Programmierung eines OPC UA .NET Clients mit C# für den SIMATIC NET OPC UA Server

SIMATIC NET OPC UA Server

https://support.industry.siemens.com/cs/ww/de/view/42014088

Siemens Industry Online Support

Gewährleistung und Haftung

Hinweis

Die Applikationsbeispiele sind unverbindlich und erheben keinen Anspruch auf Vollständigkeit hinsichtlich Konfiguration und Ausstattung sowie jeglicher Eventualitäten. Die Applikationsbeispiele stellen keine kundenspezifischen Lösungen dar, sondern sollen lediglich Hilfestellung bieten bei typischen Aufgabenstellungen. Sie sind für den sachgemäßen Betrieb der beschriebenen Produkte selbst verantwortlich. Diese Applikationsbeispiele entheben Sie nicht der Verpflichtung zu sicherem Umgang bei Anwendung, Installation, Betrieb und Wartung. Durch Nutzung dieser Applikationsbeispiele erkennen Sie an, dass wir über die beschriebene Haftungsregelung hinaus nicht für etwaige Schäden haftbar gemacht werden können. Wir behalten uns das Recht vor, Änderungen an diesen Applikationsbeispielen jederzeit ohne Ankündigung durchzuführen. Bei Abweichungen zwischen den Vorschlägen in diesem Applikationsbeispiel und anderen Siemens Publikationen, wie z.B. Katalogen, hat der Inhalt der anderen Dokumentation Vorrang.

Für die in diesem Dokument enthaltenen Informationen übernehmen wir keine Gewähr.

Unsere Haftung, gleich aus welchem Rechtsgrund, für durch die Verwendung der in diesem Applikationsbeispiel beschriebenen Beispiele, Hinweise, Programme, Projektierungs- und Leistungsdaten usw. verursachte Schäden ist ausgeschlossen, soweit nicht z.B. nach dem Produkthaftungsgesetz in Fällen des Vorsatzes, der groben Fahrlässigkeit, wegen der Verletzung des Lebens, des Körpers oder der Gesundheit, wegen einer Übernahme der Garantie für die Beschaffenheit einer Sache, wegen des arglistigen Verschweigens eines Mangels oder wegen Verletzung wesentlicher Vertragspflichten zwingend gehaftet wird. Der Schadensersatz wegen Verletzung wesentlicher Vertragspflichten ist jedoch auf den vertragstypischen, vorhersehbaren Schaden begrenzt, soweit nicht Vorsatz oder grobe Fahrlässigkeit vorliegt oder wegen der Verletzung des Lebens, des Körpers oder der Gesundheit zwingend gehaftet wird. Eine Änderung der Beweislast zu Ihrem Nachteil ist hiermit nicht verbunden.

Weitergabe oder Vervielfältigung dieser Applikationsbeispiele oder Auszüge daraus sind nicht gestattet, soweit nicht ausdrücklich von Siemens Industry Sector zugestanden.

© Siemens AG 2017 All rights reserved

Security-hinweise

Siemens bietet Produkte und Lösungen mit Industrial Security-Funktionen an, die den sicheren Betrieb von Anlagen, Lösungen, Maschinen, Geräten und/oder Netzwerken unterstützen. Sie sind wichtige Komponenten in einem ganzheitlichen Industrial Security-Konzept. Die Produkte und Lösungen von Siemens werden unter diesem Gesichtspunkt ständig weiterentwickelt. Siemens empfiehlt, sich unbedingt regelmäßig über Produkt-Updates zu informieren.

Für den sicheren Betrieb von Produkten und Lösungen von Siemens ist es erforderlich, geeignete Schutzmaßnahmen (z. B. Zellenschutzkonzept) zu ergreifen und jede Komponente in ein ganzheitliches Industrial Security-Konzept zu integrieren, das dem aktuellen Stand der Technik entspricht. Dabei sind auch eingesetzte Produkte von anderen Herstellern zu berücksichtigen. Weitergehende Informationen über Industrial Security finden Sie unter http://www.siemens.com/industrialsecurity.

Um stets über Produkt-Updates informiert zu sein, melden Sie sich für unseren produktspezifischen Newsletter an. Weitere Informationen hierzu finden Sie unter http://support.automation.siemens.com.

Inhaltsverzeichnis

Gewährleistung und Haftung ... 2

1 Automatisierungsaufgabe ... 4

1.1 Übersicht ... 5
1.2 Anforderungen ... 6

2 Automatisierungslösung ... 8

2.1 Übersicht Gesamtlösung ... 8
2.2 Beschreibung der Kernfunktionalität ... 9
Abgrenzung ... 13
2.3 Verwendete Hard- und Software-Komponenten ... 13
2.4 Alternativlösungen ... 14

3 Grundlagen ... 15

3.1 Grundlagen OPC ... 15
3.2 Grundlagen von OPC Unified Architecture ... 17
3.2.1 OPC UA Spezifikationen ... 17
3.2.2 Aufbau des OPC UA Server Adressraums ... 19
3.2.3 Schnittstelle für den Zugriff auf den OPC UA Server Adressraum ... 23
3.2.4 Protokolle und Sicherheitsmechanismen ... 26
3.2.5 Abgrenzung und Vergleich mit OPC Data Access ... 30
3.3 Grundlagen zum Thema S7 Kommunikation ... 31
3.3.1 Allgemein ... 31
3.3.2 optimierte S7-Kommunikation ... 34

4 Funktionsmechanismen dieser Applikation ... 36

4.1 OPC UA Client Helper API ... 39
4.2 Simple OPC UA Client ... 41
4.3 Komfortabler OPC UA Client ... 44
4.3.1 Oberfläche ... 44
4.3.2 Klassendiagramm ... 45
4.3.3 Sequenzdiagramme ... 47
4.4 S7 Programm ... 52

5 Konfiguration und Projektierung ... 56

5.1 Konfiguration der SIMATIC S7 Stationen ... 56
5.2 Konfiguration der OPC Server Station ... 58
5.3 Konfiguration der OPC UA Security ... 62
5.3.1 OPC UA Remote Kommunikation ... 62
5.3.2 Zertifikatsspeicher ... 63
5.3.3 Authentication, SecurityPolicy und MessageSecurityMode ... 68

6 Installation ... 70

7 Inbetriebnahme der Applikation ... 72

8 Bedienung der Applikation ... 76

8.1 Bedienung Simple Client ... 76
8.2 Bedienung komfortabler OPC UA Client ... 78

9 Weitere Hinweise, Tipps und Tricks, etc. ... 84

10 Literaturhinweis ... 85

11 Historie ... 86

© Siemens AG 2017 All rights reserved

1 Automatisierungsaufgabe

Veranlassung

Mit OPC Unified Architecture (UA) existiert jetzt im SIMATIC NET OPC Server eine zusätzliche, komfortable und performante Möglichkeit der Prozessankopplung von PC-Systemen an SIMATIC S7, die sukzessive die bestehende OPC Data Access (DA) und Alarms & Events (A&E) Funktionen ersetzen wird.

Die wesentlichen Vorteile von OPC UA gegenüber herkömmlichen OPC Schnittstellen sind:

- Kommunikation über das Internet und über Firewalls hinweg.
- Optimiertes, robustes und fehlertolerantes Protokoll mit integrierten Sicherheitsmechanismen.
- OPC UA kann direkt in Anwendungen auf verschiedenen Betriebssystemen mit verschiedenen Programmiersprachen integriert werden.
- Alle OPC Informationen wie Daten oder Alarme sind in einem Namensraum integriert.
- Informationen können mit objektorientierten Mitteln beschrieben werden.

Zielgruppe

Diese Applikation ist gedacht für Endanwender, die einen fundierten Einstieg in diese Technologie benötigen und die Erfahrung mit der professionellen Erstellung von OPC UA Clients in C# unter .NET sammeln wollen.

Inhalt

Hier verschaffen Sie sich einen Überblick über die Verwendung der Kommunikationsschnittstelle OPC UA, die Daten, Alarme und Diagnoseinformationen aus SIMATIC S7 Steuerungen anbietet. Sie lernen die verwendeten Komponenten, Standard Hard- und Softwarekomponenten, sowie eigens erstellte Anwendersoftware kennen.

Die Anwendersoftware bietet Beispiele für die Erstellung von OPC UA Clients mit C# unter .NET. Enthalten sind eine vereinfachte, wieder verwendbare API, ein einfaches Beispiel und ein komplexes Beispiel mit einer komfortablen Oberfläche. Das Beispiel gibt auch Hinweise zur Optimierung und Erweiterung der Anwendung.

Copyright © Siemens AG 2017 All rights reserved

1.1 Übersicht

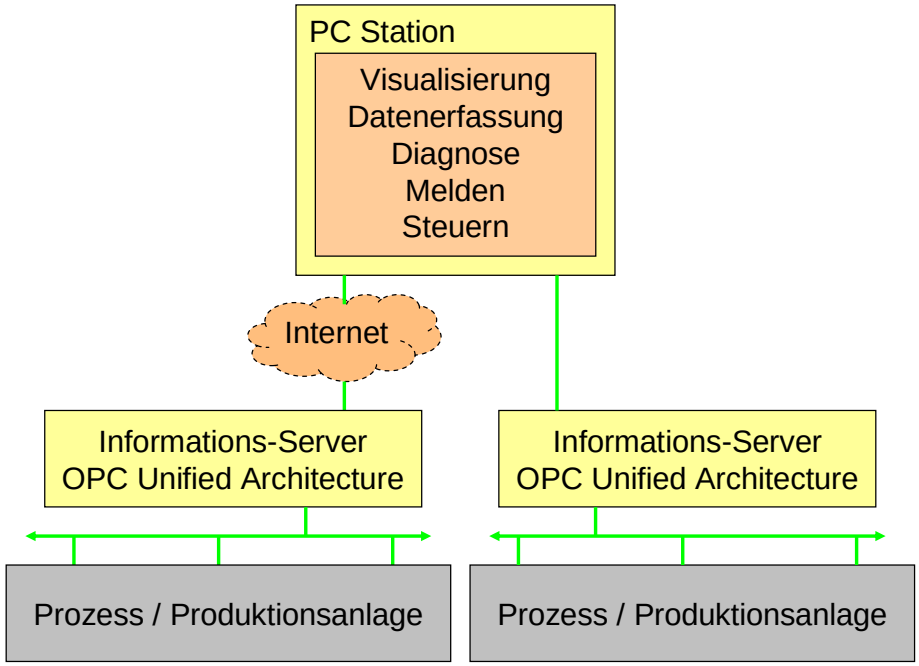
Einführung/Einleitung

Zur Realisierung einer Datenkopplung werden heute bevorzugt standardisierte Mechanismen verwendet, um einen derartigen Datenaustausch unabhängig vom eingesetzten Bussystem bzw. Protokoll oder sogar Hersteller zu halten. Auch für den Austausch von Ereignis- und Alarmmeldungen wird ein standardisierter Mechanismus für die Anbindung verschiedener Subsysteme verwendet werden. OPC UA kombiniert diese Funktionalität und bietet zusätzlich Authentifizierung und verschlüsselte Datenübertragung sowie erweiterte Diagnoseinformation.

Übersicht über die Automatisierungsaufgabe

Folgende Abbildung zeigt einen Überblick über die Automatisierungsaufgabe.

Abbildung 1-1



Beschreibung der Automatisierungsaufgabe

Im Automatisierungssystem soll der OPC UA Server als Informationsserver betrachtet werden, der einzelne Komponenten aber auch die gesamte Anlage abbilden und beschreiben kann. Durch den mit Zertifikaten geprüften und sicheren, verschlüsselten Zugriff ist auch eine Ankopplung von weiteren Standorten möglich.

Die Kernaufgabe dieses Beispiels ist der Zugriff auf Prozessdaten mit der OPC UA Schnittstelle. Dies wird erläutert durch die Erstellung einer einfachen, aber praxistauglichen, individuell erstellten Visualisierung auf Basis des neuen OPC UA Standards.

Copyright © Siemens AG 2017 All rights reserved

Die Applikation soll folgende Funktionalitäten beinhalten:

- Auswahl des Servers inklusive Sicherheitseinstellungen.
- Navigation durch den OPC UA Namensraum des Servers und Auswahl der Prozessvariablen.
- Lesen der Attribute inklusive des Wertes der ausgewählten Prozessvariablen.
- Beobachten des Wertes der ausgewählten Prozessvariablen.
- Schreiben des Wertes der ausgewählten Prozessvariablen.
- Nutzung von Blockdiensten über OPC UA.

Eine datentechnische Weiterverarbeitung (z.B. Speicherung in einer Datenbank, o.ä.) ist hier nicht Thema.

1.2 Anforderungen

Anforderungen durch die Automatisierungsaufgabe

Die Beispielapplikation ist in C# erstellt und nutzt die Schnittstellen der .NET API der OPC Foundation.

Es wird dem Anwender der praxistaugliche Umgang mit der OPC UA Schnittstelle unter .NET verdeutlicht. Basisschnittstelle ist das auf der SIMATIC NET Installation mitgelieferte .NET Client SDK der OPC Foundation.

Diese Schnittstelle bietet den vollen Funktionsumfang von OPC UA. Zur Vereinfachung der Schnittstelle wird eine Reduzierung auf die für dieses Beispiel erforderliche Funktionalität durchgeführt. Es wird eine effiziente und praxistaugliche Anleitung zu den OPC UA Diensten erarbeitet.

Mit der Gestaltung einer einfachen GUI Oberfläche wird die Basisfunktionalität von OPC UA demonstriert. Dabei wird die komplette Wirkungskette zwischen S7-Variable(n), OPC UA Namensraum und Zugriff vom Client aus in C# gezeigt:

- Anmelden, Abmelden und Authentifizierung am Server
- Durchsuchen des Namensraums nach Variablen
- Lesen, Schreiben und Beobachten von Variablen
- Einfaches Fehlerhandling

Im Beispiel wird die symbolische und absolute Adressierung gezeigt und die Verwendung der Variablen-Dienste Read, Write und Beobachten für die S7-Basistypen sowie die Nutzung der blockorientierten Dienste (Empfang und Senden von großen Datenblöcken) beschrieben.

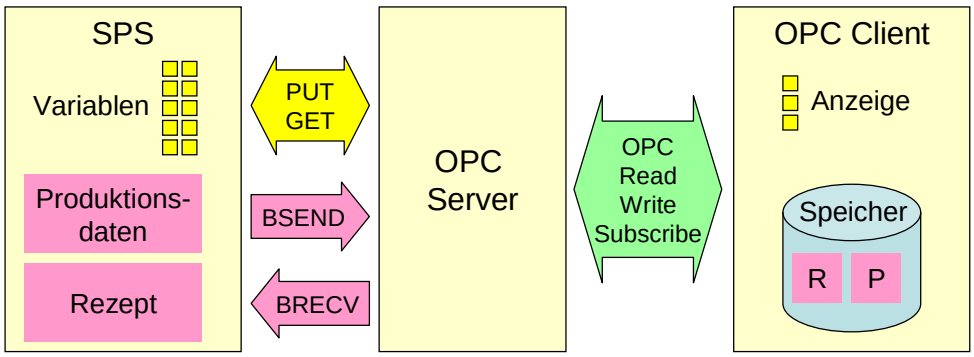
Die verschiedenen Diagnosemöglichkeiten und die programmtechnische Verarbeitung von Fehlerszenarien werden erläutert. Die Fehler können durch die Simulation von Verbindungsabbrüchen zwischen den verschiedenen Komponenten ausgelöst werden.

Copyright © Siemens AG 2017 All rights reserved

Anforderung an die Steuerung

Die Steuerung soll in der Lage sein, die erforderlichen Datenstrukturen und Datenmengen anzubieten und Werteänderungen zu simulieren. Eine konkrete Steuerungsaufgabe gibt es nicht, es soll lediglich der Zugriff auf die Daten veranschaulicht werden. Die Datenbereiche und die Interaktion mit den anderen Komponenten werden in der folgenden Abbildung gezeigt.

Abbildung 1-2



Das STEP7 Programm in der S7-CPU simuliert die Einzelwerte, die vom Client empfangen und angezeigt werden sollen (Variablen-Dienste). Als Einzelvariablen werden verschiedene Datentypen verwendet.

Das SPS Programm simuliert und generiert die nötigen Strukturen und Werte für die bidirektionale Übertragung größerer Datenmengen und ruft die blockorientierten Dienste (BSEND, BRECV) entsprechend auf. Dies wird genutzt für STRUCT oder ARRAY-Variablen mit mehreren 100 Bytes in Summe (Rezeptdaten, Produktionsdatensätze, o.ä.).

Zum Senden von Daten stößt die SPS aktiv die Übertragung eines blockorientierten Produktionsdatensatzes zum OPC UA Server an. Die SPS empfängt einen vom OPC UA Client gesendeten blockorientierten Datensatz (z.B. Rezept) und legt diesen in der entsprechenden Struktur in einem Datenbaustein ab.

Weiterhin werden die nötigen Variablentabellen in STEP 7 zum Test zur Verfügung gestellt.

Anforderung an die PC Station

Die PC Station muss die nötige physikalische Anbindung mit entsprechender Hardware und Software zur Kommunikation mit der Steuerung enthalten. Die Applikation zur Visualisierung und Steuerung soll ausschließlich die OPC UA Schnittstelle nutzen, um beliebige OPC UA Server verwenden zu können.

Das Applikationsbeispiel soll zeigen, was generell auf der Server-/Client PC-Station und den S7-Controllern projektiert werden muss, um die Kommunikationsaufgabe zu lösen.

Der SIMATIC NET OPC Server wird in STEP 7, in der Konfigurations-Konsole für die PC-Station und in den entsprechenden Konfigurationsdateien für die Aufgabe konfiguriert (Protokoll, Sicherheitseinstellungen, Zertifikate, etc.).

Das zugrunde liegende S7-Protokoll und die notwendigen Verbindungen zu den Controllern werden projektiert; inklusive der entsprechenden Schritte, die auf dem Server PC für den OPC UA Betrieb zu projektieren und parametrieren sind.

Es wird eine sichere Kommunikation zwischen Client PC und Server PC mit OPC UA Mitteln unter Windows hergestellt.

Copyright © Siemens AG 2017 All rights reserved

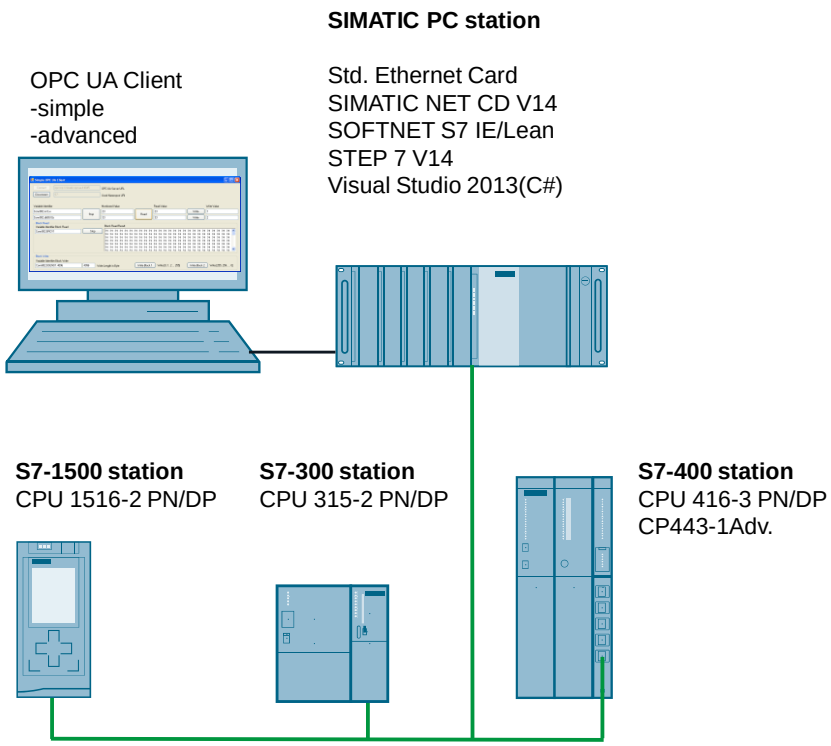
2 Automatisierungslösung

2.1 Übersicht Gesamtlösung

Übersicht

Die folgende Abbildung zeigt schematisch die wichtigsten Komponenten der Lösung:

Abbildung 2-1



Aufbau

Eine PC Station wird über Ethernet mit einer CPU 315-2 PN und einer CPU 414-2 verbunden. Im PC wird eine Standard Ethernetkarte verwendet.

OPC-UA Client Software

Der OPC-UA Client in der PC Station ist in zwei Komplexitätsstufen realisiert. Ein sehr einfach gestalteter Client (**Simple OPC UA Client**) zeigt ihnen alle Basisfunktionen für den Einstieg in OPC UA. Ein komplexerer Client (**OPC UA .NET Client**) mit komfortabler Oberfläche demonstriert Ihnen den professionellen Umgang mit wieder verwendbaren Klassen.

Die Funktionalität dieser Beispiel-Clients wird im nächsten Abschnitt erläutert.

Copyright © Siemens AG 2017 All rights reserved

2.2 Beschreibung der Kernfunktionalität

Übersicht

Den Kern der Funktionalität dieses Beispiels bildet der SIMATIC NET OPC UA Server. Dieser vereint die Funktionen und Informationen der klassischen OPC Server für Data Access und Alarm & Events in einem einzigen Namensraum und erlaubt den Zugriff auf die Informationen über eine serviceorientierte Architektur. Dabei ist die Kommunikation auch über das Internet und Firewalls hinweg sicher und performant möglich.

Die folgende Abbildung zeigt die Wirkungskette für einen Datenzugriff.

Abbildung 2-2

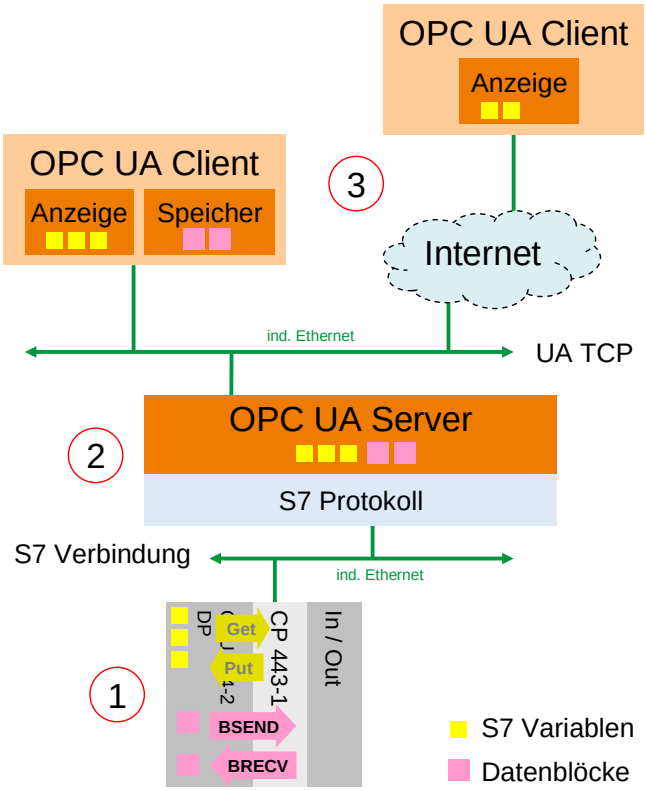


Tabelle 2-1

Nr.	Komponente	Beschreibung
1.	S7 Station	Die S7 CPU stellt in Datenbereichen wie Merkern oder Datenbausteinen S7 Variablen zur Verfügung. Über die blockorientierten Dienste BSEND und BRECV können auch größere Datenblöcke aktiv aus dem Anwenderprogramm gesendet und empfangen werden.
2.	OPC UA Server	Der OPC UA Server setzt die S7 Variablen und die Blockdienste auf OPC UA Variablen um und stellt die OPC Dienste wie Browse, Read, Write und Data Monitoring zur Verfügung.
3.	OPC UA Client	Der OPC UA Client kann eine sichere Verbindung zum Server aufbauen, durch den Namensraum des Servers navigieren und ausgewählte Variablen lesen, schreiben und beobachten.

Copyright © Siemens AG 2017 All rights reserved

Softwarekomponenten der Applikation (OPC UA .NET Client)

Die folgende Abbildung zeigt die Softwarekomponenten, welche bei der komplexeren Applikation (OPC UA .NET Client) verwendet werden. Dabei stammen der OPC UA Server und die Basisbibliotheken für die OPC UA Kommunikation auf der Clientseite von der SIMATIC NET CD.

Die für die Applikation in C# erstellten Softwarekomponenten können in wieder verwendbare Module und Beispielcode aufgeteilt werden.

Abbildung 2-3

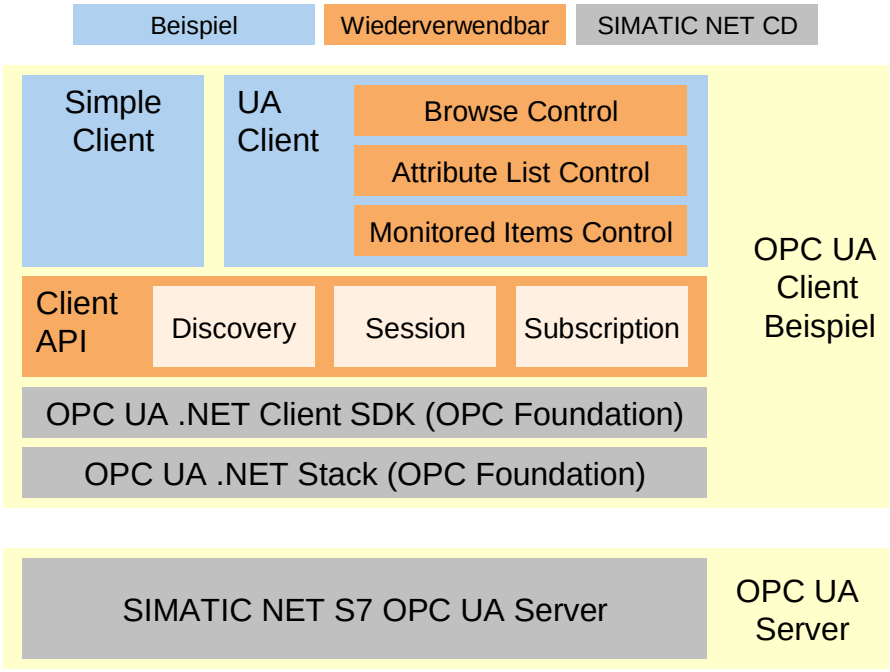


Tabelle 2-2

Modul	Beschreibung
OPC UA .NET Stack	Der .NET OPC UA Stack von der OPC Foundation für die Realisierung der Netzwerkkommunikation.
.NET Client SDK	Das .NET OPC UA Client SDK der OPC Foundation. Die beiden DLLs der OPC Foundation sind Lieferbestandteil der SIMATIC NET CD.
Client API	Wiederverwendbare, vereinfachte und auf die Aufgabe zugeschnittene .NET Client API. Sie bietet für Discovery, Session und Subscription Handling wiederverwendbare C# Klassen an.
Simple Client	Einfaches Beispiel für die Verwendung der Client API mit den Funktionen Connect, Disconnect, Read, Write und Data Monitoring. Dieses Beispiel zeigt auch die direkte Adressierung und das Handling von Namespaces.
UA Client	Komfortabler OPC UA Client mit den Funktionen Discovery, Connect, Disconnect, Browse, Read aller Attribute, Write und Data Monitoring. Allgemeine Funktionen wie Browse, Attribute Auflisten und das Monitoring von Datenvariablen sind in wiederverwendbaren Controls gekapselt. In diesem Beispiel können die symbolischen Variablen gebrowsed und direkt aus dem Browser heraus verwendet werden.
S7 OPC UA Server	Der SIMATIC NET OPC UA Server implementiert die notwendige Serverlogik für Sessions und Subscriptions und die Datenanbindung an die S7 Stationen.

Copyright © Siemens AG 2017 All rights reserved

Oberfläche des einfachen Beispiels (Simple OPC UA Client)

Die Oberfläche des einfachen OPC UA Clients (Simple OPC UA Client) wird über Buttons für die einzelnen Funktionen bedient. Das einfache Beispiel zeigt die Verwendung der direkten Adressierung von S7 Variablen.

Abbildung 2-4

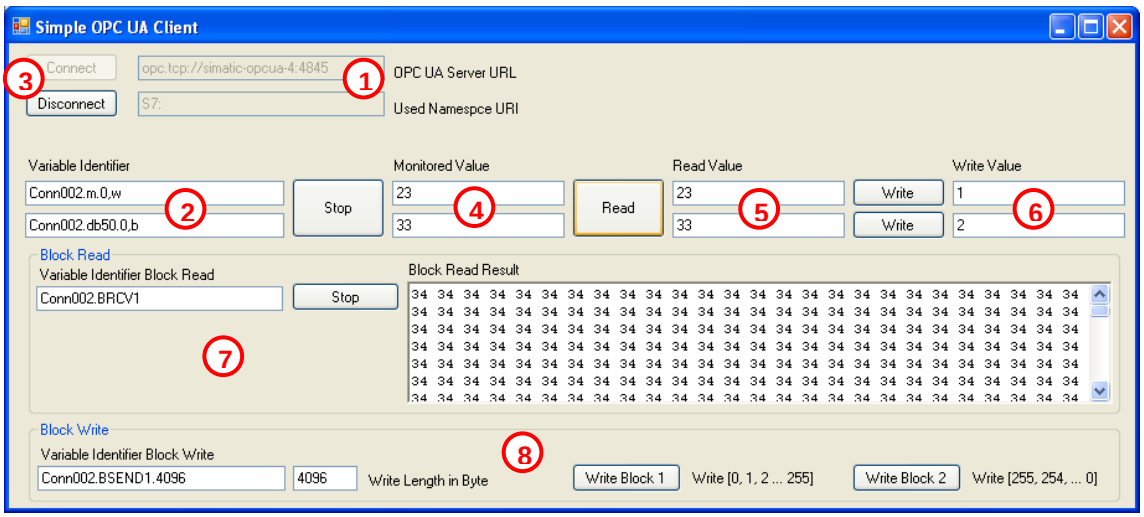


Tabelle 2-3

Nr.	Beschreibung
1.	In der Textbox für die **Server URL** kann die Server URL angegeben werden. Diese setzt sich für den SIMATIC NET OPC Server zusammen aus **opc.tcp://<Rechnername>:4845**. In der Textbox **Namespace URI** wird der verwendete Namespace angegeben. Dies ist **S7:** für direkte Adressierung, **S7COM:** für direkte Adressierung über die OPC DA kompatible Syntax und **SYM:** für symbolische Adressierung.
2.	In den Textboxen für die **Variable Identifier** wird der Kennungsteil der **NodeId** angegeben. Diese setzt sich beim Namespace **S7:** zusammen aus **<S7Verbindung>.<Datenbereich>.<Offset>,<Datentyp>** Die NodeId zum Lesen und Schreiben wird aus der Kennung und dem Namespace gebildet.
3.	Über die Buttons **Connect** und **Disconnect** kann die Verbindung zum OPC UA Server auf- und abgebaut werden. Die Verbindung wird nur ohne Sicherheit aufgebaut. Der sichere Verbindungsaufbau wird im nächsten Beispiel erläutert.
4.	Über den Button **Monitor** wird eine Subscription angelegt und mit den beiden **NodeIds** werden zwei **Monitored Items** in der **Subscription** angelegt. Die Datenänderungen werden in den Textboxen neben dem Button angezeigt. Fehler werden jeweils statt der Werte angezeigt.
5.	Über den Button **Read** werden die Werte (Value Attribut) der beiden Variablen mit den angegebenen **NodeIds** gelesen und in die Textboxen neben dem Button angezeigt.
6.	Über den Button **Write** wird der Wert aus der Textbox neben dem Button auf die durch die **NodeId** identifizierte Variable geschrieben. Für das Schreiben muss erst Read aufgerufen werden, da der Text aus der Textbox in den passenden Datentyp der Variable konvertiert werden muss. Die Konvertierung geschieht auf Basis des Datentyps, der bei Read geliefert wird.
7.	In der Gruppe "**Block Read**" können Daten empfangen werden, die von der S7 mit dem Blockdienst **BSEND** aktiv geschickt werden. Dies kann z.B. für das Senden von Ergebnisdaten von der S7 an eine PC Applikation verwendet werden.
8.	In der Gruppe "**Block Write**" können Datenblöcke an die S7 geschickt werden, die dort mit dem Blockdienst **BRECV** empfangen werden. Es können zwei Blöcke mit unterschiedlichen Inhalten geschickt werden. Dies kann z.B. für den Download von Rezeptdaten zur S7 verwendet werden.

