



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorthesis

Tobias Schönrock

**Simulation von automatisierten Anlagen mittels
Interprozesskommunikation am Beispiel eines Aufzugsystems**

*Fakultät Technik und Informatik
Studiendepartment Informations-
und Elektrotechnik*

*Faculty of Engineering and Computer Science
Department of Electronic Emgineering and Com-
puter Science*

Tobias Schönrock

**Simulation von automatisierten Anlagen mittels
Interprozesskommunikation am Beispiel eines Aufzugsystems**

Bachelorthesis eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Engineering Elektro- und Informationstechnik
am Department Informations- und Elektrotechnik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr.-Ing. Ulfert Meiners
Zweitgutachter: Prof. Dr.-Ing. Jochen Maaß

Eingereicht am: 19.02.2013

Tobias Schönrock

Thema der Arbeit

Simulation von automatisierten Anlagen mittels Interprozesskommunikation am Beispiel eines Aufzugsystems

Stichworte

Lift, Aufzug, Fahrstuhl, LabVIEW, TIA-Portal, COM-Objekt, ActiveX, Interprozesskommunikation, OPC, TCP, NetToPLCsim, Simulation

Kurzzusammenfassung

Dieses Dokument beschreibt die verschiedenen Wege der Kommunikation zwischen der Software-SPS PLCsim von Siemens und Fremdsoftware am Beispiel von LabVIEW der Firma National Instruments. Durch diesen Datenaustausch soll es möglich sein, in einer Simulationssoftware wie LabVIEW die Hardware der Anlagen im Feldbereich zu simulieren und die Steuerung der Anlage mit der Software-SPS durchzuführen. Die verschiedenen Arten des Datenaustausches werden analysiert und bewertet. Die beste Methode wird am Beispiel eines Aufzugsystems angewendet.

Tobias Schönrock

Title of the paper

Simulation of automated Systems via inter-process communication at the example of an elevatorsystems

Keywords

Lift, Elevator, LabVIEW, TIA-Porta, COM-Object, ActiveX, inter-process communication, OPC, TCP, NetToPLCsim, simulation

Abstract

This document describes different ways of communication between the Software-programmable logic controler (PLC) PLCSim by Siemens and foreign Software with an Example in LabVIEW by National Instruments. It should be possible to simulate the field-hardware-devices of the System in LabVIEW and to implement the control of the System with the Software-PLC. The different ways to communicate with foreign Software is analyzed and rated. The best methode is applied at the example of an elevator system.

Inhaltsverzeichnis

1	Motivation und Ziele	1
1.1	Einführung	1
1.2	Aufgabenstellung	1
2	Theorie	3
2.1	Interprozesskommunikation	3
2.2	Software Tataly Integrated Automation (TIA)-Portal	3
2.2.1	Allgemeines zum Softwarepaket	3
2.2.2	PLCsim	4
2.3	LabVIEW	5
2.4	NefToPLCsim	6
2.5	Loopback Device	6
2.6	Einleitende Literaturhinweise	7
3	Analyse	8
3.1	Anforderungen	8
3.2	Lösungsansätze	11
3.2.1	ActiveX mit COM	11
3.2.2	OPC-Verbindung	12
3.2.3	TCP/IP-Verbindung	13
3.3	Analyse und Bewertung der Lösungsansätze	13
3.3.1	Bewertung von ActiveX mit COM-Objekt	13
3.3.2	Bewertung der OPC-Verbindung	17
3.3.3	Bewertung der TCP-Verbindung	21
3.3.4	Problematische Netzwerkprogrammierung in LabVIEW	26
3.4	Auswahl der besten Lösung	27
4	Simulationen in LabVIEW erstellen	29
4.1	Frontpanel: Bedienoberfläche mit Anlagensvisualisierung	29
4.2	Blockdiagramm: Grafische Schnittstellenprogrammierung	29
5	Beispiel einer Aufzugsimulation	30
5.1	Allgemeines zu Aufzügen	30
5.2	Fahrwegplanung	30
5.2.1	Handsteuerung	31
5.2.2	Sammelsteuerung	31

5.2.3	Druckknopfsteuerung	31
5.2.4	Zielauswahlsteuerung	31
5.3	Beschreibung der realen Anlage	32
5.3.1	Feuerwehraufzug	33
5.3.2	Sonstige Funktionen der realen Anlage	33
5.4	Funktionen in dieser Simulation	33
5.5	Programm in LabVIEW	35
5.5.1	Aufbau des Frontpanels	35
5.5.2	Beschreibung des Blockdiagrammes	36
5.6	SPS-Programm	38
5.6.1	Funktionsbeschreibung	38
5.6.2	Programmbeschreibung	39
6	Zusammenfassung und Ausblick	44
6.1	Ergebnisse	44
6.2	Ausblick	44
6.2.1	Mögliche Erweiterungen der Schnittstelle	44
6.2.2	Mögliche Erweiterungen der Aufzugsimulation	45
6.3	Programmieren im TIA-Portal - Erfahrungen mit der neuen Software	45
6.4	Erfahrungen mit LabVIEW	46

Tabellenverzeichnis

3.1	Relevante Datenbereiche der Steuerung	9
3.2	Anforderungen an die Simulation	11
3.3	Parameter des Testrechners	13
3.4	Zeitanalyse ActiveX-Verbindung	15
3.5	Zeitanalyse der OLE for Process Control (OPC)-Verbindung	19
3.6	Zeitanalyse der TCP/IP-Verbindung	23
3.7	Ergebnisse im Vergleich	27
3.8	Vergleich der Lösungsansätze	27
5.1	Aufbau des Auftragspeichers	40

Abbildungsverzeichnis

2.1	Ausschnitt aus der Startansicht des TIA-Portals	4
2.2	Programmansicht von PLCsim	5
2.3	Logo von LabVIEW	5
3.1	Logo von ActiveX	12
3.2	Logo von OPC	12
3.3	Geschwindigkeitstest der ActiveX-Verbindung	14
3.4	ActiveX-Übertragungsdauer abhängig von der Werteanzahl	16
3.5	Auszug aus dem ActiveX-Testprogramm	17
3.6	Geschwindigkeitstest der OPC-Verbindung	18
3.7	OPC-Übertragungsdauer abhängig von der Werteanzahl	20
3.8	Auszug aus dem OPC-Testprogramm	21
3.9	Geschwindigkeitstest der TCP/IP-Verbindung mit Loopback-Interface	22
3.10	Transmission Control Protocol (TCP)-Übertragungsdauer abhängig von der Werteanzahl	24
3.11	Ausschnitt aus Testprogramm TCP/IP: Lesetelegramme	25
3.12	Aufwändige Programmierung mit Netzwerktelegrammen	26
5.1	Frontpanel der Simulation in LabVIEW	35
5.2	Aufbau der Sensorik einer Etage	38
5.3	Programmstruktur mit Bausteinaufrufen	41

Abkürzungsverzeichnis

AWL Anweisungsliste

COM Component Object Model

CPU Central Processing Unit

DCOM Distributed Component Object Model

DLL Dynamik-Link-Librarie

DB Datenbaustein

HMI Human Machine Interface

IE Industrial Ethernet

IP Internet Protocol

LAN Local Area Network

OLE Object Linking and Embedding

OPC OLE for Process Control

SCADA Supervisory Control and Data Acquisition

SCL Structured Control Language (Siemensbezeichnung für "Strukturierter Text")

SPS Speicherprogrammierbare Steuerung

TCP Transmission Control Protocol

TIA Tataly Integrated Automation

1 Motivation und Ziele

1.1 Einführung

In der heutigen Zeit wird die Möglichkeit, Anlagen im Büro zu programmieren und zu simulieren, immer wichtiger. Auf einer Baustelle soll möglichst wenig Zeit verbracht werden, ein Programm soll daher vor Ort nur noch geladen und im Optimalfall fehlerfrei in Betrieb gesetzt werden. Zu diesem Zweck können realitätsgetreue Simulationen verwendet werden, welche besondere Eigenschaften einer Anlage widerspiegeln, die vom Programmierer in Plänen schnell übersehen werden. So entfallen häufige Fehler, die sonst erst bei der Inbetriebsetzung auftreten würden. Die vielfältigen Möglichkeiten von Anlagensimulationen können auch in der Ausbildung eine breite Anwendung finden. Echte Anlagen oder entsprechend verkleinerte Nachbauten sind für viele Lehrinrichtungen nicht finanzierbar und nehmen viel Platz in einem Übungsraum ein. Eine Simulation, die realitätsnah ist, kann gleich mehrere Übungsanlagen ersetzen.

Um die Simulation mit der Steuerung zu verbinden, wird eine Schnittstelle benötigt, die dafür sorgt, dass die simulierte Anlage mit ihren Sensoren und Aktoren mit dem Programm der Steuerung verbunden wird. Die Software in der speicherprogrammierbaren Steuerung (SPS) kann so getestet werden, als würde der Programmierer direkt an der realen Anlage arbeiten. In dieser Arbeit werden verschiedene Varianten einer Schnittstelle zwischen Simulation und Steuerung untersucht. Nach der Analyse wird die beste Methode auf ein Beispiel angewandt. Dafür wird eine Simulation einer bekannten Aufzuganlage erstellt und mit einer Software-SPS verbunden, um darin ein Programm zur Steuerung der Anlage ablaufen zu lassen.

1.2 Aufgabenstellung

Die Aufgabe dieser Arbeit besteht darin, die verschiedenen Möglichkeiten des Datenaustausches zwischen der Soft-SPS PLCsim in dem Paket des TIA-Portal¹ und einer fremden Software wie LabVIEW zu analysieren und zu bewerten. Dabei wird darauf geachtet, wie

¹Neue Softwarekomplettlösung zur Programmierung und Projektierung von Automatisierungskomponenten der Firma Siemens, nähere Informationen unter: [Siemens (2013)]

1 Motivation und Ziele

benutzerfreundlich die Schnittstellen sind und wie sie zu implementieren sind. Verschiedene Aspekte wie Geschwindigkeit und Erweiterbarkeit werden untersucht. Exemplarisch soll eine Anlagensimulation in LabVIEW entworfen und dessen Steuerung mit dem TIA-Portal realisiert werden. Dafür wird die Aufzuganlage der HAW Hamburg im Gebäude Berliner Tor 7 simuliert. Am Schluss der Arbeit ist das neue TIA-Portal rückblickend auf die Erfahrungen während der Programmierarbeit kurz zu bewerten und positive oder negative Erfahrungen zu berichten.

2 Theorie

2.1 Interprozesskommunikation

Interprozesskommunikation beschreibt die Kommunikation von Anwendungen und Prozessen, die nicht denselben Speicher eines Systems nutzen. Darin sind auch netzwerkübergreifende Kommunikationsanwendungen eingeschlossen. Es wurde mit der Entwicklung der Computeranwendungen immer bedeutender, dass einige Anwendungen untereinander Daten austauschen. Dafür wurde zunächst Object Linking and Embedding (OLE) entwickelt. Daraus entstanden weitere Schnittstellen. [Wikipedia (b)]

2.2 Software TIA-Portal

2.2.1 Allgemeines zum Softwarepaket

Das TIA-Portal wurde für die Projektierung und Programmierung von speicherprogrammierbaren Steuerungen entwickelt. Es stellt eine Komplettlösung dar, die von der speicherprogrammierbaren Steuerung (SPS) bis zur Prozessvisualisierung alle Werkzeuge enthält. Die Software ist die Weiterentwicklung des bereits verbreiteten Softwarepakets SIMATIC STEP 7, das erst durch weitere Programme (z.B. WinCC) fähig ist, eine Visualisierung oder Ähnliches zu erstellen. Der SIMATIC Manager ist die Verbindungskomponente der verschiedenen Einzelwerkzeuge. Das TIA-Portal kombiniert die Funktion des SIMATIC Managers und die der verschiedenen Programme in einem Fenster. Bei der Oberflächengestaltung wird scheinbar mehr Wert auf das Design gelegt und auf Veranschaulichung der komplexen Prozesse. Schon bei der Startansicht, die in Abb. 2.1 zu sehen ist, kann man erkennen, dass viel Wert auf Benutzerfreundlichkeit gelegt wird. In der Startansicht kann der Nutzer wählen, welche Aufgabe er gerade bearbeiten möchte.

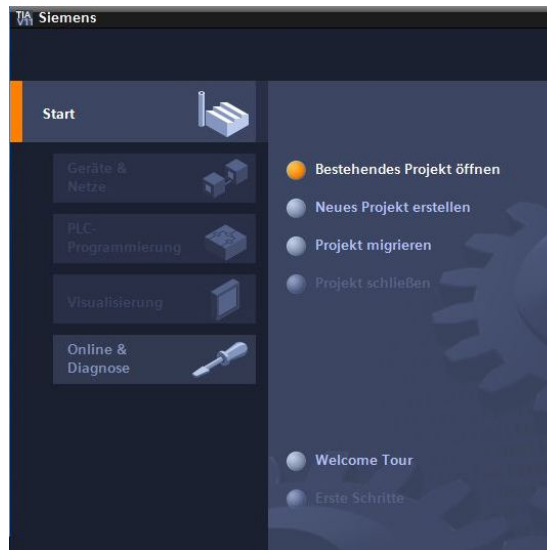


Abbildung 2.1: Ausschnitt aus der Startansicht des TIA-Portals

2.2.2 PLCsim

Bei PLCsim handelt es sich um eine Software, in der eine Speicherprogrammierbare Steuerung (SPS) auf dem lokalen Rechner nachgestellt werden kann. Sie kann geladen werden und der Benutzer kann durch Mausclick oder durch Werteingabe die Prozessvariablen der Steuerung beeinflussen. Nach und während der Programmbearbeitung können alle Speicherbereiche, z.B. Akkus oder Adressregister, beobachtet und das Ergebnis an den Ausgängen eingesehen werden. In der Abb. 2.2 ist die Programmoberfläche mit der Central Processing Unit (CPU) zu sehen. Auf der rechten Seite ist ein Unterfenster mit den Inhalten der Akkumulatoren und des Statuswortes enthalten. Der Benutzer kann sich die Oberfläche durch Anordnung der verschiedenen Unterfenster selbst anpassen.

