

Bachelor Thesis
zur Erlangung des Akademischen Grades
Bachelor of Engineering

Flexible Programmierung einer
Bürogebäudesteuerung mit einer Beckhoff-SPS
und der Verwendung von TwinCat 3

Michael Oldenburg

Matrikel Nr.7079267

Erstprüfer: Prof. Dr. Bernd Aschendorf

Zweitprüfer: Dipl. Ing. Sandra Stahlberg

Abgabedatum

Inhaltsverzeichnis

1. Eidesstattliche Erklärung	4
2. Abstract	5
3. Aufgabenstellung	6
4. Gebäudeautomation	7
5. Das Unternehmen Beckhoff	8
5.1 Gebäudeautomation mit Beckhoff-SPS	10
5.2 Demo-Koffer-Beckhoff	10
5.3 Module	11
6. Pflichtenheft	18
6.1 Zielbestimmungen	18
6.1.1 Musskriterien	18
6.1.1.1 Büro-Seminar-Raum-Lichtsteuerung	18
6.1.1.2 Trennwandsteuerung	18
6.1.1.3 Heiz-und Kühlsteuerung	18
6.1.1.4 Flur-Lichtschtaltung	19
6.1.2 Kannkriterien	19
6.1.2.2 Zentralsteuerung	19
6.2 Produkteinsätze	19
6.2.1 Anwendungsbereiche	19
6.2.2 Zielgruppen	19
6.3 Produktübersichten	19
6.3.1 Pro Büroraum	19
6.3.2 Flur	19
6.4 Produktfunktionen	20
6.4.1 Lichtstromkreise	20
6.4.1.1 Lichtstromkreise bei geschlossener Trennwand	20
6.4.2.1 Lichtstromkreis 1 Linkes Büro	20
6.4.2.2 Lichtstromkreis 2 Linkes Büro	20
6.4.1.2 Lichtstromkreise bei geöffneter Trennwand	20
6.4.2 Außen-Jalousiesteuerung	21
6.4.2.1 Außen-Jalousiesteuerung	21
6.4.3 Innen-Jalousiesteuerung	21
6.4.3.1 Innen-Jalousiesteuerungen bei geschl. Trennwand	21
6.4.4 Flurschtaltung	21
6.4.5 Trennwand	21
6.4.6 Zentralsteuerung	21
6.5 Technische Produktbetrachtungen	22

6.5.1 Software	22
6.5.1.1 Software-Informationen	22
7. Die Software TwinCat 3.....	23
7.1 Installation der Software TwinCat 3	25
7.2 Programmieren mit TwinCat 3.....	26
7.3 Konfiguration des CX9020	40
7.4 Realisierung der Jalousiefunktion	44
7.5 Realisierung der Heizungssteuerung	46
7.6 Realisierung der Flurlichtsteuerung	47
7.7 Anlegen globaler Variablen.....	48
7.8 TwinCat 3 Lizenz	49
7.9 Bootprojekt erstellen	50
8. Vergleich TwinCat 2 mit TwinCat 3.....	51
9. Aussichten	51
10 Fazit.....	52
11. Anhang	53
11.1 Literaturverzeichnis	53
11.2 Abbildungsverzeichnis.....	54
11.3 Tabellenverzeichnis	55

1.Eidesstattliche Erklärung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbständig und ohne die Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Schriften entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung noch nicht vorgelegt worden.

Bergkamen,

Unterschrift

2. Abstract

The theme of this thesis is:

„Flexible programming of an office building control with a Beckhoff PLC and the use of TwinCat 3“

Based on an exemplary programming of a seminar offices is shown that a flexible programming with a Beckhoff PLC is possible.

To edit the thesis the following therms are used:

- Basic CPU modul CX9020
- TwinCat in version 3.0
- Demo suitcase Beckhoff

In additon both versions TwinCat 2 and 3 will be compared.

3. Aufgabenstellung

Das Thema dieser Thesis lautet:

„Flexible Programmierung einer Bürogebäudesteuerung mit einer Beckhoff-SPS und der Verwendung von TwinCat 3“

Anhand einer exemplarischen Programmierung eines Seminar-Büros wird gezeigt, dass eine flexible Programmierung mit einer Beckhoff-SPS möglich ist.

Für die Bearbeitung wird folgendes verwendet:

- CPU Grundmodul CX9020
- TwinCat in der Version 3.0
- Demo-Koffer-Beckhoff

Zudem werden die beiden Versionen TwinCat 2 und 3 verglichen.

4. Gebäudeautomation

Heutzutage findet die Gebäudeautomation Einzug in Wohnhäuser aber auch in Bürogebäuden. Die Vorteile liegen auf der Hand. Im Wohnhaus ist es das Ziel, einen erhöhten Komfort durch intelligente Steuerung zu erreichen. Als gutes Beispiel dient hier die Jalousiesteuerung. Vollautomatisiert braucht der Hausbesitzer sich damit nicht mehr beschäftigen bis auf die Entscheidung, zu welcher Uhrzeit die Jalousien Hoch- oder Runterfahren. Anforderungen an die Gebäudeautomation stellen auch die Punkte Flexibilität, Sicherheit, Erweiterbarkeit, Kommunikation und die Funktionssicherheit.

In Bürogebäuden ist ein erhöhter Komfort auch nicht von der Hand zu weisen. Dagegen liegt der Fokus in großen Bürogebäuden auf Energieeinsparung, Flexibilität und Erweiterbarkeit. Über Zentralfunktionen lassen sich Stromkreise im gesamten Gebäude ausschalten. Sensoren erfassen die Anwesenheit von Personen und es wird die gewünschte Funktion geschaltet. Die Heizungssteuerung reagiert über Temperaturerfassung in jedem Büro, somit lässt sich ein ungewünschtes Aufheizen vermeiden. Eine sensorgesteuerte Abschottung realisiert ein Fahren der Außenjalousien je nach Sonnenstand in die gewünschte Position.

Es ist sowohl bei Wohn- und Bürogebäuden mit Hilfe einer Gebäudeautomation möglich, ein „smart Metering“ durchzuführen. Somit kann ermittelt werden, wo ein erhöhter Energiebedarf vorhanden ist und wie man diesen durch weitere Maßnahmen reduzieren kann.

Mit einer herkömmlichen Elektroinstallation kommt man schnell an die Grenze des Möglichen und muss hier auf eine andere Art von Elektroinstallation zurückgreifen. In den vergangenen Jahren haben sehr viele Hersteller sich mit dem Thema Gebäudeautomation beschäftigt und ihre Systeme auf den Markt gebracht. Es ist daher wichtig bei der Planung, ein System zu wählen, welches die gewünschten Funktionen abdeckt und auch unter dem Punkt Erweiterbarkeit keine großen Einschränkungen bietet.

5. Das Unternehmen Beckhoff

Das Unternehmen Beckhoff wurde im Jahre 1980 von Hans Beckhoff ins Leben gerufen. Die Firmenzentrale befindet sich in Verl im Kreis Gütersloh.

In der Elektroindustriebranche hat sich Beckhoff mit 2800 Mitarbeitern und einen Umsatz im Jahre 2014 von 510 Mio. Euro einen Namen gemacht. Allein in Deutschland gibt es 11 Niederlassungen und 33 Tochterfirmen repräsentieren den Namen Beckhoff rund um die Welt. Die Produktpalette von Beckhoff lässt sich wie folgt unterteilen:[BP]

Der IPC-Bereich

Skalierbare Industrie-PC-Lösungen

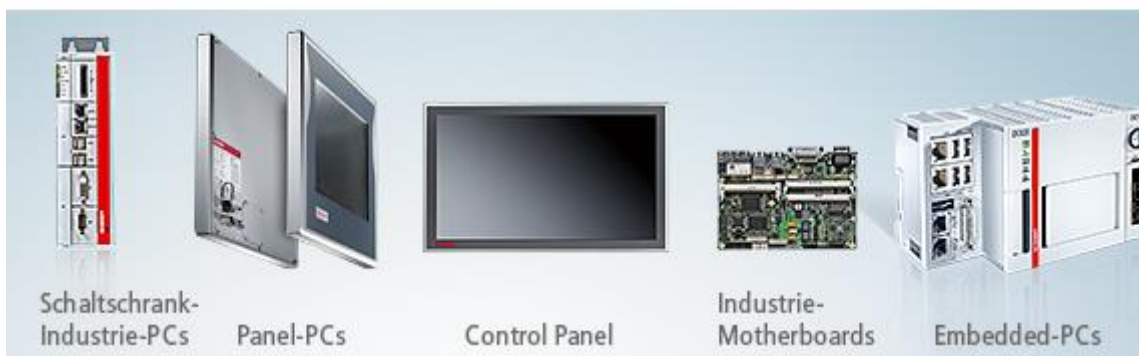


Abbildung 1: IPC-Bereich

Hier bietet Beckhoff ein breites Spektrum an Industrie-PCs und Embedded-PCs für sämtliche Anforderungen der Industrie über Schaltschrankbau, Control Panels und Hutschienenlösungen. Industrie-Motherboards werden nach Wünschen des Kunden selbst im Hause entwickelt und geliefert. [BP]

Der I/O-Bereich

Busklemmen – Der universelle Grundbaustein für die Automatisierung



Abbildung 2: Der I/O Bereich

Ein wichtiger Punkt ist die Flexibilität und das erreicht Beckhoff mit einem umfangreichen Angebot an Feldbuskomponenten. Somit ist für jede Aufgabe ein perfekt geeignetes Bussystem verfügbar Lon, EnOcean, EIB sind nur einige Systeme, die verarbeitet werden können, da die Liste an möglichen Systemen bei weitem viel länger ist. [BP]

Der Motion-Bereich



Abbildung 3: Der Motion-Bereich

Auch in der Antriebstechnik stellt Beckhoff Lösungen bereit. Da immer mehr Fertigungsstrecken automatisiert werden, um schneller und besser die gewünschten Anforderungen zu bewerkstelligen, stellt Beckhoff zahlreiche Komponenten, um diese zu erreichen. Servoverstärker für schnelle Positionierung oder starke Servomotoren sind nur 2 Beispiele aus dem umfangreichen Angebot an Hardware. [BP]

5.1 Gebäudeautomation mit Beckhoff-SPS

Das Unternehmen Beckhoff bietet mit unterschiedlichen Ein- und Ausgangsklemmen sowie verschiedenen Controllern eine hohe Bandbreite an Möglichkeiten für eine solide Gebäudeautomation. Nach Entscheidung für ein Grundmodul und einen Bus-Typ (K-Bus oder E-Bus) bietet sich ein Sortiment von über 400 verschiedenen Bus-Klemmen an, um das gewünschte Projekt mit den verbundenen Funktionen zu realisieren. Im Rahmen dieser Bachelorarbeit ist es natürlich nicht möglich, ein komplettes Bürogebäude mit allen Funktionen zu programmieren. Es werden daher einige Funktionen mit Hilfe des Demo-Koffers-Beckhoff von der FH-Dortmund realisiert.

5.2 Demo-Koffer-Beckhoff



Abbildung 4: Demo-Koffer-Beckhoff

5.3 Module



Abbildung 5: CPU Grundmodul CX9020

Das Herzstück des Systems bildet das CPU Grundmodul CX9020.

Der CX9020 ist eine kompakte, hutschienenmontierbare Ethernet-Steuerung mit 1-GHz-ARM-Cortex™-A8-CPU. Der Anschluss für die Beckhoff-I/O-Systeme ist direkt im CPU-Modul integriert. Das Gerät verfügt über eine automatische Erkennung des jeweiligen Bussystems (K-Bus oder E-Bus) und schaltet in den entsprechenden Modus. Der CX9020 umfasst die CPU mit zwei MicroSD-Kartenslots, den internen Arbeitsspeicher (RAM) und 128 kB NOVRAM als nicht-flüchtigen Speicher. Zwei geschaltete Ethernet-RJ45-, vier USB-2.0- sowie eine DVI-D-Schnittstelle gehören ebenfalls zur Basisausstattung. Die RJ45-Schnittstellen sind auf einen internen Switch geführt und bieten eine einfache Möglichkeit zum Aufbau einer Linientopologie ohne den zusätzlichen Einsatz von Ethernet-Switchen. Das Betriebssystem ist Microsoft Windows Embedded Compact 7. Durch die Automatisierungssoftware TwinCAT 3 wird das CX9020-System zu einer leistungsfähigen SPS und Motion-Control-Steuerung, die mit oder ohne Visualisierung eingesetzt werden kann. Optional kann das Gerät mit einer Feldbus-, einer seriellen oder einer Audioschnittstelle bestellt werden. [CX9020]

An dieses Grundmodul werden die Bus-Klemmen angesteckt und sind über eine seitlich angebrachte Kontaktfläche mit dem Bus verbunden. Folgende Klemmen sind im Demo-Koffer-Beckhoff verbaut.

Digitale Eingangsklemme

Digital-Eingang KL1104, KL1114

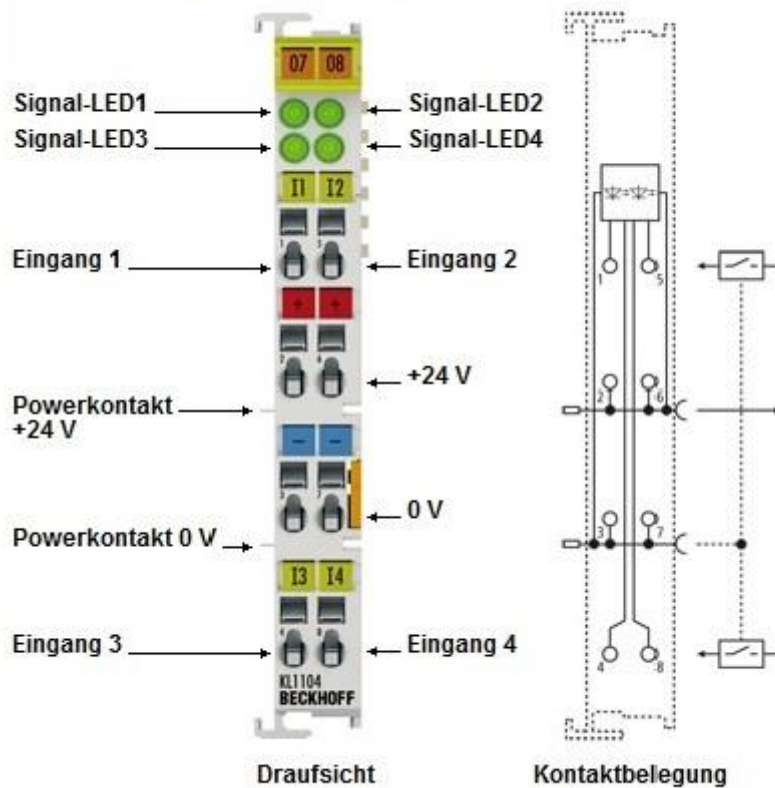


Abbildung 6 Digital-Eingang KL 1104

Die digitalen Eingangsklemmen KL1104 erfasst die binären Steuersignale aus der Prozessebene und transportiert sie galvanisch getrennt zum übergeordneten Automatisierungsgerät. Die Busklemme besitzt vier Kanäle, deren Signalzustand durch Leuchtdioden angezeigt wird. Wie bei dem Grundmodul wird diese auf die Hutschiene montiert. [BP]

Analoge Eingangsklemme

Analog-Eingang KL3454, KL3458

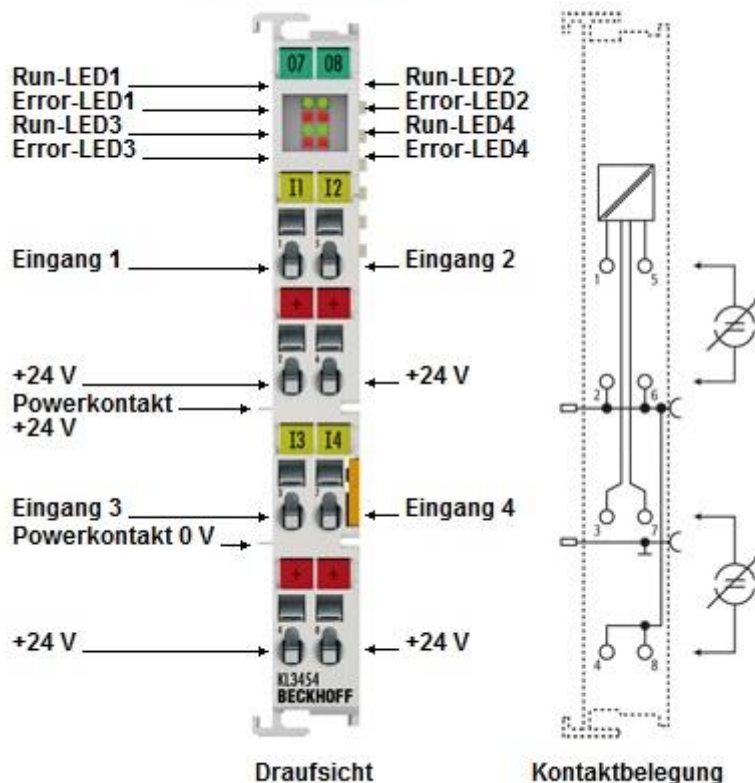
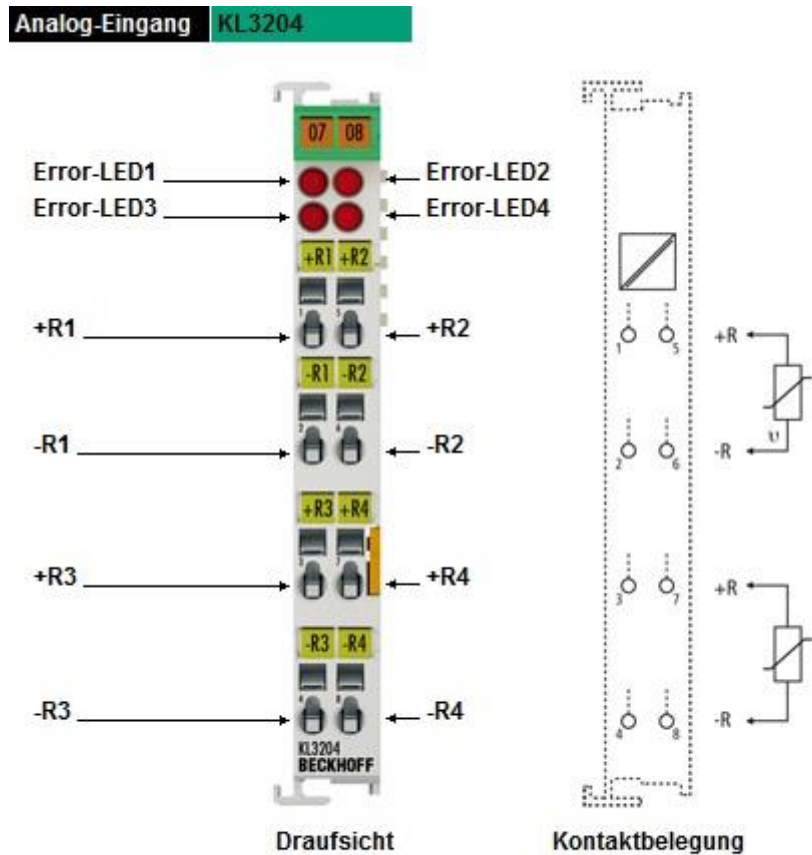