Copyright © Siemens AG 2017 All rights reserved

Übersicht und Beschreibung der komfortablen Oberfläche (OPC UA .NET Client)

Die folgende Abbildung und Tabelle beschreiben die Oberfläche des generischen OPC UA Client Beispiels, mit dem auf die Informationen des Namensraums eines OPC UA Servers komfortabel zugegriffen werden kann.

Die Oberfläche erlaubt auch das Browsen der symbolischen S7 Variablen.

Abbildung 2-5

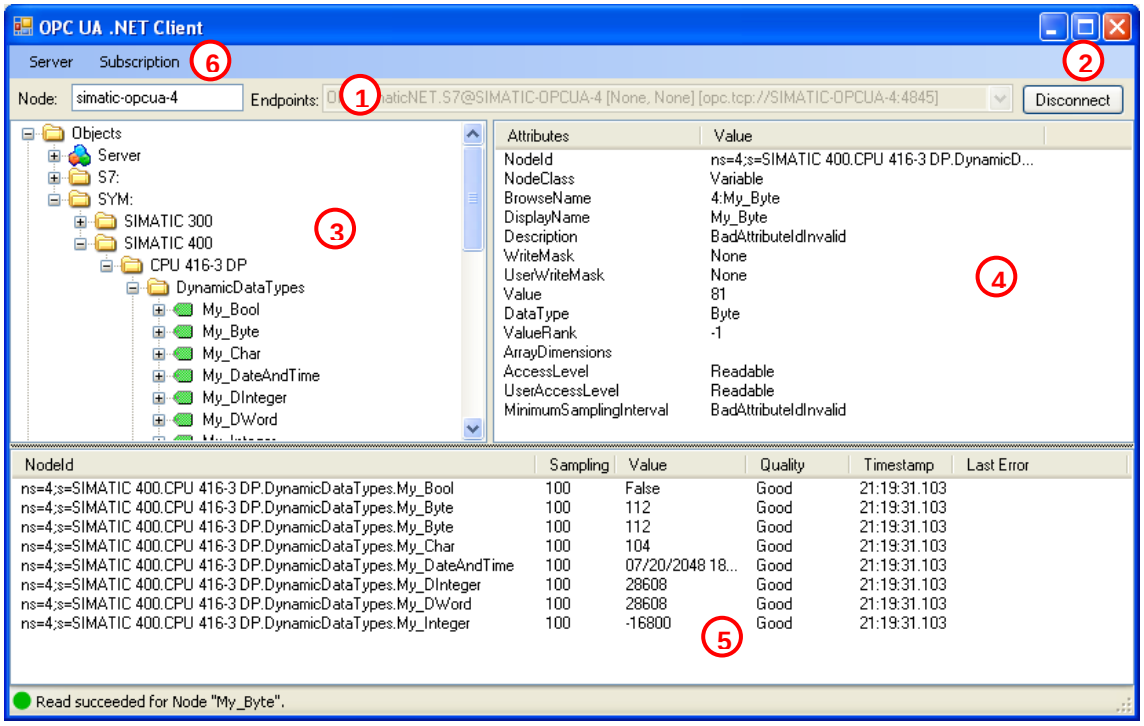


Tabelle 2-4

Nr.	Beschreibung
1.	Über die Auswahlliste **Endpoints** kann der Server ausgewählt werden. Dafür wird die Liste der verfügbaren OPC UA Server vom jeweiligen Netzwerkknoten ermittelt. Der Rechner, auf dem die Liste abgefragt werden soll, kann im Textfeld **Node** eingegeben werden. Ist das Feld leer, wird die Liste auf dem lokalen Rechner ermittelt. Die URL des OPC UA Servers kann auch manuell eingegeben werden. Die URL setzt sich für den SIMATIC NET OPC UA Server zusammen aus **opc.tcp://<Rechnername>:55101**.
2.	Über den Button **Connect** kann die Verbindung zum Server auf- bzw. abgebaut werden.
3.	Im Browse Control wird der gesamte Adressraum des verbundenen Servers in einer hierarchischen Baumansicht angezeigt. Es werden nur hierarchische Referenzen angezeigt.
4.	Für den selektierten Knoten im Browse Control werden die Attribute gelesen und in diesem Control angezeigt.
5.	Mit Drag&Drop können Variablen aus dem Browse Control in das Beobachtungsfenster gezogen werden. Für die Variable wird dann die NodeId, das Sampling Intervall, der Wert, der Zeitstempel und der Status Code angezeigt.
6.	Über das Kontextmenü im Beobachtungsfenster oder über das Applikationsmenü können die Eigenschaften der Subscription und Monitored Items verändert werden. So kann z.B. das Sampling Intervall geändert werden. Außerdem kann der Dialog zum Schreiben geöffnet werden. Dabei werden die Variablen in den Dialog übernommen, die im Beobachtungsfenster markiert sind.

Copyright © Siemens AG 2017 All rights reserved

Vorteile dieser Lösung

Die hier vorgestellte Lösung bietet Ihnen folgende Vorteile:

- ein einfacher Einstieg in die OPC UA Technologie
- Programmierung in C# für .NET
- einfache Erweiterbarkeit des Beispiels
- wieder verwendbare Komponenten
- Zugriff über das Internet und durch Firewalls möglich
- Zugriffsrechte können individuell für Benutzer vergeben werden
- Umgang mit Zertifikaten, Verschlüsselung und Authentifizierung
- Demonstration der S7 Kommunikation

Abgrenzung

Diese Applikation enthält keine Beschreibung zur Verarbeitung oder Speicherung der Daten im OPC UA Client z.B. in Datenbanken.

Vorausgesetzte Kenntnisse

Grundlegende Kenntnisse über den Umgang mit dem SIMATIC Konfigurations- und Programmiertool STEP7 sowie der Entwicklungsumgebung Microsoft Visual Studio 2008 und der Programmiersprache C# und objektorientierter Programmierung werden voraus gesetzt.

Copyright © Siemens AG 2017 All rights reserved

2.3 Verwendete Hard- und Software-Komponenten

Die Applikation wurde mit den nachfolgenden Komponenten erstellt:

Hardwarekomponenten

Tabelle 2-5

Komponente	Anz.	Artikelnummer	Hinweis
S7-400 CPU 416-3 PN/DP	1	6ES7416-3XR05-0AB0	Alternativ kann auch jede andere S7-400 mit einem PNIO-Interface verwendet werden.
CP 443-1 Advanced	1	6GK7443-1GX20-0XE0	(Optional) Alternativ kann auch jeder andere S7-fähige Ethernet CP verwendet werden.
S7-300 CPU 315-2 PN/DP	1	6ES7 315-2EH14-0AB0	Alternativ kann auch jede andere S7-300 mit einem PNIO-Interface verwendet werden.
CP 343-1 Advanced-IT	1	6GK7 343-1GX31-0XE0	(Optional) Alternativ kann auch jeder andere S7-fähige Ethernet CP verwendet werden.
S7-1500 CPU 1516-3 PN/DP	1	6ES7 516-3AN00-0AB0	Alternativ kann auch jede andere S7-1500 verwendet werden.
Standard PC als OPC UA Server	1	-	Standard-PC (z. B. PGs) mit Windows 7/8/10.

Komponente	Anz.	Artikelnummer	Hinweis
Standard PC als OPC UA Client	1	-	Alternativ kann der Client auch lokal auf dem PC betrieben werden.

Softwarekomponenten

Tabelle 2-6

Komponente	Anz.	Artikelnummer	Hinweis
SIMATIC NET DVD V14	1	6GK1700-0AA14-0AA0	-
STEP 7 Professional V14 SP1	1	6ES7822-1AA04-0YA5	-
Microsoft Visual Studio 2010	1		Alternativ sind auch neuere Versionen von VS möglich.
.NET Framework 4.5	1	-	-

Beispieldateien und Projekte

Die folgende Liste enthält alle Dateien und Projekte, die in diesem Beispiel verwendet werden.

Tabelle 2-7

Komponente	Hinweis
42014088_OPC_UAClient_DOKU_V12_de.pdf	Dieses Dokument.
42014088_OPC_UAClient_CODE_V12.zip	Diese gepackte Datei enthält den OPC UA-Client mit Quellen und das zugehörige STEP 7 V14-Projekt.

Copyright © Siemens AG 2017 All rights reserved

2.4 Alternativlösungen

OPC Data Access auf Basis von COM

Die Automatisierungsaufgabe wird heute typischerweise mit der COM basierten klassischen OPC Data Access Schnittstelle gelöst.

Vorteile der Lösung mit COM OPC Data Access:

- Große Verbreitung der Schnittstelle.
- Viele Anwendungen für verschiedene Aufgaben unterstützen die Schnittstelle.
- Einfacher Zugriff für lokale Anwendungen.

Nachteile der Lösung mit COM OPC Data Access:

- Komplizierte DCOM Konfiguration für remoten Zugriff.
- Keine Kommunikation über Firewall- oder Internetgrenzen möglich.
- OPC Clients können nur auf Windows PC Systemen betrieben werden.
- Eingeschränkte Sicherheitsmechanismen und Benutzer-Authentifizierung nur im Rahmen der DCOM Konfiguration.
- Keine benutzerbezogenen Zugriffsrechte möglich.

3 Grundlagen

3.1 Grundlagen OPC

Übersicht

Die OPC Foundation (eine Interessenvereinigung namhafter Hersteller für die Definition von Standardschnittstellen) hat in den letzten Jahren eine Vielzahl von Softwareschnittstellen definiert, um den Informationsfluss von der Prozessebene bis zur Managementebene zu vereinheitlichen. Entsprechend der unterschiedlichen Anforderungen innerhalb einer industriellen Anwendung sind in der Vergangenheit verschiedene OPC Spezifikationen entstanden: Data Access (DA), Alarm & Events (A&E), Historical Data Access (HDA) und Data eXchange (DX). Der Zugriff auf Prozessdaten ist in der DA Spezifikation beschrieben, A&E beschreibt eine Schnittstelle für ereignisbasierte Information inklusive Quittierung, HDA beschreibt Funktionen für archivierte Daten und DX definiert eine Server zu Server Querkommunikation.

Basierend auf den Erfahrungen dieser klassischen OPC Schnittstellen hat die OPC Foundation eine neue Plattform mit dem Namen OPC Unified Architecture (UA) definiert. Ziel dieses neuen Standards ist die generische Beschreibung und der einheitliche Zugriff auf alle Informationen die zwischen Systemen bzw. Applikationen ausgetauscht werden müssen. Das schließt die Funktionalität aller bisherigen OPC Schnittstellen mit ein. Außerdem soll die Möglichkeit geschaffen werden, die Schnittstelle nativ in das jeweilige System zu integrieren, unabhängig davon, auf welchem Betriebssystem das System betrieben wird und unabhängig davon, mit welcher Programmiersprache das System erstellt ist.

Dieses Beispiel beschäftigt sich mit der Schnittstelle OPC Unified Architecture. Eine detaillierte Dokumentation befindet sich auf der SIMATIC NET CD. Weitere Informationen sind unter www.opcfoundation.org zu finden.

Was ist OPC

OPC war in der Vergangenheit eine Sammlung von Softwareschnittstellen zum Datenaustausch zwischen PC Anwendungen und Prozessgeräten. Diese Softwareschnittstellen waren entsprechend den Regeln von Microsoft COM (Component Object Model) definiert und damit auf Microsoft Betriebssystemen leicht integrierbar. COM bzw. DCOM (Distributed COM) stellt die Funktionalität der Interprozesskommunikation zur Verfügung und organisiert den Informationsaustausch zwischen Anwendungen, auch über Rechnergrenzen hinweg (DCOM). Ein OPC Client (COM Client) kann damit unter Verwendung von Mechanismen des Microsoft Betriebssystems Informationen mit einem OPC Server (COM Server) austauschen.

Der OPC Server stellt Prozessinformationen eines Gerätes an seiner Schnittstelle zur Verfügung. Der OPC Client verbindet sich mit dem Server und kann auf die angebotenen Daten zugreifen.

Die Verwendung von COM bzw. DCOM führt dazu, dass OPC Server und Clients nur auf einem Windows PC oder im lokalen Netzwerk betrieben werden können und diese die Kommunikation zum entsprechenden Automatisierungssystem meist über proprietäre Protokolle realisieren müssen. Für die Netzwerkkommunikation zwischen Client und Server müssen oft zusätzliche Tunneling Tools eingesetzt werden, um durch Firewalls zu kommen oder um die komplizierte DCOM Konfiguration zu umgehen. Außerdem kann nur mit C++ Applikationen nativ auf die Schnittstelle zugegriffen werden, .NET oder JAVA Applikationen können nur über eine Wrapperschicht zugreifen. Diese Einschränkungen führen in der Praxis zu

Copyright © Siemens AG 2017 All rights reserved

zusätzlichen Kommunikations- und Softwareschichten, welche den Konfigurationsaufwand und die Komplexität erhöhen.

Durch die große Verbreitung von OPC wird der Standard immer mehr zur generellen Kopplung von Automatisierungssystemen eingesetzt und nicht mehr nur für den ursprünglichen Anwendungsfall als Treiberschnittstelle in HMI und SCADA Systemen für den Zugriff auf Prozessinformationen.

Um die genannten Einschränkungen in der Praxis zu lösen und den zusätzlichen Anforderungen gerecht zu werden, hat die OPC Foundation in den letzten fünf Jahren eine neue Plattform mit dem Namen OPC Unified Architecture definiert, welche eine einheitliche Basis für den Informationsaustausch zwischen Komponenten und Systemen bietet. OPC UA wird auch als IEC 62541 Standard verfügbar sein und bildet damit auch die Basis für andere internationale Standards.

OPC UA bietet die folgenden Features:

- Zusammenfassung aller bisherigen OPC Features und Informationen wie DA, A&E und HDA in einer generischen Schnittstelle.
- Verwendung offener und plattformunabhängiger Protokolle für die Interprozess- bzw. Netzwerkkommunikation.
- Internetzugriff und Kommunikation durch Firewalls.
- Integrierte Zugriffskontrolle und Sicherheitsmechanismen auf Protokoll- und Applikationsebene.
- Umfangreiche Abbildungsmöglichkeiten für objektorientierte Modelle, Objekte können Variablen und Methoden haben und Events feuern.
- Erweiterbares Typsystem für Objekte und komplexe Datentypen.
- Transportmechanismen und Modellierungsregeln bilden die Basis für andere Standards.
- Skalierbarkeit von kleinen embedded Systemen bis hin zu Unternehmensanwendungen und von einfachen DA Adressräumen bis hin zu komplexen, objektorientierten Modellen.

Copyright © Siemens AG 2017 All rights reserved

3.2 Grundlagen von OPC Unified Architecture

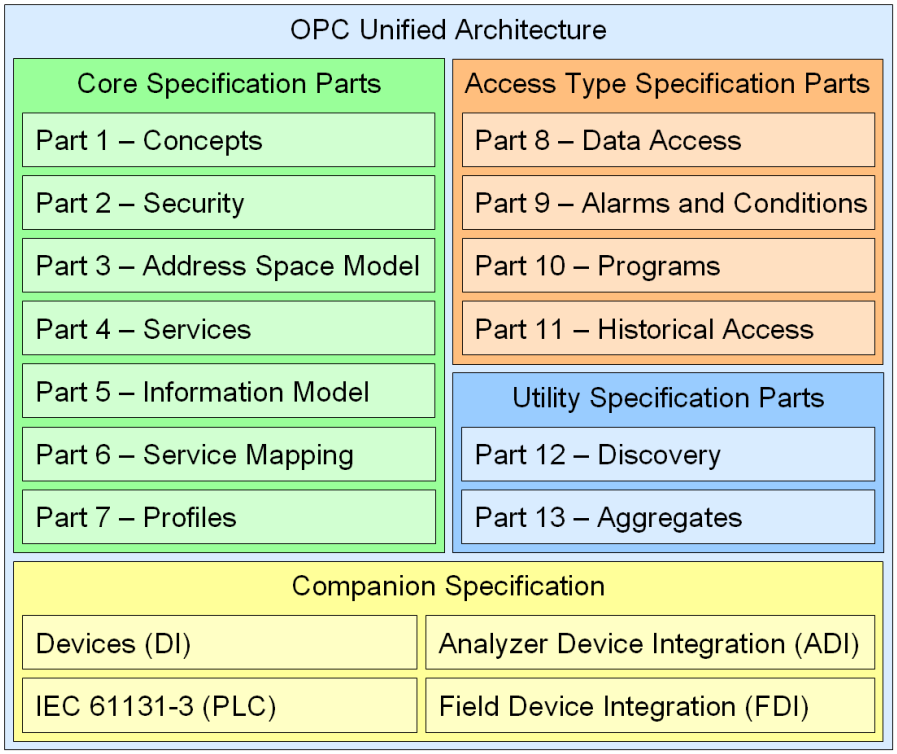
In diesem Kapitel werden die für das Beispiel notwendigen Grundlagen von OPC Unified Architecture erläutert.

3.2.1 OPC UA Spezifikationen

Übersicht

Die OPC UA Spezifikationen sind auf Grund der IEC 62541 Standardisierung in verschiedene Parts aufgeteilt. Abbildung 3-1 zeigt die verschiedenen Parts im Überblick.

Abbildung 3-1



Part 1 bis 7 bilden die Basis der Technologie und die Realisierung von OPC UA Applikationen. Vor allem die Parts 3 bis 5 bilden den Kern des Standards.

Die Parts 8 bis 11 definieren OPC spezifische Informationsmodelle zur Bereitstellung klassischer OPC Informationen wie aktuelle Prozessdaten oder Alarme.

Ergänzende Hilfsmittel sind in Part 12 und 13 definiert.

Darüber hinaus entstehen sogenannte Companion Spezifikationen, die in Zusammenarbeit mit anderen Standardisierungsorganisationen zusätzliche Informationsmodelle auf Basis von OPC UA definieren. Dabei bilden die Modelle und Informationen in anderen Standards die Grundlage und die Companion Spezifikation definiert, wie diese Informationen mit OPC UA beschrieben und transportiert werden.

Hinweis

Relevant für diese Applikation sind die Teile drei bis fünf und Teil acht. Die Beschreibung der anderen Teile ist für die Vermittlung eines kompletten Überblicks über die OPC Unified Architecture enthalten.

Copyright © Siemens AG 2017 All rights reserved

Liste der Spezifikationen

Tabelle 3-1 erläutert die Liste der Spezifikation und deren Inhalte. Die für den SIMATIC NET Server aktuell relevanten Spezifikationen sind hervorgehoben.

Tabelle 3-1

Spezifikation	Beschreibung
Part 1 – Concepts	Dieser nicht normative Teil gibt einen Überblick über den Standard.
Part 2 – Security	Die Anforderungen an Security und eine Einführung in die Grundlagen ist in dem ebenfalls nicht normativen zweiten Teil beschrieben.
Part 3 – Address Space Model	Dieser Teil definiert die Grundregeln und Grundelemente für den Aufbau des Adressraumes eines OPC UA Servers. Diese Regeln bilden die Basis für die Informationsmodelle in Part 5, 8 bis 11 und den Companion Spezifikationen.
Part 4 – Services	Dieses Dokument definiert als einziger Teil die Schnittstelle für den Zugriff auf alle OPC UA Informationen. Er spezifiziert eine Liste von Methoden, die sogenannten Services. Diese Services sind generisch und bilden die Basis für alle Informationsmodelle.
Part 5 – Information Model	Das Basis Informationsmodell definiert die Einstiegspunkte in den Adressraum und grundlegende Typen wie z.B. Datentypen oder Objekttypen. Der Teil bildet zusammen mit Part 3 und 4 den Kern von OPC UA.
Part 6 – Service Mapping	Die Services in Part 4 sind unabhängig vom verwendeten Transport-mechanismus definiert. Dieser Teil spezifiziert die Umsetzung der Services auf verschiedene Arten von Serialisierung, Sicherung und Transport-protokollen für die Nachrichten zwischen OPC UA Client und Server. Dieser Teil bildet die Basis für die Implementierung von Kommunikationsstacks und ist für Anwender der Technologie nicht relevant.
Part 7 – Profiles	Ein Profil legt eine Untermenge von OPC Funktionalitäten für verschiedene Anwendungsfälle fest, die von einem OPC UA Server angeboten bzw. von einem OPC UA Client genutzt werden können. Dieser Teil definiert die Liste der Profile für OPC UA.
Part 8 – Data Access	Dieser Teil definiert Variablentypen, Properties und Quality Status Codes für Prozessdaten. Alle anderen notwendigen Konzepte sind bereits in den Parts 3 bis 5 enthalten.
Part 9 – Alarms and Conditions	Dieser Teil definiert das Modell für die Beschreibung von Zustandsüberwachungen und Prozessalarmen und der Meldung von Statuswechseln über Events. Die notwendigen Konzepte für Events sind bereits in den Parts 3 bis 5 enthalten.
Part 10 – Programs	Dieser Teil definiert wie länger laufende Aktionen gestartet und überwacht werden können. Dies geschieht auf Basis von Zustandsmaschinen, deren Handhabung in OPC UA in Part 5 definiert ist.
Part 11 – Historical Access	Hier wird der Zugriff auf historische Daten und Events definiert.
Part 12 – Discovery	Definiert wie OPC UA Server im Netzwerk gefunden werden können.
Part 13 – Aggregates	Dieser Teil definiert Aggregatsfunktionen für die Datenkompression wie z.B. Mittelwert oder Maximalwert über ein Zeitbereich. Die Aggregate können auf aktuelle Daten oder auf historische Daten angewendet werden.
Devices (DI)	Diese Companion Spezifikation definiert ein generisches Modell für die Konfiguration und Diagnose von Geräten.
IEC 61131-3 (PLC)	Diese Companion Spezifikation definiert ein Mapping des IEC 61131-3 Softwaremodells und der standardisierten Steuerungsprogrammiersprachen auf einen OPC UA Server Adressraum.
Analyzer Device Integration (ADI)	Diese Companion Spezifikation definiert ein Modell für die Konfiguration und Datenanbindung für komplexe Geräte zur Prozessanalyse aufsetzend auf DI
Field Device Integration (FDI)	Diese Companion Spezifikation definiert ein Modell für das komplette Engineering von Feldgeräten auf Basis von Electronic Device Description Language (EDDL) und Field Device Tool (FDT).

Copyright © Siemens AG 2017 All rights reserved

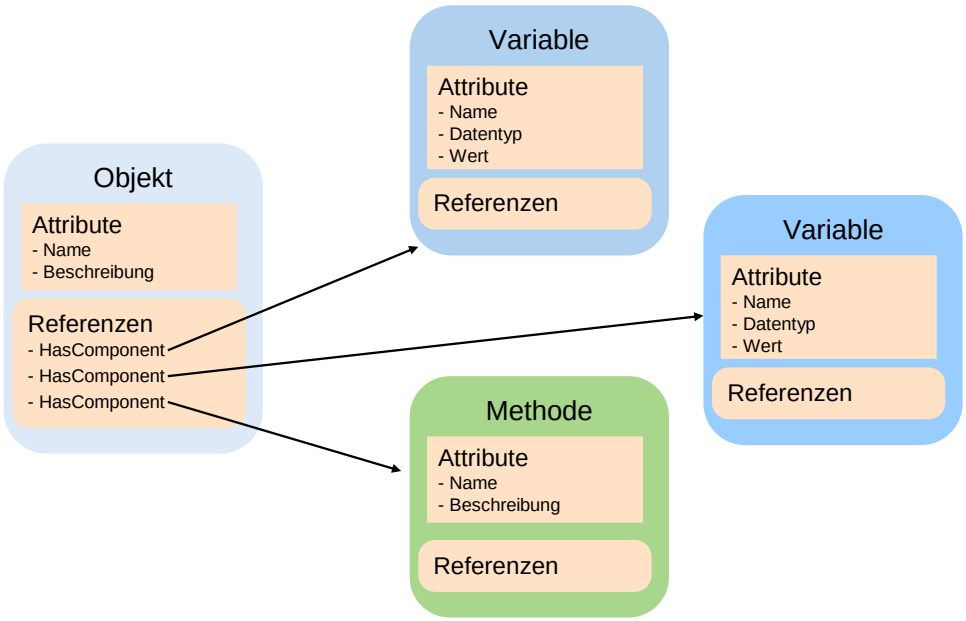
3.2.2 Aufbau des OPC UA Server Adressraums

Knoten im Adressraum

Ein Knoten im OPC UA Adressraum ist von einem bestimmten Typ wie z.B. Objekt, Variable oder Methode und wird durch eine Liste von Attributen beschrieben. Alle Knoten haben gemeinsame Attribute wie Name oder Beschreibung und spezifische Attribute wie z.B. den Wert einer Variablen. Die Liste der Attribute ist nicht erweiterbar. Zusätzliche Informationen über den Knoten können als Property ergänzt werden. Properties sind eine spezielle Art von Variablen.

Die Knoten sind mit Referenzen untereinander verbunden. Die Referenzen sind typisiert. Es gibt zwei Hauptgruppen, hierarchische Referenzen wie z.B. HasComponent für die Komponenten eines Objektes oder nicht-hierarchische Referenzen wie z.B. HasTypeDefinition für eine Verbindung von einer Objektinstanz zu einem Objekttyp. Ein Beispiel für Knoten und die verbindenden Referenzen bietet Abbildung 3-2.

Abbildung 3-2



Verfügbare Typen von Knoten im Adressraum

Die definierten Knotentypen sind in Tabelle 3-2 aufgeführt. Die Liste der Typen ist nicht erweiterbar.

Tabelle 3-2

Knotentyp	Beschreibung	Beispiel
Objekt	Ein Objekt dient als typisierte Container für Variablen, Methoden und Events.	Die Objekte, die eine S7 Verbindung repräsentieren, sind immer gleich aufgebaut.
Variable	Variablen repräsentieren die Daten von Objekten oder als Property die Eigenschaften eines Knotens.	S7 Variable in einem Datenbaustein.
Methode	Methoden sind Komponenten von Objekten und können eine Liste von Eingangs- bzw. Ausgangsparametern haben. Die Parameter	Methode blockRead( ) an einem S7 Verbindungsobjekt mit der ein Baustein aus einer S7 ausgelesen

Copyright © Siemens AG 2017 All rights reserved

Knotentyp	Beschreibung	Beispiel
	werden über definierte Properties beschrieben.	werden kann.
View	Views repräsentieren einen Teil des Adressraumes. Der Knoten dient als Einstiegspunkt und als Filter beim Browsen.	Views sind im SIMATIC NET Server nicht vorhanden.
Objekttyp	Objekttypen liefern Informationen über den Aufbau bzw. die Komponenten eines Objektes.	S7ConnectionType beschreibt die Komponenten die an einem S7 Verbindungsobjekt vorhanden sind.
Variablen Typ	VariablenTypen beschreiben typischerweise, welche Properties bzw. Datentypen an einer Instanz des Typs (Variable) gefunden werden können.	AnalogItemType definiert, dass eine Variable dieses Types die Properties EngineeringUnits und EURange zur Verfügung stellt.
Referenz Typ	Referenz Typen definieren die möglichen Arten von Referenzen zwischen Knoten.	Eine Methode wird von einem Objekt mit HasComponent referenziert.
Datentyp	Datentypen beschreiben den Inhalt des Wertes einer Variablen.	Der Wert einer Variablen kann den Datentyp Double haben.

Strukturierung des Adressraums

Die Grundstruktur des OPC UA Adressraums ist in Part 5 definiert. Abbildung 3-3 zeigt einen Teil dieser Struktur und SIMATIC NET spezifische Teile. Die verschiedenen Bereiche sind in Tabelle 3-3 beschrieben.

Abbildung 3-3

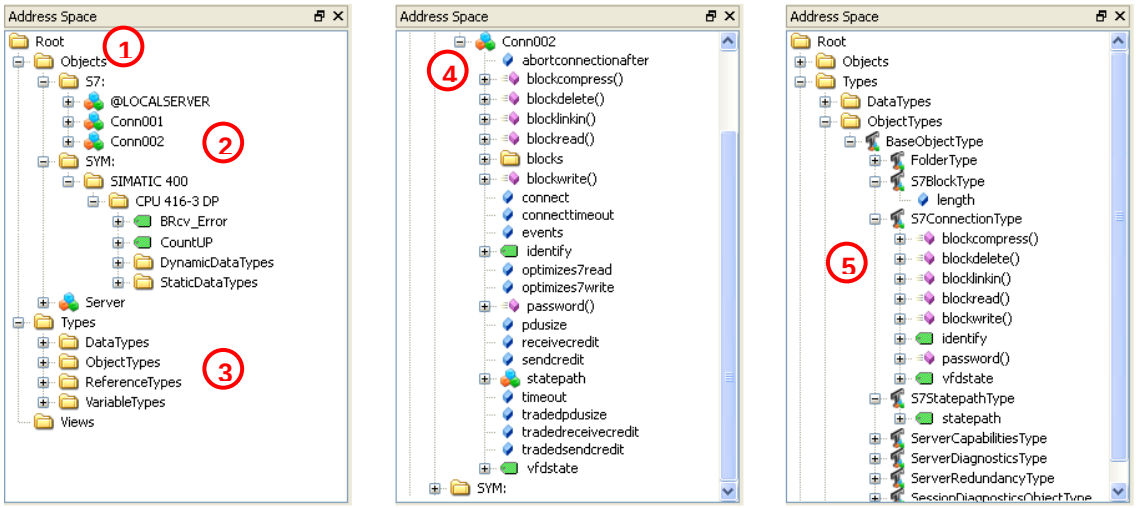


Tabelle 3-3

Nr.	Beschreibung
1.	Im Verzeichnis Objects sind die Instanzen wie Objekte und Variablen zu finden. Ein Data Access Client findet unter diesem Verzeichnis die Variablen für den Datenzugriff. Neben dem SIMATIC NET spezifischen Verzeichnissen findet man hier auch das Server Objekt, welches von OPC UA definiert ist. Es enthält Informationen über Funktionsumfang und Status des Servers.
2.	Die zwei Verzeichnisse S7: und SYM: unter Objects sind spezifisch für den SIMATIC NET OPC UA Server. Unter S7: sind die Konfigurierten S7 Verbindungen als Objekte aufgelistet. Unter SYM: sind die Symbole aus dem STEP 7 Projekt zu finden.
3.	Im Verzeichnis Types befinden sich die verschiedenen Typknoten für DataTypes, ObjectTypes, ReferenceTypes und VariableTypes.