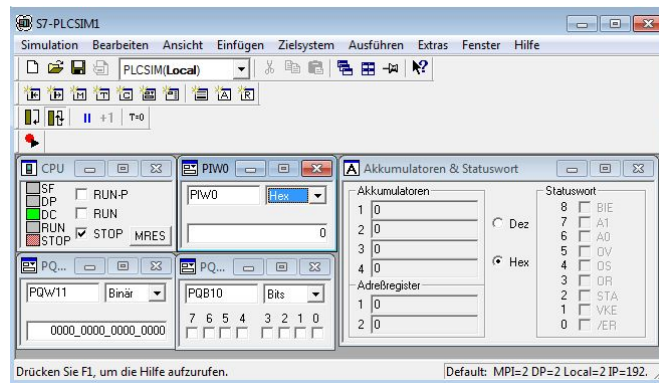


Abbildung 2.2: Programmansicht von PLCsim

2.3 LabVIEW

National Instruments stellt mit LabVIEW eine Software zur Verfügung, die auf Laborarbeit und dortige messtechnische Anforderungen spezialisiert ist. Es ist eine grafische Programmiersoftware, die es zulässt, schnell und bequem eine Windows-Fensteranwendung zu erstellen. Auf der LabVIEW-Seite beschreibt National Instruments sein Produkt mit dem Satz:

“LabVIEW ist eine leistungsstarke Systemdesignsoftware, die speziell auf die Anforderungen von Ingenieuren und Wissenschaftlern zugeschnitten ist.” [National-Instruments (a)]

Viele Nutzer in Industrie und Lehre geben dem Konzern recht, indem sie die Software einsetzen. Die Software ist in erster Linie für Laboranwendungen entwickelt worden, um eine computerbasierte Messwertauswertung schnell und einfach zu erstellen. Mittlerweile wird das Tool mit immer mehr Funktionen erweitert. Es wird auch zunehmend in der Prozessvisualisierung eingesetzt. Schwächen zeigt LabVIEW in Bereichen, die in grafischen Programmieroberflächen unübersichtlich werden, beispielsweise die Erstellung von Protokollen für Bussysteme. Dafür kann jedoch auch ein Skriptbaustein eingesetzt werden, der einen C-Code enthält. Leider ist diese Funktion nicht in dem Grundpaket der Software enthalten und muss wie viele andere Bibliotheken separat erworben werden.



Abbildung 2.3: Logo von LabVIEW

2.4 NetToPLCsim

Man kann zwar in PLCsim eine Programmierschnittstelle einstellen, jedoch handelt es sich um keine reale Schnittstelle, die über das Netzwerk direkt genutzt werden kann. Diese Schnittstelle ist eine S7Online-Schnittstelle, die von der Programmiersoftware und teilweise von dem Visualisierungssystem genutzt wird. Selbst wenn das Listenfeld auf **TCP/IP** steht, handelt es sich also nicht um eine echte Ethernet-Verbindung. Die Netzwerkerweiterung NetToPLCsim baut eine Verbindung zu PLCsim auf und leitet nun Anfragen, die über der **IP**-Adresse des Rechners mit einem bestimmten Format eintreffen, an PLCsim weiter und gibt die entsprechende Antwort zurück. NetToPLCsim ist keine von Siemens veröffentlichte Software, sondern eine Open-Source-Anwendung eines Nutzers. In den Vorgängerversionen wurde von dieser Software auch die Schnittstelle S7ProSim (siehe 3.2.1) genutzt. Die aktuelle und in diesem Test verwendete Version nutzt die beschriebene S7-Online-Verbindung.

2.5 Loopback Device

Jede Netzwerkschnittstelle hat eine Loopback-Schnittstelle. Die bekannteste ist der Localhost mit der Adresse 127.0.0.1. Der Adressraum 127.0.0.1 bis 127.255.255.254 ist für Loopback-Kanäle reserviert. Diese Adressen werden auch als Loopback-Adressen bezeichnet. Pakete, die an eine Loopback-Adresse geschickt werden verlassen die Netzwerkkarte nicht, sondern werden automatisch wieder zurückgeroutet und erscheinen, als würden sie aus dem Netzwerk kommen, selbst wenn der Rechner mit gar keinem Netzwerk physikalisch verbunden ist. In dem Betriebssystem Windows 7 ist es möglich, in der Netzwerkkonfiguration einen Loopback-Kanal mit einer **IP**-Adresse aufzubauen, die nicht im Loopback-Adressbereich liegt. Diese virtuelle Verbindung wird hier genutzt, um NetToPLCsim mit dem eigenen Rechner zu verbinden. So kann der Umstand, einen zweiten Rechner zu benötigen, umgangen werden. Bei der in Abschnitt 3.2.3 erläuterten TCP-Verbindung wird ebenfalls ein Loopback Device eingesetzt. [Schreiner (2009)]

2.6 Einleitende Literaturhinweise

Einleitend kann in diese Themen die folgende Literatur unterstützen. Bei der Einarbeitung in LabVIEW ist [Georgi und Metin (2007)] sehr hilfreich. Interprozesskommunikation wird in [Haase (2008)] behandelt. Die Einarbeitung in die SPS-Programmierung ist in [Berger (2011)] beschrieben, als Nachschlagewerk ist [Wellenreuther und Zastrow (2005)] zu empfehlen. Für die später erläuterten verschiedenen Lösungsansätze sollten jeweils folgende Werke zum Einarbeiten verwendet werden:

ActiveX: [Bettermann (2002)], [Bengel (2000)], [Emmerich (1999)], [Haase (2008)]

OPC: [Jomaa (2000)], [Lange (2002)]

TCP/IP: [Schreiner (2009)]

3 Analyse

In diesem Abschnitt werden die Anforderungen des Systems beschrieben und festgelegt. Nach der Aufstellung der Anforderungen werden diese nach Wichtigkeit bewertet und entsprechend gewichtet. Anschließend werden die Anforderungen beschrieben und mit den Lösungsansätzen verglichen. Im Ergebnis wird der Lösungsansatz, der die meisten Anforderungen erfüllt, ausgewählt.

3.1 Anforderungen

Stabilität

Die wichtigste Anforderung an Software und deren Schnittstellen ist Zuverlässigkeit und Stabilität. Als Maß für Stabilität werden die Eindrücke und die Absturzhäufigkeit während der Analysearbeiten subjektiv betrachtet.

Benutzerfreundlichkeit

Jede Software sollte benutzerfreundlich sein. Benutzerfreundlich ist eine Software dann, wenn die Benutzerzielgruppe damit gut zurechtkommt. Heute wird immer öfter das Ziel gesetzt, Software mit intuitiver Bedienung zu erstellen. Das ist jedoch eher der Konsumenten-Software vorbehalten. Die hier angesprochene Benutzergruppe umfasst Programmierer und Inbetriebsetzer, die Vorwissen mitbringen. Außerdem ist es in Industriesoftware oft nicht möglich, eine intuitive Bedienung zu realisieren, da die Software in den meisten Fällen zu komplex ist und somit eine Bedienungsanleitung erfordert. Zum Vergleich der Lösungsansätze werden die Beobachtungen aus der Softwareerstellung dokumentiert.

Reaktivität

Da sich ein reales System auch in Echtzeit verändert, sollte dieses Echtzeitverhalten auch in der Simulation enthalten sein. Deshalb sollten die Rechenzeiten der Simulation und der Übertragung an die virtuelle Steuerung so gering sein, dass die Verhaltensmuster der Anlage

wie die eines realen Systems wirken. Dabei wird jede Verbindung zum Vergleich einem Benchmarktest unterzogen, der in den folgenden Analysen noch beschrieben wird.

Abbildung auf physikalische Speicheradressen

Ein reales System ist häufig mit verschiedener SPS-Hardware aufgebaut. Dezentrale Peripherie ist fast immer verbaut. Um das Programm der Steuerung bei der Inbetriebsetzung sofort verwenden zu können, sollte es also möglich sein, die Datenbereiche in der SPS zu beschreiben und zu lesen, welche in der realen Anlage auch genutzt werden. Die hier in Frage kommenden Datenbereiche einer SPS sind in Tabelle 3.1 aufgeführt.

Datenbereich	Abk.	Beschreibung
Eingänge,Ausgänge	E/A	Lokale Hardwareanschlüsse der SPS
Peripherieeingänge/-ausgänge	PE/PA	Eingänge und Ausgänge von Peripheriegeräten
Datenbausteine	DB	Inhalte von Datenbausteinen
Merker	M	Globale Variablen für Zwischenergebnisse

Tabelle 3.1: Relevante Datenbereiche der Steuerung

Netzwerkfähigkeit in IP-Netzen

Gelegentlich ist es sinnvoll, die Simulation und die Programmiersoftware auf getrennten Rechnern aufzubauen. Die Performanz eines Rechners lässt sich deutlich verbessern, wenn aufwändige Berechnungen ausgelagert werden. Ein Vorteil eines separaten Simulationsrechners ist die Möglichkeit, mit mehreren Nutzern darauf zuzugreifen, eventuell sogar zeitgleich. Außerdem ist eine zentrale Datenhaltung verfügbar, die bei manchen Simulationen notwendig sein kann.

Erweiterbarkeit und Übertragbarkeit

Wie jede Software oder Steuerung sollte auch eine Simulation mit geringem Aufwand erweiterbar sein. Eventuell wird eine Anlage mehrmals programmiert und hat jedes Mal nur geringe Abweichungen. Besteht dann bereits eine Simulation und ein Programm, muss nur wenig geändert werden und die Inbetriebsetzung kann schneller erfolgen.

Lokaler Betrieb und Anschluss einer echten SPS möglich

Einige Verbindungen können nicht auf dem gleichen Rechner aufgebaut werden, weil es sich dabei um typische Netzwerkverbindungen handelt, bei denen eine Verbindung mit dem Localhost nicht vorgesehen ist. Dann wird entweder eine richtige SPS benötigt, um eine Simulation mit LabVIEW damit zu steuern, oder ein weiterer Rechner mit PLCsim und einer Netzwerkerweiterung, die in Abschnitt 2.4 noch erläutert wird. Eine Möglichkeit, diesen Umstand zu umgehen, ist das Loopback Device (siehe 2.5).

Nicht umsetzbare Anforderungen

Als eine weitere Anforderung ist bei Simulationen jeglicher Art zu beachten, dass sie das reale System mit allen seinen Eigenschaften und denen, die durch die Umwelt des Systems darauf einwirken, beschreiben sollen. Viele Parameter können von Software nicht beachtet werden. Mechanische Toleranzen oder Fehler, die von den Anlagenbauern oder anderen beteiligten Personen gemacht werden, können in einer Software nur sehr bedingt beachtet werden. Oft können solche aber als geringfügig eingestuft werden, da sie auf das Steuerungsprogramm meist nur geringen Einfluss haben oder kurzfristig korrigiert werden können.

Zusammenfassung der Anforderungen

In der Tabelle 3.2 sind die aufgeführten Anforderungen zusammengefasst und gewichtet. Stabilität und Benutzerfreundlichkeit wurden hoch gewichtet, weil instabile oder nicht benutzerfreundliche Software die Bearbeitung der Aufgabe verzögert, und der Zeitvorteil durch die Simulationen nicht gegeben wäre. Reaktivität ist besonders wichtig, da eine reale Anlage auch keine großen Zeitverzögerungen in der Signalübertragung hat. Sie wäre also nicht gut repräsentiert. Die Abbildung auf physikalische Speicher ist unbedingt nötig, um das erstellte SPS-Programm in der realen Anlage zu verwenden.

Pos.	Anforderung	Gewichtung
1	Stabilität	++
2	Benutzerfreundlichkeit	++
3	Reaktivität	++
4	Abbildung auf phys. Speicherbereiche	++
5	Netzwerkfähig für mehrere Nutzer	+
6	Erweiterbarkeit und Übertragbarkeit	+
7	Lokaler Betrieb	+
8	Anschluss echter SPS möglich	+

Tabelle 3.2: Anforderungen an die Simulation

Die Gewichtung bestimmt in der Auswertung der Analyse, wie sehr die Einzellwertung in die Gesamtwertung eingeht.

3.2 Lösungsansätze

3.2.1 ActiveX mit COM

Das Component Object Model (**COM**) ist ein Softwarestandard von Microsoft und die Basis für ActiveX-Steuerelemente. Anfang der 90er Jahre wurde festgestellt, dass es immer wichtiger ist, zwischen Prozessen verschiedener Software zu kommunizieren. Dadurch entstand zunächst **OLE**, eine Technologie zur Erstellung von Verbunddokumenten. Daraus entwickelten sich dann neue **COM**-gestützte Technologien, die heute besser bekannt sind als ActiveX-Steuerelemente. Mittels dieser Technik kann zwischen den Prozessen kommuniziert werden, unabhängig von der Programmiersprache der verschiedenen Programme. Sie wurde immer weiter entwickelt und hat sich bis heute auf dem Markt gehalten. Daher wird in dieser Arbeit angenommen, dass die meisten Fehler dieser Schnittstelle bereits gefunden und beseitigt wurden, was zu einer hohen Verfügbarkeit und Stabilität führt. Es stehen für viele Anwendungen bereits Schnittstellen als Dynamik-Link-Librarie (**DLL**) von den Entwicklern zur Verfügung, die Methoden enthalten, um zwischen den Prozessen Daten auszutauschen. Diese Bibliotheken werden auch als "Middleware"[[Wikipedia \(b\)](#)] bezeichnet. [[Bettermann \(2002\)](#)] beschreibt dieses Model sehr anschaulich. Es gibt in der Automatisierungstechnik viele Anwendungsfälle dieser Art. Komplexe Prozessleitsysteme wie Siemens PCS7 nutzen ActiveX für ihre Visualisierungen. Auch viele andere Visualisierungssysteme arbeiten mit ActiveX-Steuerungen. Die Software

PLCsim bietet seit Version 5.4 so eine Schnittstelle, genannt S7ProSim, in einer **DLL**-Datei an. Sie ist in [Siemens (2007)] beschrieben und kann in LabVIEW importiert werden.



Abbildung 3.1: Logo von ActiveX

3.2.2 OPC-Verbindung

OLE for Process Control ist eine Weiterentwicklung von OLE und ein Kommunikationsstandard in der Prozessleittechnik. Man kann OPC auch als Erweiterung des **COM** betrachten. [Bettermann (2002)] beschreibt die Schnittstelle in dieser Form. Ein Zusammenschluss von Herstellern der Automatisierungskomponenten hat die Entwicklung von **OPC** vorangetrieben. Es gibt die **OPC** Foundation mit mehr als 200 Mitgliedsfirmen (Quelle: [Bettermann (2002)]). Die Foundation hat bestimmte Regeln aufgestellt, um einen beliebigen **OPC** Client mit einem Server zu verbinden. In den Regeln werden bestimmte Methoden festgelegt, die in den Objekten definiert sein müssen, um mit anderen **OPC**-Objekten kommunizieren zu können. Da **OPC** neben dem **COM** auch das Distributed Component Object Model (**DCOM**) nutzt, ist es netzwerkfähig [Jomaa (2000)]. Das ist mit der zuvor in 3.2.1 beschriebenen ActiveX-Verbindung nicht möglich. Viele serverbasierende Prozessleitsysteme nutzen **OPC** für ihre Kommunikation über Netzwerkschnittstellen zu mehreren Clients. Für eine Verbindung ist eine **OPC**-Serversoftware nötig. In dieser Untersuchung wird eine Software von Siemens genutzt.