Abbildung 7 Analog-Eingang KL 3454

Die analoge Eingangsklemme kann Signale im Bereich von 4 bis 20 mA verarbeiten. Der Strom wird mit einer Auflösung von 12Bit digitalisiert, galvanisch getrennt und zum Automatisierungsgerät transportiert. Bei dieser Klemme sind die 4 Eingänge in 2-Leitertechnik ausgeführt und haben einen gemeinsamen Massepunkt. Die 24V Versorgung wird auf die Klemme geführt und ermöglicht so einen Anschluss von nicht fremdversorgten Sensoren. Die Powerkontakte sind durchverbunden und die Bezugsmasse der Eingänge ist die 0-Volt-Anschlussklemme. Eine Überlast wird erkannt und der Klemmenstatus wird übermittelt. Durch die RUN-LED ist zu erkennen, ob ein Datenaustausch mit dem Buskoppler stattfindet. Die Error-LED signalisiert Überlast oder Drahtbruch. [BP]

4-Kanal-Eingangsklemme



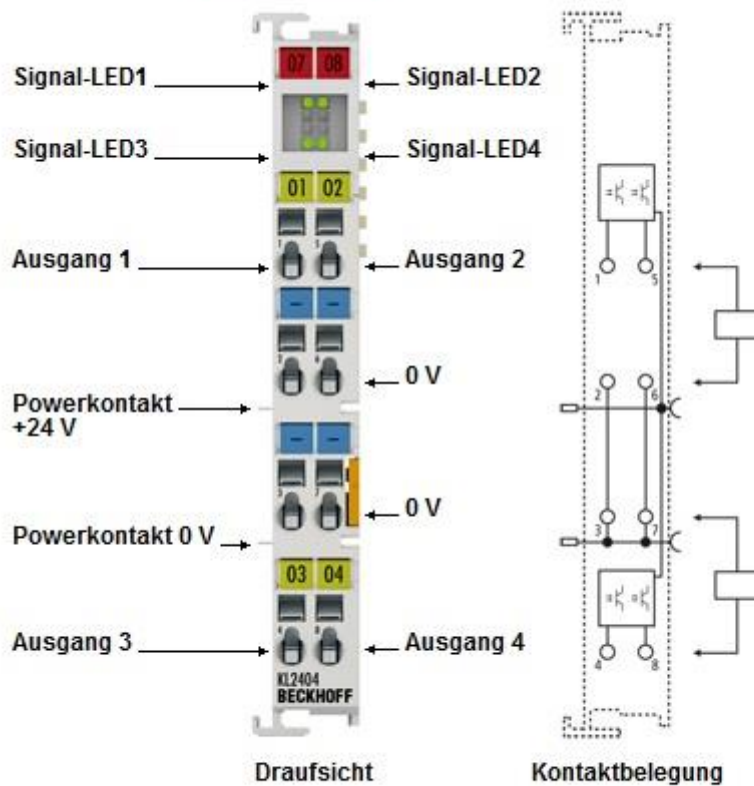
KL3204 | 4-Kanal-Eingangsklemme PT100 (RTD)

Abbildung 8: 4-Kanal-Eingangsklemme KL3204

An diese analoge Eingangsklemme können vier Widerstandssensoren direkt anschließen werden. Die Busklemme kann Sensoren in 2-Leitertechnik verarbeiten und ein Mikroprozessor realisiert die Linearisierung über den gesamten Temperaturbereich. Der Temperaturbereich ist frei wählbar und die Standardeinstellung der Klemme ist bei PT100-Sensoren 0,1 Grad Celsius. Sensorstörungen werden über LEDs angezeigt. [BP]

Digitale Ausgangsklemme

Digital-Ausgang KL2404, KL2424



KL2404, KL2424 | 4-Kanal-Digital-Ausgangsklemmen 24 V DC

Abbildung 9: Digitale Ausgangsklemme KL 2404

Die digitale Ausgangsklemme schaltet die einkommenden 24V-Steuersignale galvanisch getrennt zu dem jeweiligen Aktor. Der Signalzustand der vier enthaltenden Kanäle wird durch Leuchtdioden angezeigt. Die Busklemme ermöglicht den direkten Anschluss von vier 2-Leitertechnik Aktoren. [BP]

Serielle Schnittstellen Klemme (EnOcean)

Sonderfunktion KL6021, KL6041

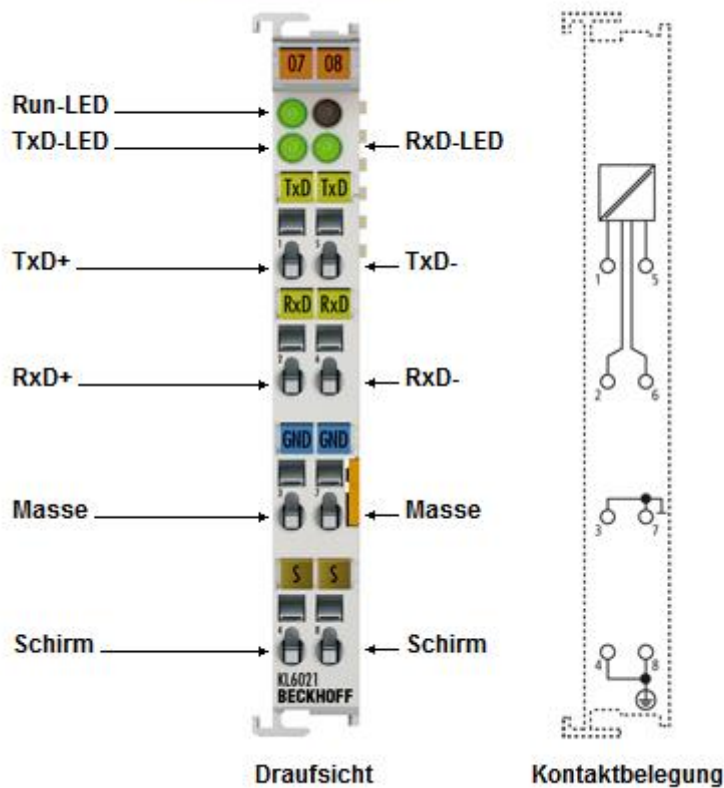


Abbildung 10: Serielle Schnittstelle KL 6021

Die seriellen Schnittstellen KL6021 und KL6041 ermöglichen den Anschluss von Geräten mit einer RS422- oder RS485-Schnittstelle. Das an die Klemme angeschlossene Gerät kommuniziert über den Buskoppler mit dem Automatisierungsgerät. Der aktive Kommunikationskanal arbeitet unabhängig vom überlagerten Bussystem im Voll- oder Halbduplexbetrieb mit bis zu 19.200 Baud (KL6021) bzw. 115,2 kBaud (KL6041). Die Differenzsignalübertragung nach RS422 garantiert hohe Störsicherheit durch galvanisch getrennte Signale. [Beckhoff] Diese Klemme ist notwendig, um den EnOcean-Receiver in das System einzubinden.

Busendklemme

Systemklemmen KL9010



Draufsicht



Kontaktbelegung

KL9010 | Endklemme

Abbildung 11: Busendklemme KL 9010

Jeder Aufbau muss auf der rechten Seite mit einer Busendklemme angeschlossen werden. Erst die Endklemme ermöglicht den Datenaustausch zwischen Buskoppler und Busklemmen. Es gibt keine weiteren Anschlussmöglichkeiten oder Funktionen. [BP]

6. Pflichtenheft

Programmierung der Bürogebäudeautomation exemplarisch an einem Seminar-Büro-Raum mit einer Beckhoff-SPS.

6.1 Zielbestimmungen

Beschrieben wird in diesem Pflichtenheft die Gebäudeautomation eines Bürogebäudes hinsichtlich der Lichtsteuerung und Abschottung über Innen- und Außenjalousien. Betrachtet wird zunächst ein Seminar-Büro-Raum, der mit Hilfe einer Trennwand in separate Büros aufgeteilt werden kann.

Die Lichtstromkreise werden mit einer sogenannten Taster-Schaltung ein- oder ausgeschaltet. Das Schalten erfolgt nicht direkt vom Taster, sondern von einem Aktor. Bei Betätigung eines Tasters wird das Eingangssignal an den jeweiligen Aktor gesendet und dieser schaltet ein oder aus. Dies wird an dieser Stelle erwähnt, da es im Verlauf des Pflichtenhefts sonst zu Missverständnissen kommen kann, wenn vom „Schalten“ in Verbindung mit Tastern gesprochen wird.

6.1.1 Musskriterien

6.1.1.1 Büro-Seminar-Raum-Lichtsteuerung

Die Lichtsteuerung besteht aus 2 Lichtstromkreisen. Über 2 Taster wird ein Signal an einen Aktor geleitet, dieser verarbeitet das Signal und schaltet die Beleuchtung ein oder aus.

6.1.1.2 Trennwandsteuerung

Die Trennwand lässt sich per Schalter aktivieren. Bei geschlossener Trennwand ändert sich die Lichtsteuerung insoweit, dass die unterteilten Büros separat angesteuert werden und zwar von der jeweiligen Eingangstür.

Ist die Trennwand offen, werden die Lichtstromkreise zusammengelegt, so dass von jeder Tür ein oder ausgeschaltet werden kann.

6.1.1.3 Heiz-und Kühlsteuerung

Für das Büro kann über ein Thermostat eingestellt werden, welche Temperatur in dem Büro eingehalten werden soll. Zudem ist ein Wechseln von Sommer auf Wintermodus möglich. Dies führt dazu, dass im Sommer gekühlt und im Winter geheizt wird.

6.1.1.4 Flur-Lichtschaltung

Die Flurbeleuchtung wird über angebrachte Taster getastet. Zudem wird über eine Zeitfunktion ein ausschalten über Zeit realisiert.

Ein installierter Lichtsensor gewährleistet, dass bei zu geringem Lichteinfall die Beleuchtung eingestellt wird.

6.1.2 Kannkriterien

6.1.2.2 Zentralsteuerung

Zentral wird dem Gebäude mitgeteilt, ob eine Anwesenheit vorhanden ist. Zusätzlich kann das Licht ein- oder ausgeschaltet werden.

6.2 Produkteinsätze

6.2.1 Anwendungsbereiche

Zu den angestrebten Anwendungsbereichen gehören Büroräume und Seminarräume.

6.2.2 Zielgruppen

Kleine bis große Bürogebäude

6.3 Produktübersichten

6.3.1 Pro Büroraum

Es werden folgende Sachen pro Büro-Raum benötigt.

- 2 Taster 1-Fach
- 1 Jalousie-Taster mit 4-Schaltfunktionen
- Jalousie-Schaltfunktion (Außen und Innen)
- Lichtsteuerung
- Temperatursensor

6.3.2 Flur

Für die Flurbeleuchtung werden folgende Sachen benötigt.

- Lichtsensor
- Lichtstromkreise
- Taster

6.4 Produktfunktionen

6.4.1 Lichtstromkreise

6.4.1.1 Lichtstromkreise bei geschlossener Trennwand

6.4.2.1 Lichtstromkreis 1 Linkes Büro

Der Lichtstromkreis Stkr1 wird von dem Taster an der Tür ein- oder ausgeschaltet. Bei Betätigung wird ein Steuersignal an den Schaltaktor geschickt und dieser schaltet den Lichtstromkreis ein oder aus.

6.4.2.2 Lichtstromkreis 2 Linkes Büro

Der Lichtstromkreis 2 wird von dem Taster an der Tür geschaltet. Bei Betätigung wird ein Steuersignal an den Schaltaktor geschickt und dieser schaltet den Lichtstromkreis ein oder aus.

6.4.1.2 Lichtstromkreise bei geöffneter Trennwand

Bei geöffneter Trennwand werden die Lichtstromkreise Stkr1 und Stkr2 aus dem einen Büro mit den Stromkreisen Stkr1 und Stkr2 aus dem anderen Büro zusammengelegt und können von beiden Türen mit Taster1 bzw. Taster2 geschaltet werden.

6.4.2 Außen-Jalousiesteuerung

6.4.2.1 Außen-Jalousiesteuerung

Die Außen-Jalousiesteuerung erfolgt über ein 4-Fach-EnOcean-Funktaster. Es wird unterschieden zwischen langem und kurzem Drücken.

Bei langem Drücken (>500ms) wird die Jalousie auf- oder abgefahren. Die Fahrt kann durch kurzes Drücken der gleichen Taste beendet werden.

6.4.3 Innen-Jalousiesteuerung

6.4.3.1 Innen-Jalousiesteuerungen bei geschl. Trennwand

Die Innen-Jalousiesteuerung erfolgt über ein 4-Fach-EnOcean-Funktaster. Es wird unterschieden zwischen langem und kurzem Drücken.

Bei langem Drücken (>500ms) wird die Jalousie auf- oder abgefahren. Die Fahrt kann durch kurzem Drücken der gleichen Taste beendet werden.

6.4.4 Flurschaltung

Die Lichtstromkreise für die Flurbeleuchtung werden über die jeweiligen Flur-Taster geschaltet und über eine Zeitfunktion wieder aus. Fällt der Lux-Wert unter den Grenzwert von 100lux wird die Flurbeleuchtung eingeschaltet.

6.4.5 Trennwand

Die Trennwand wird über einen separaten Schalter aktiviert oder deaktiviert. Beim Erreichen der Endposition werden über angebrachte Endschalter die jeweiligen Schaltfunktionen, für die Beleuchtung, der Büros aktiv oder inaktiv.

6.4.6 Zentralsteuerung

Über 2 Taster wird das Licht ein- oder ausgeschaltet. Ein Schalter simuliert die Anwesenheit im Gebäude. Daraufhin wird die Heizungssteuerung von abwesend auf anwesend geschaltet und je nach Sommer- oder Wintereinstellung wird die Funktion ausgeführt.

6.5 Technische Produktbetrachtungen

6.5.1 Software

Die Programmierung des Projekts erfolgt mit der Software TwinCat 3.

6.5.1.1 Software-Informationen

Die Software TwinCat 3 ist eine von Beckhoff selbst produzierte Software. Durch kontinuierliche Weiterentwicklung der hauseigenen Software hat Beckhoff mit TwinCat 3 die Vorgängerversion abgelöst. Somit ist es jetzt möglich mit nur einem Programm zu konfigurieren und zu programmieren. Einige Highlights der Software sind:

- Visual Studio-Integration
- Mehr Freiheitsgrade bei der Wahl der Programmiersprachen.
- Unterstützung der objektorientierten Erweiterung der IEC 61131-3
- Verwendung von C/C++ als Programmiersprache für Echtzeitanwendungen
- Anbindung an MATLAB/Simulink
- Aktive Unterstützung von Multi-Core- und 64-Bit-Betriebssystemen.
- Automatische Codegenerierung und Projekterstellung mit dem TwinCat Automation Interface
- Programmierung mit ST, AWL, KOP, FBS, ST

7. Die Software TwinCat 3

Um die Programmierung mit TwinCat 3 zu erläutern, wird die Programmierung einer Funktion im Detail beschrieben. Die beschriebene Funktion ist die im Pflichtenheft aufgeführte Lichtsteuerung des Seminar-Büros.

Die Programmierung des Bürogebäudes soll im weitesten flexibel sein und daher wird nach dem Prinzip der Instanziierung programmiert.

