Fachhochschule
Dortmund
University of Applied Sciences and Arts

Masterthesis
zur Erlangung des akademischen Grades
Master of Engineering
im Masterstudiengang
Informations- und Elektrotechnik

Entwicklung und Validierung einer Simulationsumgebung mit fernwirk- und stationsleittechnischen Funktionen und IEC 60870-5-104 Kommunikation unter Java

Dardae Fariad

Matrikelnummer 7088180

22. Juni 2018

Erstprüfer: Prof. Dr.-Ing. Kai Luppa
Zweitprüfer: Ing. Jens Flötotto, M.Eng.

Danksagung

An erster Stelle möchte ich meinem Betreuer Prof. Dr. Kai Luppa danken, der mir die Möglichkeit gegeben hat, meine Projektarbeiten und Abschlussarbeit unter seiner Betreuung anfertigen zu können. Dabei möchte mich nochmals herzlichst für das entgegengebrachte Vertrauen bedanken, das mich in meiner persönlichen Weiterentwicklung sehr gestärkt hat. Bleiben Sie so begeistert von der Netzleit- und Fernwirktechnik, denn diese Begeisterung schwappt hoffentlich bei einem weiteren Studenten über.

Des Weiteren möchte ich mich herzlich bei Jens Flötotto bedanken, der mir zu jeder Zeit mit fachlichem Rat beistand. Durch die von ihm und Prof. Dr. Kai Luppa gelebte Stimmung im Labor für Energieautomation war es mir eine Freude Mitglied dieser Gemeinschaft gewesen zu sein. Diese Zeit möchte ich nicht missen. Dazu beigetragen haben insbesondere meine Kommilitonen, bei denen ich mich für ihre motivierenden Worte für mich von ganzem Herzen bedanken möchte.

Tief dankbar bin ich meiner Freundin Nadine Tabaczynski, die für mich stets immer ein offenes Ohr hatte und mit der ich immer anregende Diskussionen führen konnte, die mir bei der Anfertigung dieser Arbeit nützlich waren.

Der größte Dank gilt meinen Eltern und Geschwistern. Besonders danke ich meiner Schwester Jinane, die mir während meines gesamten Studiums immer den Rücken gestärkt und mich in meinen Entscheidungen bestärkt hat.

Erklärung

Hiermit versichere ich an Eides statt, dass die von mir vorgelegte Prüfungsleistung selbständig und ohne unzulässige fremde Hilfe erstellt worden ist. Alle verwendeten Quellen sind in der Arbeit so aufgeführt, dass Art und Umfang der Verwendung nachvollziehbar sind.

Dortmund, 21.6.2018

Unterschrift

Abstract

The master thesis *Development and Validation of a Simulation Environment with Remote Control and Station Control Technology Functions and IEC 60870-5-104 Communication under Java* covers the implementation of a simulation environment for the demonstration of remote control and station control in combination with an IEC 60870-5-104 communication. The simulation environment is defined as an IEC 60870-5-104 server. After the station initialization and the transmission control, the simulation environment can receive and analyze telegrams in the control direction and initiate corresponding remote control and station control processes. In the reporting direction, spontaneous process changes or changes triggered by control processes are to be implemented by generating and transmitting telegrams. The simulation environment can be used to demonstrate processes triggered by IEC 60870-5-104 communication within a remote control device as well as by means of a process simulation.

Die Masterthesis *Entwicklung und Validierung einer Simulationsumgebung mit fernwirk- und stationsleittechnischen Funktionen und IEC 60870-5-104 Kommunikation unter Java* umfasst die Implementierung einer Simulationsumgebung zur Veranschaulichung fernwirk- und stationsleittechnischer Vorgänge in Kombination mit einer IEC 60870-5-104 Kommunikation. Die Simulationsumgebung ist dabei als IEC 60870-5-104-Server definiert. Nach der Stationsinitialisierung und der Übertragungssteuerung kann die Simulationsumgebung Telegramme in Steuerungsrichtung empfangen, analysieren und entsprechende fernwirk- und stationsleittechnische Vorgänge auslösen. In Melderichtung sind spontane Prozessänderungen oder durch Steuervorgänge ausgelöste Änderungen durch Generierung und Übertragung von Telegrammen umzusetzen. Mit der Simulationsumgebung können durch eine IEC 60870-5-104 Kommunikation ausgelöste Vorgänge innerhalb eines Fernwirkgerätes sowie anhand einer Prozesssimulation demonstriert werden.

Inhaltsverzeichnis

Abkürzungen

APDU	Application Protocol Data Unit
ASDU	Application Service Data Unit
BL	Blocked
CAASDU	Common Adress of Application Service Data Unit
COT	Cause of Transmission
CPU	Central Processing Unit
CSV	Character Separated Value
DCS	Double Command State
DIQ	Double-Point Information with Quality Descriptor
DPI	Double-Point Information
FWG	Fernwirkgerät
GUI	Graphical User Interface
I	Numbered Information Transfer
IEC	International Electrotechnical Commission
IOA	Informationsobjektadresse
IP	Internet Protocol

IV	Invalid
LED	Light-Emitting Diode
NT	Not Topical
OV	Overflow
qds	Quality Descriptor (separate object)
QOS	Qualifier of Set-Point Command
S	Numbered Supervisory Functions
SB	Substituted
SCADA	Supervisory Control and Data Acquisition
SIQ	Single-Point Information with Quality Descriptor
SPI	Single-Point Information
TCP/IP	Transmission Control Protocol/Internet Protocol
TK	Typkennung
U	Unnumbered Control Functions
UML	Unified Modeling Language

1 Einleitung

Die Stations- und Netzleittechnik sowie Fernwirktechnik ermöglicht den Betrieb und somit die Überwachung und Steuerung von Schaltanlagen, Umspannwerken und Netzen. Notwendige Prozessdaten werden von der Fernwirktechnik erfasst und über Kommunikationsnetze an zentrale Stellen übertragen. Abbildung 1.1 zeigt die übergeordnete Leitebene, welche über standardisierte Protokolle eine Kommunikationsverbindung zwischen dem Prozess und dem Netzleitsystem über fernwirktechnische Einrichtungen aufbauen kann.

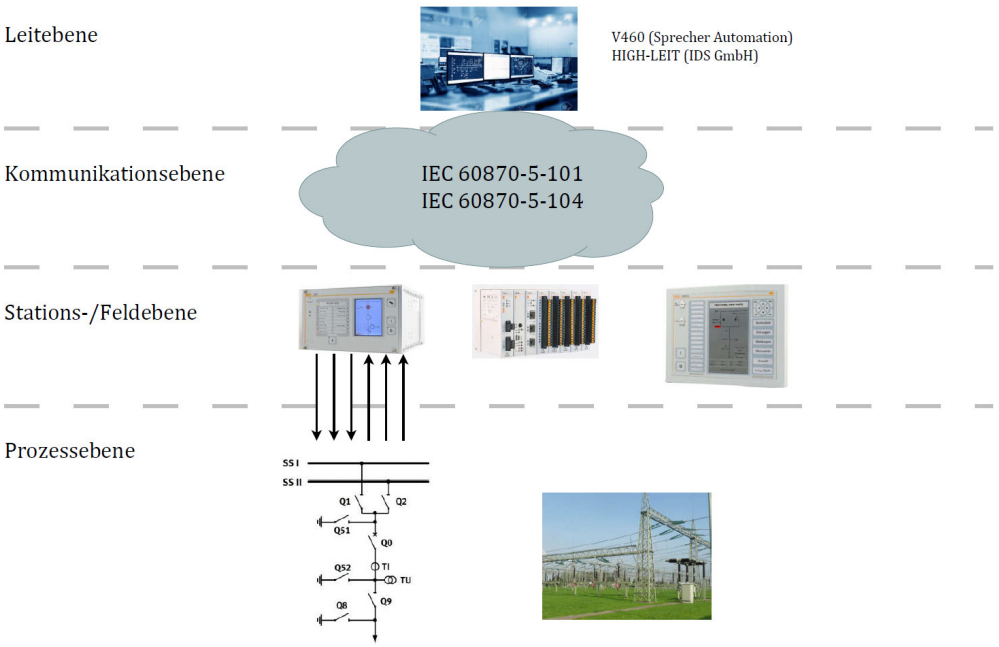


Abbildung 1.1: Leitebenen

Das Protokoll International Electrotechnical Commission (IEC) 60870-5-101 [1] besitzt allgemeine Fähigkeiten im Rahmen von Supervisory Control and Data Acquisition (SCADA)-

Anwendungen und überträgt die Daten über serielle Verbindungen. Gleiche SCADA-Fähigkeiten besitzt das Übertagungsprotokoll IEC 60870-5-104 [2]. Dieser Standard erlaubt die Kommunikation nur über ein Transmission Control Protocol/Internet Protocol (TCP/IP)-Netzwerk. Dieses Fernwirkprotokoll ermöglicht es einer zentralen Netzleitstelle mit Hilfe der Stations- und Feldleittechnik weit verteilte Prozesse zu überwachen und zu steuern.

1.1 Aufgabenstellung und Ziel der Arbeit

Im Rahmen dieser Masterthesis *Entwicklung und Validierung einer Simulationsumgebung mit fernwirk- und stationsleittechnischen Funktionen und IEC 60870-5-104 Kommunikation unter Java* soll eine bereits entwickelte Graphical User Interface (GUI) zur Simulation eines Fernwirkgerät (FWG) und der entsprechenden Funktionalitäten um eine *Parametrierung* und IEC 60870-5-104 Kommunikation erweitert werden. Dabei soll die Simulationsumgebung als Server mit einem beliebigen Client über das Kommunikationsprotokoll IEC 60870-5-104 kommunizieren. Für die Umsetzung dieser Kommunikation ist es erforderlich, den bestehenden statischen Prozess durch Informationsobjekte abzubilden.

Der in der Arbeit *Entwicklung einer grafischen Oberfläche zur Simulation eines Fernwirksystems unter JavaFX* [3] entwickelte Simulator zeigt die wesentlichen Vorgänge innerhalb eines FWG auf, in dem Ursache und Auswirkung als zusammenhängender Vorgang demonstriert werden. Der Simulator aus Abbildung 1.2 besteht aus den Komponenten FWG, *Prozesssimulation* und *Vor-Ort-Steuerung* als separate GUI. Das in Abbildung 1.2 mittig dargstellte FWG stellt ein Kompaktfernwirkgerät dar. Es besteht aus einer Spannungsversorgung, einer zentralen Prozessoreinheit, vier digitalen Eingaben, zwei digitalen Ausgaben, einer analogen Eingabe und einer analogen Ausgabe. Mit diesem FWG wird ein Schaltfeld aus einem Leistungsschalter, einem Erdungsschalter sowie einem Messwandler und einer Sollwert-Anzeige abgebildet.

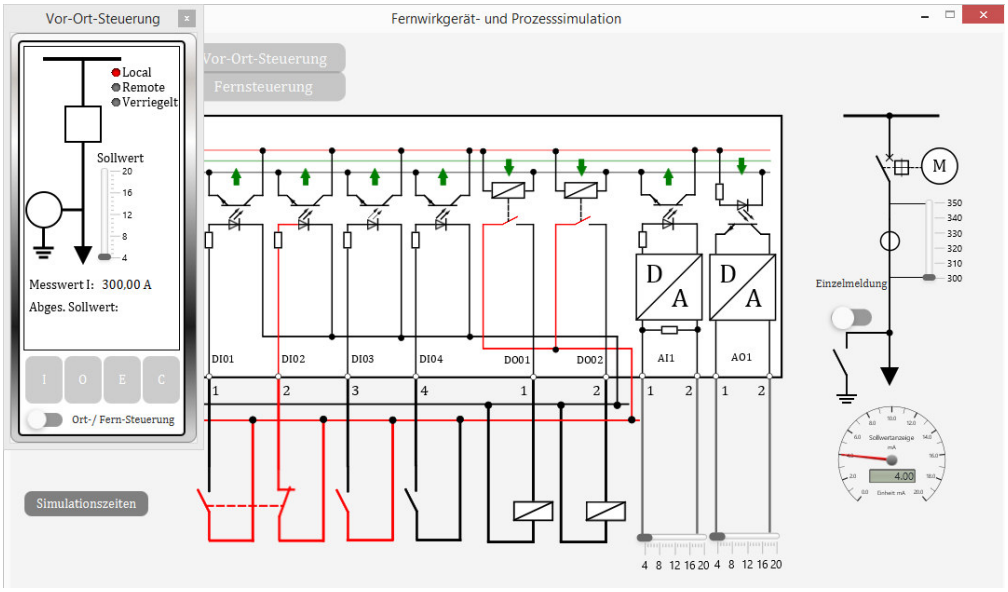


Abbildung 1.2: grafischen Oberfläche zur Simulation eines Fernwirksystems

Mit der links positionierten *Vor-Ort-Steuerung* kann der Leistungsschalter angesteuert und ein Zustandswechsel initiiert werden. Dabei beginnt bei einer Befehlsgabe an den Leistungsschalter die Simulation im FWG. Die internen Relais der digitalen Ausgabe schließen zunächst für die Dauer der Gerätelaufzeit als Simulationszeit. Im Verlaufe dieser Befehlsgabe wechseln die an den digitalen Eingängen angeschlossenen Endlagenschalter ihre Position analog zu der Position des Leistungsschalters in der auf der Benutzeroberfläche rechts angeordneten *Prozesssimulation*. Über Optokoppler wird der aktuelle Zustand der analog Eingabe über den Bus an die *CPU* der zentralen Prozessoreinheit übertragen. Der entsprechende Prozesszustand wird in der *Vor-Ort-Steuerung* angezeigt. Mittels eines Buttons kann der Zustand des Erdungsschalters in der *Prozesssimulation* verändert werden. Über einen im Abgang angeschlossenen *Slider* erfolgt die Messwertsimulation. Bei Bewegung des *Sliders* wird aktuelle Messwert in der *Vor-Ort-Steuerung* angezeigt. Mit der Sollwert-Anzeige in der *Prozesssimulation* kann der von der *Vor-Ort-Steuerung* abgesetzte Sollwert-Stellbefehl angezeigt werden.

Mit dem Gedanken zunächst eine vom Kommunikationsprotokoll IEC 60870-5-104 losgelöste Steuerung eines Prozesses über die *Vor-Ort-Steuerung* zu realisieren, soll durch diese Arbeit eine *Parametrierung* dieses Prozessdatenumfanges durchgeführt werden und

somit eine Fernsteuerung erfolgen können. In Abbildung 1.3 ist eine Übersicht der Erweiterungen des Simulators skizziert. Dabei ist die vom Fernwirkprotokoll unabhängige Steuerungsmöglichkeit des Prozesses mittels der Vor-Ort-Steuerung rot umrandet.

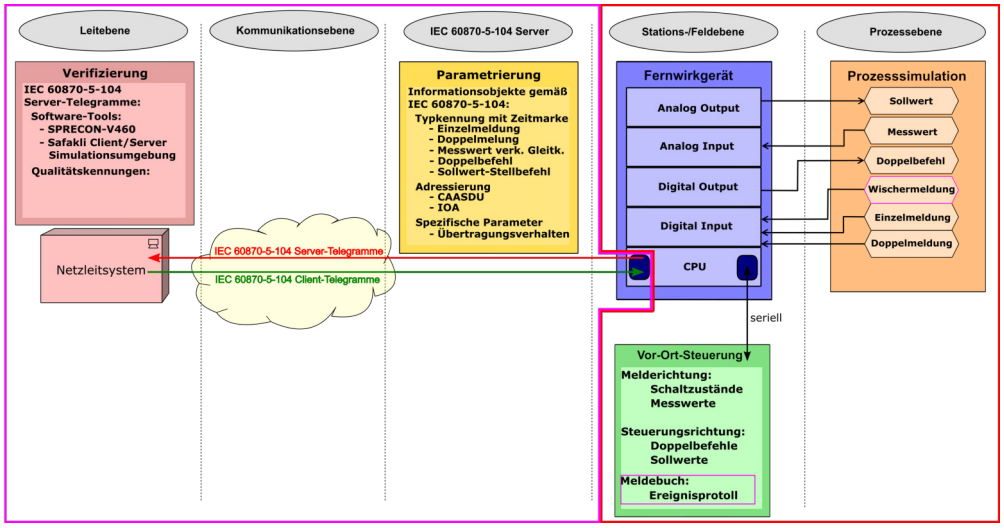


Abbildung 1.3: Übersicht zur Projektaufgabe

Der in magenta umrahmte Bereich stellt die im Rahmen dieser Arbeit umzusetzenden Funktionalitäten. Dabei soll innerhalb der *Prozesssimulation* die Möglichkeit geschaffen werden, eine Wischermeldung zu simulieren. Um alle Prozessänderungen protokollieren zu können, soll innerhalb der *Vor-Ort-Steuerung* ein *Meldebuch* integriert werden. Dabei ist dort tabellarisch die Prozessänderung mit entsprechendem Zeitstempel aufzuführen.

Das Ziel dieser Arbeit ist, dass das FWG in der Stations- und Feldebene die von einer Gegenstation erhaltenen IEC 60870-5-104 Client-Telegramme empfangen, analysieren und ggf. einen entsprechenden Vorgang im Prozess auslösen kann. Zudem soll die Anwendung IEC 60870-5-104 Server-Telegramme generieren, die beispielsweise eine erfolgreiche oder nicht mögliche Befehlsprozedur kommunizieren. Mit der Übertragung von Server-Telegrammen hinsichtlich der Prozessänderungen wird dem Client immer der aktuelle Prozesszustand mitgeteilt.

Für eine Client-/Server-Kommunikation gemäß dem Fernwirkprotokoll IEC 60870-5-104 ist eine *Parametrierung* des FWG erforderlich. Die *Parametrierung* der in Tabelle 1.1

aufgelisteten Datenpunkte soll dabei mittels einer Rangiertabelle durchgeführt werden können. In der Tabelle ist für jede zu implementierende Typkennung (TK) die entsprechende Prozessgröße aufgeführt, dessen Wert oder Zustand abgebildet oder verändert werden soll.

TK	**Prozessgröße**
Doppelbefehl (TK 59)	Ansteuerung Leistungsschalter
Sollwert-Stellbefehl verkürzte Gleitkommazahl (TK 63)	Sollwert-Anzeige
Doppelmeldung (TK 31)	Zustand Leistungsschalter
Einzelmeldung (TK 30)	Zustand Erdungsschalter
Einzelmeldung (TK 30)	Wischermeldung
Messwert verkürzte Gleitkommazahl (TK 36)	Stromwandler

Tabelle 1.1: Prozessabbild als Datenpunktliste

Die TK der zu parametrierenden Informationsobjekte werden alle mit Zeitmarke parametriert. Die Common Adress of Application Service Data Unit (CAASDU) (Statiosadresse) soll als globale Größe parametriert und für jedes Informationsobjekt übernommen werden. Die Parametrierung der Informationsobjektadresse (IOA) soll für jedes Informationsobjekt individuell erfolgen. Weiterhin sollen definierte spezifische Parameter eingestellt werden können, die das Übertragungsverhalten der Informationsobjekte in Überwachungsrichtung bestimmen. Zusätzlich zu der Parametrierung der in Tabelle 1.1 aufgelisteten Informationsobjekte als Abbild des Prozesses sollen vier weitere Systemmeldungen parametriert werden können. Dabei soll hierbei nur die Parametrierung der CAASDU und der IOA möglich sein. Die Übertragung der Systemmeldungen erfolgt durch den entsprechenden Vorgang spontan. Für folgende Ereignisse sind vier Einzelmeldungen der TK 30 in der *Rangiertabelle* zu integrieren:

- für die Umschaltung zwischen Ort-/Fernsteuerung
- als Wischermeldung bei einem Verriegelungsverstoß
- für die Prüfung des Baugruppenzustands (Baugruppenausfall)
- für die Prüfung einer Meldesperre (ausgelöst durch Nahsteuerung)

Hinsichtlich der Qualitätskennungen sind Simulationsfälle zu entwickeln, die alle in Tabelle 1.2 aufgeführten Qualitätsbits setzen und übertragen können. Für die Qualitätskennungen Quality Descriptor (separate object) (qds) des Messwertes, Single-Point Information with Quality Descriptor (SIQ) der Einzelmeldung und Double-Point Information with Quality Descriptor (DIQ) der Doppelmeldung ist eine Anwendung der Qualitätsbits durchzuführen.

Qualitätsbit	**Beschreibung**
Blocked (BL): (blockiert)	Das entsprechende Informationsobjekt ist für die Übertragung blockiert. Der Wert verbleibt in dem Status, der vor der Blockierung herrschte und ändert sich nicht.
Substituted (SB): (ersetzt)	Der Wert des Informationsobjektes wird durch einen Vorgegebenen Wert ersetzt.
Not Topical (NT): (nicht aktuell)	Der Wert des Informationsobjektes ist nicht aktuell und steht der zeit nicht zur Verfügung.
Invalid (IV): (ungültig)	Der Wert des Informationsobjektes ist ungültig, wenn er nicht korrekt erfasst werden konnte. Der Wert darf somit nicht verwendet werden.
Overflow (OV): (überlauf)	Liegt der Wert des Informationsobjektes außerhalb eines definierten Bereiches, liegt ein Überlauf vor.

Tabelle 1.2: Qualitätsbits für alle Qualitätskennungen

Die Datenübertragung bei einer Kommunikationsverbindung zwischen dem zu implementierenden IEC 60870-5-104-Server und einem IEC 60870-5-104-Client erfolgt gemäß dem Übertragungsprotokoll über das TCP/IP-Netzwerk. Im Rhamen dieser Arbeit sollen neben dem Aufbau einer Kommunikationsverbindung entsprechende Prüfprozeduren zur Prüfung dieser Verbindung und Start/Stopp-Übertragungssteuerungen zur Steuerung des Datenverkehrs auf der Verbindung realisiert werden. Diese Festlegungen sind der Norm IEC 60870-5-104 [2] in Kapitel 5 beschrieben.

Die Simulationsumgebung soll nach erfolgreicher Parametrierung in der Lage sein Steuerinformationen aus der implementierten *Vor-Ort-Steuerung* oder einer beliebigen Fernsteuerung umzusetzen. Dabei ist für definierte Anwendungsfälle die Übertragungsursache Cause of Transmission (COT) festgelegt und in Abbildung 1.4 dargestellt.

Mittels des *ToggleButtons* der *Vor-Ort-Steuerung* soll dabei die Schalthoheit bestimmt werden können. Im Falle der Schalthoheit *Ort* und einer Kommunikationsverbindung zu einer Gegenstation als Client, sollen alle Zustände, welche durch einen örtlichen Befehl verändert wurden mit einer COT *Retloc* (Rückmeldung verursacht durch einen örtl. Befehl) an das übergeordnete Netzleitsystem übertragen werden. Im Rhamen dieser Arbeit wird die Doppelmeldung als Rückmeldung für einen Doppelbefehl mit dieser Ursache übertragen. Befehle von der Fernsteuerung sollen nicht entgegen genommen und mit einer Ursache *ActCon_NEGA* beantwortet werden.

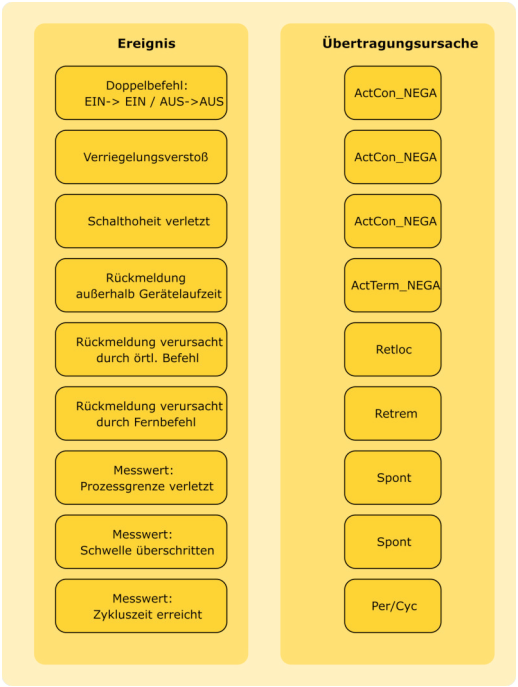


Abbildung 1.4: Anwendungsfälle für Übertragungsverhalten

Besteht die Schalthoheit *Fern*, so sollen bei einem Generalabfragebefehl durch das Netzleitsystem alle Informationsobjekte mit ihren aktuellen Zuständen übertragen werden. Dabei sollen die generierten Telegramme eine TK ohne Zeitmarke enthalten. Im Falle einer Zustandsänderung, welche durch einen Doppelbefehl verursacht wurde, soll die Befehlsaktivierung bestätigt werden und je nach Parametrierung die Zwischenstellung gefolgt von der Endstellung des Leistungsschalters übertragen werden. Dabei ist die

COT als *Retrem* (Rückmeldung verursacht durch Fernbefehl) festgelegt. Nach erfolgreich durchgeführter Befehlsgabe wird die Aktivierung des Befehls beendet. Als Übersicht ist in Abbildung 1.4 für jeden Anwendungsfall die entsprechende COT aufgeführt. Insgesamt soll für jedes Ereignis ein entsprechendes IEC 60870-5-104-Server-Telegramm erzeugt und an die Gegenstelle übertragen werden.

Zur Validierung der Ergebnisse soll das Netzleitsystem *SPRECON-V460* der Firma *Sprecher Automation GmbH*[1] eingesetzt und entsprechend parametriert werden. Zusätzlich ist das *Simulationsprogramm für das IEC 60870-5-104 Protokoll* [4] heranzuziehen. Mit diesen Tools sollen die generierten und an den Client übertragenen IEC 60870-5-104-Server-Telegramme auf Richtigkeit geprüft werden.

Die GUI ist unter Verwendung der Programmiersprache *Java*, dem entsprechenden GUI-Framework *JavaFX* und dem Oberflächengestaltungs-Tool *SceneBuilder*[2] zu erstellen. Mit diesem Tool soll das Design der Oberfläche gestaltet und im weiteren Verlauf mittels *JavaFX* mit Funktionalitäten erweitert werden.

1.2 Abgrenzung

Für die Umsetzung dieses Projektes muss eine Abgrenzung nicht behandelter Aspekte vorgenommen werden. Diese soll den Schwerpunkt dieser Arbeit verdeutlichen und legt die Grenze des Umfangs der Aufgabenstellung fest. Die Parametrierung des in der Masterstudienarbeit [3] festgelegten Prozesses wird innerhalb dieser Arbeit mittels TK mit Zeitmarke durchgeführt. Eine Möglichkeit der Parametrierung der Informationsobjekte mit TK ohne Zeitmarke wird nicht implementiert. Das Setzen der in Abschnitt 1.1 beschriebenen Qualitätsbits wird innerhalb dieser Arbeit ausschließlich für die beschrieben Informationsobjekte in Melderichtung durchgeführt. Hinsichtlich der Implementierung des eines IEC 69870-5-104-Servers wurde für die Umsetzung die Mastertstudienarbeit *Entwicklung einer Serverkommunikation auf Basis der Norm IEC 60870-5-104 unter Java* [5] herangezogen. Dabei wird die Anwendung auf die Gegebenheiten dieser Arbeit angepasst werden. Dabei werde für die Analyse der Telegramme vom IEC 60870-5-104-Client alle Telegramme im Unnumbered Control Functions (U)-Format

[1] `https://www.sprecher-automation.com`
[2] `http://gluonhq.com/products/scene-builder/`

und Numbered Supervisory Functions (S)-Format zu implementieren. Eine Begrenzung von Telegrammen im Numbered Information Transfer (I)-Format in Steuerungsrichtung erfolgt für Application Service Data Unit (ASDU) der TK 100 (Generalabfragebefehl), TK 59 (Doppelbefehl) und TK 63 (Sollwert-Stellbefehl verk. Gleitkommazahl). Die Parametrierung von Schaltverriegelungen ist im Rhamen dieser Arbeit nicht vorgesehen. Sie werden dennoch berücksichtigt und fest in der Implementierung kodiert.

Mit dieser Abgrenzung soll betont werden, dass es sich in dieser Arbeit um keine frei parametrierbare IEC 60870-5-104-Server-Anwendung handelt. Die Parametrierung ist eingeschränkt möglich. Mit dem gegebenen statischen Prozess sind definierte Anwendungsfälle zu implementieren. Dabei sind die IEC 60870-5-104-Adressierung und das Übertragungsverhalten der Informationsobjekte in Melderichtung parametrierbar.

Alle Bergriffe und Funktionen der Norm IEC 60870-5-101 [1] und IEC 60870-5-104 [2], die im Rahmen dieser Arbeit zur Erwähnung kommen werden als bekannt vorausgesetzt.

1.3 Gliederung der Arbeit

Die Arbeit beschreibt zunächst im nachfolgenden Kapitel 2 die Erweiterung des Simulators. Dabei wird zunächst der Ausgangszustand des bestehenden Simulators dargestellt. Daraufhin erfolgt die Beschreibung der einzelnen gestalteten Oberflächen für die Erweiterung des Simulators um eine Parametrierung. Das Kapitel wird durch die Ausführung eines Telegrammverkehrs zwischen der zu entwickelnden Simulationsumgebung und der Gegenstelle abgeschlossen. Im Anschluss wird die Implementierung der definierten Funktionen in Kapitel 3 aufgeführt. Innerhalb des Kapitels 4 werden die Ergebnisse dieser Arbeit dargestellt. Den inhaltlichen Abschluss dieser Arbeit bildet das Kapitel 5 mit einer Zusammenfassung und einem Ausblick.

Im Folgenden wird aus Gründen der besseren Lesbarkeit ausschließlich die männliche Form benutzt. Es können dabei aber sowohl männliche als auch weibliche Personen gemeint sein.

2 Planung zur Erweiterung des Simulators

Die Erweiterung des Simulators umfasst sowohl eine Ergänzung der bestehenden GUI als auch die Integration zur Möglichkeit der Herstellung einer Kommunikationsverbindung über das Fernwirkprotokoll IEC-60870-5-104. Mit der Anpassung der GUI gemäß der in der Aufgabenstellung in Abschnitt 1.1 beschriebenen Anforderungen erfolgt innerhalb im nachfolgenden Abschnitt 2.1 eine kurze Beschreibung des zu Beginn dieses Projektes bestehenden Ausgangszustandes des Simulators. Basierend auf diesem Stand wird im Abschnitt 2.2 die Gestaltung der zu ergänzenden *Wischermeldung*, des *Meldebuches* sowie der *Parametrierung* spezifiziert. Mit Hilfe von Sequenzdiagrammen wird die Grudnlage für die Realisierung des Simulator als IEC 60870-5-104-Server in Abschnitt 2.3 geschaffen.

2.1 Status Quo des Simulators

Die zu Beginn dieses Projektes zur Verfügung stehende Anwendung dient als eine softwarebasierte Simulation eines FWG. Die in Abbildung 2.1 dargestellte Oberfläche der Anwendung umfasst neben der grafischen Darstellung der internen Verschaltung der einzelnen Baugruppen eines FWG inklusiver Verschaltung zu prozessseitigen Endlagenschaltern und Relais (magenta umrahmt) eine in orange umrandete *Prozesssimulation* in einpoliger Darstellung, dessen Zustand mit Hilfe des FWG abgebildet und verändern werden kann. Mittels des Informationsumfangs des Schaltfeldes bestehend aus Leistungsschalter (Doppelbefehl mit Doppelmeldung als Rückmeldung), Erdungsschalter (Einzelmeldung), Stromwandler (Messwert) sowie einem Anzeigeinstrument (Sollwert-Stellbefehl) soll das Verhalten der einzelnen Komponenten in Abhängigkeit des Simulationsfalls anhand von Schalterbewegungen und Einfärbungen animiert werden.

Über den blau umrahmten *Button Simulationszeiten* öffnet sich eine separate Oberfläche zur Anpassung der Animationszeiten der Schalterbewegung des Leistungsschalters, Erdungsschalters sowie der Auslösekontakte der Koppelrelais der digitalen Ausgabe. Mittels des gelb umrandeten *Buttons Ortsteuerung* lässt sich die separate Oberfläche der *Vor-Ort-Steuerung* öffnen.

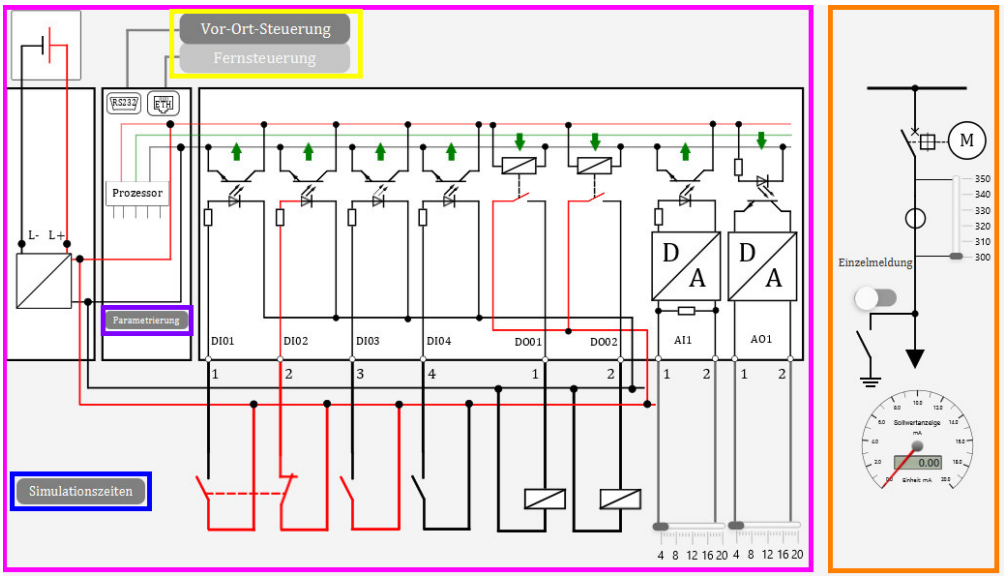


Abbildung 2.1: Oberfläche des Simulators - FWG und Prozesssimulation

Die von einer netzwerkbasierten Kommunikation losgelöste Möglichkeit der Steuerung des Prozesses erfolgt mittels der in Abbildung 2.2 dargestellten *Vor-Ort-Steuerung*, welche eine Bedien- und Beobachtungseinheit abbildet. Diese dient dem Auslösen eines Steuervorgangs zum Veranlassen der Darstellung der internen Vorgänge eines FWG und der damit einhergehenden Auswirkungen auf den Prozess. Mittels des *ToggleButtons* kann die *Schalthoheit* festgelegt werden. Dabei ist das Absetzen eines Befehls durch die *Vor-Ort-Steuerung* ausschließlich in der Position *Ort* möglich. Über Light-Emitting Diode (LED) werden die aktuellen Zustände hinsichtlich *Schalthoheit* und *Verriegelung* signalisiert.

Eine Befehlsgabe an den Leistungsschalter erfolgt durch Auswahl des Symbols des Schaltgerätes auf der Oberfläche, wobei hierdurch die unten angeordneten *Buttons* (*I* Schaltrichtung Ein, *O* Schaltrichtung Aus, *E* Bestätigung, *C* Abbruch) freigeschaltet werden. Nach Auswahl der Schaltrichtung kann die Befehlsgabe bestätigt oder abgebrochen werden.

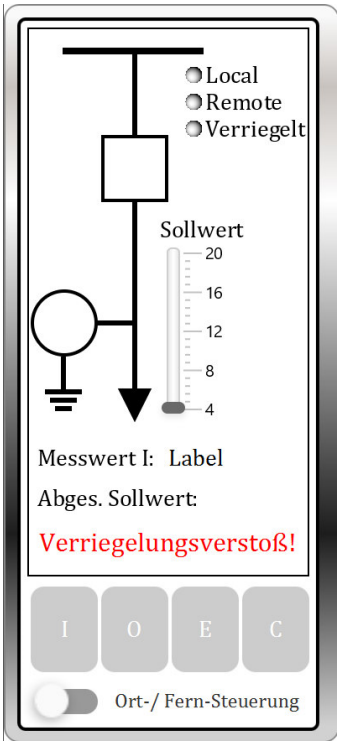


Abbildung 2.2: Vor-Ort-Steuerung

Dieser Vorgang löst den Beginn der Animation der Befehlsgabe aus. Es wird der Auslösekontakt der für die Schaltrichtung zuständigen digitalen Ausgabe des FWG für die Dauer Simulationszeit des Leistungsschalters geschlossen. Gleichzeitig wird die Schalterbewegung des Leistungsschalters in der *Prozesssimulation* und der Zustandswechsel der Endlagenschalter der digitalen Eingaben animiert. Der Zustandswechsel wird inklusiver Differenzstellung in der *Vor-Ort-Steuerung* visualisiert. Nach Beendigung der Befehlsgabe nach der Gerätelaufzeit fährt der Auslösekontakt des Koppelrelais der digitalen Ausgabe wieder in seine ursprüngliche Position.

Eine spontane Änderung des Prozesszustandes kann mittels des Erdungsschalters der *Prozesssimulation* simuliert werden. Dabei wird durch Selektion des *ToggleButtons Einzelmeldung* der Vorgang der Schalteranimation eingeleitet. Der Endlagenschalter der Klemme *DI03* der digitalen Eingabe nimmt parallel zum Erdungsschalter der *Prozesssimulation* den entsprechenden Zustand ein. Dieser wird ebenfalls in der *Vor-Ort-Steuerung* visualisiert.

Die Messwertsimulation wird durch Bewegung des *Sliders* der *Prozesssimulation* angestoßen, wobei hierbei die simulierte Prozessgröße auf die für den analogen Eingang erforderliche Einheitsgröße skaliert wird. In der *Vor-Ort-Steuerung* wird der aktuelle Messwert als Prozessgröße angezeigt. Analog zur Befehlsgabe an den Leistungsschalter wird der Sollwert-Stellbefehl durch die *Vor-Ort-Steuerung* abgesetzt. Hierfür wird der dort positionierte *Slider* auf den entsprechenden Sollwert geschoben. Durch eine Bestätigung dieses Wertes erfolgt eine Anzeige dieses im Anzeigeinstrument der *Prozesssimulation*.

Insgesamt ist der Simulator in der Lage mittels einer Prozesssimulation eine Messwertsimulation und eine Simulation einer Einzelmeldung durchzuführen. Weiterhin ist es möglich über die *Vor-Ort-Steuerung* einen Doppelbefehl mit entsprechender Doppelmeldung als Rückmeldung zu generieren sowie einen Sollwert-Stellbefehl absetzen zu können. Zum jeweiligen Simulationsfall werden entsprechende Animationen durchgeführt und die internen Abläufe innerhalb des FWG visualisiert. Eine detaillierte Beschreibung dieser Anwendung ist in der Masterstudienarbeit [3] nachzulesen.

2.2 Erweiterungen der grafischen Benutzeroberfläche

Die Erweiterung der GUI des Simulators um Komponenten zur Umsetzung der definierten Funktionalitäten erfolgt innerhalb dieses Abschnittes und den enthaltenen Unterabschnitten zunächst durch die Gestaltung der Oberflächen und Komponenten. Abbildung 2.3 zeigt alle Erweiterungen, welche auf der Oberfläche der *Vor-Ort-Steuerung*, des *FWG* und der *Prozesssimulation* umgesetzt werden. Die Beschreibung der entsprechenden Ergänzung erfolgt dabei in den nachfolgenden Abschnitten, in denen auf diese Abbildung referenziert wird.

Abbildung 2.3 zeigt mit den eingefärbten Rhamen die einzelnen Komponenten, um die die Oberfläche zu erweitern ist. In den nachfolgenden Unterabschnitten werden die Erweiterungen hinsichtlich der Wischermeldung, des Meldebuches sowie der Parametrierung beschrieben. An dieser Stelle werden drei weitere Erweiterungen erläutert, dessen Notwendigkeit und Entwicklung mit geringem Aufwand verbunden war.