Abbildung 3.2: Logo von OPC

3.2.3 TCP/IP-Verbindung

Der aktuelle Netzwerkstandard für Local Area Network (**LAN**) ist eine Verbindung aus dem Transmission Control Protocol (**TCP**) und dem Internet Protocol (**IP**). Eine **SPS** stellt auch eine **TCP/IP**-Verbindung zur Verfügung. Diese Art der Peripherieanbindung wird immer mehr in der Industrie eingesetzt und unter dem Namen Industrial Ethernet (**IE**) geführt¹. Auch die verwendete **Soft-SPS** stellt mit der Software NetToPLCsim eine Netzwerkschnittstelle zur Verfügung. Um diese Variante auf einem Rechner nutzen zu können, ist ein Loopback Device nötig (siehe 2.5), da nur so die Telegramme wieder an das eigene System zurückgesandt werden und trotzdem so erscheinen, als kämen Sie aus einem Netzwerk. Ohne das Loopback Device wäre ein zweiter Rechner nötig, um diese Lösung nutzen zu können.

3.3 Analyse und Bewertung der Lösungsansätze

Um die Lösungsansätze durch Tests zu analysieren, muss das verwendete Gerät beachtet werden. Die Daten des für die Tests verwendeten Rechners sind in der Tabelle 3.3 aufgeführt.

Parameter	Wert / Typ
CPU	Intel Core i7-2600 3,80 GHz
RAM	16GB
Betriebssystem	Windows 7 Enterprise SP1 64Bit

Tabelle 3.3: Parameter des Testrechners

3.3.1 Bewertung von ActiveX mit COM-Objekt

Stabilität

Subjektiv betrachtet erscheint diese Variante sehr stabil. Während der Untersuchung ist es zu keiner Instabilität gekommen.

Benutzerfreundlichkeit

Die Benutzerfreundlichkeit kann insgesamt als gut bezeichnet werden. Der Benutzer findet sich mit der Dokumentation der Schnittstelle und den einzelnen Methoden schnell zurecht und der Einstieg ist einfach.

¹Siemensprodukte führen den Namen ProfiNet.

Reaktivität

Das Echtzeitverhalten wird untersucht, indem 1000 Datenworte² über die Verbindung geschrieben werden. Die Soft-SPS PLCsim ist programmiert, diese Werte in einen anderen Speicher zu kopieren, welcher dann wieder ausgelesen wird. Dadurch wird ein Programmdurchlauf mit dem Schreiben und Lesen von Daten simuliert. Die Zeitdauer für diese Übertragung wird automatisch gemessen. Das Vorgehen ist in Abbildung 3.3 schematisch dargestellt.

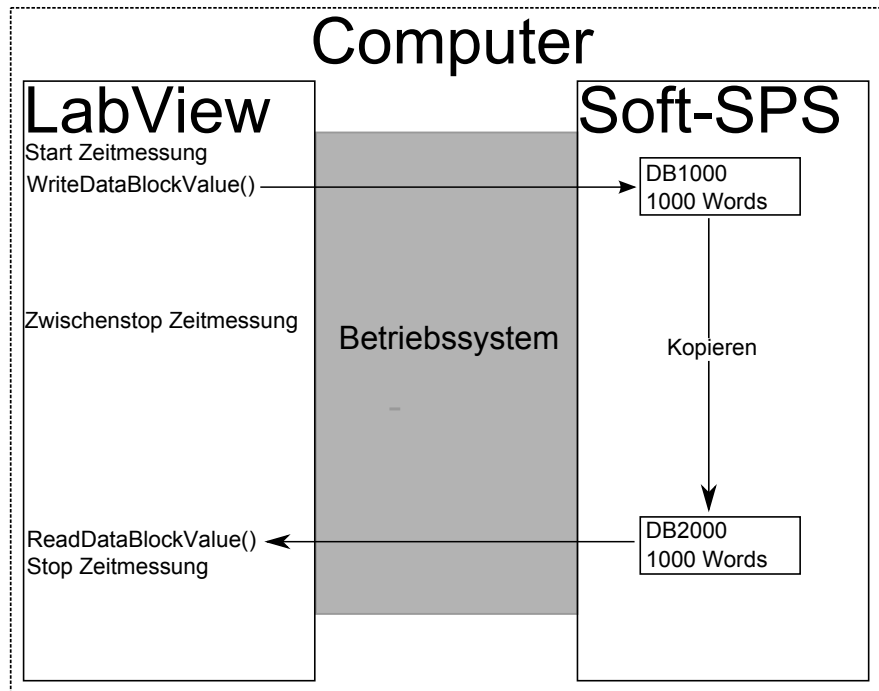


Abbildung 3.3: Geschwindigkeitstest der ActiveX-Verbindung

Das Bild zeigt, dass nach Beginn der Zeitmessung eine Schreibmethode aus der Schnittstelle aufgerufen wird. Sie wird 1000 mal aufgerufen, um jeweils ein Datenwort zu senden. Die Datenworte werden im Zyklus der SPS in einen anderen Datenbaustein (DB) kopiert. Nachdem alle Werte gesendet wurden, wird die Lesemethode mehrmals aufgerufen, um die Werte aus dem anderen DB anzufordern. Das Betriebssystem verwaltet die eventgestützten Methodenaufrufe. Mit dieser Messung wird das Zeitverhalten untersucht, indem die Zeit für das Schreiben und das Lesen jeweils gemessen werden. Es wurden zehn Messungen durchgeführt, um einen statistisch aussagekräftigen Wert für das Zeitverhalten zu erhalten. Bei der Messung müssen die Rechnerparameter beachtet werden, die am Anfang dieses Abschnitts in

²1 Datenwort=2 Byte

Tabelle 3.3 aufgeführt sind. Die Zykluszeit der SPS kann hier durch die wenigen Operationen im Programm vernachlässigt werden. Die Ergebnisse sind in Tabelle 3.4 zu erkennen.

Messung	Zeitdauer[ms]		
	Schreiben	Lesen	Summe
1	331	307	638
2	316	287	603
3	338	290	628
4	337	290	627
5	289	288	577
6	315	287	602
7	331	289	620
8	324	289	613
9	312	290	602
10	316	288	604
Mittelwert	320,9	290,5	611,4

Tabelle 3.4: Zeitanalyse ActiveX-Verbindung

Die Übertragungsgeschwindigkeit erscheint recht niedrig, wobei jedoch die Anzahl der Werte und die Rechnerleistung beachtet werden muss. Hintergrundprozesse können die Zeiten beeinflussen. Man kann hier zwar nicht mehr von Echtzeit sprechen, jedoch kann hier noch eine Signallaufzeit in einer echten Anlage mit langen Leitungswegen und Übergangswiderständen recht gut dargestellt werden. Eine weitere Untersuchung zeigt, dass Einzelbits schneller übertragen werden. Die Zeitdauer ist abhängig von der Anzahl der zu übertragenden Werte. Die Ergebnisse sind in dem Diagramm in Abb. 3.4 zu sehen.

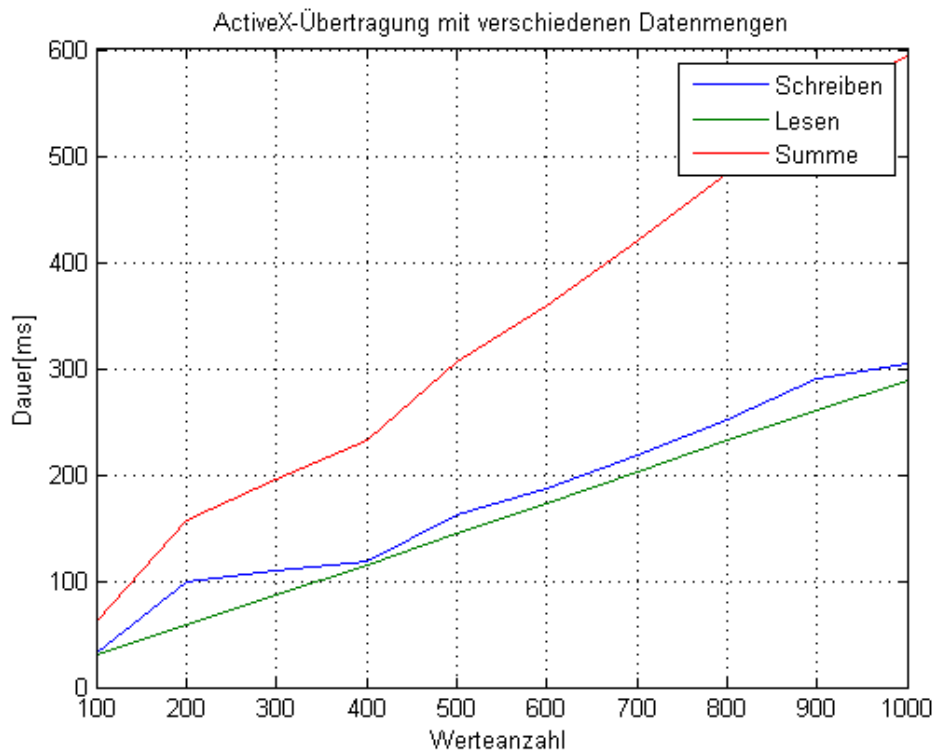


Abbildung 3.4: ActiveX-Übertragungsdauer abhängig von der Werteanzahl

Abbildung auf physikalische Speicherbereiche

Auf die Hardwareeingänge der SPS kann nicht direkt zugegriffen werden, jedoch ist es möglich, auf die Peripherie-Variablen direkt zuzugreifen. Das entspricht vielen modernen Anlagen, weil Sensorik und andere Feldgeräte fast immer über dezentrale Peripherie angebunden sind. Somit kann das Programm später mit hoher Wahrscheinlichkeit auch für die reale Anlage verwendet werden.

Netzwerkfähigkeit in IP-Netzen

Da ActiveX-Verbindungen innerhalb eines Rechners ablaufen, ist ohne eine weitere Kommunikationssoftware keine Verbindung über die Rechnergrenze hinweg möglich.

Erweiterbarkeit und Übertragbarkeit

Die Simulation kann jederzeit erweitert werden. Die zu schreibenden Operanden werden durch Konstanten an den Methoden definiert und können leicht geändert werden. Dadurch steigt jedoch die Rechenzeit und die Reaktivität kann sich verschlechtern. Die Blöcke für die Variablen werden hintereinander geschaltet und es kann jederzeit ein weiterer Block eingefügt werden. Der Aufbau ist in Abbildung 3.5 zu sehen, wobei es sich um einen Auszug aus dem Testprogramm handelt. Hier wird die Schreibmethode in einer Schleife gezeigt. Der Byteindex wird mit den Schleifendurchläufen erhöht, um den **DB** von oben nach unten zu füllen. Nach dem Schreibvorgang muss die Steuerung wieder in den “ContinuousScan”-Modus geschaltet werden, weil sonst kein Zyklus durchgeführt wird. So soll vermieden werden, dass während eines Zyklus Datenmanipulationen durchgeführt werden können, die zu Inkonsistenzen führen.

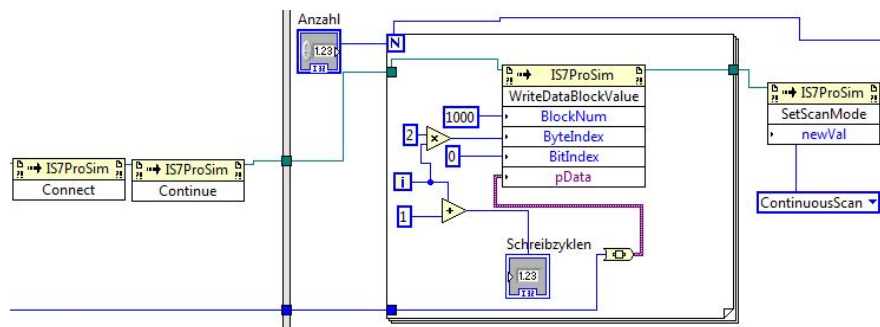


Abbildung 3.5: Auszug aus dem ActiveX-Testprogramm

3.3.2 Bewertung der OPC-Verbindung

Stabilität und Benutzerfreundlichkeit

Die Projektierung einer **OPC**-Verbindung ist recht aufwendig und sollte nur durchgeführt werden, wenn ein Netzwerkzugriff auf eine Steuerung unbedingt nötig ist. Zudem ist es nicht mit jeder Serversoftware möglich, eine Verbindung zu einem **COM**-Objekt auf dem eigenen Rechner aufzubauen. Aus diesem Grund ist in diesem Test eine “echte” Steuerung am Simulationsarbeitsplatz nötig, um die Simulation durchzuführen. Zur sicheren Übertragbarkeit des Programmes sollte sie identisch mit der Steuerung der realen Anlage sein, da sonst eventuell unabsichtlich zu viele Datenbausteine, Zähler oder Timer verwendet werden.