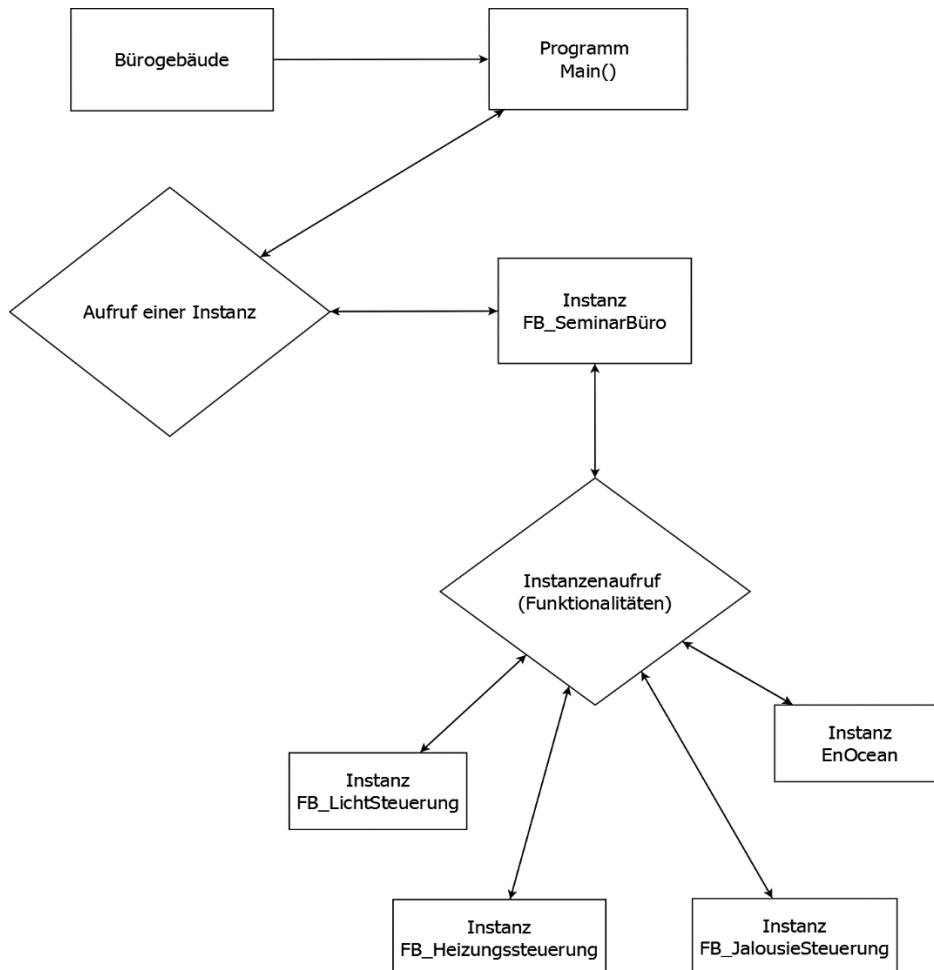


Abbildung 12 Schaubild Instanziierung

Um dies kurz zu erläutern: Im Hauptprogramm - Main - wird die gewünschte Instanz aufgerufen. Die aufgerufene Instanz beinhaltet unterschiedliche Instanzen, welche für diese Instanz relevant sind und zwar in Bezug auf die Funktionalität. Die Instanz - Seminarbüro - enthält z.B. die Instanzen Jalousiesteuerung, Heizungssteuerung, Lichtsteuerung und die EnOcean-Instanz. Die Variablen werden den entsprechenden Instanzen weitergegeben. Die „echten“ Ein- und Ausgangsvariablen werden nur im Programm Main verarbeitet. Um die Übersicht nicht zu verlieren, sollte hier eine eindeutige und gut erkennbare Bezeichnung gewählt werden. Angenommen das geplante Bürogebäude besteht aus sieben Etagen und zwei Untergeschosse. Die Etagen beinhalten 7 Einzelbüros, 1 Seminarbüro, Bad und eine Teeküche. Eine mögliche Bezeichnung des Seminarbüros im Erdgeschoss könnte wie in folgender Tabelle gezeigt aussehen:

Tabelle 1: Mögliche Variablenvergabe

Etage	Raum	Raum-Nummer	Funktion
E00	SB	001	Lstkr1_1
E00	SB	001	Lstkr2_1
E00	SB	002	Lstkr1_2
E00	SB	002	Lstkr2_2
E00	SB	001	LichtTaster1
E00	SB	001	LichtTaster2
E00	SB	002	LichtTaster1
E00	SB	002	LichtTaster2
E00	SB	001	Ja_A_Auf
E00	SB	001	Ja_A_Ab
E00	SB	001	Ja_In_Auf
E00	SB	001	Ja_In_Ab
E00	SB	002	Ja_A_Auf
E00	SB	002	Ja_A_Ab
E00	SB	002	Ja_In_Auf
E00	SB	002	Ja_In_Ab
E00	SB	001	RaumTemp
E00	SB	001	Soll_Temp
E00	SB	002	RaumTemp
E00	SB	002	Soll_Temp
E00	SB	001	Heizen
E00	SB	001	Kühlen
E00	SB	002	Heizen
E00	SB	002	Kühlen
E00	SB	001	Trennwand

Tabelle 2: Legende

E00...E04	Etagen Erdgeschoss bis 4 Etage
U01..U02	Untergeschosse
SB	Seminarbüro
WC	Bad
TK	TeeKüche
EB	Einzelbüro
Lstkr	Lichtstromkreis
Ja_In_Auf	Innenjalousie Auf
Ja_A_Auf	Außenjalousie Auf

7.1 Installation der Software TwinCat 3

Beckhoff bietet auf der Homepage ein Installationspaket zum Download an. Es benötigt eine Anmeldung, um es zu laden. Nach erfolgreicher Anmeldung steht einem das gewünschte Paket zum Download zur Verfügung. Eine Anmeldung bietet den Vorteil, dass über diesen Weg fehlende Bibliotheken nachträglich downgeloadet und installiert werden können. Wenn das Programm auf deutscher Sprache installiert werden soll, muss vorher Microsoft Visual Studio 2013 Shell und das entsprechende Sprachpaket installiert werden. Erst dann sollte die Installation von TwinCat 3 erfolgen. Zudem wurde vom Support bestätigt, dass die Version 2013 und nicht die Version 2015 verwendet werden soll. Diese Information wurde bei Erstellung dieser Arbeit eingeholt und könnte nicht mehr dem aktuellen Stand entsprechen. Nach erfolgreicher Installation und Starten des Programms erscheint dieses Fenster. (Abbildung 13)

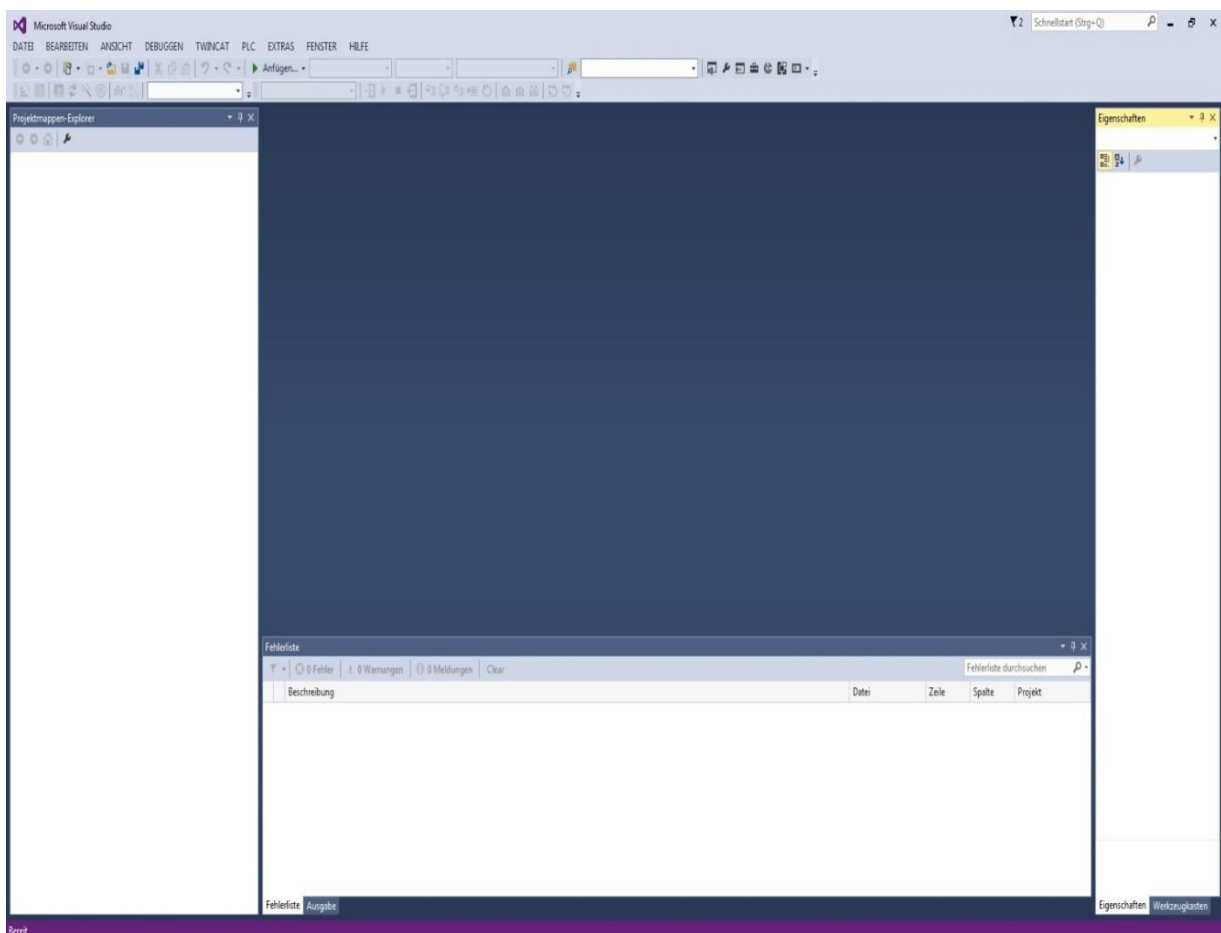


Abbildung 13: TwinCat3 Start

7.2 Programmieren mit TwinCat 3

Über die Menüleiste - Datei Neu, Projekt - wird ein „Neues Projekt“ angelegt. Es öffnet sich ein Fenster, in dem Parameter festgelegt werden z.B. Projekt-Name, Speicher-Ort und Projekt-Mappen-Name. Ausgewählt wird hier unter TwinCat Projekt ein XAE (eXtended Automation Engineering) Projekt im XML-Format.

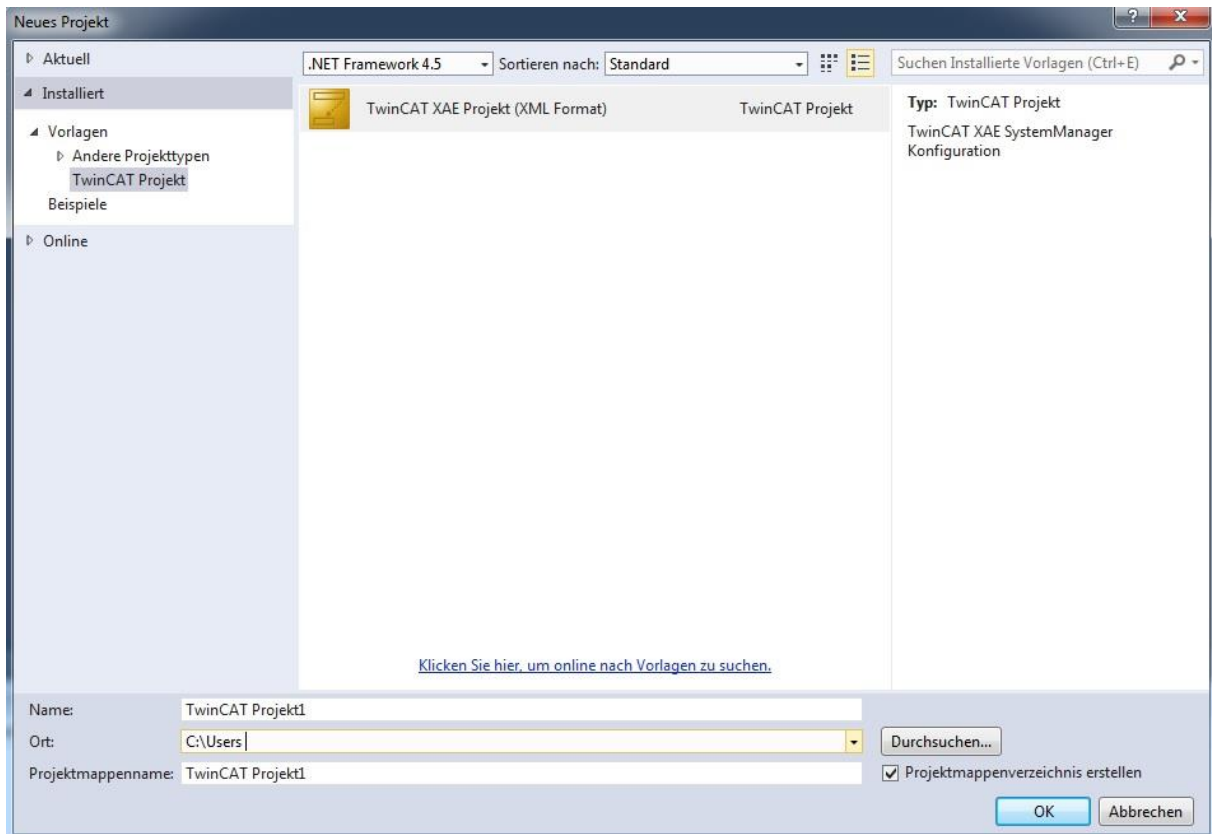


Abbildung 14: Projekt anlegen

Sind alle Parameter eingegeben, wird durch Klicken auf OK das Projekt angelegt und es erscheint im Projektmappen-Explorer (Abbildung 15). Dieser Editor bietet eine genaue Übersicht über das gesamte Projekt.

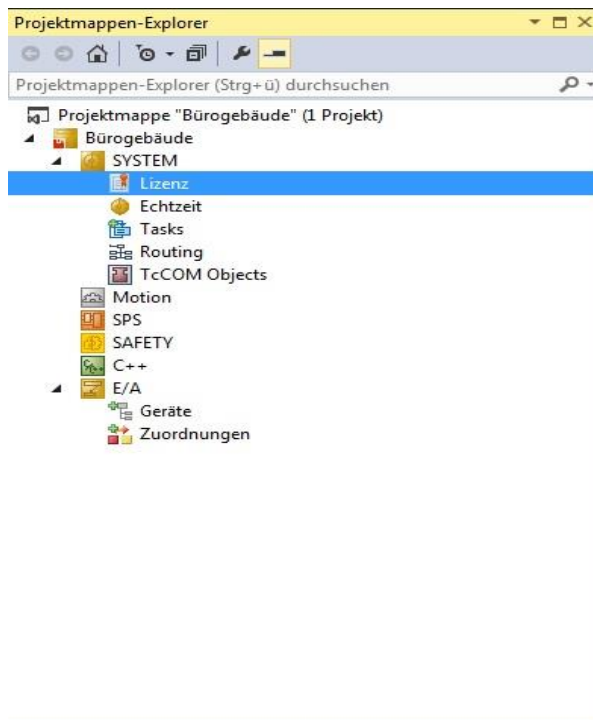


Abbildung 15: Projektmappen-Explorer

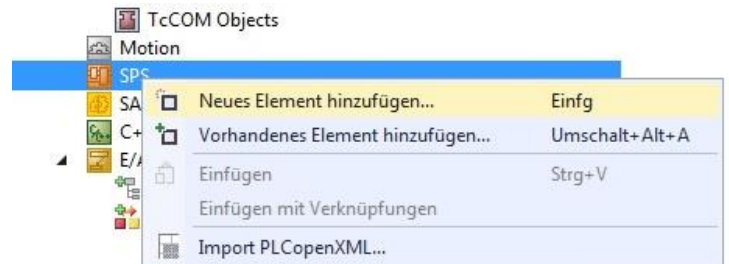


Abbildung 16: Neues Element

Wie es in Abbildung 15 zu sehen ist, kann über den Explorer auf Eigenschaften bzw. Einstellung direkt zugegriffen werden. Fehlende Lizenzen können nachgetragen oder auch Veränderungen vorgenommen werden. Dazu gehört z.B. ein Hinzufügen des Tasks, welcher bei Erstellung des Projekts automatisch eingestellt wird und nicht wie in der Vorgängerversion TwinCat2 von Hand hinzugefügt werden muss.

Um ein SPS-Programm zu schreiben, muss hier auf den Reiter SPS mit Rechtsklick ein neues Element (Abbildung 16) sowie ein Projekt PLC (Programmable Logic Controller) hinzugefügt werden. Es erscheint ein Fenster, in dem das Projekt ausgewählt und mit einem Namen versehen wird.