Copyright © Siemens AG 2017 All rights reserved

Nr.	Beschreibung
4.	Ein S7 Verbindungsobjekt stellt diverse Statusinformationen und Methoden zur Verfügung. So können z.B. Blöcke in der S7 über Methoden bearbeitet oder ausgelesen werden. Neben den Methoden liefern Properties Informationen über die Konfiguration der S7 Verbindung.
5.	Der zu dem S7 Verbindungsobjekt zugehörige S7ConnectionType kann im Verzeichnis ObjectTypes gefunden werden. Er beschreibt, welche Methoden und Variablen an der Instanz mindestens vorhanden sind. Die Regeln für das Typsystem sind in /2/ ausführlich beschrieben.

Namespaces und NodeId

Jeder Knoten im OPC UA Adressraum wird durch eine NodeId eindeutig identifiziert. Diese NodeId setzt sich zusammen aus einem Namespace zur Unterscheidung von Kennungen aus verschiedenen Subsystemen und einer Kennung, die entweder ein numerischer Wert, ein String oder eine GUID sein kann.

Für die Kennung werden typischerweise Strings verwendet. Dies ist analog zu OPC Data Access, wo die ItemId als Kennung auch ein String ist. Numerische Werte werden für statische Namensräume wie z.B. Typsystem eingesetzt.

OPC UA definiert einen Namespace für die von OPC definierten Knoten. Die OPC UA Server definieren zusätzlich einen oder mehrere Namespaces. Tabelle 3-6 listet die relevanten Namespaces für den SIMATIC NET OPC UA Server auf.

Tabelle 3-4

Namespace	Beschreibung
http://opcfoundation.org/UA/	Wird verwendet für Knoten, die in der OPC UA Part 5 definiert werden. Dies sind Knoten, welche die Grundstruktur des Adressraumes bilden und Knoten die von OPC UA definierten Typen repräsentieren.
S7:	Namensraum für direkte Adressierung von S7 Variablen mit einer optimierten Syntax.
S7COM:	Namensraum für Adressierung mit alten Bezeichnern. Der Namensraum ist S7-OPC-DA-kompatibel. Dieser Namensraum kann mit UA verwendet werden, ist jedoch nicht durchsuchbar.
S7AREAS:	Namensraum für Bereiche mit Alarmhierarchie.
SYM:	Namensraum für symbolische Adressierung von S7 Variablen. Die Symbolinformationen werden aus dem STEP 7 Projekt exportiert.

Copyright © Siemens AG 2017 All rights reserved

Attribute der Knoten

Die wichtigsten Attribute von Knoten sind exemplarisch in der folgenden Tabelle aufgelistet. Der Schwerpunkt liegt dabei auf dem Knotentyp Variable.

Tabelle 3-5

Attribute	Knotentyp	Beschreibung
NodeId	Alle	Eindeutige Adresse des Knotens.
DisplayName	Alle	Lokalisierter Anzeigename für den Knoten. Die Sprache hängt von der vom Client angefragten Sprache für die Verbindung und von den vom Server unterstützten Sprachen ab.
BrowseName	Alle	Nicht lokalisierter Name für den Knoten. Der Namen enthält eine Namespace und ist vor allem bei der Verwendung von Typen relevant.
NodeClass	Alle	Art des Knotens wie z.B. Objekt, Variable oder Methode.
Description	Alle (optional)	Optionale lokalisierte Beschreibung des Knotens.
Value	Variable	Wert der Variable. Zeitstempel und Status des Wertes werden wie bei allen anderen Attributen beim Lesen zusammen mit dem Wert des Attributs geliefert.
DataType	Variable	Datentyp der Variable bzw. des Value Attributes. Datentypen sind z.B. OPC UA definierte Datentypen wie Int32, Double oder String oder auch strukturierte Datentypen.
ValueRank	Variable	Gibt an, ob der Wert (Value Attribut) ein skalarer Wert, ein Array oder ein multidimensionales Array ist.
AccessLevel	Variable	Gibt an, ob die Variable gelesen oder geschrieben werden kann.

Copyright © Siemens AG 2017 All rights reserved

3.2.3 Schnittstelle für den Zugriff auf den OPC UA Server Adressraum

Kommunikationskanal und Applikationsobjekte

Abbildung 3-4 zeigt die verschiedenen Objekte, die beim Datenaustausch zwischen OPC Client und Server anlegt werden können. Die Objekte werden in Tabelle 3-6 beschrieben.

Abbildung 3-4

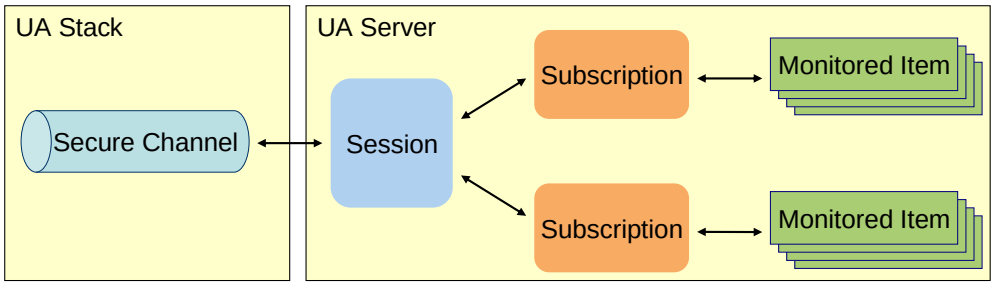


Tabelle 3-6

Objekt	Beschreibung
Secure Channel	Der gesicherte Kommunikationskanal wird im OPC UA Stack realisiert. Die Objekte auf Applikationsebene sind davon unabhängig lebensfähig. Sie können aber nur im Kontext eines Secure Channel erzeugt, verwendet oder verändert werden. Wird nach einer Verbindungsunterbrechung ein neuer Secure Channel aufgebaut, muss dieser der Session auf Applikationsebene zugeordnet werden.
Session	Die Session im Server ist die logische Verbindung zwischen OPC UA Client und Server. Sie hält die Benutzerinformationen und Spracheinstellungen für die Verbindung. Die Session wird vom Server gelöscht wenn innerhalb des Timeouts keine Aufrufe vom Client empfangen werden. Der Timeout wird vom Client vorgegeben. Die Session ist an einen Secure Channel gebunden, kann aber auf einen neuen Secure Channel zugewiesen werden, wenn die Kommunikation gestört war.
Subscription	Ein Subscription Objekt kann vom Client zur Gruppierung von Monitored Items erzeugt werden. Monitored Items dienen zur Beobachtung von Werteänderungen oder zum Empfangen von Ereignismeldungen. Die Subscription wird vom Server gelöscht, wenn innerhalb des Timeouts keine Daten oder KeepAlive Nachricht an den Client geschickt werden konnte. Der Timeout wird vom Client vorgegeben.

Copyright © Siemens AG 2017 All rights reserved

Methoden zum Verbindungsaufbau

Tabelle 3-7 erläutert die wichtigsten Methoden der OPC UA Schnittstelle zum Verbindungsaufbau.

Tabelle 3-7

Methode	Beschreibung
OpenSecureChannel	Öffnet einen sicheren Kommunikationskanal zwischen Client und Server. Zum Öffnen der Verbindung sind die URL des Servers, die Applikationszertifikate und die Sicherheitseinstellungen notwendig.
CreateSession	Anlegen einer Applikationssession im Kontext eines Secure Channels.
ActivateSession	Aktivieren der Session durch Übergabe der Benutzerauthentifizierung und der Spracheinstellungen. Diese Methode wird auch verwendet, um eine bestehende Session auf einen neuen Secure Channel zuzuweisen bzw. um den Benutzer zu wechseln.
CloseSession	Schließt die Applikationssession.

Methoden des Session Objektes

Tabelle 3-8 erläutert die wichtigsten Methoden der OPC UA Schnittstelle bezüglich der Session.

Tabelle 3-8

Methode	Beschreibung
Browse	Liefert die Liste von Knoten, die von einem Startknoten aus über eine Referenz erreichbar sind. Die Menge der Knoten kann durch Filter eingeschränkt werden. Für jeden Knoten werden Informationen geliefert, die z.B. für die Anzeige in einer Baumansicht notwendig sind.
Read	Liest eine Liste von Attributen von Knoten. Mit der Methode können sowohl Werte von Variablen (Value Attribut) als auch Metadaten wie z.B. der Datentyp einer Variablen (DataType Attribut) gelesen werden.
Write	Schreibt eine Liste von Attributen von Knoten. Mit dieser Methode werden typischerweise Werte von Variablen geschrieben. Wenn der Server dies erlaubt, können auch andere Attribute geschrieben werden.
CreateSubscription	Anlegen einer Subscription für das Empfangen von Datenänderungen oder Ereignismeldungen. Die Subscription dient der Gruppierung von Informationen, die beobachtet werden sollen. Alle neuen Daten bzw. Ereignisse werden als Paket in einstellbaren Zeitintervallen für eine Subscription geliefert.
DeleteSubscription	Löschen einer Subscription.

Methoden des Subscription Objektes

Tabelle 3-9 erläutert die wichtigsten Methoden der OPC UA Schnittstelle bezüglich der Subscription.

Tabelle 3-9

Methode	Beschreibung
ModifySubscription	Ändert die Einstellungen einer Subscription wie z.B. das Publish Intervall in der neue Daten für den Client gesammelt und zusammen geschickt werden.
CreateMonitoredItems	Anlegen einer Liste von Monitored Items in einer Subscription. Ein Monitored Item dient entweder zur Überwachung eines Wertes einer Variablen oder zur Überwachung von Ereignismeldungen. Beide Arten von Monitored Items können in einem Aufruf an diese Methode gemischt werden. In dieser Applikation werden nur Datenänderungen überwacht.
ModifyMonitoredItems	Ändert die Einstellungen einer Liste von MonitoredItems wie z.B. das Sampling Intervall für die Überwachung auf Werteänderungen.

Copyright © Siemens AG 2017 All rights reserved

Methode	Beschreibung
DeleteMonitoredItems	Löscht eine Liste von Monitored Items in einer Subscription.
Publish	Methode zur Übertragung von Datenpaketen für eine Subscription mit Werteänderungen und Ereignismeldungen im Publish Intervall. Diese Methode wird in der Client API nicht sichtbar. Die Funktionalität ist dort als Callback an die Client Applikation realisiert.

Copyright © Siemens AG 2017 All rights reserved

3.2.4 Protokolle und Sicherheitsmechanismen

OPC UA Kommunikationsarchitektur

Die Services für den Zugriff auf die Informationen in einem OPC UA Server Adressraum wie Browse, Read und Write sind abstrakt und unabhängig vom Transportprotokoll in Part 4 spezifiziert.

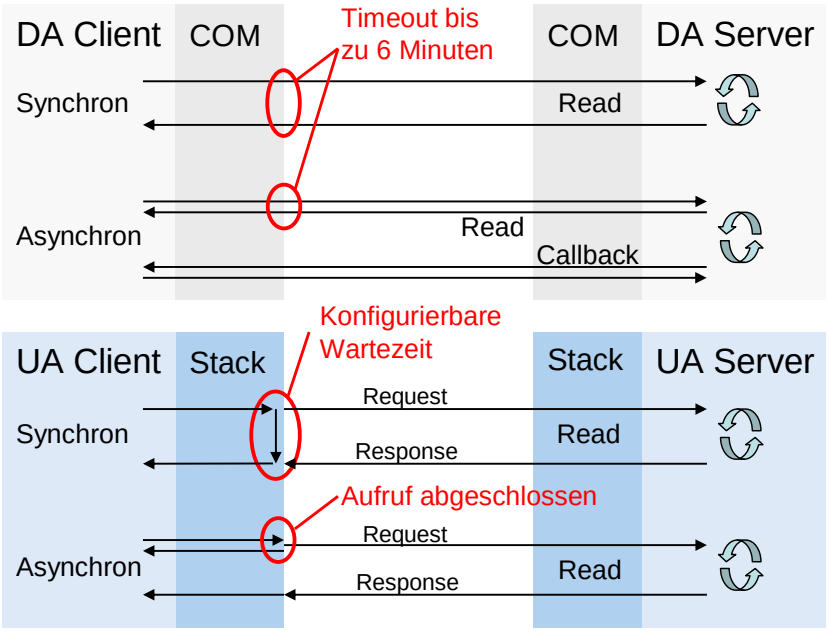
Die verschiedenen Bindings für die Übertragung der Service Nachrichten zwischen OPC UA Client und Server sind in Part 6 definiert. Ein Binding setzt sich aus Protokoll, Sicherheitsmechanismus und Serialisierungsart für die Daten zusammen.

Die Bindings werden in Kommunikationsstacks implementiert. Zurzeit gibt es von der OPC Foundation drei Implementierungen, nämlich in ANSI C, C# / .NET und JAVA. In dieser Applikation wird der C# / .NET Stack verwendet.

Die Methoden an der API des Stacks für die Applikation entsprechen den Services in Part 4 mit konkreten Datentypen aus der jeweiligen Programmiersprache. Damit kann in der Applikationsentwicklung auf eine native API in der entsprechenden Programmiersprache zugegriffen werden. Die Applikation kann auch unabhängig vom verwendeten Binding implementiert werden. Neue Bindings können durch Austausch des OPC UA Stacks erweitert werden.

Synchrone und asynchrone Aufrufe

Abbildung 3-5



Bei COM sind alle Aufrufe zum Server synchron. Aus diesem Grund wurden dort für wenige Aktionen wie Read und Write zusätzliche asynchrone Funktionen definiert. Ein synchroner Aufruf startet die Aktion im Server. Nach Abschluss der Aktion wird vom Server ein synchroner Callback an den Client geschickt. Durch den synchronen Aufruf zum Starten der Aktion können auch asynchrone Aufrufe blockieren wenn die Netzwerkverbindung unterbrochen ist.

Bei OPC UA sind alle Aufrufe zum Server asynchron, eine Unterscheidung zwischen synchronen und asynchronen Methoden wird in der Spezifikation nicht gemacht. Nachdem die Request Nachricht auf das Netzwerk geschrieben ist, wird

Copyright © Siemens AG 2017 All rights reserved

der asynchrone Aufruf an die Client Applikation zurückgegeben. Ein Blockieren ist deshalb beim asynchronen Aufruf nicht möglich. Da ein asynchroner Aufruf immer synchron gemacht werden kann, bieten die Stacks alle OPC UA Methoden auch als synchrone Aufrufe an. Dafür wird der Aufruf so lange im Stack festgehalten, bis die Response Nachricht vom Server eingetroffen ist oder bis der Timeout abgelaufen ist. Die Timeouts können pro Aufruf individuell eingestellt werden. Für den Server gibt es keinen Unterschied zwischen synchronen und asynchronen Aufrufen.

Sicherheitsschichten

Die verschiedenen Sicherheitsschichten von OPC UA werden in Abbildung 3-6 und Tabelle 3-10 beschrieben.

Abbildung 3-6

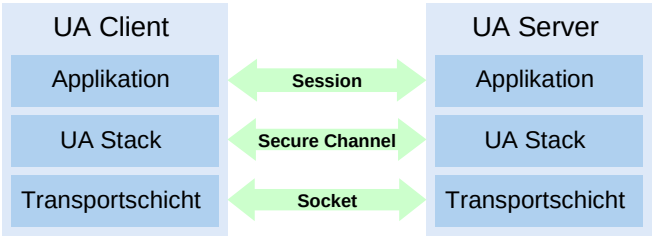


Tabelle 3-10

Schicht	Beschreibung
Socket	Auf Socketebene kann zusätzlich oder alternativ zum Secure Channel eine verbindungsorientierte Sicherung der Socketverbindung durch Secure Socket Layer (SSL) oder durch ein Virutal Private Network (VPN) eingesetzt werden.
SecureChannel	Auf der SecureChannel Ebene erfolgt eine gegenseitige Authentifizierung der Applikationen und eine nachrichtenbasierte Sicherung der Kommunikation. Dabei wird jede Nachricht signiert und verschlüsselt, um die Integrität und die Geheimhaltung der Nachrichten sicher zu stellen. Basis dieser Mechanismen sind Zertifikate, welche die Applikationen eindeutig auf Basis eines Public Key Infrastructure (PKI) Systems identifizieren. Eine detaillierte Beschreibung dieser Mechanismen findet sich in /2/. Der Austausch dieser Zertifikate als wichtigster Schritt bei der Sicherheitskonfiguration ist im nächsten Abschnitt beschrieben.
Session	Auf Sessionebene wird eine Benutzerauthentifizierung durchgeführt.

Konfigurationsoptionen für Security

Tabelle 3-11 beschreibt die verschiedenen Konfigurationsoptionen für die Sicherheitsmechanismen.

Tabelle 3-11

Option	Beschreibung
Security Policy	**None** – Es wird keine Sicherheit im Secure Channel verwendet. **Basic128Rsa15** – Set von Algorithmen für die Security. **Basic256** – Set von Algorithmen für die Security mit längeren Schlüsseln.
Message Security Mode	**None** – Die Nachrichten werden nicht gesichert. **Sign** – Die Nachrichten werden signiert. **Sign&Encrypt** – Die Nachrichten werden signiert und verschlüsselt.
User Authentication	**Anonymous** – Es ist keine Benutzerauthentifizierung notwendig. **User Password** – Die Benutzerauthentifizierung wird mit Benutzernamen und Passwort durchgeführt. **Certificate** – Die Benutzerauthentifizierung erfolgt durch ein Zertifikat.

Copyright © Siemens AG 2017 All rights reserved

Austausch der Zertifikate

Der Austausch der Zertifikate zwischen Client und Server und das Akzeptieren der Zertifikate ist in Abbildung 3-7 und Tabelle 3-12 erläutert.

Wenn alle beteiligten Applikationen die Richtlinien von OPC UA zur Sicherheitskonfiguration umsetzen, ist für den Austausch der Zertifikate nur ein manueller Schritt beim Server notwendig, da die Zertifikate zwischen den Applikationen automatisch ausgetauscht werden und nur das Akzeptieren der Zertifikate durch einen Administrator erfolgen muss.

Da noch nicht alle Applikationen die automatischen Schritte umsetzen, wird der manuelle Austausch der Zertifikate in Kapitel 5.3.2 erläutert.

Abbildung 3-7

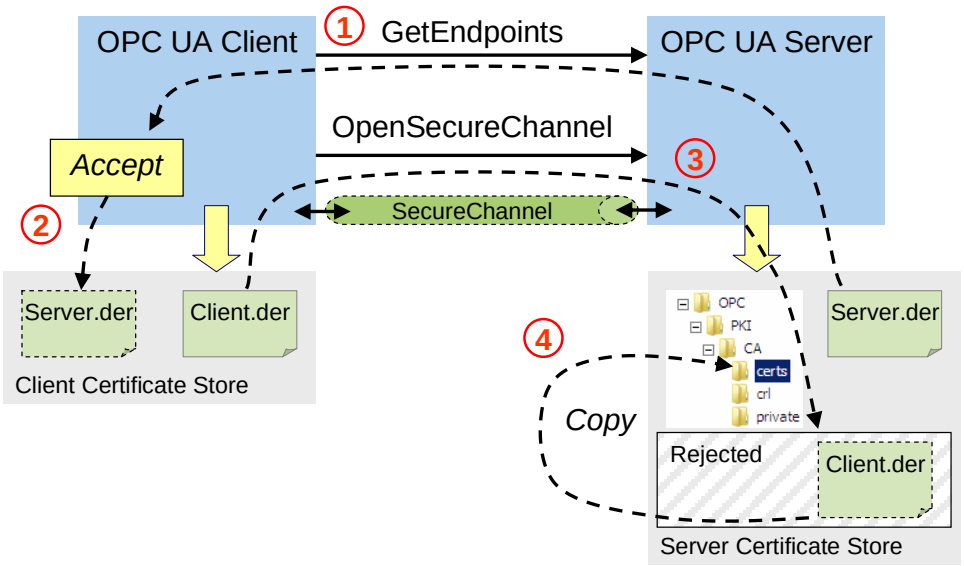


Tabelle 3-12

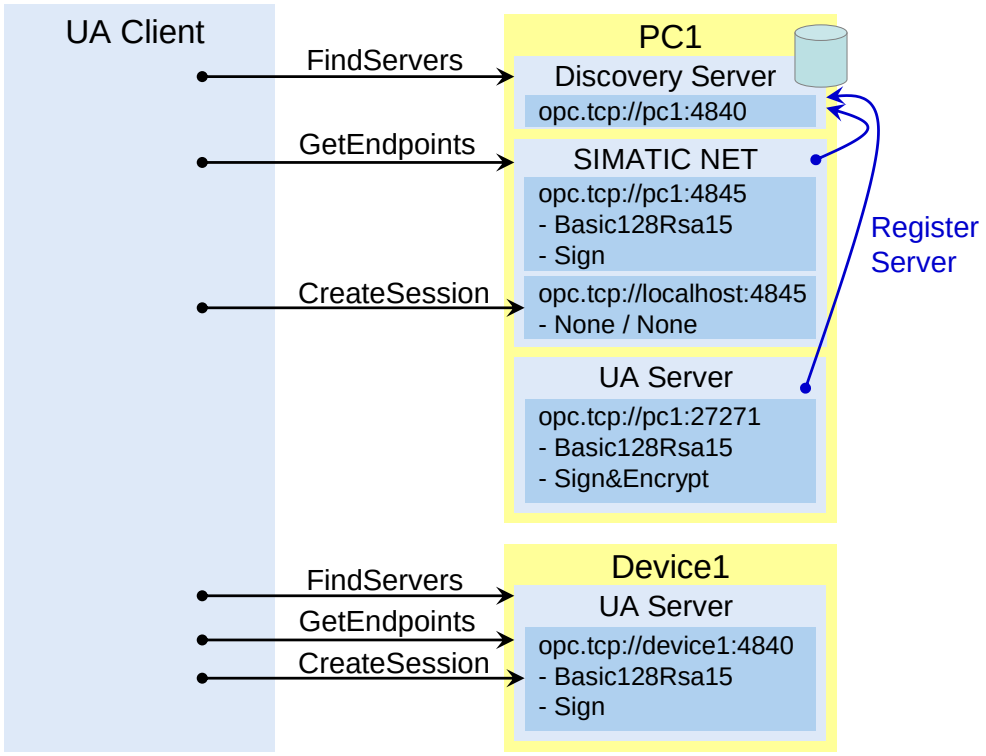
Schritt	Beschreibung
1.	Bevor der Client sich mit einem Server verbinden kann, benötigt er die notwendigen Informationen wie die vom Server geforderten Sicherheitsmechanismen, Protokoll und die Adresse zum Verbinden. Diese Informationen kennzeichnen einen sogenannten Endpoint. Mit dem Aufruf GetEndpoints werden die verfügbaren Endpoints eines Servers geliefert. Mit der Beschreibung des Endpoints liefert der Server auch sein Zertifikat.
2.	Nach der Auswahl des Endpoints mit den Sicherheitseinstellungen wird der Benutzer gefragt, ob er das Zertifikat akzeptieren will. Wenn ja, wird es im Zertifikatsspeicher des Clients abgelegt.
3.	Beim Aufruf von OpenSecureChannel wird das Client Zertifikat an den Server übergeben. Ist das Zertifikat im Server nicht bekannt, wird es in einem Rejected Verzeichnis gespeichert.
4.	Mit einem Konfigurationstool des Servers können Zertifikate aus dem Rejected Verzeichnis akzeptiert werden. Dabei werden sie in den Zertifikatsspeicher des Servers verschoben.

Server Discovery

Bisher ist für OPC UA Discovery ein Local Discovery Service (LDS) definiert, der von der grundlegenden Funktionalität her mit dem OPC Enum bei klassischem OPC vergleichbar ist.

Copyright © Siemens AG 2017 All rights reserved

Abbildung 3-8



Ein LDS liefert eine Liste der lokal auf dem Netzwerkknoten verfügbaren OPC UA Server. Der LDS läuft standardmäßig auf dem für OPC UA registrierten Port 4840. Damit ist der LDS immer über **opc.tcp://<Node>:4840** als URL ansprechbar. Die Server auf einem PC registrieren sich mit dem LDS.

Ein Client kann mit den folgenden Schritten einen Server aussuchen und eine Verbindung aufbauen:

- Verbindungsaufbau ohne Security mit Port 4840 und Aufruf von **FindServers**. Dieser Aufruf liefert die Liste der verfügbaren Server und deren Discovery URL.
- Verbindungsaufbau ohne Security zur Discovery URL des gewünschten Servers und Aufruf von **GetEndpoints**. Dieser Aufruf liefert die Liste der Enpoints mit den Endpoint URLs und den Sicherheitseinstellungen der Endpoints.
- Verbindungsaufbau mit der Endpoint URL und den geforderten Sicherheitseinstellungen. Danach kann mit **CreateSession** eine Applikationssession mit dem Server geöffnet werden.

Ist auf einem System nur ein OPC UA Server verfügbar, kann dieser auf dem Standardport 4840 laufen, da alle Server auch FindServers implementieren müssen und dann nur sich selbst liefern. Auch die Endpoints verwenden in diesem Fall den Port 4840.

Copyright © Siemens AG 2017 All rights reserved

3.2.5 Abgrenzung und Vergleich mit OPC Data Access

Überblick

OPC Unified Architecture greift alle Features der klassischen OPC Schnittstellen auf und setzt diese lediglich in einem gemeinsamen Ansatz für die verschiedenen klassischen OPC Schnittstellen um.

Diese Applikation beschäftigt sich mit der Data Access Funktionalität in OPC UA. Aus diesem Grund wird die Umsetzung von OPC Data Access auf OPC UA in diesem Kapitel erläutert.

Aufbau des Adressraumes

Tabelle 3-13 erläutert die Umsetzung des OPC DA Adressraumes auf die Konzepte von OPC UA.

Tabelle 3-13

OPC Data Access	OPC Unified Architecture
Knoten im Adressraum	
Verzeichnisse werden zur hierarchischen Strukturierung des Adressraumes verwendet.	Verzeichnisse können mit Folder Objekten realisiert werden. Die Hierarchie wird mit Organizes Referenzen aufgebaut.
OPC Items repräsentieren Datenpunkte im Adressraum. Sie sind die Blätter der Verzeichnisse.	Variable Knoten werden für die Abbildung von OPC Items verwendet.
Properties und Attribute	
ItemId für OPC Items	NodeId für alle Knoten im Adressraum
Property Item Canonical Data Type	Attribute DataType, ValueRank und ArrayDimension
Properties Item Value, Quality und Timestamp	Das Value Attribut liefert Wert, Status und Zeitstempel.
Property Item Access Rights	Attribute AccessLevel und UserAccessLevel

Copyright © Siemens AG 2017 All rights reserved

Zugriff auf Daten

Tabelle 3-14 erläutert die Umsetzung des OPC DA Zugriff auf Daten auf die Konzepte von OPC UA.

Tabelle 3-14

OPC Data Access	OPC Unified Architecture
Kontext	
COM Objekt OPCServer	OPC UA Session
COM Objekt OPCGroup	OPC UA Subscription
OPCItem in einer Gruppe	Data Monitored Item in einer Subscription
Kontext erzeugen	
CoInitializeEx CoInitializeSecurity CoCreateInstanceEx erzeugt OPCServer	OpenSecureChannel CreateSession ActivateSession
AddGroup RemoveGroup IOPCGroupStateMgt::SetState	CreateSubscription DeleteSubscriptions ModifySubscription
AddItems RemoveItems	CreateMonitoredItems DeleteMonitoredItems

OPC Data Access	OPC Unified Architecture
Zugriff auf Informationen	
ChangeBrowsePosition / BrowseOPCItemIDs GetItemID / QueryAvailableProperties	Browse
IOPCItemIO::Read IOPCSyncIO::Read IOPCSyncIO2::ReadMaxAge IOPCAsyncIO2::Read IOPCAsyncIO3::ReadMaxAge IOPCItemProperties::GetItemProperties	Read
IOPCItemIO::WriteVQT IOPCSyncIO::Write IOPCSyncIO2::WriteVQT IOPCAsyncIO2::Write IOPCAsyncIO3::WriteVQT	Write
OnDataChange	Publish
GetStatus ShutdownEvent	Lesen oder Monitoren der ServerState und ServerStatus Variablen

Copyright © Siemens AG 2017 All rights reserved

3.3 Grundlagen zum Thema S7 Kommunikation

3.3.1 Allgemein

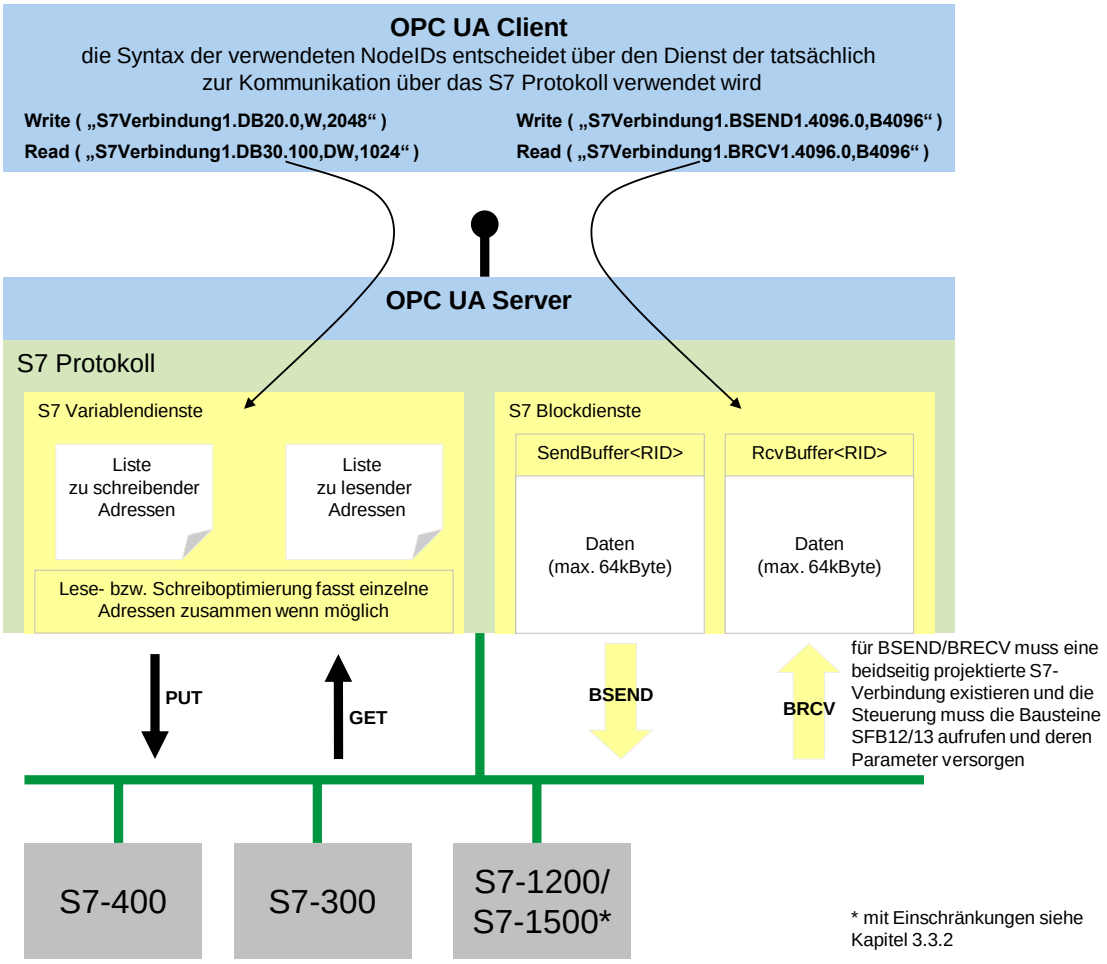
Hier wird der Zugriff mit Hilfe des SIMATIC S7-Protokolls über den UA-Server auf S7-Steuerungen beschrieben.