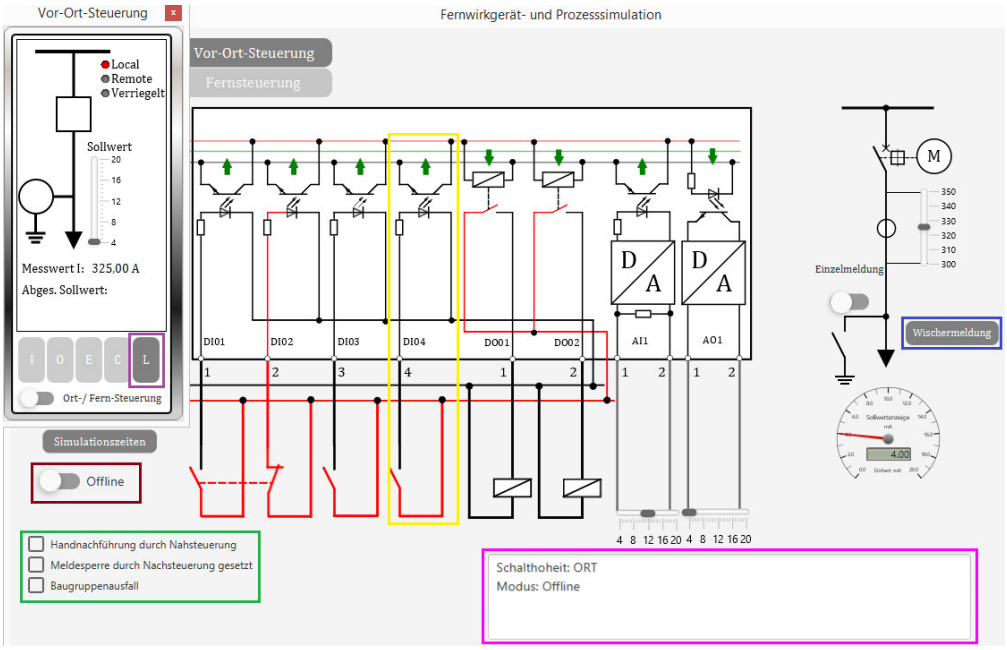


Abbildung 2.3: Erweiterte grafische Oberfläche des Simulators

Beginnend mit dem in Abbildung 2.3 braun umrandete *ToggleButton* mit der Beschriftung *Offline*, soll mit diesem durch Selektion die Anwendung in den *Online*-Modus gewechselt werden. Dabei kann in diesem Modus eine Kommunikationsverbindung von einem IEC 60870-5-104-Client zu dieser Anwendung als IEC 60870-5-104-Server hergestellt werden. Die Simulationsfälle zum Setzen definierter Qualitätsbits sollen mittels der grün umrahmten *CheckBoxen* ausgeführt werden. Wird die obere *CheckBox* selektiert, so wird der Simulationsfall *Handnachführung durch Nahsteuerung* abgebildet. Dabei wird das Qualitätsbit SB für alle Informationsobjekte gesetzt und eine Übertragung dieser Informationsobjekte mit entsprechenden Qualitätskennungen an den IEC 60870-5-104-Client übertragen. Ist die *CheckBox* nicht ausgewählt, so weist das SB-Bit den Wert 0 auf.

Analog hierzu wird bei den anderen Simulationsfällen verfahren. Im Simulationsfall der *CheckBox Meldesperre durch Nahsteuerung* ist das Qualitätsbit BL zu setzen. Die Qualitätsbits IV und NT werden im Falle der Simulation eines *Baugruppenausfalls* gesetzt. Um alle wichtigen Ereignisse für den Anwender zu protokollieren ist im magenta umfassten Bereich der GUI in Abbildung 2.3 ein *TextArea* platziert. In diesem wird jedes Ereignis mit seinem Status gelistet. Dabei handelt es sich beispielsweise um Ereignisse wie die Umschaltung der Schalthoheit, dem Modus der Kommunikationsverbindung oder einem Verriegelungsverstoß.

2.2.1 Wischermeldung

Eine Wischermeldung ist eine sehr kurzzeitig anstehende Einzelmeldung, bei der nur das Kommen des Prozess-Signals zeitrichtig erfasst und weiterverarbeitet wird. Sie hat verschiedenste Anwendungsfälle. Bei einem *Leistungsschalterfall* wird die Wischermeldung verwendet, um die Zustandsänderung des Leistungsschalters von Ein nach Aus aufgrund einer Schutzanregung zu signalisieren. Somit wird der Zustandswechsel gemeldet, welcher nicht durch einen Steuervorgang ausgelöst wurde. Eine Wischermeldung wird weiterhin verwendet, um anzuzeigen, dass eine Befehlsrückmeldung nicht innerhalb der vorgegeben Zeit eingetroffen ist.

Die Komponenten für die Erweiterung einer Wischermeldung sind in Abbildung 2.3 in gelb und blau umrahmt. In der *Prozesssimulation* erfolgt die Simulation der Wischermeldung mittels des blau umrandeten *Buttons Wischermeldung*. Durch einen Mausklick auf diesen wird die Simulation am Endlagenschalter an der Klemme *DI04* der digitalen Eingabe durchgeführt. Dabei wird der Schließer geschlossen und direkt wieder geöffnet. Hierdurch kann die Wischermeldung als ein kurzzeitig anstehenden Signals an einem digitalen Eingang simuliert werden.

2.2.2 Meldebuch der Vor-Ort-Steuerung

Die Protokollierung von Zustandsänderungen kann für eine *Vor-Ort-Steuerung* in einem *Meldebuch* erfolgen. Jede auftretende Änderung im Prozess wird dort mit dem entsprechenden Zeitstempel gelistet. Es können zudem Befehle, Alarmmeldungen für Prozessgrenzenverletzungen oder Verriegelungsverstöße protokolliert werden.

Das für den Simulator entwickelte *Meldebuch* zeichnet ausschließlich Zustandsänderungen auf. Über den *L-Button*, welcher in der Abbildung 2.3 in der *Vor-Ort-Steuerung* lila umrahmt ist, öffnet sich das in Abbildung 2.4 dargestellte *Meldebuch*. Es besteht aus einer Tabelle mit den Spalten *Datum/Uhzeit* und *MELD*. Bei einer Zustandsänderung wird die Tabelle um das aktuelle Ereignis mit Zeitstempel ergänzt. Es sollen keine Einträge gelöscht werden können. Mit einem *ScrollBar* kann bei Erreichen der maximal sichtbaren Einträge durch die Tabelle geblättert werden. Durch ein Mausklick auf dem *<-Button* erfolgt ein Wechsel der Ansicht zur Steuerung der *Vor-Ort-Steuerung*.

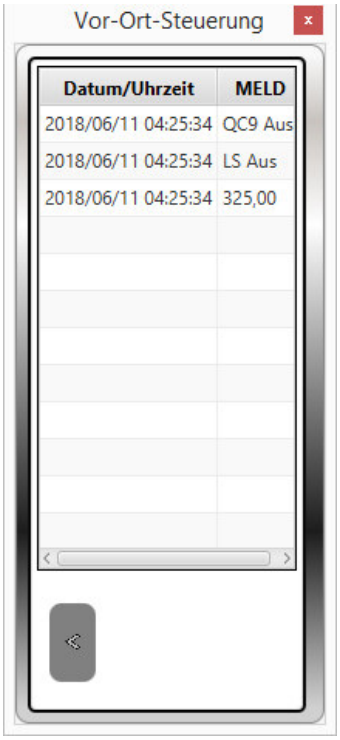


Abbildung 2.4: Meldebuch der Vor-Ort-Steuerung

2.2.3 Parametrierung gemäß IEC 60870-5-104

Ein Hauptbestandteil der Erweiterung des Simulators liegt in der Entwicklung einer Möglichkeit der Parametrierung gemäß des Fernwirkprotokolls IEC 60870-5-104. Hierfür sollen die in der Aufgabenstellung unter Abschnitt 1.1 definierten Informationsobjekte hinsichtlich ihrer IEC 60870-5-104-Adressierung und ihres Übertragungsverhaltens para-

metriert werden. Für die Entwicklung einer Parametrierung wird die Gegenüberstellung von Parametrier-Tools namhafter Hersteller von Fernwirksystemen aus der Projektarbeit *Darstellung und Dokumentation von Fernwirksystemen unterschiedlicher Hersteller* [6] herangezogen.

Aus den unterschiedlichen Tools sind Aspekte bezüglich der Bedienung dieser für diese Arbeit berücksichtigt. Dabei soll beispielsweise möglichst auf die Anwendung von Kontextmenüs verzichtet werden. Bei nicht plausiblen oder ungültigen Eingagen soll der Anwender entsprechend darüber informiert werden. Allgemeine Parameter wie die CAASDU und die IOA sollen zunächst innerhalb der Rangiertabelle vergeben werden. Sobald ein Informationsobjekt hinsichtlich seines individuellen Übertragungsverhaltens konfiguriert werden soll, ist dies über eine separate Maske durchzuführen. Somit liegt eine übersichtliche Rangiertabelle vor.

In Anbetracht der allgemeinen Prozedur zur Parametrierung eines FWG und der Anbindung dieses an ein Netzleitsystem orientiert sich die Entwicklung der Parametrierung des Simulators an folgendem Ablauf:

1. Ermittlung des Prozessdatenumfangs
2. Auswahl der Hardware-Komponenten des FWG
3. Hardwarekonfiguration
4. Parametrierung einzelner Informationsobjekte
5. Fernwirkankopplung

Da der Prozessdatenumfang und somit die Auswahl der Hardware-Komponenten des FWG fest definiert ist, soll die Hardwarekonfiguration ein statisches Abbild der erforderlichen Baugruppen sein. Es ist somit keine Bestückung des Baugruppenträgers zu realisieren. Das Abbild des FWG des Simulators ist in Abbildung 2.5 dargestellt. Diese Ansicht öffnet sich, sobald der *Button Parametrierung* geklickt wird, welcher auf der Oberfläche des Simulators in der Central Processing Unit (CPU) positioniert ist. Abgebildet sind alle Baugruppen mit ihren Anschlussklemmen. Zur Übersicht sind im Rahmen dieses Projektes die Bezeichner jedes Informationsobjektes an der zugehörigen Klemme platziert. Dabei liegt für diesen Informationsumfang eine bereits parametrierte Rangiertabelle als Vorlage vor, welche im weiteren Verlauf beschrieben wird. Mit dem

Button Weiter, welcher in Abbildung 2.5 unten rechts angeordnet ist, soll eine Oberfläche zur Parametrierung des FWG gemäß IEC 60870-5-104 geöffnet werden.

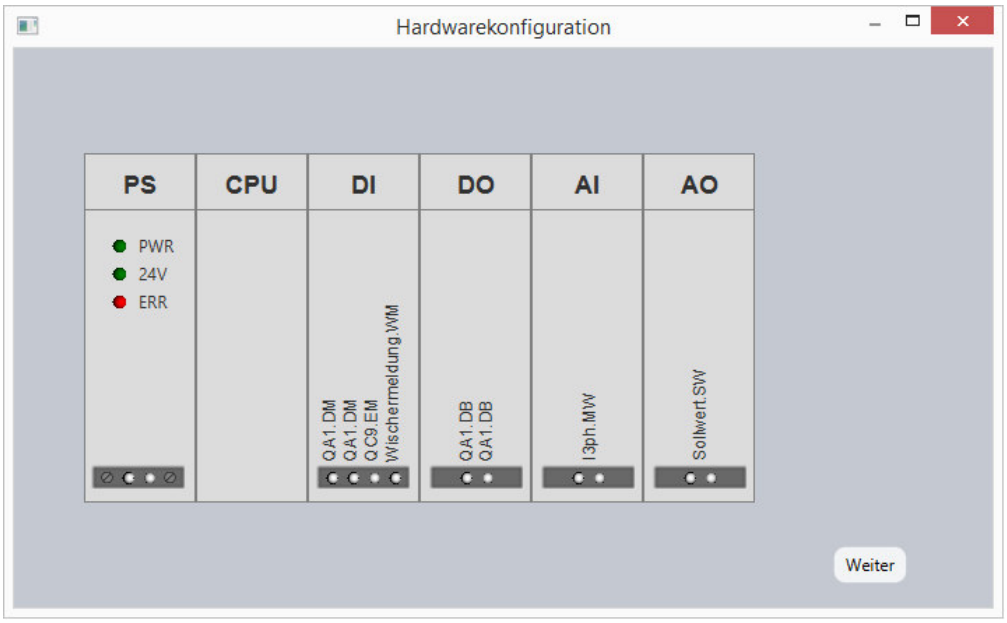


Abbildung 2.5: Parametrierung: Hardwarekonfiguration

Die auf der Hardwarekonfiguration basierende Parametrierung beginnt mit einer *Gerätespezifischen Parametrierung*. Die *TabPane* in Abbildung 2.6 zeigt im *Tab Allgemein* die Maske zur Eingabe der stationsspezifischen Parameter. Im Feld *CAASDU* erfolgt die Vergabe der Stationsadresse. Da diese für alle Informationsobjekte innerhalb dieser Unterstation gilt, soll die Festlegung global erfolgen. Alle zu paramentrierenden Informationsobjekte übernehmen folglich diese CAASDU. Hierdurch ist es nicht erforderlich für jedes Datenpunkt diese Adresse erneut einzugeben. Insgesamt erfolgt im Rahmen dieser Arbeit keine Strukturierung der Adressierung (CAASDU und IOA) und somit keine Bit-Kodierung. Die Eingabe wird als gemeinsamer dezimaler Zahlenwert vorausgesetzt und interpretiert. Bei der Implementierung der Prüfung des Eingabewertes ist der in der Norm IEC 60870-5-104 definierte Adressbereich der CAASDU zu berücksichtigen. Es ist somit nur die Eingabe eines Wertes zwischen *0* und *65535* zu akzeptieren, andernfalls soll eine entsprechende Meldung zum erlaubten Wertebereich erscheinen.

Mit dem Feld *Herkunftsadresse* wird eine Adresse angegeben, mit der die Quelle identifiziert werden kann, welche einen Vorgang in Steuerungsrichtung aktiviert bzw. aktivieren darf. Sie dient somit zur Identifizierung des Absenders eines Telegramms beim Empfänger. Für den weiteren Verlauf dieser Arbeit wird diese Adresse mit *0* festgelegt. Das bedeutet, dass keine Zentralstation explizit in der Unterstation definiert ist und Steuervorgänge eines beliebigen Clients vom FWG akzeptiert werden. Eine Anpassung dieser *Herkunftsadresse* ist für diese Arbeit zunächst nicht vorgesehen. Der *Button Zurück* ermöglicht es dem Anwender zurück zur Ansicht der *Hardwarekonfiguration* zu gelangen. Dieser ist sowohl für den *Tab Allgemein* als auch für die *Tabs Schnittstellenparametrierung* und *Informationsobjekte* unten links positioniert.

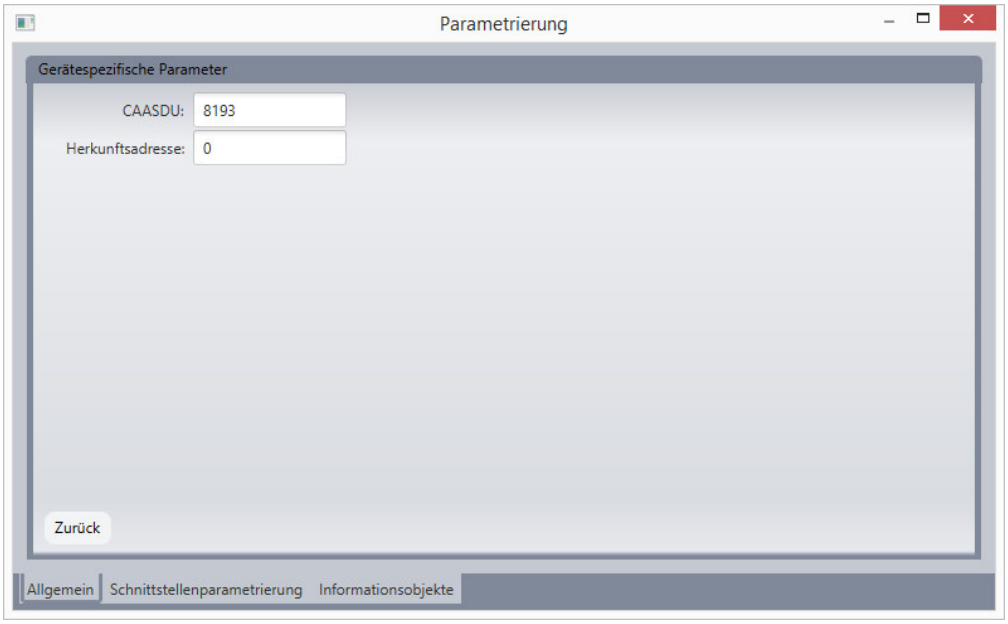


Abbildung 2.6: Parametrierung: Allgemein

Der Wechsel zum den jeweilig gewünschten *Tab* erfolgt durch Anwahl dieses in der unteren Leiste. Durch Auswahl des *Tabs Schnittstellenparametrierung* öffnet sich die in Abbildung 2.7 dargestellte Maske. Diese dient dazu, die Notwendigkeit der Vergabe einer Internet Protocol (IP)-Adresse im Falle einer fernwirktechnischen Anbindung an ein Netzleitsystem zu verdeutlichen. Die hierfür erforderliche *Schnittstelle* in einem *SplitMenuButton* als *Ethernet(ETH)*-Kommunikationsschnittstelle angezeigt. Zur Vervollständigung der Anzeige aller im FWG von der zentralen Prozessoreinheit

zur Verfügung stehender Kommunikationsschnittstellen wird die serielle Schnittstelle *RS232(COM)* aufgeführt. Mit dieser soll verdeutlicht werden, dass die *Vor-Ort-Steuerung* diese Schnittstelle zur Kopplung mit dem FWG verwendet. Beide *SplitMenuButtons* weisen keine Funktionalitäten auf und sind als statische Elemente festgelegt. Um mit der *Ethernet(ETH)*-Schnittstelle eine netzwerkbasierte Kommunikationsverbindung zu einem Client aufbauen zu können, sind entsprechende *Adresseinstellungen* für diese durchzuführen. Mit den Textfeldern *Lokale IP-Adresse*, *Subnetzmaske* sowie *Standardgateway* soll innerhalb dieses Schrittes der Parametrierung deutlich gemacht werden, dass mit diesen Einstellungen das FWG innerhalb des lokalen Netzwerkes eindeutig identifiziert wird. Beim Ausführen der Simulationsumgebung als Anwendung soll die IP-Adresse des PC, auf dem diese ausgeführt wird, ermittelt und in der Oberfläche initial angezeigt werden. Da die Ermittlung der übrigen Adresseinstellungen für dieses Projekt zunächst nicht relevant sind, werden diese zur Übersicht ohne weitere Funktionalitäten dargestellt.

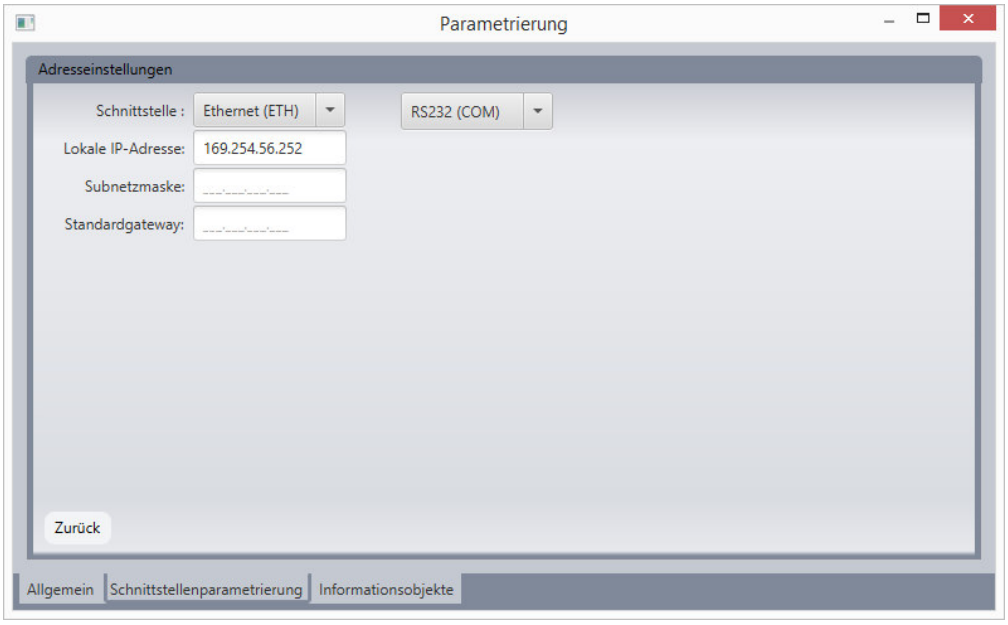


Abbildung 2.7: Parametrierung: Kommunikationsschnittstelle

Die Parametrierung des FWG hinsichtlich der IEC 60870-5-104 -Adressierung und des Übertragungsverhaltens kann im *Tab Informationsobjekte* durchgeführt werden. In Abbildung 2.8 ist die Rangiertabelle mit den entsprechenden Datenpunkten abgebil-

det. Dabei sind dort alle Informationsobjekte bereits initial parametriert als Vorlage bereitgestellt. In der Tabelle sind neben Informationsobjekten mit Hardwarebezug zusätzliche Informationsobjekte ohne Hardwarebezug als Systemmeldungen aufgelistet. Nicht editierbar sind die Spalten *Zeitmarke*, *Klemme*, *Baugruppe*, *TK* sowie *CAASDU*. Da die *CAASDU* bereits in der *Gerätespezifischen Parametrierung* vergeben wird und diese von jedem Informationsobjekt übernommen wird, sind die Zellen dieser Spalte nicht editierbar. Die *TK* ist für jeden Datenpunkt der Rangiertabelle individuell festgelegt. Anhand dieser richtet sich die zu einem späteren Zeitpunkt beschriebene spezifische Parametrierung. Für die hardwarebezogenen Informationsobjekte ist in den Spalten *Baugruppe* sowie *Klemme* die entsprechende Quelle am FWG zu entnehmen. Alle Systemmeldungen ohne Hardwarebezug sind der fiktiven Baugruppe *sysInfo* als Systeminformation zugeordnet, wobei die Klemmen durch nummeriert sind. Die in der Spalte *Zeitmarke* positionieren *CheckBoxen* sind ausgehend von der Aufgabenstellung zunächst nicht zu deaktivieren, da alle TK innerhalb dieses Projektes mit Zeitmarke parametriert werden sollen.

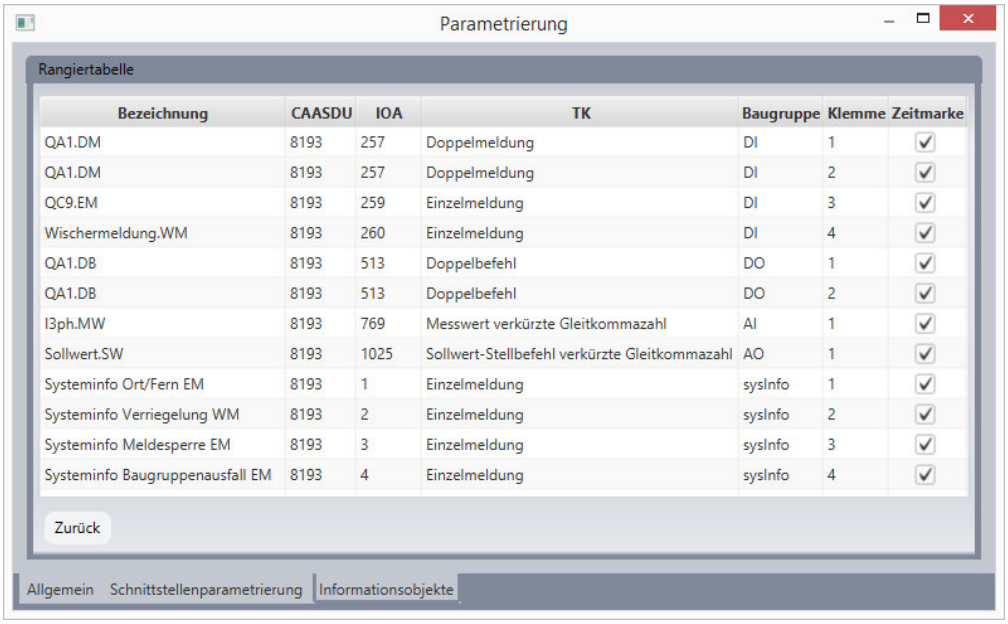


Abbildung 2.8: Parametrierung: Informationsobjekte

Für jedes Objekt können innerhalb dieser Tabelle durch ein Mausklick in die entsprechende Zelle die *Bezeichnung* und *IOA* angepasst werden. Anhand der *Bezeichnung* kann jedes Informationsobjekt als individueller Datenpunkt identifiziert und mit seiner Funktion gedeutet werden. Dabei richtet die Vergabe der initialen *Bezeichnung* an der primärtechnischen Größe gefolgt vom Typ des Informationsobjektes. Das erste in der Tabelle aufgeführte Informationsobjekt *<QA1.DM>* besagt beispielsweise, dass es sich um eine Doppelmeldung *DM* des Leistungsschalters *QA1* handelt. Für die Systemmeldungen ist die Funktion dieser Meldung im Klartext ausgeschrieben gefolgt vom Typ der Meldung.

Bei der Parametrierung der IOA ist darauf zu achten, dass für die Informationsobjekte der TK *Doppelmeldung* und *Doppelbefehl* der vergebende Wert für beide Hardware-Klemmen übernommen wird. Die Eingabe der IOA ist als gemeinsamer dezimaler Zahlenwert der in der Norm IEC 60870-5-104 festgelegten drei Oktett durchzuführen. Zudem ist bei Parametrierung der genormte Adressbereich zu prüfen. Wird ein Wert außerhalb des Bereiches zwischen *1* und *16777215* bei der Eingabe festgestellt, ist eine Meldung zur Verletzung des Wertebereiches zu generieren. Da eine IOA zur Identifizierung genau eines Informationsobjektes dient, soll eine Prüfung auf vergebener IOA implementiert werden. Bei Eingabe einer bereits vergebenen IOA ist eine Meldung zu erzeugen, welche den Anwender darauf hinweisen soll. Der getätigte Eintrag soll verworfen und nicht übernommen werden. Um eine aufeinanderfolgende Anpassung der IOA zügig gewährleisten zu können, soll nach Änderung einer IOA die Selektion des als nächstes anzupassenden Informationsobjektes genügen, um die Tastatureingabe fortzuführen.

Wie bereits erwähnt wird mittels der Rangiertabelle die IEC 60870-5-104 - Parametrierung durchgeführt. Die Vergabe der IOA erfolgt innerhalb der Tabelle. Das Übertragungsverhalten und weiterer Parameter hingegen wird im Rahmen einer spezifischen Parametrierung für jedes Informationsobjekt separat realisiert. Dabei kann das Verhalten zur Festlegung der Übertragung ausschließlich für Informationsobjekte in Melderichtung mit Hardwarebezug durchgeführt werden. Alle Systemmeldungen sind entsprechend ihrer Funktion zu übertragen. Die spezifische Parametrierung richtet sich nach der entsprechenden TK des Informationsobjektes. Auf Basis dieser sind unterschiedliche Parameter einzustellen. Aufgrund des statischen Prozesses sind bestimmte

Parameter innerhalb dieser Arbeit teilweise fest vorgegeben. An entsprechender Stelle wird dabei Bezug auf die Prozessgröße genommen.

In den nachfolgenden Unterabschnitten dieses Abschnittes werden die spezifischen Parameter des *Doppelbefehls*, der *Doppelmeldung*, *Einzelmeldung* sowie des *Messwertes* beschrieben. Dabei öffnet sich das Fenster der *spezifischen Parameter* durch ein Doppelklick auf das entsprechende Informationsobjekt. Für den *Sollwert-Stellbefehl* ist im Folgenden keine Vorstellung der Oberfläche vorgesehen, da für dieses Informationsobjekt keine spezifische Parametrierung im Rahmen dieser Arbeit durchführt wird. Mit der festen Vorgabe eines Sollwertes als Einheits- und Prozessgröße zwischen *4-20mA* ist das Informationsobjekt in einem Netzleitsystem zu parametrieren.

Jede Oberfläche der *spezifischen Parametrierung* ist als *Accordion* mit den drei *Titled-Panes Allgemeine Parameter*, *IEC 60870 Adresse* sowie *Informationsobjekt* aufgebaut. Da sich die *spezifische Parametrierung* von TK zu TK unterscheiden, erfolgt zunächst an dieser Stelle der Beschreibung der Fensterbereiche *Allgemeine Parameter* sowie *IEC 60870 Adresse*. Die Funktionalitäten und der Aufbau der in Abbildung 2.9 dargestellten Textfelder ist für die *spezifische Parametrierung* alle TK identisch zu realisieren. Am Beispiel des Informationsobjektes der TK *Sollwert-Stellbefehl* ist in der Abbildung 2.9 die Übernahme der individuellen *Bezeichnung*, *CAASDU* sowie *IOA* aus der Rangiertabelle zu erkennen. Eine Änderung der *Bezeichnung* und *IOA* ist innerhalb dieser Maske nicht vorgesehen. Diese Fensterbereiche dienen lediglich der Zusammenfassung der in der Rangiertabelle parametrierbaren Größen.

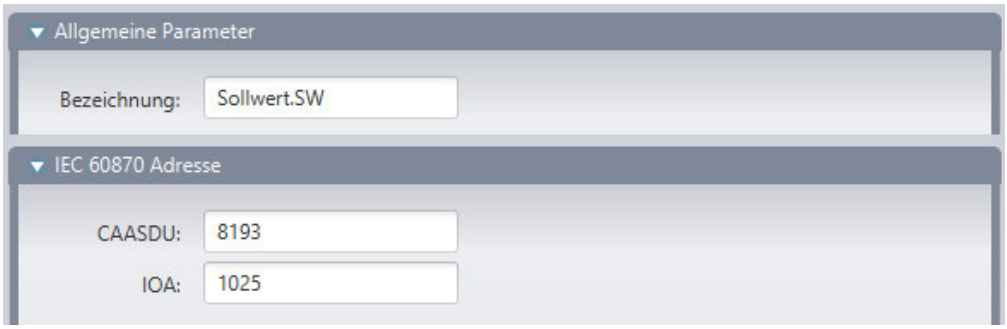


Abbildung 2.9: Spezifische Parameter: Allgemeine Parameter und IEC 60870 Adresse

2.2.3.1 Doppelbefehl

Abbildung 2.10 zeigt die Oberfläche der *spezifischen Parameter* des *Doppelbefehls* im *TitledPane Informationsobjekt*. Der Aufbau der grün umrahmten Felder ist für alle nachfolgenden Oberflächen identisch, sodass eine Beschreibung dieser an dieser Stelle einmalig erfolgt. Dabei gibt das Feld *Typkennung* die TK des aktuell ausgewählten Informationsobjektes wieder. Aus der individuellen *Baugruppe* und *Klemme* des Objektes wird aus der Rangiertabelle die *Objektquelle* im gleichnamigen Feld zusammengesetzt und angezeigt. Im Feld *Zeitmarke* wird für dieses Projekt jedes Informationsobjekt mit einer Zeitmarke von 7 Oktett versehen.

In der Rangiertabelle befindet sich ein einziges Informationsobjekt mit einer TK in Steuerungsrichtung, für das eine *spezifische Parametrierung* zu realisieren ist. Diese sind in Abbildung 2.10 rot umrandet. Dabei besteht beim *Doppelbefehl* der Einstellung einer *Rückmeldung*. In einer *ChioceBox* werden alle in Melderichtung zur Verfügung stehenden Informationsobjekte der digitalen Eingabe zur Auswahl angezeigt. Gemäß der in abgebildeten Einstellung wird neben der TK auch die Objektquelle und die parametrierte IOA angezeigt. Mit der *Befehlsgabezeit* kann eine Zeit parametriert werden, in der das Relais der digitalen Ausgabe im Zuge eines Schaltbefehls angesteuert werden soll und für genau diese Dauer eine Befehlsgabe durchführt. Dabei ist die an dieser Stelle parametrierte Zeit entsprechend in den Animationsvorgang des Simulators einzubinden.

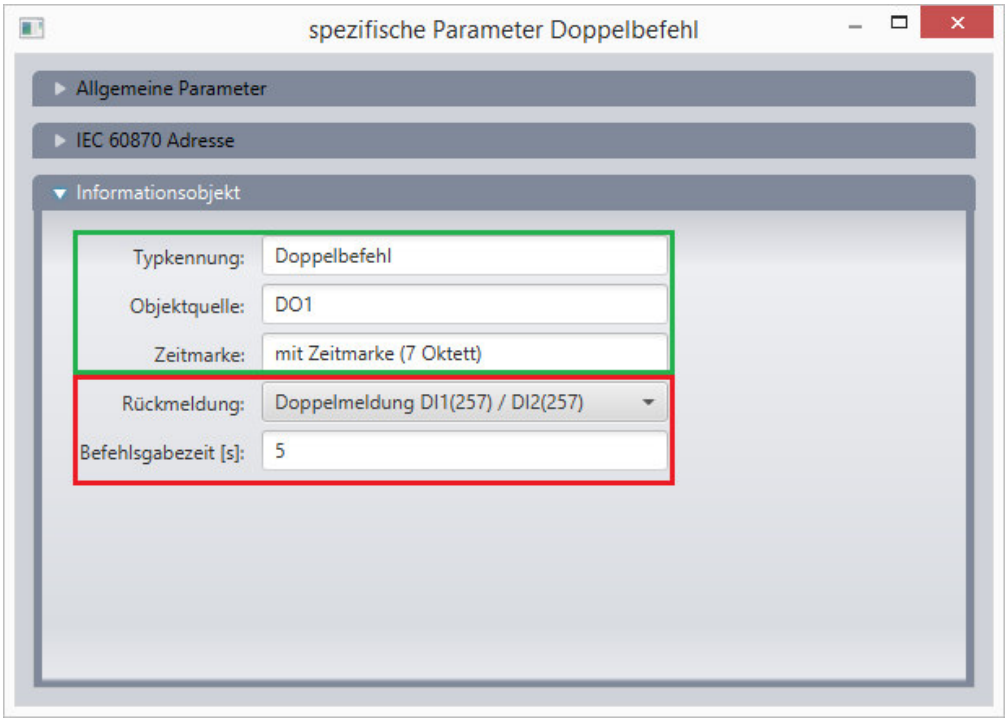


Abbildung 2.10: Spezifische Parameter: Doppelbefehl

2.2.3.2 Doppelmeldung

Hinsichtlich der *spezifischen Parametrierung* der *Doppelmeldung* können die beiden in Abbildung 2.11 rot umrahmten Parameter eingestellt werden. Mit der Gerätelaufzeit wird die Gesamtlaufzeit des Schaltgerätes (Leistungsschalter) angegeben. Mit diesem Parameter kann eine Rückmeldeüberwachung realisiert werden. Dabei wird im Rahmen einer Befehlsverarbeitung eine zeitliche Überwachung durchgeführt. Ab Beginn der Befehlsgabe wird kontrolliert, ob das Schaltgerät innerhalb der parametrierten Gerätelaufzeit die gewünschte Endstellung erreicht hat. Trifft die Rückmeldung innerhalb der Gerätelaufzeit ein, so kann die Befehlsbeendigung als erfolgreich (COT:ActTerm) zurückgemeldet werden. Erfolg keine Rückmeldung innerhalb der Gerätelaufzeit, wird die Befehlsbeendigung negativ zurückgemeldet (COT:ActTerm_NEGA). Der beschriebene Anwendungsfall für die Verwendung des Gerätelaufzeit wird im Rahmen dieser Arbeit verwendet, um entsprechende Server-Telegramme zu generieren. Es kommen andere Anwendungsfälle in Frage, welche für diese Anwendung nicht zu implementieren sind.

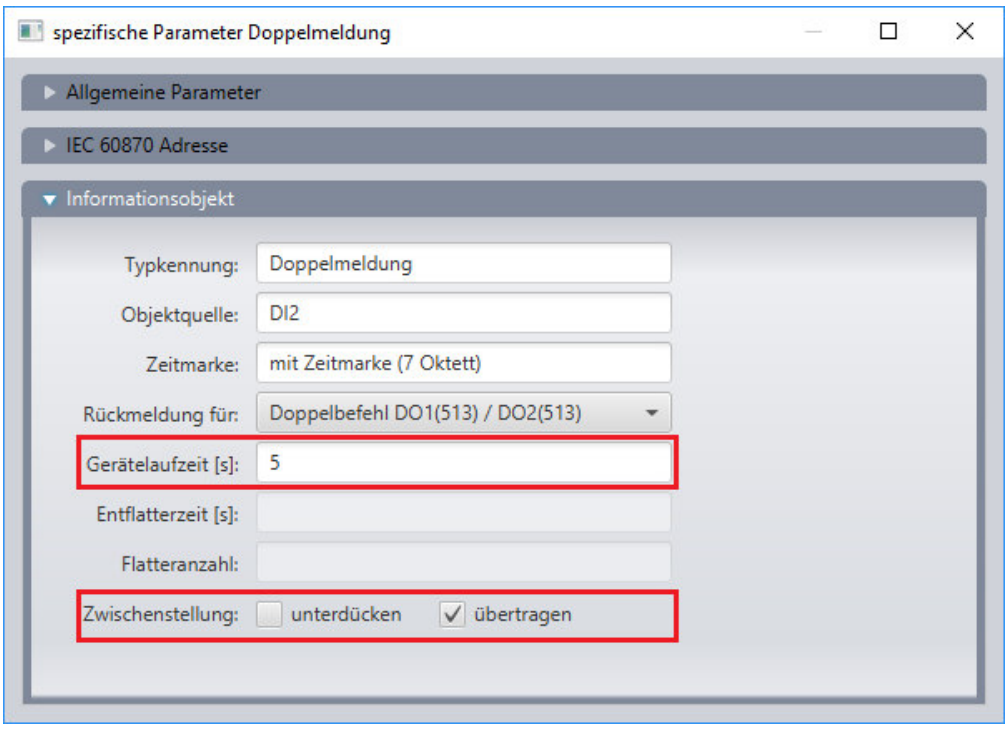


Abbildung 2.11: Spezifische Parameter: Doppelmeldung

Das Übertragungsverhalten der *Zwischenstellung* wird in der Maske *spezifische Parameter Doppelmeldung* mit Hilfe der *CheckBoxen* eingestellt. Ist die *CheckBox unterdrücken* selektiert, erfolgt keine Übertragung der Zwischenstellung vom Server an den Client. Andernfalls wird das entsprechende Telegramm in Melderichtung generiert und übertragen, welches die Zwischenstellungsposition enthält. Dabei gilt diese Einstellung ausschließlich für eine Befehlsaktivierung durch eine *Fernsteuerung*. Bei einer *Vor-Ort-Steuerung* wird die Zwischenstellung nicht an den Client übertragen.

Die *ChioceBox Rückmeldung für:* zeigt lediglich den Doppelbefehl an, für den diese Doppelmeldung als Rückmeldung parametriert wurde. Darüber hinaus weist dieses Element keine Funktionalität auf. Eine Parametrierung einer *Entflatterzeit* und *Flatteranzahl* ist für diese Doppelmeldung nicht vorgesehen. Da dies prinzipiell für eine Doppelmeldung möglich wäre, sind die Felder in der Maske aufgeführt jedoch deaktiviert. Eine Anwendung der Entflatterung von Meldungen wird im nachfolgenden Unterabschnitt für eine *Einzelmeldung* beschrieben.