Reaktivität

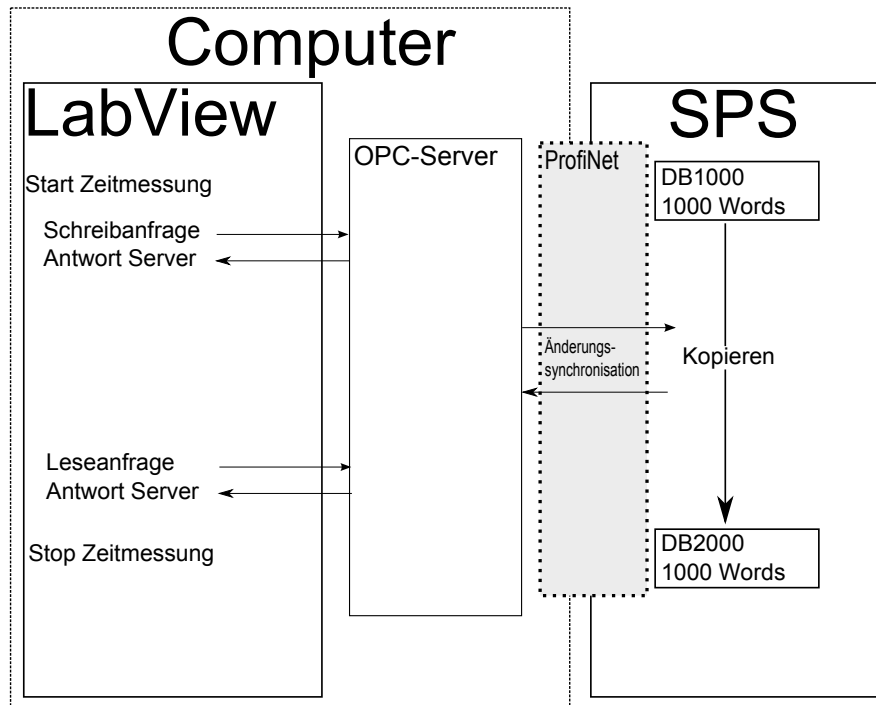


Abbildung 3.6: Geschwindigkeitstest der OPC-Verbindung

Der **OPC-Server** von Siemens ist nicht in der Lage, eine Verbindung zum eigenen Rechner aufzubauen sondern durchsucht nur das Netzwerk. Es ist also nicht möglich, damit eine Verbindung zu PLCsim herzustellen, wenn die Soft-**SPS** auf dem gleichen Rechner läuft wie der **OPC-Server**. Die Soft-**SPS** lässt sich nicht öffnen, wenn Anwendungen mit "echten" CPUs verbunden sind. Wird der Komponenten Konfigurator auf dem Rechner ausgeführt und ist mit einem Subnetz verbunden³, kann PLCsim nicht zeitgleich laufen. Auch der Versuch an einem Rechner mit zwei Netzwerkkarten ist daher fehlgeschlagen. Verwendet man eine echte **SPS** oder installiert PLCsim mit NetToPLCsim auf einem anderen Rechner im Netzwerk, ist die Verbindung möglich. Durch die Möglichkeit, eine echte **SPS** anzuschließen scheint diese Simulationsvariante sehr realitätsnah, da auf alle Datenbereiche zugegriffen werden kann (siehe Tab. 3.1). Die **OPC-Verbindung** wird auf dem gleichen Rechner (siehe Tab. 3.3) mit ebenfalls 1000 Datenworten getestet. Diesmal muss jedoch eine echte **SPS** genutzt werden, da eine Verbindung zu PLCsim auf dem gleichen Rechner nicht möglich ist. Das Testverfahren ist in Abb. 3.6

³Er muss verbunden sein, sonst kann er keine Variablen lesen

dargestellt. Darin ist zu sehen, dass die eigentliche Kommunikation mit der Steuerung von dem OPC-Server über eine ProfiNet-Verbindung ausgeführt wird. Die Zeitmessung wird gestartet und die zu schreibenden Daten werden an den OPC-Server gesendet. Der Server antwortet mit einer Bestätigung der Schreiboperation. Im Server sind die Variablen der Steuerung hinterlegt und werden regelmäßig bei Änderungen synchronisiert. Die Daten werden also nicht direkt aus der SPS abgefragt. Werden Daten im OPC-Server geändert, werden diese an die Steuerung weitergeleitet. Eine Datenanfrage beantwortet der OPC-Server mit den Daten, die bei ihm hinterlegt sind. Sind die Daten im OPC-Server zum Abfragezeitpunkt nicht aktuell, ist das für den abfragenden Prozess nicht ersichtlich, was zu Fehlern führen könnte. Die Ergebnisse des Tests sind wie zuvor ausgewertet in Tabelle 3.5.

Messung	Zeitdauer[ms]		
	Schreiben	Lesen	Summe
1	1354	508	1862
2	1276	509	1785
3	1257	509	1766
4	1271	510	1781
5	1291	509	1800
6	1321	508	1829
7	1349	509	1858
8	1282	508	1790
9	1305	509	1814
10	1309	508	1817
Mittelwert	1301,5	508,7	1810,2

Tabelle 3.5: Zeitanalyse der OPC-Verbindung

Die Zeitdauer des Schreibens ist abhängig von der Anzahl der zu übertragenden Werte, die Zeitdauer des Lesens ist konstant bei etwa 509ms (siehe Abb. 3.7).

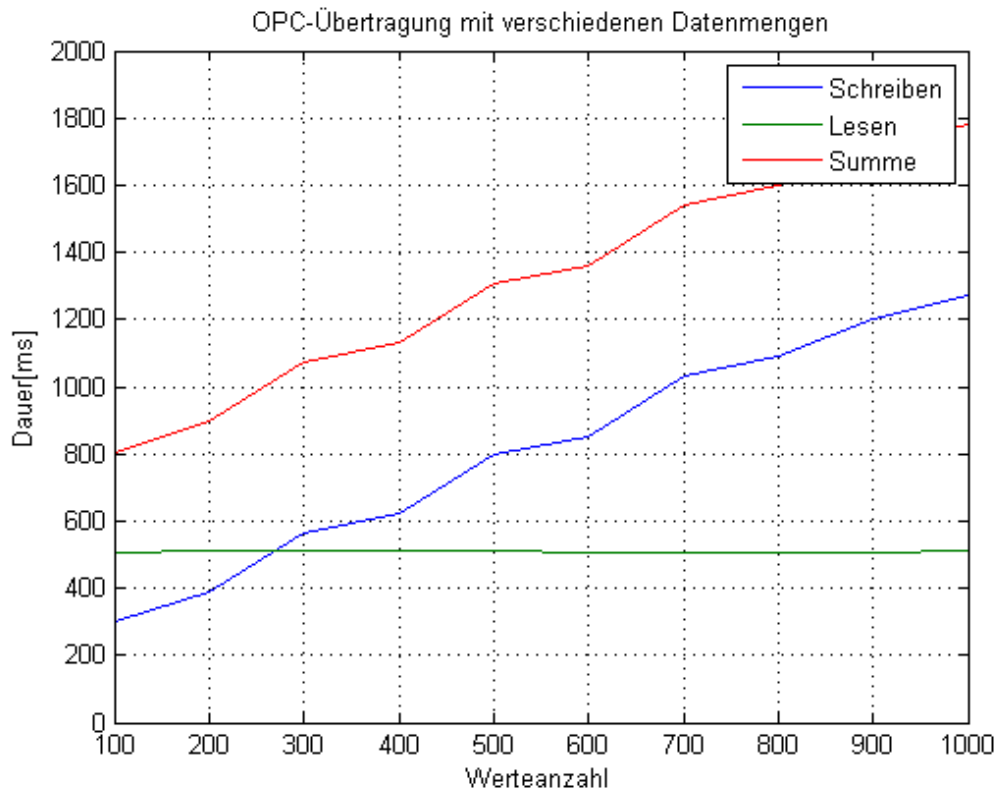


Abbildung 3.7: OPC-Übertragungsdauer abhängig von der Werteanzahl

Es wurde beobachtet, dass die Rechenzeiten bei mehreren Messungen sehr abweichend sind. Das lässt die Vermutung zu, dass Hintergrundprozesse die Zeitbilanz sehr stark beeinträchtigen, was im Rahmen der Anforderungen nicht tragbar wäre. Es hat sich gezeigt, dass eine Werteabfrage mit LabVIEW nicht automatisch ein erneutes Holen der Werte aus der SPS anstößt. Erst nach etwa einer Minute werden die Variablen aus der SPS gelesen und im OPC-Server erneuert. Nur, wenn neben LabVIEW auch der Siemens OPC-Scout aktiv ist und auf "Beobachten" geschaltet ist, sind die gelesenen Werte aktuell. Das macht einen Echtzeitbetrieb noch schwieriger, da zusätzliche Rechnerleistung für den OPC-Scout genutzt wird. Die OPC-Verbindung kann auf alle Datenbereiche, auch den E/A-Bereich zugreifen. Diese Verbindung bietet also vielseitigere Möglichkeiten, als die ActiveX-Verbindung, ist aber deutlich langsamer und benötigt eine richtige SPS.

Netzwerkfähigkeit in IP-Netzen

OPC ermöglicht auch einen rechnerübergreifenden Zugriff auf Daten.

Erweiterbarkeit und Übertragbarkeit

Eine Erweiterung ist immer vergleichsweise aufwändig, weil mehrere Systeme geladen werden müssen. Eine Änderung der Variablen muss separat in den OPC-Server geladen werden und in LabVIEW muss eine neue Verbindung angelegt werden, mit einer URL zu jeder einzelnen Variable (siehe Abbildung 3.8). Das ist viel Projektierungsaufwand im Gegensatz zu den anderen Lösungsansätzen.

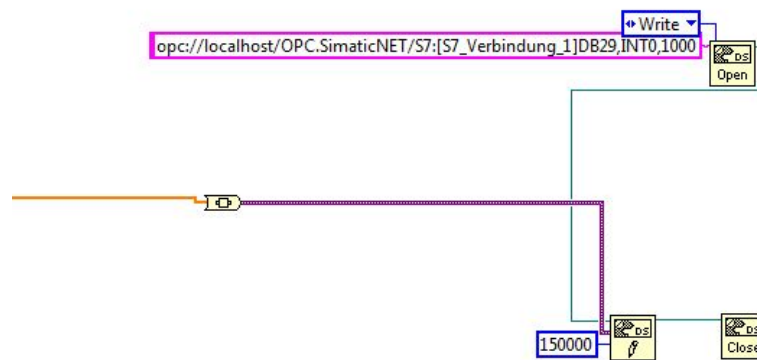


Abbildung 3.8: Auszug aus dem OPC-Testprogramm

Eine Übertragung in ein anderes Projekt ist auch sehr aufwendig, da die Variablen im OPC-Server genau auf das Programm abgestimmt werden müssen. Auch der Name müsste überall angepasst werden. Eventuell gibt es mit einem OPC-Zusatzpaket für LabVIEW noch weitere Möglichkeiten, die bei dieser Untersuchung nicht zur Verfügung standen. [National-Instruments (b)]

3.3.3 Bewertung der TCP-Verbindung

Für den Test wurde aus einem Forum von National Instruments eine Bibliothek für ProfiNet genutzt, die von einem Nutzer erstellt wurde und unter [Community] zu finden ist.

Stabilität und Benutzerfreundlichkeit

Die Stabilität von NetToPLCsim ist auf dem Testrechner nicht gut ausgefallen. Schon das Aktivieren des Bildschirmschoners hat das Programm zum Absturz gebracht. Die Benutzer-

freundlichkeit ist durch den komplexen Aufbau der Telegramme und die aufwändige Implementierung nur befriedigend, jedoch besser als bei der OPC-Variante. Eventuell kann hier mit kostenpflichtiger Soft- und Hardware von National Instruments Abhilfe geschaffen werden [National-Instruments (b)]. Es gibt Lösungen von National Instruments für die Nutzung von ProfiNet mit spezieller Hardware.

Abbildung auf physikalische Speicherbereiche

Es können alle Datenbereiche beschrieben werden. Das Programm kann gut in der realen Anlage verwendet werden.

Reaktivität

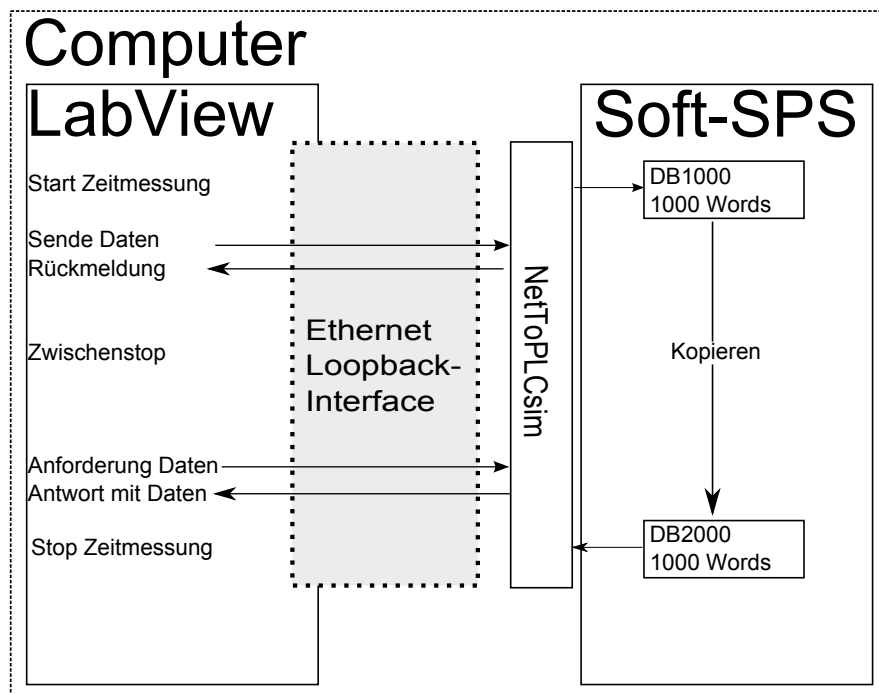


Abbildung 3.9: Geschwindigkeitstest der TCP/IP-Verbindung mit Loopback-Interface

In der Grafik ist zu sehen, dass die Telegramme nach Beginn der Zeitmessung zunächst an den Ethernetadapter geschickt werden. Durch das Loopback-Interface werden sie wieder an das eigene Gerät zurückgeschickt. Auf dem Rechner ist neben der Soft-SPS die Software NetToPLCSim installiert. Diese Software ist über eine S7Online-Verbindung mit der Soft-SPS verbunden, worüber die Daten an die CPU übergeben werden. Darin werden die Daten in

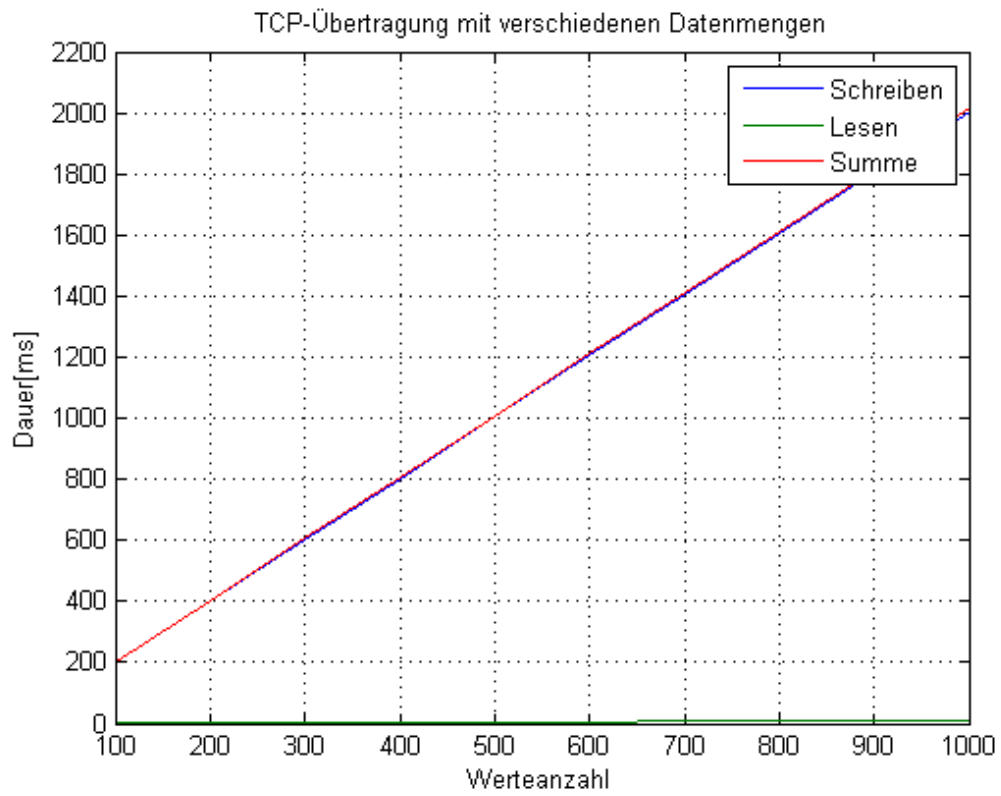
einen anderen DB kopiert und von LabVIEW mit einem Lesetelegramm angefordert. Nach dem Empfang des letzten Antwortpaketes wird die Zeitmessung wieder gestoppt.