Wirkungskette der Kommunikation

Die S7 Kommunikation wird in zwei grundlegend verschiedene Kommunikationsdienste unterteilt, in Variablendienste und Blockdienste. Diese sind auf der Ebene des OPC UA Servers fast vollständig verdeckt. Nur anhand der NodeId lässt sich erkennen, welcher Kommunikationsdienst zur Steuerung hin verwendet wird. Der Namespace "S7:" legt fest, dass es sich um die neue Adressierungsart handelt, die an die Syntax von Anypointern angelehnt ist. Vor allem bei einer großen Anzahl von Variablen bietet sie einen deutlich performanteren Zugriff. Zur Kompatibilität mit der früheren Syntax der ItemIDs beim klassischem OPC Data Access bleibt der alte Aufbau der Bezeichner unter dem Namespace "S7COM:" erhalten.

Der OPC UA Server zerlegt intern die NodeId in ihre Bestandteile und erkennt an ihrem Aufbau, über welchen Kommunikationsdienst zur S7 Steuerung kommuniziert werden muss. Hierbei identifiziert der Verbindungsname den Kommunikationspartner (dieser Name repräsentiert unter anderem z.B. eine IP Adresse) und das Schlüsselwort "BRCV" bzw. "BSEND" bewirkt die Verwendung der Blockdienste anstelle der Variablendienste. Der S7 Typbezeichner und die Offsetadresse geben die Position der Daten innerhalb der Steuerung an und der Datentyp bestimmt die Interpretation dieser Daten.

Abbildung 3-9



Copyright © Siemens AG 2017 All rights reserved

Variablendienste

Eine S7 Steuerung antwortet auf Anfragen über Variablendienste, hierzu ist nur eine einseitig projektierte Verbindung erforderlich. Jede S7-Steuerung ist ein so genannter "S7-Server" und beantwortet PUT/GET Anfragen, ohne dass dafür im Steuerungsprogramm der SPS etwas implementiert werden muss. Es kann direkt auf alle Datenbereiche der Steuerung zugegriffen werden (E, A, M, DB, etc.). Dieser Kommunikationsdienst ist sehr flexibel und vor allem einfach zu verwenden.

NodeIds für Variablendienste

Syntax im Namespace **S7:**

<connectionname>.<S7object>.<address>{,{<S7type>}{,<quantity>}}

Beispiel: S7-Verbindung_3.DB10.20,W

Eine Variable vom Typ Wort (16Bit ohne Vorzeichen), die sich im Datenbaustein 10 befindet und an der Offset-Byteadresse 20 anfängt (also aus den Bytes 20 und 21 besteht). Diese Variable wird mit Put/Get über die Verbindung mit dem Namen "S7-Verbindung_3" geholt, also von der S7-Steuerung, die sich hinter dieser Verbindung verbirgt.

Symbolische NodeIDs

Neben der direkten Adressierung über die neue Syntax ("S7:"Namespace) sowie der alten kompatiblen Adressierung (S7COM:-Namespace) gibt es die Möglichkeit der symbolischen Adressierung. Hierzu wird von STEP 7 aus der Adressraum generiert. Für alle symbolischen Bezeichner der Datenpunkte in den S7-Steuerungen, die über eine S7-Verbindung mit einem OPC Server verbunden sind, kann ein Symbolexport angestoßen werden. Die daraus entstehende Symboldatei mit der Endung ATI wird dem OPC UA Server per Download aus STEP 7 oder per XDB-Import bekannt gemacht. Die ATI-Datei (Advanced Tag Information) beinhaltet eine Abbildung der symbolischen Namen auf die direkten Adressen.

Hinweis Symbolische Adressierung in der schnellen, hochoptimierten ATI Variante ist nur für Variablendienste des S7 Protokolls verfügbar. Anders gesagt: alle Symbole werden letztendlich über PUT/GET von den Steuerungen geholt. Symbole, die eine BSEND bzw. BRCV Variable abbilden, sind nicht möglich.

Blockdienste

Für den Austausch großer Datenmengen steht der effektivere Blockdienst zur Verfügung. Auf einer beidseitig projektierten Verbindung können große Datenmengen (bis 64kByte) ausgetauscht werden. Die Kommunikation basiert auf dem Austausch von Datenpuffern. Allerdings müssen im Steuerungsprogramm hierzu entsprechende Systemfunktionsbausteine (BSEND/BRECV) aufgerufen werden. Der OPC UA Server stellt auf dem PC die entsprechenden Gegenstücke bereit, wenn die dazugehörigen NodeIds (früher OPC Items) angelegt werden.

Aufbau der NodeIds für Blockdienste

Syntax BRCV im Namespace **S7:**

<connectionname>.brcv<rid>.<address>{,{<S7type>}{,<quantity>}}

Beispiel: S7-Verbindung_5.brcv3

Der komplette Empfangspuffer für das BSEND/BRECV Paar mit der ID 3, dass über die Verbindung mit dem Namen "S7-Verbindung_5" angeschlossen ist, wird für OPC UA in einen ByteString abgebildet. Dieser ByteString enthält immer die zuletzt vom Kommunikationspartner (auf der anderen Seite von "S7-Verbindung_5") mit BSEND gesendeten Daten. Auf einer S7-Verbindung können mehrere zusammengehörende BSEND/BRECV Pärchen existieren, die über ihre RID verbunden sind. Hier ist es der BRECV, der zum BSEND mit der ID 3 gehört.

Syntax BSEND im Namespace **S7:**

<connectionname>.bsend<rid>.<bufferlength>.<address>{,{<S7type>}{,<quantity>}}

Beispiel: S7-Verbindung_2.bsend1.1024.100,W,20

Beim Schreiben auf diese NodeId wird ein Array of Words (unsigned integer 16 Bit) mit 20 Elementen ab der Byte-Offset-Adresse 100 in den Sendepuffer der Länge 1024 geschrieben. Es wird in dem 1024 Byte großen Puffer der Bereich von 100 bis 140 überschrieben. Der gesamte Block wird mit der ID 1 zum Kommunikationspartner gesendet, der einen BRECV mit der ID 1 und einer Mindestlände von 1024 Bytes bereitstellen muss, um die Daten zu empfangen.

Hinweis Für die Verwendung der Blockdienste BSEND/BRCV muss eine beidseitig projektierte Verbindung existieren und die Steuerung muss selbstständig die Bausteine SFB12/13 aufrufen und deren Parameter versorgen.

Copyright © Siemens AG 2017 All rights reserved

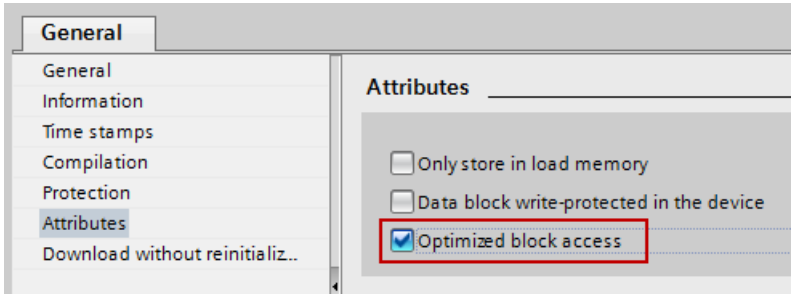
Zum Thema blockorientierte Dienste beachten Sie auch die Hinweise im SIMATIC NET Handbuch "Industrielle Kommunikation – Band 2 - Schnittstellen".

3.3.2 optimierte S7-Kommunikation

Hintergrund

Mit einer S7-Verbindung ist der Zugriff auf optimierte Datenbausteine in den S7-1200/S7-1500 CPUs nicht möglich. Um mit einem OPC Server über eine S7-Verbindung Daten aus einer S7-1200/S7-1500 abzuholen, dürfen die Datenbausteine, aus denen gelesen werden soll, nicht optimiert sein.

Abbildung 3-10 Eigenschaften eines Datenbausteins

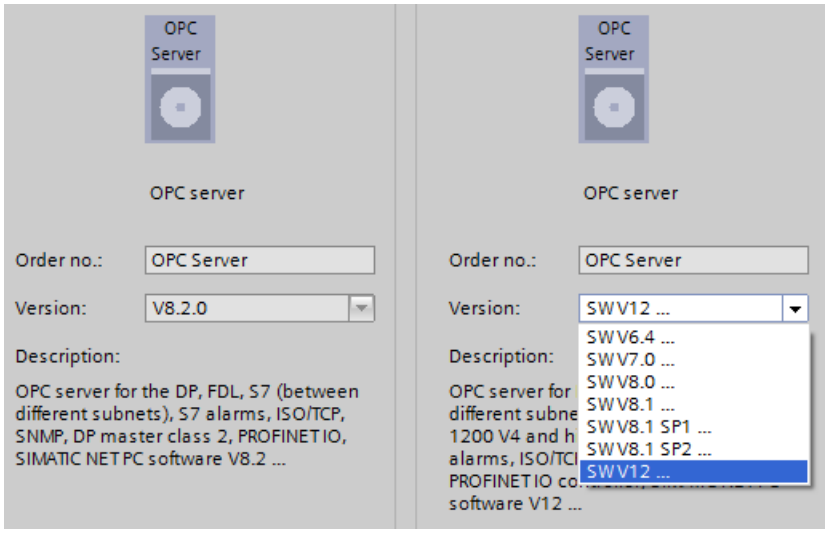


Nachteil: Die Performance der innovierten Steuerungen S7-1200/S7-1500 wird durch die Verwendung von nicht zugriffsoptimierten Datenbausteinen beeinträchtigt.

Abhilfe

Um keine Performance-Verluste in den S7-1200/S7-1500 Steuerungen durch nicht zugriffsoptimierte Datenbausteine zu erhalten, gibt es ab SIMATIC NET OPC V12 die sogenannten "optimierte S7-Kommunikation".

Abbildung 3-11



Diese Verbindungsart wird bei Verwendung des SIMATIC NET OPC Servers V14 und einer innovierten Steuerung automatisch angelegt. Mit Hilfe der optimierten S7-Verbindung ist dann auch ein Zugriff auf zugriffsoptimierte Datenbausteine möglich.

Copyright © Siemens AG 2017 All rights reserved

Zugriff mit OPC-Client

Der Zugriffpunkt auf die optimierten S7-Verbindungen erfolgt über den SimaticNET.S7OPT-Server auf Port 50101.

Copyright © Siemens AG 2017 All rights reserved

4 Funktionsmechanismen dieser Applikation

Gesamtübersicht

Abbildung 4-1

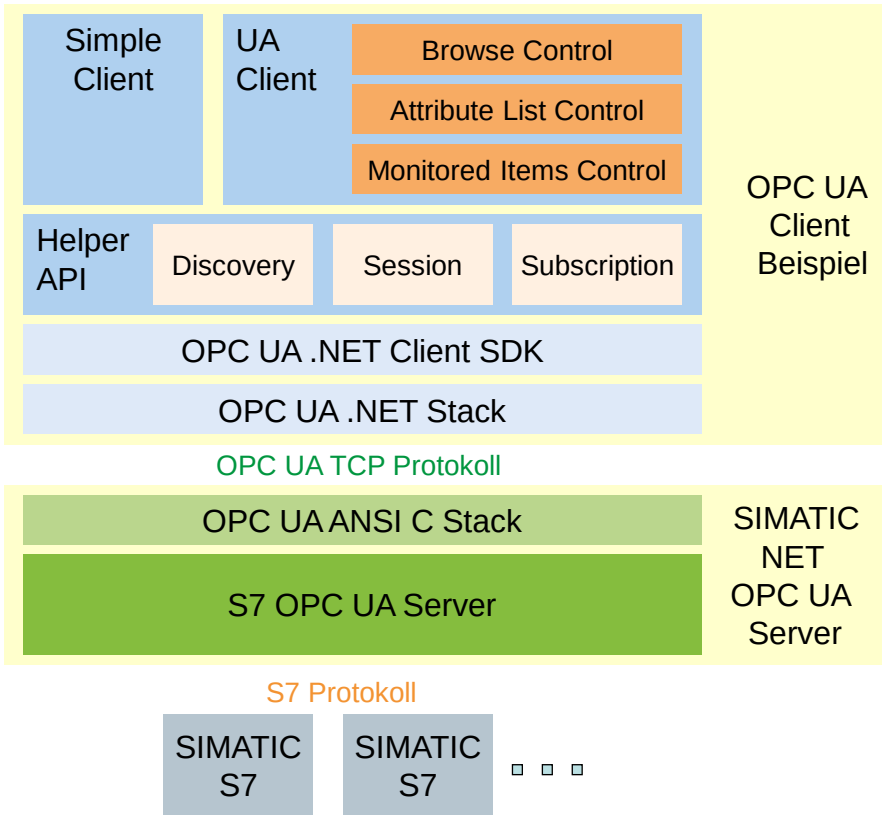


Tabelle 4-1

Modul	Beschreibung
OPC UA .NET Stack	Der .NET basierte OPC UA Kommunikationsstack von der OPC Foundation.
.NET Client SDK	Das .NET basierte OPC UA Client SDK der OPC Foundation.
Helper API	Wiederverwendbare und vereinfachte .NET Klasse. Sie bietet Funktionen für Discovery, Session und Subscription Handling.
Simple Client	Einfaches Beispiel für die Verwendung der Client API mit den Funktionen Connect, Disconnect, Read, Write und Daten Monitoring. In diesem Beispiel wird auch die direkte Adressierung und das Handling von Namespaces gezeigt.
UA Client	Komfortabler OPC UA Client mit den Funktionen Discovery, Connect, Disconnect, Browse, Read aller Attribute, Write und Data Monitoring. Allgemeine Funktionen wie Browse, Attribute Auflisten und das Monitoring von Datenvariablen ist in wiederverwendbaren Controls gekapselt. In diesem Beispiel können die symbolischen Variablen gebrowst und direkt aus dem Browser heraus verwendet werden.
ANSI C UA Stack	Der SIMATIC NET OPC UA Server verwendet den optimierten und portablen OPC UA ANSI C Stack der OPC Foundation.
S7 OPC UA Server	Der SIMATIC NET OPC UA Server implementiert die notwendige Serverlogik für Sessions und Subscriptions und die Datenanbindung an die S7 Stationen.

Copyright © Siemens AG 2017 All rights reserved

.NET Client SDK

Das verwendete SDK der OPC Foundation wird von der OPC Foundation nicht mehr gepflegt oder supportet. Das beeinträchtigt die Funktionalität des Applikationsbeispiels nicht.

Empfohlen für die Verwendung in eigenen Applikationen sind SDKs von kommerziellen Anbietern wie zum Beispiel von Unified Automation (siehe \7\).

Programmübersicht

Die folgende Abbildung zeigt die Funktionsblöcke im OPC UA Client und das Zusammenspiel mit dem OPC UA Server.

Abbildung 4-2

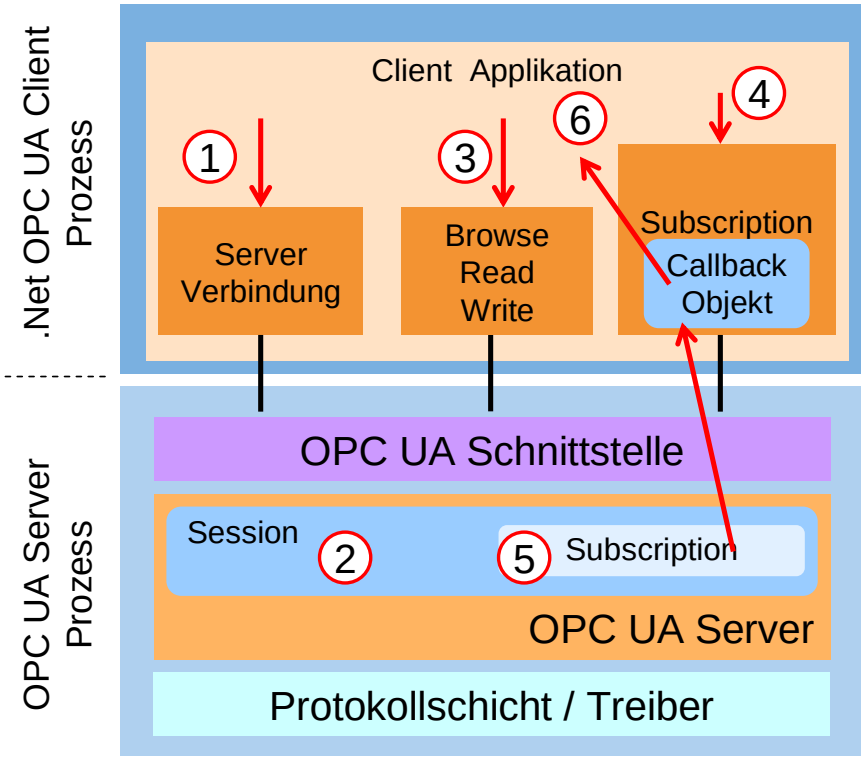


Tabelle 4-2

Nr.	Beschreibung
1	Beim Verbindungsaufbau der Oberfläche mit dem OPC UA Server wird auf Clientseite ein Client API Objekt erzeugt. Dieses Objekt verwaltet die Verbindung mit dem Server (2). Es stellt darüber hinaus alle OPC UA Services zur Verfügung mit Ausnahme von Services, die sich auf eine Subscription beziehen.
2	Über die OPC UA Schnittstelle wird ein Session Objekt im Server erzeugt.
3	Beim Verbindungsaufbau der Oberfläche mit dem OPC UA Server wird auch die erste Ebene des Adressraums des OPC UA Servers dargestellt. Dabei wird der Browse Service der OPC UA Schnittstelle genutzt. Selektiert man einen Knoten in der Baumansicht des Browse Controls im Client, werden mittels des Read Services die Attributwerte des Knotens in einem weiteren Fenster angezeigt.

Copyright © Siemens AG 2017 All rights reserved

4	Beim Anmelden von Variablen zur Beobachtung von Wertänderungen wird ein Subscription Objekt erzeugt, das alle OPC UA Services zur Verfügung stellt, die sich auf eine Subscription beziehen.
5	Über die OPC UA Schnittstelle wird im Server ein Subscription Objekt erzeugt, das alle Subscription relevanten Einstellungen verwaltet.
6	Um Wertänderungen vom Server empfangen zu können, wird eine Callback-Verbindung aufgebaut. Dafür wird ein SubscriptionCallback Objekt im Client angelegt und mit der Subscription im Server verbunden. Werden Änderungen vom Server an den Client geschickt, trägt dieser die Änderungen in das Monitoring Fenster ein.

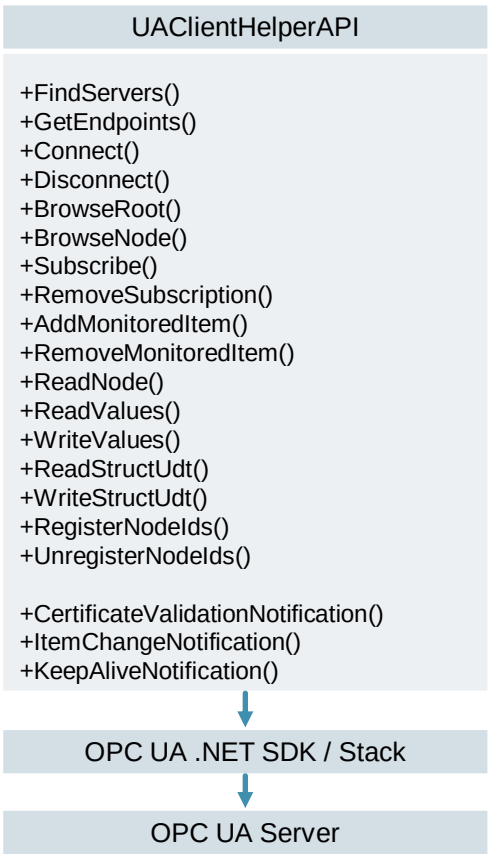
Copyright © Siemens AG 2017 All rights reserved

4.1 OPC UA Client Helper API

Die nachfolgende Abbildung zeigt das Klassendiagramm zur Klasse "UAClientHelperAPI". Die wichtigsten Zugriffsmethoden auf einen OPC UA Server sind in dieser Klasse gekapselt und zusammengefasst.

Die UAClientHelperAPI greift auf die .NET-Assemblys Opc.UA.Client.dll und Opc.UA.Core.dll der OPC Foundation zurück.

Abbildung 4-3



Copyright © Siemens AG 2017 All rights reserved

Methodenbeschreibung

Die nachfolgende Tabelle erläutert die Funktionen der public-Methoden innerhalb der Klasse "UAClientHelperAPI", über die die OPC UA Client Funktionalitäten realisiert sind:

Tabelle 4-3

Methode	Beschreibung
FindServers	Sucht nach OPC UA Servern im Netzwerk. Voraussetzung: Ein LDS (Local Discovery Server) oder GDS (Global Discovery Server) muss vorhanden sein.
GetEndpoints	Ermittelt die verfügbaren Endpoints auf einem Server, über die eine Verbindung aufgebaut werden kann.
Connect	Baut die Verbindung zu einem Server auf und erzeugt einen Secure Channel und eine Session zum Server.
Disconnet	Beendet eine bestehende Session und baut die Verbindung zum Server ab.
BrowseRoot	Liefert eine Collection von Nodes zurück, die im Root-Verzeichnis des Servers zu finden sind.
BrowseNode	Liefert eine Collection von Nodes zurück, die unter einer spezifischen Node zu finden sind.
Subscribe	Erstellt eine Subscription auf dem Server.
RemoveSubscription	Löscht eine spezifische Subscription vom Server.
AddMonitoredItem	Fügt einer bestehen Subscription ein MonitoredItem zur Beobachtung hinzu.
RemoveMonitoredItem	Löscht ein bestehendes MonitoredItem einer Subscription.
ReadNode	Liest die Metadaten einer spezifischen Node aus.
ReadValues	Liest die Werte von Variablen-Nodes aus.
WriteValues	Schreibt Werte auf Variablen-Nodes.
ReadStructUdt	Liest Werte benutzerdefinierter Strukturen und UDTs unter Zuhilfenahme des "TypeDictionary" des Servers aus.
WriteStructUdt	Schreibt Werte auf benutzerdefinierter Strukturen und UDTs, die zuvor gelesen wurden.
RegisterNodeIds	Registriert Node Ids beim Server für einen optimierten Zugriff auf diese.
UnregisterNodeIds	Löscht die Registrierung von bereits registrierten Node Ids.
CertificateValidation Notification	Event, das gefeuert wird, wenn ein Serverzertifikat nicht akzeptiert werden kann.
ItemChangeNotification	Event, das gefeuert wird, wenn sich die Werte eines MonitoredItems geändert haben.
KeepAliveNotification	Event, das gefeuert wird, wenn eine KeepAliveNotification eintrifft.

Copyright © Siemens AG 2017 All rights reserved

4.2 Simple OPC UA Client

Der Simple Client stellt ein einfaches Beispiel für die Verwendung der Client API dar. Die wichtigsten Funktionen wie Connect, Disconnect, Read, Write und Monitoring von Daten werden in einer Datei bzw. Klasse mit einem Dialog gezeigt. Der Code für das Beispiel kann im Projekt SimpleClient in der Datei MainForm.cs gefunden werden.

Oberfläche des einfachen Beispiels

Die Oberfläche wird über Buttons für die einzelnen Funktionen bedient.

Abbildung 4-4

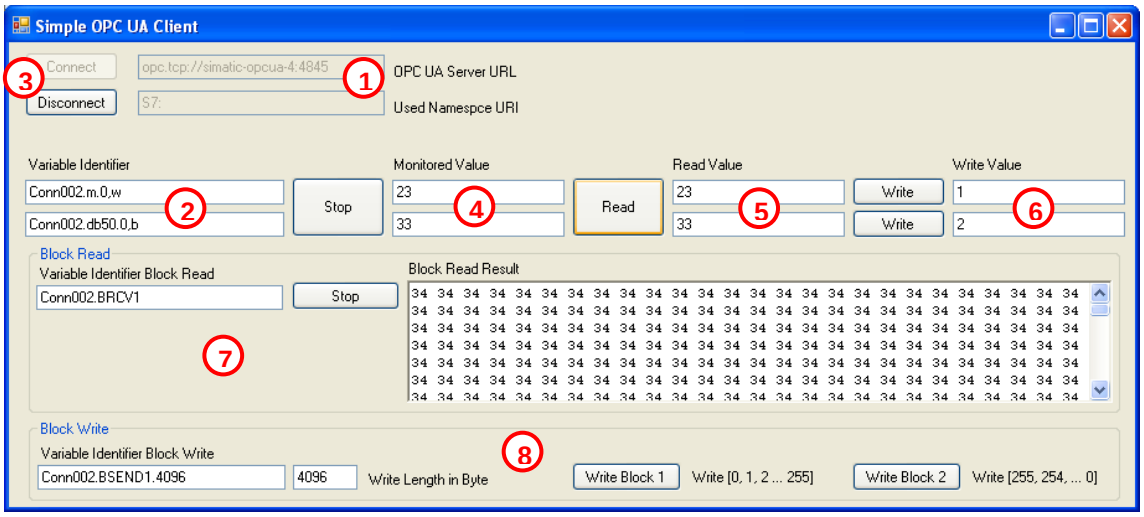


Tabelle 4-4

Nr.	Beschreibung
1.	In der Textbox für die Server URL kann die Server URL angegeben werden. Diese setzt sich für den SIMATIC NET OPC Server zusammen aus **opc.tcp://<Rechnername>:55101**. In der Textbox Namespace URI wird der verwendete Namespace angegeben. Dies ist **S7:** für direkte Adressierung, **S7COM:** für direkte Adressierung über die OPC DA kompatible Syntax und **SYM:** für symbolische Adressierung.
2.	In den Textboxen für die Variable Identifier wird der Kennungsteil der NodeId angegeben. Diese setzt sich ist beim Namespace S7: z.B. zusammen aus **<S7Verbindung>.<Datenbereich>.<Offset>,<Datentyp>** Die NodeId zum Lesen und Schreiben wird aus der Kennung und dem Namespace Index gebildet. Der Namespace Index ergibt sich aus der Position des Namespaces in der Namespace Tabelle des Servers. Diese Tabelle kann mit Read vom Server gelesen werden.
3.	Über die Buttons Connect und Disconnect kann die Verbindung zum OPC UA Server auf- und abgebaut werden. Die Verbindung wird ausschließlich ohne Sicherheit aufgebaut. Der sichere Verbindungsaufbau wird im nächsten Beispiel erläutert.
4.	Über den Button Monitor wird eine Subscription angelegt und mit den beiden NodeIds werden zwei Monitored Items in der Subscription angelegt. Die Datenänderungen werden in den Textboxen neben dem Button angezeigt. Fehler werden jeweils an Stelle der Werte angezeigt.
5.	Über den Button Read werden die Werte (Value Attribut) der beiden Variablen mit den angegebenen NodeIds gelesen und in den Textboxen neben dem Button angezeigt.
6.	Über den Button Write wird der Wert aus der Textbox neben dem Button auf die durch die NodeId identifizierte Variable geschrieben. Für das Schreiben muss erst Read aufgerufen werden, da der Text aus der Textbox in den passenden Datentyp der Variable konvertiert werden muss. Die Konvertierung geschieht auf Basis des Datentyps, der bei Read geliefert wird.

Copyright © Siemens AG 2017 All rights reserved

Nr.	Beschreibung
7.	In der Gruppe "Block Read" können Daten empfangen werden, die von der S7 mit dem Blockdienst BSEND aktiv geschickt werden. Dies kann z.B. für das Senden von Ergebnisdaten von der S7 an eine PC Applikation verwendet werden.
8.	In der Gruppe "Block Write" können Datenblöcke an die S7 geschickt werden, die dort mit dem Blockdienst BRECV empfangen werden. Es können zwei Blöcke mit unterschiedlichen Inhalten geschickt werden. Dies kann z.B. für den Download von Rezeptdaten zur S7 verwendet werden.

Funktionen des einfachen Beispiels

Die Funktionen können in der Klasse MainForm in der Datei MainForm.cs gefunden werden. In den Funktionen ist ein einfaches Fehlerhandling implementiert. Tritt bei OPC UA Aufrufen eine Exception auf, wird ein Dialog mit der Fehlermeldung aufgeblendet. Tritt bei den variablenbezogenen Aufrufen ein Fehler bei einer oder mehreren Variablen auf, wird der Fehler in den zugehörigen Textboxen ausgegeben.

Tabelle 4-5

Funktion	Beschreibung
btnConnect_Click	In dieser Funktion wird die Verbindung zum OPC UA Server über die Funktion Server::Connect() der Client API aufgebaut. Dabei wird der URL String aus der entsprechenden Textbox übergeben. Ist die Verbindung erfolgreich aufgebaut worden, wird die Namespace Tabelle vom Server über die Funktion Server::ReadValues() gelesen. In der zurückgegebenen Tabelle wird der Namespace aus der Textbox Namespace URI gesucht. Der Index in der Tabelle wird in einer Klassenvariablen gespeichert.
btnDisconnect_Click	In dieser Funktion wird die Verbindung zum OPC UA Server über die Funktion Server::Disconnect() der Client API abgebaut
btnRead_Click	In dieser Funktion werden zuerst die beiden NodeIds aus dem Namespace Index und den Identifier Texten gebildet. Danach werden die beiden Werte über die Funktion Server::ReadValues() gelesen. Das Ergebnis wird in die zugehörige Textbox geschrieben. Das Ergebnis kann entweder der gelesene Wert oder ein Fehlercode sein.
btnMonitor_Click	Wenn keine Subscription angelegt ist, wird in dieser Funktion zuerst mit Server::Subscribe() eine Subscription angelegt. Danach werden die beiden NodeIds aus dem Namespace Index und den Identifier Texten gebildet. Anschließend werden die beiden Monitored Items mit Subscription::AddMonitoredItem() angelegt. Dabei wird die zugehörige Textbox für die Ausgabe der Werte als Client Handle übergeben. Ist eine Subscription angelegt, wird diese mit Server::RemoveSubscription() gelöscht.
ClientApi_ValueChanged	Die Funktion wird bei Subscription::AddMonitoredItem() als Callback Funktion angegeben. In der Funktion wird zuerst geprüft, ob der Aufruf im Haupt-Thread des Dialogs ankommt. Wenn nicht, wird der Aufruf über BeginInvoke an den Haupt-Thread des Dialogs übergeben. Ansonsten ist ein Zugriff auf den Dialog nicht möglich. Danach wird geprüft, ob das Client Handle eine Textbox ist. Wenn nicht, handelt es sich um einen Callback für den Block Read. Handelt es sich um den Wert einer normalen Variablen, wird das Client Handle zurück auf eine Textbox gecastet und nach der Prüfung des Status entweder der Wert oder ein Fehlercode in die Textbox eingetragen. Handelt es sich dagegen um den Wert einer Blockvariablen, wird das Bytearray extrahiert und als Folge von HEX-Werten für die einzelnen Elemente des Bytearrays ausgegeben.