2.2.3.3 Einzelmeldung

Für Informationsobjekte vom Typ *Einzelmeldung* kann eine Entflatterung parametriert werden. Mit den in Abbildung 2.12 rot umrahmten Parametern *Entflatterzeit* und *Flatteranzahl* kann das Übertragungsverhalten der *Einzelmeldungen* der Informationsobjekte *QC9.EM* und *Wischermeldung.WM* definiert werden. Bei der Entflatterung handelt es sich um eine Unterdrückung der Übertragung von IEC 60870-5-104 -Server-Telegrammen, im Falle eines flatterns dieser Meldung am digitalen Eingang. Werden demnach mehr Zustandsänderungen als die parametrierte *Flatteranzahl* am digitalen Eingang innerhalb der parametrierten *Entflatterzeit* erkannt, so wird die Übertragung der Zustandsänderungen bis zum Ablauf der *Entflatterzeit* unterdrückt. Danach beginnt eine erneute Überwachung des digitalen Eingangs. Am Beispiel der in Abbildung 2.12 eingestellten Überwachungszeit und Anzahl an Zustandsänderungen bedeutet dies, dass sobald innerhalb von 5 Sekunden mehr als 2 Zustandswechsel erkannt werden, die Übertragung der folgenden Wechsel nicht an den Client übertragen wird. Die Verwendung der Einzelmeldungen der Rangiertabelle als Rückmeldung für ein Befehl und die Parametrierung einer Gerätelaufzeit ist in dieser Arbeit nicht vorgesehen. Diese Felder sind in der Maske aufgeführt, jedoch deaktiviert.

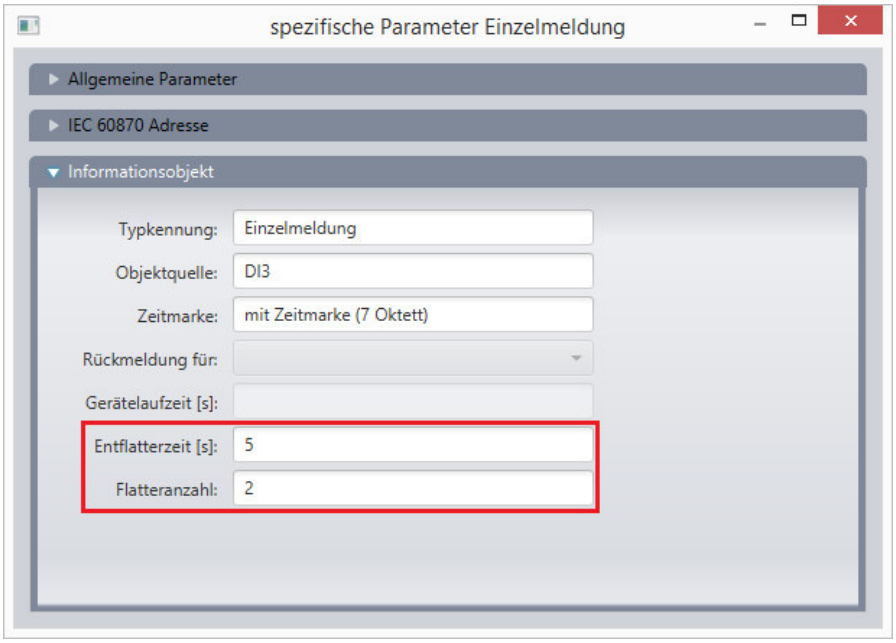


Abbildung 2.12: Spezifische Parameter: Einzelmeldung

2.2.3.4 Messwert

Das Übertragungsverhalten des Informationsobjektes der TK *Messwert* kann mit diversen Verfahren eingestellt werden. In Abbildung 2.13 sind mögliche Parametriermöglichkeiten rot umrandet. Aufgrund des definierten Prozesses der *Prozesssimulation* sind teilweise einzelne Parameter fest vorgegeben. Der *Erfassungsbereich* ist aus der Festlegung des *Sliders* der analogen Eingabe *AI1* des FWG des Simulators abgeleitet. Mit einem Bereich von *4-20mA* als Einheitsgröße wird der *Erfassungsbereich* des Hardwareeingangs im Rahmen dieses Projektes als Auswahlmöglichkeit der *ChioceBox* festgelegt.

Infolge der begrenzten Prozessgröße des *Sliders* zur *Messwertsimulation* innerhalb der *Prozesssimulation* werden auch der *Minimale* und *Maximale Prozesswert* vorgegeben. Dabei soll über den am Abgang angebrachten Stromwandler als *Slider* ausgeführt ein Prozesswert im Bereich zwischen 300A und 350A simuliert werden. Hieraus folgt der in Abbildung 2.13 vorgegebene Bereich des Prozesswertes. Mit diesem erfolgt im weiteren Verlauf die lineare Skalierung des Wertebereiches der analogen Eingabe *AI1*.

Mit den darunter angebrachten und rot umrahmten *CheckBoxen* wird die Anzeige der *Einheit* der nachfolgend zu parametrierenden Größen festgelegt. Initial ist werden alle Größen als *relative* Größen angezeigt. Bei einem Wechsel zur Anzeige *absoluter* Größen ist eine entsprechende Umrechnung der aktuellen Werte durchzuführen. Diese Um rechnung ist gleichermaßen vorzunehmen, wenn von der Anzeige von *absoluten* zu *relativen* Größen gewechselt wird.

Hinsichtlich der *Nullpunktunterdrückung* wird zur Veranschaulichung dieser Parametrierung ein Wert von *30%* relativ bzw. *6mA* absolut vorgegeben. Dabei soll bei diesem Verfahren der zu übertragene Messwert auf Null gesetzt werden, sobald der erfasste Messwert die im Parameter *Nullpunktunterdrückung* eingestellte Grenze unterschreitet.

Zur Identifizierung eines Drahtbruches ist der Parameter *Live-Zero Schwelle* in der *spezifischen Parametrierung* des Messwertes aufgeführt. Das *Live-Zero-Prinzip* ermöglicht dabei zur Erkennung eines schaltungsinternen Leitungsbruches und dient zu Fehlererkennung. Da der Messbereichsanfang von *4mA* (Erfassungsbereich) festgelegt wurde, gilt dieser als Offset. Tritt ein Leitungsbruch auf, ergibt sich ein Signal mit *0mA*. Mit der

parametrierten *Live-Zero-Schwelle* wird bei einem Messwert kleiner *4mA* ein Leitungsbruch eindeutig detektiert werden. Im Rahmen dieser Arbeit wird diese Schwelle als fester Parameter aufgeführt, jedoch findet die Anwendung eines Leitungsbruches innerhalb der Simulationsfälle keine Verwendung, das der *Erfassungsbereich* zwischen *4-20mA* festgelegt wird.

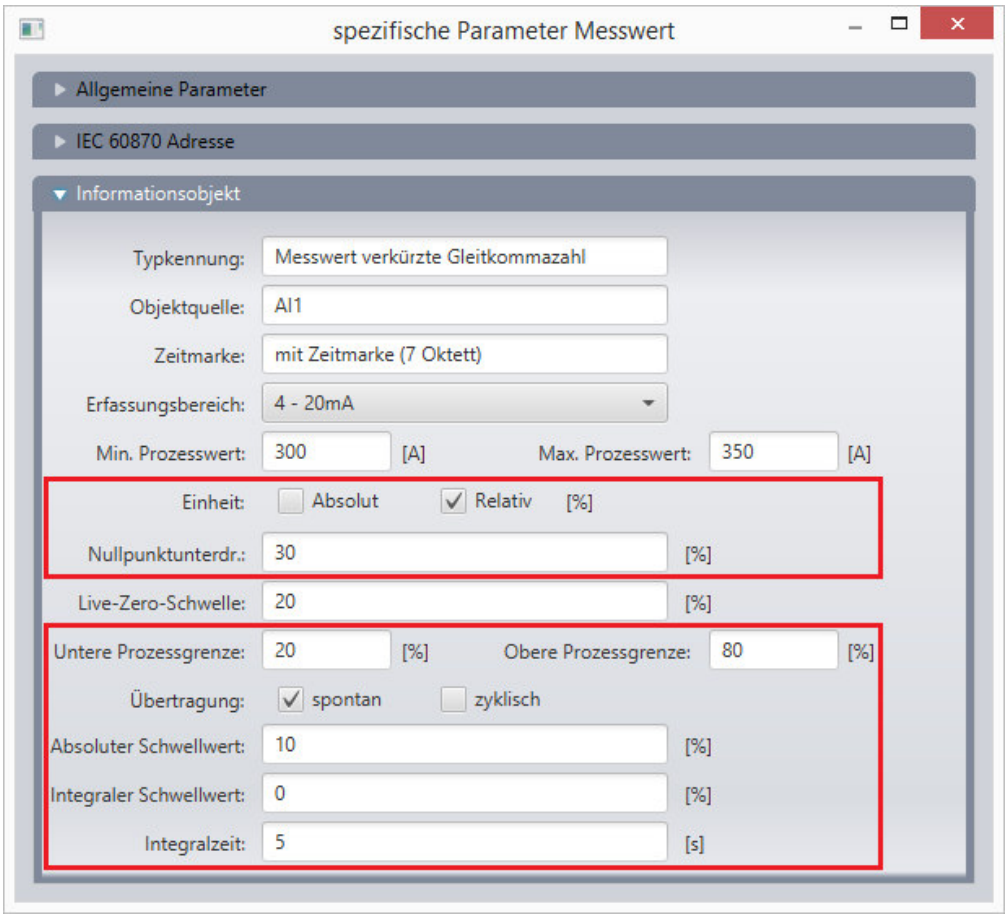


Abbildung 2.13: Spezifische Parameter: Messwert

Mit den Parameter *Untere Prozessgrenze* sowie *Obere Prozessgrenze* sollen Grenzen parametriert werden, die zur Überwachung der Grenzwertverletzung dienen. Unterschreitet der aktuell erfasste Messwert die parametrierte *untere Prozessgrenze* oder überschreitet dieser die *obere Prozessgrenze*, soll der Messwert mit einer Überlaufkennung versehen und spontan übertragen werden. Dabei kann durch das Setzen des Qualitätsbits OV eine

entsprechende Grenzwertverletzung identifiziert. Die zur Demonstration dieses Übertragungsverhaltens gewählten Prozessgrenzen sollen eine spontane Messwertübertragung anstoßen, sobald bei der Messwertsimulation der Messwert von *310A* unterschritten oder *340A* überschritten wird. Dabei erfolgt die Übertragung unabhängig davon, welches Übertragungsverhalten des Messwertes innerhalb der Prozessgrenzen eingestellt ist. Dieses wird mittels den *CheckBoxen spontan* und *zyklisch* definiert. Wird die *CheckBox zyklisch* angewählt, kann ein Intervall (Zykluszeit) in Sekunden angegeben werden, in dem der erfasste Messwert mit der COT *Per/Cyc* übertragen werden soll.

Soll die Messwertübertragung ereignisorientiert (COT:Spont) erfolgen, stehen bei der *spezifischen Parametrierung* bei Selektion der *CheckBox spontan* zwei Verfahren zur Verfügung. Dabei können beide gemeinsam oder jeder einzeln verwendet werden. Mit dem Schwellwertverfahren *Absoluter Schwellwert* wird ein Wert angegeben, ab dem der aktuelle Messwert spontan übertragen werden soll. Dabei erfolgt eine Messwertübertragung, wenn die Abweichung zwischen dem aktuellen Messwert und dem zuletzt übertragenen Messwert im Betrag dem parametrierte Schwellwert entspricht. Dabei wird gemäß Gleichung 2.1 der aktuell erfasste Messwert als x_i und der zuletzt übertragene Messwert als x_0 definiert.

$$|x_i - x_0| > \quad absoluter\ Schwellwert \tag{2.1}$$

In der *spezifischen Parametrierung* in Abbildung 2.13 ist ein absoluter Schwellwert von *10%* als Default-Wert festgelegt. Das bedeutet, dass bei einer Differenz von *5A* zwischen aktuellem und zuletzt übertragenen Messwert ein Messwerttelegramm mit dem aktuellen Messwert spontan übertragen wird. Bei einer Eingabe von *0* als *Absoluter Schwellwert* wird dieses Verfahren nicht angewandt.

Das zweite Verfahren zur spontanen Übertragung eines Messwertes erfolgt nach dem *Integralverfahren*. Für die Anwendung dieses Schwellwertverfahrens wird angegeben, ab welchem Wert der aktuelle Messwert übertragen werden soll. Gleichung 2.2 zeigt die Ermittlung des aktuellen Schwellwertes, um diesen mit dem parametrierten Schwellwert zu vergleichen.

$$\Big|\sum \frac{(x_i - x_0) * \Delta t}{Z_{mw}}\Big| > \quad integraler\ Schwellwert \tag{2.2}$$

Gemäß der Gleichung 2.2 wird die Differenz zwischen aktuellem Messwert (x_i) und dem zuletzt übertragenen Messwert (x_0) addiert und mit dem Erfassungszyklus (Z_{mw}) des Messwertes gewichtet. Ist der Betrag dieses ermittelten Schwellwertes größer als die parametrierte Schwelle, soll der aktuelle Messwert spontan übertragen werden. Die Integralzeit (Δt) beschreibt das Zeitintervall nach dessen Ablauf die Ermittlung des Schwellwertes durchgeführt wird. Erfolgt eine Parametrierung eines *Integralen Schwellwertes* oder einer *Integralzeit* von *0*, wird dieses Verfahren zur Messwertverarbeitung nicht verwendet.

Mittels dieser *spezifischen Parametrierung* der Informationsobjekte kann das Übertragungsverhalten von Zustandsänderungen oder Messwertänderungen individuell definiert werden. Die mit der zu implementierenden Simulationsumgebung realisierbaren Anwendungsfälle werden im nachfolgenden Abschnitt anhand von Sequenzdiagrammen näher beschrieben.

2.3 RtuSimulator als IEC 60870-5-104 Server

Mit der *Parametrierung* des FWG gemäß IEC 60870-5-104 soll die Simulationsumgebung als IEC 60870-5-104 -Server mit einem beliebigen IEC 60870-5-104 -Client kommunizieren. Hierbei sollen entsprechende Informationsobjekte mit ihren Zuständen oder Werten ausgetauscht werden. Im Rahmen dieser Arbeit sind für definierte Anwendungsfälle passende Telegramme von der Simulationsumgebung zu generieren bzw. zu analysieren. In diesem Abschnitt werden die einzelnen Anwendungsfälle und der entsprechende Telegrammaustausch zwischen Client und Server anhand von Sequenzdiagrammen vorgestellt. Dabei sind alle nachfolgend alle von der Simulationsumgebung als IEC 60870-5-104 -Server zu spiegelnden Telegramme in grün dargestellt. In blau sind Client-Telegramme definiert, für die eine Analyse und Identifizierung zu implementieren ist.

Zunächst ist bei einer Kommunikationsverbindung zwischen IEC 60870-5-104 -Server und IEC 60870-5-104 -Client eine *Stationsinitialisierung* gemäß Abbildung 2.14 durchzuführen. Dabei ist das Client-Telegramm im *STARTDT act* U-Format als Aktivierung zum Austausch von Anwenderdaten zu identifizieren. Die Simulationsumgebung soll als Antwort eine entsprechende Bestätigung der Aktivierung (*STARTDT con*) als gespiegeltes Telegramm mit angepasster COT generieren und übertragen. Gleichermaßen ist zu verfahren, wenn der Austausch von Anwenderdaten vom Client mittels *STOPDT act* vom Client deaktiviert wird.

Im Falle eines aktivierten Anwenderdaten-Austausches ist die Identifizierung von U-Telegrammen zur periodischen Prüfung der hergestellten Verbindung seitens des Clients zu implementieren. Auf die Aktivierung des Prüftelegramms *TESTFR act* hat der Server mit einem gespiegelten Telegramm mit der COT *con* als Bestätigung der Aktivierung zu antworten. Diese *Prüfprozedur* soll im Rahmen dieses Projektes ausschließlich durch den Client initiiert werden.

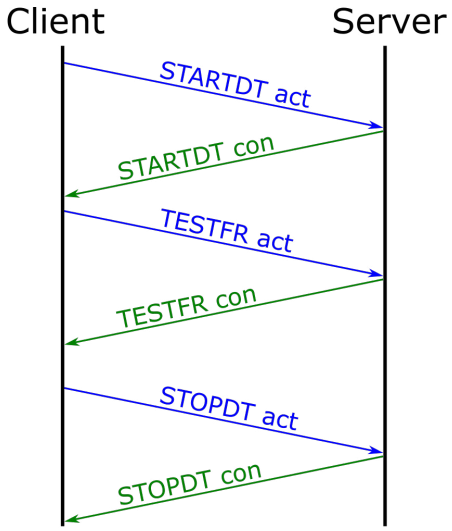


Abbildung 2.14: Stationsinitialisierung

Die Implementierung einer Prüfung der in den Application Protocol Data Unit (APDU) enthaltenen Folgenummern ist zusätzlich für den *Schutz gegen Verluste und Vervielfachung von Nachrichten*. Der IEC 60870-5-104 -Server soll dabei den Empfang eines Client-Telegramms im S-Format identifizieren und die im Telegramm erhaltene Empfangsfolgenummer mit der im eigenen Speicher gesicherten Sendefolgenummer vergleichen. Stimmen Empfangsfolgenummer vom Client mit der Sendefolgenummer des Servers überein, deckt die Anzahl der vom Server gesendeten Telegramme im I-Format mit den vom Client empfangenen Telegramme im I-Format. Hierfür ist entsprechend bei jedem gesendeten Telegramm im I-Format die Sendefolgenummer um eins zu erhöhen. Bei nicht identischen Folgenummern ist die Kommunikationsverbindung zu beenden. Gemäß der Norm IEC 60870-5-104 sollen nach Herstellung einer TCP/IP-Verbindung die Sende- und Empfangsfolgenummern auf null gesetzt. Im Rahmen dieses Projektes ist nur die Identifizierung von S-Telegrammen und der Vergleich der Folgenummern zu implementieren. Ein Senden von S-Telegramm mit entsprechender Empfangsfolgenummer durch den Server ist nicht zu implementieren.

Die Simulationsumgebung soll in der Lage sein einen Generalabfragebefehl zu identifizieren und gemäß Abbildung 2.15 entsprechende Antwort-Telegramme zu generieren und zu übertragen. Zunächst soll die Aktivierung des Generalabfragebefehls (TK=100) mit einem gespiegelten Telegramm und einer COT *ActCon+* bestätigt werden. Darauffolgend sind alle Informationsobjekte mit ihrer jeweilig parametrierten IOA und Qualitätskennung als Telegramme ohne Zeitmarke an den Client zu senden. Dabei enthalten die Qualitätskennungen SIQ der *Einzelmeldung* und DIQ der *Doppelmeldung* den entsprechenden Prozesszustand und zusätzlich den aktuellen Zustand der Qualitätsbits. Das Messwert-Telegramm enthält den aktuellen Prozesswert und ebenfalls zusätzlich den aktuellen Zustand der Qualitätsbits. Mit der COT *Inrogen* werden die Informationsobjekte für die Ursache einer *Stationsabfrage* definiert. Abgeschlossen wird der Generalabfrage mit der Beendigung der Aktivierung als gespiegeltes Telegramm mit der COT *ActTerm+*.

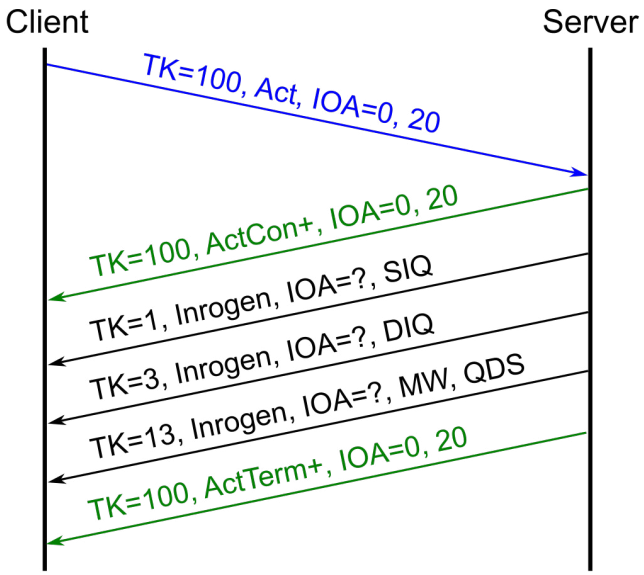


Abbildung 2.15: Generalabfrage

Mit dem Sequenzdiagramm in Abbildung 2.16 wird der festgelegt, der im Falle einer erfolgreichen Befehlsgabe durch den Client zu implementieren ist. Neben der Analyse des vom Client gesendeten *Doppelbefehls* der TK=59 mit Zeitmarke ist zudem mit dem Doppelbefehlszustand Double Command State (DCS) der Befehlskennung die Schaltrichtung zu identifizieren. Diese soll für die Bestätigung der Aktivierung (*ActCon+*) genutzt werden. Je nachdem, ob die Übertragung der Zwischenstellung der *Doppelmeldung* parametriert wurde, soll zunächst diese als Doppelmeldungszustand Double-Point Information (DPI) der Qualitätskennung DIQ mit der COT *Retrem* als Rückmeldung verursacht durch einen Fernbefehl übertragen werden. Dabei ist die parametrierte IOA dieses Informationsobjektes für das Telegramm zu verwenden. Nach Ablauf der Simulationszeit des Leistungsschalters und Erreichen der Endstellung dieses, hat die Übertragung der Endstellung zu erfolgen. Mit dem gespiegelten Befehlstelegramm wird die positive Beendigung der Aktivierung mit der COT *ActTerm+* abgeschlossen.

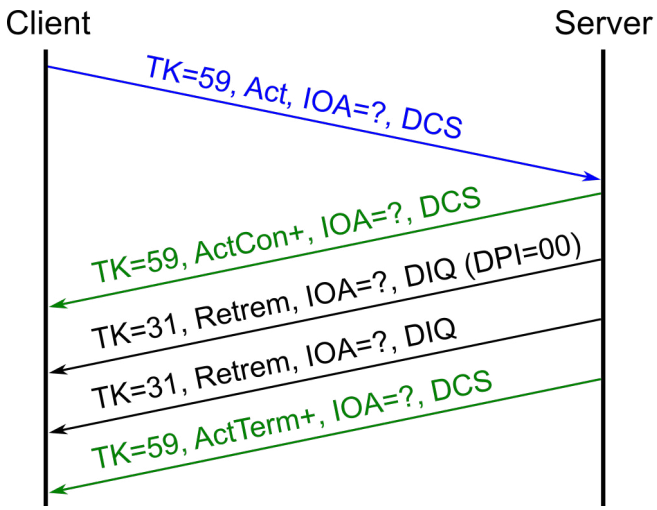


Abbildung 2.16: Erfolgreiche Befehlsgabe: Doppelbefehl

Um eine negative Befehlsaktivierung zu simulieren, ist der Fall eines Verriegelungsverstoßes mittels der in Abbildung 2.17 dargestellten Telegramme zu implementieren. Hierfür ist im Falle einer Befehlsaktivierung durch den Client zunächst festzustellen, ob der Erdungsschalter als Einzelmeldung ausgeführt eingeschaltet ist. Trifft dies zu, soll die Aktivierung des Befehls als gespiegeltes Telegramm mit der COT *ActCon-* als negative Befehlsaktivierung beantwortet werden. Zusätzlich ist zur Identifizierung der Ursache der negativen Befehlsaktivierung eine *Wischermeldung* zu übertragen. Diese ist in der Rangiertabelle parametrierbar, sodass die entsprechende IOA dieses Informationsobjektes für die Telegrammgenerierung verwendet werden kann. Das Telegramm dieser *Wischermeldung* aus zwei aufeinanderfolgenden Informationsobjekten der mit identischer IOA und Zeitmarke aufgebaut. Es unterscheiden sich lediglich die Zustände der jeweiligen *Einzelmeldung*. Hierfür soll für das erste Informationsobjekt der Zustand Single-Point Information (SPI)=Ein und das zweite Informationsobjekt der Zustand SPI=Aus übertragen werden. Mit dieser Folge ist dieses Telegramm als *Wischermeldung* definiert.

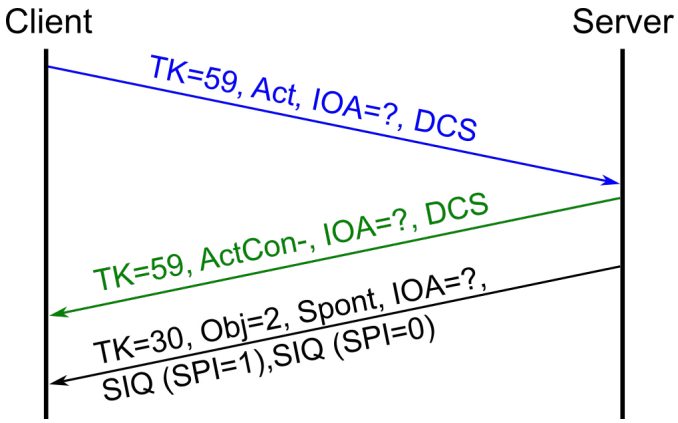


Abbildung 2.17: Negative Befehlsaktivierung: Verriegelungsverstoß

Zwei weitere Anwendungsfälle zur Simulation einer negativen Befehlsaktivierung werden mittels des Sequenzdiagramms in Abbildung 2.18 beschrieben. Dabei wird die Befehlsaktivierung durch den Client vom Server negativ mit der COT *ActCon-* beantwortet. Dies erfolgt im Falle der eingestellten Schalthoheit *Ort* in der *Vor-Ort-Steuerung* oder bei einer Befehlsgabe mit einer Schaltrichtung auf einen bereits existierenden Schaltzustand der *Doppelmeldung*. Das bedeutet, dass eine Befehlsaktivierung negativ beantwortet wird, wenn beispielsweise ein Ein-Befehl vom Client abgesetzt wird und der entsprechende Leistungsschalter bereits dieses Schaltzustand aufweist. Gleiches gilt für die Schaltrichtung Aus.

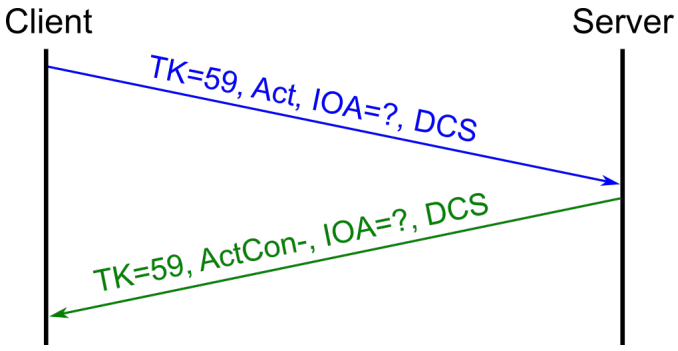


Abbildung 2.18: Negative Befehlsaktivierung: Schalthoheit Ort, EIN->EIN/AUS->AUS

Das Sequenzdiagramm in Abbildung 2.19 zeigt die vom Server zu genierenden IEC 60870-5-104 -Telegramme für den Fall, dass die Rückmeldung zu einem Doppelbefehl nicht innerhalb der parametrierten Gerätelaufzeit eintrifft. Es ist hierfür zunächst der aktivierte Doppelbefehl seitens des Client durch den Server mit einem gespiegelten Telegramm und der COT *ActCon+* bestätigt, wodurch die Befehlsgabe im FWG angestoßen wird. Ist für die Doppelmeldung die Übertragung der Zwischenstellung in der Parametrierung aktiviert, wird diese mit der COT *Retrem* an den Client gesendet. Läuft die parametrierte Gerätelaufzeit der Doppelmeldung ab und der Leistungsschalter befindet sich aufgrund der eingestellten Simulationszeit noch Bewegung in Richtung Schaltrichtung, soll eine negative Beendigung des Befehls an den Client übertragen werden. Die daraufhin vom FWG erfasste Zwischenstellung wird mit der COT *Spont* übertragen. Nach Ablauf der Simulationszeit des Leistungsschalters erreicht dieser seine Zielposition. Dieser erfasste Zustand wird ebenfalls mit der COT *Spont* an den Client zur Aktualisierung des Prozesszustandes gesendet.

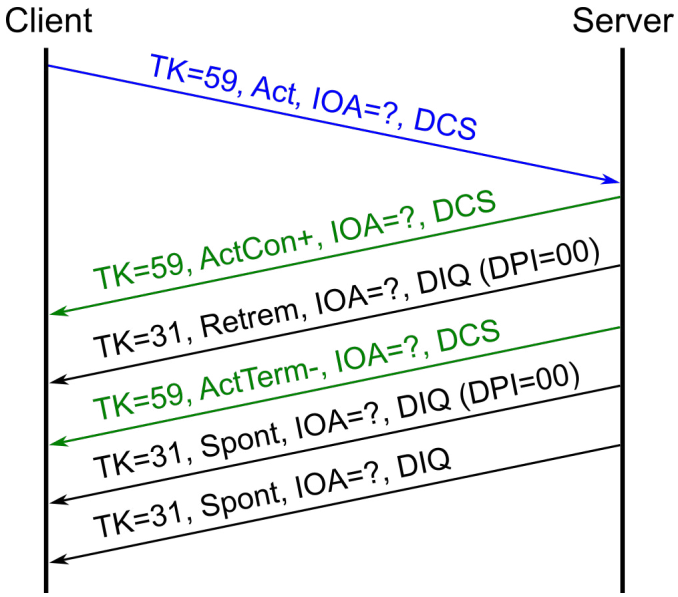


Abbildung 2.19: Negative Befehlsbeendigung: Rückmeldung außerhalb der Gerätelaufzeit

Bei Absetzen eines Doppelbefehls an den Leistungsschalter durch die *Vor-Ort-Steuerung* ist bei Erreichen der Zielposition diese entsprechend an den Client mit der COT *Retloc* zu übertragen. Hierdurch wird die Ursache der Rückmeldung festgelegt, dass diese durch einen örtlichen Befehl verursacht wurde.

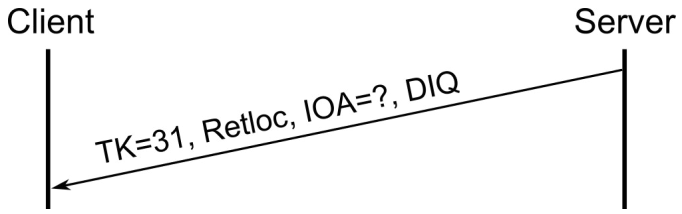


Abbildung 2.20: Rückmeldung verursacht durch einen örtl. Befehl

Zusätzlich zum *Generalabfragebefehl* und dem *Doppelbefehl* soll die Simulationsumgebung einen *Sollwert-Stellbefehl* der TK=63 mit Zeitmarke identifizieren. Zur Umsetzung dieses *Sollwert-Stellbefehls* ist eine Analyse des empfangenen Sollwertes durchzuführen. Im Gegensatz zu einer Befehlsgabe eines Doppelbefehls erfolgt hierfür keine Rückmeldung als Messwert. Es soll lediglich die Aktivierung des Befehls als gespiegeltes Telegramm mit der COT *ActCon+* gefolgt von einem gespiegeltem Telegramm mit der COT *ActTerm+* für die Beendigung der Befehlsaktivierung übertragen werden. Ist in der *Vor-Ort-Steuerung* die Schalthoheit *Ort* eingestellt, soll bei einem *Sollwert-Stellbefehl* vom Client die Befehlskativierung negativ beantwortet werden. Dabei ergibt ein Telegrammverkehr analog zu dem aus Abbildung 2.18, wobei die TK=63 als *Sollwert-Stellbefehl* zu definieren ist und der Sollwert zu spiegeln ist. Insgesamt ist keine Analyse der Qualitätskennung für den *Sollwert-Stellbefehl* Qualifier of Set-Point Command (QOS) zu realisieren.

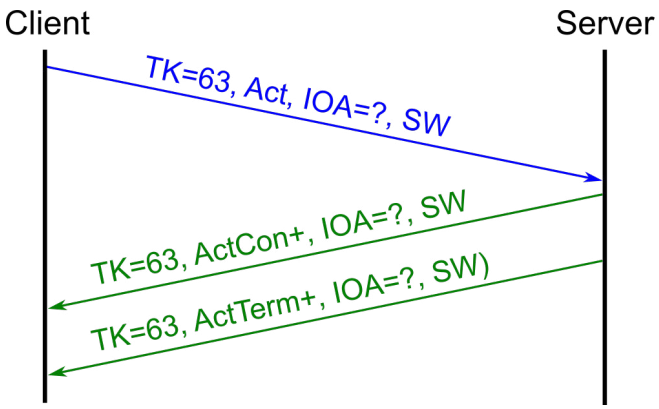


Abbildung 2.21: Erfolgreiche Befehlsgabe: Sollwert-Stellbefehl

Für die Generierung und Übertragtagung von IEC 60870-5-104 -Telegrammen in Melderichtung, welche nicht durch einen Steuervorgang ausgelöst werden, sind die entsprechenden Anwendungsfälle im Sequenzdiagramm in Abbildung 2.22 aufgeführt. Das erste Telegramm ist eine *Einzelmeldung* und spiegelt den Zustand des Erdungsschalters wieder. Bei jeder Zustandsänderung durch den *ToggleButton Einzelmeldung* in der *Prozesssimulation* des Simulators, ist ein IEC 60870-5-104 -Telegramm mit aktueller Qualitätskennung SIQ an den Client mit der COT *Spont* zu übertragen. Das in Abbildung 2.22 vom Server an den Client zu übertragene Telegramm der TK=30 (*Einzelmeldung*) stellt die in der *Prozesssimulation* angebrachte *Wischermeldung* dar. Durch ein Mausklick auf den *Button Wischermeldung* soll dieses Telegramm generiert werden. Dabei besteht es aus zwei aufeinanderfolgenden Informationsobjekten, dessen IOA und Zeitmarke identisch ist. Es unterscheidet sich jeweils der Zustand SPI von *1* gefolgt von *0*. Hierdurch ist ein Telegramm einer *Wischermeldung* gebildet.

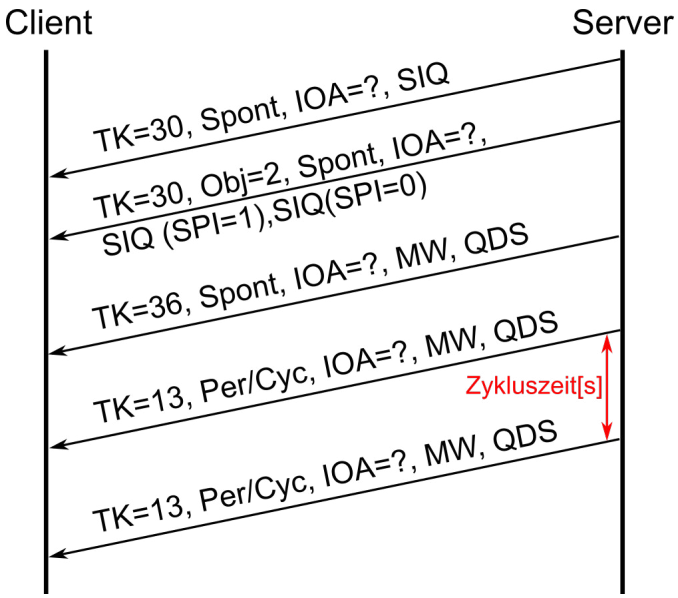


Abbildung 2.22: Spontane Telegrammübertragung

Je nach Festlegung des Übertragungsverhaltens des Messwertes bei der Parametrierung unterscheidet sich die TK. Soll eine ereignisorientierte Messwertübertragung erfolgen, ist bei der Generierung des Messwert-Telegramms die TK=36 mit Zeitmarke zu wählen. Gemäß der parametrierten Schwellen wird bei Überschreiten dieser ein solches Telegramm erzeugt und mit dem aktuellen Messwert (*MW*) und der zugehörigen Qualitätskennung (*qds*) spontan an den Client übertragen. Soll der Messwert periodisch im Intervall der parametrierten *Zykluszeit* an den Client gesendet werden, ist ein Messwert-Telegramm mit der TK=13 ohne Zeitmarke zu generieren. In dem parametrierten Zyklus soll das Telegramm in Melderichtung mit aktuellem Messwert (*MW*) und aktueller Qualitätskennung (*qds*) an den Client übertragen werden.

Die einzelnen Qualitätsbits der bis zum jetzigen Zeitpunkt verwendeten Qualitätskennungen werden mit dem Simulator durch *CheckBoxen* gesetzt. Abbildung 2.23 zeigt den Telegrammverkehr für jeden einzelnen der drei Simulationsfälle. Für den Simulationsfall *1* sollen bei aktivieren/deaktivieren der *CheckBox Handnachführung durch Nahsteuerung* für die *Einzelmeldung* des Erdungsschalters, die Doppelmeldung des *Leistungsschalters* sowie den Messwert des *Stromwandlers* das Qualitätsbit SB der jeweiligen Qualitätskennung mit *0* oder *1* gesetzt werden. Dabei sollen diese Informationsobjekte mit ak-

tueller Qualitätskennung an den Client übertragen werden. Für den zweiten Simulationsfall soll gleichermaßen verfahren werden. Hierbei ist bei aktivieren/deaktivieren der *CheckBox Meldesperre durch Nahsteuerung* für jede Qualitätskennung der betroffenen Informationsobjekte das Qualitätsbit BL mit *0* oder *1* zu belegen. Im Simulationsfall *Baugruppenausfall* erfolgt das Setzen der Qualitätsbits IV und NT mit anschließender Übertragung der Server-Telegramm an den Client. Eine kombinierte Anwendung der drei Simulationsfälle soll zudem ermöglicht werden.

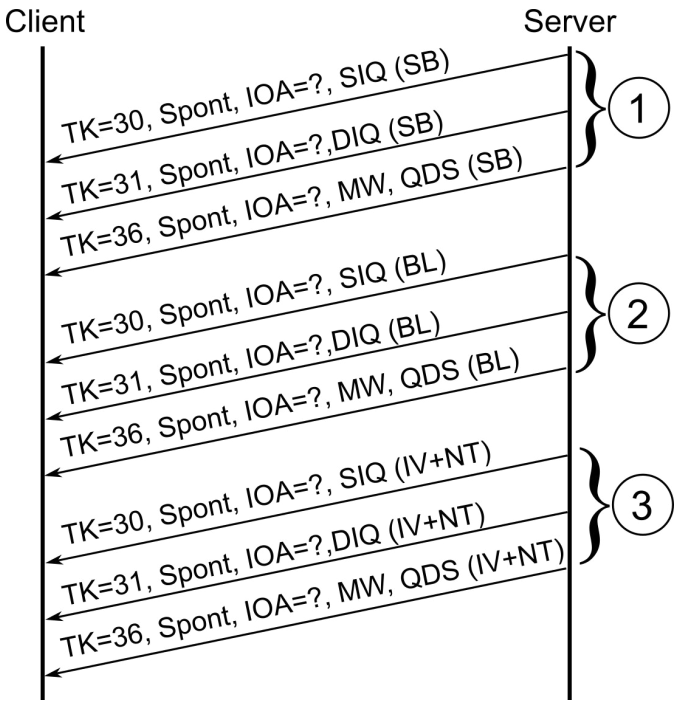


Abbildung 2.23: Spontane Telegrammübertragung: Simulationsfälle

3 Implementierung der Anwendung

In diesem Kapitel erfolgt die Beschreibung der Implementierung zur Erweiterung des Simulators. Die entwickelten Oberflächen zur Parametrierung gemäß IEC 60870-5-104 werden mit dem implementieren Code verknüpft. Die Beschreibung der Umsetzung erfolgt in den nachfolgenden Abschnitten mit Hilfe von Programmablaufplänen. Dabei kann dem Quellcode im Anhang C eine detaillierte Kommentierung entnommen werden. Zusätzlich sind im Anhang B Unified Modeling Language (UML)-Klassendiagramme aufgeführt.