Messung	Zeitdauer[ms]		
	Schreiben	Lesen	Summe
1	2005	10	2015
2	2005	10	2015
3	2006	10	2016
4	2009	10	2019
5	2007	10	2017
6	2008	10	2018
7	2008	10	2018
8	2007	10	2017
9	2006	10	2016
10	2008	10	2018
Mittelwert	2007	10	2017

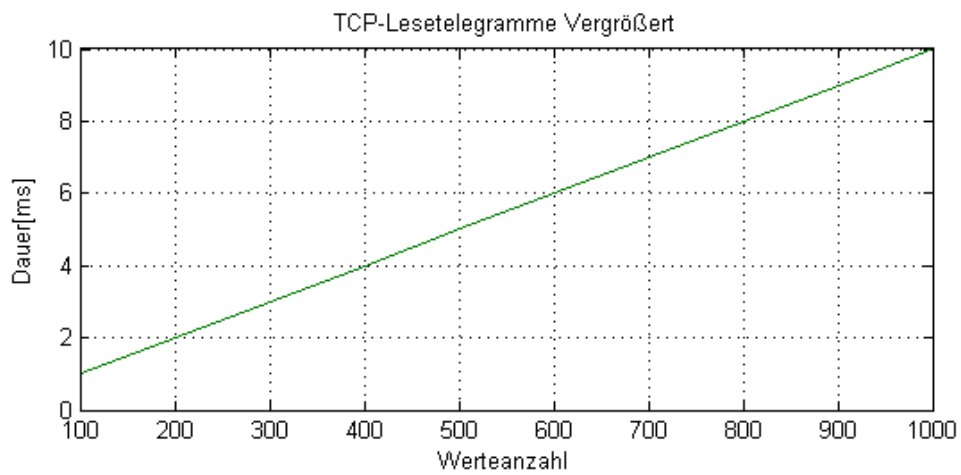
Tabelle 3.6: Zeitanalyse der TCP/IP-Verbindung

Bei der Messung sind sehr hohe Zeiten für das Schreiben auffällig. In der Schreiboperation muss bei dieser Variante für jedes Byte ein eigenes Telegramm gesendet werden, hingegen dazu wird beim Lesen jeweils für 200 Byte⁴ ein Telegramm gesendet. Aus diesem Grund entstehen die beschriebenen großen Zeitdifferenzen.

⁴Maximal kann ein Telegramm 222 Byte Daten enthalten. Hier wurden $10 \cdot 200 \text{ Byte} = 2000 \text{ Byte} = 1000 \text{ Worte}$ verwendet.



(a) Gesamtansicht



(b) Lesedauer vergrößert

Abbildung 3.10: TCP-Übertragungsdauer abhängig von der Werteanzahl

Die Übertragungsdauer ist, wie Abb. 3.10 zeigt, abhängig von der Warteanzahl. Die Lesedauer steigt linear von 0 bis 10 ms an. In diesem Fall ist ebenfalls zu sehen, dass die Übertragung jedes Telegrammes etwa eine Millisekunde dauert. Daher ist die Schreibdauer durch die vielen Telegramme hoch.

Netzwerkfähigkeit in IP-Netzen

Da es sich um eine **TCP/IP**-Verbindung handelt, ist diese Variante auch netzwerkfähig.

Erweiterbarkeit und Übertragbarkeit

Das Programm ist gut übertragbar, da mit diesem Zugriffsverfahren alle Datenbereiche beschrieben und gelesen werden können. Eine Erweiterung um einzelne Daten kann auch implementiert werden, da die Lese- und Schreibblöcke wie bei der ActiveX-Variante (siehe 3.3.1) hintereinander geschaltet werden und jederzeit ein weiteres Telegramm eingefügt werden kann. Das ist in Abb. 3.11 gut zu sehen und vergleichbar zu dem Aufbau der Methoden der **COM**-Schnittstelle, die in Abb. 3.5 dargestellt sind.

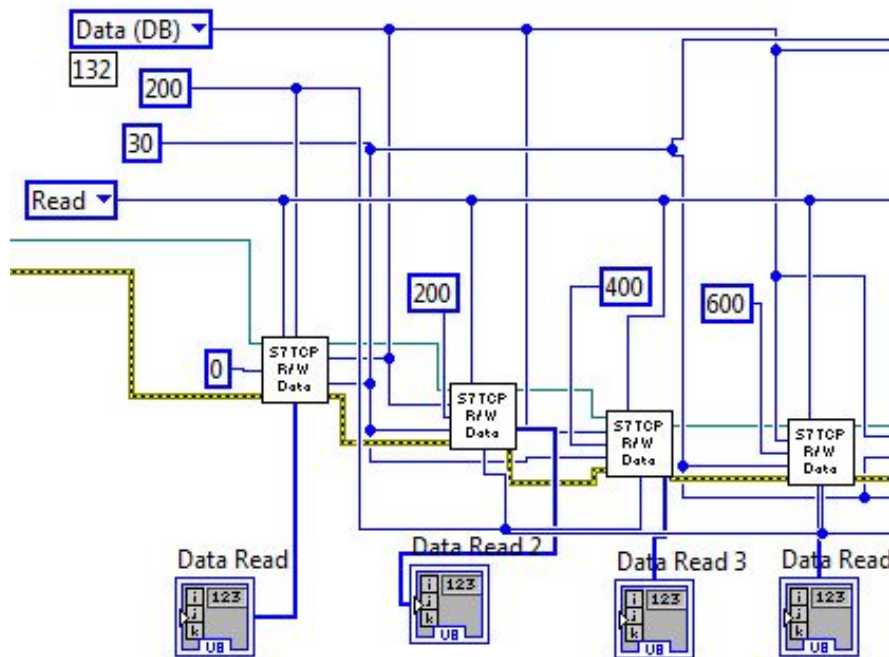


Abbildung 3.11: Ausschnitt aus Testprogramm TCP/IP: Lesetelegramme

3.3.4 Problematische Netzwerkprogrammierung in LabVIEW

Die **TCP/IP**-Verbindung ist ein gutes Beispiel dafür, dass LabVIEW nicht für derartige Programmierungen geeignet ist. In Abb. 3.12 ist erkennbar, dass sehr viele Einzeltelegramme in einer Schleife platziert sind. Es wurde beobachtet, dass ein Schleifendurchlauf mehr Rechenzeit benötigt, als die Aneinanderreihung der Telegramme. Um einen Kompromiss zwischen Implementierungsaufwand und Rechenzeitgewinn zu erzielen, wurden 100 Telegramme in einer Schleife 20 mal durchlaufen, um den Test durchzuführen. Es wurde versucht, die Programmstruktur eines Telegrammes so anzupassen, dass ein Schreiben von mehreren Werten möglich wird, was aber nicht gelang⁵. LabVIEW ist wie schon anfangs (siehe 2.3) beschrieben, nicht geeignet, um komplexe Netzwerkprotokolle zu programmieren. Mit einem Erweiterungspaket zur Erstellung von Skriptbausteinen wäre die Netzwerkprogrammierung einfacher [National-Instruments (b)]. Da der Umfang dieser Ausarbeitung nicht darauf basiert, eine TCP-Kommunikation aufzubauen, wird diese Problematik nicht weiter verfolgt.

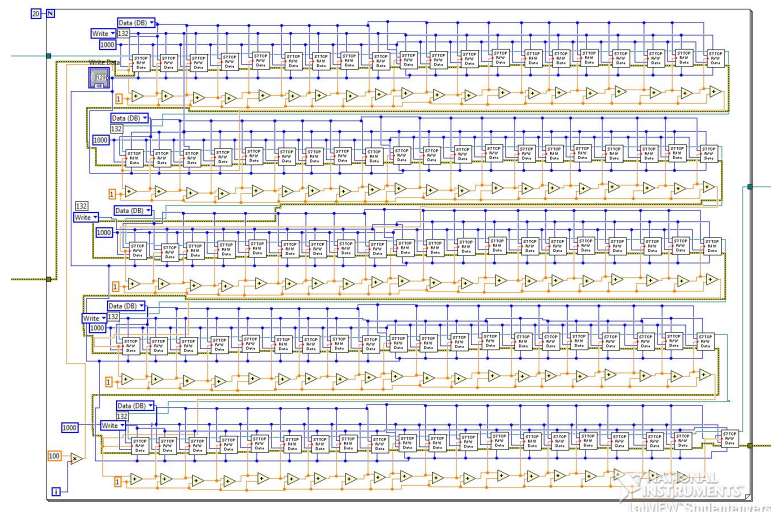


Abbildung 3.12: Aufwändige Programmierung mit Netzwerktelegrammen

⁵Der genaue Protokollaufbau von ProfiNet wird von Siemens nicht veröffentlicht. Man muss in eine kostenpflichtige Nutzergruppe eintreten um den genauen Telegrammaufbau zu erhalten.

3.4 Auswahl der besten Lösung

Pos.	Anforderung	ActiveX	OPC	TCP/IP
1	Stabilität	sehr gut	befriedigend	befriedigend
2	Benutzerfreundlichkeit	sehr gut	aufwendig	befriedigend
3	Reaktivität	611,4ms	1008ms	(2017ms)
4	Abbildung auf phys. Speicherbereiche	DB,PE/PA,M	Alle	Alle
5	Netzwerkfähig für mehrere Nutzer	nein	ja	ja
6	Erweiterbarkeit und Übertragbarkeit	gut	mäßig	gut
7	Lokaler Betrieb	ja	nein	ja
8	Anschluss echter SPS möglich	nein	ja	ja

Tabelle 3.7: Ergebnisse im Vergleich

Die Übersichtstabelle 3.7 zeigt, dass die ActiveX-Verbindung die durchschnittlich schnellste und benutzerfreundlichste ist. Die Zeit bei der TCP-Verbindung ist eingeklammert, weil es mit hoher Wahrscheinlichkeit durch Veränderung des Testprogrammes oder durch eine kommerzielle Lösung noch schneller ginge und die höchste Geschwindigkeit hier erwartet werden kann. In Tabelle 3.8 erfolgt die Punktevergabe. Besonders wichtige Anforderungen (siehe 3.1) sind mit “++” gekennzeichnet und gehen doppelt in die Gesamtwertung ein.

Pos.	Anforderung	Gewichtung	ActiveX	OPC	TCP/IP
1	Stabilität	++	+++	+	+
2	Benutzerfreundlichkeit	++	+++	+	++
3	Reaktivität	++	++	+	+
4	Abbildung auf phys. Speicherbereiche	++	++	+++	+++
5	Netzwerkfähig für mehrere Nutzer	+	–	++	+
6	Erweiterbarkeit und Übertragbarkeit	+	++	+	++
7	Lokaler Betrieb	+	++	–	+
8	Anschluss echter SPS möglich	+	–	+	++
	Summe		24	15	21

Tabelle 3.8: Vergleich der Lösungsansätze

Die Ergebnisse zeigen, dass die ActiveX-Verbindung die meisten Punkte erhält und somit die beste Lösung ist. Sie sticht hervor durch Benutzerfreundlichkeit und Stabilität. Trotz der

Einschränkung, die lokalen Hardwareanschlüsse der Steuerung nicht beschreiben oder lesen zu können, und der fehlenden Netzwerkfähigkeit, werden hier die meisten Anforderungen besser erfüllt als von den anderen Lösungsansätzen. Zudem ist es die von Siemens empfohlene Schnittstelle, um solche Simulationen durchzuführen. Die einleitende Anforderungsanalyse in Abschnitt 3.1 beschreibt die Anforderungen und das Vorgehen bei der Analyse.

4 Simulationen in LabVIEW erstellen

4.1 Frontpanel: Bedienoberfläche mit Anlagensvisualisierung

Im Frontpanel des sogenannten “LabVIEW VIs” befindet sich die Anlagensvisualisierung. Die Anlage wird hier schematisch dargestellt. LabVIEW bietet nicht so viele Möglichkeiten wie eine professionelle HMI- oder SCADA-Software¹, wie z. B. Siemens SIMATIC WinCC. Es können die gängigen Fenstersteuerelemente wie Schieberegler und Schaltflächen verwendet werden. Zusätzlich bietet LabVIEW noch einige unterschiedliche Binär- und Messwertanzeigen sowie Diagrammfelder. Auch Oszillogramme können dargestellt werden. In dieser Simulation wird der Aufzug mit einem Schieberegler dargestellt. Dieser kann auch als Anzeigeelement konfiguriert werden. Die Taster in der Kabine und in den Stockwerken werden mit normalen Schaltflächen nachgestellt. Binäranzeigen werden für die Leuchtmelder in den Tasten verwendet. Die Türen werden wie die Aufzugskabinen mit Schiebereglern dargestellt. Die Lichtschranken werden durch Schalter simuliert. Sie müssen gelegentlich nach dem Start des VIs eingeschaltet werden, da Lichtschranken aus Sicherheitsgründen drahtbruchsicher angeschlossen sind, müssen sie ein “1-Signal” führen, wenn sie nicht unterbrochen sind.

4.2 Blockdiagramm: Grafische Schnittstellenprogrammierung

Im Blockdiagramm wird die eigentliche Programmierung durchgeführt. Jedes Element, welches zuvor im Frontpanel platziert wurde, erscheint mit seinem Namen versehen auch gleichzeitig im Blockdiagramm. Die Elemente werden mit Linien mit anderen grafischen Programmelementen verschaltet. Zusätzlich können Programmteile in sogenannte Sub-VIs untergliedert werden. Die Sub-VIs empfehlen sich auch, wenn einige Programmteile mehrmals aufgerufen werden sollen. Es ist zu beachten, dass die betreffenden VIs ablaufinvariant ausgeführt werden, da sonst nur eine Instanz von ihnen gebildet und dauernd überschrieben wird².

¹Software zum Überwachen und Steuern von Prozessen[Wikipedia (c)] als Mensch-Maschine-Schnittstelle

²Dafür muss in den VI-Eigenschaften in der VI-Hierarchie im Abschnitt “Ausführung” die “Ablaufinvariante Ausführung” gewählt werden.