Copyright © Siemens AG 2017 All rights reserved

Funktion	Beschreibung
btnWrite1_Click	Die Funktion setzt die NodeId für Variable 1 zusammen und ruft die Funktion writeNewValue mit der NodeId, dem neuen Wert als Text aus der Textbox und mit dem zuletzt gelesenen Wert auf.
btnWrite2_Click	Die Funktion setzt die NodeId für Variable 2 zusammen und ruft die Funktion writeNewValue mit der NodeId, dem neuen Wert als Text aus der Textbox und mit dem zuletzt gelesenen Wert auf.
writeNewValue	In dieser Funktion wird zuerst der neue Wert von einem Text auf den Datentyp konvertiert, der beim letzten Lesen geliefert wurde. Danach wird der Wert über die Funktion Server::WriteValues() geschrieben und das Ergebnis geprüft.
btnMonitorBlock_Click	Wenn keine Block Subscription angelegt ist, wird in dieser Funktion zuerst mit Server::Subscribe() eine Subscription angelegt. Danach wird die NodeId aus dem Namespace Index und den Identifier Text gebildet. Anschließend wird das Monitored Item mit Subscription::AddMonitoredItem() angelegt. Ist eine Subscription angelegt, wird diese mit Server::RemoveSubscription() gelöscht.
btnWriteBlock1_Click	Die Funktion setzt die NodeId für die Blockvariable zusammen und simuliert Datensatz 1. Die Simulation legt ein Bytearray mit der in der Oberfläche angegeben Länge an und befüllt die Werte mit einem Variablenwert, der bei 0 beginnt und nach jeder Zuweisung inkrementiert wird. Danach wird die Funktion writeNewBlockValue mit der NodeId und dem neuen Wert aufgerufen.
btnWriteBlock2_Click	Die Funktion ist identisch zu btnWriteBlock1_Click; allerdings beginnt der hier zugewiesene Variablenwert mit 255 und wird nach jeder Zuweisung dekrementiert.
writeNewBlockValue	In dieser Funktion wird der Wert über die Funktion Server::WriteValues() geschrieben und dann das Ergebnis geprüft.

Copyright © Siemens AG 2017 All rights reserved

4.3 Komfortabler OPC UA Client

4.3.1 Oberfläche

Die folgende Abbildung und Tabelle beschreiben die Oberfläche des generischen OPC UA Clients Beispiels mit dem auf die Informationen des Namensraums eines OPC UA Servers komfortabel zugegriffen werden kann.

Abbildung 4-5

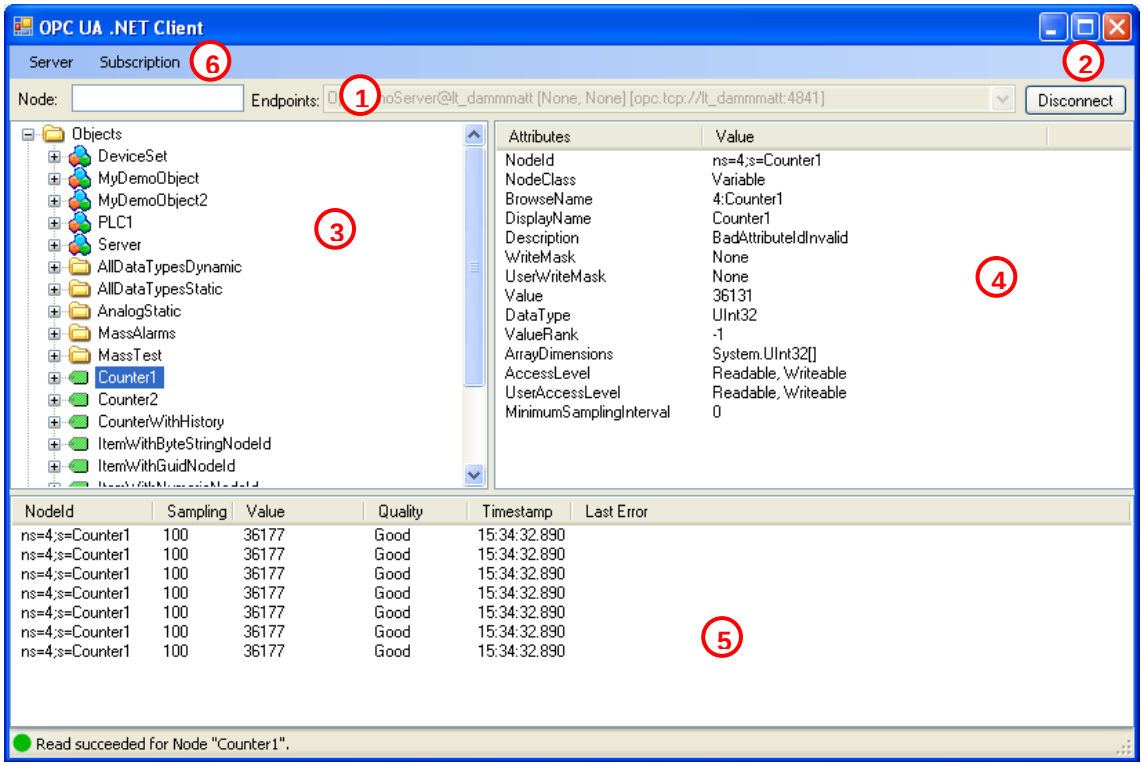


Tabelle 4-6

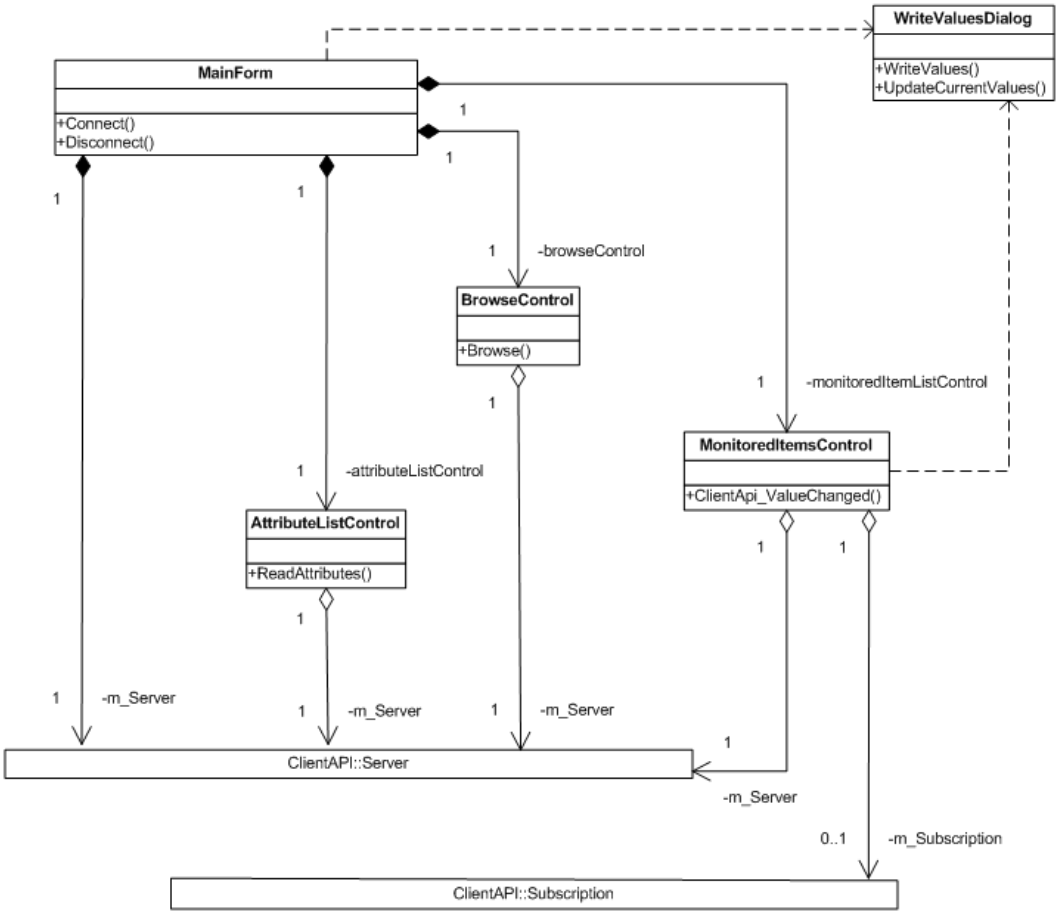
Nr.	Beschreibung
1.	Über die Auswahlliste kann der Server Endpoint ausgewählt werden. Dafür wird die Liste der Server und Endpoints vom Discovery Server ermittelt. Der Rechner auf dem der Discovery Server abgefragt werden soll, kann im Textfeld Node eingegeben werden. Ist das Feld leer, wird der lokale Discovery Server angesprochen.
2.	Über den Button Connect kann die Verbindung zum Server auf- bzw. abgebaut werden.
3.	Im Browse Control wird der gesamte Adressraum des verbundenen Servers in einer hierarchischen Baumansicht angezeigt. Es werden nur hierarchische Referenzen angezeigt.
4.	Für den selektierten Knoten im Browse Control werden die Attribute gelesen und in diesem Control angezeigt.
5.	Mit Drag&Drop können Variablen aus dem Browse Control in das Beobachtungsfenster gezogen werden. Für die Variable wird dann die NodeId, das Sampling Intervall, der Wert, der Zeitstempel und der Status Code angezeigt.
6.	Über das Kontextmenü im Beobachtungsfenster oder über das Applikationsmenü können die Eigenschaften der Subscription und Monitored Items verändert werden. So kann z.B. das Sampling Intervall geändert werden. Außerdem kann der Dialog zum Schreiben geöffnet werden. Dabei werden diejenigen Variablen in den Dialog übernommen, die im Beobachtungsfenster markiert sind.

Copyright © Siemens AG 2017 All rights reserved

4.3.2 Klassendiagramm

Das folgende Klassendiagramm zeigt die Klassen des OPC UA Beispielclients. Diese Klassen realisieren die Funktionalität der Oberfläche und verwenden die Klassen der UA Client API. Auf den folgenden Seiten werden die einzelnen Klassen im Detail erläutert.

Abbildung 4-6



Copyright © Siemens AG 2017 All rights reserved

Klasse MainForm

Die in der folgenden Tabelle beschriebene Klasse **MainForm** implementiert die Funktionalität des Hauptdialoges der Client Applikation.

Der Klassenname entspricht dem Dateinamen im Projekt UAClient.

Tabelle 4-7

Methode	Funktionalität
MainForm	Konstruktor der Klasse.
Connect	Implementiert die Funktionalität, um die Verbindung mit dem Server aufzubauen und initialisiert das Browse Control.
Disconnect	Baut die Verbindung zum Server ab.

Klasse BrowseControl

Die in der folgenden Tabelle beschriebene Klasse **BrowseControl** implementiert die Funktionalität für das Anzeigen des Adressraums des Servers.

Der Klassenname entspricht dem Dateinamen im Projekt UAClient.

Tabelle 4-8

Methode	Funktionalität
BrowseControl	Konstruktor der Klasse.
Browse	Navigiert durch die Baumansicht ausgehend vom selektierten Knoten.

Klasse AttributeListControl

Die in der folgenden Tabelle beschriebene Klasse **AttributeListControl** implementiert die Anzeige der Werte aller Attribute eines Knotens im Adressraum.

Der Klassenname entspricht dem Dateinamen im Projekt UAClient.

Tabelle 4-9

Methode	Funktionalität
AttributeListControl	Konstruktor der Klasse.
ReadAttributes	Liest die Werte der Attribute eines Knotens und zeigt sie in einer Liste an.

Copyright © Siemens AG 2017 All rights reserved

Klasse MonitoredItemsControl

Die in der folgenden Tabelle beschriebene Klasse **MonitoredItemsControl** implementiert die Erzeugung einer Subscription und von MonitoredItems und die Anzeige der Werte bzw. des Status der Monitored Items.

Der Klassenname entspricht dem Dateinamen im Projekt UAClient.

Tabelle 4-10

Methode	Funktionalität
MonitoredItemsControl	Konstruktor der Klasse.
ClientApi_ValueChanged	Callback Funktion für die Data Change Events von der Client API. Der neue Wert, Zeitstempel und Status wird in die entsprechende Zeile eingetragen. Dafür wird das Client Handle auf die Objekt gecastet, welches die Zeile repräsentiert.
MonitoredItems_DragDrop	In dieser Funktion wird eine NodeId, die über Drag&Drop auf das Control gezogen wird, als Data Monitored Item angelegt. Ist noch keine Subscription vorhanden, wird erst eine Subscription erzeugt.

Klasse WriteValuesDialog

Die in der folgenden Tabelle beschriebene Klasse **WriteValuesDialog** implementiert das Anzeigen und das Ändern der aktuellen Werte einer oder mehrerer Variablen.

Der Klassenname entspricht dem Dateinamen im Projekt UAClient.

Tabelle 4-11

Methode	Funktionalität
WriteValuesDialog	Konstruktor der Klasse.
WriteValues	Schreibt den Wert einer oder mehrerer Variablen.
UpdateCurrentValues	Aktualisiert die Werte im Dialog durch einen Aufruf von Read.

Verwendung der Client API im Beispiel

In der folgenden Tabelle sind die Dateien und Funktionen aufgelistet in denen die Client API verwendet wird.

Tabelle 4-10

Client API	Verwendet in
Klasse Discovery	
FindServers	MainForm.cs in der Funktion UrlCB_DropDown
GetEndpoints	MainForm.cs in der Funktion UrlCB_DropDown
Klasse Server	
Connect	MainForm.cs in der Funktion Connect
Disconnect	MainForm.cs in der Funktion Disconnect
Browse	BrowseControl.cs in der Funktion Browse
Read	AttributeListControl.cs in der Funktion ReadAttributes
ReadValues	WriteValuesDialog.cs in der Funktion UpdateCurrentValues
WriteValues	WriteValuesDialog.cs in der Funktion WriteValues
AddSubscription	MonitoredItemsControl.cs in der Funktion MonitoredItems_DragDrop
ModifySubscription	MonitoredItemsControl.cs in der Funktion MonitoringMenu_PublishingInterval_Click
RemoveSubscription	MonitoredItemsControl.cs in der Funktion RemoveSubscription
Klasse Subscription	
AddDataMonitoredItem	MonitoredItemsControl.cs in der Funktion MonitoredItems_DragDrop
ModifyMonitoredItem	MonitoredItemsControl.cs in der Funktion MonitoringMenu_SamplingInterval_Click
RemoveMonitoredItem	MonitoredItemsControl.cs in der Funktion MonitoringMenu_RemoveItems_Click

Copyright © Siemens AG 2017 All rights reserved

4.3.3 Sequenzdiagramme

Verbindung zum OPC UA Server aufbauen und beenden

Das folgende Sequenzdiagramm zeigt die Abläufe, die notwendig sind, um die Verbindung zum OPC UA Server aufzubauen. Durch Anklicken der **Combobox Endpoints** wählt der Benutzer zunächst einen verfügbaren Endpoint aus.

Der Verbindungsaufbau kann über den **Button Connect** in der Oberfläche oder über das **Menü Server** gestartet werden. Nach einem erfolgreichen Verbindungsaufbau erscheint auf dem **Button Connect** die Aufschrift **Disconnect.** Das Sequenzdiagramm zeigt auch die Abläufe, die durch die Aktion "Disconnect Server" über den **Button Disconnect** angestoßen werden.

Abbildung 4-7

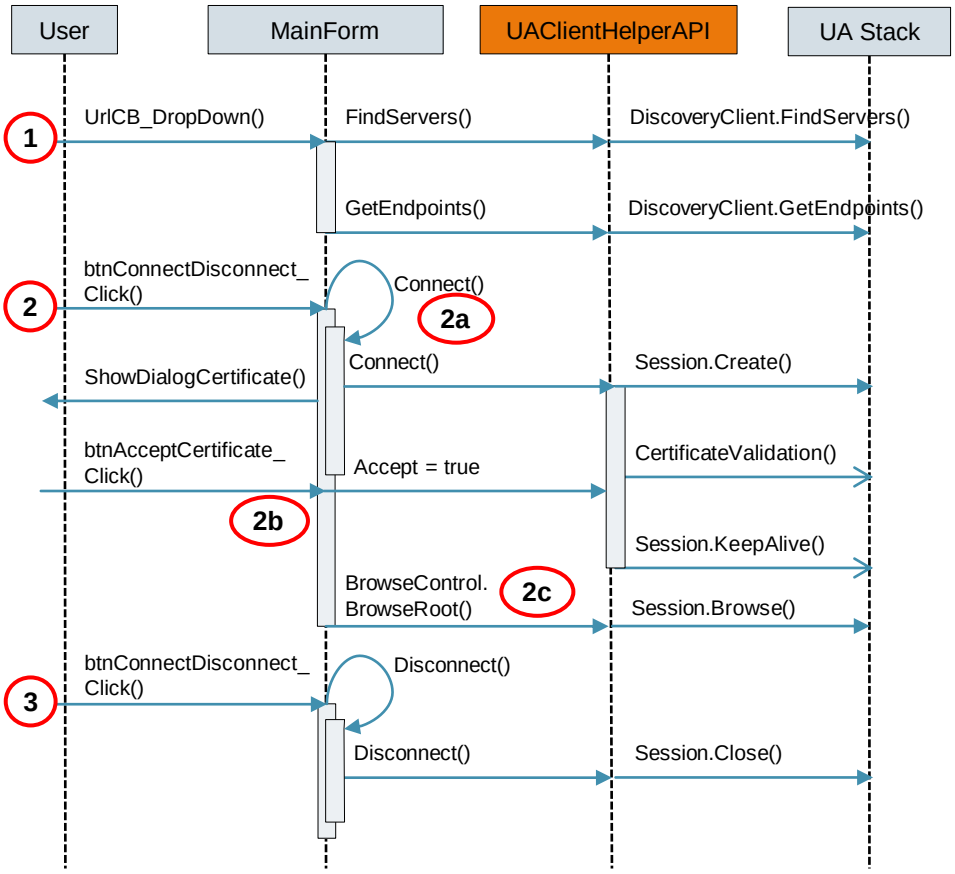


Tabelle 4-12

Nr.	Beschreibung
1	Beim Öffnen der Auswahlliste wird ein ClientAPI::DiscoveryClient Objekt erzeugt und dort die Methoden **FindServers** und **GetEndpoints** aufgerufen.
2	Durch die Benutzeraktion "Connect" wird die Methode **ConnectDisconnect_Click** am Objekt MainForm aufgerufen. In dieser Methode werden die Aktionen zum Verbindungsaufbau mit dem Server durchgeführt. Im ersten Schritt wird das ClientAPI Objekt erzeugt und dort Methode **Connect** aufgerufen (2a). Damit wird die Verbindung zum - durch die URL definierten - OPC UA Server aufgebaut. Wird ein Zertifikat als nicht vertrauenswürdig eingestuft, kann der Benutzer dieses dennoch akzeptieren und der Verbindungsaufbau wird fortgesetzt (2b). Im Fall einer bestehenden Verbindung wird in einem zweiten Schritt die Methode **BrowseRoot** am ClientAPI Objekt aufgerufen (2c). Dadurch wird die erste Ebene des Adressraums des OPC UA Servers im BrowseControl angezeigt.
3	Durch die Benutzeraktion "Disconnect Server" wird ebenfalls die Methode **ConnectDisconnect_Click** am Objekt MainForm aufgerufen. Im Fall einer bereits existierenden Verbindung wird in dieser Methode die Methode **Disconnect** am ClientAPI Objekt aufgerufen. Damit wird die Verbindung zum OPC UA Server beendet.

Copyright © Siemens AG 2017 All rights reserved

Lesen von Attributen

Das folgende Sequenzdiagramm zeigt die Abläufe beim Lesen von Attributwerten.

Abbildung 4-8

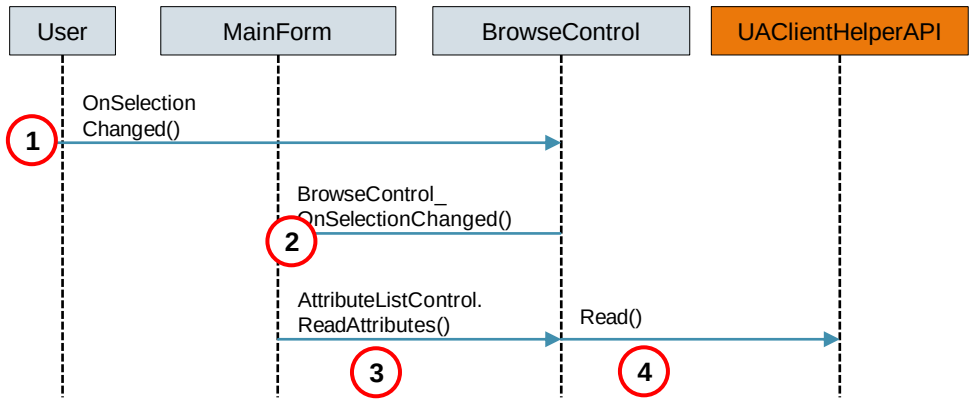


Tabelle 4-13

Nr.	Beschreibung
1	Wird ein Knoten im Baum des BrowsControls selektiert, wird ein OnSelectionChanged Event ausgelöst.
2	Der Event wird an die MainForm weiter geleitet.
3	Durch den Event wird die Methode **ReadAttributes** des AttributeListControl aufgerufen.
4	Dort wird Read am ClientAPI Objekt aufgerufen und die Werte werden angezeigt.

Copyright © Siemens AG 2017 All rights reserved

Schreiben von Werten

Das folgende Sequenzdiagramm zeigt die Abläufe beim Schreiben von Werten von Variablen.

Abbildung 4-9

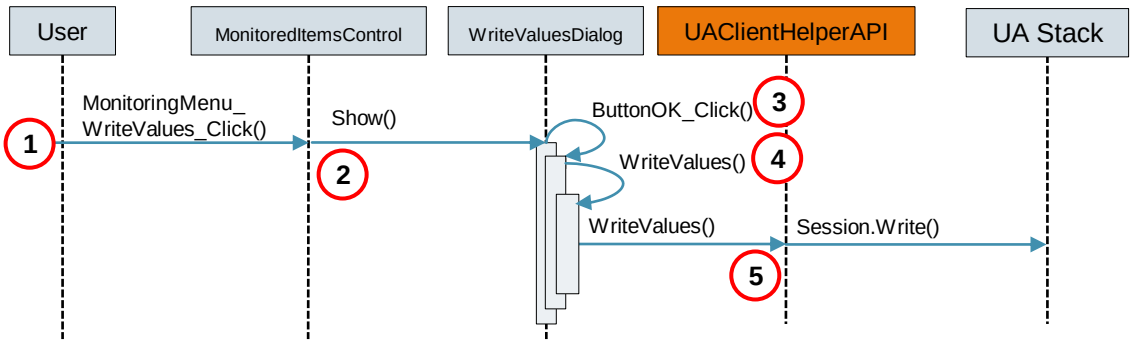


Tabelle 4-14

Nr.	Beschreibung
1+2	Beim Anklicken des Eintrags "WriteValues" im Kontextmenü von **MonitoredItemsControl** (1) wird der Dialog "Write Values" geöffnet (2). Dort erscheinen die aktuellen Werte der zuvor im **MonitoredItemsControl** selektierten Variablen.
3+4	Betätigt der Benutzer nach dem Eintragen neuer Werte den **Ok Button** (Aktion **buttonOk_Click**), wird am WriteValuesDialog Objekt **WriteValues** aufgerufen.
5	Dort wird am ClientAPI Objekt wiederum **WriteValues** aufgerufen.

Copyright © Siemens AG 2017 All rights reserved

Variablen zur Beobachtung anmelden

Das folgende Sequenzdiagramm zeigt die Abläufe beim Anmelden von Variablen für die Beobachtung von Wertänderungen.

Abbildung 4-10

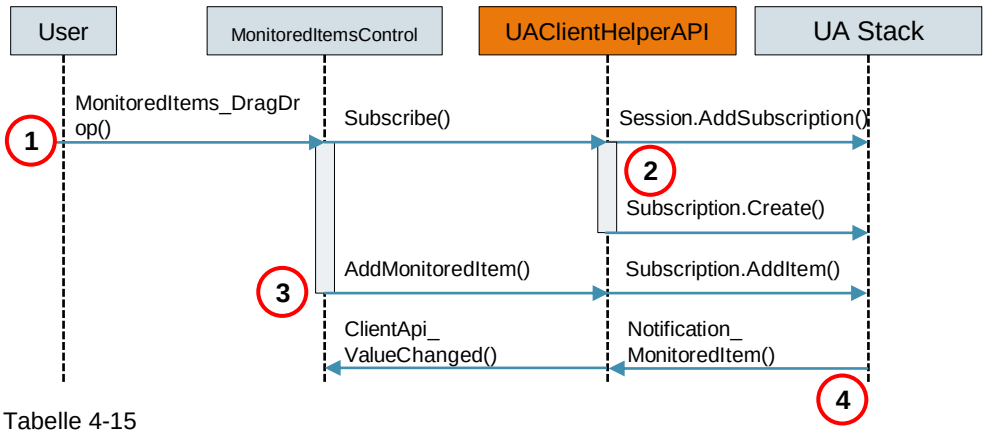


Tabelle 4-15

Nr.	Beschreibung
1	Eine Drag&Drop-Aktion mit dem MonitoredItemsControl als Ziel prüft zunächst, ob bereits eine Subscription existiert. Wenn nicht, wird **Subscribe** an der Client API aufgerufen.
2	Dabei wird ein Subscription Objekt erzeugt und anschließend ein **Create** am Subscription Objekt aufgerufen.
3	Im nächsten Schritt wird **AddMonitoredItem** am Client API Objekt aufgerufen.
4	Ab jetzt werden Wertänderungen vom Server empfangen und an das MonitoredItemsControl weitergeleitet, um dort angezeigt zu werden.

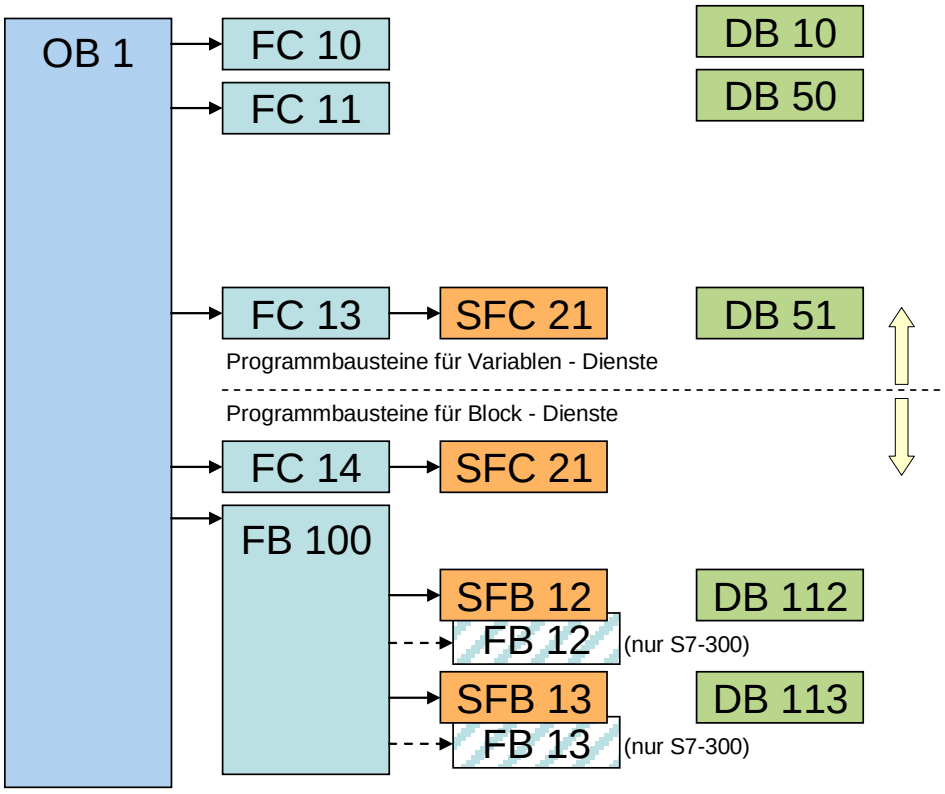
Copyright © Siemens AG 2017 All rights reserved

4.4 S7 Programm

Übersicht

Das S7 Programm teilt sich im Wesentlichen in zwei Teile. Zunächst werden die dynamischen Daten für die Variablendienste simuliert, anschließend werden die Sendedaten simuliert und im FB 100 werden die Blockdienste aufgerufen.

Abbildung 4-11



ACHTUNG	**Das S7-Programm lässt sich auf Grund der verwendeten absoluten Adressierung nicht zusammen mit optimierten Bausteinen verwenden!**

Hinweis Das S7-Programm für die S7-1500 enthält lediglich den Datenbaustein "StaticDataTypesOpt" (DB1) in dem Variablen optimiert abgelegt sind. Das S7-Programm dient nur dazu, Variablen-Nodes für den S7-Opt-Server (Port 55105) bereit zu stellen.

Copyright © Siemens AG 2017 All rights reserved

Simulation dynamische Daten

Die folgende Tabelle gibt eine kurze Übersicht über die Programmteile und ihre Funktion zur Datensimulation. Auf Details wird hier bewusst verzichtet, weitere Kommentare befinden sich im AWL-Code.

Tabelle 4-16

Baustein	Anmerkung
OB1	Cyclic Main hier wird zunächst ein variabler Timer gesetzt, mit dessen Intervall die weiteren Programmfunktionen aufgerufen werden. Die Datenänderungsgeschwindigkeit kann über DB10 Byte 0 gesetzt werden.
FC10	ChangeDateAndTime inkrementiert einen Datumswert sowie Uhrzeit im DB51.
FC11	ChangeSimpleTypes inkrementiert die Daten des DB51. 8Bit Typen werden mit +1, 16Bit Typen mit +100 und 32Bit Typen mit +1000 inkrementiert.
FC13	ChangeString inkrementiert einen String der Länge 10 im DB51.
DB10	SimulationConfiguration enthält globale Variablen zur Konfiguration der Datensimulation.
DB50	StaticDataTypes enthält einfache Datentypen, denen symbolische Namen gegeben wurden. Die Werte sind mit maximalem Wertebereichsendwert vorinitialisiert.
DB51	DynamicDataTypes enthält einfache Datentypen, denen symbolische Namen gegeben wurden. Die Werte werden mit den Funktionen FC10 bis FC13 entsprechend ihrer Wertebereiche inkrementiert.
SFC21	FILL Hilfsfunktion, um Datenbereiche mit Werten zu befüllen, Speicherinitialisierung.