Grundsätzlich gliedert sich die Strukturierung des Software-Projektes in drei Pakte, dessen Aufbau im vereinfachten Klassendiagramm in Abbildung 3.1 dargestellt ist. Das Paket *rtuSimulator* enthält eine Sammlung von Klassen, die Daten verwaltet und anwendungsspezifische Algorithmen realisiert. Zu erkennen sind diese Klassen an der Endung *Model*. Weiterhin befinden sich innerhalb dieses Paketes alle *Controller*-Klassen. Diese sind mit ihrer entsprechenden zur Darstellung zuständigen *FXML*-Datei verknüpft und dienen für die Ereignisbehandlung von Benutzereingaben sowie die Umsetzung von Reaktionen und Logik. Die jeweiligen *FXML*-Dateien sind in der Übersicht in Abbildung 3.1 nicht aufgeführt.

Im Paket *tcpIpServerIec104* sind Klassen implementiert, welche für die Herstellung einer TCP/IP-Kommunikationsverbindung erforderlich sind. Weiterhin sind dort Klassen entgalten, welche die Generierung von-Telegrammen vereinfachen und somit einen Telegrammverkehr gemäß IEC 60870-5-104 ermöglichen. Die Umsetzung der Verwaltung der Folgenummern, welche für das Generieren und Empfangen von Telegrammen sowie Quittieren erforderlich sind, erfolgt mit Klassen und Schnittstellen des Paketes *rtuIec104ServerFolgenummer*.

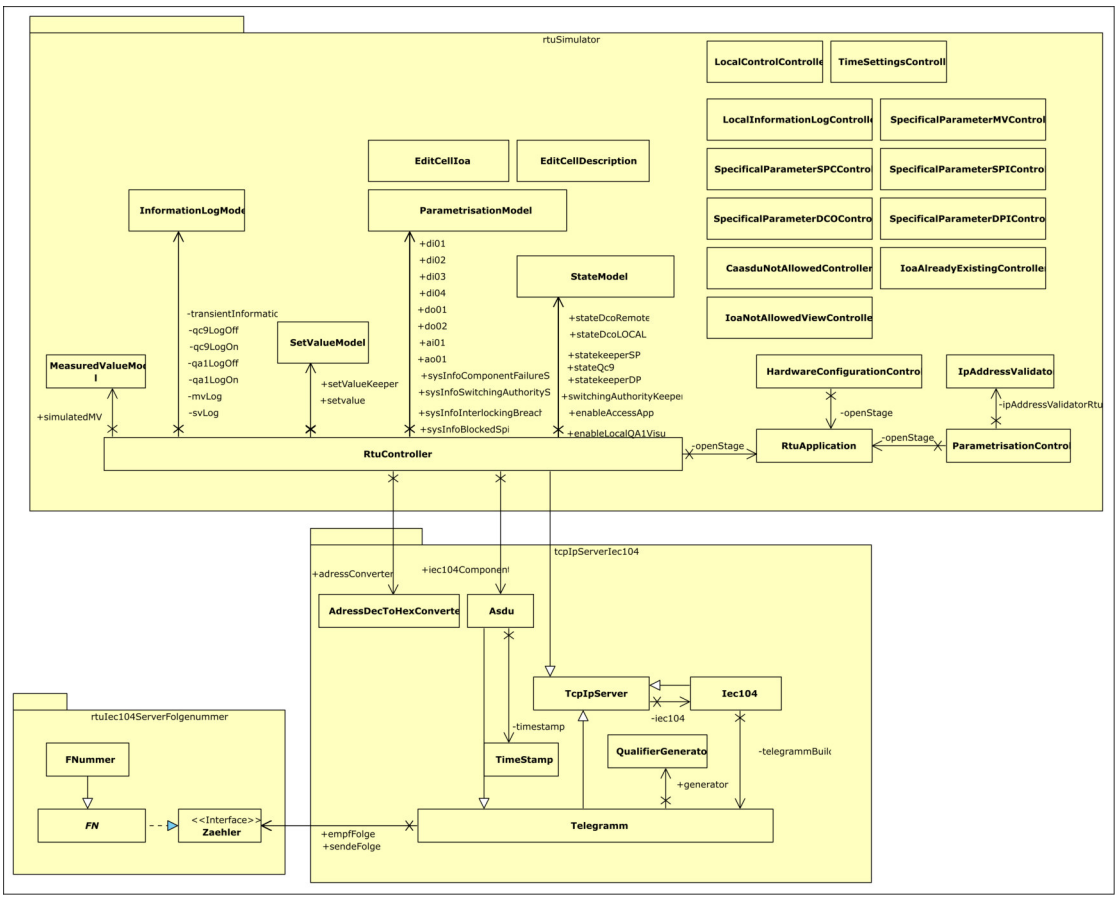


Abbildung 3.1: Klassendiagramm Projekt `RtuSimulator`

Die im nachfolgenden Abschnitt 3.1 beschriebenen *Model*-Klassen `InformationLogModel` und `ParametrisationModel` beliefern die *Controller*-Klasse `RtuController` mit entsprechenden Daten hinsichtlich spezifischer Daten von Informationsobjektes sowie derer Ereignismeldungen. Insgesamt übernimmt der *Controller* `RtuController` eine wesentliche Aufgabe im Rahmen der Simulation des FWG. Die Beschreibung der bereits bestehenden Klassen kann der Arbeit [3] entnommen werden, sodass nachfolgend der Fokus auf ergänzte Klassen besteht.

3.1 Modelle zur Datenverwaltung

Im Rahmen dieses Projektes wurden zwei zusätzliche *Model*-Klassen entwickelt, welche für die Datenverwaltung des *Meldebuches* sowie der Parameter jedes einzelnen Informationsobjektes erforderlich sind. Die *Model*-Klasse `InformationLogModel` aus Abbildung 3.2 ermöglicht es Informationen für die Protokollierung von Ereignissen im *Meldebuch* zur Verfügung zu stellen. Mit einer Instanziierung eines Objektes dieser Klasse stellt die Klassenvariable `info` für den Erdungsschalter, den Leistungsschalter, den Messwert sowie der Wischermeldung den geänderten Zustand/Wert zur Verfügung. Mittels der entsprechenden Setter-Methode `setInfo()` kann der jeweilige Zustand gesetzt und mittels der Getter-Methode `getInfo()` abgefragt werden. Zu jeder Zustandsänderung wird der entsprechende Zeitstempel mit der Methode `setDateTime()` gesetzt und kann zum Befüllen des *Meldebuches* mit der Methode `getDateTime()` abgefragt werden.

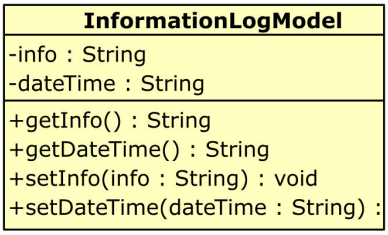


Abbildung 3.2: UML-Diagramm der Klasse `InformationLogModel`

Eine bedeutende Klasse zur Abfrage der individuellen Parameter der parametrierten Informationsobjekte bildet die *Model*-Klasse `ParametrisationModel`. Die implementierten Klassenvariablen und Methoden sind im UML-Klassendiagramm in Abbildung 3.3 dargestellt. Für jedes Informationsobjekt wird eine Instanz dieser Klasse gebildet. Die einzelnen Parameter jedes Objektes können mit der entsprechenden Setter-Methode gesetzt werden. Fest definierte Parameter wie die TK (`timetag`), Baugruppe (`module`), Klemme (`terminal`) sowie Zeitmarke (`timetag`) werden initial gesetzt und sollen im Verlauf der Laufzeit der Anwendung nicht angepasst werden können. Da im Rahmen dieses Projektes eine vordefinierte Rangiertabelle zur Verfügung gestellt wird, erfolgt zudem das initiale Setzen der CAASDU (`caasdu`), IOA (`ioa`) sowie Beschreibung (`description`). Im Laufe der Parametrierung sollen diese individuell angepasst und abgefragt werden können.

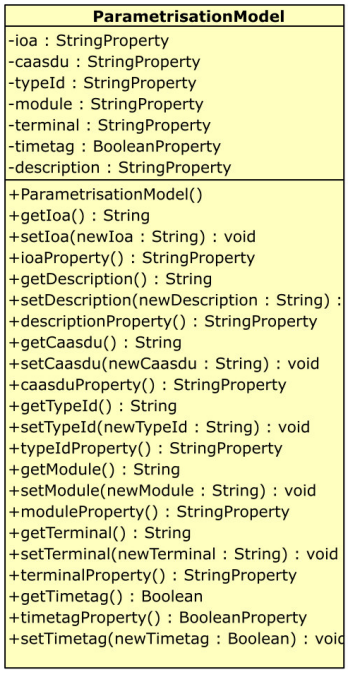


Abbildung 3.3: UML-Diagramm der Klasse `ParametrisationModel`

3.2 Stationsinitialisierung und Übertragungssteuerung

Die Herstellung einer Kommunikationsverbindung zwischen der entwickelten Simulationsumgebung und einem beliebigen Client erfolgt auf Basis einer TCP/IP-Verbindung. Die hierfür implementierte Klasse `TcpIpServer` ermöglicht den Aufbau einer Verbindung durch die Bereitstellung eines `serverSocket`. Dabei wird der in Abbildung 3.4 dargestellte Vorgang durch den auf der Benutzeroberfläche positionierten *ToggleButton* ausgelöst. Sobald der Modus von *Offline* in *Online* umgestellt wird, wird die lokale IP-Adresse ermittelt und der Port *2404* für die Anwendung des Fernwirkprotokolls IEC 60870-5-104 definiert, sodass die Anwendung auf eine Verbindungsanfrage seitens des Clients wartet. Sobald eine Anfrage empfangen wird, kann der auf TCP/IP basierende Verbindungsaufbau und somit die Stationsinitialisierung erfolgreich durchgeführt werden. Über die von der Klasse `TcpIpServer` zur Verfügung gestellte Methode `getInputStream()` können Datenströme vom Client eingelesen und für die nachfolgende Telegrammanalyse verwendet werden.

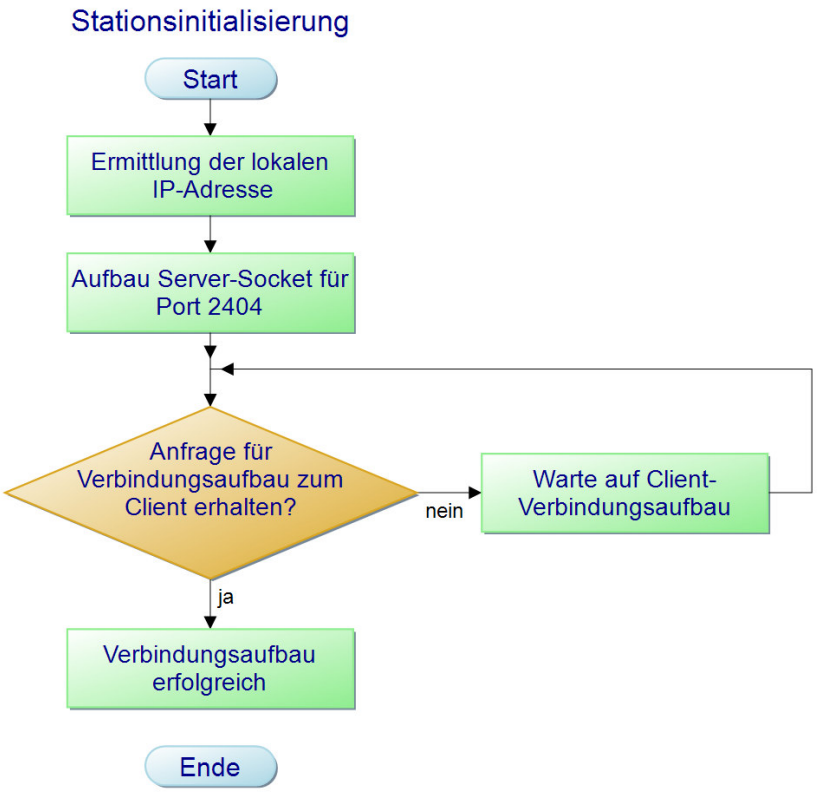


Abbildung 3.4: Programmablaufplan: Stationsinitialisierung

Voraussetzung für einen Austausch von Anwenderdaten zwischen der Simulationsumgebung als IEC 60870-5-104 -Server und einem IEC 60870-5-104 -Client ist die Durchführung einer Aktivierungsprozedur. Hierfür werden nicht benummerte Telegramme im U-Format vom Client erwartet. Zur Identifizierung des Formates eines empfangenen Telegramms wird der eingelesene Datenstrom der Methode `teleAnylse()` der Klasse `Iec104` übergeben. Innerhalb dieser erfolgt eine Auswertung des Formates. Abbildung 3.5 zeigt die Prozedur der Telegrammanalyse und die darauffolgenden möglichen Schritte.

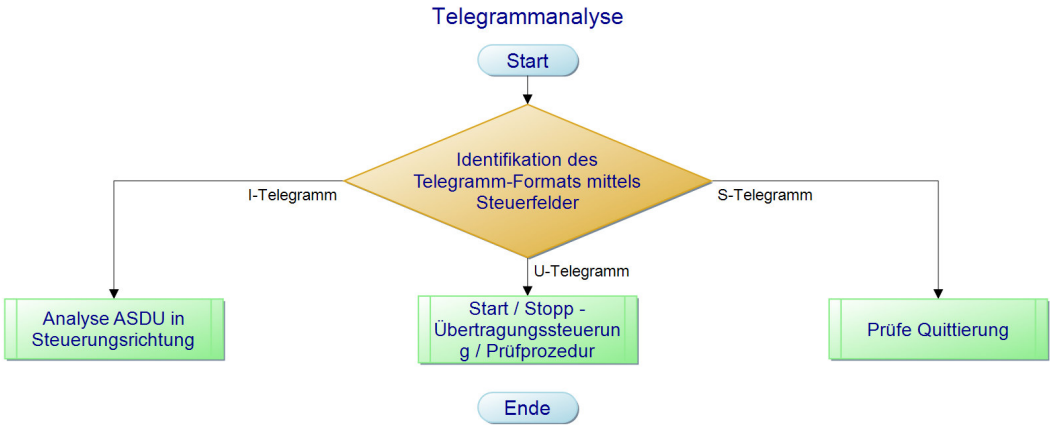


Abbildung 3.5: Programmablaufplan: Telegrammanalyse

Bei der Identifizierung eines empfangenen Telegramms im I-Format erfolgt eine weitere Analyse der ASDU hinsichtlich der TK. In Abschnitt 3.4 wird die Vorgehensweise näher erläutert. Der Empfang eines U-Telegramms vom Client veranlasst eine Prüfung der Folgenummern zur Quittierung gesendeter I-Telegramme. Im nachfolgenden Abschnitt 3.3 wird anhand eines Programmablaufplanes dieser Prüfvorgang beschrieben. Stimmt das Format des empfangenen Telegramms mit dem Format eines U-Formates überein, wird dieses für eine weitere Auswertung der Methode `utele()` übergeben. Je nach Funktion des Telegramms handelt es sich entweder um eine Start/Stopp- Übertragungssteuerung oder eine Prüfprozedur. Eine Beschreibung der Prüfprozedur der hergestellten Verbindung wird im nachfolgenden Abschnitt 3.3 durchgeführt.

Die Aktivierung des Austausches von Anwenderdaten in Form von I-Telegrammen wird durch den Client mit einem U-Telegramm mit der Funktion *STARTDT act* eingeleitet. In Abbildung 3.6 ist der Ablauf für diesen Vorgang abgebildet. Dabei ist zunächst eine erfolgreiche Stationsinitialisierung und somit eine aktive Verbindung zwischen Server und Client erforderlich. Für die Unterscheidung des empfangenen U-Telegramms hinsichtlich seiner Funktion als Prüftelegramm (*TESTFR*) oder als Steuerung der Datenübertragung (*STARTDT*/*STOPDT*) wird in der Methode `utele()` der Klasse `Telegramm` das erste Oktett des Steuerfeldes analysiert. Ein Telegramm mit der Funktion *STARTDT act* führt zu einer entsprechenden Generierung einer Antwort auf diese Aktivierung. Innerhalb der Methode `utele()` wird ein entsprechendes Telegramm im U-Format als Bestätigung

STARTDT con an den Client unter Anwendung der Methode `outWrite()` der Klasse `TcpIpServer` gesendet. Entsprechend diesem Prozedere wird verfahren, wenn die Simulationsumgebung als IEC 60870-5-104 -Server eine Deaktivierung der Datenübertragung *STOPDT act* empfängt. Als Antwort wird eine Bestätigung an den Client mit einem U-Telegramm *STOPDT con* übertragen.

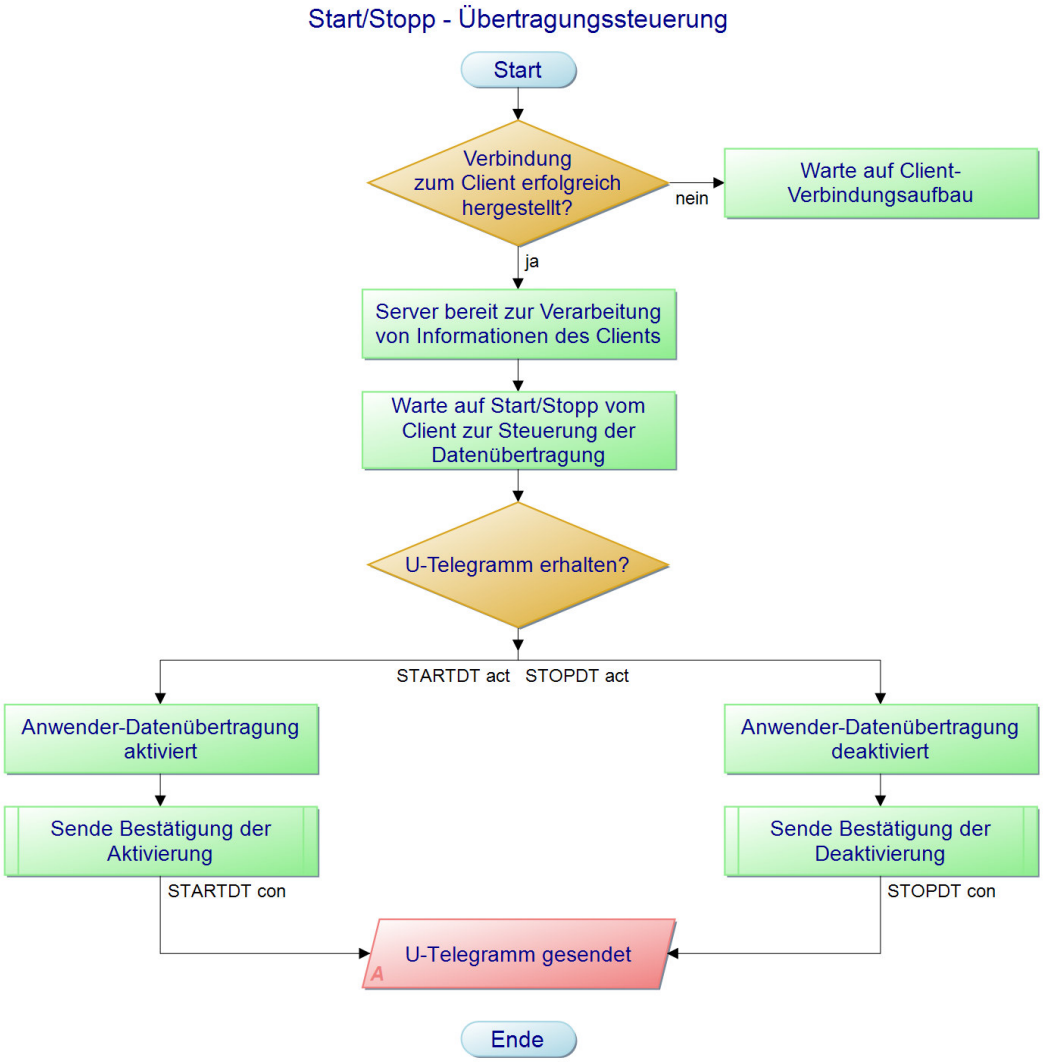


Abbildung 3.6: Programmablaufplan: Start/Stopp - Übertragungssteuerung

3.3 Prüfprozedur und Quittierung

Nachdem eine erfolgreiche Stationsinitialisierung und die Datenübertragung zwischen dem IEC 60870-5-104 -Server und IEC 60870-5-104 -Client aktiviert ist, muss diese Verbindung im Falle eines nicht vorhandenen Austausches von Anwenderdaten (I-Telegrammen) periodisch geprüft werden. Laut der Norm IEC 60870-5-104 dürfen beide Stationen (Unterstation (Server)/Zentralstation (Client)) Prüftelegramme an die Gegenstation senden. Im Rahmen dieser Arbeit wird die Aktivierung der Prüfprozedur ausschließlich durch den Client durchgeführt. Das bedeutet, dass gemäß Abbildung 3.7 die Simulationsumgebung auf die Aktivierung bei Erhalt eines Prüftelegramms reagiert und die Prüfprozedur somit abschließt.

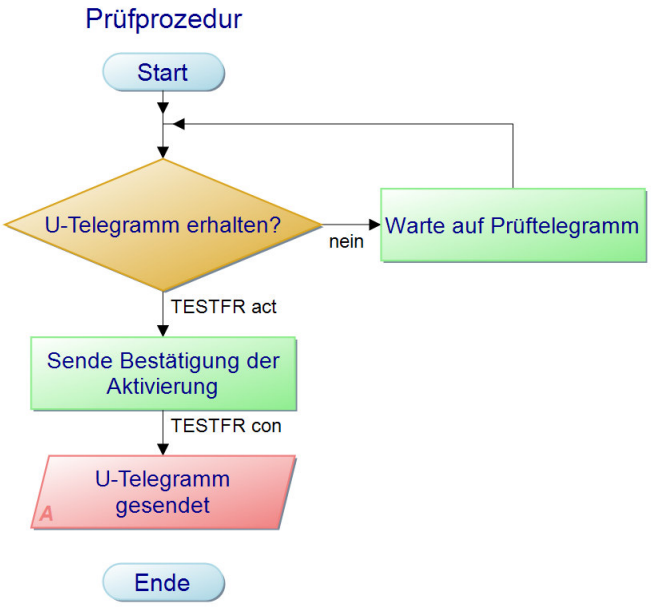


Abbildung 3.7: Programmablaufplan: Prüfprozedur

Nach einer im Client festgelegten Zeit (Empfehlung gemäß IEC 60870-5-104 : 20s) ohne Austausch von Anwenderdaten leitet dieser die Prüfprozedur ein. Durch ein U-Telegramm mit der Funktion *TESTFR act* wird diese aktiviert. Der Server identifiziert dieses Prüftelegramm innerhalb der Methode `utele()` als Aktivierungsvorgang und beantwortet die Aktivierung mit einer Bestätigung *TESTFR con*.

Zum Schutz gegen Verlust und Vervielfachung von Nachrichten ist im Rahmen einer Quittierung der Vergleich der Folgenummern durchzuführen. Bei jeder gesendeten APDU (I-Telegramm) wird die Sendefolgenummer um eins erhöht. Der Empfänger dieser erhöht wiederum die Empfangsfolgenummer um eins. Jeder Sender einer APDU hat seine Sendefolgenummer in einem internen Speicher vorzuhalten. Nach Erreichen einer Anzahl gesendeter APDU von acht, hat der Empfänger diese empfangenen Telegramme mit einem U-Telegramm zu quittieren. Dieses Telegramm enthält wiederum die gespeicherte Empfangsfolgenummer. Der Empfänger des S-Telegramms kann die empfangene Empfangsfolgenummer mit der im eignen Speicher vorliegenden Sendefolgenummer vergleichen. Stimmen beide Folgenummern überein, besteht kein Verlust gesendete und empfangener Telegramme. Werden innerhalb einer festgelegten Überwachungszeit (Empfehlung gemäß IEC 60870-5-104 : 10s) keine APDU ausgetauscht werden, ist ebenfalls ein S-Telegramm mit der Sendefolgenummer an die Gegenstation zu senden. Im Rahmen dieser Arbeit erfolgt keine Implementierung einer serverseitigen Quittierung empfangener APDU. Es wird ausschließlich das vom Client empfangene S-Telegramm zum Vergleich mit der intern gesicherten Sendefolgenummer genutzt.

Nach Empfang eines Client-Telegramms und der Identifizierung dieses durch die Methode `teleAnalyse()` als S-Telegramm, wird die in Abbildung 3.8 aufgezeigte Prüfung vollzogen. Dabei wird das empfangene Telegramm der Methode `stele()` übergeben, welche die im S-Telegramm enthaltene Empfangsfolgenummer ausliest und entsprechend zur weiteren Verarbeitung bereitstellt. Dabei erfolgt der durchzuführende Vergleich der Folgenummern innerhalb der Methode `checkExchangedTele()` der Klasse `Iec104`. Die im S-Telegramm enthaltene Empfangsfolgenummer sollte der im internen Speicher vorliegenden Sendefolgenummer entsprechen. Dabei wird die Sendefolgenummer bei jedem vom IEC 60870-5-104 -Server gesendetem I-Telegramm um eins erhöht. Hierfür wird die Methode `outZahl()` der Klasse `FNummer` verwendet. Diese ermöglicht durch die Schnittstelle `Zaehler` das halten der Sendefolgenummer in einem Speicher. Sind die Folgenummern im Rahmen des Vergleiches identisch, ist die Quittierung erfolgreich. Weichen die Stände voneinander ab, besteht ein Fehler in der Folgenummer und die Verbindung zum Client muss mittels der Methode `disconnect()` der Klasse `TcpIpServer` getrennt werden. Nach einer erneuten Stationsinitialisierung werden Sende- und Empfangsfolgenummern auf null gesetzt.

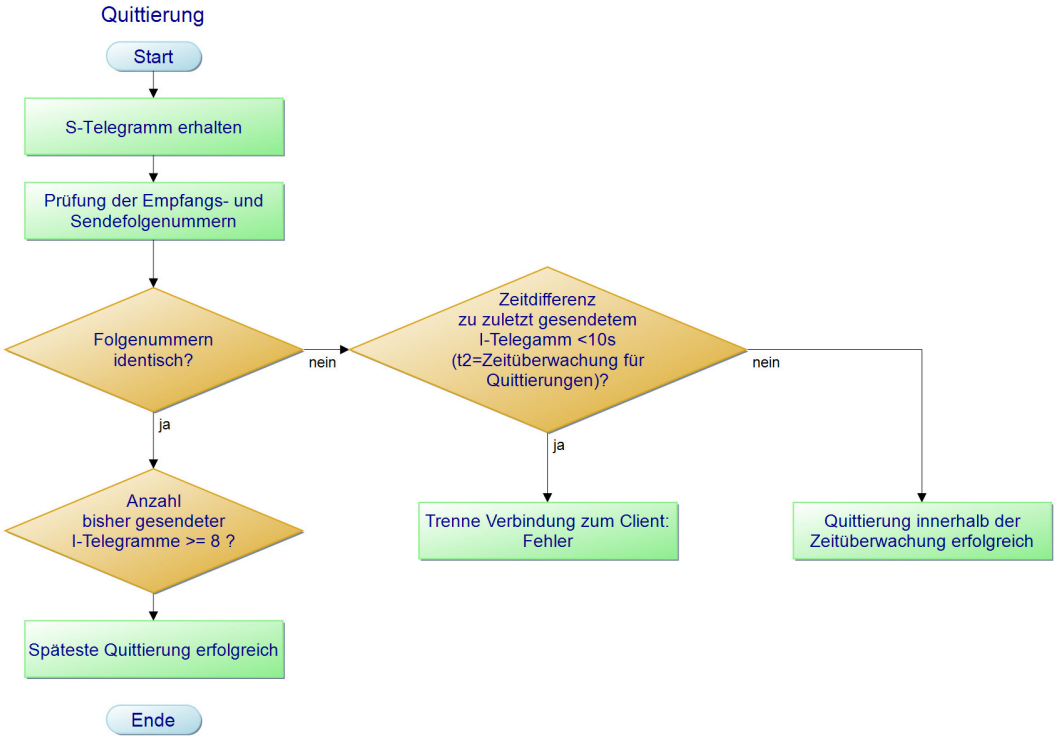


Abbildung 3.8: Programmablaufplan: Quittierung

3.4 Telegrammempfang und -analyse

Nach Empfang eines IEC 60870-5-104 -Telegramms vom Client erfolgt mittels der Methode `teleAnalyse()` der Klasse `Iec104` die Identifizierung des Formates. Wird ein Telegramm im I-Format erkannt, erfolgt zunächst eine Analyse der ASDU in Steuerungsrichtung. Der in Abbildung 3.9 stellt den Vorgang bei dieser Auswertung dar. Dabei wird das empfangene I-Telegramm an die Methode `itele()` der Klasse `Telegramm` übergeben. Dort wird zunächst die Empfangsfolgenummer um eins hochgezählt. Daraufhin wird die in der ASDU enthaltene TK identifiziert. Dabei ist die Simulationsumgebung in der Lagen drei ASDU in Steuerungsrichtung zu verarbeiten. Durch Abfrage der TK des empfangenen I-Telegramm kann in ein *Generalabfragebefehl* der TK=100, *Doppelbefehl* der TK=59 sowie *Sollwert-Stellbefehl* der TK=63 unterschieden werden. Die weitere Prozedur zum Vorgehen nach Erkennen des Steuervorgangs durch den Client wird im

Abschnitt 3.5 beschrieben. Dabei wird der vom Steuerungsvorgang abhängige Vorgang und die entsprechende Generierung von Antworttelegrammen näher beschrieben.

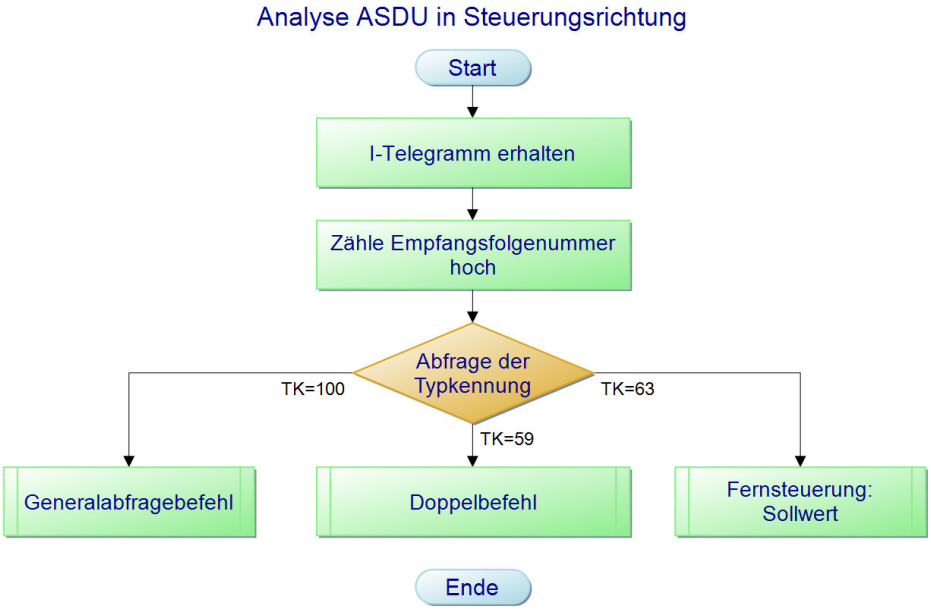


Abbildung 3.9: Programmablaufplan: Analyse ASDU in Steuerungsrichtung

3.5 Telegrammgenerierung und -übertragung

Unabhängig davon, ob eine Telegramm aufgrund eines ausgelösten Steuerungsvorgangs oder einer spontanen Zustands-/Messwertänderungen generiert werden soll, erfolgt innerhalb dieses Abschnittes die Beschreibung der Generierung und Übertragung entsprechender Server-Telegramme anhand der vordefinierten Anwendungsfälle. Hierfür liegt die Klasse `Asdu` vor, welche bei der Erzeugung von I-Telegramm verwendet wird. Sie stellt Methoden zur Verfügung, welche das zu generierende Telegramm als Rückgabewert wiedergeben. Dabei sind die individuellen Übergabeparameter dieser Methoden die COT, IOA sowie ggf. die Qualitätskennung, falls diese nicht fest kodiert ist. Bei Soll- und Messwerten besteht der entsprechende Soll-/ Messwert zusätzlich als Übergabeparameter. Innerhalb der Klasse `Asdu` stehen somit Telegrammbausteine für alle in diesem Projekt zum Einsatz kommenden TK zur Verfügung. Abbildung 3.10 zeigt die zur Anwendung kommenden TK in Melderichtung. Es können ASDU sowohl mit als auch ohne Zeitmarke

erzeugt werden. Die für Aktivierungsvorgänge vom Server zu spiegelnden Telegramme in Steuerungsrichtung sind in der Abbildung nicht dargestellt, werden allerdings innerhalb der Klasse `Asdu` zur Verfügung gestellt.

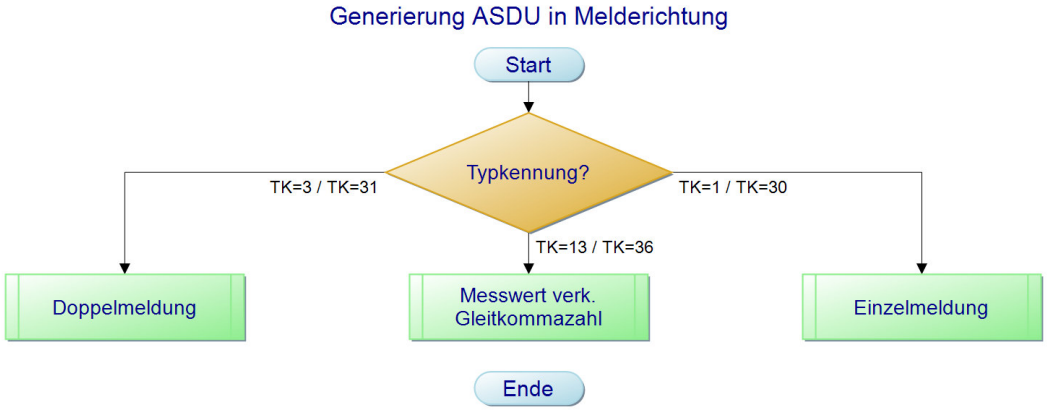


Abbildung 3.10: Programmablaufplan: Generierung ASDU in Melderichtung

Wurde bei einem Telegrammempfang eine entsprechende Analyse durchgeführt und ein *Generalabfragebefehl* identifiziert, erfolgt gemäß Abbildung 3.11 zunächst innerhalb der Methode `itele()` die Feststellung einer Aktivierung (*Act*) dieses Vorgangs. Mittels des entsprechenden Telegrammbausteins der Klasse `Asdu` wird ein Telegramm zur Bestätigung der Aktivierung (*ActCon*) als gespiegeltes Telegramm vom Server an den Client übertragen. Daraufhin werden alle Informationsobjekte mit ihren aktuellen Prozesswerten (Meldungszustand/Messwert) und Qualitätskennungen und der COT *Inrogen* gesendet. Mit dem gespiegelten Telegramm in Steuerungsrichtung mit der COT *ActTerm* wird die Befehlsaktivierung beendet.

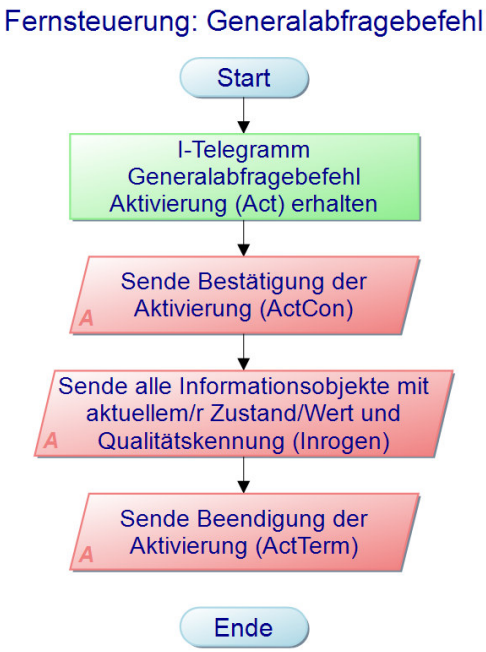


Abbildung 3.11: Programmablaufplan: Fernsteuerung - Generalabfragebefehl

Im Rahmen einer Befehlsgabe durch den Client an den Leistungsschalter ist die in Abbildung 3.12 dargestellte Prozedur implementiert. Nach Erhalt eines I-Telegramms der TK=59 erfolgt zunächst eine Prüfung der in der *Vor-Ort-Steuerung* eingestellten Schalthoheit. Besteht die Schalthoheit *Ort* wird mittels der Methode `commandBlocked()` der Klasse `Telegramm` eine negative Bestätigung der Aktivierung (*ActCon_NEGA*) an den Client übertragen. Für den Fall, dass der Client berechtigt ist einen Steuerungsvorgangs auszulösen, folgen weitere Prüfungen. Dabei wird im Falle eines Verriegelungsverstoßes aufgrund eines geschlossenen Erdungsschalters mittels der Methode `sendeInterlockingBreach()` ebenfalls eine negative Befehlsaktivierung übertragen. Nach Identifizierung der Schaltrichtung mittels der Analyse de Befehlskennung des empfangenen Doppelbefehls wird der Meldungszustand der Doppelmeldung des Leistungsschalters ermittelt. Befindet sich der Leistungsschalter bereits in dem Zustand der Schaltrichtung wird die Befehlsaktivierung negativ beantwortet.