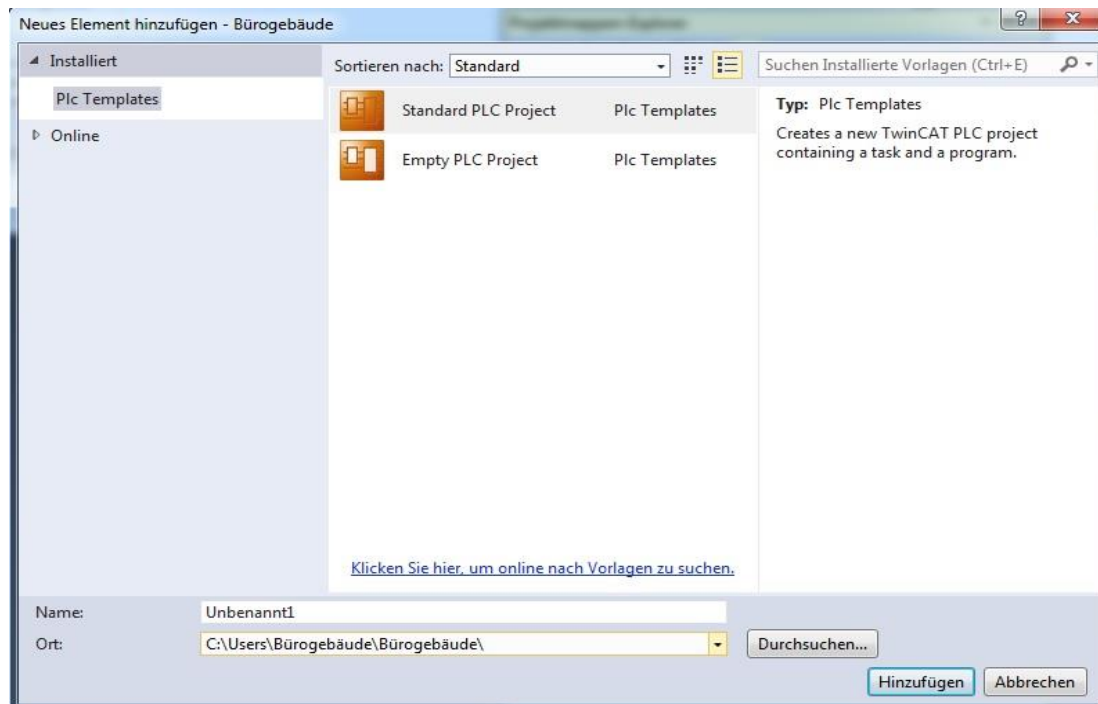


Abbildung 17: PLC Projekt hinzufügen

Es erscheint hier die Auswahl zwischen einem Standard PLC oder einem leeren (Empty) PLC Projekt. (Abbildung 17) Bei der Auswahl eines Standard PLC Projekts wird ein Projekt mit einem Task und einem Programm erstellt. Somit kann sofort mit der Programmierung gestartet werden. Nach Bestätigen durch einen Klicken auf Hinzufügen taucht das Projekt im Projektmappen-Explorer auf. (Abbildung 18)

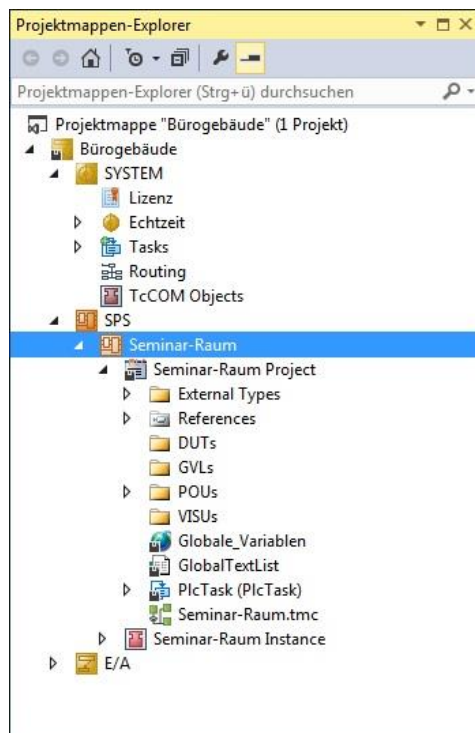


Abbildung 18: Projektmappen-Explorer

Programmiert wird der CX9020 mit Hilfe von POU (Program Organization Unit). Beckhoff beschreibt die POU mit folgender Definition.

„Programmierobjekte (POUs):

Dies sind Programme, Funktionen, Funktionsblöcke, Methoden, Schnittstellen, Aktionen, Datentyp-Definitionen usw.

Das Instanzieren erfolgt dadurch, dass ein Programm-POU von der Task einer Applikation aufgerufen wird“ [BK]

Mit Rechtsklick auf POU im Projektmappen-Editor lassen sich POUs hinzufügen. Als Erstes sollte immer ein POU mit Namen Main angelegt werden, welches als Hauptprogramm dient und andere POUs aufrufen kann. TwinCat 3 unterstützt alle Standard-Programmiersprachen gemäß der Norm IEC 61131-3. So kann mit der hinzugefügten POU (Abbildung 19) eine Programmiersprache ausgewählt werden.

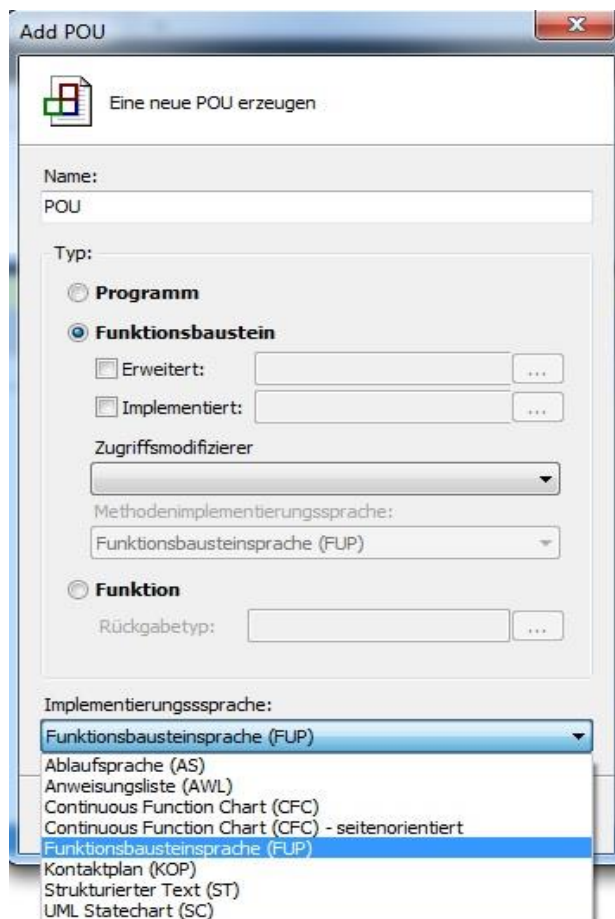


Abbildung 19: POU erstellen

Um die Lichtsteuerung zu programmieren, wird ein Funktionsbaustein „FB Lichtsteuerung“ mit der Programmiersprache Funktionsbaustein (FUB) erstellt. (Abbildung 20)

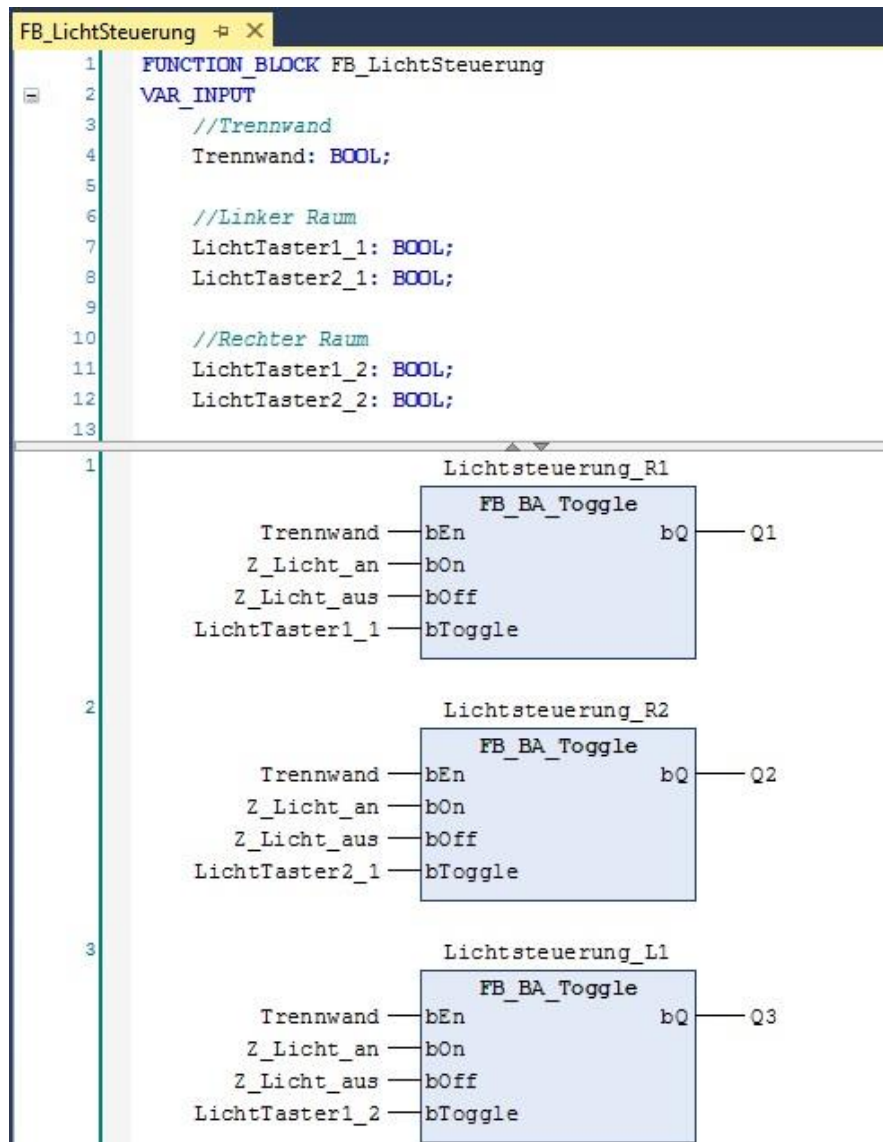


Abbildung 20: POU FB_LichtSteuerung

Das Einfügen von Funktionsblöcken funktioniert auf unterschiedliche Weisen. Zum einen können über den Werkzeugkasten rechts im Fenster Standardfunktionen per „Drag and Drop“ in das gewünschte Netzwerk gezogen werden. Eine andere Möglichkeit, Funktionsbausteine hinzuzufügen, ist über ein Auswahlfenster, welches mit einem Rechtsklick auf das Netzwerk erscheint. Über diesen Weg kann der gewünschte Baustein aus installierten Bibliotheken ausgewählt werden. In Abbildung 21 ist rechts der Werkzeugkasten zu sehen und die Methode über Rechtsklick auf ein Netzwerk einen Funktionsbaustein hinzuzufügen.

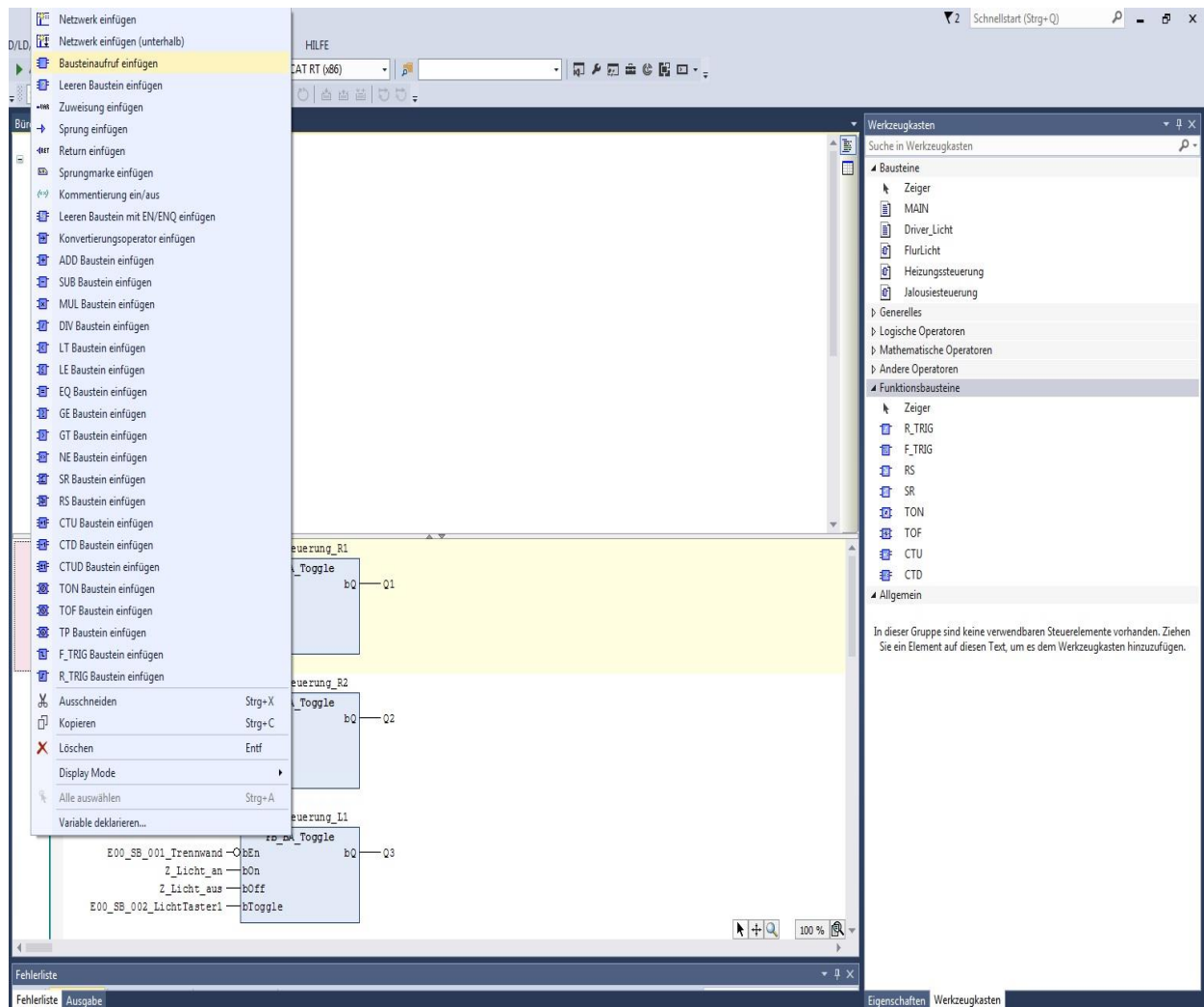


Abbildung 21: Funktionsbausteine hinzufügen

Um die geplante Funktion der Lichtschaltung im Seminar-Büro zu programmieren, wird ein Toggle (Umschalten) Baustein verwendet. (Abbildung 22). Über den Eingang bEn wird der FUB aktiviert. Die Eingänge bOn und bOff sind für das Zentrale Ein- und Ausschalten zuständig. Über den Eingang bToggle lässt sich der Ausgang des FUBs ein- oder ausschalten.

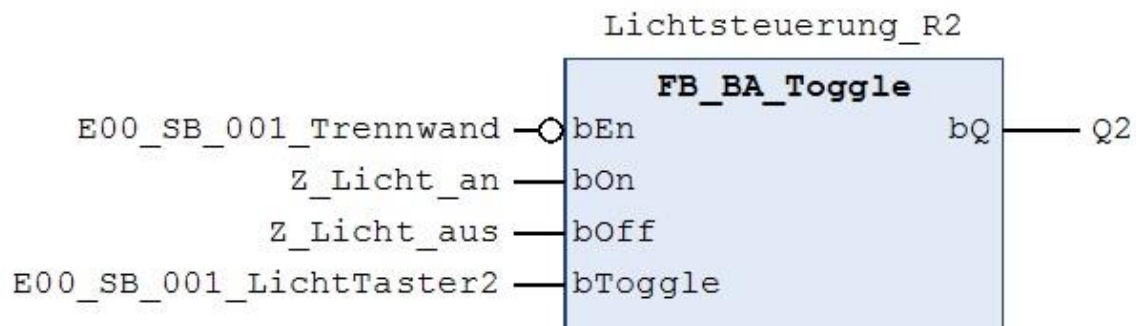


Abbildung 22: Funktionsbaustein Toggle

Sind alle benötigten Ein- und Ausgangsvariablen deklariert, kann mit der Zuweisung der Ein- und Ausgängen an den entsprechenden Funktionsbausteinen fortgefahren werden.

```

FB_LichtSteuerung  ▢ X
1  FUNCTION_BLOCK FB_LichtSteuerung
2  VAR_INPUT
3      //Trennwand
4      Trennwand: BOOL;
5
6      //Linker Raum
7      LichtTaster1_1: BOOL;
8      LichtTaster2_1: BOOL;
9
10     //Rechter Raum
11     LichtTaster1_2: BOOL;
12     LichtTaster2_2: BOOL;
13
14
15  END_VAR
16  VAR_OUTPUT
17     // Stromkreise rechtes Büro
18     LStkr1_1: BOOL;
19     LStkr2_1: BOOL;
20     //Stromkreise linkes Büro
21     LStkr1_2: BOOL;
22     LStkr2_2: BOOL;
23  END_VAR
24  VAR
25     Lichtsteuerung_R1:FB_BA_Toggle;
26     Lichtsteuerung_R2:FB_BA_Toggle;
27     Lichtsteuerung_L1:FB_BA_Toggle;
28     Lichtsteuerung_L2:FB_BA_Toggle;
29     Lichtsteuerung_01: FB_BA_Toggle;
30     Lichtsteuerung_02: FB_BA_Toggle;
31
32     //Hilfs_Variablen
33     Q1:BOOL;
34     Q2:BOOL;
35     Q3:BOOL;
36     Q4:BOOL;
37     Q5: BOOL;
38     Q6: BOOL;
39     RS1: RS;
40  END_VAR
41

```

Abbildung 23: Variablen

Das Hinzufügen des Funktionsbausteins über die Methode - Rechtsklick auf das Netzwerk - öffnet ein Fenster, in dem aus den mitgelieferten Bibliotheken der gewünschte Baustein ausgewählt wird. (Abbildung 24)