Copyright © Siemens AG 2017 All rights reserved

Blockorientierte Daten

Die S7-Kommunikation bei der S7-300 wird über FBs (ladbare Funktionsbausteine) realisiert und nicht anhand von SFBs (integrierten Systemfunktionsbausteinen) wie bei der S7-400. Wenn Sie im S7-Programm der S7-300 jedoch einen SFB statt FB aufrufen, liefert der Baustein einen "ERROR" und gibt den "STATUS = 27" aus. Dieser Status weist darauf hin, dass der Funktionsbaustein für die S7-Kommunikation auf der S7-300 nicht vorhanden ist. Die Kommunikations-FBs für die S7-300 befinden sich in der Task Card im Reiter Anweisungen Unter Kommunikation > S7 Kommunikation.

Die folgende Tabelle gibt eine kurze Übersicht über die Programmteile und ihre Funktion bezüglich BSEND/BRCV. Auf Einzelheiten wird hier bewusst verzichtet, weitere Kommentare befinden sich im AWL-Code.

Tabelle 4-17

Baustein	Anmerkung
OB1	Cyclic Main Aufruf des Sendebausteins (SFB12) für BSEND und des Empfangsbausteins (SFB13) für BRCV über den Funktionsblock 100. Für S7-300 werden FB12 und FB13 aus der Bibliothek verwendet da die S7-300 über ladbare FBs und nicht über Systemfunktionen kommuniziert.
FC14	ChangeSendData inkrementiert ein Byte, welches dann auf den gesamten Sendepuffer kopiert wird
FB100 + DB100 (Instanz-DB)	InvokeBSENDandBRCV ruft die echten Kommunikationsbausteine auf und versorg deren Parameter
DB112	SendData Datenbaustein mit 4096 Bytes Länge, der den Sendebaustein übergeben wird
DB113	ReceiveData Datenbaustein mit 4096 Bytes Länge, der den Empfangsbaustein übergeben wird
SFB12 + DB12 (Instanz-DB)	BSEND Der Sendebaustein übergibt den Sendepuffer an den Kommunikationsprozessor (CP), der ihn dann entsprechend RID und VerbindungsID an den Kommunikationspartner sendet.
FB12	BSEND (nur S7-300)
SFB13 + DB13 (Instanz-DB)	BRCV Der Empfangsbaustein holt das zuletzt empfangene Datenpaket entsprechend RID und VerbindungsID vom Kommunikationsprozessor (CP) ab und legt es in den Empfangspuffer.
FB13	BRCV (nur S7-300)
SFC21	FILL Hilfsfunktion um Datenbereiche mit Werten zu befüllen, Speicherinitialisierung.

Die Programmlogik sendet bzw. empfängt Datenblöcke und versorgt die entsprechenden Parameter über den FB100. Wenn größere Datenpakete gesendet werden ist ein mehrfacher Aufruf des BSEND bzw. BRCV Bausteins erforderlich. Dieser Mehrfachaufruf wird vom FB100 durchgeführt.

Copyright © Siemens AG 2017 All rights reserved

Hinweis Werden Sende- und Empfangsbaustein zyklisch aufgerufen kann zum einen eine hohe Kommunikationslast entstehen und zum anderen besteht die Gefahr, dass die Datenpuffer überschrieben werden, bevor sie von der Gegenseite verarbeitet werden konnten. Die Programmlogik sollte deshalb eine Flusskontrolle beinhalten, um die Datenkonsistenz sicherzustellen. Hierzu sind der Parameter DONE (fertig) und NDR (neue Daten empfangen) zu verwenden.

Um Daten zwischen zwei S7-300 Stationen über eine projektierte S7-Verbindung austauschen zu können, müssen im S7-Programm Kommunikationsfunktionen aufgerufen werden. Der Baustein FB12 "BSEND" dient zum Senden von Daten und der Baustein FB13 "BRCV" zum Empfangen von Daten.

Hierbei muss die S7-Verbindung beidseitig projektiert werden, da die S7-Kommunikation mittels FB12 "BSEND" und FB13 "BRCV" auf dem Client-Client Prinzip basiert.

Copyright © Siemens AG 2017 All rights reserved

5 Konfiguration und Projektierung

5.1 Konfiguration der SIMATIC S7 Stationen

Allgemein

Es wird vorausgesetzt, dass alle Hardware- und Softwarekomponenten erfolgreich installiert und entsprechend verkabelt wurden.

Hier finden Sie eine Übersicht der im Beispielprojekt verwendeten IP-Adressen:

Tabelle 5-1

Addresse	Station	Anmerkung
192.168.0.200	PC OPC Server Station	
192.168.0.2	S7-300, CP 343-1	
192.168.0.3	S7-400, CP 443-1 EX/GX 20	optional
192.168.0.4	S7-1500, CPU 1516-3 PN/DP	optional
192.168.0.1	Laptop OPC Client Station	optional

Copyright © Siemens AG 2017 All rights reserved

Besonderheit S7-1200 ab FW V4.0 und S7-1500

Wenn Sie optimierte Datenbausteine in einer S7-1200 Station ab FW V4.0 oder einer S7-1500 Station und verwenden, dann müssen Sie den OPC Server V12 oder höher einsetzen, um optimiert auf die Datenbausteine der Steuerungen zuzugreifen.

STEP 7 legt dann automatisch "optimierte S7-Verbindungen" an, da der Standardzugriff auf optimierte Datenbausteine per OPC nicht möglich ist.

Der OPC Client muss dann über den Port 55105 auf den SimaticNET.S7OPT-Server zugreifen.

Wenn Sie keine optimierten S7-Verbindungen verwenden wollen, dann müssen die folgenden Bedingungen erfüllt sein:

1. Verwenden des OPC-Servers maximal in der Version 8.2.
2. Verwenden nicht-optimierter Datenbausteine.

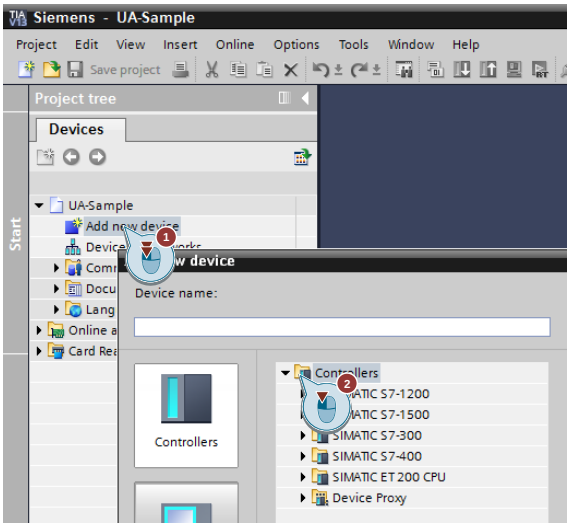
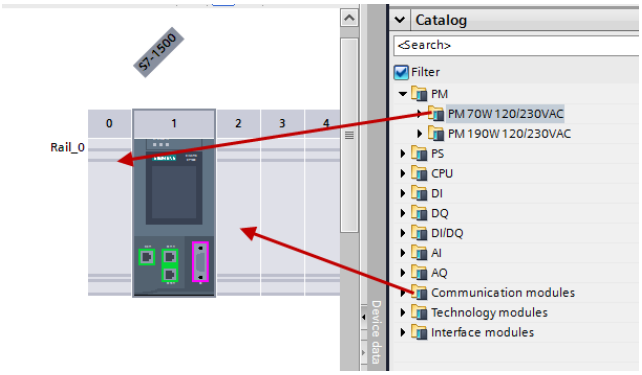
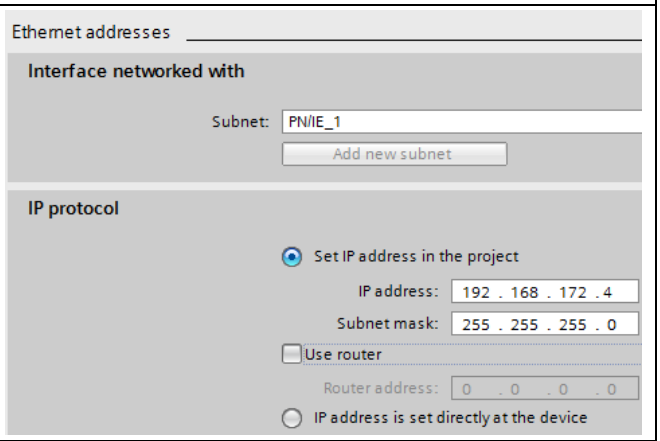
Vorgehen

Die folgenden Projektierungs- und Konfigurationsschritte der SIMATIC S7-Stationen beschreiben beispielhaft die Vorgehensweise. Nehmen Sie selbstständig Anpassungen entsprechend Ihrer Hardware vor.

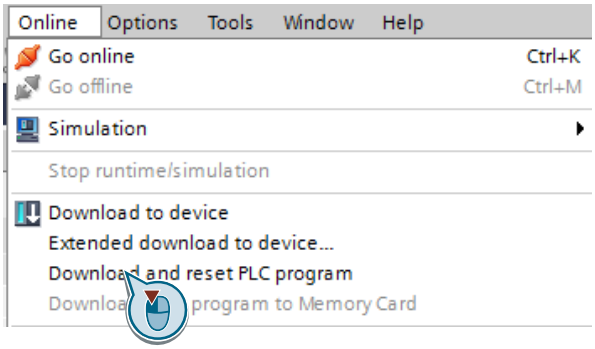
Hinweis Nach dem Speichern und Übersetzen werden alle Konfigurations- und Projektierungsinformationen überschrieben.

Die folgende Tabelle zeigt die Konfiguration und Projektierung der SIMATIC S7-Stationen.

Tabelle 5-2

Nr.	Aktion	Anmerkung
1.	STEP 7 V14 öffnen und neues Projekt anlegen.	Hier wurde der Name "UA-Sample" verwendet.
2.	SIMATIC 400 Station einfügen und Namen vergeben (hier "S7-400"). SIMATIC 300 Station einfügen und Namen vergeben (hier "S7-300"). SIMATIC 1500 Station einfügen und Namen vergeben (hier "S7-1500).	
3.	SIMATIC Stationen in der Gerätesicht (device view) öffnen und evtl.verwendete CPs und andere Komponenten entsprechend der tatsächlich eingesetzten Hardware einfügen.	
4.	Wechseln Sie zum Einstellen der IP-Adresse im Inspektorfenster in den Reiter Eigenschaften. Wählen Sie dort Ethernet Adressen ("Ethernet-addresses"). Einstellen der IP-Adresse und Subnetmaske ein. Anlegen eines Ethernet Netzwerkes. Eine MAC Adresse wird nur dann eingegeben, wenn die Station über ISO Transport Layer 4 kommunizieren soll.	
5.	Wiederholen Sie die Schritte entsprechend für alle SIMATIC Stationen.	
6.	S7-Verbindungen anlegen	Die Verbindungsprojektierung wird zusammen mit der Konfiguration der PC Station beschrieben (im Kapitel 6.2).

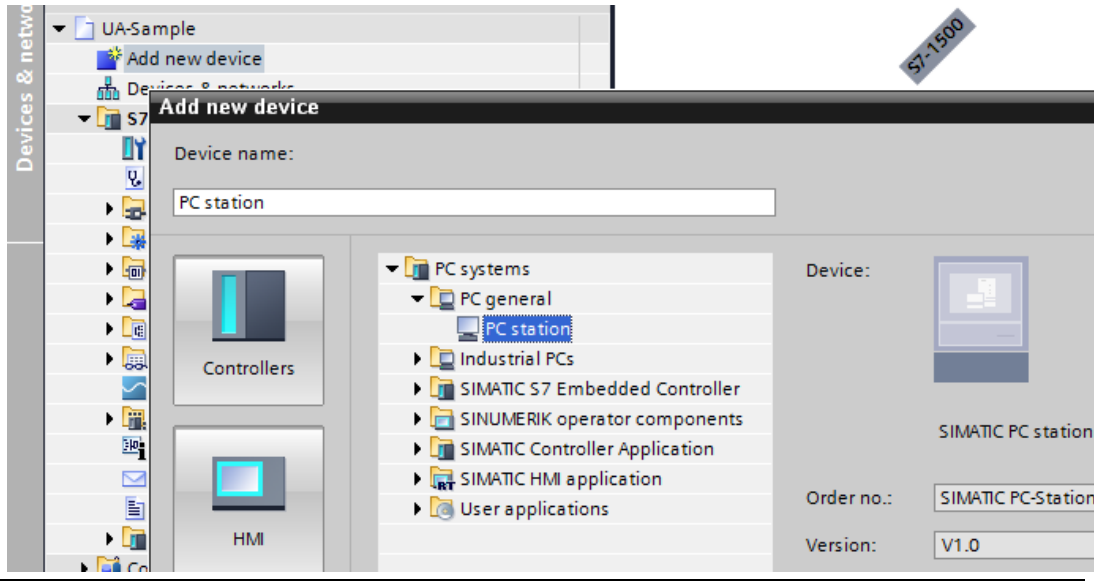
Copyright © Siemens AG 2017 All rights reserved

Nr.	Aktion	Anmerkung
7.	Wenn alle Einstellungen betätigt sind, dann laden Sie die Projektierung mit einem Neustart in die S7-CPUs.	

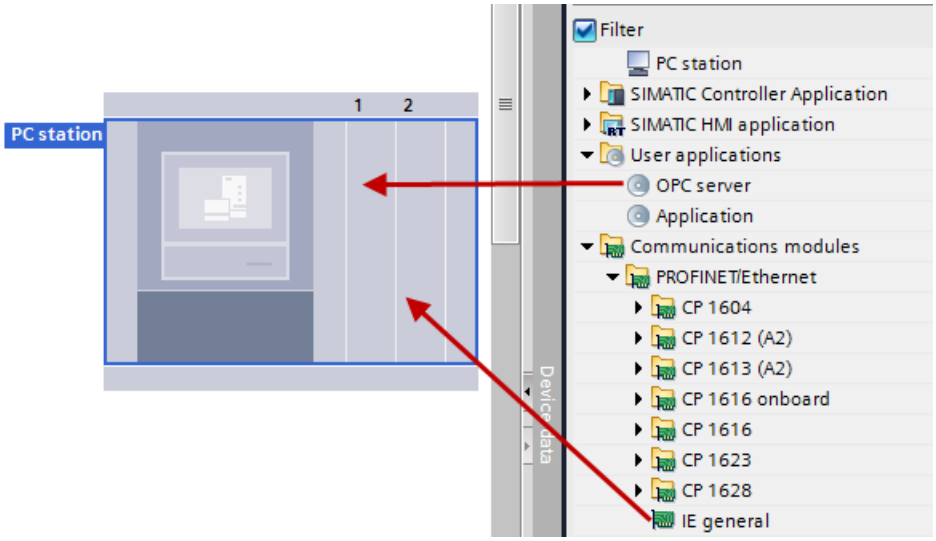
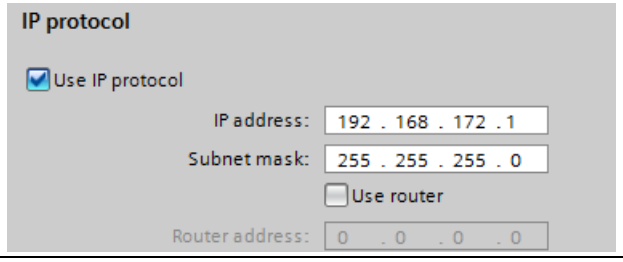
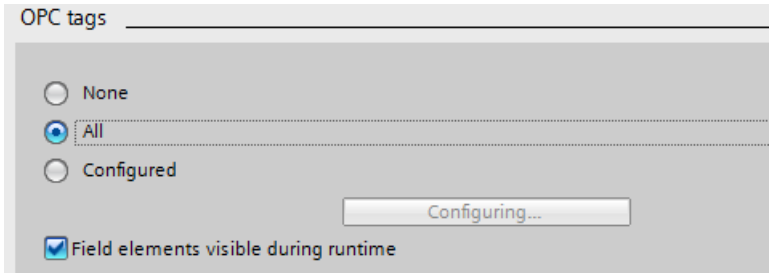
5.2 Konfiguration der OPC Server Station

Die Konfiguration und Projektierung der SIMATIC PC Station wird mittels STEP 7 vorgenommen und schrittweise beschrieben. Alternativ kann auch mit Hilfe des NCM-PC Softwarepakets eine Projektierung vorgenommen werden. Die Vorgehensweise ist identisch, es werden jedoch einseitig projektierte Verbindungen verwendet.

Tabelle 5-3

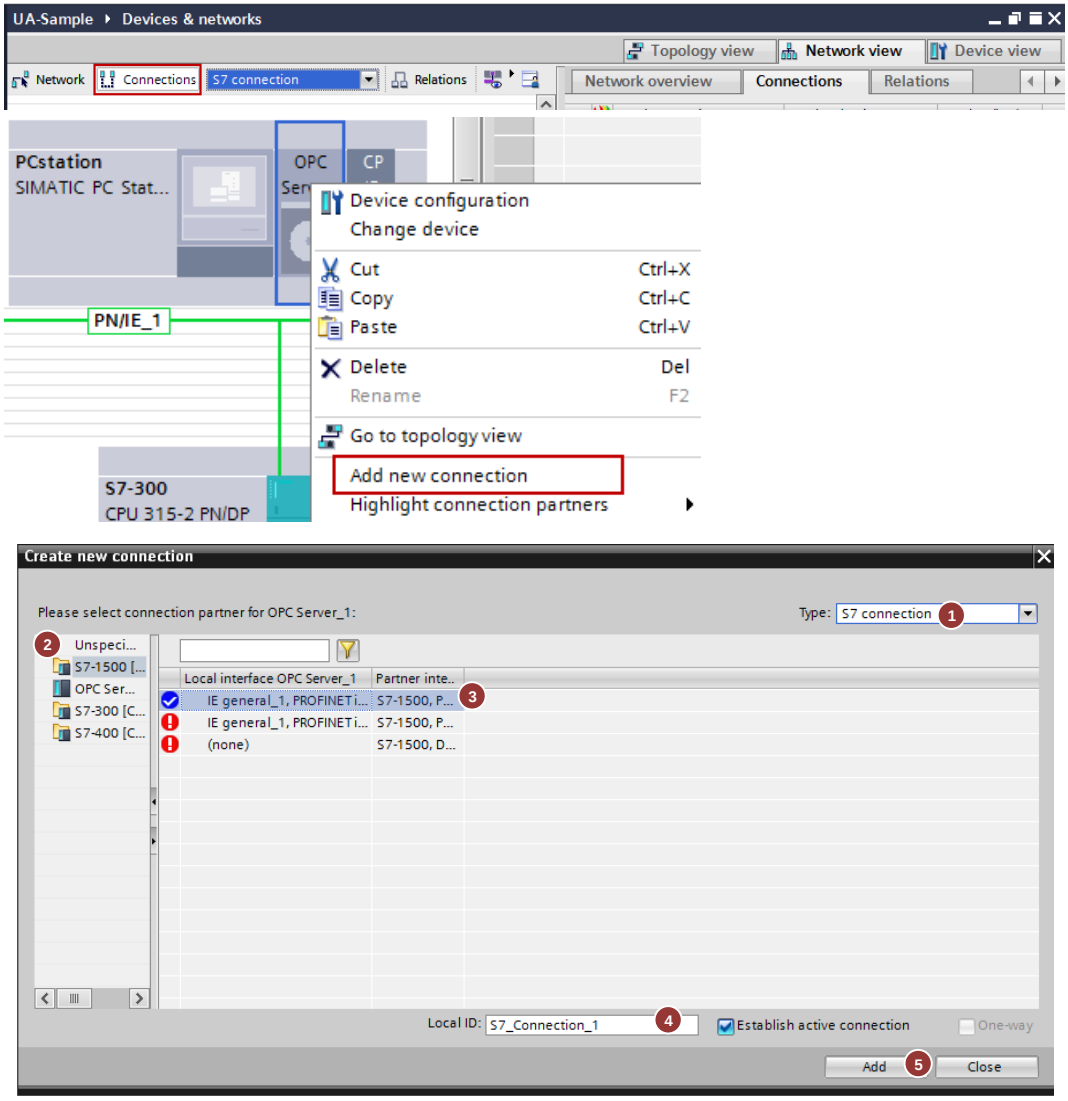
Nr.	Aktion Anmerkung
1.	Starten von STEP 7 V14. Öffnen des zuvor angelegten Projektes "UA-Sample".
2.	SIMATIC PC Station einfügen und Namen vergeben. 

Copyright © Siemens AG 2017 All rights reserved

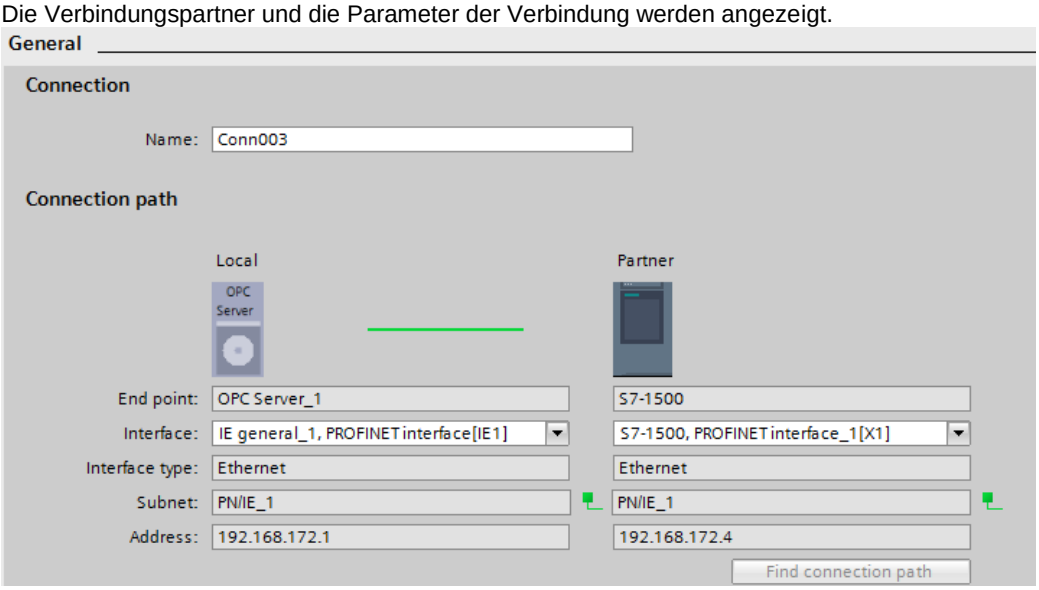
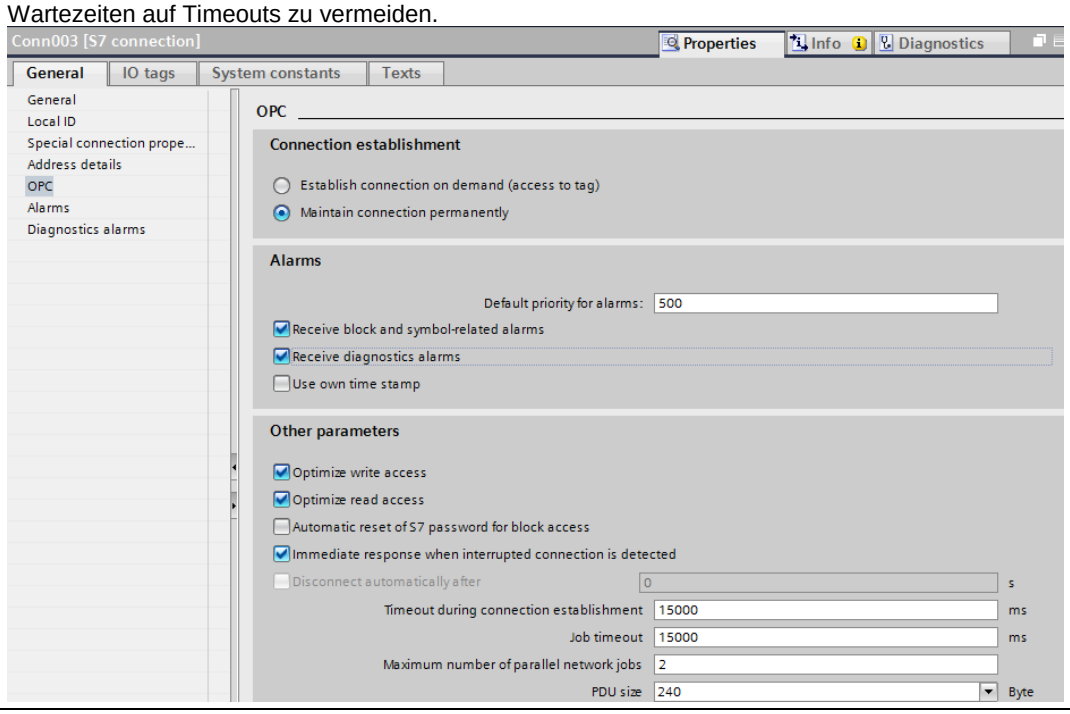
3.	Öffnen Sie die Gerätesicht der PC Station. Fügen Sie dort die Ethernetkarte und den OPC-Server ein. **Hinweis** Wollen Sie eine S7-1200 oder eine S7-1500 CPU mit optimierten Datenbausteinen verwenden, dann müssen Sie den OPC Server V12 oder höher verwenden. Der Steckplatz muss identisch mit dem Index sein, der in der Konfigurationskonsole vergeben wurde, hier Index "2" für Ethernetkarte. Die Applikation "OPC Server" wurde auf Steckplatz "1" gesteckt. 
4.	Für die Ethernetkarte wird in den Eigenschaften eine IP-Adresse (hier "192.168.172.1") vergeben und die Karte wird mit dem Ethernetnetzwerk verbunden. 
5.	In den Eigenschaftsseiten des OPC Servers wird im Register "S7" die Verwendung der symbolischen Adressierung aktiviert. 

Copyright © Siemens AG 2017 All rights reserved

6.	In der Netzsicht werden drei S7-Verbindungen vom OPC Server zu den S7-Steuerungen angelegt. Die Verbindungen werden über Ethernet angelegt.



Copyright © Siemens AG 2017 All rights reserved

7.	S7-Verbindung über Ethernet: Nachdem die S7-Verbindungen angelegt wurden, werden über einen Klick auf die Verbindung und den Reiter Eigenschaften im Inspektorfenster weitere Einstellungen vorgenommen. Nachdem der Verbindungsweg gewählt wurde, kann der Verbindungsname geändert werden (hier "Conn001" für die Verbindung zur S7-300, "Conn002" für die Verbindung zur S7-400 und "Conn003"). Die Verbindungspartner und die Parameter der Verbindung werden angezeigt. 
8.	Im den Eigenschaften der S7-Verbindungen wird das Menü "OPC" selektiert. Hier werden verbindungsspezifische Einstellungen vorgenommen. Der Verbindungsaufbau wird permanent eingestellt, damit die Verbindung gehalten wird, auch wenn gerade nicht kommuniziert wird. Die Verbindung wird für die Übertragung von Alarmen und Diagnose-Events konfiguriert. Weiterhin wird die sofortige Reaktion auf Verbindungsunterbrechungen aktiviert um unnötige Wartezeiten auf Timeouts zu vermeiden. 
9.	Nachdem die Verbindungen angelegt sind, muss die neue Konfiguration in die Stationen geladen werden. Die PC Station kann auch mit der XDB-Datei, wie in Kapitel 6 beschrieben, geladen werden.

Copyright © Siemens AG 2017 All rights reserved

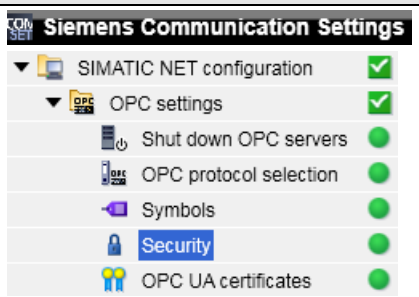
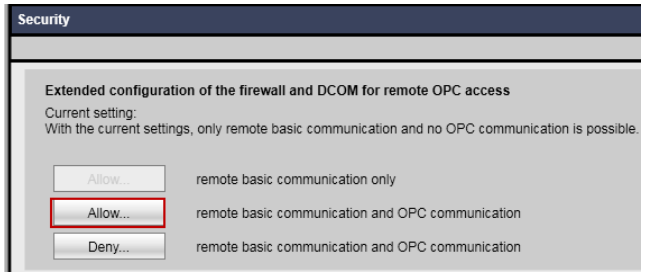
5.3 Konfiguration der OPC UA Security

Die Sicherheitsmechanismen von OPC Unified Architecture werden in verschiedenen Ebenen eingestellt. Es können Verschlüsselung und Signierung der Übertragung sowie Authentifizierung zum Verbindungsaufbau getrennt voneinander eingestellt werden. Nach der Installation des SIMATIC NET OPC UA Servers sind grundsätzlich sichere Verbindungen möglich. Neben dieser verschlüsselten Kommunikation ist auch eine unverschlüsselte Verbindung möglich. Der Server akzeptiert Authentifizierung mit Benutzer und Passwort oder auch den anonymen Verbindungsaufbau. Diese Einstellungen sind "unsicher" und dienen nur der Erleichterung der Inbetriebnahme. Der OPC UA Server kann so konfiguriert werden, dass er ausschließlich eine verschlüsselte Übertragung mit Benutzer-Authentifizierung akzeptiert.

5.3.1 OPC UA Remote Kommunikation

Alle serverseitig erforderlichen Einstellungen bezüglich der Windows Firewall lassen sich einfach mit dem Konfigurationstool "PC Station einstellen" (Configuration Console) einstellen und auch wieder entfernen.

Tabelle 5-4

Nr.	Aktion	Anmerkung
1.	Starten Sie das Konfigurationstool: "Kommunikations-Einstellungen" ("`Communication Settings`") und selektieren Sie das Menü "SIMATIC Net-Konfiguration > OPC-Einstellungen > `Sicherheit`" ("SIMATIC NET configuration > OPC settings > Security")	
2.	Mit einem einzelnen Knopfdruck werden alle erforderlichen Einstellungen in der Firewall vorgenommen um eine remote Kommunikation zu erlauben bzw. auch wieder zu sperren.	

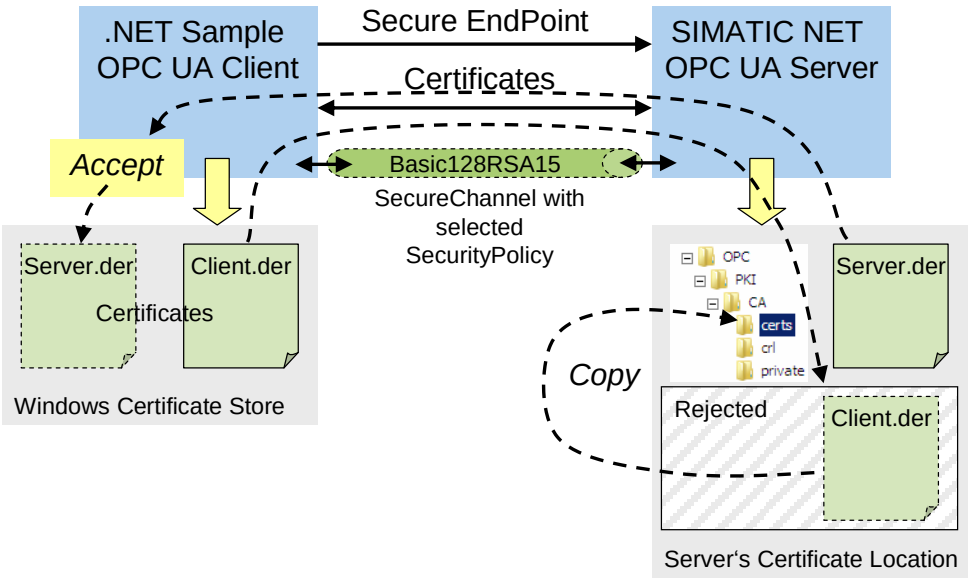
Hinweis Beachten Sie, dass an der Firewall des PC "OPC Client Station" auch eine Ausnahme für die Applikation und den TCP-Port (4845) eingetragen werden muss.