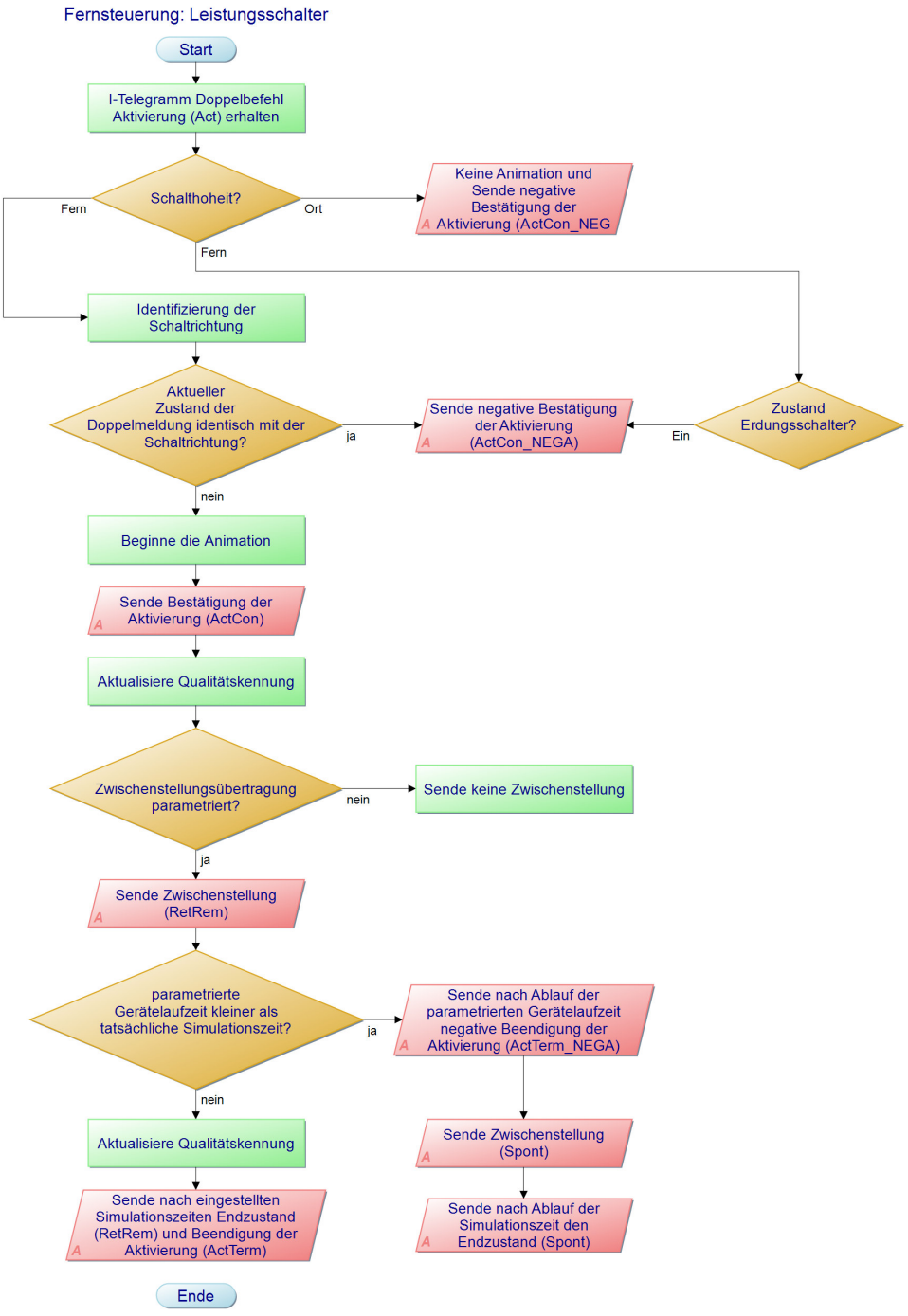


Abbildung 3.12: Programmablaufplan: Fernsteuerung - Leistungsschalter

Unterscheiden sich Doppelmeldungszustand und Schaltrichtung, kann der Befehl von der Simulationsumgebung ausgeführt werden. Dabei wird die Animation der Befehlsgabe in der *Controller*-Klasse `RtuController` durch die Instanz `stateDcoRemote` angestoßen. Mittels der Methode `sendDoublePointInformation()` wird die Telegrammgenerierung und -übertragung umgesetzt. Die Befehlsaktivierung wird mit einem gespiegelten Telegramm und der COT *ActCon* bestätigt. Sobald sich der Leistungsschalter in der Differenzstellung befindet, wird dieser Zustand als Server-Telegramm der TK=31 mit der COT *Retrem* übertragen. Dabei erfolgt zuvor eine Abfrage der in der Parametrierung vorgenommenen Einstellung hinsichtlich der Zwischenstellungsübertragung. Wenn der Leistungsschalter innerhalb der parametrierten Gerätelaufzeit der Doppelmeldung seine Endposition erreicht, wird die entsprechende Qualitätskennung DIQ aktualisiert, da sich der Doppelmeldungszustand DPI nach der Gerätelaufzeit geändert hat. Mit Erreichen der Endposition wird diese als Server-Telegramm mit dem aktuellen Zustand übertragen. Die Befehlsgabe ist abgeschlossen und wird mit *ActTerm* beendet. Für den Fall, dass die parametrierte Gerätelaufzeit kleiner als die festgelegte Simulationszeit des Leistungsschalters ist, wird nach Ablauf der Gerätelaufzeit die Befehlsaktivierung negativ beendet (*ActTerm_NEGA*). Daraufhin folgt die spontane Übertragung der Differenzstellung. Läuft die Simulationszeit des Leistungsschalters ab, wird der Endzustand ebenfalls mit der COT *Spont* übertragen.

Da der Leistungsschalter gleichermaßen über die implementierte *Vor-Ort-Steuerung* angesteuert werden kann, wird dieser Fall anhand des Programmablaufplanes in Abbildung 3.13 beschrieben. Dabei wird bei Absetzen eines Doppelbefehls durch die *Vor-Ort-Steuerung* in der *Controller*-Klasse `RtuController` mit der Methode `simulateDcoWithResponse()` die Schalthoheit geprüft. Bei einer Schalthoheit *Fern* wird keine Animation angestoßen. Andernfalls wird zunächst geprüft, welchen Zustand der Erdungsschalter aufweist. Ist dieser ausgeschaltet kann die Animation und Simulation der Befehlsgabe begonnen werden. Hierbei wird die Instanz `bpSendDpiRetLoc` gesetzt, welche in der Klasse `Telegramm` dazu führt, dass nach Beendigung der Befehlsgabe der entsprechend aktuelle Meldungszustand der Doppelmeldung mit der COT *Retloc* an den Client übertrage wird.

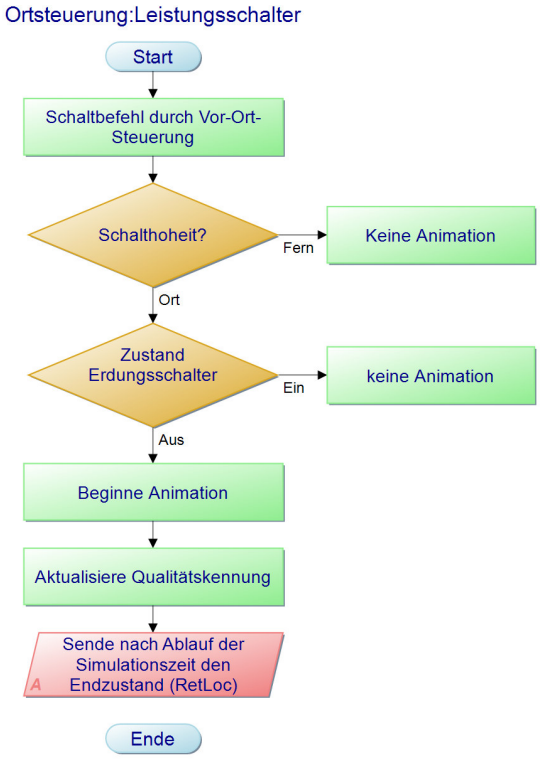


Abbildung 3.13: Programmablaufplan: Ortsteuerung - Leistungsschalter

Im Anwendungsfall einer Fernsteuerung des Sollwertes ist die in Abbildung 3.14 aufgeführte Abfolge implementiert. Es wird nachdem ein *Sollwert-Stellbefehl* vom Client empfangen und in der Methode `itele()` der Klasse `Telegramm` analysiert wurde, zu Beginn die Schalthoheit geprüft. Darf der Befehl aufgrund der einstellten Schalthoheit *Ort* in der *Vor-Ort-Steuerung* nicht umgesetzt werden, ist die Befehlsaktivierung umgehend mit einer negativen Bestätigung zu beantworten. Bei einer eingestellten Schalthoheit *Fern* wird der erhaltene Sollwert vom empfangenen I-Telegramm mittels der Methode `setPointValHexToFloatConverter()` der Klasse `Telegramm` von Hexadezimaler Schreibweise in eine Gleitkommazahl umgewandelt. Im Zuge dessen wird die Aktivierung des *Sollwert-Stellbefehls* bestätigt. Der konvertierten Sollwert kann als Gleitkommazahl genutzt werden, um diesen in der Benutzeroberfläche im Anzeigeinstrument der *Prozesssimulation* anzuzeigen. Die Befehlsgabe wird mit dem gespiegelten Telegramm in Steuerungsrichtung mit der COT *ActTerm* beendet.

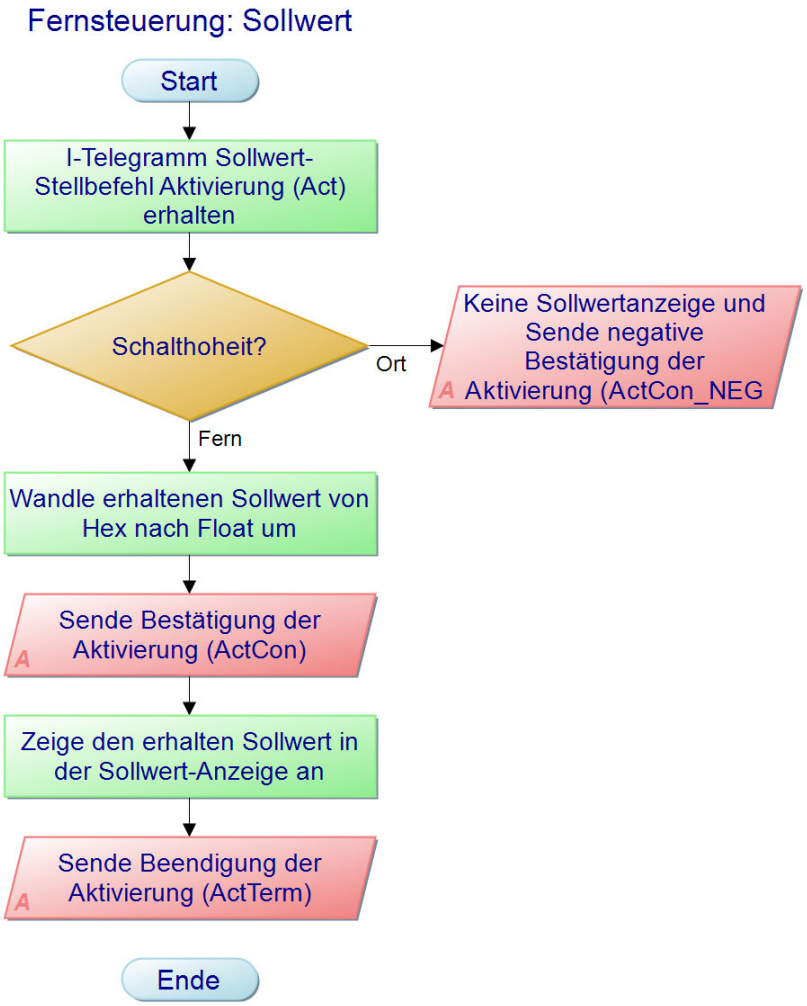


Abbildung 3.14: Programmablaufplan: Fernsteuerung - Sollwert

Mit den auf der Benutzeroberfläche positionierten *CheckBoxen* zur Aktivierung der definierten Simulationsfälle sollen diverse Qualitätsbits aller Qualitätskennungen gesteuert werden. In Abbildung 3.15 sind die implementierten Vorgänge in Abhängigkeit vom Simulationsfall dargestellt. Mit der Methode `simulationCasesListenerRegistration()` der Klasse `Telegramm` werden alle Listener registriert, die für das Setzen der entsprechenden Qualitätsbits zuständig sind. Ist die *CheckBox* des Simulationsfalls *Meldesperre durch Nahsteuerung* aktiviert, wird in der *Controller*-Klasse `RtuController` durch die Metho-

de `simulationCasesQualityDescriptor()` die boolsche Variable `bpBlocked` gesetzt. Diese löst innerhalb der Methode `simulationCasesListenerRegistration()` die entsprechende Generierung und Übertragung der I-Telegramme der Informationsobjekte aus. Dabei erfolgt innerhalb der Methode `simulationCasesListenerRegistration()` vor der Telegrammgenerierung und -übertragung eine Aktualisierung der einzelnen Qualitätskennungen durch die Methoden `spiSiqHandler()`, `dpiDiqHandler()` sowie `mVQdsHandler()`. Hierdurch können die Telegramme mit gesetztem BL-Bit übertragen werden. Mit dieser Vorgehensweise wird ebenso für die beiden Simulationsfälle *Handnachführung durch Nahsteuerung* und *Baugruppenausfall* verfahren. Es besteht außerdem die Möglichkeit, der Aktivierung aller drei Simulationsfälle. Das bedeutet, dass für die letzte Aktivierung die gesetzten Qualitätsbits BL, SB. IV und NT übertragen werden.

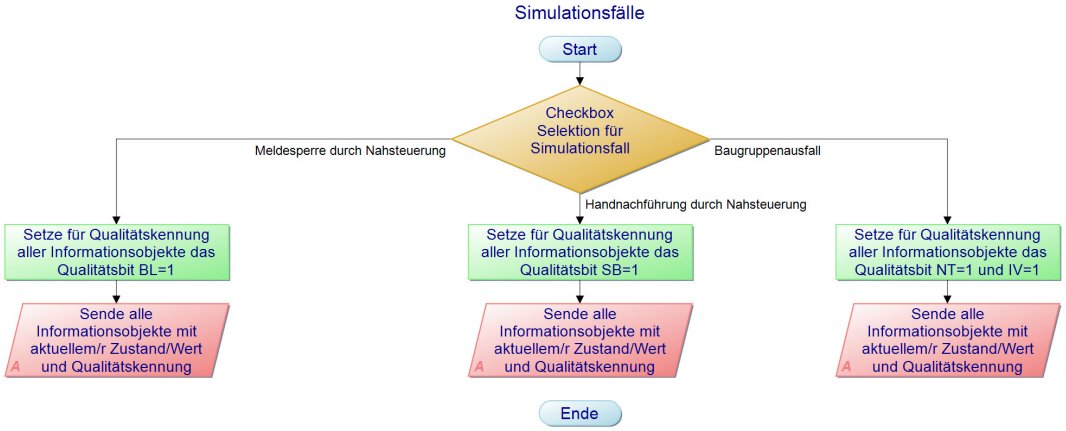


Abbildung 3.15: Programmablaufplan: Simulationsfälle

Eine Übertragung einer Zustandsänderung, welche nicht durch einen Steuervorgang ausgelöst wird, erfolgt im Rahmen dieses Projektes mit der COT *Spont*. Den Anwendungsfall zeigt Abbildung 3.16 anhand einer Einzelmeldung. Die in der *Prozesssimulation* festgelegten Einzelmeldungen als *Erdungsschalter* und *Wischermeldung* lösen jeweils den abgebildeten Vorgang aus. Unabhängig vom Auslöser eine Zustandsänderung durch den *ToggleButton Einzelmeldung* oder den *Button Wischermeldung* wird zu Beginn die individuell parametrierte Flatteranzahl und Entflatterzeit abgefragt. Hierfür sind die Methoden `chattterSuppressionSinglePointInfo()` zur Flatterunterdrückung der *Einzelmeldung* und `chatterSuppressionTransientInfo()` zur Flatterunterdrückung

der *Wischermeldung* in der *Controller*-Klasse `RtuController` implementiert. Mit jedem Mausklick auf dem *Button* bzw. Wechsel der Selektion des *ToggleButtons* wird der Zähler `ipClickCountSpi` um eins erhöht. Dieser dient zum Vergleich des Zustandswechsels mit der parametrierten Flatteranzahl. Dabei wird der Zähler nach Ablauf der Entflatterzeit wieder auf null gesetzt. Übersteigt die Anzahl der Zustandswechsel die parametrierte Flatteranzahl, so wird die Übertragung für die restliche Dauer der Entflatterzeit unterdrückt. Andernfalls werden I-Telegramme erzeugt, die den aktuellen Zustand enthalten, und werden mit der COT *Spont* an den Client gesendet.

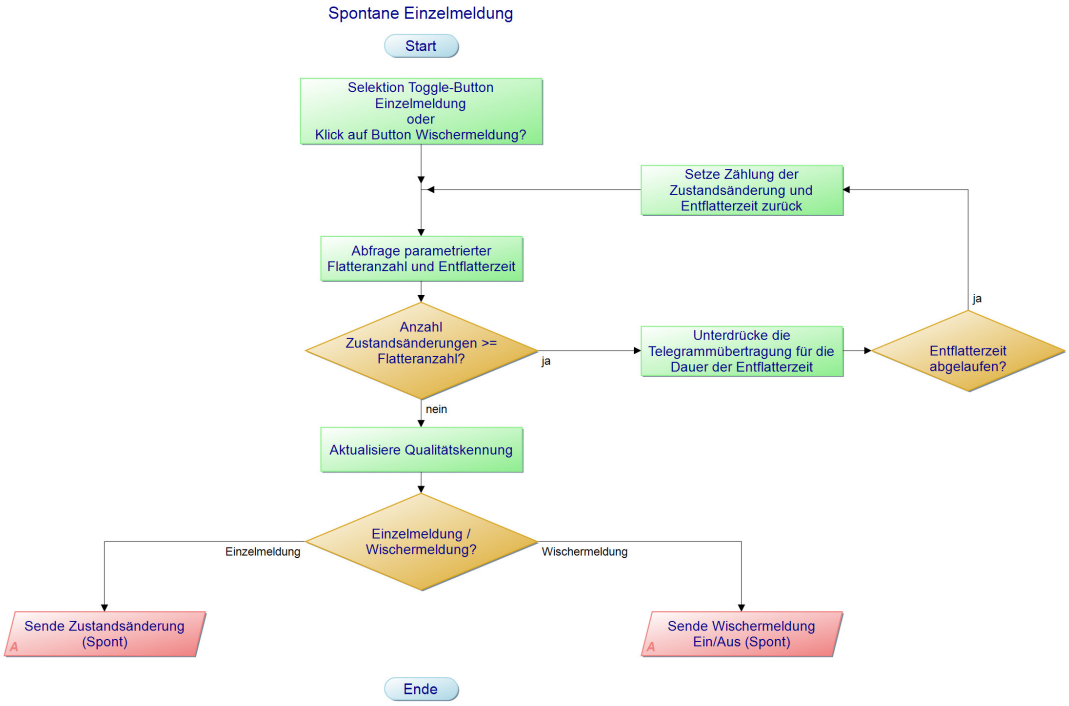


Abbildung 3.16: Programmablaufplan: Spontane Einzelmeldung

Die Messwertübertragung kann je nach parametriertem Übertragungsverhalten spontan oder zyklisch erfolgen. In Abbildung 3.17 ist die Implementierung der Vorgänge für eine spontane Messwertübertragung aufgezeigt. Dabei muss zur Anwendung einer spontanen Übertragung die entsprechende Parametrierung für den Messwert durchgeführt werden. Es besteht die Möglichkeit den Messwert nach zwei Schwellwertverfahren zu übertragen. Ist in der spezifischen Parametrierung des Messwertes

ein *Absoluter Schwellwert* parametriert, wird dieser ermittelt. Mit der Methode `transmitMvAbsoluteThresholdValueSpont()` der Klasse `Telegramm` wird zu jeder Messwertänderung, welche durch den *Slider* in der *Prozesssimulation* umgesetzt werden kann, eine Berechnung der aktuellen Schwellen durchgeführt. Bei einer Parametrierung eines *Integralen Schwellwertes* für die Messwertübertragung wird die parametrierte Schwelle sowie die Integralzeit mittels der Methode `measuredValFloatToHexConverter()` ermittelt und der aktuelle Schwellwert berechnet. Dabei wird mit Hilfe der Klasse `Timer` durch einen Zeitplaner in einem Zeitintervall der parametrierten Integralzeit geprüft, ob die derzeitig aktuell berechnete Schwelle die parametrierte Schwelle überschreitet.

Gleichermaßen wird durch die Parametrierung von *Prozessgrenzen* ein zusätzliches Übertragungsverhalten definiert. In der Methode `processLimitsHandler()` werden die parametrierte obere und untere Prozessgrenze ermittelt und bei jeder Messwertänderung durch den *Slider* in der *Prozesssimulation* eine Über-/Unterschreitung der Prozessgrenzen festgestellt. Wird eine Prozessgrenzenverletzung erfasst, wird die Überlaufkennung des OV-Bits der Qualitätskennung des Messwertes gesetzt. Wird die *integrale* oder *absolute* Schwelle überschritten oder tritt eine Verletzung der parametrierten Prozessgrenzen auf, wird bevor das Telegramm zur Übertragung erzeugt wird eine Prüfung durchgeführt, ob der erfasste Messwert sich im Bereich der *Nullpunktunterdrückung* befindet. Die Methode `zeroSuppression()` bestimmt dabei den parametrierten Nullpunkt und führt diese Untersuchung durch. Unterschreitet der Messwert den parametrierten Wert der *Nullpunktunterdrückung*, wird ein Messwerttelegramm mit einem Prozesswert 0 generiert und übertragen. Unterliegt der aktuell erfasste Messwert keiner *Nullpunktunterdrückung*, so muss der vom *Slider* zur Messwertsimulation erzeugt Messwert mit der Methode `measuredValFloatToHexConverter()` von einer Gleitkommazahl in einen hexadezimalen Wert umgewandelt werden, sodass dieser in das Messwerttelegramm eingebunden werden kann. Nach Aktualisierung der Qualitätskennung des Messwertes mt der Methode `mVQdsHandler()` kann der aktuelle Messwert mit der COT *Spont* an den Client übertragen werden.

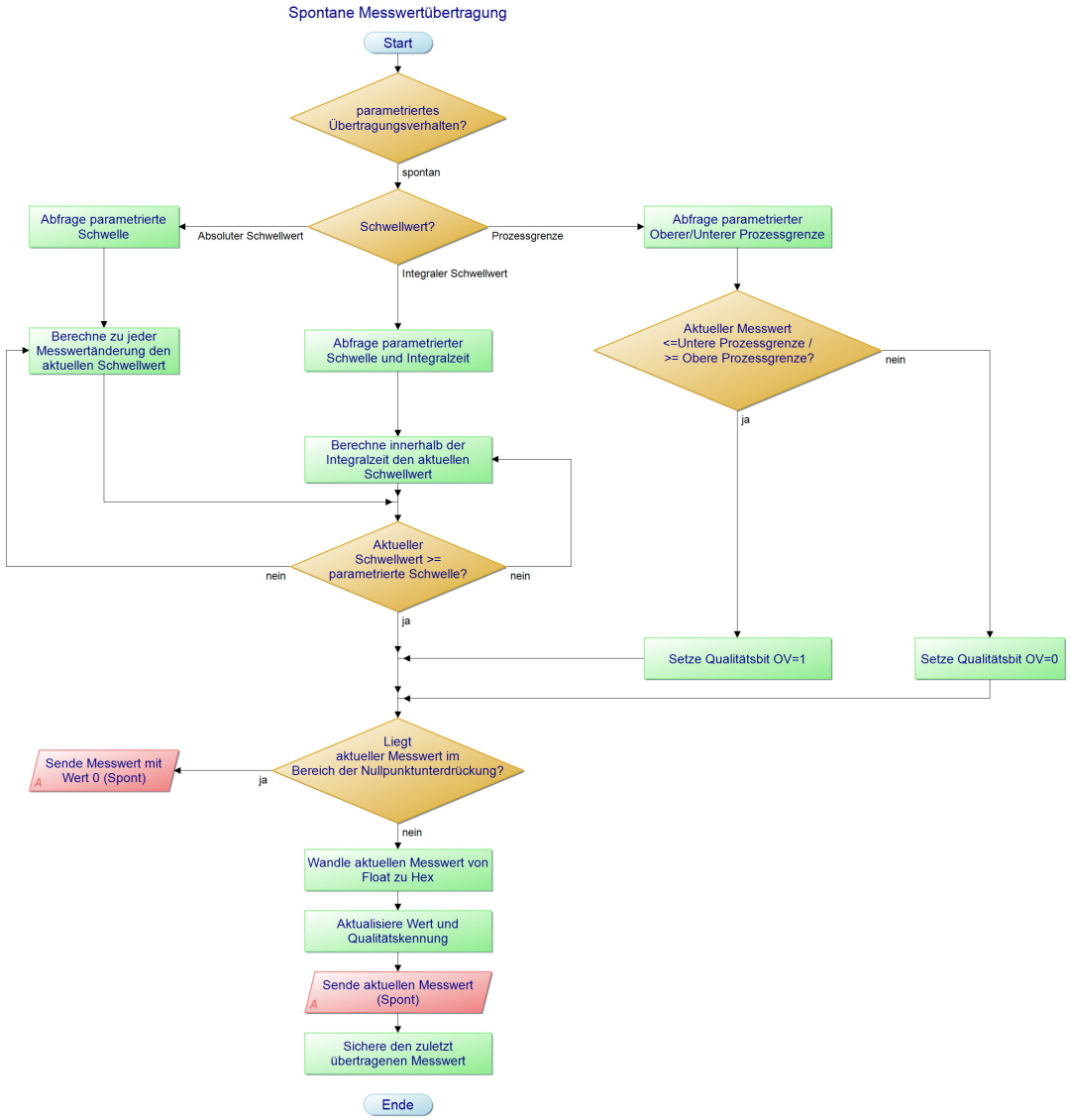


Abbildung 3.17: Programmablaufplan: Spontane Messwertübertragung

Die zyklische Messwertübertragung wird im Rahmen dieser Arbeit mit der Methode `transmitMvCyc()` der Klasse `Telegramm` realisiert. Der Ablauf aus Abbildung 3.18 zeigt die Vorgehensweise der Implementierung. Voraussetzung für die Durchführung einer zyklischen Übertragung ist eine entsprechende Parametrierung des Übertragungsverhaltens des Messwertes. Dabei wird mittels der Methode `transmitMvCyc()` dieses zunächst ab-

gefragt. Mit den Klassen `Timer` und `TimerTask` wird der zeitgesteuerte Ablauf geplant. Dabei wird als Periode die parametrierte Zykluszeit übergeben. Die auszuführende Aufgabe ist mittels der Klasse `TimerTask` definiert und fragt den aktuellen Wert des *Sliders* zur Messwertsimulation im parametrierten Intervall ab. Bei Erreichen des Bereiches der *Nullpunktunterdrückung* wird ein Messwert mit der COT *Per/Cyc* von 0 an die Client übertragen. Dabei wird dieser Nullwert solange periodisch übertragen, bis der Nullpunktbereich verlassen wird. Soll keine *Nullpunktunterdrückung* durchgeführt werden, muss der ermittelte Wert des *Sliders* mittels der Methode `measuredValFloatToHexConverter()` von einer Gleitkommazahl zu einer hexadezimalen Zahl umgewandelt werden. Im parametrierten Intervall wird entsprechend ein Messwerttelegramm erzeugt und mit der COT *Per/Cyc* übertragen.

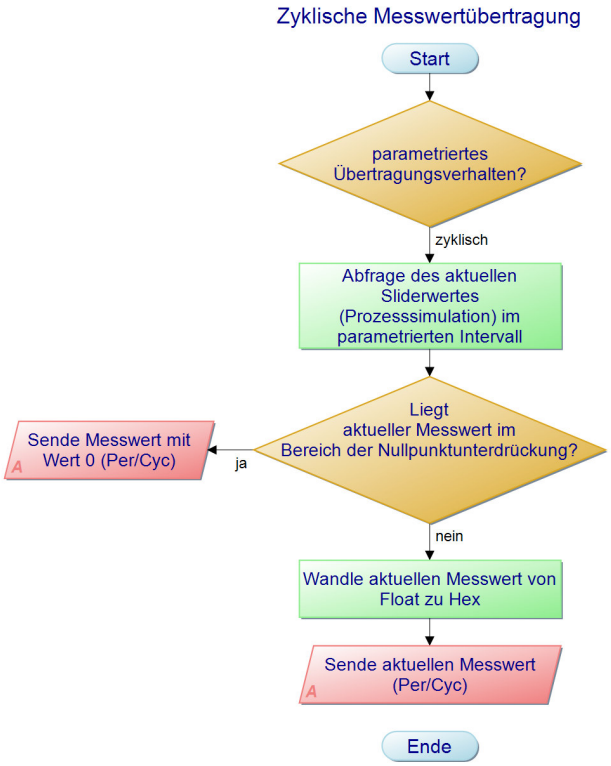


Abbildung 3.18: Programmablaufplan: Zyklische Messwertübertragung

4 Ergebnisdarstellung

Innerhalb dieses Kapitels wird die fertigstellte Anwendung als IEC 60870-5-104 -Server mit ihren Funktionalitäten vorgestellt. Dabei wird auf Basis einer Client/Server-Kommunikation über das Fernwirkprotokoll IEC 60870-5-104 zu einem das Netzleitsystem *SPRECON-V460* der Firma *Sprecher Automation GmbH*[1] die Verifizierung der Funktionalitäten vorgestellt. Weiterhin findet die *Client/Server Simulations- und Testumgebung für eine frei parametrierbare IEC 60870-5-104 Kommunikation* aus [4] Anwendung. Dabei wird mit dem Netzwerkanalyse-Tool *Wireshark*[3] der Telegrammverkehr mitgeschnitten und die aufbereitete Analyse dargestellt. Die Kompatibilitätsliste für diesen IEC 60870-5-104 -Server ist im Anhang A aufgeführt, in der die von der entwickelten Simulationsumgebung unterstützten Parameter angegeben sind.

Abbildung 4.1 zeigt die Simulationsumgebung als IEC 60870-5-104 -Server. Dabei besteht diese aus einem *FWG*, einer *Prozesssimulation* sowie einer *Vor-Ort-Steuerung*. Nach Start der Anwendung besteht die Möglichkeit eine Parametrierung gemäß IEC 60870-5-104 vorzunehmen. Dies ist für diese Anwendung nicht zwingend notwendig, das bereits eine vorab parametrierte Rangiertabelle zur Verfügung gestellt wird und die Simulationsumgebung ohne Parametrierung angewandt werden kann. Mit der in Abbildung 4.1 geöffneten und links angeordneten *Vor-Ort-Steuerung* kann neben der Durchführung einer Befehlsgabe außerdem die Schalthoheit eingestellt werden. Initial ist diese auf *Ort* gesetzt. Hinsichtlich der Meldungszustände der Schaltgeräte der Prozesssimulation befinden sich alle im Zustand *Aus*. Der initiale am *Slider* der Messwertsimulation Messwert weist einen Prozesswert von *325A* auf. Die für die Simulationsfälle unten links positionieren *CheckBoxen* sind nach Start der Anwendung nicht selektiert. Im Ausgabefenster unten rechst ist der Status der Schalthoheit und des Kommunikationsmodus im Ausgangszustand aufgeführt.

[3] `https://www.wireshark.org/`[??]

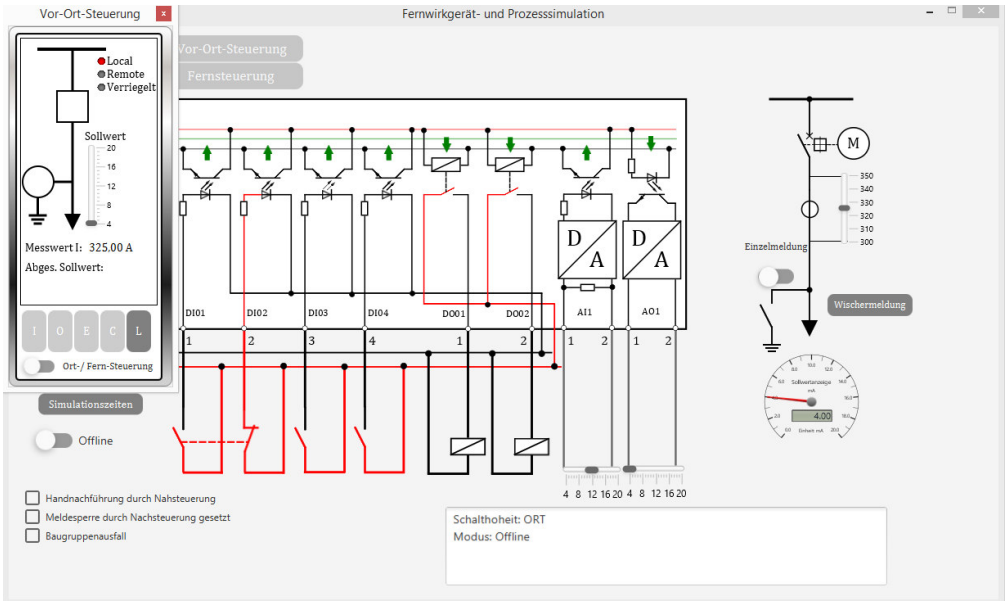


Abbildung 4.1: Oberfläche des Simulators mit Vor-Ort-Steuerung

Durch Selektion des *ToggleButtons* vom *Offline*-Modus in den *Online*-Modus, steht die Simulationsumgebung für eine Kommunikationsverbindung zu einem IEC 60870-5-104 -Client bereit. Wird eine Anfrage zur Herstellung einer Verbindung empfangen, ist der Simulator als IEC 60870-5-104 -Server bereit sowohl IEC 60870-5-104 -Telegramme in Steuerungsrichtung zu empfangen und zu interpretieren als auch Telegramme in Meldeerichtung zu generieren und übertragen.

4.1 Verifizierung mit Netzleitsystem SPRECON-V460

Für die Verifizierung der Funktionalität mit dem Netzleitsystem *SPRECON-V460* der Firma *Sprecher Automation GmbH*[1] ist eine entsprechende Parametrierung eines Prozessabbildes erforderlich. Nach Anlegen aller IEC 60870-5-104 Informationsobjekte, welche in der Rangiertabelle der Parametrierung des FWG des Simulators enthalten sind, können diese jeweils mit ausgewählten Symbolen zur visuellen Darstellung des Prozessabbildes verknüpft werden. Jede einzelne IEC 60870-5-104 -Variable, die angelegt wurde, kann hinsichtlich ihres Status konfiguriert werden. Abbildung 4.2 zeigt das Konfigurationsfenster der IEC 60870-5-104 -Variable für die Einzelmeldung des Erdungsschalters. Dabei sind im oberen linken Bereich Zustandsdefinitionen aufgeführt, für dessen Aus-

weitung eine Anzeigemöglichkeit festgelegt werden kann. In den rot umrandeten Zeilen ist für die entsprechende Einzelmeldung ausschließlich die Auswertung des Zustandsbits SPI festgelegt. Dabei erfolgt auf Basis der Auswertung des Zustandes *Ein* oder *Aus* eine jeweilige Darstellungsform des mit dieser Variable verknüpften Symbols. Die in grün markierten Zeilen werten neben dem Zustand zusätzlich das SB-Bit des Informationsobjektes der Einzelmeldung aus. Das Qualitätsbits BL wird mittels der in blau umrahmten Zeilen ausgewertet. Die in magenta umrandeten Zeilen dienen der Analyse der Qualitätsbits IV sowie NT.

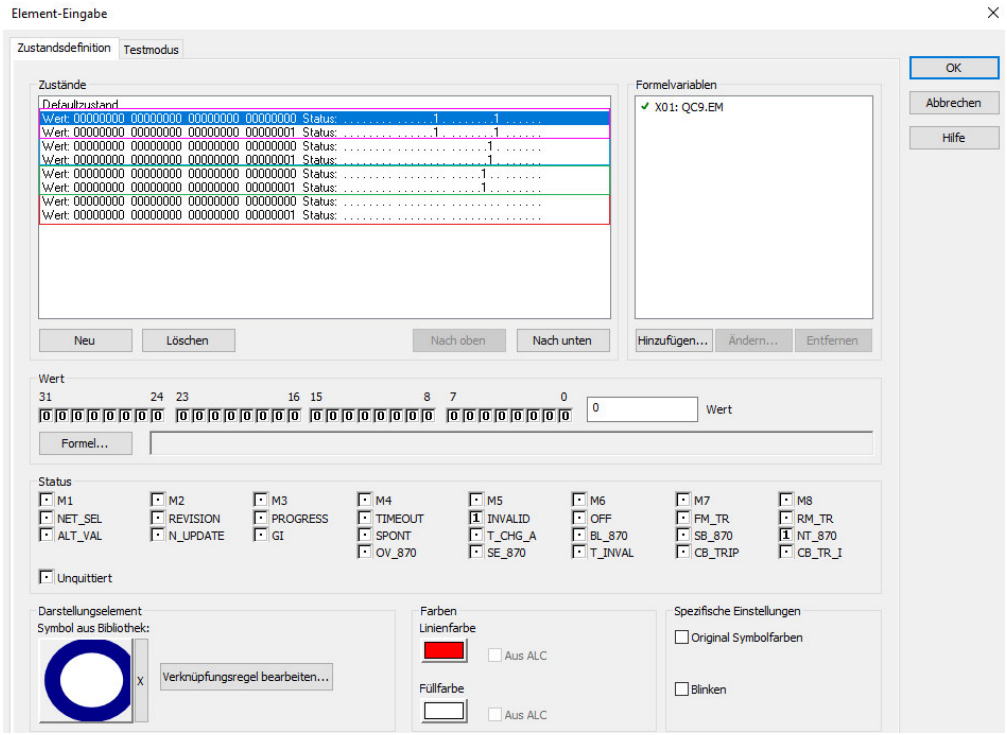


Abbildung 4.2: Variablenkonfiguration im SPRECON-V460

In Abbildung 4.3 ist ein vergrößerter Ausschnitt des Konfigurationsfensters abgebildet, welche für die Definition der auszuwertenden Werte und Qualitätsbits verwendet wird. Über den *Wert* kann dort Wert für den auszuwertenden Zustand der Variable festgelegt werden. Im Bereich *Status* können diverse Qualitätsbits oder interne Statusinformationen abgefragt werden. Das in Abbildung 4.3 mit 1 gesetzte Statusbit *BL_870* ist das in der Norm IEC 60870-5-104 definierte Qualitätsbit BL. Durch Setzen dieses Elementes

auf 1 wird das eingelesene IEC 60870-5-104 -Telegramm für dieses Informationsobjekt an genau diesem Qualitätsbit auf den Wert 1 geprüft. Die im Laufe der Simulationsfälle der Simulationsumgebung möglich setzbaren Qualitätsbits, dessen Auswertung im Netzleitsystem *SPRECON-V460* zu konfigurieren sind, sind in der Auflistung im Handbuch für Statusverarbeitung [7] definiert. Das bedeutet das Qualitätsbit SB kann mit dem Statusbit *SB_870*, NT mit dem Statusbit *NT_870* und das Qualitätsbits IV mit dem Statusbits *INVALID* ausgewertet werden.

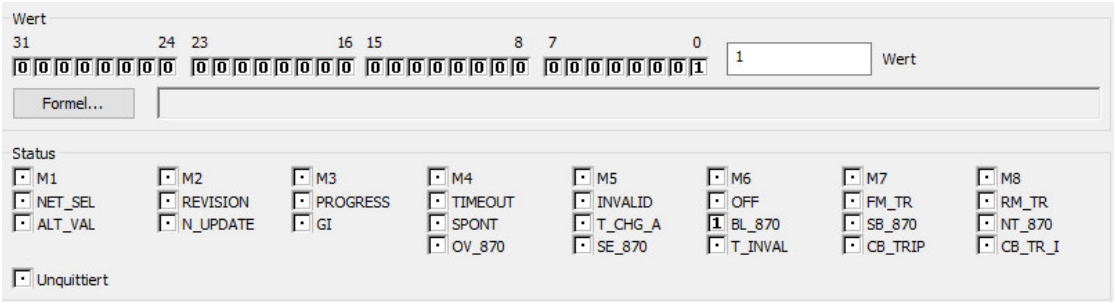


Abbildung 4.3: Konfiguration von Wert und Statusbits im SPRECON-V460

Nach dem für jede Variable die Auswertung der Statusbits und die entsprechende Darstellungsform parametriert wurde, kann eine Kommunikationsverbindung vom Netzleitsystem als IEC 60870-5-104 -Client zur Simulationsumgebung als IEC 60870-5-104 -Server aufgebaut werden. Abbildung 4.4 zeigt das Abbild des Schaltfeldes der Prozesssimulation bestehend aus Leistungsschalter für die Befehlsgabe mit Rückmeldung, Erdungsschalter, einem Messwert sowie einem Sollwertgeber für den Sollwert-Stellbefehl. In Abbildung 4.4 ist der Simulationsfall *Baugruppenausfall* dargestellt. Der über der Sammelschiene rot eingefärbte Bereich ist für die Zustandsdarstellung der Systemmeldung *Systeminfo Baugruppenausfall EM* der Rangiertabelle zuständig. Nachdem in der Simulationsumgebung die *CheckBox Baugruppenausfall* selektiert ist, werden alle Informationsobjekte mit gesetztem IV- und NT-Bit an das Netzleitsystem übertragen. Nach Auswertung gemäß der im Konfigurationsfenster eingestellten Darstellungsformen wird für ein eingeschaltetes Schaltgeräte die Symboleinfärbung in rot durchgeführt. Unabhängig von der Statusauswertung werden für alle abgebildeten Informationsobjekte mit gesetztem Qualitätsbit IV und NT in Abbildung 4.4 mit einem roten Quadrat an der Ecke des Elementes versehen. Daran kann erkannt werden, dass diese Elemente nicht den aktuellen Prozesswert enthalten.