5 Beispiel einer Aufzugsimulation

Dieses Beispiel beschreibt die Simulation mit dem in [3.2.1](#) beschriebenen Verbindungsverfahren über Active-X-Kommandos. Hauptsächlich wird, um eine echte Anlage realistisch nachzustellen, der Peripheriebereich beschrieben und gelesen. Im Folgenden wird der Aufbau der Simulation beschrieben. In der Simulation wurde ein Integralregler genutzt, der unter dem Forumlink [[Forum](#)] gefunden werden kann. Zunächst werden einige allgemeine Informationen zu Aufzügen dargelegt.

5.1 Allgemeines zu Aufzügen

Die ersten Aufzüge wurden von Aristoteles und Archimedes erfunden. Bereits im antiken Rom wurden Aufzüge zur Personenbeförderung genutzt. Heute sind sie aus vielen Gebäuden nicht mehr wegzudenken, da die Gebäude in den letzten Jahrzehnten stark an Stockwerken gewonnen haben. Sogar Hochgeschwindigkeitsaufzüge mit Geschwindigkeiten von bis zu zehn Metern pro Sekunde (Burj Khalifa, Dubai) werden verbaut. In Forschungsinstituten wird auch schon an modernen Paternosteraufzügen gearbeitet, in denen die Kabinen seillos durch Magnetfelder gehalten und durch magnetische Linearantriebe bewegt werden. Diese Technologie soll in den nächsten Jahren in hohen Gebäuden sehr effizient Personen befördern. Durch die seillose Verarbeitung soll auch ein Schachtwechsel möglich sein, wodurch sehr kurze Fahrwege erreicht werden können. Die Fahrwegplanung ist in der Aufzugprogrammierung die größte Herausforderung. Die Steuerung sollte den Aufzug so sinnvoll verfahren, dass alle Nutzer in kürzester Zeit ihr Zielstockwerk erreichen und die Wartezeiten auf die Fahrkabine möglichst kurz sind. Dafür gibt es verschiedene Prinzipien, die in den folgenden Abschnitten erläutert werden.

5.2 Fahrwegplanung

Die Fahrwegplanung beschäftigt sich damit, eine Reihenfolge der abzuarbeitenden Kommandos, also der Fahraufträge, aufzustellen. Es gibt dafür verschiedene Prinzipien. Diese Steuerungskonzepte werden in [Wikipedia](#) ([a](#)) beschrieben:

5.2.1 Handsteuerung

“Bis Mitte des 20. Jahrhunderts hinein waren Aufzüge überwiegend handgesteuert (Hebelsteuerung). Ein Aufzugführer wurde mit einer Rufanlage auf Fahrgäste aufmerksam, öffnete und schloss die Türen, bediente einen Fahrshalter und fuhr die Stockwerke auf mündlichen Wunsch an.” [Wikipedia (a)]

5.2.2 Sammelsteuerung

“Die Steuerung speichert Außenrufe und Innenkommandos. Diese werden, sofern sie in der aktuellen Fahrtrichtung der Kabine liegen, auf dem Weg nacheinander abgearbeitet. Man unterscheidet zwischen Abwärts- bzw. Aufwärtssammelsteuerung und Vollsammelsteuerung. Letztere setzt zwei Druckknöpfe voraus (Auf- und Absammelnd). Nach Eingabe eines Fahrtwunsches über die Außentableaus (Ruf) oder vom Kabinentableau (Kommando) wird in der Steuerung die Richtungs- auswahl bestimmt (Ruf ober- oder unterhalb des jeweiligen Kabinenstandortes). Bei Aufzugsgruppen können Gruppensammelsteuerungen eingesetzt werden, die ebenfalls als Abwärts-, Aufwärts- oder Vollsammelsteuerung ausgeführt sind.” [Wikipedia (a)]

5.2.3 Druckknopfsteuerung

“Im Gegensatz zur Sammelsteuerung wird jeweils nur ein Kommando oder Ruf gespeichert, wobei Kabinenkommandos Vorrang vor Außenrufen haben. Nach Eingabe eines Kommandos oder Rufes wird dieser gespeichert und blockiert bis zum Ende der Fahrt alle weiteren Eingaben. Die Druckknopfsteuerung wird heute nur noch bei kleineren Wohnhäusern bis zirka 5 Haltestellen eingesetzt.” [Wikipedia (a)]

5.2.4 Zielauswahlsteuerung

“Die Zielrufsteuerung wurde 1997 von Schindler Aufzüge AG Ebikon/CH erfunden. Bei der Zielauswahlsteuerung muss beim Rufen des Aufzugs bereits das Zielstockwerk eingegeben werden. Dies erlaubt der Steuerung ein gezieltes Disponieren und steigert die Kapazität erheblich. Ein System mit Express- und Nahaufzügen kann so flexibler betrieben werden: Der Fahrgast braucht gar nicht mehr zu wissen, welcher Aufzug der Expressaufzug ist, und Einschränkungen wie „Halt nur an jedem 10. Stockwerk“ sind nicht nötig. Zielauswahlsteuerungen ergeben eher in

nichtöffentlichen Gebäuden Sinn, in denen mehrere Aufzüge zu einer sogenannten „Gruppe“ zusammengefasst sind und die Fahrgäste mit der Bedienung vertraut sind, da die Bedienung von der konventionellen Art abweicht. Jeder Aufzug der Gruppe ist mit einer festen Bezeichnung (durchnummeriert oder mit Buchstaben von A-...) versehen. Beim manuellem Eingeben des Zielstockwerks oder mittels Karte (Badge mit fest zugeteiltem Stockwerk bei Geschäftshäusern) in das zentrale Terminal wird dem Fahrgast der entsprechende Aufzug an diesem zentralen Terminal angezeigt. Der Fahrgast begibt sich zu diesem Aufzug. Beim nächsten Öffnen der Türen betritt er den Fahrkorb und wird an sein Zielstockwerk gefahren. In der Kabine befinden sich keine Stockwerkstaster. Das Bedientableau besteht lediglich aus einem Notrufknopf, einem Tür-Auf- sowie gegebenenfalls aus einem Tür-Zu-Taster. Sollte der Fahrgast seinen Fahrtwunsch ändern wollen, muss er erst den bestehenden Fahrtwunsch abfahren und an seinem Ziel einen erneuten Fahrtwunsch an dem zentralen Terminal eingeben. Intelligente Zielauswahlsteuerungen verfügen über eine Rollstuhloption, damit die Steuerung den erhöhten Platzbedarf eines Rollstuhls berücksichtigt und weniger Personen als üblich der nächsten freien Kabine zuteilt. Bei Mehrkabinenaufzügen (zwei Kabinen in einem Schacht) ist diese Steuerung Voraussetzung für den effizienten Einsatz.“ [Wikipedia (a)]

5.3 Beschreibung der realen Anlage

Die Anlage befindet sich in der HAW Hamburg im Gebäude Berliner Tor 7. Sie besteht aus sechs Aufzügen, welche auf die Zielstockwerke aufgeteilt sind, wobei drei ungerade Stockwerknummern anfahren und drei die geraden Stockwerknummern. Jeder Aufzug fährt in das Erdgeschoss und in den ersten Stock, wo die Mensa ist. Die neunte Etage beinhaltet technische Räume, die nur von wenigen Personen betreten werden dürfen. Hier hält nur der Feuerwehraufzug mit Schlüsselbedienung.

Jeweils drei Aufzüge sind nebeneinander. Die anderen drei Aufzüge sind auf der gegenüberliegenden Seite nebeneinander. Die Aufzüge für die ungeraden Stockwerke sind mit A1, B1 und C1 bezeichnet, die für die geraden Stockwerke mit A2, B2 und C2. Die Aufzüge werden im Folgenden mit Aufzuggruppe 1 und Aufzuggruppe 2 bezeichnet, entsprechend den Ziffern in ihren Namen. Jede Aufzuggruppe besitzt eigene Ruftasten. In der Gruppe 2 ist ein Feuerwehraufzug (Aufzug C2) enthalten, der einige Sonderfunktionen erfüllt und alle Stockwerke anfahren kann. Er ist separat in Abschnitt 5.3.1 beschrieben. In den Kabinen sind Sprechanlagen und Alarmfunktionen enthalten sowie digitale Stockwerkanzeigen. An den

Türen außen befinden sich die gleichen Anzeigen und Leuchtmelder, um die Fahrtrichtung des ankommenden Aufzuges anzuzeigen. Ein akustisches Signal kündigt die Ankunft eines Aufzuges in dem Stockwerk an, in welchem der Aufzug gerufen wurde.

5.3.1 Feuerwehraufzug

Der Feuerwehraufzug ist im Normalbetrieb ein Aufzug für gerade Stockwerke in der Gruppe 2. Er erfüllt unter Anwendung eines Schlüssels einige Sonderfunktionen. Der Feuerwehraufzug ist in der Lage, jedes Stockwerk anzufahren. In den ungeraden Stockwerken sind Schlüsselschalter und ein zusätzliches Ruftableau montiert.

5.3.2 Sonstige Funktionen der realen Anlage

Die reale Anlage hat noch weitere Funktionen, wie eine Gewichtsüberwachung, die bei Überlast Alarm auslöst, und weitere Hardwaremeldungen, die in dieser Simulation jedoch nicht weiter berücksichtigt werden.

5.4 Funktionen in dieser Simulation

In dieser Simulation können aus Gründen des Umfangs nicht alle Funktionen der realen Anlage berücksichtigt werden. Die folgenden Funktionen sind nicht implementiert worden:

1. Feuerwehraufzug
2. Sonderfahrt und Außensteuerungen durch Schlüsselschalter
3. Alarmtaster
4. Hardwaremeldungen (z. B. Automatenfall¹, Motorschutzschalter)
5. Sondereinrichtungen für Sicherheit (z.B. Gewichtsmessung, Seilüberwachungen)
6. Alarmfunktion
7. Sprechanlage
8. Akustisches Signal bei Ankunft eines angeforderten Aufzuges

¹Auslösen eines Leitungsschutzschalters

5 *Beispiel einer Aufzugsimulation*

Der Vollständigkeit halber sind die nötigen Taster, Schlüsselschalter und Leuchtmelder für den Feuerwehraufzug und die Alarmtaster in der Übersicht enthalten, aber ohne Funktion². Es werden alle Funktionen simuliert und programmiert, die ein “normaler Benutzer” ohne Schlüssel oder ähnliches nutzen kann.

²Mit Ausnahme der zusätzlichen Ruftableaus in den ungeraden Stockwerken.

5.5 Programm in LabVIEW

5.5.1 Aufbau des Frontpanels



Abbildung 5.1: Frontpanel der Simulation in LabVIEW

Jeder Aufzug ist als Rechteck dargestellt. Die Kabine wird durch einen Schieberegler repräsentiert. Die Stockwerkstasten sind so platziert, dass sie auch zur Stockwerknummerierung dienen. Die Aufzugruftasten sind auf der rechten Seite versetzt platziert. Sie sind aufgeteilt wie in der realen Anlage, die zwischen geraden und ungeraden Stockwerken geteilt ist (siehe 5.3). Die Türen der Aufzüge werden ebenfalls durch Schieberegler repräsentiert. Sie sind in der Übersicht rechts waagerecht angeordnet. Wenn die Tür geschlossen ist, wird der Hintergrund der Schieberegler blau gefärbt. Unterhalb der Türdarstellungen befinden sich Schalter, die die Lichtschranken darstellen. Sie sind als Öffner ausgelegt, um Drahtbruchsicherheit zu gewährleisten.

5.5.2 Beschreibung des Blockdiagrammes

Das Blockdiagramm ist zum Zweck der Übersicht so aufgebaut, dass man mit einem ausreichend großen Bildschirm³ nur in eine Richtung scrollen muss, um sich darin zu bewegen. Ganz oben wird zunächst mit der Methode “Connect” die Verbindung aufgebaut und die Stellung des Betriebsartenschalters abgefragt. Die Stellung wird in einem Feld auf dem Frontpanel angezeigt. Nach unten fortlaufend sind die sechs Aufzüge mit ihren Kabinentastern und Leuchtmeldern angeordnet. Sie werden jeweils zu Clustern zusammengefasst und als Worte übertragen. Auf der rechten Seite des Hauptdiagrammes befinden sich für jeden Aufzug zwei SubVIs. Eines ist für die Tür und ein anderes für die Verschiebung der Kabine und dessen Positionserkennung. Die darin enthaltenen Blockdiagramme werden im Weiteren noch genauer erläutert (siehe unten). Unter den sechs einzelnen Aufzügen sind die Ruftableaus angeordnet. Sie werden zu Demonstrationszwecken wortweise in den Merkerbereich übertragen. Es ist auffällig, dass an allen Lesemethoden am Anschluss “pData” eine nicht bezeichnete Konstante verbunden ist. Diese Konstante ist nötig, damit für die gelesenen Variablen Speicher bereitgestellt wird. Ein Name für diese Konstanten ist nicht von Belang. Der Speicher wird von LabVIEW verwaltet. Der Datentyp der Konstante ist “Variant”, das ist eine Art Pointer auf einen Speicher ohne klassischen Datentyp. Daher müssen die zu lesenden Daten wie die zu schreibenden zunächst entsprechend der Richtung umgewandelt werden. Ein direktes Überführen der Daten von “Variant” in “Cluster” und umgekehrt ist nicht möglich. Es muss zunächst eine Umwandlung in einen klassischen Datentyp wie Integer (I16) erfolgen. Integer kann in ein Boolesches Array umgewandelt werden, um die einzelnen Bits zu lesen oder zu bearbeiten. Die Methoden für die Kommunikation sind alle hintereinander verbunden. Am Ende eines jeden Programm-durchlaufes wird die Verbindung getrennt. Zwischendurch muss nach den Schreiboperationen auf “Continuous Scan” umgeschaltet werden. Das passiert nicht nach jedem Schreibvorgang, sondern einmal gesammelt am Ende des Durchlaufes.

Blockdiagramm der Tür

Die Tür wird durch einen Schieberegler dargestellt, dessen Wert aus einem Integralregler gespeist wird. Es gibt einen Booleschen Wert zum Öffnen der Tür und einen zum Schließen. Es wird entweder ein positiver Eingangswert oder ein negativer auf den Integrator geschaltet. Enthalten beide Bits eine logische “0”, wird der Integrator mit dem Wert 0 gespeist. Der Integrator hat eine untere und eine obere Grenze, wobei die Grenzen den Endlagen der Tür entsprechen und die Integrationszeit T_i die Geschwindigkeit der Türbewegung bestimmt. Es

³Es wurde auf einem 22"-Monitor erstellt.

wird davon ausgegangen, dass beide Türhälften von demselben Motor angetrieben werden und die Umsetzung der Bewegungsrichtung mechanisch erfolgt. Um die Rückmeldungen der Endlagen zu simulieren, wird der Integratorausgang auf Vergleicher verschaltet, die bei den Werten der Endlagen die Bits für die Rückmeldungen an die Schreibmethode übergeben. Die Daten werden an das am Eingang des SubVIs übergebene Peripherieeingangswort übertragen und aus dem Peripherieausgangswort mit der gleichen Nummer gelesen.