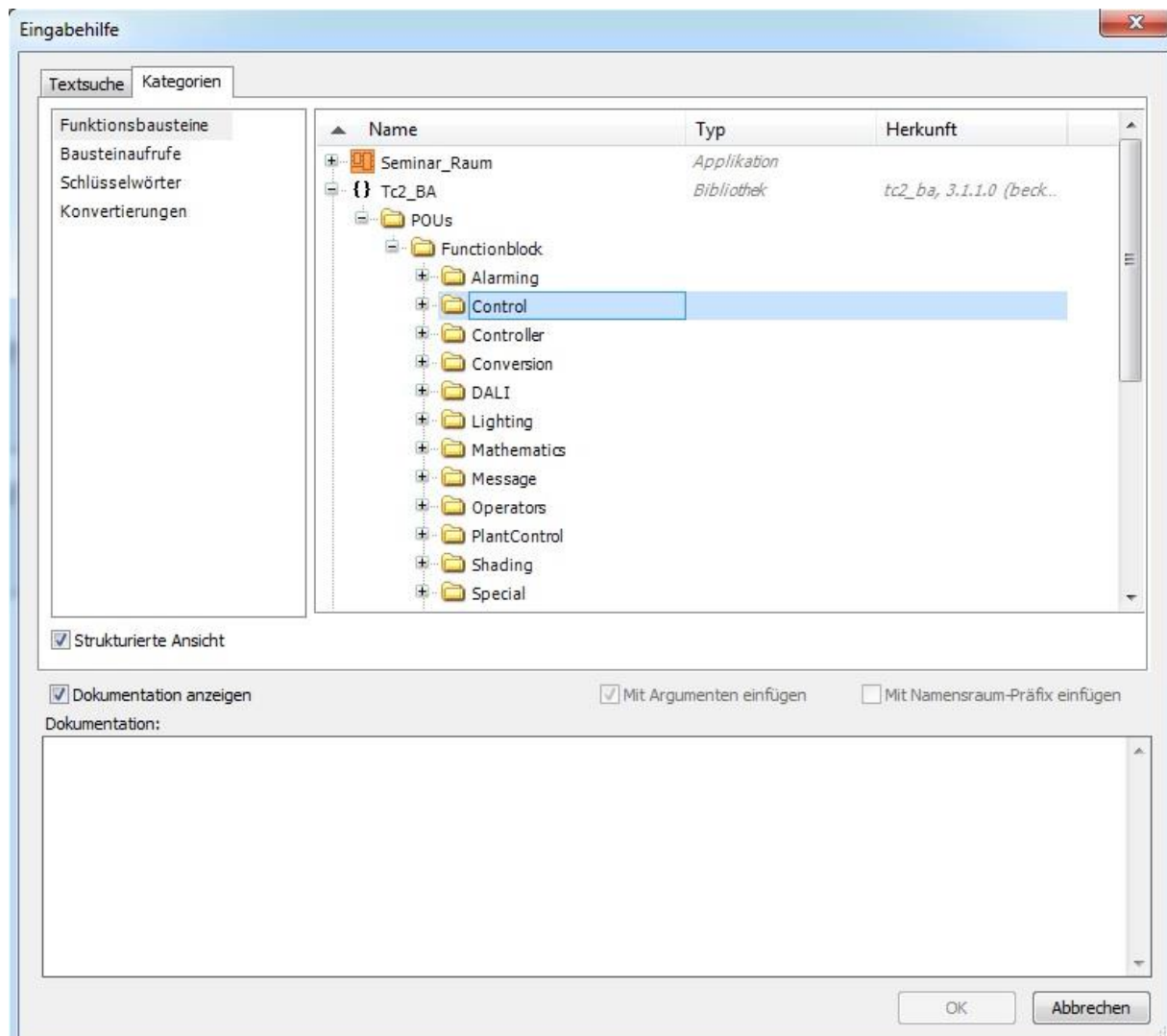


Abbildung 24: Fub-Bibliothek

Ist einem der Baustein bekannt, kann auch ein „leerer“ FUB hinzugefügt und die Bezeichnung eingetragen werden. Für die zu programmierende Funktion werden sechs Bausteine hinzugefügt. Sie bekommen ebenfalls eine Bezeichnung, die nachvollziehbar ist. Da bei der Programmierung Funktionsblöcke ausgeschaltet werden, kommen Hilfsvariablen zum Einsatz.

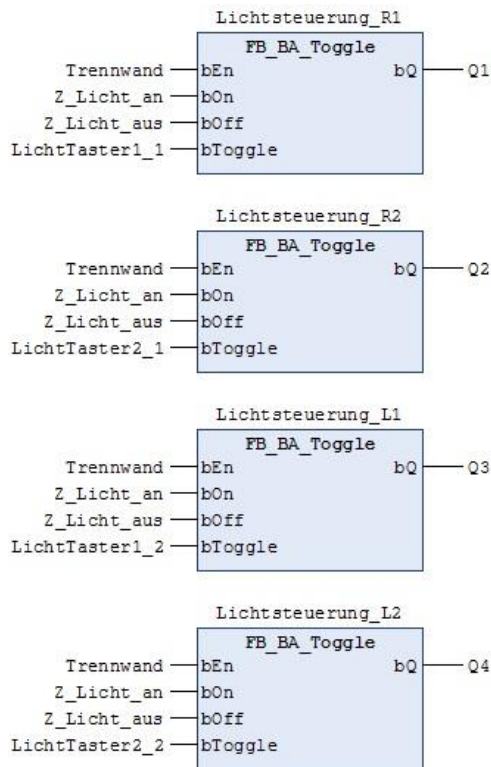


Abbildung 25: Lichtsteuerung einzeln

Für die Lichtschaltung gibt es zwei Zustände: die Trennwand ist offen oder geschlossen. Abbildung 25 zeigt die Funktion, wenn die Trennwand geschlossen wurde. An den Eingängen *bEn* liegt eine 1 (True) an und somit wird der Baustein aktiv. Über die Eingänge *bToggle* können die Lichtstromkreise ein oder ausgeschaltet werden. Wird die Trennwand geöffnet, ändern sich die Eingänge an *bEn* und die Bausteine werden inaktiv und der zweite Zustand der Schaltung aktiv (Abbildung 26). Die Hilfsvariablen *Q1* bis *Q6* schalten je nach Zustand der Trennwand die Ausgänge für die Lichtstromkreise.

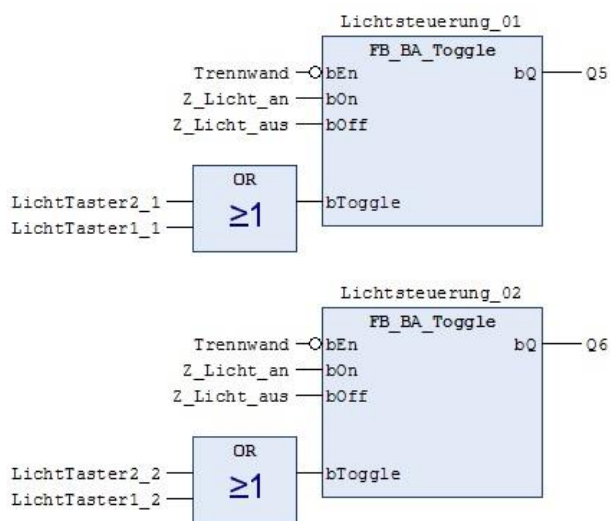


Abbildung 26: Lichtsteuerung Zusammen

Um einem Funktionsblock Ein- oder Ausgänge zuzuweisen, wird auf dem gewünschten Ein bzw. Ausgang geklickt und es erscheint ein Fenster, welches alle vorher deklarierten Variablen anzeigt. Mit einem weiteren Klick auf die gesuchte Variable wird diese dem Ein- oder Ausgang zugewiesen. Es kann auch durch eintippen des Variablennamens eine Zuweisung durchgeführt werden (Abbildung 27)

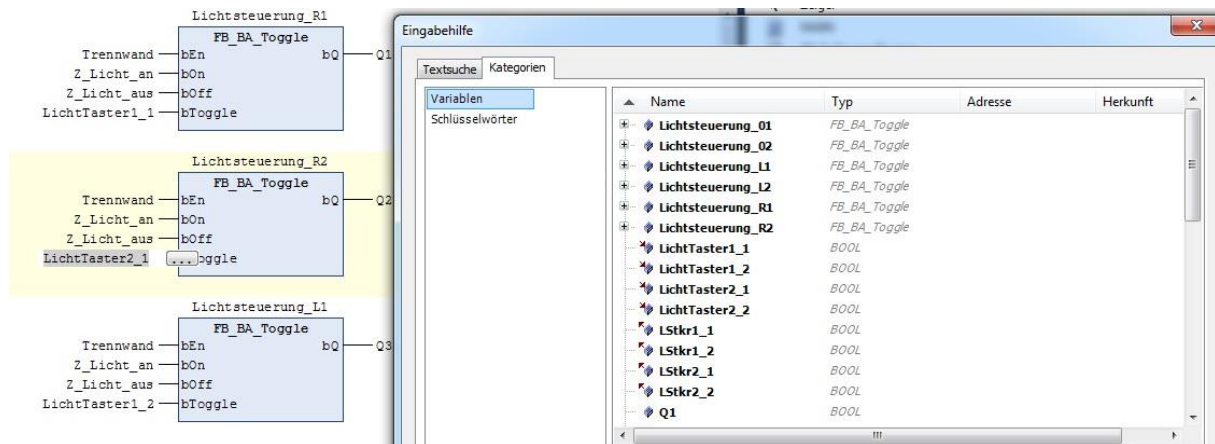


Abbildung 27: Zuweisung von Variablen an einen FUB

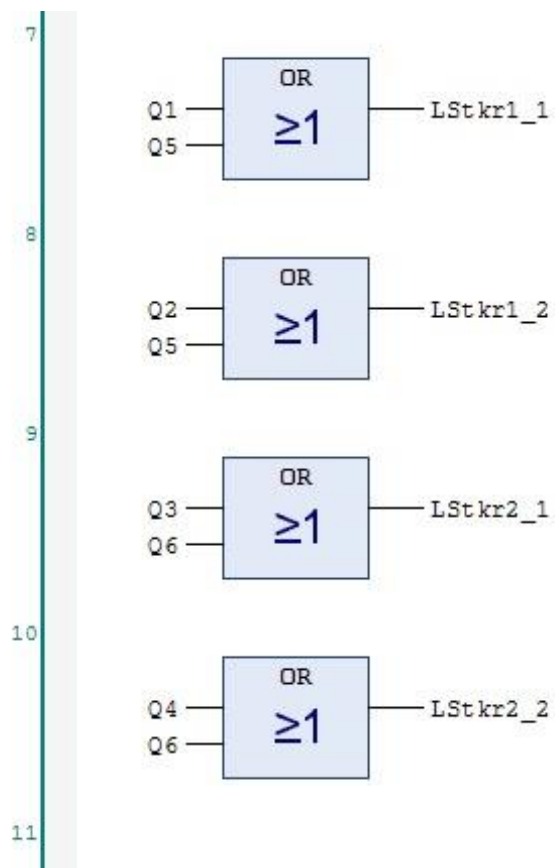


Abbildung 28: Ausgänge Lichtsteuerung

Wie in Abbildung 28 zu sehen ist, werden die Lichtstromkreise je nach Schaltzustand einzeln oder im Verbund geschaltet. Die Hilfsvariablen Q1 bis Q4 sind aktiv, wenn die Trennwand geschlossen ist. Somit werden separat die Lichtstromkreise in den getrennten Büros geschaltet. Bei geöffneter Trennwand sind die Hilfsvariablen Q1 bis Q4 inaktiv und Q5, Q6 aktiv, wodurch ein anderes Schalten eingeleitet wird.

Die Lichtstromkreise *LStkr1_1* und *LStkr1_2* werden zusammengelegt. Über die Zentralfunktion können die Lichtstromkreise des gesamten Büros ein- oder ausgeschaltet werden. Abschließend wird das Programm kompiliert und somit auf Fehler überprüft. Der Funktionsbaustein FB_LichtSteuerung (Abbildung 29) ist somit fertig programmiert. Des Weiteren wird nun der Funktionsbaustein FB_SeminaBuero erstellt, der die Instanz Lichtsteuerung beinhaltet. Der Funktionsbaustein Seminar-Büro wird in der Programmiersprache ST angelegt.

Ein- und Ausgänge werden deklariert. Hier ist es wichtig darauf zu achten, welche Ein- und Ausgangsvariablen nötig sind. Im unteren Teil des Fensters wird nun mit Rechtsklick ein Popup-Fenster geöffnet und Eingabehilfe ausgewählt (Abbildung 30).

```

FB_LichtSteuerung
1 FUNCTION_BLOCK FB_LichtSteuerung
2 VAR_INPUT
3   //Trennwand
4   Trennwand: BOOL;
5
6   //Linker Raum
7   LichtTaster1_1: BOOL;
8   LichtTaster2_1: BOOL;
9
10  //Rechter Raum
11  LichtTaster1_2: BOOL;
12  LichtTaster2_2: BOOL;
13
14
15  END_VAR
16  VAR_OUTPUT
17  // Stromkreise rechtes Büro
18  LStkr1_1: BOOL;
19  LStkr2_1: BOOL;
20  //Stromkreise linkes Büro
21  LStkr1_2: BOOL;
22  LStkr2_2: BOOL;
23
24  END_VAR
25  VAR
26    Lichtsteuerung_R1:FB_BA_Toggle;
27    Lichtsteuerung_R2:FB_BA_Toggle;
28    Lichtsteuerung_L1:FB_BA_Toggle;
29    Lichtsteuerung_L2:FB_BA_Toggle;
30    Lichtsteuerung_01: FB_BA_Toggle;
31    Lichtsteuerung_02: FB_BA_Toggle;
32
33    //Hilfs Variablen
34    Q1:BOOL;
35    Q2:BOOL;
36    Q3:BOOL;
37    Q4:BOOL;
38    Q5: BOOL;
39    Q6: BOOL;
40    RS1: RS;
41  END_VAR

```

Abbildung 29: FB_LichtSteuerung

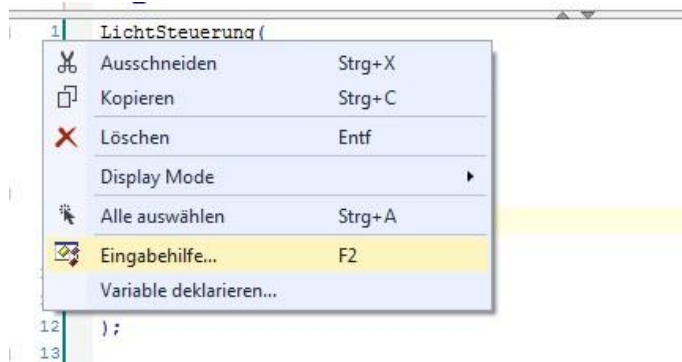


Abbildung 30: Eingabehilfe

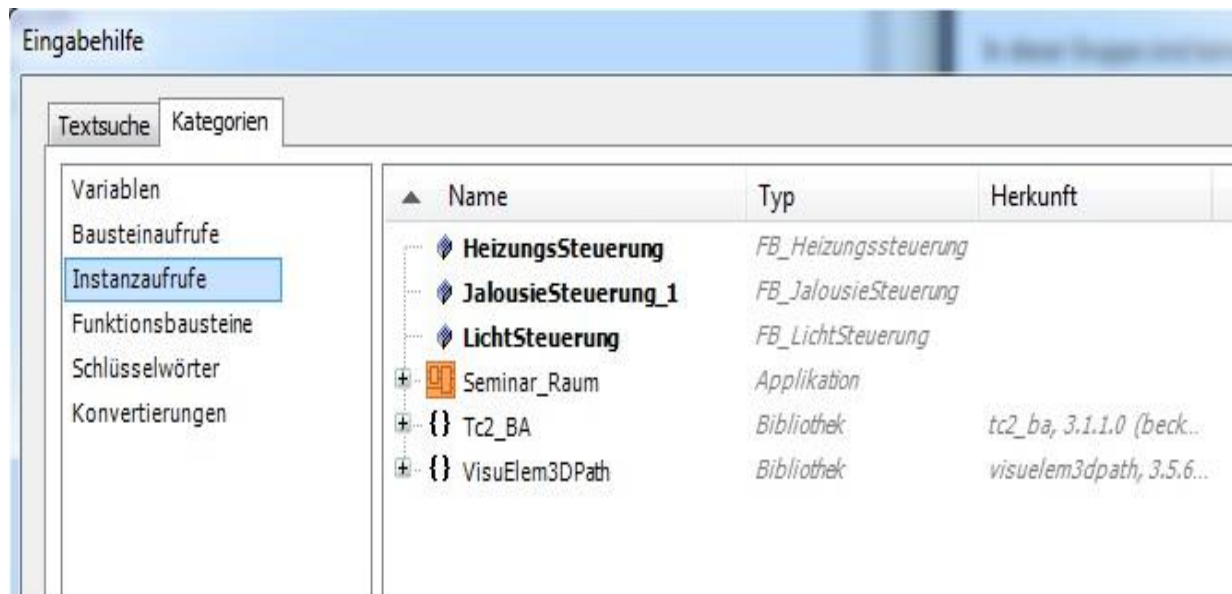


Abbildung 31: Instanzaufufe

Nach der Betätigung von der Eingabehilfe erscheint ein neues Fenster (Abbildung 31), in dem alle angelegten Instanzen angezeigt werden. Ausgewählt wird die zuvor erstellte Instanz LichtSteuerung.

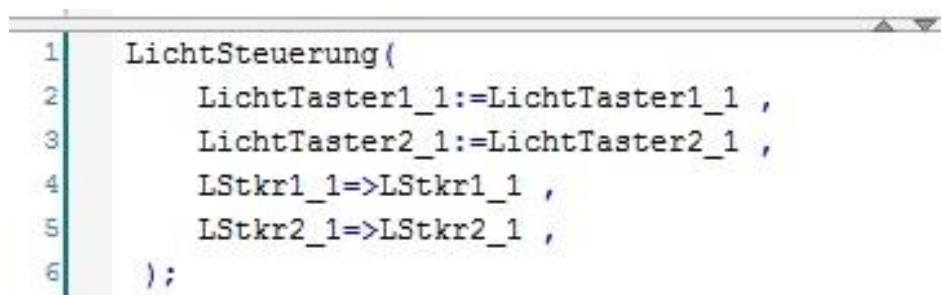


Abbildung 32: Instanz LichtSteuerung

Es werden die Variablen deklariert und können somit weitergegeben werden (Abbildung 32). Damit ist der Funktionsbaustein SeminarBüro mit Aufruf der Instanz Lichtsteuerung programmiert. Somit fehlt noch der Aufruf im Programm Main der Instanz Seminarraum und der Übergabe der entsprechenden Variablen.

Die POU Main wird mit der Programmiersprache „Strukturierter Text (ST)“ angelegt (Abbildung 33), erscheint im Projektmappen-Editor und besteht aus zuvor erwähntem Aufbau (Abbildung 34).

Auch im Hauptprogramm Main werden im oberen Fenster die Ein- und Ausgangsvariablen deklariert sowie die verwendeten Funktions-Bausteine.