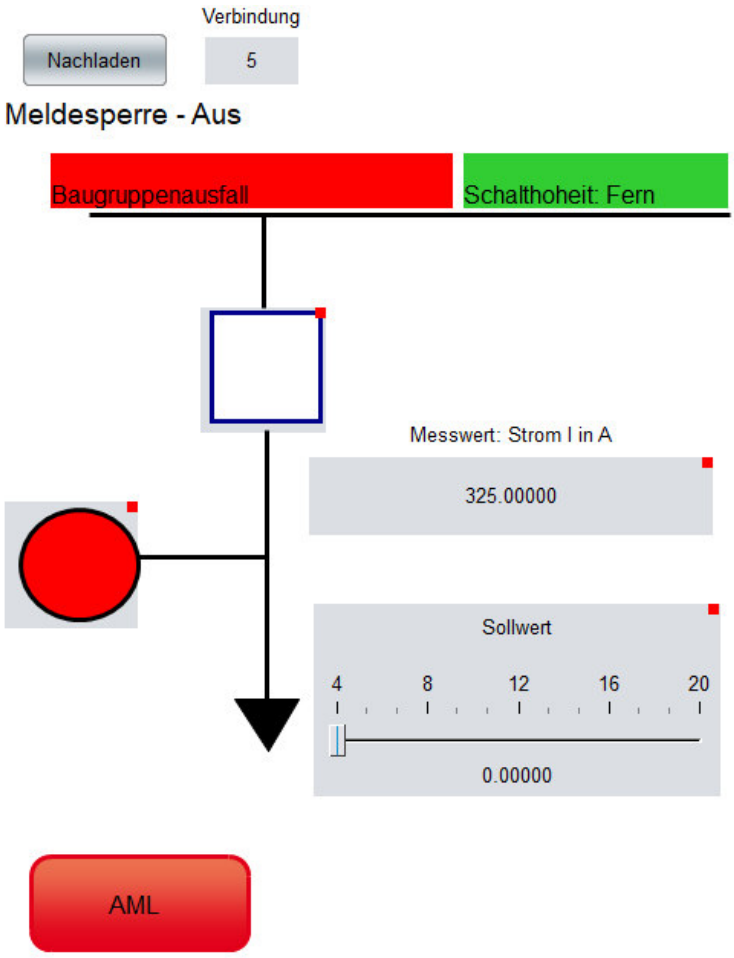


Abbildung 4.4: Simulationsfall: Baugruppenausfall

Im Falle des Simulationsfalles *Meldesperre durch Nahsteuerung* wird um das entsprechende Schaltfeld im Netzleitsystem aus Abbildung 4.5 eine Rahmenfläche in blinkender Darstellungsform angezeigt. Mit dem Meldetext wird der entsprechende Zustand nochmal angezeigt, jedoch ist mit der blinkenden Darstellungsform deutlich erkennbar, dass beispielsweise Wartungsarbeiten an dem Schaltfeld durchgeführt werden. Für Schaltgeräte im Zustand ein erfolgt zudem die Einfärbung in rot.

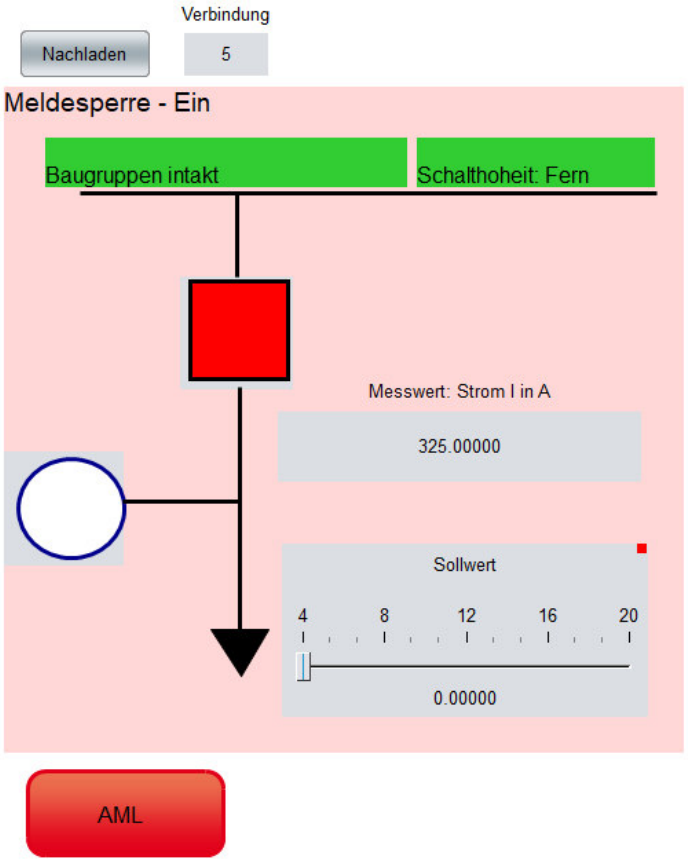


Abbildung 4.5: Simulationsfall: Meldesperre durch Nahsteuerung

Hinsichtlich des parametrierten Messwertes sind zwei Grenzwerte im Netzleitsystem angegeben. Bei Unterschreiten einer unteren Prozessgrenze von *310* sowie bei Überschreiten einer oberen Prozessgrenze von *340* wird der entsprechende Messwert mit einem Meldetext in eine *Alarmmeldeliste* eingetragen. In Abbildung 4.6 kann mit der Schaltfläche *AML* zu dieser Liste gelangt werden. Bei Identifizierung einer Prozessgrenzenverletzung erfolgt die Anzeige dieses Alarm rot hinterlegt in der oberen Leiste. Diese enthält den Zeitstempel, den Variablennamen, den entsprechenden Wert sowie den Meldetext. In der Regel müssen Alarme quittiert werden, sodass durch ein Doppelklick auf die rote Zeile die *Alarmmeldelist* geöffnet wird und die Quittierung zur Kenntnisnahme des Alarmes durchgeführt werden kann.

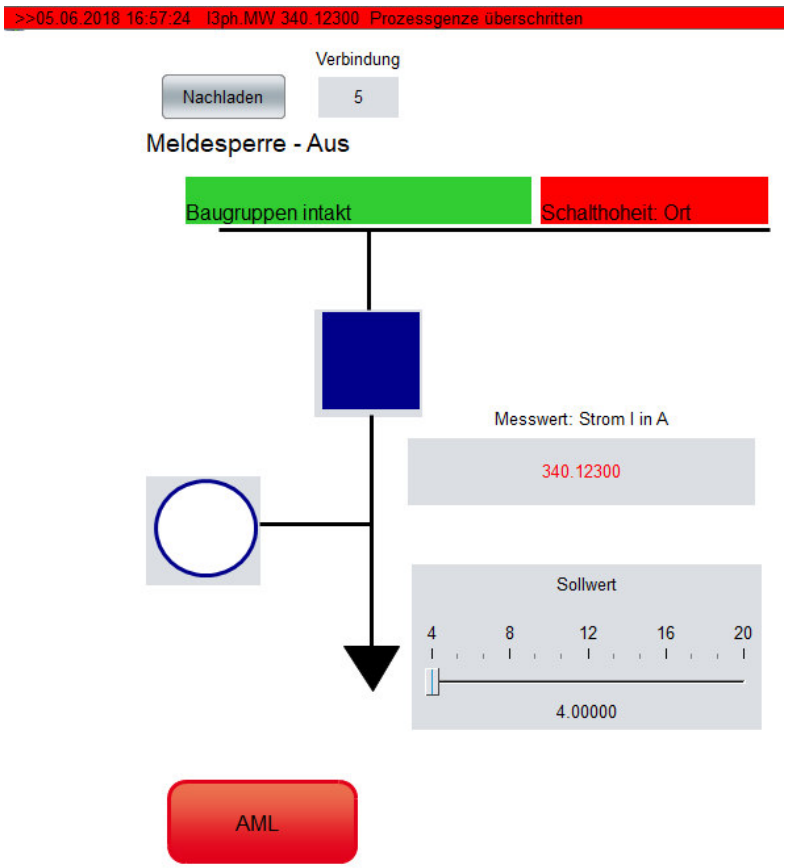
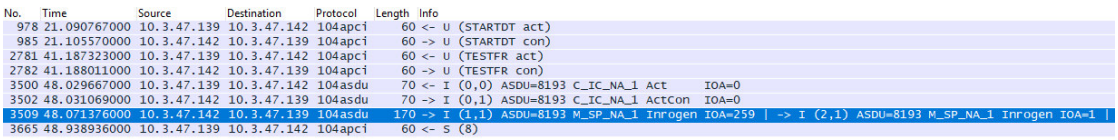


Abbildung 4.6: Prozessgrenzenverletzung

4.2 Verifizierung mit Client/Server-Simulationsumgebung

Bei der Verwendung der Client/Server-Simulationsumgebung [4] erfolgt der Einsatz der Anwendung als IEC 60870-5-104 -Client. Mit diesem werden Telegramme in Steuerungsrichtung generiert und an die Simulationsumgebung dieser Arbeit als IEC 60870-5-104 -Server übertragen. Vom IEC 60870-5-104 -Server empfangene Telegramme in Melderichtung können entsprechend analysiert und können als interpretierte Telegramme ausgegeben werden. Nachfolgend wird mit Hilfe von Protokollmitschnitten des Telegrammverkehrs zwischen Client und Server anhand einer Bit-genauen Untersuchung die Funktionsweise der Simulationsumgebung als IEC 60870-5-104 -Server aufgezeigt.

Unter Verwendung der bereits vorgegebenen und parametrierten Rangiertabelle kann eine Kommunikationsverbindung zwischen Server und Client hergestellt werden. Am Protokollmitschnitt in Abbildung 4.7 soll dabei zunächst die Übertragungssteuerung sowie die Generalabfrage dargestellt werden.



No.	Time	Source	Destination	Protocol	Length	Info
978	21.090767000	10.3.47.139	10.3.47.142	104apci	60	<- U (STARTDT act)
985	21.105570000	10.3.47.142	10.3.47.139	104apci	60	-> U (STARTDT con)
2781	41.187323000	10.3.47.139	10.3.47.142	104apci	60	<- U (TESTFR act)
2782	41.188011000	10.3.47.142	10.3.47.139	104apci	60	-> U (TESTFR con)
3500	48.029667000	10.3.47.139	10.3.47.142	104asdu	70	<- I (0,0) ASDU=8193 C_IC_NA_1 Act IOA=0
3502	48.031069000	10.3.47.142	10.3.47.139	104asdu	70	-> I (0,1) ASDU=8193 C_IC_NA_1 ActCon IOA=0
3509	48.071376000	10.3.47.142	10.3.47.139	104asdu	170	-> I (1,1) ASDU=8193 M_SP_NA_1 Inrogen IOA=259 | -> I (2,1) ASDU=8193 M_SP_NA_1 Inrogen IOA=1 |
3665	48.938936000	10.3.47.139	10.3.47.142	104apci	60	<- S (8)

Abbildung 4.7: Übertragungssteuerung und Generalabfrage

In den ersten beiden Zeilen des Mitschnitts wird nach erfolgreicher Stationsinitialisierung die Aktivierung des Datenaustausches durch den Client durchgeführt. Die Simulationsumgebung antwortet entsprechend mit einer Bestätigung. Die in den beiden Zeilen darauffolgende Prüfprozedur dient zur Prüfung der Verbindung. Dabei kann die Simulationsumgebung im Falle einer Aktivierung eines Prüftelegramms eine entsprechende Bestätigung rückmelden. Mit einem vom IEC 60870-5-104 -Client abgesetzten *Generalabfragebefehl* reagiert die IEC 60870-5-104 -Server Simulationsumgebung zunächst mit einem gespiegelten Telegramm und der COT *ActCon* als Bestätigung der Aktivierung. Daraufhin folgt die Übertragung aller Informationsobjekte, welche in Abbildung 4.8 insgesamt aufgeführt sind. Dabei enthalten die einzelnen Telegramme die in der Rangiertabelle parametrierte IOA und den jeweils aktuellen Prozesswert. Abschlossen wird die Generalabfrage mit einem gespiegelten Telegramm in Steuerungsrichtung mit der COT *ActTerm* als Beendigung der Aktivierung. Die Letzte Zeile aus Abbildung 4.7 zeigt den Empfang eines S-Telegramms mit der Empfangsfolgenummer des Clients als Quittierung. Das empfangene S-Telegramm wurde aufgrund der erreichten Anzahl von acht I-Telegramme vom Client als späteste Quittierung versendet.

```
⊞ IEC 60870-5-104-Asdu: ASDU=8193 M_SP_NA_1 Inrogen IOA=259 'single-point information'
⊞ IEC 60870-5-104-Apci: -> I (2,1)
⊞ IEC 60870-5-104-Asdu: ASDU=8193 M_SP_NA_1 Inrogen IOA=1 'single-point information'
⊞ IEC 60870-5-104-Apci: -> I (3,1)
⊞ IEC 60870-5-104-Asdu: ASDU=8193 M_SP_NA_1 Inrogen IOA=3 'single-point information'
⊞ IEC 60870-5-104-Apci: -> I (4,1)
⊞ IEC 60870-5-104-Asdu: ASDU=8193 M_SP_NA_1 Inrogen IOA=4 'single-point information'
⊞ IEC 60870-5-104-Apci: -> I (5,1)
⊞ IEC 60870-5-104-Asdu: ASDU=8193 M_DP_NA_1 Inrogen IOA=257 'double-point information'
⊞ IEC 60870-5-104-Apci: -> I (6,1)
⊞ IEC 60870-5-104-Asdu: ASDU=8193 M_ME_NC_1 Inrogen IOA=769 'measured value, short floating point number'
⊞ IEC 60870-5-104-Apci: -> I (7,1)
⊞ IEC 60870-5-104-Asdu: ASDU=8193 C_IC_NA_1 ActTerm IOA=0 'interrogation command'
```

Abbildung 4.8: Generalabfrage: Informationsobjekte

Der Protokollmitschnitt aus Abbild 4.9 zeigt den Telegrammverkehr zwischen Server und Client im Falle einer erfolgreichen Befehlsgabe durch den Client. Hierfür ist zunächst in der *Vor-Ort-Steuerung* der Server-Simulationsumgebung die Schalthoheit auf *Fern* umzustellen. Das in der Rangiertabelle parametrierte Informationsobjekt der TK=30 und der IOA=1 dient als Systemmeldung und wird beim Umschalten der Schalthoheit übertragen. Das entsprechende Telegramm für die Umschaltung der Schalthoheit ist im Protokollmitschnitt in der ersten Zeile aufgezeigt. Daraufhin besitzt der Client die Berechtigung Steuerungsvorgänge zu aktivieren. Mit der Aktivierung eines Doppelbefehls mit Zeitmarke kann das in der Rangiertabelle parametrierte Informationsobjekte mit der IOA 513 angesprochen werden.

No.	Time	Source	Destination	Protocol	Length	Info
364	6.967093000	10.3.47.142	10.3.47.139	104asdu	77	-> I (0,0) ASDU=8193 M_SP_TB_1 Spont IOA=1
834	11.900117000	10.3.47.139	10.3.47.142	104asdu	77	<- I (0,1) ASDU=8193 C_DC_TA_1 Act IOA=513
838	11.901312000	10.3.47.142	10.3.47.139	104asdu	77	-> I (1,1) ASDU=8193 C_DC_TA_1 ActCon IOA=513
1043	13.910557000	10.3.47.142	10.3.47.139	104asdu	77	-> I (2,1) ASDU=8193 M_DP_TB_1 Retrem IOA=257
1440	18.907924000	10.3.47.142	10.3.47.139	104asdu	77	-> I (3,1) ASDU=8193 M_DP_TB_1 Retrem IOA=257
1448	18.949057000	10.3.47.142	10.3.47.139	104asdu	77	-> I (4,1) ASDU=8193 C_DC_TA_1 ActTerm IOA=513

Abbildung 4.9: Befehlsgabe

Der IEC 60870-5-104 -Server antwortet mit einem gespiegelten Telegramm in Steuerungsrichtung die Aktivierung mit einer Bestätigung *ActCon*. Vorausgesetzt die Zwischenstellungsübertragung ist in der Parametrierung aktiviert, erfolgt die erste Rückmeldung für die Differenzstellung zwei Sekunden nach Bestätigung der Aktivierung, da die Simulationszeit für die Schalterbewegung des Auslösekontaktes der digitalen Ausgabe in den Simulationszeiten mit zwei Sekunden eingestellt ist. Das Bedeutet, dass erst nach dieser Simulationszeit der Auslösekontakt geschlossen ist und die Schaltspannung am digitalen Ausgang anliegt. Die Rückmeldung des Endzustandes des Leistungsschalters wird nach der eingestellten Simulationszeit des Leistungsschalters

im Fenster *Simulationszeiten*. Nach den festgelegten fünf Sekunden wird gemäß des Protokollmitschnitts aus Abbildung 4.9 die Doppelmeldung des Informationsobjektes der IOA 257 mit dem Endzustand und der COT *Retrem* übertragen. Im Anschluss wird die Befehlsaktivierung mit dem in der letzten Zeile aufgeführten gespiegelten Telegramm beendet (*ActTerm*).

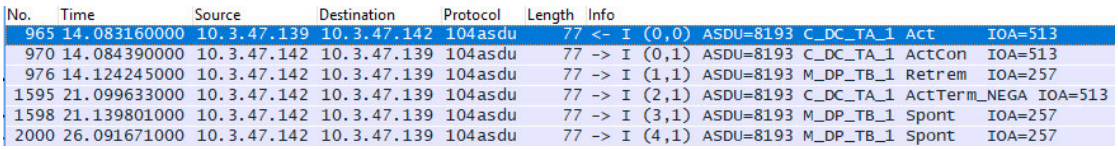
Abbildung 4.10 zeigt in der ersten Zeile eine Rückmeldung des Informationsobjektes der IOA 257, welche durch einen örtlichen Befehl verursacht wurde (*Retloc*). Das bedeutet, dass durch die *Vor-Ort-Steuerung* der Leistungsschalter angesteuert wurde und ein Zustandswechsel stattgefunden hat. Mit einer bestehenden Schalthoheit *Ort* wird durch den Client in der zweiten Zeile eine Befehlsaktivierung an de Leistungsschalter durchgeführt. Der IEC 60870-5-104 -Server antwortet mit einer negativen Befehlsaktivierung mit der COT *ActCon_NEGA*, da eine Verletzung der Schalthoheit besteht. In der vierten Zeile wird durch die *Vor-Ort-Steuerung* die Schalthoheit auf *Fern* umgeschaltet, sodass der Client wieder über eine Schaltberechtigung verfügt. Der Leistungsschalter wurde durch die *Vor-Ort-Steuerung* in den Zustand Ein geschaltet. Setzt der Client, wie in Zeile fünf des Protokollmitschnitts aus Abbildung 4.10 dargestellt, einen Doppelbefehl mit der Schaltrichtung, dessen Zustand bereits der Leistungsschalter aufweist, muss die Aktivierung negativ bestätigt werden (*ActCon_NEGA*).

No.	Time	Source	Destination	Protocol	Length	Info
852	10.707366000	10.3.47.142	10.3.47.139	104asdu	77	-> I (0,0) ASDU=8193 M_DP_TB_1 Retloc IOA=257
1653	19.487411000	10.3.47.139	10.3.47.142	104asdu	77	<- I (0,1) ASDU=8193 C_DC_TA_1 Act IOA=513
1654	19.488449000	10.3.47.142	10.3.47.139	104asdu	77	-> I (1,1) ASDU=8193 C_DC_TA_1 ActCon_NEGA IOA=513
2088	24.397439000	10.3.47.142	10.3.47.139	104asdu	77	-> I (2,1) ASDU=8193 M_SP_TB_1 Spont IOA=1
2749	31.271528000	10.3.47.139	10.3.47.142	104asdu	77	<- I (1,3) ASDU=8193 C_DC_TA_1 Act IOA=513
2750	31.271993000	10.3.47.142	10.3.47.139	104asdu	77	-> I (3,2) ASDU=8193 C_DC_TA_1 ActCon_NEGA IOA=513

Abbildung 4.10: *Vor-Ort-Steuerung*, Ortverriegelung, EIN->EIN-Befehl

Für den in Abbildung 4.11 hervorzurufenden Telegrammverkehr zwischen dem IEC 60870-5-104 -Client und der Simulationsumgebung als IEC 60870-5-104 -Server, wird innerhalb der Simulationszeiten die Laufzeit des Leistungsschalters von 5 auf 10 Sekunden erhöht. Die ersten drei Telegramm des Protokollmitschnitts zeigen einen üblichen Telegrammverkehr bei einer Befehlsaktivierung. Es wird durch den IEC 60870-5-104 -Client zunächst eine Befehlsaktivierung *Act* für den Leistungsschalter an den Server übertragen. Der Server bestätigt die Aktivierung mit einem gespiegelten Telegramm und überträgt, sobald die Befehlsgabe durch die digitale Ausgabe gestartet wird, die Zwischenstellung als Rückmeldung an den Client. Da durch die parametrierte Gerätelaufzeit

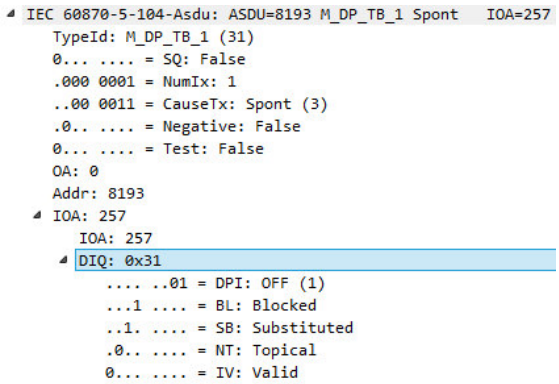
der Rückmeldung nach 5 Sekunden eine Rückmeldung erwartet wird, der Leistungsschalter aber eine Gesamtlaufzeit von 10 Sekunden besitzt, muss die Befehlsbeendigung negativ beendet werden (*ActTerm_ NEGA*). Die daraufhin vom FWG erfassten Zustände werden entsprechend mit der COT *Spont* vom Server übertragen. Das letzte in Abbildung 4.11 übertragene Telegramm enthält den Endzustand des Leistungsschalters und wird nach Ablauf seiner festgelegten Gesamtlaufzeit übertragen.



No.	Time	Source	Destination	Protocol	Length	Info
965	14.083160000	10.3.47.139	10.3.47.142	104asdu	77	<- I (0,0) ASDU=8193 C_DC_TA_1 Act IOA=513
970	14.084390000	10.3.47.142	10.3.47.139	104asdu	77	-> I (0,1) ASDU=8193 C_DC_TA_1 ActCon IOA=513
976	14.124245000	10.3.47.142	10.3.47.139	104asdu	77	-> I (1,1) ASDU=8193 M_DP_TB_1 Retrem IOA=257
1595	21.099633000	10.3.47.142	10.3.47.139	104asdu	77	-> I (2,1) ASDU=8193 C_DC_TA_1 ActTerm_NEGA IOA=513
1598	21.139801000	10.3.47.142	10.3.47.139	104asdu	77	-> I (3,1) ASDU=8193 M_DP_TB_1 Spont IOA=257
2000	26.091671000	10.3.47.142	10.3.47.139	104asdu	77	-> I (4,1) ASDU=8193 M_DP_TB_1 Spont IOA=257

Abbildung 4.11: Rückmeldung außerhalb der Gerätelaufzeit

Mit der detaillieren Analyse des Telegramms einer Einzelmeldung in Abbildung 4.12 sollen die Funktionalitäten der Simulationsfälle *Meldesperre durch Nahsteuerung* und *Handnachführung durch Nahsteuerung* aufgezeigt werden. Dabei werden bei Selektion eines Simulationsfalls alles Informationsobjekte mit aktualisierter Qualitätskennung an den Client übertragen. Bei dem in Abbildung 4.12 ausgewählten Informationsobjekt handelt es sich um die Doppelmeldung des Leistungsschalters. Die ab der blau markierten Zeile zusammengesetzt Qualitätskennung DIQ dieses Informationsobjektes weist einen Meldungszustand DPI *Aus* auf. Mit den Qualitätskennungen BL und SB kann identifiziert werden, dass die beiden genannten Simulationsfälle aktiviert sind.



```
▾ IEC 60870-5-104-Asdu: ASDU=8193 M_DP_TB_1 Spont   IOA=257
    TypeId: M_DP_TB_1 (31)
    0... .... = SQ: False
    .000 0001 = NumIx: 1
    ..00 0011 = CauseTx: Spont (3)
    .0.. .... = Negative: False
    0... .... = Test: False
    OA: 0
    Addr: 8193
  ▾ IOA: 257
      IOA: 257
    ▾ DIQ: 0x31
          .... ..01 = DPI: OFF (1)
          ...1 .... = BL: Blocked
          ..1. .... = SB: Substituted
          .0.. .... = NT: Topical
          0... .... = IV: Valid
```

Abbildung 4.12: Meldesperre und Handnachführung durch Nahsteuerung

Anhand des in Abbildung 4.13 dargestellten I-Telegramms der TK=30 soll die Realisierung der *Wischermeldung* vorgestellt werden. Dabei ist bereits in der blau markierten Titel-Zeile der Beschreibung *IOA[2]=260* zu entnehmen, dass sich dieses Telegramm aus zwei Informationsobjekten mit identischen IOA besteht. Anhand der Qualitätskennung SIQ der einzelnen Informationsobjekte ist ein Wechsel des Meldungszustands SPI von *Ein* nach *Aus* zu erkennen. Dabei wird durch den Vergleich der Zeitmarken beider Informationsobjekte die *Wischermeldung* deutlich.

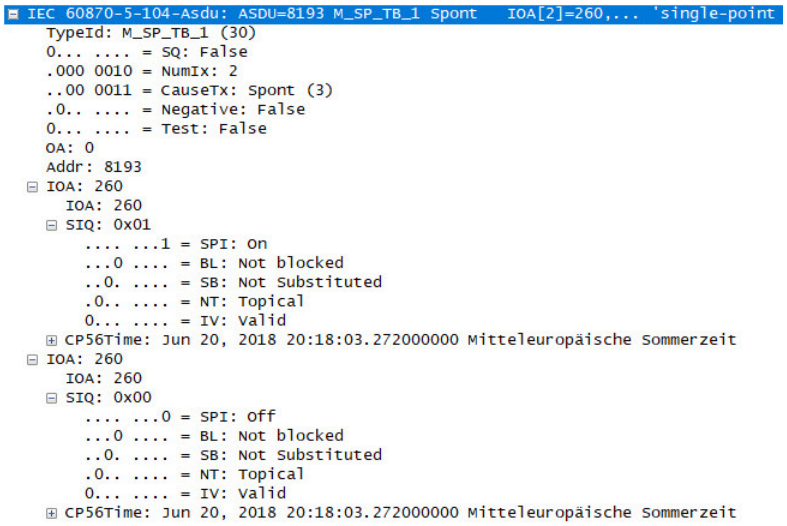


Abbildung 4.13: Wischermeldung

5 Zusammenfassung und Ausblick

In der Masterthesis *Entwicklung und Validierung einer Simulationsumgebung mit fernwirk- und stationsleittechnischen Funktionen und IEC 60870-5-104 Kommunikation unter Java* erfolgt die Entwicklung einer Anwendung zur Client-/Server-Kommunikation gemäß dem Fernwirkprotokoll IEC 60870-5-104. Die Anwendung besteht aus den Komponenten Prozesssimulation, FWG mit einer Parametriermöglichkeit sowie einer Vor-Ort-Steuerung.

Der statische Prozess der Prozesssimulation als Abbild eines Schaltfeldes bestehend aus einem Leistungsschalter, Erdungsschalter, Messwandler, einer Sollwertanzeige sowie einer Wischermeldung kann gemäß dem Fernwirkprotokoll IEC 60870-5-104 parametriert werden. Die erforderlichen Informationsobjekte liegen in Form einer Rangiertabelle vor. Jedes einzelne Informationsobjekt weist eine Typkennung mit Zeitmarke auf. Als Globale Adresse kann die Stationsadresse CAASDU für alle Informationsobjekte definiert werden. Neben der individuellen Vergabe der IOA kann im Rhamen der Parametrierung für bestimmte Informationsobjekte das Übertragungsverhalten eingestellt werden. Dabei kann für die Einzelmeldungen eine Flatterunterdrückung und für die Doppelmeldung eine Zwischenstellungsunterdrückung festgelegt werden. Es kann eine Auswahl zwischen zyklischer oder spontaner Messwertübertragung durchgeführt werden. Bei einer ereignisorientierten Übertragung besteht die Möglichkeit der Parametrierung von Schwellen. Zusätzlich kann mittles der Parametrierung einer oberen und unteren Prozessgrenze eine Grenzwertverletzung signalisiert werden, indem für das Messwert-Telegramm entsprechend das OV-Bit (Messwertüberlauf) gesetzt ist.

Durch die Umsetzung der Simulationsumgebung als IEC60870-5-104 Server ist die Anwendung in der Lage ein Doppelbefehl sowie Sollwert-Stellbefehl als Client-Telegramm zu empfangen, zu analysieren und entsprechen die Simulation des Vorgangs zu beginnen. Dabei werden alle für eine erfolgreiche oder nicht mögliche Befehlsprozedur erforderlichen

Telegramme vom IEC60870-5-104 Server generiert und an die Gegenstelle übertragen. Bei einer nicht durch einen Steuervorgangs ausgelösten Prozessänderung werden die Änderungen entsprechend der Ursache als Telegramme an den IEC60870-5-104 Client versendet.

Eine Validierung der Funktionalität der Simulationsumgebung wurde mittels des Netzleitsystems SPRECON-V460 der Firma *Sprecher Automation GmbH*[4] durchgeführt. Dabei konnten alle möglichen Qualitätsbits der Qualitätskennungen der Informationsobjekte ausgewertet werden. Zusätzlich wurde die Client/Server Simulations- und Testumgebung der Arbeit [4] herangezogen.

Mit der Simulationsumgebung können mittels einer Vor-Ort-Steuerung oder über eine externe Fernsteuerung Schalthandlungen oder Sollwert-Änderungen vorgenommen werden. Für die Simulationsumgebung wurde keine freie Parametrierung der Herkunftsadresse realisiert. In diesem Zusammenhang kann eine Lösung für eine Kommunikationsaufbau zu mehreren Clients integriert werden. In Rahmen dieser Erweiterung besteht die Möglichkeit eines Anwendungsfalls der Parametrierung einer Herkunftsadresse, mit der Schaltberechtigungen an den verbundenen IEC 60870-5-104 -Clients vergeben werden können. Zudem ist im Rahmen dieses Projektes die Analyse von Telegrammen in Steuerungsrichtung ausschließlich für TK mit Zeitmarke realisiert. Für die Integration von TK in Steuerungsrichtung ohne Zeitmarke gilt es zunächst diese Einstellung in der Parametrierung durchführen zu können sowie einer Erweiterung der implementierten Methode um die entsprechende TK. Zur Verdeutlichung des Simulationsfalls *Handnachführung durch Nahsteuerung* ist die Möglichkeit der Vorgabe eines Ersatzwertes innerhalb der spezifischen Parametrierung der Informationsobjekte. Somit kann nach Aktivierung des Simulationsfalls eine Änderung des Zustandes durch die Nahsteuerung beobachtet werden. Damit die im Rahmen der Parametrierung durchgeführten IEC 60870-5-104 Parameter bei erneutem Start der Simulationsumgebung nicht neu eingegeben werden müssen, kann die Funktion der Sicherung der Rangiertabelle als Character Separated Value (CSV)-Datei eingebunden werden.

[4] `https://www.sprecher-automation.com`[1]

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[1] *Fernwirkeinrichtungen und -systeme – Teil 5-101: Übertragungsprotokolle – Anwendungsbezogene Norm für grundlegende Fernwirkaufgaben (IEC 60870-5-101:2003).* Norm. DIN EN 60870-5-101, Dez. 2003.

[2] *Fernwirkeinrichtungen und -systeme – Teil 5-104: Übertragungsprotokolle – Zugriff für IEC 60870-5-101 auf Netze mit genormten Transportprofilen.* Norm. DIN EN 60870-5-104, Sep. 2007.

[3] Dardae Fariad. "Entwicklung einer grafischen Oberfläche zur Simulation eines Fernwirksystems unter JavaFX". Masterstudienarbeit. Fachhochschule Dortmund.

[4] Muhammed Safakli. "Entwicklung einer Client/Server Simulations- und Testumgebung für eine frei parametrierbare IEC 60870-5-104 Kommunikation unter Java". Masterthesis. Fachhochschule Dortmund.

[5] Muhammed Safkli. "Entwicklung einer Serverkommunikation auf Basis der Norm IEC 60870-5-104 unter Java". Masterstudienarbeit. Fachhochschule Dortmund.

[6] Dardae Fariad. *Darstellung und Dokumentation von Fernwirksystemen unterschiedlicher Hersteller.* Projektarbeit. Fachhochschule Dortmund - Labor für Energieautomation und Netzführung.

[7] Sprecher Automation GmbH. "SPRECON-V460 Handbuch Statusverarbeitung". Handbuch. Sprecher Automation GmbH, 2014.

A Kompatibilitätsliste

9 Kompatibilität

Diese anwendungsbezogene Norm gibt Parametersätze und Alternativen vor, aus denen Untermengen auszuwählen sind, um bestimmte Fernwirksysteme zu erstellen. Bestimmte Parameter, wie die Auswahl von „strukturierten“ oder „unstrukturierten“ Feldern der ADRESSE DES INFORMATIONSOBJEKTS von ASDU, schließen sich gegenseitig aus. Das bedeutet, dass nur ein Wert des festgelegten Parameters je System zulässig ist. Andere Parameter, wie der aufgelistete Satz unterschiedlicher Prozessinformation in Befehls- und Überwachungsrichtung, erlauben die Festlegung von Gesamt- oder Untermengen, die für die gegebene Anwendung geeignet sind. Dieser Abschnitt fasst die Parameter der vorstehenden Abschnitte zusammen, um eine geeignete Auswahl für eine bestimmte Anwendung zu ermöglichen. Wird ein System aus mehreren Systemkomponenten unterschiedlicher Hersteller zusammengesetzt, ist es erforderlich, dass alle Partner den ausgewählten Parametern zustimmen.

Die Kompatibilitätsliste ist wie in IEC 60870-5-101 festgelegt und um Parameter ergänzt, die in dieser Norm angewendet werden. Die zugehörigen Beschreibungen in dieser anwendungsbezogenen Norm nicht zulässiger Parameter sind durchgestrichen (das zugehörige Kontrollfeld ist geschwärzt).

ANMERKUNG Für die vollständige Festlegung eines Systems kann zusätzlich die individuelle Auswahl bestimmter Parameter für bestimmte Teile eines Systems erforderlich sein, z. B. die individuelle Auswahl von Skalierungsfaktoren für individuell adressierbare Messwerte.

Die ausgewählten Parameter sollten in den weißen Kontrollfeldern wie folgt markiert werden:

☐ Funktion oder ASDU wird nicht benutzt

☒ Funktion oder ASDU wird wie genormt benutzt (Vorzugswert)

☐ Funktion oder ASDU wird im Umkehrmodus benutzt

☐ Funktion oder ASDU wird im Regel- und Umkehrmodus benutzt

Die mögliche Auswahl (leer, X, R oder B) ist für jeden Abschnitt oder Parameter festgelegt.

Ein schwarzes Kontrollfeld zeigt an, dass die Auswahl in dieser anwendungsbezogenen Norm nicht durchgeführt werden kann.

9.1 System oder Gerät

(systembezogener Parameter, die Festlegung System oder Gerät ist durch Markieren eines der folgenden Kontrollfelder mit „**X**“ anzuzeigen)

☐ Systemfestlegung

☐ Festlegung für die Zentralstation

☒ Festlegung für die Unterstation

9.2 Netzkonfiguration

(netzbezogener Parameter, alle angewendeten Konfigurationen sind mit „**X**“ zu markieren)

■ ~~Point-to-point~~ ■ ~~Multipoint~~

■ ~~Multiple point-to-point~~ ■ ~~Multipoint star~~

9.3 Physikalische Schicht

(netzbezogener Parameter, alle angewendeten Schnittstellen und Datenraten sind mit „**X**“ zu markieren)

Übertragungsgeschwindigkeit (Steuerungsrichtung)

Unsymmetrische Schnittstelle V.24/V.28 Genormt		Unsymmetrische Schnittstelle V.24/V.28 Empfohlen bei > 1200 bit/s		Symmetrische Schnittstelle X.24/X.27			
☐	~~100 bit/s~~	☐	~~2400 bit/s~~	☐	~~2400 bit/s~~	☐	~~56 000 bit/s~~
☐	~~200 bit/s~~	☐	~~4800 bit/s~~	☐	~~4800 bit/s~~	☐	~~64 000 bit/s~~
☐	~~300 bit/s~~	☐	~~9600 bit/s~~	☐	~~9600 bit/s~~		
☐	~~600 bit/s~~			☐	~~19 200 bit/s~~		
☐	~~1200 bit/s~~			☐	~~38 400 bit/s~~		

Übertragungsgeschwindigkeit (Überwachungsrichtung)

Unsymmetrische Schnittstelle V.24/V.28 Genormt		Unsymmetrische Schnittstelle V.24/V.28 Empfohlen bei > 1200 bit/s		Symmetrische Schnittstelle X.24/X.27			
☐	~~100 bit/s~~	☐	~~2400 bit/s~~	☐	~~2400 bit/s~~	☐	~~56 000 bit/s~~
☐	~~200 bit/s~~	☐	~~4800 bit/s~~	☐	~~4800 bit/s~~	☐	~~64 000 bit/s~~
☐	~~300 bit/s~~	☐	~~9600 bit/s~~	☐	~~9600 bit/s~~		
☐	~~600 bit/s~~			☐	~~19 200 bit/s~~		
☐	~~1200 bit/s~~			☐	~~38 400 bit/s~~		

9.4 Verbindungsschicht

(Netzbezogener Parameter, alle angewendeten Auswahlen sind mit „**X**“ zu markieren. Die maximale Telegrammlänge ist festzulegen. Ist für unsymmetrische Übertragungsdienste eine von der Regel abweichende [en. non-standard] Zuweisung von Anwenderdaten zur Datenklasse 2 eingeführt, sind Typkennung und Übertragungsursache aller der Datenklasse 2 zugewiesenen Anwenderdaten anzugeben.)