Copyright © Siemens AG 2017 All rights reserved

5.3.2 Zertifikatsspeicher

Beim Aufbau einer sicheren Verbindung zwischen OPC UA Client und OPC UA Server werden Zertifikate ausgetauscht. Beide Applikationen müssen das entsprechende Zertifikat der Gegenseite prüfen und akzeptieren, damit ein Verbindungsaufbau zustande kommt.

Abbildung 5-1



Der OPC UA Client dieses Beispiels verwendet die **Windows Zertifikatsverwaltung** (**Windows Certificate Store**). Hier befindet sich das öffentliche Zertifikat des Clients.

Die Beispiel Clients erzeugen beim ersten Start in dieser Zertifikatsverwaltung ein öffentliches Zertifikat (UA Test Client). Der Testclient muss hierzu beim ersten Start mit Administrationsrechten gestartet werden.

Beim Verbindungsaufbau einer verschlüsselten Verbindung tauschen Server und Client ihre Zertifikate aus. Der Client zeigt das Zertifikat an und der Anwender muss diesem Zertifikat vertrauen. Durch das Akzeptieren wird das Serverzertifikat in der Windows Zertifikatsverwaltung hinterlegt.

Der OPC UA Server verwendet ein **eigenes** Zertifikatsverzeichnis und ist unabhängig von der Windows Zertifikatsverwaltung. Aus Sicherheitsgründen lehnt der OPC UA Server zunächst jedes Zertifikat eines unbekannten Client ab und speichert dieses in einem "Quarantäne-Ordner" (rejected-folder). Wie bei anderen Serverdiensten muss ein Administrator dieses Client-Zertifikat in die Liste der vertrauenswürdigen Zertifikate kopieren, um dem entsprechenden Client den Zugriff auf den Server zu gestatten. Der Ort, an dem der OPC UA Server seine eigenen und die Zertifikate von OPC UA Clients speichert und verwaltet, ist das Datenverzeichnis des OPC UA Servers.

Copyright © Siemens AG 2017 All rights reserved

C:\Documents and settings\All Users\Application Data\Siemens\OPC\PKI\CA

Hier befinden sich drei Unterordner mit folgendem Inhalt:

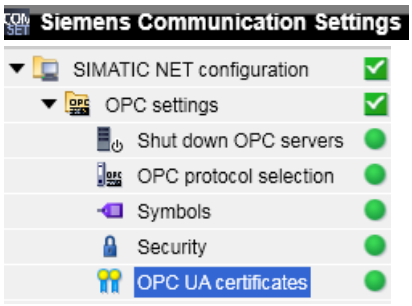
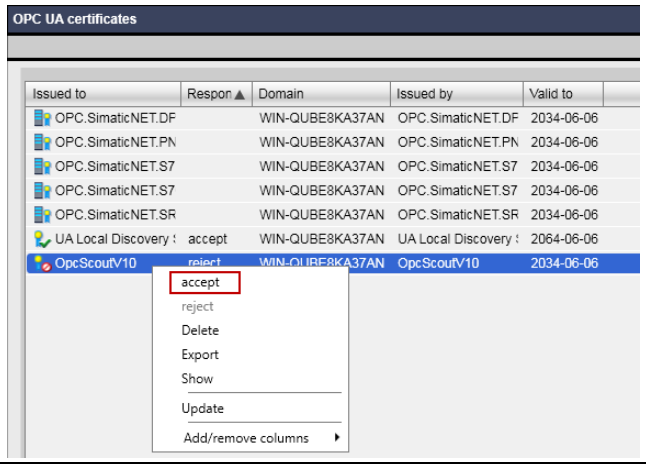
- **\certs**
  enthält das öffentliche Zertifikat des Servers sowie alle vertrauenswürdigen Zertifikaten von Clients. In diesen Ordner müssen öffentliche Zertifikate von OPC UA Clients kopiert werden damit der Server sie akzeptiert.
- **\crl**
  enthält eine Datei mit einer Liste von nicht vertrauenswürdigen Zertifikaten, die sogenannte "RevocationList"
- **\private**
  enthält das private Zertifikat des OPC UA Servers. Dieses Zertifikat darf niemandem zugänglich gemacht werden.

Der Server legt selbstständig unterhalb von \certs\ den Ordner \reject\ an und speichert alle unbekannten Client-Zertifikate zunächst in diesem "Quarantäne-Ordner". Durch einfaches "Verschieben" der Datei kann das Zertifikat vertrauenswürdig gemacht werden.

Konfiguration Server mit SIMATIC NET DVD ab V8.2

Ab der SIMATIC NET CD V8.0 ist dies mit dem Konfigurationstool "Siemens Kommunikations-Einstellungen" möglich.

Tabelle 5-5

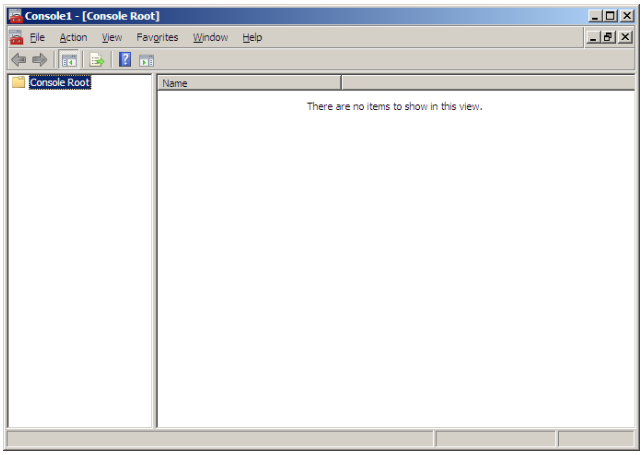
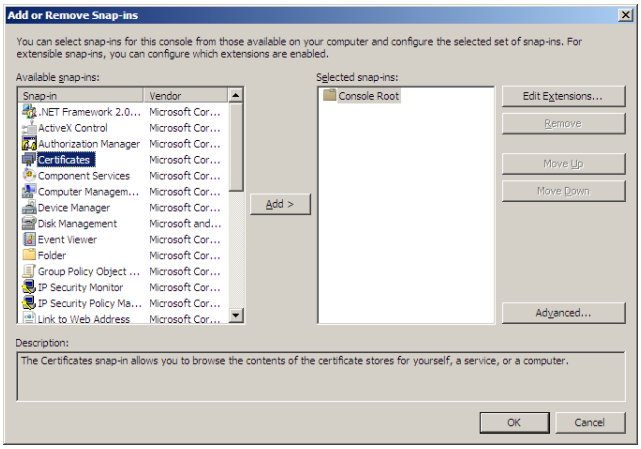
Nr.	Aktion	Anmerkung
1.	Starten Sie das Konfigurationstool: "`Siemens Kommunikations-Einstellungen`" und selektieren Sie "SIMATIC Net-Konfiguration > OPC-Einstellungen > `OPC-UA-Zertifikate`" ("SIMATIC NET configuration > OPC settings > OPC-UA-certificates")	Auf dem Server-PC. 
2.	Hier finden Sie alle öffentlichen UA Zertifikate die der Server kennt. Mit Rechtsklick auf ein ausgewähltes Zertifikat können Sie dieses akzeptieren.	

Copyright © Siemens AG 2017 All rights reserved

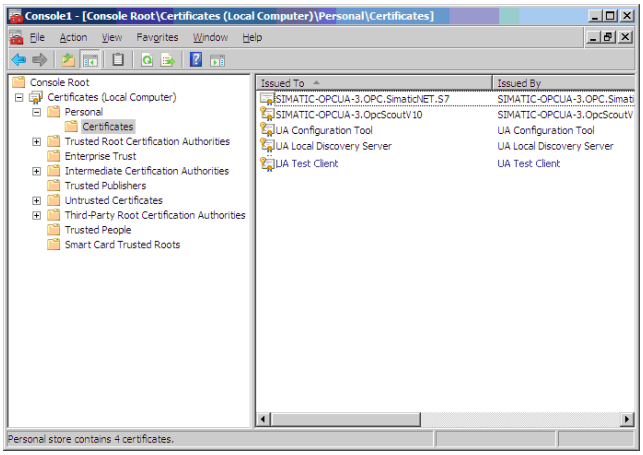
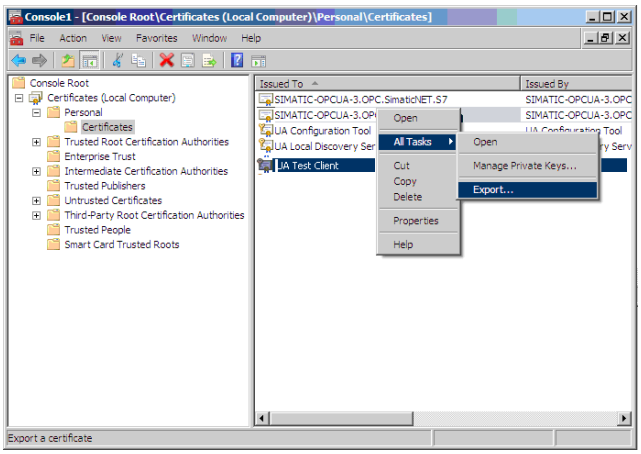
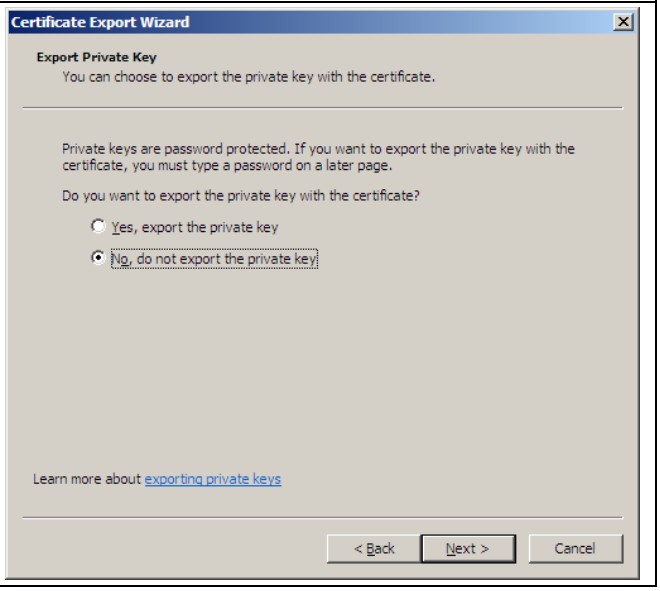
Konfiguration Server mit SIMATIC NET CD 2008 (V7.1)

In der SIMATIC NET CD 2008 (V7.1 oder niedriger) müssen vertrauenswürdige Client Zertifikate manuell in das Zertifikatsverzeichnis des Servers kopiert werden. Dazu werden sie zunächst auf dem Client PC aus der Windows-Zertifikatsverwaltung exportiert.

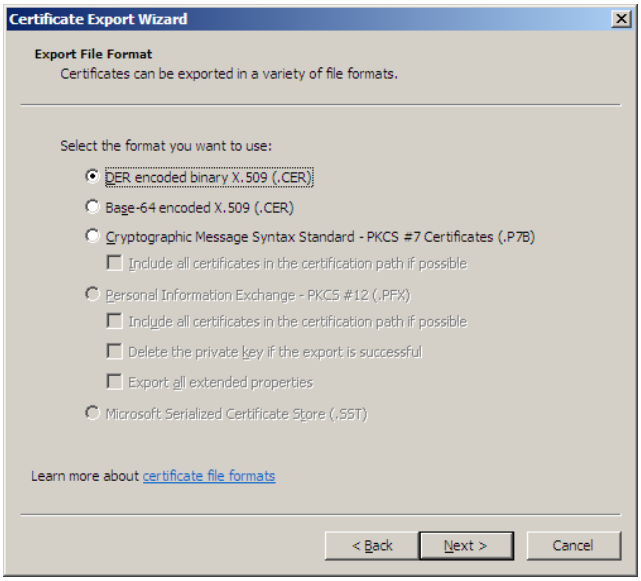
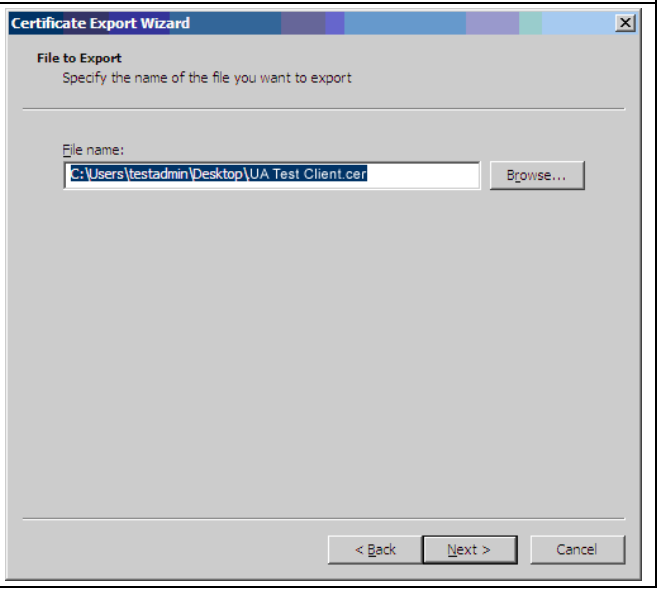
Tabelle 5-6

Nr.	Aktion	Anmerkung
3.	Öffnen Sie die Windows Zertifikatsverwaltung in der Management Konsole.	Auf dem Client PC "OPC Client Station" Start→ Ausführen→ "mmc" 
4.	Im Menu Datei wählen Sie "SnapIn hinzufügen/entfernen" und wählen Sie anschließend "Add>".	

Copyright © Siemens AG 2017 All rights reserved

Nr.	Aktion	Anmerkung
5.	Wählen Sie die Zertifikate für das "Computerkonto" des lokalen Computers.	
6.	Wählen Sie das zu exportierende Zertifikat (Rechtsklick→ Alle Aufgaben → exportieren).	In diesem Beispiel das Zertifikat **UA Test Client**, das von den OPC UA Beispiel Clients beim ersten Start erzeugt wurde. Der Export Wizard wird gestartet 
7.	Der private Schlüssel wird NICHT exportiert, sondern nur der "öffentliche" Schlüssel	

Copyright © Siemens AG 2017 All rights reserved

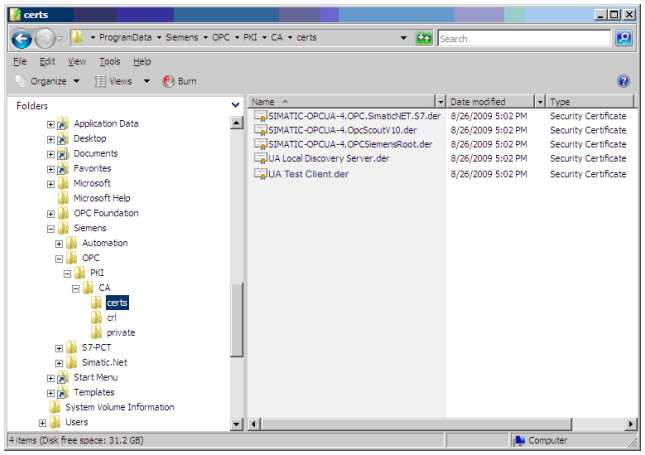
Nr.	Aktion	Anmerkung
8.	Wählen Sie **DER** Kodierung	
9.	Geben Sie einen Dateinamen an und speichern Sie die Zertifikatsdatei ab.	

Copyright © Siemens AG 2017 All rights reserved

Manueller Import des Client Zertifikats beim Server

In der SIMATIC NET CD 2008 (V7.1) müssen vertrauenswürdige Client Zertifikate direkt in das Zertifikatsverzeichnis des Servers kopiert werden. Dazu wird die auf dem Client PC aus der Windows-Zertifikatsverwaltung exportierte Zertifikatsdatei umbenannt und auf den Server PC kopiert (ins Zertifikatsverzeichnis des Servers).

Tabelle 5-7

Nr.	Aktion	Anmerkung
1.	Öffnen Sie das Verzeichnis im Windows Explorer	Auf dem Server PC "OPC Server Station"
2.	Kopieren Sie die Datei auf den Server PC "OPC Server Station" und ändern sie die Dateiendung auf "**DER"**	Hier im Beispiel: "UA Test Client.der"
3.	Im Zertifikatsverzeichnis des Servers finden Sie alle öffentlichen OPC UA Zertifikate denen der Server vertraut. An diese Stelle muss das Client-Zertifikat kopiert werden, um vom Server akzeptiert zu werden.	Das Verzeichnis befindet sich im Datenverzeichnis der SIMATIC NET Software. C:\Documents and settings\All Users\Application Data \Siemens\OPC\PKI\CA\ 

5.3.3 Authentication, SecurityPolicy und MessageSecurityMode

Der OPC UA Server unterstützt Authentifizierung von Clients beim Verbindungsaufbau. Es werden zwei Arten der Authentifizierung unterstützt:

- Anonymous
- UserName / Password

Nach der Installation sind beide Modi aktiv, um eine Inbetriebnahme zu erleichtern. Der Server kann umkonfiguriert werden, damit keine anonymen Anmeldungen mehr möglich sind. Benutzername und Passwort müssen vom Client angegeben werden und der Server prüft diese gegen die Windows Benutzerverwaltung. Somit können sich nur noch Clients verbinden, die einen Windows-Account auf der Servermaschine besitzen.

Der Server besitzt zwei Verbindungsendpunkt-Konfigurationen, die von Clients verwendet werden können. Jeder dieser Endpunkte repräsentiert einen anderen Verschlüsselungsmechanismus für die Datenübertragung.

- None
- Basic128RSA15

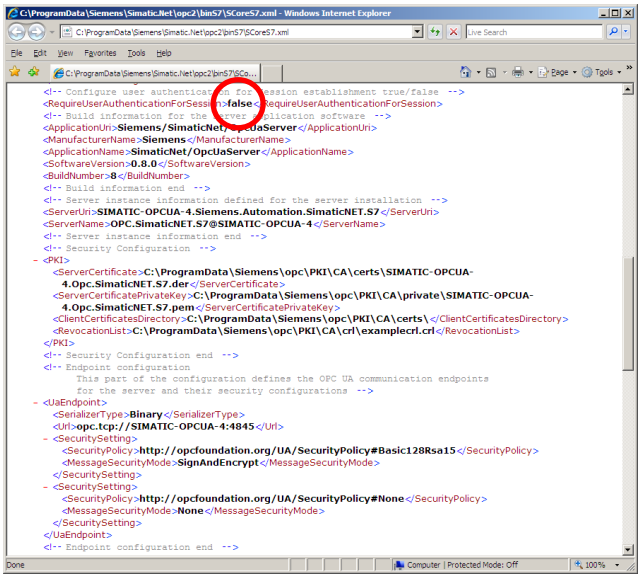
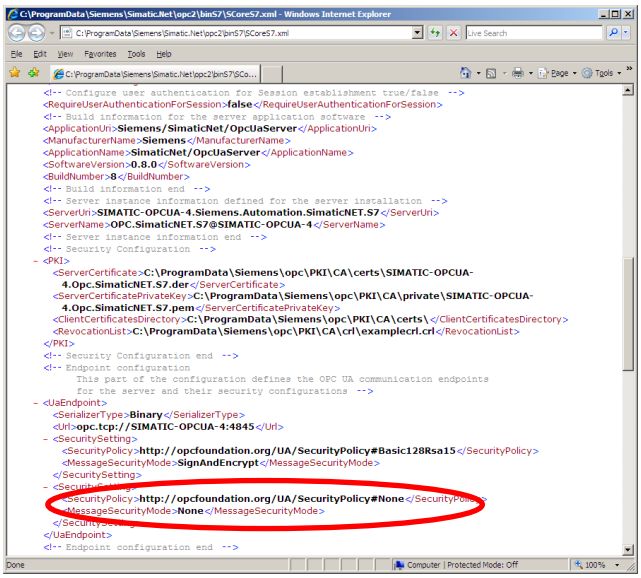
Copyright © Siemens AG 2017 All rights reserved

Nach der Installation sind beide SecurityPolicies aktiv, um eine Inbetriebnahme zu erleichtern. Der Server kann umkonfiguriert werden, um ausschließlich sichere verschlüsselte Verbindungen zu verwenden. Somit können sich nur noch Clients verbinden, die die Basic128RSA15 Verschlüsselung beherrschen.

Wenn Sie die Sicherheitsmechanismen des OPC UA Servers verändern wollen, editieren Sie die Konfigurationsdatei des OPC UA Servers und starten Sie den Server anschließend neu.

C:\ Documents and settings\All Users\Application Data \Siemens\Simatic.Net\opc2\bins7\ SCoreS7.xml

Tabelle 5-8

Nr.	Aktion	Anmerkung
1.	Öffnen der OPC UA Serverkonfigurationsdatei "SCoreS7.xml" in einem Editor (z.B. Notepad)	Die Datei befindet sich im Datenverzeichnis der SIMATIC NET Software. C:\Docoments and Settings\All Users\Application Data\Siemens\Simatic.Net\opc2\bins7
2.	Stellen Sie RequireUserAuthenticationForSession auf "true", wenn jeder OPC UA Client sich mit Benutzername und Passwort am Server authentifizieren soll, um eine Session aufbauen zu dürfen.	
3.	Löschen Sie den kompletten Eintrag SecuritySetting für SecurityPolicy "none" und MessageSecurityMode "none" aus der UAEndpoint Konfiguration. Der Server erlaubt danach ausschließlich verschlüsselte Übertragung vom Typ "Basic128RSA15" mit Signierung und Encryption. Nur Clients, die ebenfalls diesen Verschlüsselungstyp unterstützen, können sich verbinden (falls ihrem Zertifikat vertraut wird).	

Copyright © Siemens AG 2017 All rights reserved

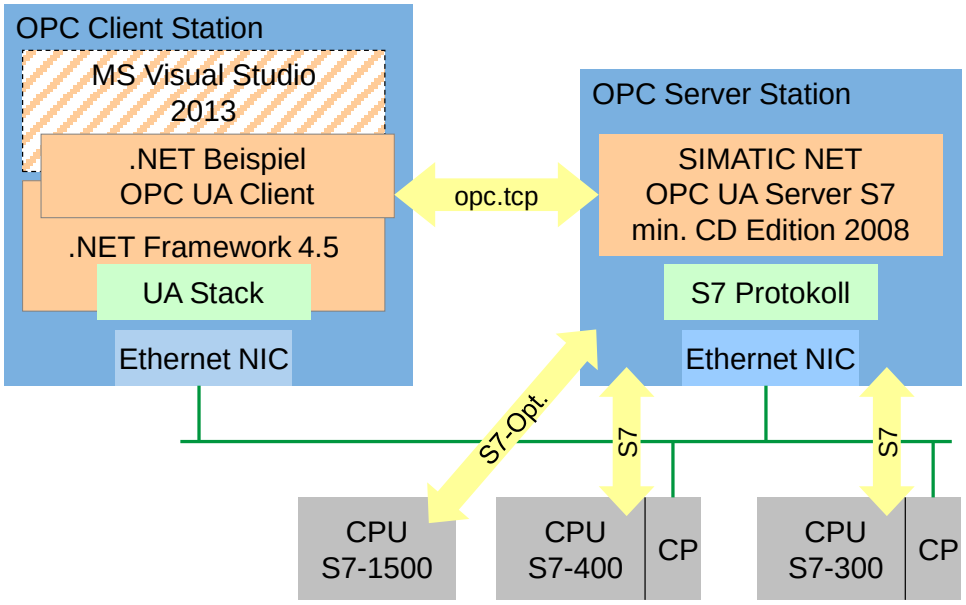
6 Installation

Dieses Kapitel beschreibt, welche Hard- und Softwarekomponenten installiert werden müssen. Die Beschreibungen, Handbücher und Lieferinformationen, die mit den entsprechenden Produkten ausgeliefert werden, sind in jedem Fall zu beachten.

Installation der Hardware

Nachfolgendes Bild zeigt den Hardwareaufbau der Anwendung sowie die erforderlichen Softwarekomponenten.

Abbildung 6-1



Hinweis Die Aufbaurichtlinien für Industrial Ethernet Netzwerke sind generell zu beachten.

Copyright © Siemens AG 2017 All rights reserved

Installation der Software

Dieses Kapitel beschreibt die Schritte zur Installation der Software, die für die Applikation benötigt wird.

Tabelle 6-1

Nr.	Aktion	Anmerkung
1	Installieren Sie SIMATIC NET DVD V14 auf die SIMATIC OPC Server Station.	
2	Installieren Sie STEP 7 V14 SP1 auf die Engineering Station (z.B. Laptop).	
3	Installieren Sie das .NET Framework 4.5 auf die OPC Client Station.	Dieser Schritt ist nur dann nötig wenn das Framework noch von keiner anderen Applikation installiert wurde.
4	Installieren Sie Microsoft Visual Studio 2010 auf die OPC UA Client Station. Installieren Sie die C# Entwicklungsumgebung.	Dieser Schritt ist nur dann nötig, wenn der Code verifiziert oder modifiziert werden sollte.

Installation der Beispiele

Dieses Kapitel beschreibt die Schritte zur Installation des Beispielcodes.

Tabelle 6-2

Nr.	Aktion	Anmerkung
1	Installieren Sie die Applikationssoftware auf die OPC UA Client Station. Kopieren Sie die Executables sowie die Assemblies (Ordner: "bin") und auch den Quellcode (Ordner: "scr") in ein Verzeichnis, auf das Sie Zugriffsrechte besitzen.	Den Quellcode benötigen Sie nur, wenn sie Modifikationen vornehmen wollen oder einzelne Funktionen verifizieren wollen.
2	Installieren Sie das STEP 7 Programm. Kopieren und entpacken Sie das STEP 7 Projekt.	Nehmen Sie ggf. Anpassungen bezüglich ihrer Hardware vor.

Copyright © Siemens AG 2017 All rights reserved

7 Inbetriebnahme der Applikation

Es wird vorausgesetzt, dass alle Hardware- und Softwarekomponenten erfolgreich installiert und verkabelt wurden.

Hinweis Die mit diesem Beispiel ausgelieferte Projektdatei (XDB) enthält die vollständig konfigurierte und projektierte PC-Station. Diese Datei kann nur dann ohne Anpassung verwendet werden, wenn die Hardware identisch mit der Projektierung ist.

Falls abweichende Hardware verwendet wird, muss die Konfiguration der PC Station angepasst werden. Dies ist speziell dann erforderlich, wenn die Hardwareausgabestände unterschiedlich sind oder die Ethernetadressen abweichen. Das mitgelieferte STEP 7 Projekt muss dann geöffnet und entsprechend angepasst werden (siehe hierzu Kapitel 6). Nach dem Speichern und Übersetzen werden alle Konfigurations- und Projektierungsinformationen in der XDB-Datei überschrieben.

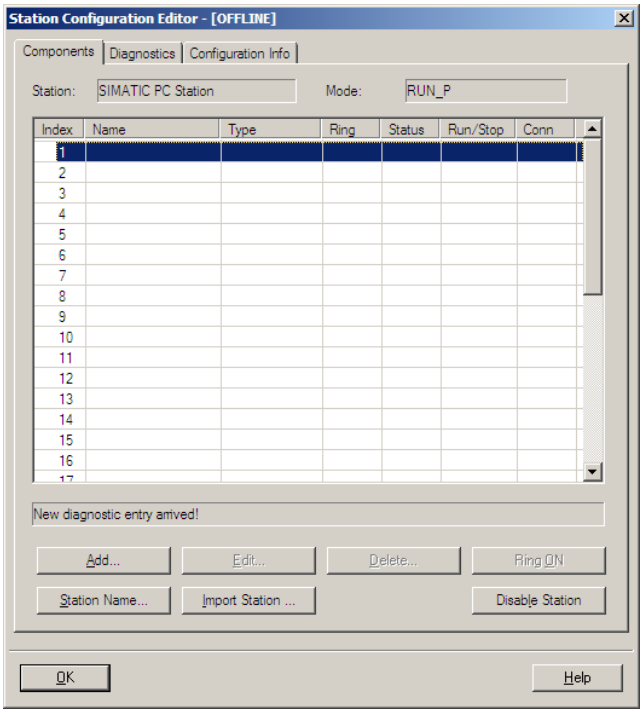
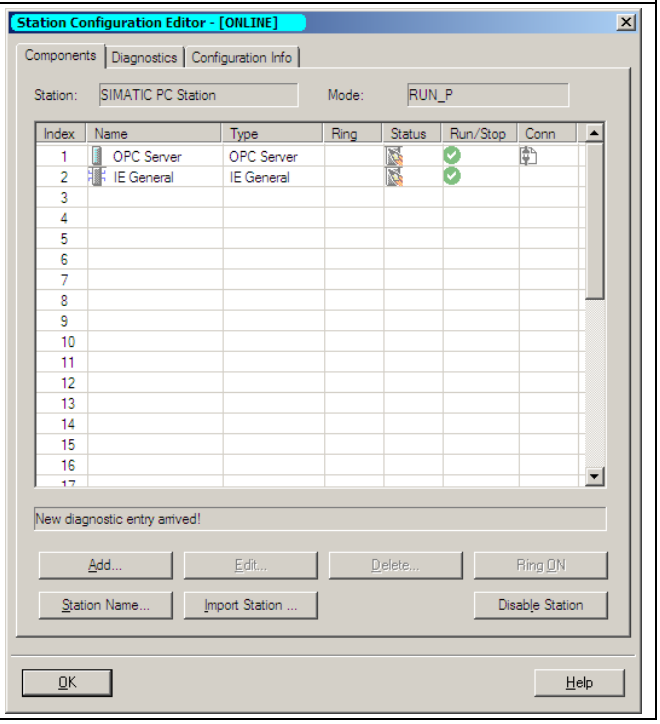
Inbetriebnahme der PC Station

Die folgende Tabelle zeigt die Konfiguration und Projektierung der PC Station mittels Import einer XDB Datei.

Tabelle 7-1

Nr.	Aktion	Anmerkung
1.	Projekt dearchivieren	Entpacken Sie die Datei **OPC_UA_STEP7_CODE_V12.zip**
2.	Projektdatei mit STEP 7 öffnen und die CPU-Stationen konfigurieren und die Konfiguration auf die Steuerungen laden	Nehmen Sie entsprechende Anpassungen vor, falls Sie unterschiedliche Hardware verwenden. Auch die PC-Station kann "online" konfiguriert und geladen werden. Alternative: laden Sie die PC-Station mit Hilfe der XDB-Datei: siehe nachfolgende Schritte.