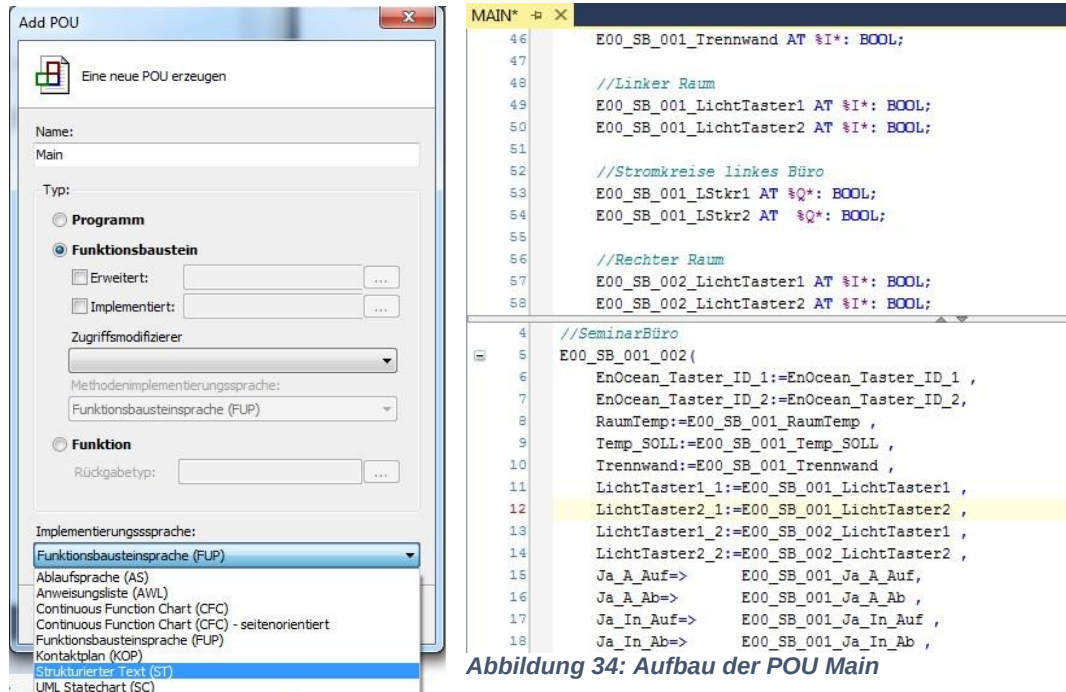


Abbildung 33: POU Main

Die Deklaration von Eingängen und Ausgängen sieht wie folgt aus:

```

43
44 //Deklaration von SeminarBüros mit entsprechender Bezeichnung
45 E00_SB_001_002: FB_SeminarBueroRaum;
46
47 //Lichtsteuerung
48 //Trennwand
49 E00_SB_001_Trennwand AT %I*: BOOL;
50
51 //Linker Raum
52 E00_SB_001_LichtTaster1 AT %I*: BOOL;
53 E00_SB_001_LichtTaster2 AT %I*: BOOL;
54
55 //Stromkreise linkes Büro
56 E00_SB_001_LStkr1 AT %Q*: BOOL;
57 E00_SB_001_LStkr2 AT %Q*: BOOL;
58
59 //Rechter Raum
60 E00_SB_002_LichtTaster1 AT %I*: BOOL;
61 E00_SB_002_LichtTaster2 AT %I*: BOOL;
62
63 //Stromkreise rechtes Büro
64 E00_SB_002_LStkr1 AT %Q*: BOOL;
65 E00_SB_002_LStkr2 AT %Q*: BOOL;
66

```

Abbildung 35: Deklaration

In der Abbildung 35 gut zu erkennen, wird mit der Zeile *E00_SB_001_002: FB_SeminarBuero* ein Raum mit der Bezeichnung *E00_SB_001_002* als Instanz SeminarBüroRaum deklariert.

Bis zu dieser Stelle wurde nur eine Instanziierung im Detail besprochen. Im Seminarbüro werden jedoch weitere Instanzen mit entsprechender programmierter Funktionalität benötigt. Die Vorgehensweise ist für die benötigten Instanzen wie Jalousiesteuerung sowie Heizungssteuerung die gleiche. Es werden Funktionsbausteine erstellt und als Instanz aufgerufen. Nach erfolgreicher Programmierung der anderen Funktionalitäten und erneutem Aufruf der Instanz Seminarbüro im Hauptprogramm Main werden alle benötigten Variablen angezeigt (Abbildung 36) und müssen den entsprechenden Ein- und Ausgängen zugewiesen werden. Die Realisierung der anderen Instanzen wird im weiteren Verlauf angesprochen. Aber zu nächst wird der CX9020 konfiguriert, um das bis jetzt erstellte Programm testen zu können.

```

5
6 E00_SB_001_002(
7     EnOcean_Taster_ID_1:=EnOcean_Taster_ID_1 ,
8     EnOcean_Taster_ID_2:=EnOcean_Taster_ID_2,
9     RaumTemp:=E00_SB_001_RaumTemp ,
10    Temp_SOLL:=E00_SB_001_Temp_SOLL ,
11    Trennwand:=E00_SB_001_Trennwand ,
12    LichtTaster1_1:=E00_SB_001_LichtTaster1 ,
13    LichtTaster2_1:=E00_SB_001_LichtTaster2 ,
14    LichtTaster1_2:=E00_SB_002_LichtTaster1 ,
15    LichtTaster2_2:=E00_SB_002_LichtTaster2 ,
16    Ja_A_Auf=>      E00_SB_001_Ja_A_Auf,
17    Ja_A_Ab=>      E00_SB_001_Ja_A_Ab ,
18    Ja_In_Auf=>     E00_SB_001_Ja_In_Auf ,
19    Ja_In_Ab=>     E00_SB_001_Ja_In_Ab ,
20    Ja_A_Auf_2=>    E00_SB_002_Ja_A_Auf ,
21    Ja_A_Ab_2=>    E00_SB_002_Ja_A_Ab ,
22    Ja_In_Auf_2=>   E00_SB_002_Ja_In_Auf ,
23    Ja_In_Ab_2=>   E00_SB_002_Ja_In_Ab ,
24    Heizen=>      E00_SB_001_Heizen ,
25    Kuehlen=>     E00_SB_001_Kuehlen ,
26    LStkr1_1=>E00_SB_001_LStkr1 ,
27    LStkr2_1=>E00_SB_001_LStkr2 ,
28    LStkr1_2=>E00_SB_002_LStkr1 ,
29    LStkr2_2=>E00_SB_002_LStkr2 );

```

Abbildung 36: Erneuter Aufruf der Instanz

7.3 Konfiguration des CX9020

Um das Programm testen bzw. auf den Controller übertragen zu können, muss dieser eingerichtet werden. Als erstes muss der Controller über ein Lan-Kabel direkt mit dem PC verbunden werden. Ist ein Netzwerk vorhanden, kann der Controller auch über einen Switch angesprochen werden. Es hat sich aber herausgestellt, dass eine direkte Verbindung für die Konfiguration besser geeignet ist. Falls die Verbindung steht und der Controller sofort angezeigt wird, kann er als Zielsystem ausgewählt werden (Abbildung 37). Erscheint der Controller nicht in der Liste, kann eine Broadcast-Suche gestartet werden, wie in Abbildung 38 gezeigt wird.

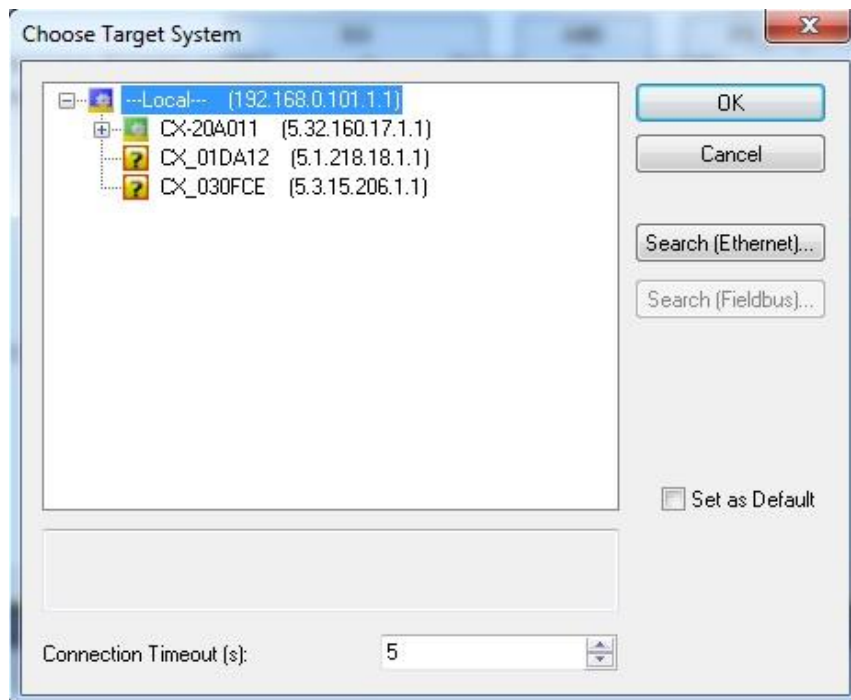


Abbildung 37: Zielsystem

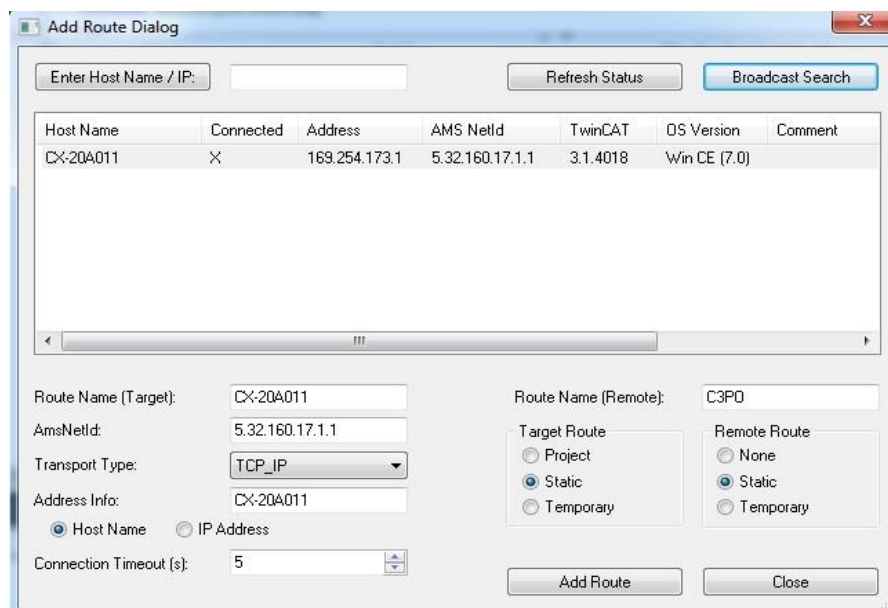


Abbildung 38: Broadcastsuche

Bei erfolgreicher Suche des Controllers und Zuweisung einer Route wird der Controller im Projektmappen-Editor unter System aufgeführt.

Desweiteren kann nun dem Controller eine feste IP-Adresse vergeben (Abbildung 39) und nach angeschlossenen Klemmen gesucht werden. Die Suche nach angeschlossenen Klemmen kann über den Projektmappen-Editor mit einem Rechtsklick auf Devices und im darauf folgenden Fenster mit einem Scan gestartet werden. (Abbildung 40)

Abbildung 39: IP-Konfiguration

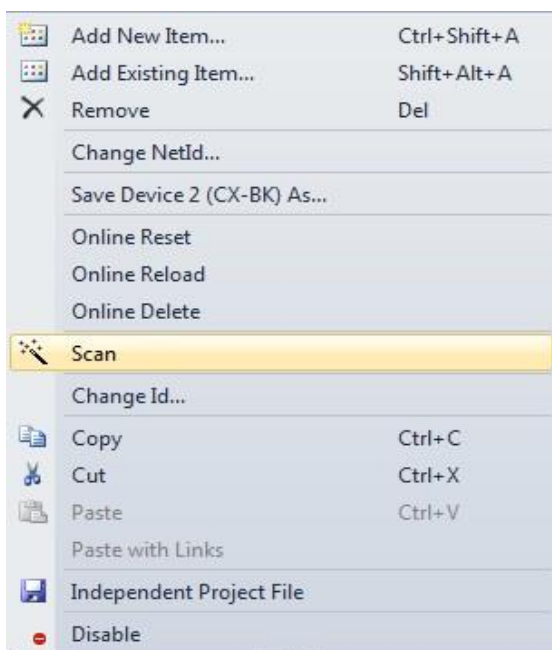


Abbildung 40: Scannen nach Bus-Klemmen

Die gefundenen Klemmen werden im Editor angezeigt (Abbildung 41) und sind farblich je nach Art der Klemme markiert. Die Ein- und Ausgängen können im Online-Modus ausgelesen und den entsprechenden Tastern bzw. Ausgängen zugewiesen werden. Hier ist wichtig die zuvor angesprochene Beschriftung beizubehalten, um den richtigen Ein- und Ausgängen die entsprechende Variable zuordnen zu können.

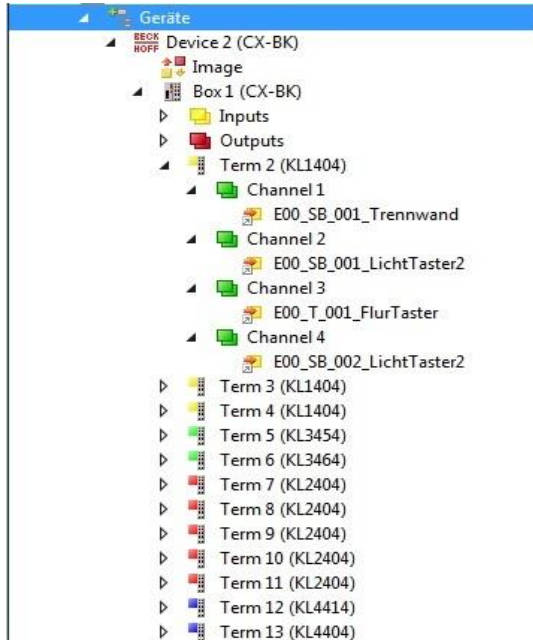


Abbildung 41: Angeschlossene Bus-Klemmen

Sind die Ein- und Ausgänge mit den entsprechenden Variablen verknüpft und das Kompilieren ergab keine Fehler, kann die Konfiguration geladen und getestet werden.

Bei der Aktivierung der Konfiguration (Abbildung 42) wird die Verlinkung der Variablen mit den Ein- und Ausgängen überprüft und auf den Controller CX9020 übertragen. Bei erfolgreicher Übertragung ist es nun möglich, sich einzuloggen und das übertragene Programm auf dem CX9020 zu starten (Abbildung 43).

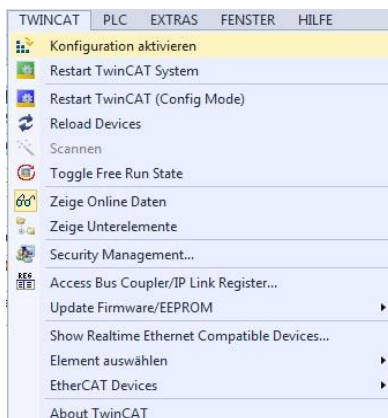


Abbildung 42: Konfiguration aktivieren

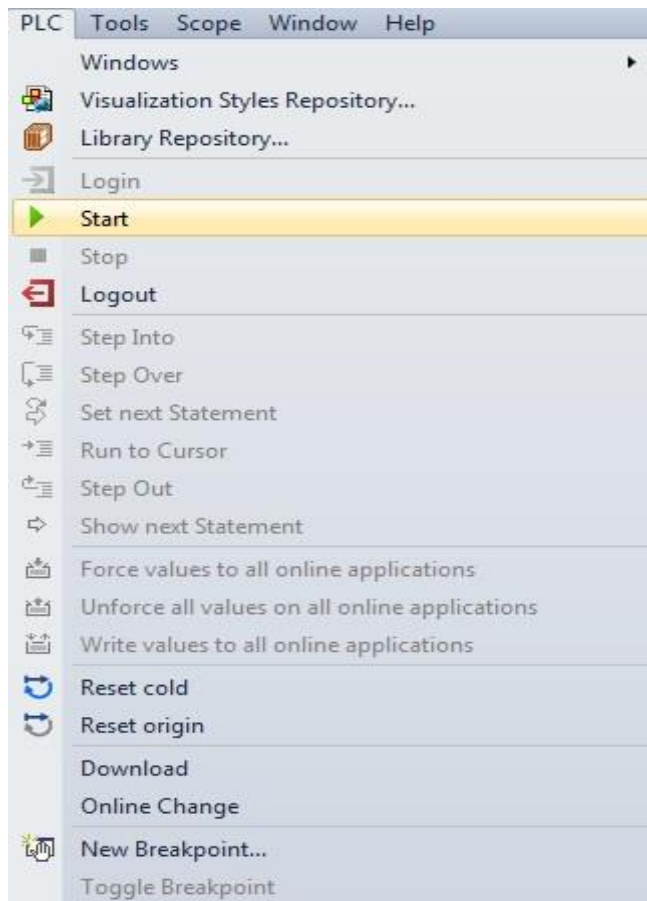


Abbildung 43: Starten des Programms auf CX9020

Ist das geschriebene Programm ordnungsgemäß auf dem Controller übertragen, wird beim Betätigen eines Tasters angezeigt, welche Variable den Wert True (Ein) oder False (Aus) annimmt.