~~Nach dieser anwendungsbezogenen Norm werden ausschließlich Telegrammformat FT 1.2, Einzelzeichen 1 und das feste Zeitüberwachungsintervall benutzt.~~

Übertragungsprozedur der Verbindungsschicht

- [ ] ~~Symmetrische Übertragung~~
- [ ] ~~Unsymmetrische Übertragung~~

Adressfeld der Verbindungsschicht

- [ ] ~~Nicht vorhanden (nur symmetrische Übertragung)~~
- [ ] ~~Ein Oktett~~
- [ ] ~~Zwei Oktette~~

Telegrammlänge

- [ ] ~~Strukturiert~~
- [ ] ~~Unstrukturiert~~
- [ ] ~~Maximale Länge L (Anzahl der Oktette)~~

Wird unsymmetrisch übertragen, werden die folgenden ASDU als Anwenderdaten mit den angegebenen Übertragungsursachen mit der Datenklasse 2 (niedrige Priorität) zurückübertragen:

- [ ] ~~Die genormte Zuweisung von ASDU zur Datenklasse 2 wird wie folgt angewendet:~~

Typkennung	Übertragungsursache
9, 11, 13, 21	<1>

- [ ] ~~Eine spezielle Zuweisung von ASDU zur Datenklasse 2 wird wie folgt angewendet:~~

Typkennung	Übertragungsursache

~~ANMERKUNG (Als Antwort auf eine Anforderung nach Daten der Klasse 2 darf eine Unterstation Daten der Datenklasse 1 übertragen, wenn keine Daten der Datenklasse 2 vorhanden sind.~~

9.5 Anwendungsschicht

Übertragungsmode für Anwendungsdaten

Nach dieser ~~anwendungsbezogenen~~ Norm wird ausschließlich Modus 1 (niedrigstwertiges Oktett zuerst) nach 4.10 von IEC 60870-5-4 benutzt.

Gemeinsame Adresse der ASDU

(systembezogener Parameter, alle angewendeten Konfigurationen sind mit „**X**“ zu markieren)

- [ ] ~~Ein Oktett~~
- [X] Zwei Oktette

Adresse des Informationsobjekts

(systembezogener Parameter, alle angewendeten Konfigurationen sind mit „**X**“ zu markieren)

- [ ] ~~Ein Oktett~~
- [ ] ~~Zwei Oktette~~
- [X] Drei Oktette
- [ ] Strukturiert
- [X] Unstrukturiert

Übertragungsursache

(systembezogener Parameter, alle angewendeten Konfigurationen sind mit „**X**“ zu markieren)

- [ ] Ein Oktett
- [X] Zwei Oktette (mit Herkunftsadresse). Mit 0 vorbesetzt, falls Herkunftsadresse nicht vorhanden

Länge der APDU

(systembezogener Parameter, die maximale Länge der APDU je System ist festzulegen)

Die maximale Länge der APDU beträgt in beiden Übertragungsrichtungen 253. Dies ist ein fester Systemparameter.

- [ ] Maximale Länge der APDU je System in Steuerungsrichtung
- [ ] Maximale Länge der APDU je System in Überwachungsrichtung

Auswahl genormter ASDU

Prozessinformation in Überwachungsrichtung

(stationsbezogener Parameter, jede nur in regulärer Richtung angewendete Typkennung ist mit „**X**“ zu markieren, mit „**R**“ falls nur in entgegengesetzter Richtung und mit „**B**“ falls in beiden Richtungen angewendet)

[X]	<1> := Einzelmeldung	M_SP_NA_1
[ ]	~~<2> := Einzelmeldung mit Zeitmarke~~	~~M_SP_TA_1~~
[X]	<3> := Doppelmeldung	M_DP_NA_1
[ ]	~~<4> := Doppelmeldung mit Zeitmarke~~	~~M_DP_TA_1~~
[ ]	<5> := Stufenstellungsmeldung	M_ST_NA_1
[ ]	~~<6> := Stufenstellungsmeldung mit Zeitmarke~~	~~M_ST_TA_1~~
[ ]	<7> := Bitmuster von 32 bit	M_BO_NA_1
[ ]	~~<8> := Bitmuster von 32 bit mit Zeitmarke~~	~~M_BO_TA_1~~
[ ]	<9> := Messwert, normierter Wert	M_ME_NA_1
[ ]	~~<10> := Messwert, normierter Wert mit Zeitmarke~~	~~M_ME_TA_1~~
[ ]	<11> := Messwert, skalierter Wert	M_ME_NB_1
[ ]	~~<12> := Messwert, skalierter Wert mit~~ Zeitmarke	~~M_ME_TB_1~~
[X]	<13> := Messwert, verkürzte Gleitkommazahl	M_ME_NC_1

■	~~<14> := Messwert, verkürzte Gleitkommazahl mit Zeitmarke~~	~~M_ME_TC_1~~
☐	<15> := Zählwerte	M_IT_NA_1
■	~~<16> := Zählwerte mit Zeitmarke~~	~~M_IT_TA_1~~
■	~~<17> := Schutzereignis mit Zeitmarke~~	~~M_EP_TA_1~~
■	~~<18> := Geblockte Anregungen des Schutzes mit Zeitmarke~~	~~M_EP_TB_1~~
■	~~<19> := Geblockte Auslösungen des Schutzes mit Zeitmarke~~	~~M_EP_TC_1~~
☐	<20> := Geblockte Einzelmeldungen mit Zustandsanzeige	M_PS_NA_1
☐	<21> := Messwert, normierter Wert ohne Qualitätskennung	M_ME_ND_1
☒	<30> := Einzelmeldung mit Zeitmarke CP56Time2a	M_SP_TB_1
☒	<31> := Doppelmeldung mit Zeitmarke CP56Time2a	M_DP_TB_1
☐	<32> := Stufenstellungsmeldung mit Zeitmarke CP56Time2a	M_ST_TB_1
☐	<33> := Bitmuster von 32 bit mit Zeitmarke CP56Time2a	M_BO_TB_1
☐	<34> := Messwert, normierter Wert mit Zeitmarke CP56Time2a	M_ME_TD_1
☐	<35> := Messwert, skalierter Wert mit Zeitmarke CP56Time2a	M_ME_TE_1
☒	<36> := Messwert, verkürzte Gleitkommazahl mit Zeitmarke CP56Time2a	M_ME_TF_1
☐	<37> := Zählwerte mit Zeitmarke CP56Time2a	M_IT_TB_1
☐	<38> := Schutzereignis mit Zeitmarke CP56Time2a	M_EP_TD_1
☐	<39> := Geblockte Anregungen des Schutzes mit Zeitmarke CP56Time2a	M_EP_TE_1
☐	<40> := Geblockte Auslösungen des Schutzes mit Zeitmarke CP56Time2a	M_EP_TF_1

In dieser anwendungsbezogenen Norm ist nur die Benutzung des ASDU-Satzes <30> bis <40> für ASDU mit Zeitmarke erlaubt.

Prozessinformation in Steuerungsrichtung

(stationsbezogener Parameter, jede nur in regulärer Richtung angewendete Typkennung ist mit „**X**“ zu markieren, mit „**R**“ falls nur in entgegengesetzter Richtung und mit „**B**“ falls in beiden Richtungen angewendet)

☐	<45> := Einzelbefehl	C_SC_NA_1
☐	<46> := Doppelbefehl	C_DC_NA_1
☐	<47> := Stufenstellbefehl	C_RC_NA_1
☐	<48> := Sollwert-Stellbefehl, normierter Wert	C_SE_NA_1
☐	<49> := Sollwert-Stellbefehl, skalierter Wert	C_SE_NB_1
☐	<50> := Sollwert-Stellbefehl, verkürzte Gleitkommazahl	C_SE_NC_1
☐	<51> := Bitmuster von 32 bit	C_BO_NA_1
☐	<58> := Einzelbefehl mit Zeitmarke CP56Time 2a	C_SC_TA_1

R	<59> :=	Doppelbefehl mit Zeitmarke CP56Time 2a	C_DC_TA_1
	<60> :=	Stufenstellbefehl mit Zeitmarke CP56Time 2a	C_RC_TA_1
	<61> :=	Sollwert-Stellbefehl mit Zeitmarke CP56Time 2a, normierter Wert	C_SE_TA_1
	<62> :=	Sollwert-Stellbefehl mit Zeitmarke CP56Time 2a, skalierter Wert	C_SE_TB_1
R	<63> :=	Sollwert-Stellbefehl mit Zeitmarke CP56Time 2a, verkürzte Gleitkommazahl	C_SE_TC_1
	<64> :=	Bitmuster von 32 bit mit Zeitmarke CP56Time 2a	C_BO_TA_1

Es wird entweder der ASDU-Satz <45> bis <51> oder der Satz <58> bis <64> angewendet.

Systeminformation in Überwachungsrichtung

(stationsbezogener Parameter, jede nur in regulärer Richtung angewendete Typkennung ist mit „**X**“ zu markieren, mit „**R**“ falls nur in entgegengesetzter Richtung und mit „**B**“ falls in beiden Richtungen angewendet)

	<70> :=	Initialisierungsende	M_EI_NA_1

Systeminformation in Steuerungsrichtung

(stationsbezogener Parameter, jede nur in regulärer Richtung angewendete Typkennung ist mit „**X**“ zu markieren, mit „**R**“ falls nur in entgegengesetzter Richtung und mit „**B**“ falls in beiden Richtungen angewendet)

R	<100>:=	(Stations-)Abfragebefehl	C_IC_NA_1
	<101>:=	Zähler-Abfragebefehl	C_CI_NA_1
	<102>:=	Abfragebefehl	C_RD_NA_1
	<103>:=	Uhrzeit-Synchronisationsbefehl (wahlweise, siehe 7.6)	C_CS_NA_1
	~~<104>:=~~	~~Prüfbefehl~~	~~C_TS_NA_1~~
	<105>:=	Prozess-Rücksetzbefehl	C_RP_NA_1
	~~<106>:=~~	~~Befehl zur Telegrammlaufzeit-Erfassung~~	~~C_CD_NA_1~~
	<107>:=	Prüfbefehl mit Zeitmarke CP56time2a	C_TS_TA_1

Parameter in Steuerungsrichtung

(stationsbezogener Parameter, jede nur in regulärer Richtung angewendete Typkennung ist mit „**X**“ zu markieren, mit „**R**“ falls nur in entgegengesetzter Richtung und mit „**B**“ falls in beiden Richtungen angewendet)

	<110>:=	Parameter für Messwerte, normierter Wert	P_ME_NA_1
	<111>:=	Parameter für Messwerte, skalierter Wert	P_ME_NB_1
	<112>:=	Parameter für Messwerte, verkürzte Gleitkommazahl	P_ME_NC_1
	<113>:=	Parameter für Aktivierung	P_AC_NA_1

Dateiübertragung

(stationsbezogener Parameter, jede nur in regulärer Richtung angewendete Typkennung ist mit „**X**“ zu markieren, mit „**R**“ falls nur in entgegengesetzter Richtung und mit „**B**“ falls in beiden Richtungen angewendet)

	<120>:=	Datei bereit	F_FR_NA_1

☐ <121>:= Abschnitt bereit F_SR_NA_1

☐ <122>:= Abfrage Dateiverzeichnis, -auswahl, -abfrage, Abschnittsabfrage F_SC_NA_1

☐ <123>:= Letzter Abschnitt, letztes Segment F_LS_NA_1

☐ <124>:= Dateibestätigung, Abschnittsbestätigung F_AF_NA_1

☐ <125>:= Segment F_SG_NA_1

☐ <126>:= Dateiverzeichnis [leer oder **X**, nur in Überwachungsrichtung verfügbar (genormt)] F_DR_TA_1

☐ <127>:= QueryLog – Anforderung einer Archiv-Datei F_SC_NB_1

Zuweisungen der Übertragungsursachen zu den Typkennungen

(stationsbezogene Parameter)

Schattierte Felder: werden nicht benötigt.

Schwarze Felder stellen in dieser anwendungsbezogenen Norm nicht zulässige Kombinationen dar.

Leer [N1)]: Funktion oder ASDU wird nicht angewendet.

Markierung der Kombinationen Typkennung/Übertragungsursache mit:

„**X**“, falls nur in der Normrichtung angewendet,

„**R**“, falls nur in der entgegengesetzten Richtung angewendet,

„**B**“, falls in beiden Richtungen angewendet.

Typkennung		Übertragungsursache																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	20 bis 36	37 bis 41	44	45	46	47
<1>	M_SP_NA_1														X					
~~<2>~~	~~M_SP_TA_1~~			■		■						■	■							
<3>	M_DP_NA_1														X					
~~<4>~~	~~M_DP_TA_1~~			■		■						■	■							
<5>	M_ST_NA_1																			
~~<6>~~	~~M_ST_TA_1~~			■		■						■	■							
<7>	M_BO_NA_1																			
~~<8>~~	~~M_BO_TA_1~~			■		■														
<9>	M_ME_NA_1																			
~~<10>~~	~~M_ME_TA_1~~			■		■														
<11>	M_ME_NB_1																			
~~<12>~~	~~M_ME_TB_1~~			■		■														

Typkennung		Übertragungsursache																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	20 bis 36	37 bis 41	44	45	46	47
<13>	M_ME_NC_1	X													X					
~~<14>~~	~~M_ME_TC_1~~			■		■														
<15>	M_IT_NA_1																			
~~<16>~~	~~M_IT_TA_1~~			■												■				
~~<17>~~	~~M_EP_TA_1~~			■																

N1) Nationale Fußnote: Gemeint sind die weißen Felder in der anschließenden Zuweisungstabelle.

Typkennung		Übertragungsursache																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	20 bis 36	37 bis 41	44	45	46	47
~~<18>~~	~~M_EP_TB_1~~			■																
~~<19>~~	~~M_EP_TC_1~~			■																
<20>	M_PS_NA_1																			
<21>	M_ME_ND_1																			
<30>	M_SP_TB_1			X																
<31>	M_DP_TB_1			X								X	X							
<32>	M_ST_TB_1																			
<33>	M_BO_TB_1																			
<34>	M_ME_TD_1																			
<35>	M_ME_TE_1																			
<36>	M_ME_TF_1			X																
<37>	M_IT_TB_1																			
<38>	M_EP_TD_1																			
<39>	M_EP_TE_1																			
<40>	M_EP_TF_1																			
<45>	C_SC_NA_1																			
<46>	C_DC_NA_1																			
<47>	C_RC_NA_1																			
<48>	C_SE_NA_1																			
<49>	C_SE_NB_1																			
<50>	C_SE_NC_1																			
<51>	C_BO_NA_1																			
<58>	C_SC_TA_1																			
<59>	C_DC_TA_1						R	X			X									
<60>	C_RC_TA_1																			
<61>	C_SE_TA_1																			
<62>	C_SE_TB_1																			
<63>	C_SE_TC_1						R	X			X									
<64>	C_BO_TA_1																			
<70>	M_EI_NA_1*																			
<100>	C_IC_NA_1						R	X			X									
<101>	C_CI_NA_1																			
<102>	C_RD_NA_1																			
<103>	C_CS_NA_1			■																
~~<104>~~	~~C_TS_NA_1~~						■	■									■	■	■	■
<105>	C_RP_NA_1																			
~~<106>~~	~~C_CD_NA_1~~			■			■	■									■	■	■	■
<107>	C_TS_TA_1																			
<110>	P_ME_NA_1																			
<111>	P_ME_NB_1																			
<112>	P_ME_NC_1																			
<113>	P_AC_NA_1																			
<120>	F_FR_NA_1																			
<121>	F_SR_NA_1																			
<122>	F_SC_NA_1																			
<123>	F_LS_NA_1																			
<124>	F_AF_NA_1																			
<125>	F_SG_NA_1																			
<126>	F_DR_TA_1*																			
<126>	F_DR_TA_1*																			
<127>	F_SC_NB_1																			

* Leer oder nur **X**.

9.6 Grundlegende Anwendungsfunktionen

Stationsinitialisierung

(stationsbezogener Parameter, bei Anwendung mit „**X**“ markieren)

☒ Stationsinitialisierung

Zyklische Datenübertragung

(stationsbezogener Parameter, jede nur in regulärer Richtung angewendete Typkennung ist mit „**X**“ zu markieren, mit „**R**“ falls nur in entgegengesetzter Richtung und mit „**B**“ falls in beiden Richtungen angewendet)

☒ Zyklische Datenübertragung

Abrufprozedur

(stationsbezogener Parameter, jede nur in regulärer Richtung angewendete Typkennung ist mit „**X**“ zu markieren, mit „**R**“ falls nur in entgegengesetzter Richtung und mit „**B**“ falls in beiden Richtungen angewendet)

☐ Abrufprozedur

Spontane Übertragung

(stationsbezogener Parameter, jede nur in regulärer Richtung angewendete Typkennung ist mit „**X**“ zu markieren, mit „**R**“ falls nur in entgegengesetzter Richtung und mit „**B**“ falls in beiden Richtungen angewendet)

☒ Spontane Übertragung

Doppelübertragung von Informationsobjekten mit der Übertragungsursache spontan

(stationsbezogener Parameter, jeder Informationstyp ist mit „**X**“ zu markieren, falls bei einer einzigen spontanen Änderung eines Informationsobjekts eine Typkennung ohne Zeitmarke und die zugehörige Typkennung mit Zeitmarke übertragen werden)

Die folgenden Typkennungen dürfen nacheinander in Folge eines einzigen Zustandswechsels eines Informationsobjekts übertragen werden. Die einzelnen Adressen der Informationsobjekte, die für die Doppelübertragung vorgesehen sind, werden in einer projektspezifischen Liste festgelegt.

☐ Einzelmeldung M_SP_NA_1, M_SP_TA_1, M_SP_TB_1 und M_SP_NA_1

☐ Doppelmeldung M_DP_NA_1, M_DP_TA_1 und M_DP_TB_1

☐ Stufenstellungsmeldung M_ST_NA_1, M_ST_TA_1 und M_ST_TB_1

☐ Bitmuster von 32 bit M_BO_NA_1, M_BO_TA_1 und M_BO_TB_1 (falls für ein bestimmtes Projekt festgelegt)

☐ Messwert, normierter Wert M_ME_NA_1, M_ME_TA_1, M_ME_ND_1 und M_ME_TD_1

☐ Messwert, skalierter Wert M_ME_NB_1, M_ME_TB_1 und M_ME_TE_1

☐ Messwert, verkürzte Gleitkommazahl M_ME_NC_1, M_ME_TC_1 und M_ME_TF_1

Stationsabfrage

(stationsbezogener Parameter, jede nur in regulärer Richtung angewendete Typkennung ist mit „**X**“ zu markieren, mit „**R**“ falls nur in entgegengesetzter Richtung und mit „**B**“ falls in beiden Richtungen angewendet)

☒ Global

☐ Gruppe 1 ☐ Gruppe 7 ☐ Gruppe 13

☐ Gruppe 2	☐ Gruppe 8	☐ Gruppe 14
☐ Gruppe 3	☐ Gruppe 9	☐ Gruppe 15
☐ Gruppe 4	☐ Gruppe 10	☐ Gruppe 16
☐ Gruppe 5	☐ Gruppe 11	☐
☐ Gruppe 6	☐ Gruppe 12	

Die Zuweisung der Adressen der Informationsobjekte zu jeder einzelnen Gruppe muss in einer getrennten Tabelle festgelegt werden.

Uhrzeitsynchronisation

(stationsbezogener Parameter, jede nur in regulärer Richtung angewendete Typkennung ist mit „**X**“ zu markieren, mit „**R**“ falls nur in entgegengesetzter Richtung und mit „**B**“ falls in beiden Richtungen angewendet)

☐ Uhrzeitsynchronisation

☐ Wochentag benutzt

☐ Bit RES1 oder GEN (Zeitmarke ersetzt bzw. nicht ersetzt) benutzt

☐ Bit SU (Sommerzeit) benutzt

Wahlfrei, siehe 7.6.

Befehlsübertragung

(objektbezogener Parameter, jede nur in regulärer Richtung angewendete Typkennung ist mit „**X**“ zu markieren, mit „**R**“ falls nur in entgegengesetzter Richtung und mit „**B**“ falls in beiden Richtungen angewendet)

☒ Direkte Befehlsübertragung

☒ Direkte Sollwert-Befehlsübertragung

☐ Befehl „Anwahl und Ausführung“

☐ Sollwertbefehl „Anwahl und Ausführung“

☐ C_SE ACTTERM angewendet

☐ Keine zusätzliche Festlegung

☐ Kurze Befehlsausführungsdauer (Ausführungsdauer durch einen Systemparameter in Unterstation bestimmt)

☐ Lange Befehlsausführungsdauer (Ausführungsdauer durch einen Systemparameter in Unterstation bestimmt)

☐ Dauerbefehl

☐ Überwachung der maximalen Verzögerung von Befehlen und Sollwertbefehlen in Befehlsrichtung

☐ Maximal zulässige Verzögerung von Befehlen und Sollwertbefehlen

Übertragung von Zählwerten

(stations- oder objektbezogener Parameter, jede nur in regulärer Richtung angewendete Typkennung ist mit „**X**“ zu markieren, mit „**R**“ falls nur in entgegengesetzter Richtung und mit „**B**“ falls in beiden Richtungen angewendet)

☐ Modus A: Örtliches Umspeichern mit spontaner Übertragung

☐ Modus B: Örtliches Umspeichern mit Zählerabfrage

☐ Modus C: Umspeichern und Übertragen durch Zähler-Abfrage bei Umspeichern und Übertragen durch Zähler-Abfragebefehle

☐ Modus D: Umspeichern durch Zähler-Abfragebefehl, umgespeicherte Werte werden spontan übertragen

☐ Zählerabfrage

☐ Zähler umspeichern ohne Rücksetzen

☐ Zähler umspeichern mit Rücksetzen

☐ Zähler rücksetzen

☐ Allgemeine Zählerabfrage

☐ Zählerabfrage Gruppe 1

☐ Zählerabfrage Gruppe 2

☐ Zählerabfrage Gruppe 3

☐ Zählerabfrage Gruppe 4

Laden eines Parameters

(objektbezogener Parameter, jede nur in regulärer Richtung angewendete Typkennung ist mit „**X**“ zu markieren, mit „**R**“ falls nur in entgegengesetzter Richtung und mit „**B**“ falls in beiden Richtungen angewendet)

☐ Schwellenwert

☐ Glättungsfaktor

☐ Unterer Grenzwert für Messwertübertragung

☐ Oberer Grenzwert für Messwertübertragung

Parameter für Aktivierung

(objektbezogener Parameter, jede nur in regulärer Richtung angewendete Typkennung ist mit „**X**“ zu markieren, mit „**R**“ falls nur in entgegengesetzter Richtung und mit „**B**“ falls in beiden Richtungen angewendet)

☐ Act/deact der zyklischen oder periodischen Übertragung des adressierten Objekts

Prüfprozedur

(stationsbezogener Parameter, jede nur in regulärer Richtung angewendete Typkennung ist mit „**X**“ zu markieren, mit „**R**“ falls nur in entgegengesetzter Richtung und mit „**B**“ falls in beiden Richtungen angewendet)

☒ Prüfprozedur

Dateiübermittlung

(stationsbezogener Parameter, bei Anwendung mit „**X**“ markieren)

Dateiübermittlung in Überwachungsrichtung

☐ Transparente Datei

☐ Übermittlung von Störfalldaten aus Schutzeinrichtungen

☐ Übermittlung von Ereignisfolgen

☐ Übermittlung von Folgen aufgezeichneter Analogwerte

Dateiübermittlung in Steuerungsrichtung

☐ Transparente Datei

Hintergrundabfrage

(stationsbezogener Parameter, jede nur in regulärer Richtung angewendete Typkennung ist mit „**X**“ zu markieren, mit „**R**“ falls nur in entgegengesetzter Richtung und mit „**B**“ falls in beiden Richtungen angewendet)

☐ Hintergrundabfrage

Telegrammlaufzeit-Erfassung

(stationsbezogener Parameter, jede nur in regulärer Richtung angewendete Typkennung ist mit „**X**“ zu markieren, mit „**R**“ falls nur in entgegengesetzter Richtung und mit „**B**“ falls in beiden Richtungen angewendet)

■ ~~Telegrammlaufzeit-Erfassung~~

Festlegungen für Zeitüberwachungen

Parameter	Falls kein anderer Wert festgelegt	Bemerkungen	Ausgewählter Wert
t_0	30 s	Zeitüberwachung für die Verbindungsherstellung	
t_1	15 s	Zeitüberwachung für gesendete APDU oder Test-APDU	
t_2	10 s	Zeitüberwachung für Quittierungen, falls keine Datentelegramme übertragen werden $t_2 < t_1$	10
t_3	20 s	Zeitüberwachung für gesendete Testtelegramme im Falle langer Ruhezustände	

Maximalbereich der Zeitüberwachungswerte t_0 bis t_2 : 1 bis 255 s, Genauigkeit 1 s

Empfohlener Bereich des Zeitüberwachungswertes t_3 : 0 s bis 48 h, Auflösung 1 s

Große Zeitüberwachungswerte t_3 werden in Spezialfällen benötigt, wo Satelliten- oder Wählverbindungen verwendet werden (zum Beispiel bei nur täglichem oder wöchentlichem Verbindungsaufbau zur Datenübertragung).

Maximale Anzahl *k* der unquittierten APDU im I Format und späteste APDU-Quittierung (*w*)

Parameter	Falls kein anderer Wert festgelegt	Bemerkungen	Ausgewählter Wert
k	12 APDU	Maximale Differenz Anzahl der Empfangsfolgen zur Anzahl der Sendefolgen	
w	8 APDU	Späteste Quittierung nach Empfang von *w* APDU im I-Format	8

Maximaler Wertebereich *k*: 1 bis 32 767 ($2^{15} - 1$) APDU, Genauigkeit 1 APDU.

Maximaler Wertebereich *w*: 1 bis 32 767 APDU, Genauigkeit 1 APDU (Empfehlung: *w* sollte Zweidrittel von *k* nicht überschreiten).

Portnummer

Parameter	Wert	Bemerkungen
Portnummer	2404	in allen Fällen

Redundante Verbindungen

☐ Anzahl N der benutzten Verbindungen in einer Redundanzgruppe

RFC-2200-Sammlung

RFC 2200 ist ein offizieller Internet-Standard, der den Stand der Normung im Internet angewendeter Protokolle beschreibt, wie sie durch das Internet Architecture Board (IAB) festgelegt sind. Es bietet ein breites Spektrum aktueller, im Internet angewendeter Standards. Die geeignete Auswahl in der vorliegenden Norm festgelegter Dokumente aus RFC 2200 für vorgegebene Projekte ist durch den Anwender dieser Norm auszuwählen.

☐ Ethernet 802.3

☐ Serielle Schnittstelle X.21

☐ Andere Auswahl aus RFC 2200:

Liste der anzuwendenden RFC-2200-Dokumente

1. ……..
2. ……..
3. ……..
4. ……..
5. ……..
6. ……..
7. usw.

10 Redundante Verbindungen

10.1 Einleitung

Diese anwendungsbezogene Norm legt den Netzwerkzugriff für IEC 60870-5-101 unter Nutzung des Transport-Profils TCP/IP fest und konzentriert sich dabei hauptsächlich auf eine einzelne TCP-Verbindung.

In vielen Fällen ist jedoch zur Erhöhung der Verfügbarkeit des Kommunikationssystems Redundanz erforderlich. In diesen Fällen sollten mehrere redundante Verbindungen zwischen den beiden Stationen aufgebaut werden. Dieses Kapitel legt die Anforderungen an die Kompatibilität fest, die entstehen, wenn Standby-Verbindungen als redundante Verbindungen benutzt werden.

10.2 Allgemeine Anforderungen

Redundante Kommunikation in einem System, das IEC 60870-5-104 benutzt, kann durch die Unterstützung der Möglichkeit erreicht werden, mehr als eine logische Verbindung zwischen zwei Stationen aufbauen zu können. Eine logische Verbindung wird durch die eindeutige Kombination von zwei IP-Adresse und zwei Portnummern festgelegt, nämlich dem IP-Adress/Portnummer-Paar der Zentralstation und dem IP-Adress/Portnummer-Paar der Unterstation.