Blockdiagramm der Kabine

Die Positionsbestimmung und das Verfahren der Kabine im Schacht ist ähnlich wie die Tür realisiert worden. Es handelt sich hier ebenfalls um Schieberegler, die durch Integratoren angesteuert werden. Es ist eine Schleichfahrt und eine normale Fahrgeschwindigkeit vorgesehen. Es gibt drei Eingangsbits für die Bewegung der jeweiligen Kabine. Eines ist für die Schleichfahrt, eines für die normale Fahrt und eines für die Richtung zuständig. In diesem Plan sind auch die gesamten Positionsschalter simuliert. Die Position wird anhand des Integratorausgangs bestimmt. Mit Vergleichen wird geprüft, ob der Integratorwert in einem von mehreren positionsabhängigen Wertebereichen liegt. Da sich der Integratorausgang zu schnell ändert, ist es nicht möglich, nur mit einem Wert zu vergleichen. Es muss ein Bereich sein, der ausgewertet wird. In einer echten Anlage mit schnell bewegten Teilen ist das ähnlich. Ein Sensor muss mindestens so lange ausgelöst sein, dass die Steuerung ihn sicher erfassen kann. Das bedeutet, er muss immer länger betätigt sein, als die Zykluszeit der Steuerung beträgt. Hier werden für jedes Stockwerk drei Lageschalter simuliert. Einer für den Schleichfahrtbereich in Abwärtsrichtung, die Etagenendlage, und einer für den Schleichfahrtbereich in Aufwärtsrichtung. Der Aufbau ist in Abbildung 5.2 schematisch dargestellt. Der Motor wird über das Peripherieausgangswort angesteuert, welches an das SubVI parametrierbar ist. Die Rückmeldungen für die Etagenendlagen werden durch das Peripherieeingangswort mit der gleichen Nummer an die Steuerung übertragen. Die Schleichfahrtschalter in Abwärtsrichtung werden an das 22. Peripherieeingangswort unterhalb der parametrierbaren Nummer übertragen, die Schleichfahrtschalter in Aufwärtsrichtung an das 24. unterhalb der parametrierbaren Nummer⁴. Die SubVIs mussten für die verschiedenen Aufzuggruppen separat erstellt werden. Bei der Gruppe 2 gelten andere Werte in gleicher Weise.

⁴Wenn an das VI eine 90 parametrierbar ist, führt das zu dem Aufbau:

Motor: PAW90

Endlagen in PEW90

Schleichfahrt aufwärts: PEW66

Schleichfahrt abwärts: PEW68

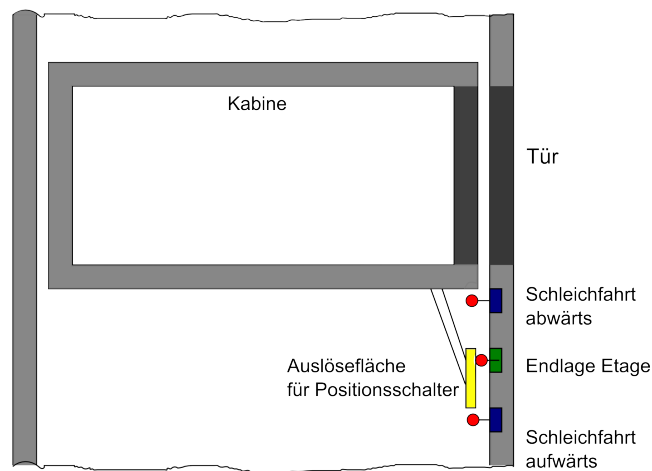


Abbildung 5.2: Aufbau der Sensorik einer Etage

5.6 SPS-Programm

5.6.1 Funktionsbeschreibung

Aufzugruf durch Stockwerkstasten

Bei einem Aufzugruf muss der Fahrgast angeben, in welcher Richtung sein Fahrziel liegt. Ein Aufzugruf in einem Stockwerk, in dem einer der Aufzüge steht, ist nicht möglich. Nach einem Aufzugruf wird anhand der folgenden Kriterien der Aufzug gewählt, dem der Aufzugruf zugeteilt wird:

1. Hat ein Aufzug einen Haltewunsch an der Haltestelle, aus der der Ruf kommt, und bewegt er sich in der gewünschten Richtung?
2. Welcher freie Aufzug liegt in geringster Entfernung?
3. Welcher Aufzug hat die wenigsten Aufträge im Speicher?

Die Kriterien werden von oben nach unten geprüft. Trifft eines nicht zu, wird das nächste überprüft. Sollte nach der Prüfung ein Auftrag noch keinem Aufzug zugewiesen sein, werden die übrigen Aufträge aus dem Speicher an die Aufzüge A und B der jeweiligen Gruppe übergeben. So wird erreicht, dass kein Auftrag unerfüllt bleibt.

Zielwahl durch Wahltasten in der Kabine

Die Zielwahl erfolgt über die Taster in der jeweiligen Kabine. Die gewählten Ziele werden direkt in den entsprechenden Auftragspeicher geschrieben. Wenn der Auftragspeicher leer ist, bestimmt die erste Zielwahl in der Kabine die Fahrtrichtung. Die Fahrtrichtungsfestlegung spielt in der Auftragsabwicklung eine Rolle (siehe 5.6.1).

Auftragsabwicklung

Vor der Abfahrt⁵ wird das nächste Ziel gewählt. Die Tür ist ab dem ersten Druck auf einen Wahltaster oder bei einem Zwischenhalt nach dem Öffnen der Tür eine vorgegebene Zeit geöffnet. Danach wird das Ziel gewählt und abgefahren. Dafür wird der Auftragspeicher von der Position aus, der Fahrtrichtung entsprechend, durchsucht⁶. Sobald eine logische "1" gefunden ist, wird das Ziel festgelegt und abgefahren. Wird nach mehreren Halten in der gewählten Fahrtrichtung kein Ziel mehr gefunden, wird in die andere Richtung von der jeweiligen Ausgangsposition gesucht. Bei Erreichen eines Zieles wird dieses aus dem Auftragspeicher gelöscht. Dieses Verfahren ergibt eine Einschränkung in der Funktion und zwar wird bei der Auftragsabwicklung davon ausgegangen, dass ein Fahrgast sein Ziel wählt, sobald er den Aufzug betritt, also vor dem Schließen der Tür. Fahrtwünsche, die nach der Abfahrt getätigt werden, können nur auf der Strecke berücksichtigt werden, wenn sie oberhalb des gerade festgelegten Zieles liegen. Sonst wird erst bis zum letzten Ziel in Fahrtrichtung gefahren, bevor das zu spät gewählte Ziel angefahren wird⁷.

Sicherheitsfunktionen

Wird die Lichtschranke ausgelöst oder der "Tür Öffnen"-Taster während der Türschließung betätigt, öffnet die Tür wieder. Ein erneutes Schließen der Tür wird nach einer definierten Wartezeit gestartet. Während der Fahrt lässt sich die Tür nicht öffnen.

5.6.2 Programmbeschreibung

Das SPS-Programm ist für Steuerungen der Firma Siemens erstellt worden. Es wurde für die Soft-SPS PLCsim erstellt, daher ist zu beachten, dass keine Hardwareanpassungen er-

⁵Die Abfahrt beginnt mit dem Schließen der Tür.

⁶Befindet sich der Aufzug im 1. Stockwerk und es ist die Aufwärtsrichtung durch einen Fahrtwunsch in den 7. Stock vorgewählt, wird der Auftragspeicher von Bit 1 an nach links laufend durchsucht. Es wird an Bit 4 eine logische "1" gefunden. Das ist das Ziel 7. Stock.

⁷Wählt ein Fahrgast während der Fahrt in den 7. Stock sein Ziel im 5. Stock würde erst der 7. Stock angefahren und dann der 5. Wählt er den 11. Stock, dann wird der Fahrauftrag nach dem 7. Stock wie üblich angefahren.

folgt sind. Das Programm ist ohne Hardwaredaten in die Soft-SPS zu laden, da es sonst zu Peripheriezugriffsfehlern kommen kann.

Auftragsspeicher und Datenstruktur

Der Auftragsspeicher besteht für jeden Aufzug aus einem Datenwort⁸. Der Aufbau des Speicherwortes ist in Tabelle 5.1 dargestellt. Die Auftragsspeicher sind alle in einem globalen DB hinterlegt, damit alle Bausteine auf die Daten zugreifen können.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Ger. Stockw.										13	11	7	5	3	1	EG
Ung. Stockw.								14	12	10	8	6	4	2	1	EG

Tabelle 5.1: Aufbau des Auftragsspeichers

Der Speicher ist in der gleichen Weise aufgebaut wie die Peripherieeingangsworte, die Stockwerkendlagen und Schleichfahrtpositionen enthalten. So können Vergleichsoperationen praktisch genutzt werden. Zusätzlich gibt es einen Speicher für die Aufträge, die in der Kabine aufgegeben wurden und die Aufzugrufe von außen in Aufwärts- bzw. Abwärtsrichtung. Diese Speicher werden auch direkt durch die Leuchtmelder der jeweiligen Taster ausgegeben.

Bausteinaufteilung im Programm

Das Programm ist in verschiedene Bausteine aufgeteilt. Die Aufzüge für die geraden und die ungeraden Stockwerke werden in den meisten Fällen getrennt behandelt. Die Programme in den Bausteinen sind zwar nahezu gleich, aber es sind wiederum viele Parameter verschieden, besonders bei den Maskierungen der Eingangsworte. Zudem werden in beiden Bausteinen auch unterschiedliche Bausteine, die ebenfalls für die Aufzuggruppen getrennt erstellt wurden, aufgerufen. Ein Programm, welches alle Aufzüge über den gleichen parametrierbaren Baustein bearbeiten würde, wäre erheblich unübersichtlicher. Die Bausteinaufrufe sind in Abb. 5.3 dargestellt.

⁸16 Bit, jedes repräsentiert ein Stockwerk

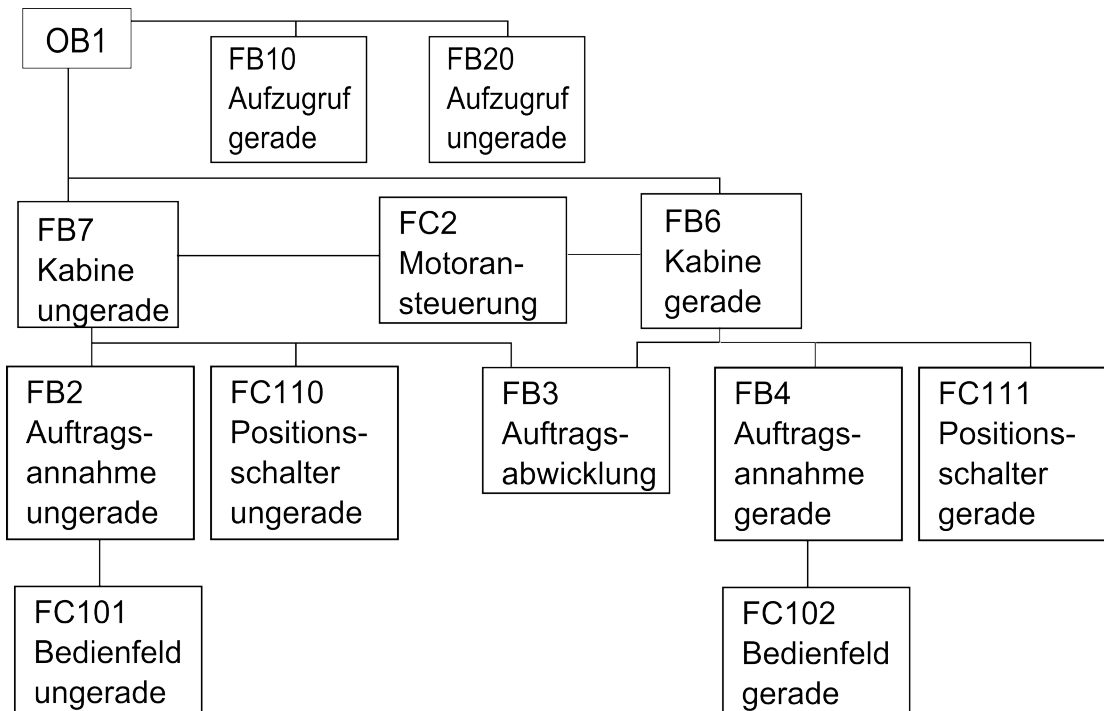


Abbildung 5.3: Programmstruktur mit Bausteinaufrufen

Die Bausteine für die Motoransteuerung und die Auftragsabwicklung sind identisch für beide Gruppen, daher mussten sie nicht getrennt für die Gruppen angelegt werden und können von beiden aufgerufen werden.

Positionsfeststellung

Die Positionsschalter sind wie in Abb. 5.2 gezeigt in jeder Etage angeordnet. Befindet sich der Aufzug zwischen zwei Etagen, ist kein Positionsschalter aktiv. Daher wird die Position zwischengespeichert und von der Funktion auch als Integer-Wert ausgegeben. Dieser dient für die Positionsanzeige in der Kabine und ist für Berechnungen sehr hilfreich.

Außenruf eines Aufzugs

Zunächst müssen die Eingangsworte der jeweiligen Ruftasten verändert werden, um an die Speicherstruktur (siehe Tab. 5.1) angepasst zu werden. Zudem wird die Etage als Integerwert zwischengespeichert. Die Position jedes Aufzuges der Gruppe wird ebenfalls als Integerwert gespeichert, um damit den minimal entfernten Aufzug zu berechnen. Die Zuweisung des Auftrags an den bestmöglichen Aufzug passiert bereits, während der Ruftaster betätigt wird,

nach dem in der Funktionsbeschreibung (siehe 5.6.1) erläuterten Verfahren. Wird eine Etage von dem gerufenen Aufzug erreicht und die Tür geöffnet, wird der Auftrag gelöscht. Besonders wichtig ist hier die Unterscheidung zwischen dem Aufwärtsruf und dem Abwärtsruf. Möchten mehrere Fahrgäste in einem Stockwerk in unterschiedliche Richtungen fahren, sind zwei Aufzüge nötig. Daher wurde mit sechs Booleschen Variablen die jeweilige Richtungszuweisung eines Aufzuges gespeichert. Ein Aufzug kann nicht in beide Richtungen gleichzeitig angefordert werden. Sind nach den Auftragszuweisungen noch Aufträge in keinem der einzelnen Auftragspeicher enthalten, werden Sie in Abwärtsrichtung dem Aufzug A und in Aufwärtsrichtung dem Aufzug B der jeweiligen Gruppe zugeordnet. So geht kein Auftrag verloren.

