/Catalog/Products/9300068?tree=CatalogTree#

SIE201 Siemens AG. (14. März 2020). Siemens Produktkatalog Layer 3 Switches / Routers. abgerufen am 18.03.2020 von https://mall.industry.siemens.com/mall/de/de/Catalog/Products/10167934?tree=CatalogTree#

VOG17 Vogel-Heuser, B., Bauernhansl, T., & Hompel, M. (2017). Handbuch Industrie 4.0 Bd. 2 Automatisierung 2. Auflage. Springer Vieweg.

WIL19 Wilmes, R. (2019). DI-Informationsmodell im OPC-UA-Standard. elektro automation, Heft PK1, S. 22 - 23.