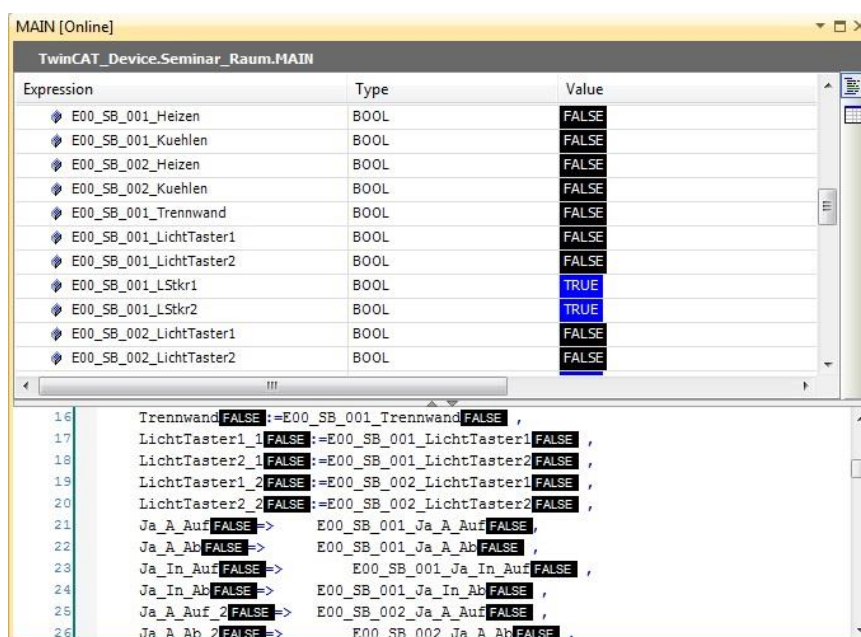


Abbildung 44: Programmausführung

7.4 Realisierung der Jalousiefunktion

Die Schaltfunktionen der Jalousie wie Hoch, Runter und Stopp werden über ein 4-fach-Funktaster realisiert. Um diesen Funktaster ins System einzubetten, wird ein EnOcean-Receiver in Verbindung der seriellen Schnittstellenklemme verwendet.

Durch die bereits mitgelieferten Bibliotheken kann hier auf ein EnOcean-Receiver-Baustein zurückgegriffen werden, um die Funksignale zu verarbeiten. Es wurde ein FB_EnOceanReceive (Abbildung 45) mit den entsprechenden Ein- und Ausgängen angelegt. Der Baustein EnOceanPTM200 (Abbildung 46) wird nur in der Funktion aufgerufen, in der EnOcean-Funktaster verwendet werden sollen.

Die Ein- und Ausgangvariablen *EnOcean_DataIn* und *EnOcean_DataOut* müssen deklariert werden, um die Kommunikation mit dem EnOcean-Adapter zu gewährleisten.

```
EnOcean_DataIn AT %I*:ST_EnOceanInData;
EnOcean_DataOut AT %Q*:ST_EnOceanOutData;
```

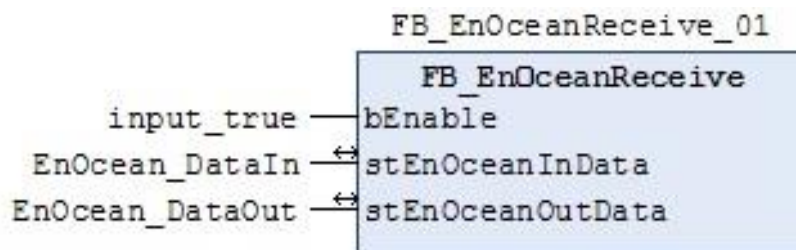


Abbildung 45: EnOceanReceive

Um die Funksignale des 4-fach-Funktaster auszuwerten zu können, muss ein weiterer Baustein eingebettet werden, der diese Funktion übernimmt.

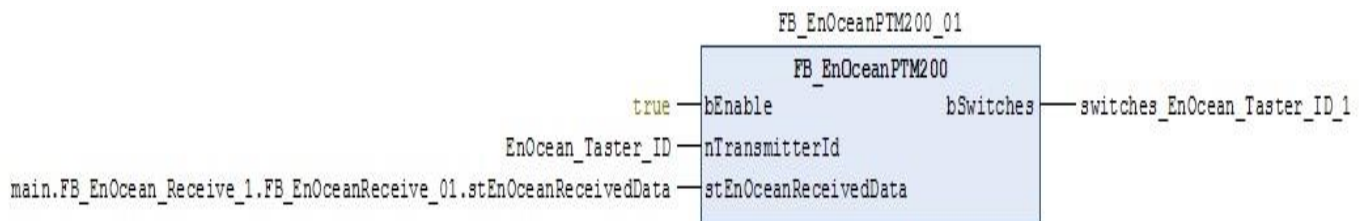


Abbildung 46: EnOceanPTM200

```
EnOcean_Taster_ID_1:UDINT;
switches_EnOcean_Taster_ID_1:ARRAY[0..3] OF BOOL;
```

Für die Deklaration wird die ID des Funksenders benötigt und der jeweiligen Instanz übergeben. Die Ausgangsvariable muss als Array deklariert werden, da bei Betätigung des Tasters der Zustand *True* auf ein Item des Arrays gelegt wird. Die Null in den eckigen Klammern der Zuweisung gibt an, welche Taste hier die Funktion auslösen soll.

```
switches_EnOcean_Taster_ID_1[0]
```

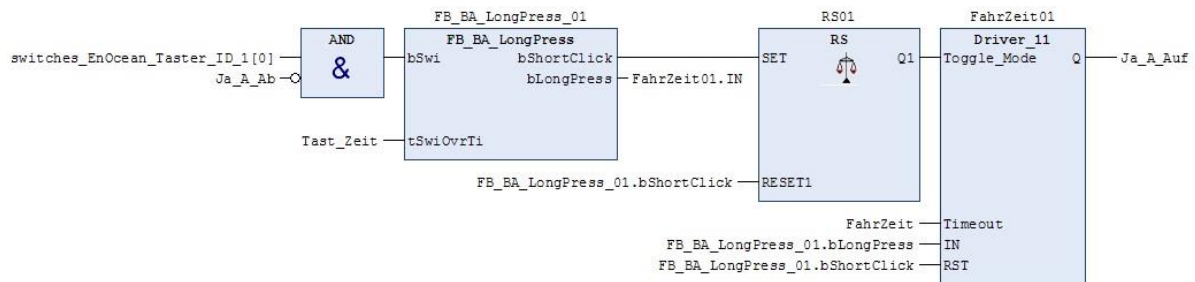


Abbildung 47: JalousieSteuerung Auf

In Abbildung 47 ist die Programmierung für die Auffahrt der Außenjalousie zu sehen. Simuliert wird die Fahrzeit der Jalousie mit Hilfe eines Funktionsbaustein Driver, welcher aus einer externen Bibliothek aufgenommen wurde. [OSCAT]

Bei der weiteren Programmierung für die anderen Fahrfunktionen wurden die Bausteine kopiert, eingefügt und sinngemäß beschriftet. Die Zuweisung, wie oben schon erwähnt, musste nur mit einer Änderung der Item-Zahl in den eckigen Klammern dahin gehend verändert werden, um mit einer weiteren Taste des Funktasters die Funktion zu starten.

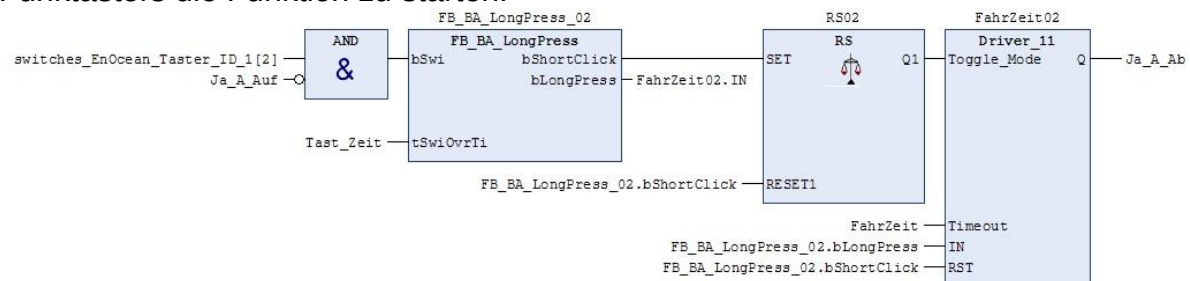


Abbildung 48: JalousieSteuerung Ab

Damit die Kommunikation zwischen Adapter und SPS funktioniert, müssen die empfangenden Signale an die Ein- und Ausgänge übertragen werden. Dies wird im Projektmappen-Editor durchgeführt. Die Ein- und Ausgangssignale müssen den Variablen EnOcean_DataIn bzw. EnOcen_DataOut zugewiesen werden. Mit dem Befehl Multi-Verknüpfung (Abbildung 49) kann die Zuweisung ausgeführt werden.

Name	Online	Type	Größe	> Adre...	Ein/A...	User ID	Verknüpft mit
Ser. State	X	USINT	1.0	88.0	Einga...	0	Jalousiesteuerung.EnOc...
Data In 0	X	USINT	1.0	89.0	Einga...	0	Jalousiesteuerung.EnOc...
Data In 1	X					0	Jalousiesteuerung.EnOc...
Data In 2	X					0	Jalousiesteuerung.EnOc...
Data In 3	X					0	Jalousiesteuerung.EnOc...
Data In 4	X					0	Jalousiesteuerung.EnOc...
Data In 5	X					0	Jalousiesteuerung.EnOc...
Data In 6	X					0	Jalousiesteuerung.EnOc...
Data In 7	X					0	Jalousiesteuerung.EnOc...
Data In 8	X					0	Jalousiesteuerung.EnOc...
Data In 9	X					0	Jalousiesteuerung.EnOc...
Data In 10	X					0	Jalousiesteuerung.EnOc...
Ser. Ctrl	X					0	Jalousiesteuerung.EnOc...
Data Out 0	X					0	Jalousiesteuerung.EnOc...
Data Out 1	X					0	Jalousiesteuerung.EnOc...
Data Out 2	X					0	Jalousiesteuerung.EnOc...
Data Out 3	X					0	Jalousiesteuerung.EnOc...
Data Out 4	X					0	Jalousiesteuerung.EnOc...
Data Out 5	X					0	Jalousiesteuerung.EnOc...
Data Out 6	X					0	Jalousiesteuerung.EnOc...
Data Out 7	X					0	Jalousiesteuerung.EnOc...
Data Out 8	X					0	Jalousiesteuerung.EnOc...
Data Out 9	X					0	Jalousiesteuerung.EnOc...
Data Out 10	X					0	Jalousiesteuerung.EnOc...

Abbildung 49: Multi-Verknüpfung

7.5 Realisierung der Heizungssteuerung

Für die Programmierung der FB_HeizungsSteuerung (Abbildung 50) wurde ein Hysterese Baustein verwendet. Zudem kann durch Tastendruck von Winter auf Sommer umgeschaltet werden. Im Winterbetrieb wird auf die gewünschte Raumtemperatur aufgeheizt und im Sommerbetrieb gekühlt. Die Temperaturen werden über 2 PT100 Temperatursensoren erfasst. In Abbildung 50 ist die Programmierung für die Heizfunktion zusehen.

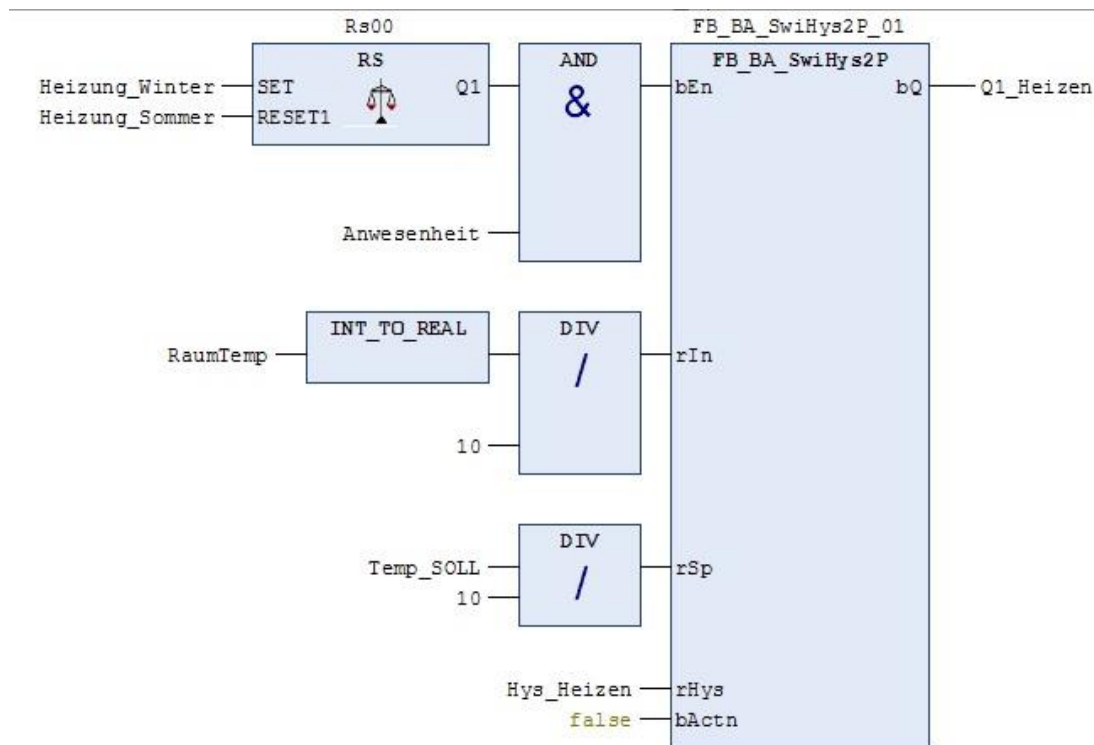


Abbildung 50: Heizungssteuerung

7.6 Realisierung der Flurlichtsteuerung

Für die Programmierung der FB_Flurlicht (Abbildung 51) wurde ein Baustein mit der Funktion Toggle (Umschalten) verwendet. Durch die Flurtaster wird die Flurbeleuchtung für eine eingestellte Zeit eingeschaltet fällt der Lux-Wert unter einem festgelegtes Niveau wird die Flurbeleuchtung eingeschaltet und die Tastenfunktion deaktiviert. Die Zeitschaltung erfolgt durch den Funktionsbaustein Driver. [OSCAT]

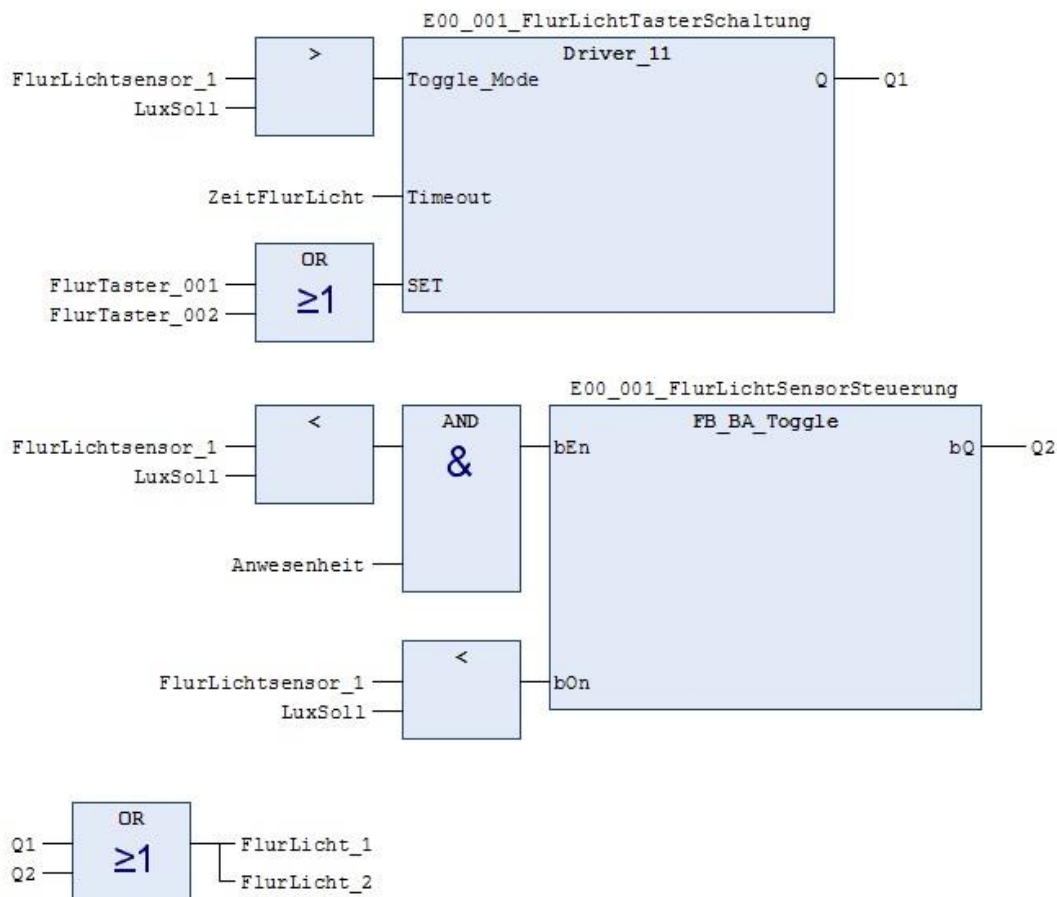


Abbildung 51: Flurlichtsteuerung

7.7 Anlegen globaler Variablen

Bei der Programmierung muss man sich im Vorfeld klarmachen, welche Variablen lokal zu Verfügung stehen sollen oder aber über das ganze Projekt von Bedeutung sind. In TwinCat kann für globale Variablen eine Liste erstellt werden, die dann im gesamten Projekt vorhanden sind. Wichtig ist hier zu erwähnen, dass bei der Variablenvergabe immer nur eine Zuordnung möglich ist. Um eine globale Variablenliste zu erstellen, erfolgt ein Klick in den Projektmappen-Editor auf globale Variablen (Abbildung 52) und es erscheint ein Fenster, in dem man die gewünschten Variablen deklarieren kann. (Abbildung 53)