Aufnahme von Kabinenaufträgen

Kabinenaufträge werden sofort in den Auftragspeicher geschrieben. Es wird durch eine Oder-Verknüpfung zwischen dem Auftragspeicher und einer Maskierung für das jeweilige Stockwerk das Bit gesetzt, welches das jeweilige Stockwerk als Ziel definiert (siehe 5.1). Das geschieht nicht, wenn der Aufzug sich bereits in der als Ziel gewählten Etage befindet. Der erste Kabinenauftrag legt die Fahrtrichtung fest, in die zuerst gefahren wird. Liegt das zuerst gewählte Ziel oberhalb der Aufzugposition, wird die Fahrtrichtung aufwärts festgelegt. Wenn die Kabine zuvor mit dem Richtungswunsch abwärts gerufen wurde, wird trotzdem aufwärts gefahren, da es keinen anderen Fahrauftrag gibt.

Auftragsabwicklung und Zielfestlegung

Das Vorgehen in der Auftragswahl ist bereits in der Funktionsbeschreibung kurz erläutert. Es wird zunächst eine Suchvariable mit dem Wert der aktuellen Position initialisiert. Danach wird zur Aufwärts- oder zur Abwärtssuche gesprungen. Die Suchvorgänge sind prinzipiell identisch aufgebaut. Die Bitfolge der Suchvariable wird immer um eine Stelle in die entsprechende Richtung verschoben. Nach der Verschiebung wird sie mit dem Auftragspeicher UND-verknüpft. Ist das Ergebnis ungleich Null, wurde das nächste Ziel gefunden und die Suche wird beendet. Dieser Vorgang wird mehrmals wiederholt, wenn kein Ziel gefunden wurde. Hat die Suchvariable das letzte Bit im Wort erreicht, ist in der Suchrichtung kein Ziel vorhanden. Dann wird noch einmal in der Gegenrichtung gesucht.

Verfahren des Aufzuges

Der Baustein "Motoransteuerung" steuert das Verfahren des Aufzuges wobei ein Zielstockwerk und die Aktuelle Position an ihn übergeben wird, und der Aufzug in das Zielstockwerk gefahren wird. Die Türen werden geschlossen, es erfolgt die Abfahrt. Durch Vergleiche wird festgelegt, ob das Ziel größer oder kleiner als die aktuelle Position des Aufzuges ist, und danach wird die Fahrtrichtung bestimmt. Es wird entweder das Bit für die Fahrtrichtung gesetzt, dann wird aufwärts gefahren, oder nicht, dann wird abwärts gefahren. Sobald die in Fahrtrichtung liegenden Schleichfahrtpositionen des Zielstockwerks erreicht werden, also der Wert des Eingangswortes der Schleichfahrtschalter gleich dem Zielstockwerk ist, wird der Aufzug in Schleichfahrt geschaltet, bis die Positionsschalter des Zielstockwerks erreicht sind.

Koordination einer Kabine

Die Koordination der genannten Funktionen übernimmt der jeweilige Kabinenbaustein. Darin werden auch Aufträge, die abgearbeitet wurden, gelöscht und Notöffnungen der Tür durchgeführt. In ihm werden auch die Wartezeiten durch einen Timer gesteuert und die Leuchtmelder in der Kabine angesteuert.

6 Zusammenfassung und Ausblick

Es wurden drei verschiedene Schnittstellen untersucht: ActiveX, OPC und TCP/IP. Sie wurden mit der LabVIEW Basissoftware erstellt. Es wurden keine Zusatzpakete verwendet. Dafür wurden teilweise bestehende Lösungen aus Foren und Communities genutzt.

6.1 Ergebnisse

Die Analyse zeigt, dass die ActiveX-Schnittstelle bei Benutzerfreundlichkeit, Stabilität und Reaktivität am besten abschneidet. Sie ist zwar im Gegensatz zu den beiden anderen Schnittstellen nicht netzwerkfähig, jedoch ist das auch nur selten erforderlich. Darüber hinaus können nicht alle, aber die wichtigsten physikalischen Speicher beschrieben und gelesen werden. OPC ist die Schnittstelle, die am aufwendigsten zu projektieren ist und am schlechtesten in der Gesamtwertung abschneidet. Im Benchmarktest kann die TCP/IP-Verbindung nicht repräsentativ getestet werden, da es in der genutzten Bibliothek Einschränkungen in der Telegrammlänge gibt, sie hätte sonst vermutlich am besten abgeschnitten. Das zeigt die auffällig schnellere Lesezeit mit größeren Telegrammlängen. Aus diesen Gründen wird für die Aufzugsimulation die ActiveX-Verbindung gewählt.

6.2 Ausblick

6.2.1 Mögliche Erweiterungen der Schnittstelle

Das Zeitverhalten der TCP/IP-Verbindung kann durch Anpassung des ProfiNet-Telegramms verbessert werden. Dafür ist es ratsam, die Implementierung des Telegrammes mit einem Skriptbaustein durchzuführen. Dafür ist jedoch ein kostenpflichtiges Zusatzpaket für LabVIEW nötig. Alternativ kann auch von National Instruments eine spezielle ProfiNet-Implementierung erworben werden. Bisher kann nur ein Byte Daten je Telegramm versandt werden. Es sollte möglich sein, bis 222 Byte in einem Telegramm zu senden. Diese Funktion würde die Übertragungsdauer von 1000 Datenworten deutlich verbessern. Dann wäre diese Schnittstelle

voraussichtlich mehrfach schneller als die ActiveX-Schnittstelle. Die OPC-Verbindung könnte mit einer anderen Serversoftware eventuell auch auf einem Rechner funktionieren.

6.2.2 Mögliche Erweiterungen der Aufzugsimulation

Die Aufzugsimulation ist bereits so erstellt worden, dass viele Sensoren und Aktoren für eine Erweiterung um die Sonderfunktionen der realen Anlage bereits in der Simulation vorhanden sind. Das SPS-Programm kann entsprechend erweitert werden. Diese Erweiterung würde hauptsächlich den Feuerwehraufzug umfassen. Ebenfalls kann die Simulation noch um Hardwarefunktionen wie Überlastmessungen und andere Meldungen erweitert werden, die in dieser Simulation noch nicht enthalten sind. Die Auftragsabwicklung könnte durch die Implementierung einer dynamischen Zielfestlegung verbessert werden. Dadurch wäre die Funktionseinschränkung, die in Abschnitt 5.6.1 beschrieben wurde, beseitigt.

6.3 Programmieren im TIA-Portal - Erfahrungen mit der neuen Software

Das neue TIA-Portal hat durchaus an manchen Stellen an Übersicht und guten Funktionen gewonnen. Sehr praktisch und in der bisherigen SIMATIC-Software¹ von vielen vermisst, gibt es jetzt Programmfunktionen ähnlich einer Tabellenkalkulation. Wenn der Benutzer mehrere Elemente nummeriert einfügen möchte, kann er in der Spalte unten links mit der Maus einen "Anfasser" benutzen, um den Text nach unten zu kopieren². Die Funktion ist jedoch gelegentlich noch fehlerhaft. Besonders das Einfügen großer Datenmengen scheint teilweise problematisch. Große Datenmengen sind generell noch ein Problem mit dem Programm. Es läuft noch nicht sehr stabil. Die Programmeditoren in Anweisungsliste (AWL) und Structured Control Language (SCL) haben an Übersicht gewonnen. Auch die Online-Beobachtungsfunktion im SCL-Editor ist jetzt gut realisiert und übersichtlich. Beim Übersetzen eines AWL-Programmes kam es im Laufe der Programmerstellung häufig zu dem Fehler: "Die Referenzinformationen sind inkonsistent. Der Baustein kann nicht übersetzt werden." Dabei handelt es sich vermutlich um einen Fehler in dem Compiler. Es kommt zu diesem Fehler nach manchen Schnittstellenänderungen, die das Datenkonstrukt des Bausteins im Speicher inkonsistent werden lassen. Für den Benutzer ist nicht ersichtlich, warum es dazu kommt. Der inkonsistente Baustein muss gelöscht und komplett neu angelegt werden. Wird in SCL in einer IF-Anweisung ein Baustein aufgerufen und die IF-Anweisung verändert, kommt es gelegentlich dazu, dass der Bausteinaufruf rot eingefärbt

¹SIMATIC-Manager mit Unterprogrammen

²Das ist auch eine sehr bekannte Funktion aus Microsoft Excel.

wird. Es erscheint dann der Fehler, die übergebenen Parameter seien nicht kompatibel mit ihren Datentypen der Bausteinschnittstelle. Löscht man den Aufruf und erstellt ihn mit den gleichen Parametern neu, ist die Funktion wieder gegeben. Auch das scheint ein Fehler in der Software zu sein. Das TIA-Portal ist eine gute Idee, aber es enthält noch viele Fehler. Viele Firmen werden den Umstieg wohl noch weiter hinauszögern.

6.4 Erfahrungen mit LabVIEW

LabVIEW hat eine einfache Programmierumgebung zur Verfügung gestellt. Die Übersicht ist jedoch durch die mangelhafte automatische Linienanordnung schnell verloren gegangen. Eine Funktion, die Verknüpfungslinien beim Anklicken einfärben oder blinken ließe, würde die Arbeit erleichtern. Das Löschen von Verbindungslinien geschieht immer nur stückweise, was auch schnell lästig wird. Für die Programmierung von Übertragungsprotokollen ist LabVIEW nicht zu empfehlen, da der komplexe Aufbau die Möglichkeiten der Software sehr stark ausreizt. Die Programme werden sehr schnell unübersichtlich. Kann man wie in dem Beispiel eine fertige Schnittstelle importieren, ist die Arbeit mit LabVIEW gut möglich, solange das Programm nicht zu groß wird. Schon in dieser nicht sehr umfangreichen Aufzugsimulation ist das Programm für außenstehende Programmierer nur schwer nachvollziehbar. Ein Vorteil von LabVIEW besteht in den vielen bereits projektierten Lösungen, die im Internet meistens öffentlich zugänglich sind.

Literaturverzeichnis

- [Bengel 2000] BENGEL, Günther: *Verteilte Systeme - Client Server Computing für Studenten und Praktiker*. Vieweg, 2000
- [Berger 2011] BERGER, Hans ; AG, Siemens (Hrsg.): *Automatisieren mit STEP 7 in AWL und SCL*. 7. Auflage. Publicis Publishing, 2011
- [Bettermann 2002] BETTERMANN, Thomas: *Anwendung von Microsoft Softwarestandards in der Automatisierungstechnik*. expert Verlag, 2002
- [Community] COMMUNITY, National I.: *S7 PLC TCP/IP Protocol Reference Example*. – URL <https://decibel.ni.com/content/docs/DOC-5467>. – Zugriffsdatum: 12.11.2012, 13.06 Uhr
- [Emmerich 1999] EMMERICH, Wolfgang: *Engineering Distributed Objects*. Wiley, 1999
- [Forum] FORUM, LabView: *I-Regler in LabView*. – URL <http://www.labviewforum.de/Thread-Reiner-I-Regler-in-LabVIEW>. – Zugriffsdatum: 17.12.2012, 15.34 Uhr
- [Georgi und Metin 2007] GEORGI, Wolfgang ; METIN, Ergun: *Einführung in LabView*. 3. Auflage. Hanser, 2007
- [Haase 2008] HAASE, Oliver: *Kommunikation in Verteilten Anwendungen*. 2. Auflage. Oldenburg Verlag, 2008
- [Jomaa 2000] JOMAA, Hichem: *Untersuchung zur Interoperabilität unterschiedlicher OPC-Clients und Server*, HAW Hamburg, Diplomarbeit, 2000
- [Lange 2002] LANGE, Frank I.: *OPC - Grundlagen, Implementierung und Anwendung*. 2. Auflage. Hüthig, 2002
- [National-Instruments a] NATIONAL-INSTRUMENTS: *LabVIEW Internetpräsenz*. – URL www.ni.com/labview/d/. – Zugriffsdatum: 08.11.2012, 11.23 Uhr

- [National-Instruments b] NATIONAL-INSTRUMENTS: *NI LabView Produktübersicht*. – URL <http://www.ni.com/labview/products/d/>. – Zugriffsdatum: 05.02.2013, 11.45 Uhr
- [Schreiner 2009] SCHREINER, Rüdiger: *Computernetzwerke*. 3. Auflage. Hanser, 2009
- [Siemens 2007] SIEMENS: *SIMATIC S7ProSim V5.4 COM-Objekt*, Januar 2007
- [Siemens 2013] SIEMENS: *TIA-Portal Produktseite*. Februar 2013. – URL <http://www.industry.siemens.com/topics/global/de/tia-portal/seiten/default.aspx>. – Zugriffsdatum: 15.02.2013, 11.37 Uhr
- [Wellenreuther und Zastrow 2005] WELLENREUTHER, Günter ; ZASTROW, Dieter: *Automatisieren mit SPS - Theorie und Praxis*. 3. Auflage. Vieweg, 2005
- [Wikipedia a] WIKIPEDIA: *Aufzugsanlage*. – URL <http://de.wikipedia.org/wiki/Aufzugsanlage>. – Zugriffsdatum: 01.01.2013, 20.52 Uhr
- [Wikipedia b] WIKIPEDIA: *Interprozesskommunikation*. – URL <http://de.wikipedia.org/wiki/Interprozesskommunikation>. – Zugriffsdatum: 30.01.2013, 08.15 Uhr
- [Wikipedia c] WIKIPEDIA: *Supervisory Control and Data Acquisition*. – URL http://de.wikipedia.org/wiki/Supervisory_Control_and_Data_Acquisition. – Zugriffsdatum: 01.01.2013, 20.52 Uhr

Anhang

Der Anhang dieser Ausarbeitung ist bei Prof. Dr.-Ing. Ulfert Meiners und Prof. Dr.-Ing. Jochen Maaß hinterlegt und besteht aus einem Datenträger mit den folgenden Inhalten:

1. Testprogramme aus der Analyse für den Test der Übertragungsdauer
 - LabVIEW-Programme
 - SPS-Programme
2. Aufzugsimulation
 - LabVIEW-Programm
 - SPS-Programm
3. Software NetToPLCsim
4. Dokumentation S7ProSim

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §16(5) APSO-TI-BM ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen habe ich unter Angabe der Quellen kenntlich gemacht.

Hamburg, 19.02.2013

 Tobias Schönrock