B UML-Diagramme

B.1 Paket rtuSimulator

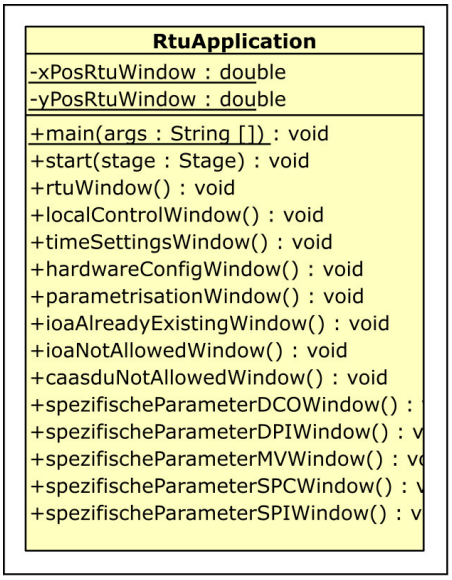


Abbildung B.1: UML-Klassendiagramm `RtuApplication`

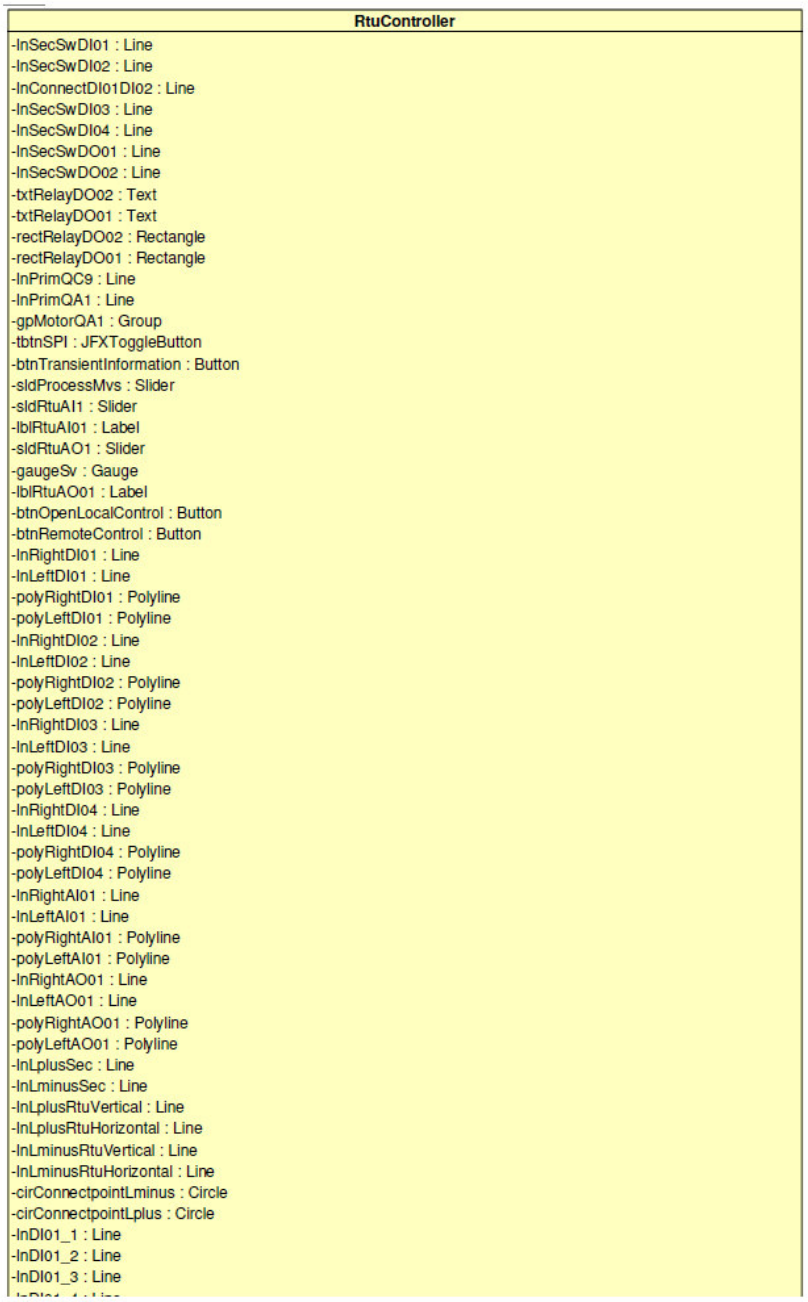


Abbildung B.2: UML-Klassendiagramm `RtuController` - Teil 1

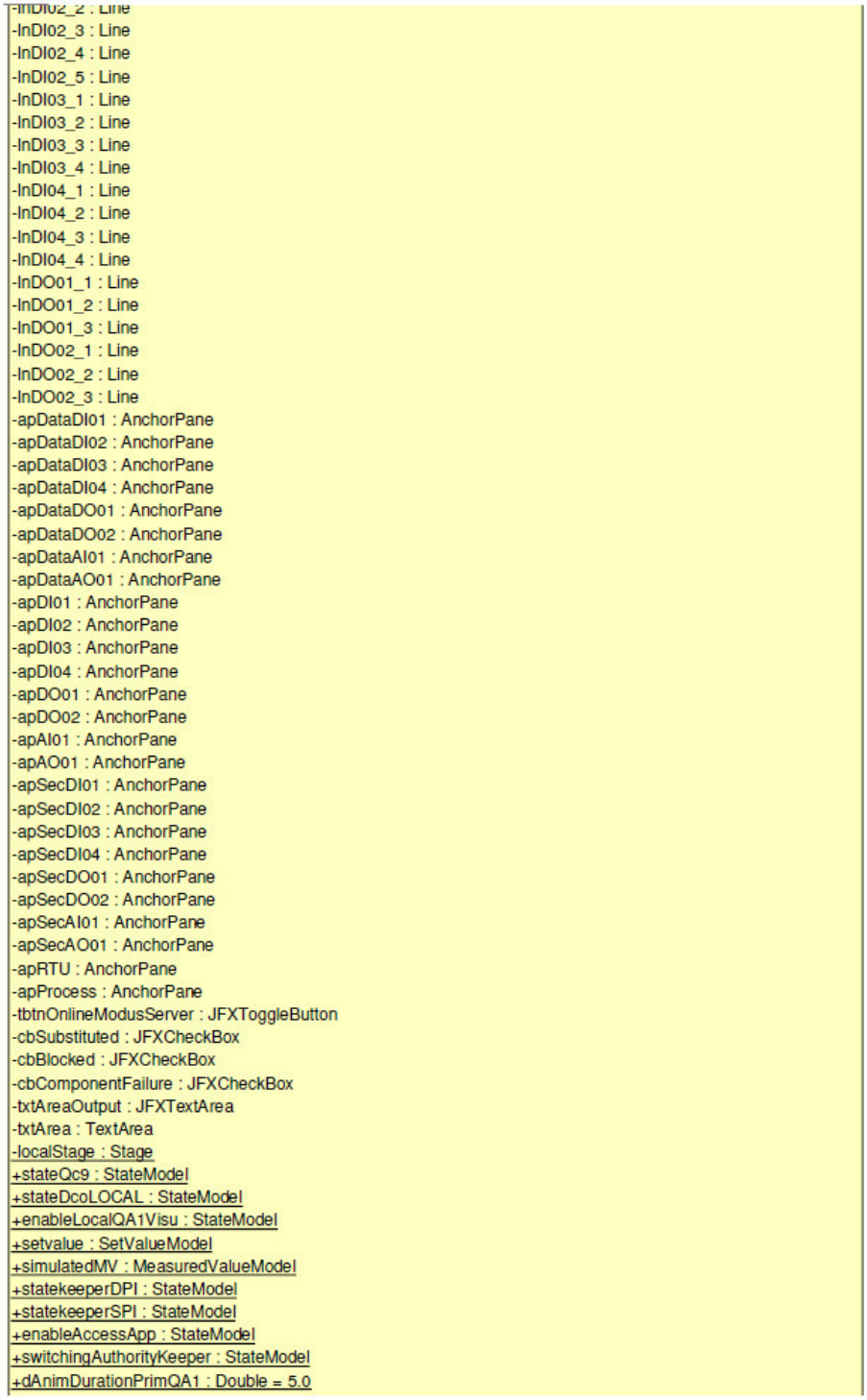


Abbildung B.3: UML-Klassendiagramm `RtuController` - Teil 2

Abbildung B.4: UML-Klassendiagramm `RtuController` - Teil 3

Abbildung B.5: UML-Klassendiagramm `RtuController` - Teil 4

Abbildung B.6: UML-Klassendiagramm `RtuController` - Teil 5

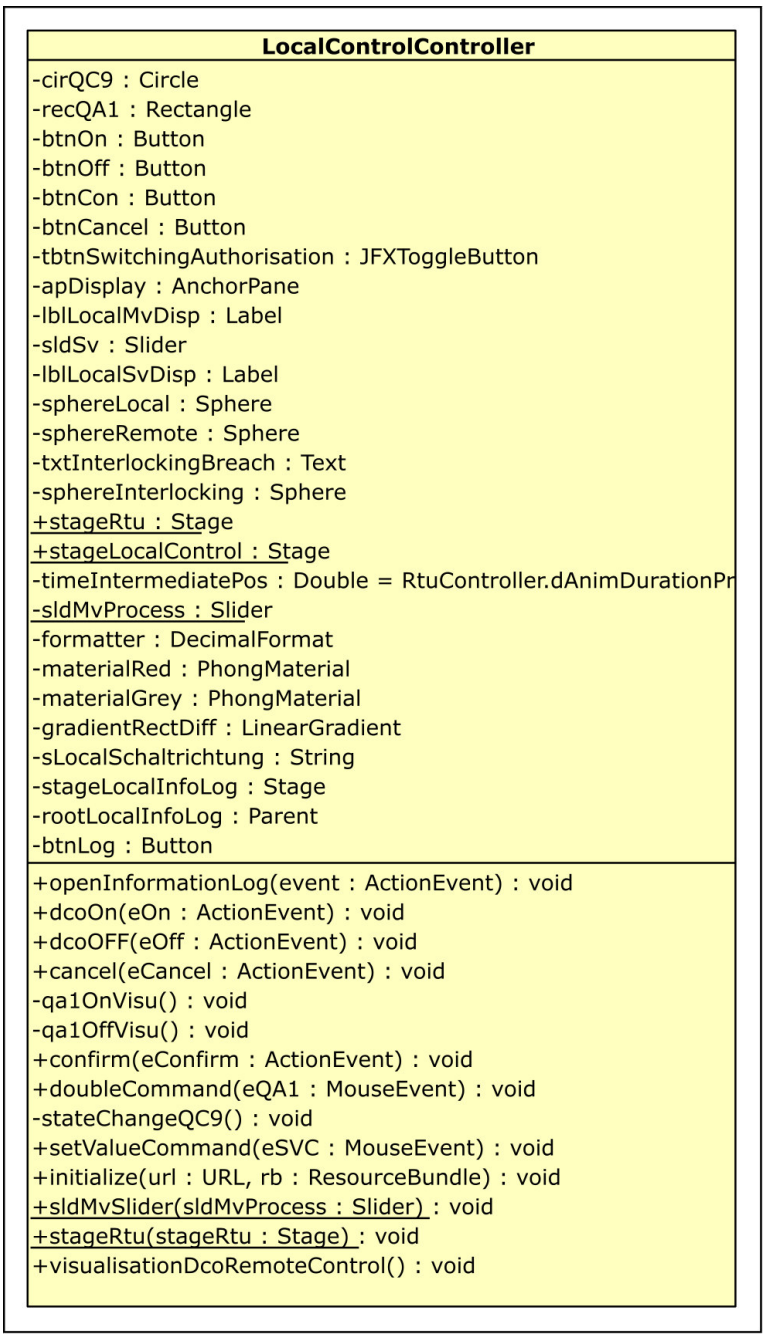


Abbildung B.7: UML-Klassendiagramm `LocalControlController`

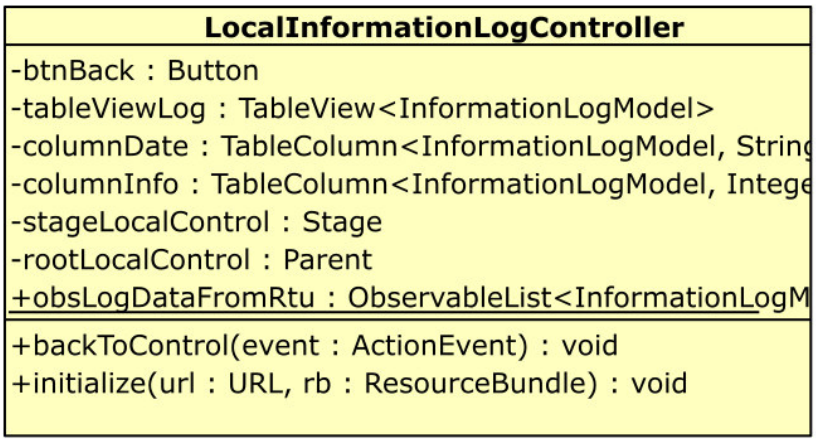


Abbildung B.8: UML-Klassendiagramm `LocalInfoLogController`

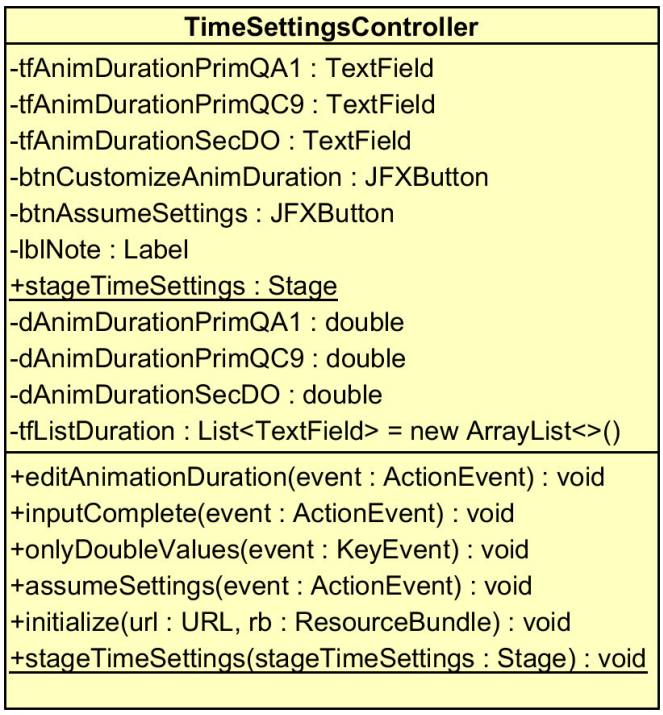


Abbildung B.9: UML-Klassendiagramm `TimeSettingsController`

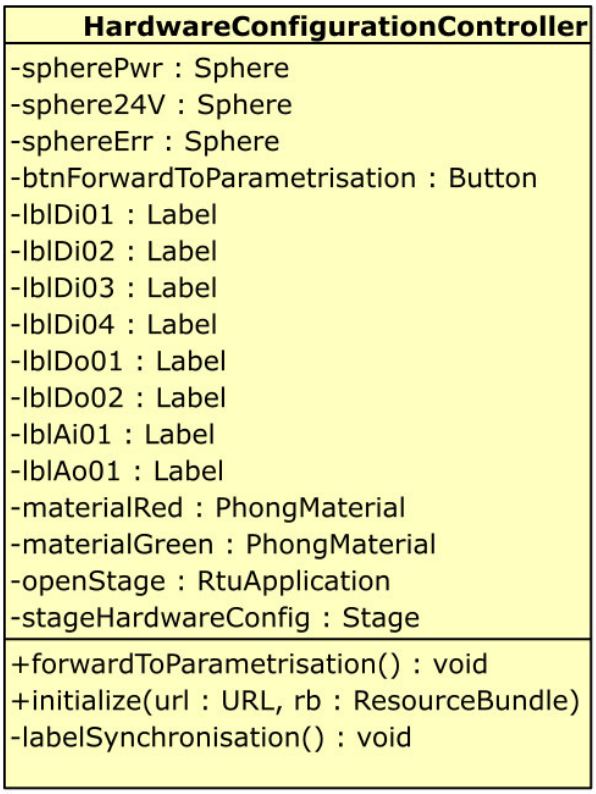


Abbildung B.10: UML-Klassendiagramm `HardwareConfigurationController`

Abbildung B.11: UML-Klassendiagramm `ParametrisationController`

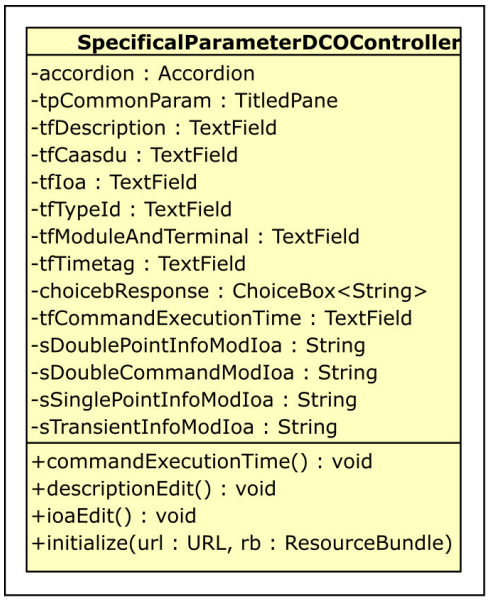


Abbildung B.12: UML-Klassendiagramm `SpecificalParameterDCOController`

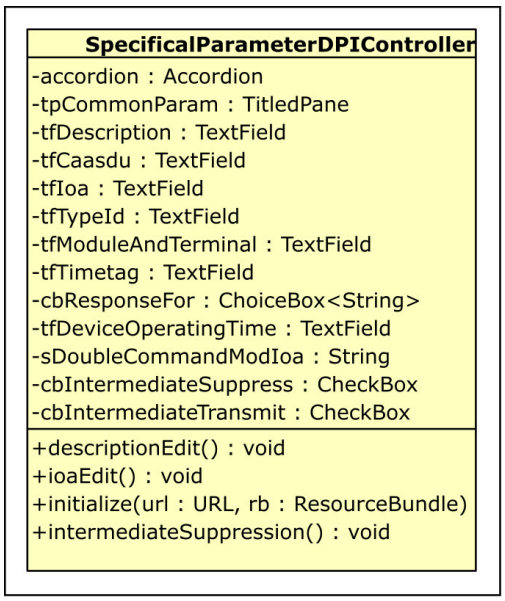


Abbildung B.13: UML-Klassendiagramm `SpecificalParameterDPIController`

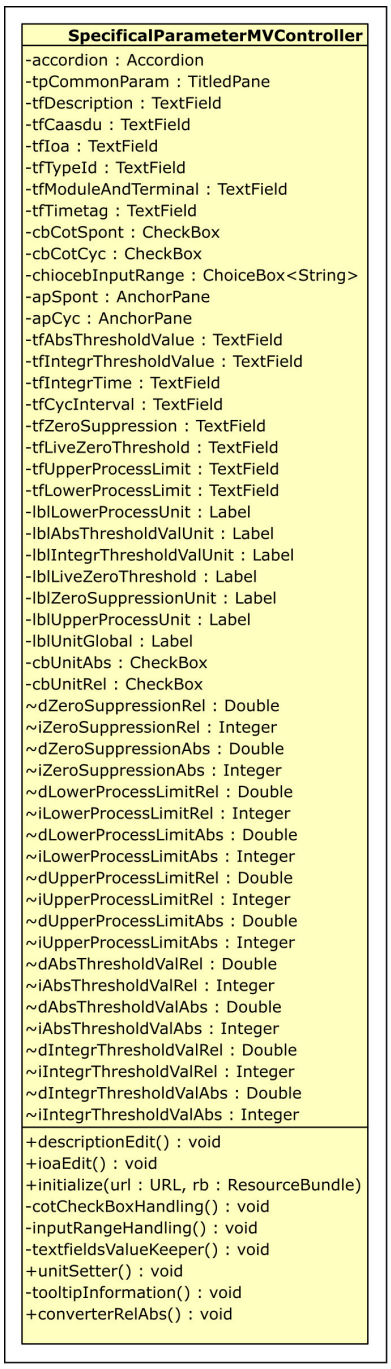


Abbildung B.14: UML-Klassendiagramm `SpecificalParameterMVController`

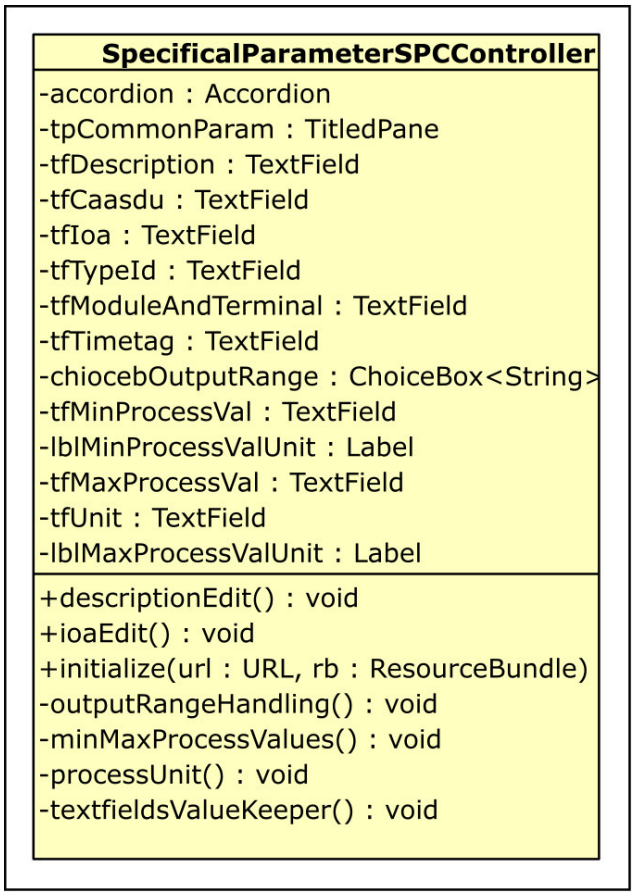


Abbildung B.15: UML-Klassendiagramm `SpecificalParameterSPCController`

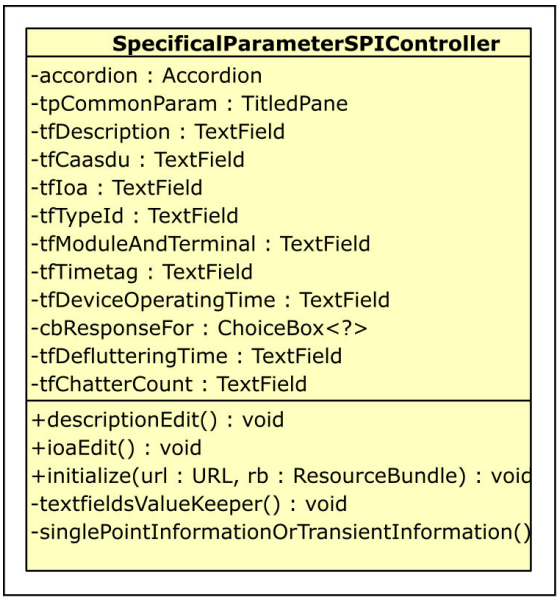


Abbildung B.16: UML-Klassendiagramm `SpecificalParameterSPIController`

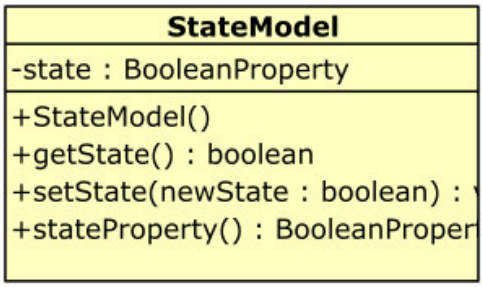


Abbildung B.17: UML-Klassendiagramm `StateModel`

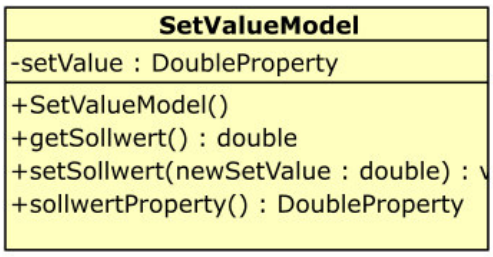


Abbildung B.18: UML-Klassendiagramm `SetValueModel`

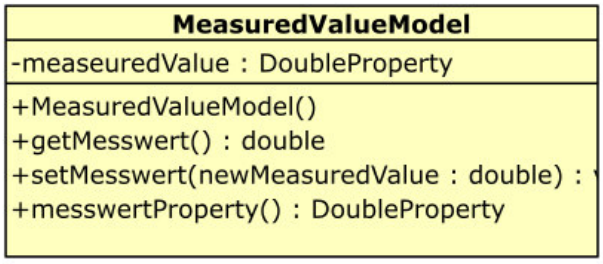


Abbildung B.19: UML-Klassendiagramm `MeasuredValueModel`

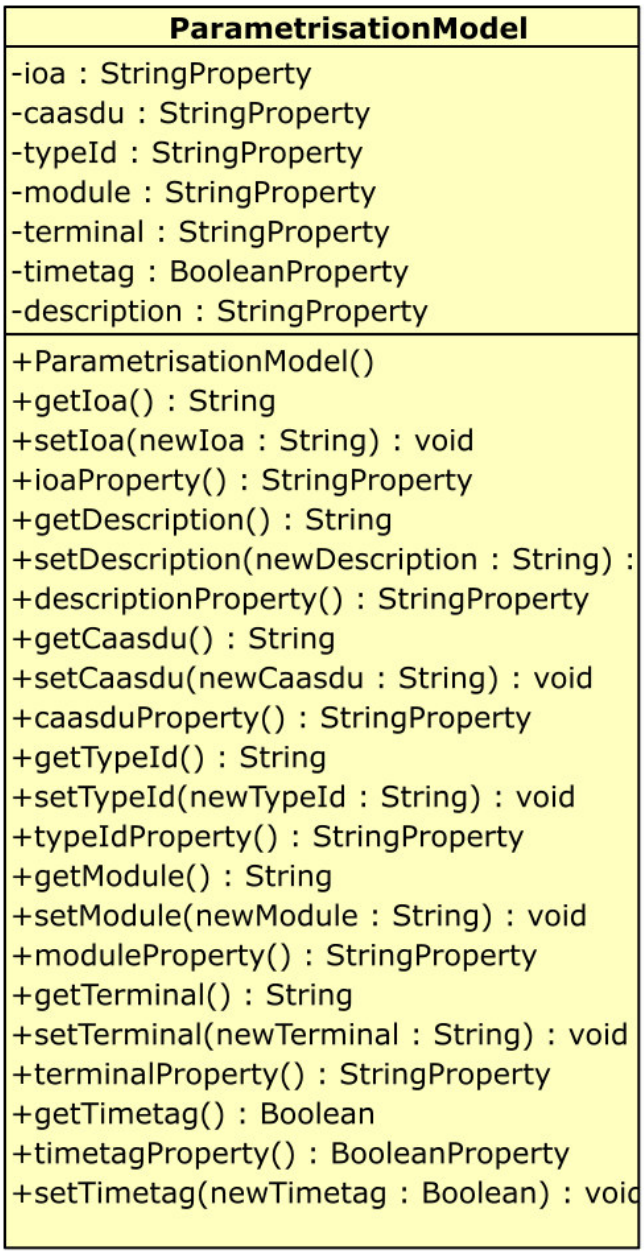


Abbildung B.20: UML-Klassendiagramm `ParameterisationModel`

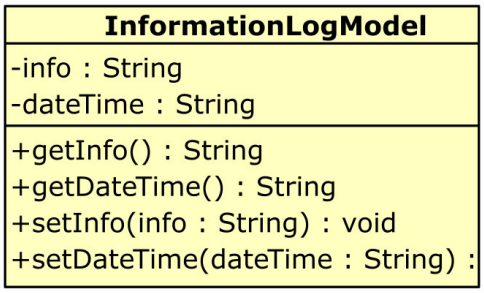


Abbildung B.21: UML-Klassendiagramm `InformationLogModel`

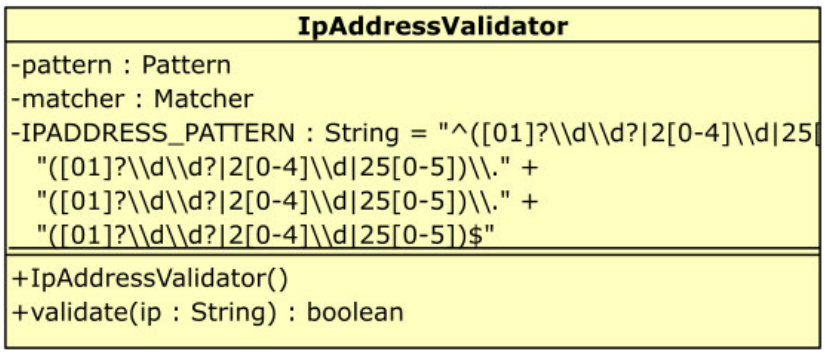


Abbildung B.22: UML-Klassendiagramm `IpAdressValidator`

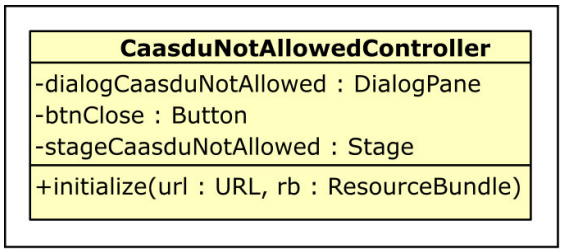


Abbildung B.23: UML-Klassendiagramm `CaasduNotAllowedController`

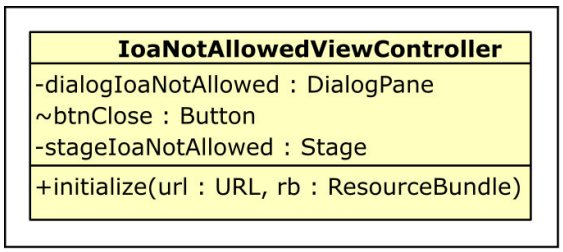


Abbildung B.24: UML-Klassendiagramm `IoaNotAllowedViewController`

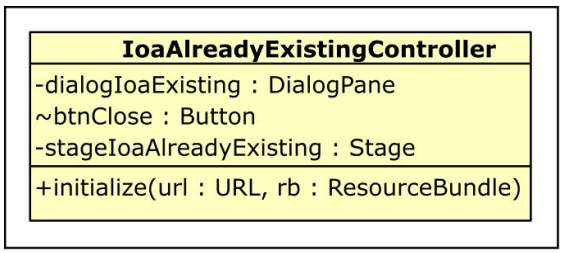


Abbildung B.25: UML-Klassendiagramm `IoaAlreadyExistingController`

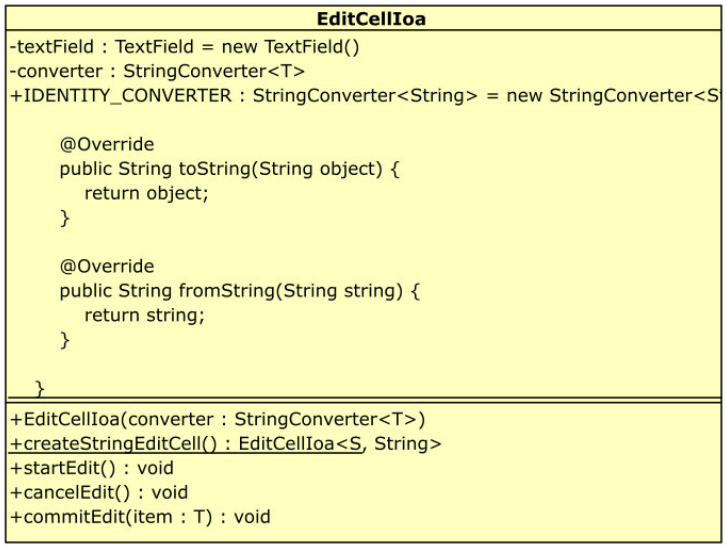


Abbildung B.26: UML-Klassendiagramm `EditCellIoa`

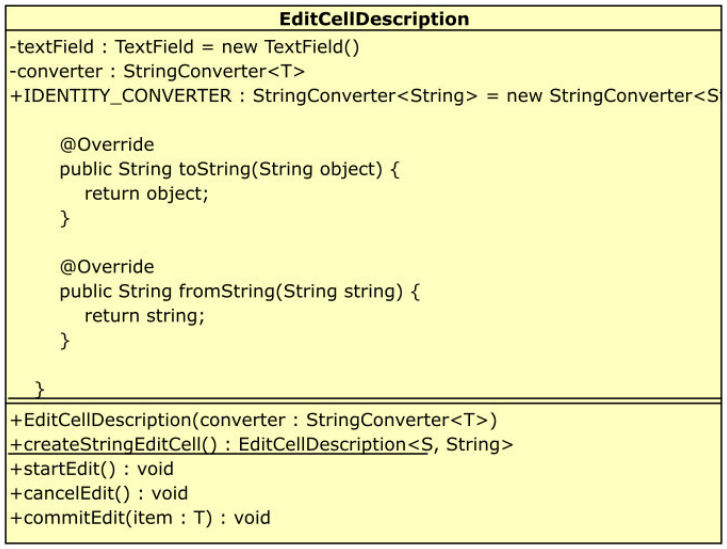


Abbildung B.27: UML-Klassendiagramm `EditCellCaasdu`

B.2 Paket tcpipServer104

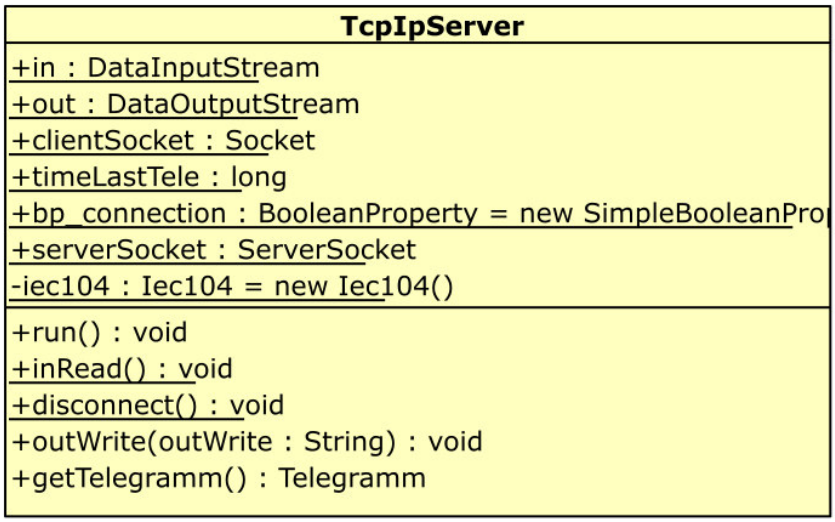


Abbildung B.28: UML-Klassendiagramm `TcpIpServer`

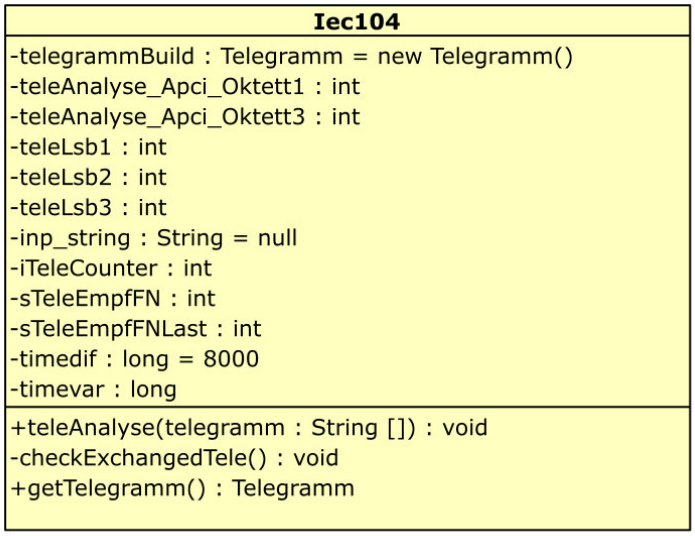


Abbildung B.29: UML-Klassendiagramm `Iec104`

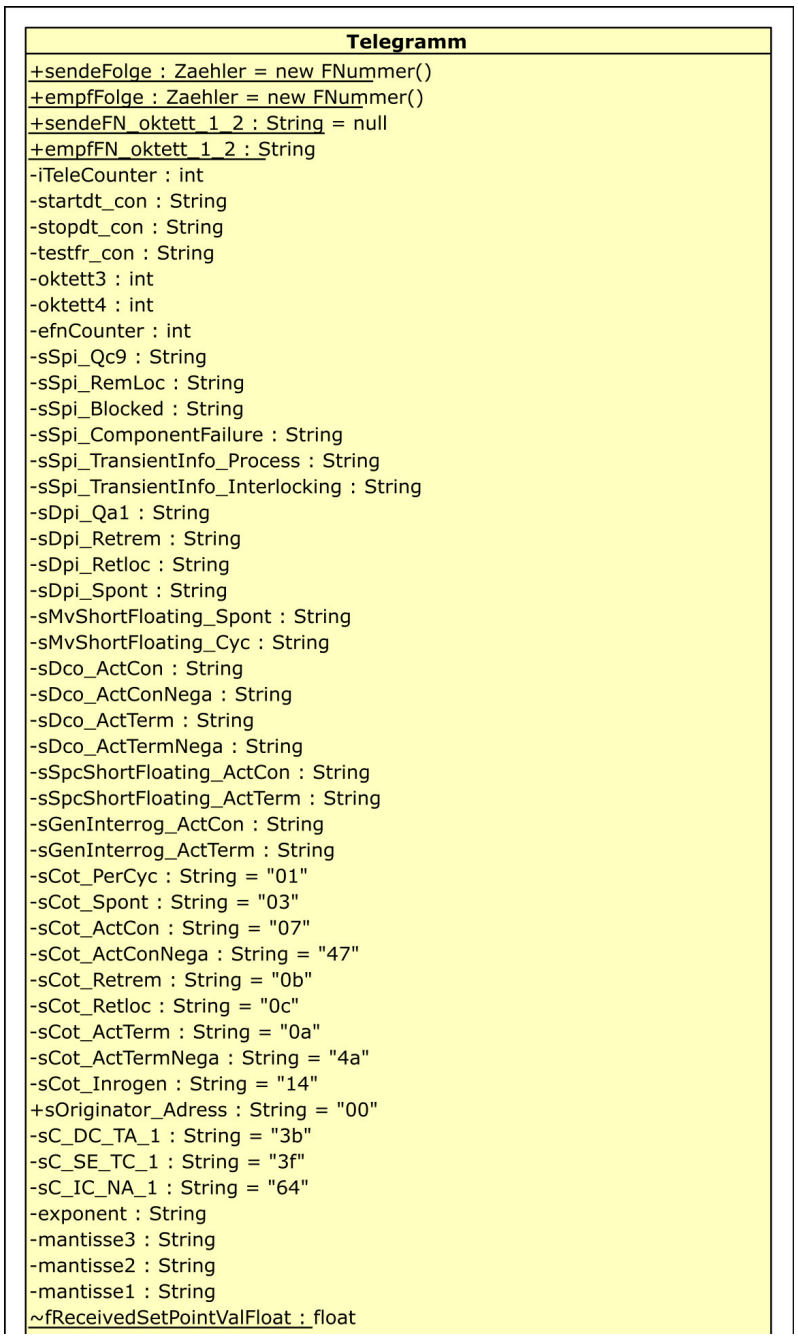


Abbildung B.30: UML-Klassendiagramm `Telegramm` - Teil 1

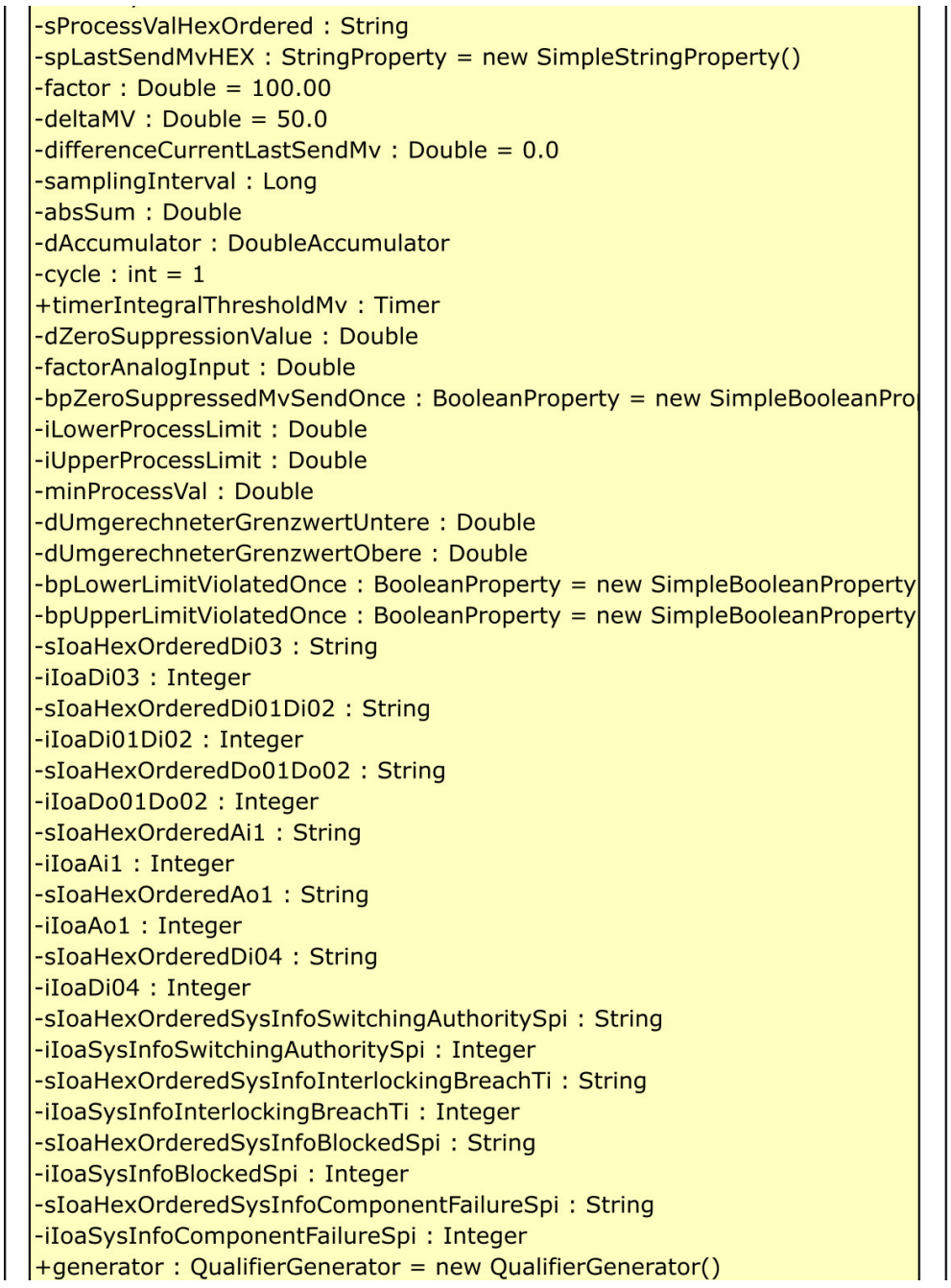


Abbildung B.31: UML-Klassendiagramm `Telegramm` - Teil 2

Abbildung B.32: UML-Klassendiagramm `Telegramm` - Teil 3

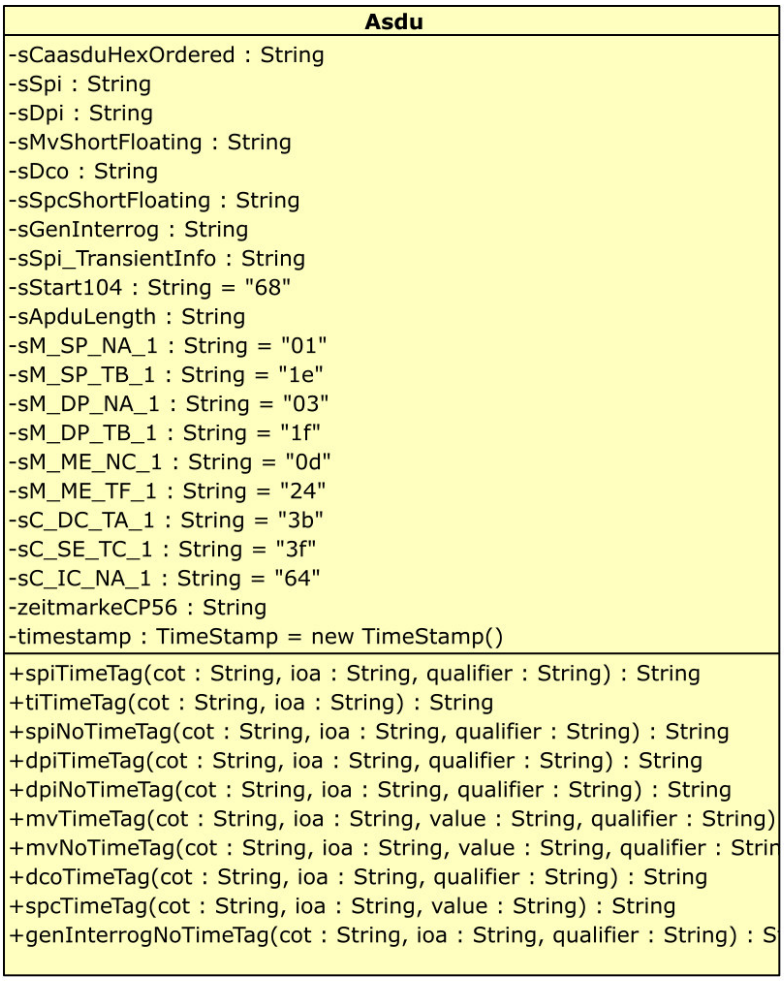


Abbildung B.33: UML-Klassendiagramm `Asdu`

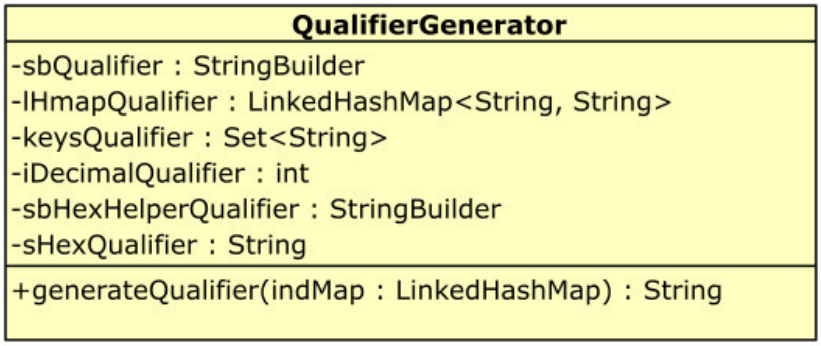


Abbildung B.34: UML-Klassendiagramm `QualifierGenerator`

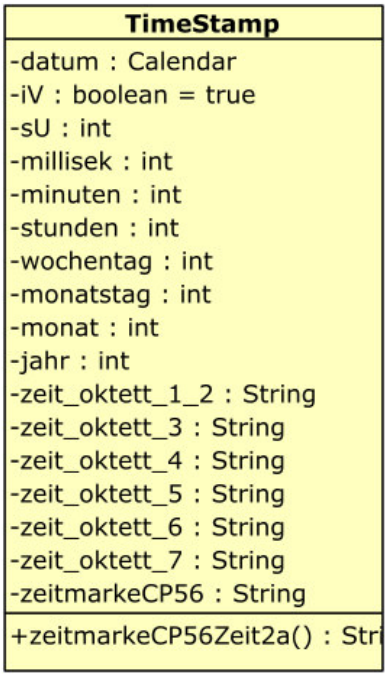


Abbildung B.35: UML-Klassendiagramm `TimeStamp`

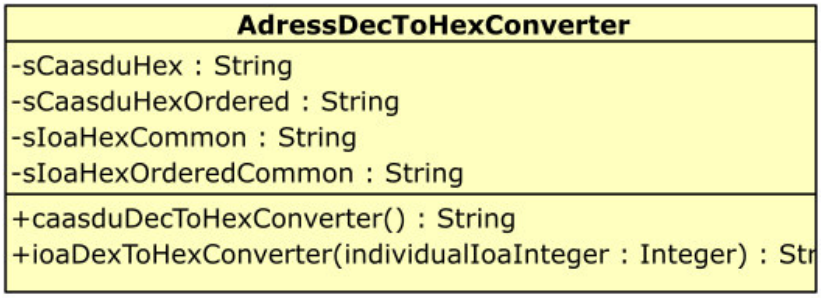


Abbildung B.36: UML-Klassendiagramm `AdressDecToHexConverter`

B.3 Paket rtuServerFolgenummer

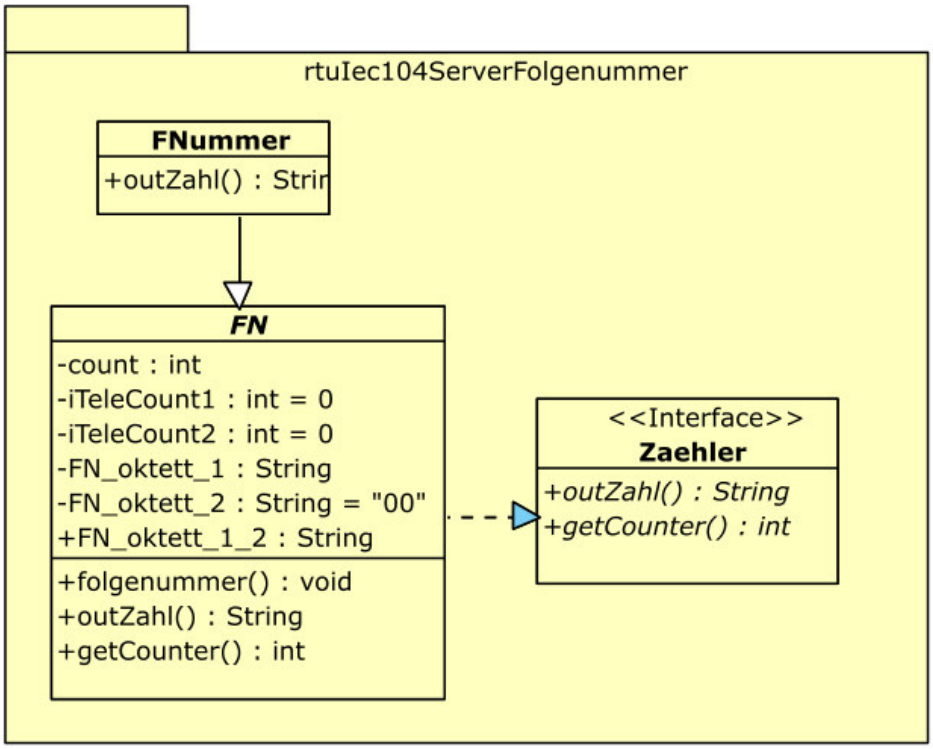


Abbildung B.37: UML-Klassendiagramm `FNummer`, Schnittstellendiagramm `FN` und `Zaehler`

C Quellcode

Der vollständige Quelltext ist aus Gründen der Übersichtlichkeit im PDF-Dokument auf der beiliegenden CD im Anhang zu finden.