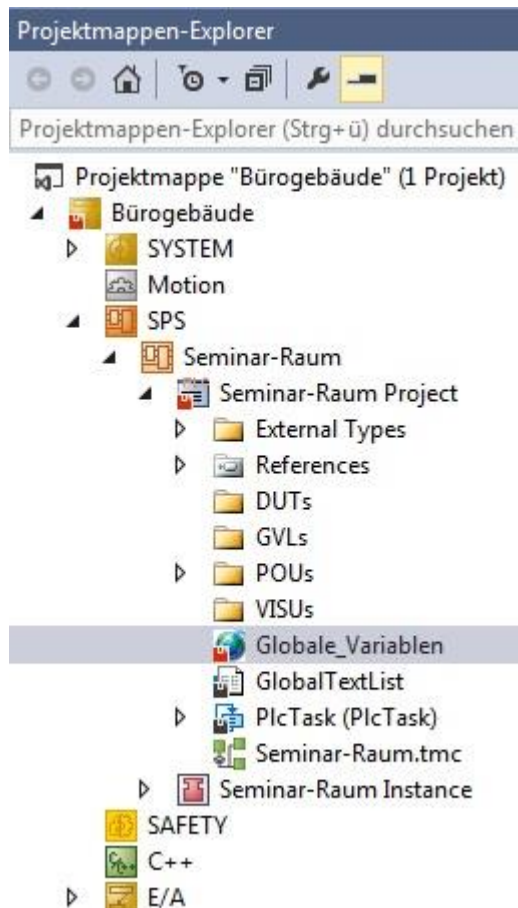


Abbildung 53: Globale-Variablen

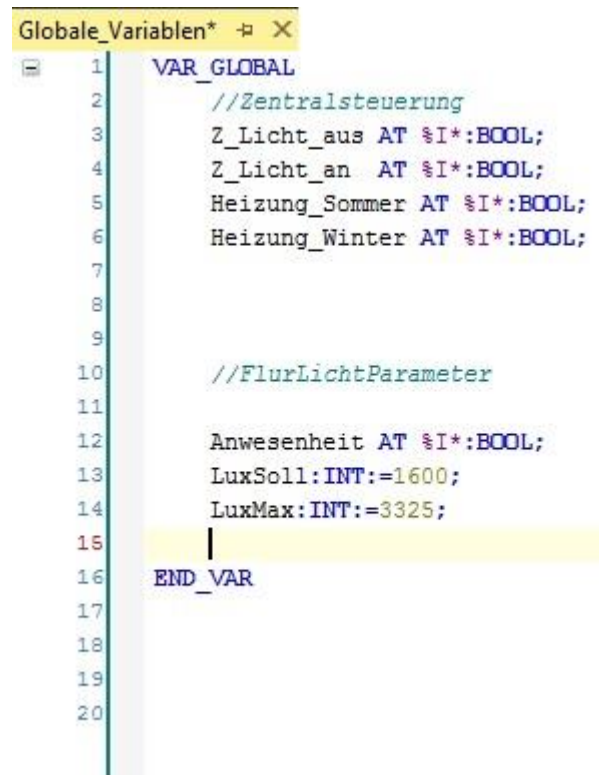


Abbildung 52: Globale Variablenliste erstellen

7.8 TwinCat 3 Lizenz

Beckhoff hat mit TwinCat 3 ein auftretendes Problem elegant gelöst. Es kann nach der Installation eine 7 Tageslizenz aktiviert und auch nach Ablauf der Zeit erneut aktiviert werden. Um die Testlizenz zu aktivieren, klickt man im Projektmappen-Editor mit einem Doppelklick auf Lizenz und es öffnet sich ein neues Fenster. In diesem Fenster kann man sowohl eine gekaufte Lizenz aktivieren oder wie schon erwähnt die Testlizenz aktivieren. Abbildung 54 zeigt, wie das aussieht.

Bestell Nr	Lizenz	Instanzen	Current Status
TC1200	TC3 PLC	CPU Lizenz	missing
TF4110	TC3 Temperature-Controller	CPU Lizenz	missing
TF8040	TC3 Building Automation	CPU Lizenz	missing

Abbildung 54: Lizenz aktivieren

Hat man den Button für die Testlizenz angeklickt, erscheint eine Aufforderung eine Buchstabenkombination (Abbildung 55) einzugeben. Wurde diese erfolgreich eingegeben, ist die Lizenz für die nächsten 7 Tage aktiv und TwinCat bestätigt das mit einer Meldung. (Abbildung 56)

Abbildung 55: Sicherheitscode

Abbildung 56: 7 Tages Lizenz

7.9 Bootprojekt erstellen

Damit das erstellte Programm auch auf dem Controller CX9020 bleibt, muss ein Bootprojekt erzeugt werden. Dazu klickt man im Projektmappen-Editor auf das erstellte PLC Projekt mit einem Rechtsklick und markiert im Fenster Autostart Boot Projekt (Abbildung 57). Zusätzlich kann über die Einstellungen des CX9020 ein Autologin eingestellt werden, was dazu führt, dass, wenn der Controller aus- und wieder eingeschaltet wird, das Programm wieder direkt aktiv ist.

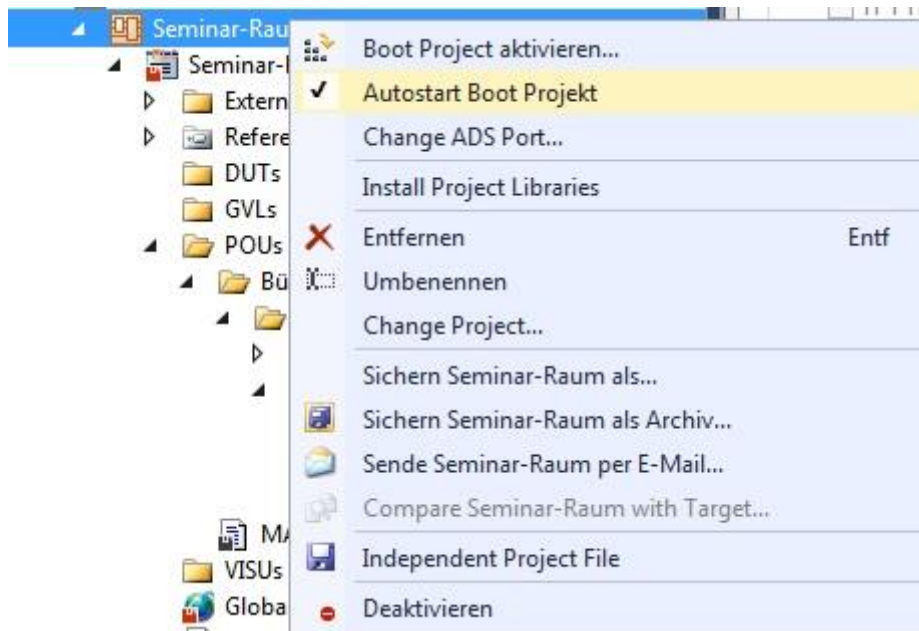


Abbildung 57: Boot Projekt

8. Vergleich TwinCat 2 mit TwinCat 3

Die Vorteile von TwinCat 3 liegen auf der Hand. Bei der Bedienung fällt einem nach sehr kurzer Zeit auf, dass kein Wechsel zwischen PLC Manager und PLC Control stattfindet. Bei der Bearbeitung ist deutlich geworden, dass TwinCat 3 stabiler läuft als die Version 2. Gab es mit der älteren Version noch unerwartetes Schließen bis hin zum totalen Aufhängen der Software, ist das Arbeiten mit der Version 3 doch um ein vielfaches angenehmer. Was zuerst negativ aufgefallen ist, war der geringe Platz bzw. die Aufteilung der Fenster. Doch kann man einzelne Fenster aus der Umgebung lösen und auf die gesamte Größe des Monitors setzen, was bei größeren bzw. längeren Programmen eine Hilfe ist. Des Weiteren ist es positiv zu bewerten, dass mit der Installation von TwinCat 3 Bibliotheken gleich mitgeliefert werden. Des Weiteren ist TwinCat 3 im Microsoft Visual Studio integriert was dem Anwender die Möglichkeit bietet auf die Hochsprachen C# und C++ zu zugreifen. Eine weitere Erneuerung ist die Anbindung an MATLAB/Simulink, die dem Anwender noch mehr Möglichkeiten bietet. Mit TwinCat 3 ist es auch möglich ein Programm welches noch mit TwinCat 2 auf einem Kern ausgeführt wird nun auf mehrere Kerne dupliziert werden und erreicht somit eine vielfache Performance. [DK] Die TwinCat 2-Programme können natürlich übernommen und in TwinCat 3 importiert werden.

Zu erwähnen ist noch, dass eine Installation von Microsoft Visual Studio nur dann erforderlich ist, wenn mit den oben angesprochenen Hochsprachen programmiert werden soll. Ansonsten funktioniert TwinCat 3 als eigenständiges Engineering-Tool. [ER]

9. Aussichten

Mit TwinCat 3 hat Beckhoff eine sehr starke Software entwickelt, die es weithin zu untersuchen gilt und zwar hinsichtlich Visualisierung und Verknüpfung zweier Controller. Man sollte auch über eine Überarbeitung des Demo-Koffers nachdenken, da es Klemmen gibt, die nicht untersucht werden können und mehr Programmierspielraum bieten. Ein wichtiger Punkt wäre z.B. das Erfassen von Verbrauchern hinsichtlich Smart Metering. Beckhoff bietet hier als Beispiel die Messklemme *KL3403* an. Es müsste zwar mit Stromwandlern gearbeitet werden, da diese Klemme nur für 1 Ampere ausgelegt ist, aber das sollte kein Problem darstellen. Für weitere Untersuchungen wären Dimmer-Klemmen *KL2751* und auch Jalousieklemmen *KL2712* interessant. Ein anderer wichtiger Punkt wäre zu untersuchen, wie sich das Grundmodul CX9020 in Kopplung mit einem weiteren CX9020 verhält und ob sich somit eine leichte Erweiterbarkeit realisieren lässt.

10 Fazit

Abschließend ist zu sagen, dass Beckhoff aus Fehlern lernt und seine Produkte immer weiter verbessert. Sicherlich ist darauf zu achten, für welche Programmiersprache man sich entscheidet, da es sich im Verlauf der Bearbeitung gezeigt hat, dass umso größer das Projekt wird, es unübersichtlicher wird mit der Programmierung mit Funktionsbausteinen. Aber mit einer durchdachten Variablenvergabe und auf das Projekt abgestimmte Bezeichnung lässt sich flexibel und schnell programmieren. Dies funktioniert nur, wenn die Variablenbezeichnung eindeutig hervorgeht. Die verwendete Bezeichnung *E00_SB_001_LichtTaster1* lässt sich ohne viel Aufwand für die nächste Etage ändern und zwar in *E01_SB_001_LichtTaster1*. Für Untergeschosse wird das *E* einfach durch ein *U* ersetzt. Die Programmierung wurde nach dem Prinzip der Instanziierung durchgeführt, was ein hohes Maß an Flexibilität bietet. Angenommen nach der Fertigstellung des Programms werden durch Umbauarbeiten neue Räume gebaut dann kann durch Aufrufen der entsprechenden Instanz die „neuen“ Büroräume unkompliziert in das Programm aufgenommen werden. Zusätzlich ist für eine flexible Programmierung wichtig, dass auch andere Bussysteme eingebettet werden können. Beckhoff bietet im Rahmen des Bus-Klemmen-Angebots die Möglichkeit an, zahlreiche Systeme einzubinden. Um einige interessante Beispiele zu nennen: KNX/EIB, EnOcean, DMX. Mit solch einer Vielfalt an Möglichkeiten steht einer flexiblen Programmierung nichts mehr im Wege. Mit der Software TwinCat in der Version 3.0 geht Beckhoff in die richtige Richtung und hat hier ein stabiles Programm entwickelt. Leider konnte im Rahmen der Bearbeitung der Thesis nicht alles getestet werden. Was zuerst einen negativen Eindruck machte, ist die Integration in das Microsoft Visual Studio. Beckhoff hat sich dafür entschieden, da diese Entwicklungsumgebung weit verbreitet und auch bei vielen beliebt ist. Zusätzlich ermöglicht diese Integration die Nutzung einiger Hochsprachen. Aus der Sicht von Beckhoff wird das Visual Studio zunächst nur als Rahmen für die eigenen Plug-Ins, Konfiguration und Programmierung von SPS-Steuerungen verwendet. [ER] Der entstandene negative Eindruck verfliegt aber nach einiger Zeit. Da nicht mehr zwischen PLC Control und PLC Manager gewechselt werden muss, bekommt man den Eindruck vom flüssigeren Arbeiten.

11. Anhang

11.1 Literaturverzeichnis

- [CX9020] Produktdatenblatt der Firma Beckhoff
- [BP] Betriebliche Praxis Arbeit, Michael Oldenburg
- [BK] Homepage der Firma Beckhoff www.Beckhoff.de
- [DK] Die Konvergenz der Technologien Interview mit dem Produktmanager Dr. Josef Papenfort
<http://www.all-electronics.de/interview-mit-dr-josef-papenfort-produktmanager-twincat-bei-beckhoff/>
- [ER] Engineering und Runtime unter 64 Bit Interview mit Stefan Hoppe
- [OSCAT] Open Source Community for Automation Technology
<http://www.oscat.de/>

11.2 Abbildungsverzeichnis

ABBILDUNG 1: IPC-BEREICH	8
ABBILDUNG 2: DER I/O BEREICH	9
ABBILDUNG 3: DER MOTION-BEREICH.....	9
ABBILDUNG 4: DEMO-KOFFER-BECKHOFF	10
ABBILDUNG 5: CPU GRUNDMODUL CX9020	11
ABBILDUNG 6 DIGITAL-EINGANG KL 1104.....	12
ABBILDUNG 7 ANALOG-EINGANG KL 3454.....	13
ABBILDUNG 8: 4-KANAL-EINGANGSKLEMME KL3204	14
ABBILDUNG 9: DIGITALE AUSGANGSKLEMME KL 2404.....	15
ABBILDUNG 10: SERIELLE SCHNITTSTELLE KL 6021	16
ABBILDUNG 11: BUSENDKLEMME KL 9010	17
ABBILDUNG 12 SCHAUBILD INSTANZIIERUNG.....	23
ABBILDUNG 13: TWINCAT3 START	25
ABBILDUNG 14: PROJEKT ANLEGEN.....	26
ABBILDUNG 15: PROJEKTMAPPEN-EXPLORER.....	27
ABBILDUNG 16: NEUES ELEMENT	27
ABBILDUNG 17: PLC PROJEKT HINZUFÜGEN	28
ABBILDUNG 18: PROJEKTMAPPEN-EXPLORER.....	28
ABBILDUNG 19: POU ERSTELLEN	29
ABBILDUNG 20: POU FB_LICHTSTEUERUNG.....	30
ABBILDUNG 21: FUNKTIONSBAUSTEINE HINZUFÜGEN	31
ABBILDUNG 22: FUNKTIONSBAUSTEIN TOGGLE.....	31
ABBILDUNG 23: VARIABLEN.....	32
ABBILDUNG 24: FUB-BIBLIOTHEK	33
ABBILDUNG 25: LICHTSTEUERUNG EINZELN	34
ABBILDUNG 26: LICHTSTEUERUNG ZUSAMMEN	34
ABBILDUNG 27: ZUWEISUNG VON VARIABLEN AN EINEN FUB	35
ABBILDUNG 28: AUSGÄNGE LICHTSTEUERUNG.....	35
ABBILDUNG 29: FB_LICHTSTEUERUNG	36
ABBILDUNG 30: EINGABEHILFE	36
ABBILDUNG 31: INSTANZAUF RUFE.....	37
ABBILDUNG 32: INSTANZ LICHTSTEUERUNG.....	37
ABBILDUNG 33: POU MAIN.....	38
ABBILDUNG 34: AUFBAU DER POU MAIN	38
ABBILDUNG 35: DEKLARATION	38
ABBILDUNG 36: ERNEUTER AUFRUF DER INSTANZ.....	39
ABBILDUNG 37: ZIELSYSTEM	40
ABBILDUNG 38: BROADCASTSUCHE	40
ABBILDUNG 39: IP-KONFIGURATION	41
ABBILDUNG 40: SCANNEN NACH BUS-KLEMMEN	41
ABBILDUNG 41: ANGESCHLOSSENE BUS-KLEMMEN	42
ABBILDUNG 42: KONFIGURATION AKTIVIEREN	42
ABBILDUNG 43: STARTEN DES PROGRAMMS AUF CX9020	43
ABBILDUNG 44: PROGRAMMAUSFÜHRUNG.....	43
ABBILDUNG 45: ENOCEANRECEIVE	44

ABBILDUNG 46: ENOCEANPTM200.....	44
ABBILDUNG 47: JALOUSIESTEuerung AUF	45
ABBILDUNG 48: JALOUSIESTEuerung AB	45
ABBILDUNG 49: MULTI-VERKNÜPFUNG.....	45
ABBILDUNG 50: HEIZUNGSTEuerung	46
ABBILDUNG 51: FLURLICHTSTEuerung	47
ABBILDUNG 52: GLOBALE VARIABLENLISTE ERSTELLEN.....	48
ABBILDUNG 53: GLOBALE-VARIABLEN.....	48
ABBILDUNG 54: LIZENZ AKTIVIEREN	49
ABBILDUNG 55: SICHERHEITSCODE	49
ABBILDUNG 56: 7 TAGES LIZENZ.....	49
ABBILDUNG 57: BOOT PROJEKT	50

11.3 Tabellenverzeichnis

TABELLE 1: MÖGLICHE VARIABLENVERGABE	24
TABELLE 2: LEGENDE.....	24