Copyright © Siemens AG 2017 All rights reserved

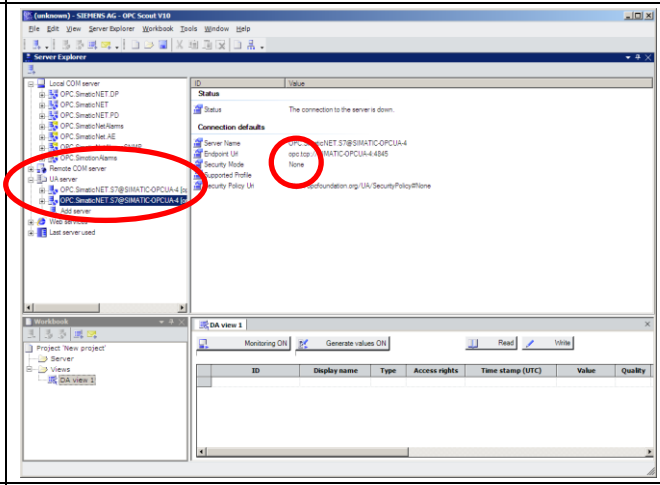
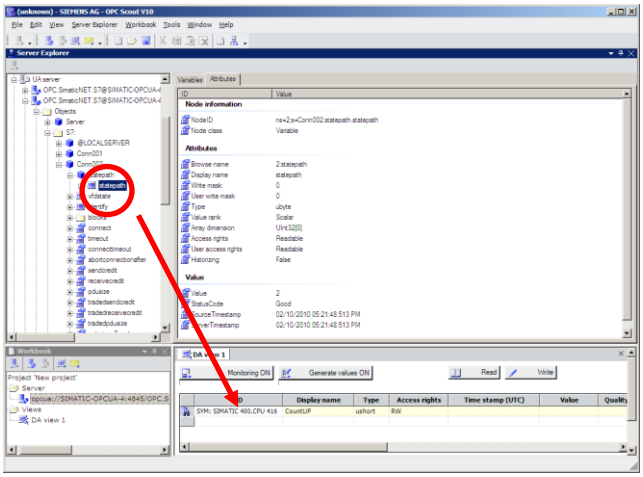
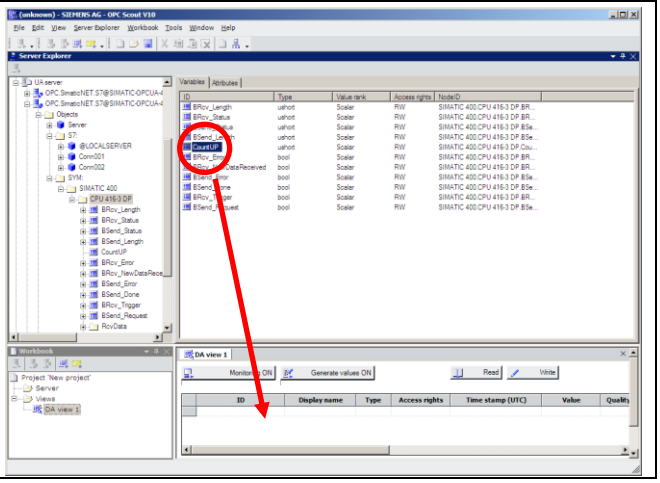
Nr.	Aktion	Anmerkung
3.	Öffnen des `Station Configurator` durch Doppelklick auf das Symbol in der Taskleiste.	
4.	`Import Station` anklicken	Bestätigen Sie die folgende Abfrage mit Ja.
5.	Auswahl der XDB-Datei	Die XDB-Datei befindet sich im Unterverzeichnis XDBs der entpackten ZIP-Datei im Verzeichnisbaum des STEP7 Projektes.
6.	Nach dem Import der XDB-Datei ist die PC-Station konfiguriert. Hierbei werden nicht nur die Verbindungsinformationen übernommen, sondern auch die Symboldatei aus der XDB extrahiert und dem Server verfügbar gemacht.	

Copyright © Siemens AG 2017 All rights reserved

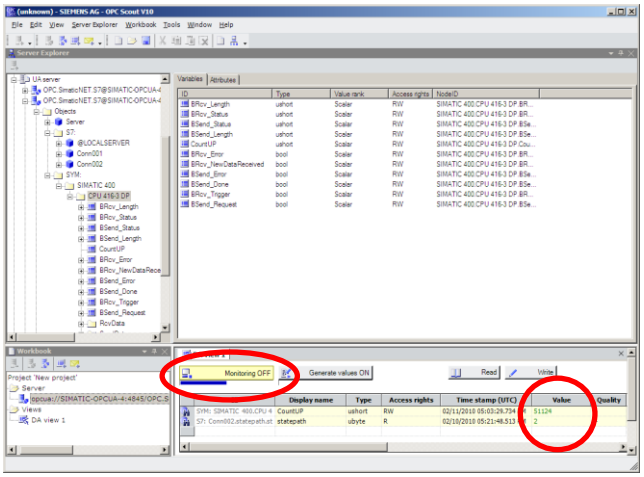
Testen mit dem OPC Scout V10

Der neue OPC Scout V10 ist ein kombinierter OPC Client, der sowohl die klassischen COM OPC Schnittstellen als auch das neue OPC UA beherrscht. Dieser Client wird mit der SIMATIC NET CD ausgeliefert und installiert.

Tabelle 7-2

Nr.	Aktion	Anmerkung
1.	Starten Sie den OPC Scout V10 aus dem Startmenü	Auf dem Server PC "OPC Server Station"
2.	Verbinden Sie sich mit dem "unsicheren" Endpoint "none". Die Verbindung wird aufgebaut sobald Sie den entsprechenden Eintrag in der Baumansicht aufblättern.	
3.	Browsen Sie in den Unterordner "S7:" und fügen Sie die Variable "&statepath" aus der Verbindung "Conn002" per Drag&Drop in das untere Fenster ein.	
4.	Browsen Sie in den Unterordner "SYM:" und fügen Sie die Variable "CountUP" per Drag&Drop in das untere Fenster ein.	

Copyright © Siemens AG 2017 All rights reserved

Nr.	Aktion	Anmerkung
5.	Schalten Sie die Überwachung ein und prüfen Sie, ob in der Spalte Value entsprechende Werte angezeigt werden. Die Verbindungsstatusvariable muss den Wert "2" (aufgebaut) haben und der Zähler sollte inkrementieren.	

Starten der Beispielapplikation

Tabelle 7-3

Nr.	Aktion	Anmerkung
1.	Starten Sie die entsprechende Applikation "Client" oder "SimpleClient".	Die Dateien befinden sich im Unterordner "bin" des entpackten Beispiels.
2.	Bedienung der Beispielapplikation	Siehe Kapitel 8.

Copyright © Siemens AG 2017 All rights reserved

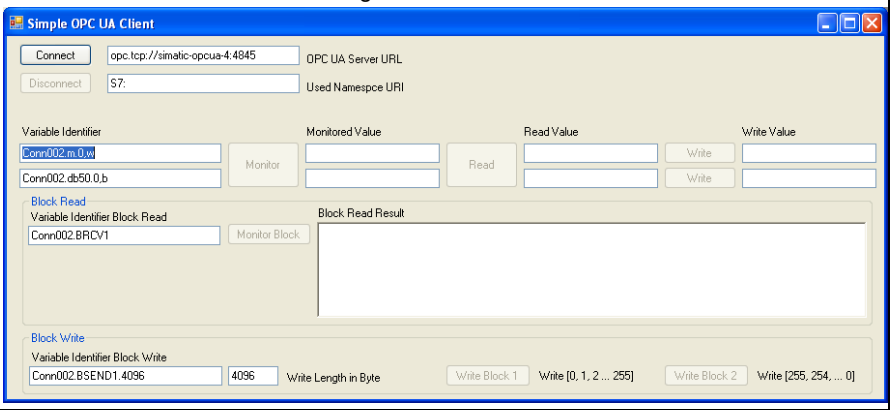
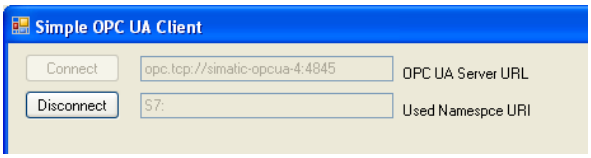
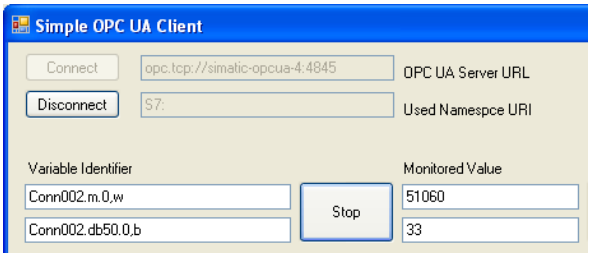
8 Bedienung der Applikation

8.1 Bedienung Simple Client

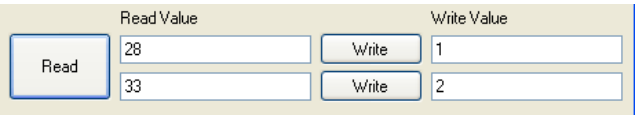
Zugriff mit direkter Adressierung

Im ersten Teil des einfachen Beispiels werden Daten von S7 Variablen über direkte Adressierung beobachtet, gelesen und geschrieben.

Tabelle 8-1

Nr.	Aktion	Anmerkung
1	Starten des Simple Clients.	Nach dem Starten der Anwendung erscheint diese Benutzeroberfläche. 
2	Verbindung zum Server aufbauen	Die Server URL muss manuell eingegeben werden. Alle Identifier für NodeIds in der Oberfläche verwenden den gleichen Namespace. Als Vorgabe ist der Namespace **S7:** für die direkte Adressierung angegeben. Über den **Connect** Button kann die Verbindung zum Server aufgebaut werden.  **Hinweis** Um auf optimierte Datenbausteine der S7-1200/S7-1500 CPUs zugreifen zu können, müssen Sie mindestens den OPC-Server V12 projektiert haben und mit dem Client über den SimetcNET.S7OPT (Port 55105) auf den OPC Server zugreifen.
3	Monitoring von Daten	Für das Beobachten von Datenänderungen, Lesen und Schreiben können zwei Variablen angegeben werden. Die NodeId wird aus dem Identifier und dem Namespace gebildet. Über den Button **Monitor** kann die Beobachtung eingeschaltet werden. Die gemeldeten Datenänderungen werden in den Textfeldern neben dem Button angezeigt. 
4	Lesen und Schreiben	Für das Lesen und Schreiben werden die gleichen NodeIds verwendet wie für das Beobachten. Durch Drücken des Buttons **Read** werden beide Variablen gelesen. Über die beiden **Write** Buttons können die Variablen einzeln

Copyright © Siemens AG 2017 All rights reserved

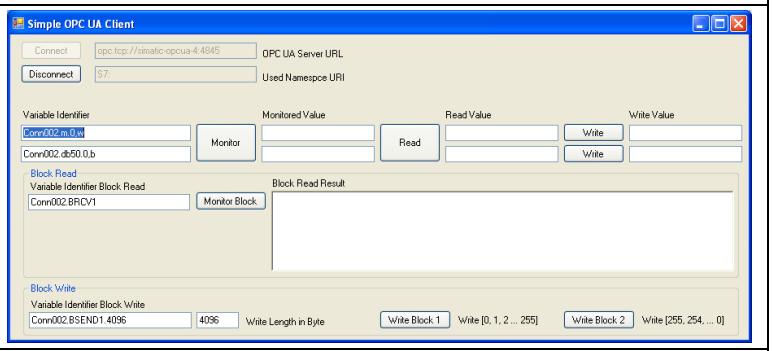
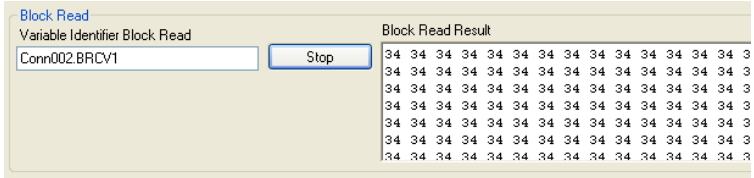
Nr.	Aktion	Anmerkung
		geschrieben werden. Vor dem Schreiben muss erst gelesen werden, da die Datenkonvertierung für das Schreiben auf Basis der gelesenen Werte geschieht. 

Verwendung der Blockdienste

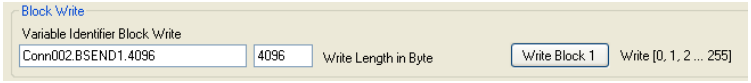
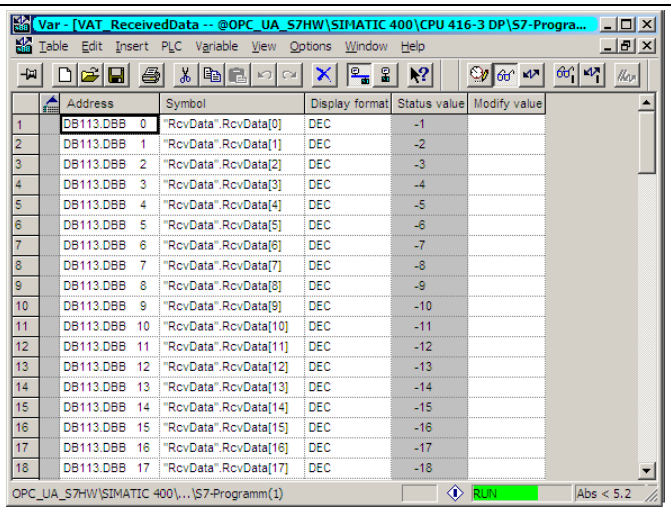
Als anschauliches, einfaches Beispiel wird hier ein typischer Anwendungsfall für blockorientierte Dienste gezeigt: Das Senden von Rezeptdaten an die S7 bzw. das Empfangen von Ergebnisdaten von der S7 unter Verwendung der Blockdienste BSEND und BRCV.

Die S7 Steuerung empfängt ein Rezept von der PC Station "OPC Client Station". Hierzu wurden im OPC UA Client zwei "Rezepte" simuliert, die nach Betätigen des jeweiligen Buttons durch einen Schreibauftrag auf die NodeID der BSEND Variablen an den OPC UA Server gesendet werden. Der OPC UA Server setzt einen BSEND Aufruf an die entsprechende Steuerung ab. Die S7 wiederum muss einen passenden BRCV aufrufen, um die Daten zu erhalten. Der BRCV muss mit korrekter RID auf der richtigen Verbindung mit der nötigen Mindestlänge zum richtigen Zeitpunkt verfügbar sein. Gehen Sie entsprechend der nachfolgenden Schrittanweisung vor.

Tabelle 8-2

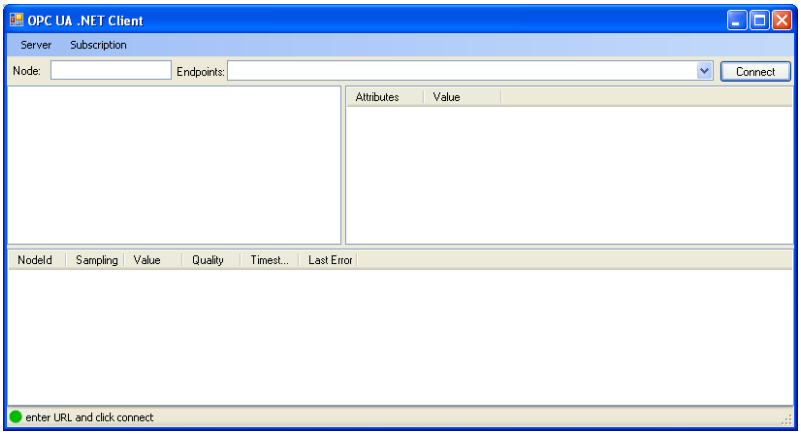
Nr.	Aktion	Anmerkung
1.	Starten Sie STEP 7 und öffnen Sie die Variablentabelle (VAT) des Empfangspuffers.	Gehen Sie zur SIMATIC Steuerung -> CPU ->S7-Program -> Bausteine und öffnen Sie "VAT_ReceivedData". Schalten Sie in die Online Ansicht (Brille)
2.	Starten Sie den OPC UA Client und verbinden Sie sich mit dem OPC UA Server	
3.	Empfangen von der S7	Zum Empfangen von der S7 muss eine Subscription mit Monitored Item angelegt werden. Dazu muss in der Gruppe "Block Read" der Button "Monitoring Block" angeklickt werden. Im Feld neben dem Button werden die Ergebnisdaten als HEX-Code angezeigt: Beachten Sie die R_ID als Bestandteil der NodeId. Mit dieser ID muss die S7 den BSEND Baustein versorgen 
4.	Senden der Daten an die S7	Wählen Sie das "Rezept", in dem Sie entweder "Write Block 1" oder "Write Block 2" drücken. Der erste Button schreibt in den Datenblock,

Copyright © Siemens AG 2017 All rights reserved

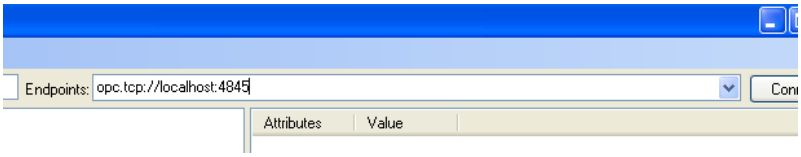
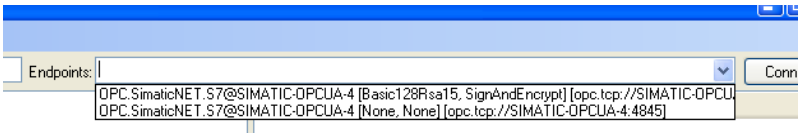
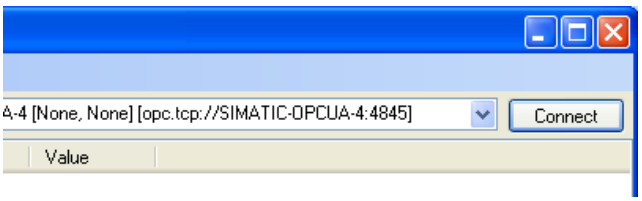
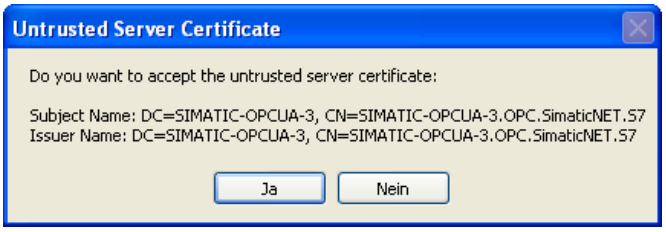
Nr.	Aktion	Anmerkung
		indem ein Wert für jeden Arrayeintrag ab 0 inkrementiert wird. Beim zweiten werden die Arrayeinträge ab 255 dekrementiert. Beachten Sie die R_ID als Bestandteil der NodeId. Mit dieser ID muss in der S7 der BRCV Baustein versorgen werden. 
5.	Prüfen Sie mit der Variablentabelle (VAT), ob die Daten angekommen sind.	
6.	Verarbeitung in S7	An dieser Stelle sollte das Rezept in der S7 verarbeitet werden und anschließend der BRCV wieder empfangsbereit geschaltet werden.
7.	Empfangen wieder aktivieren	Nachdem die Daten verarbeitet wurden, schalten Sie den BRCV wieder "scharf" (dies geschieht im Beispiel automatisch).
8.	Erneut senden	Senden Sie nun das zweite Rezept mit dem OPC UA Client.

8.2 Bedienung komfortabler OPC UA Client

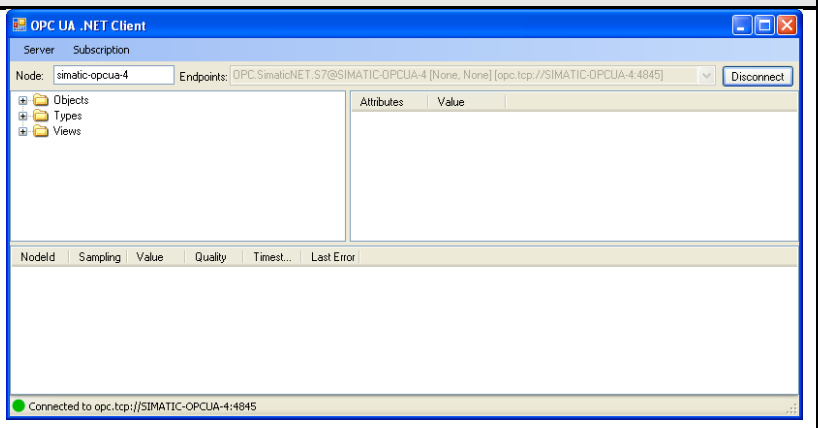
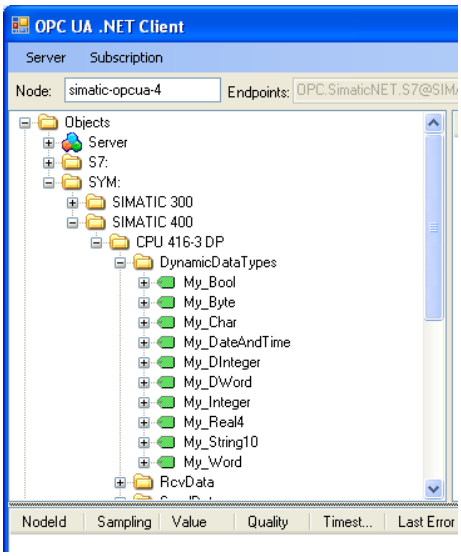
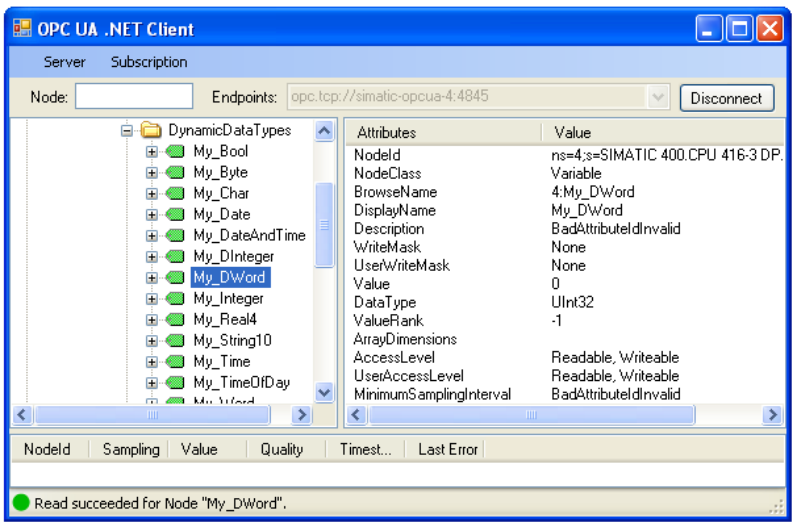
Tabelle 8-3

Nr.	Aktion	Anmerkung
1	Starten des OPC UA Clients (dieses Beispiel)	Nach dem Starten der Anwendung erscheint diese Benutzeroberfläche. 
2	Manuelle Eingabe einer Server URL	Um eine Verbindung zu einem OPC UA Server aufzubauen, kann die URL des Servers in das Textfeld **Endpoints** manuell eingegeben werden. Ist der SIMATIC NET OPC UA Server auf dem gleichen Rechner installiert, kann die URL **opc.tcp://localhost:55101** verwendet werden. Bei der manuellen Eingabe wird versucht, die Verbindung ohne Sicherheit

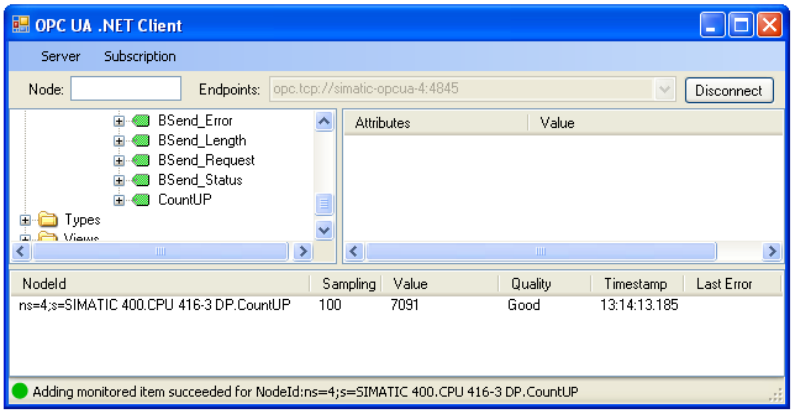
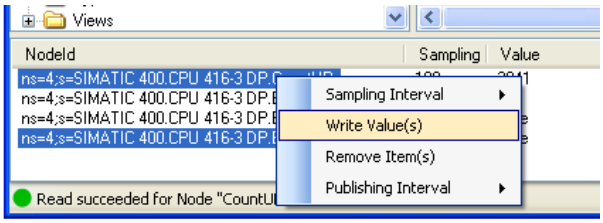
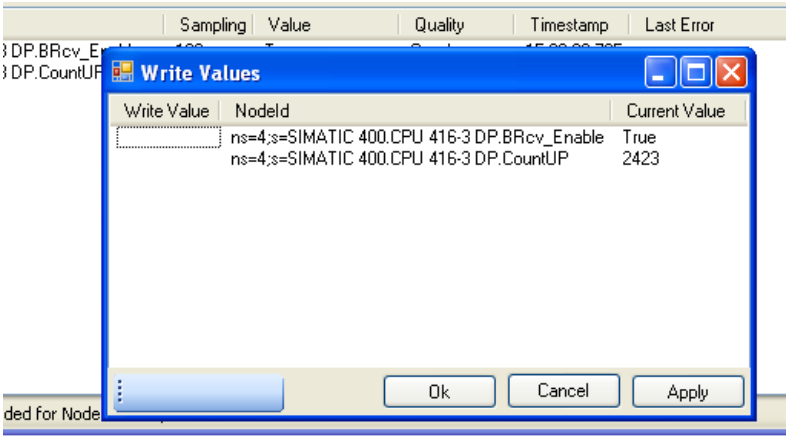
Copyright © Siemens AG 2017 All rights reserved

Nr.	Aktion	Anmerkung
		aufzubauen. **Hinweis** Um auf optimierte Datenbausteine der S7-1200/S7-1500 CPUs zugreifen zu können, müssen Sie mindestens den OPC-Server V12 projektiert haben und mit dem Client über den SimetcNET.S7OPT (Port 55101) auf den OPC Server zugreifen. 
3	Auflisten von verfügbaren Endpunkten	Über die Auswahlliste **Endpoints** kann die Liste der verfügbaren Endpoints aufgelistet werden. Ist das Textfeld **Node** leer, werden die lokalen Server bzw. deren Endpoints aufgelistet. Die Eingabe eines Remote-Rechnernamens in das Textfeld **Node** und das anschließende Anklicken der Auswahlliste **Endpoints** zeigt die verfügbaren Endpunkte auf dem Remote-Rechner an. Neben dem Server-Namen werden die Sicherheitseinstellungen und die URL des Endpoints angezeigt. 
4	Verbindungsaufbau zum Server	Über die Schaltfläche **Connect** wird die Verbindung zum Server aufgebaut.  Ist das Zertifikat des Servers nicht als vertrauenswürdig eingestuft, wird der Benutzer gefragt, ob er das Zertifikat dennoch akzeptieren will. Die notwendigen Schritte, um das Client Zertifikat beim Server zu akzeptieren, sind im Kapitel 5.3.2 beschrieben.  War der Aufbau der Verbindung erfolgreich, wird die oberste Ebene des Adressraums des OPC UA Servers im Fenster links oben angezeigt:

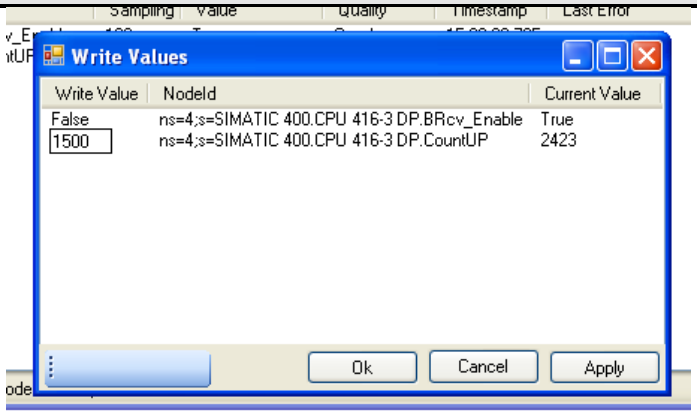
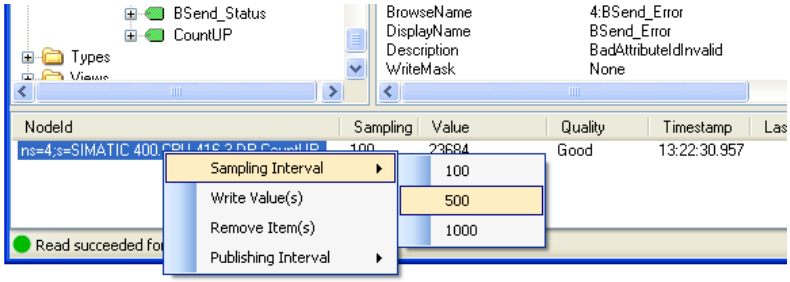
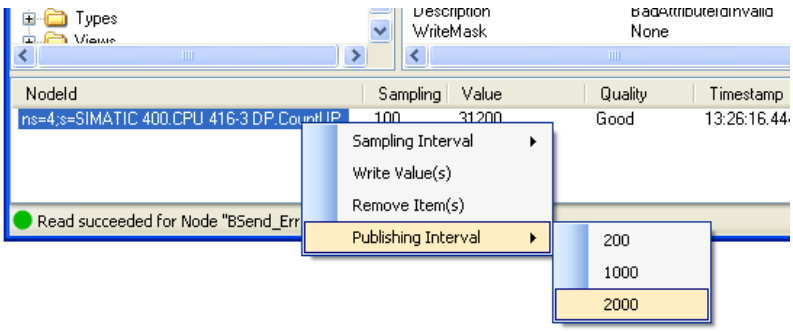
Copyright © Siemens AG 2017 All rights reserved

Nr.	Aktion	Anmerkung
		 Auf der Schaltfläche erscheint jetzt **Disconnect**.
5	Browse	Die Baumansicht ermöglicht das Navigieren im Adressraum des Servers. 
6	Attribute eines Knotens lesen	Durch Selektieren eines Knotens in der Baumansicht werden dessen Attribute und deren Werte in der Liste rechts daneben angezeigt. 

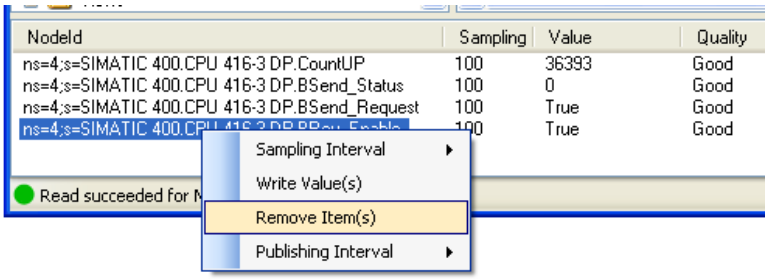
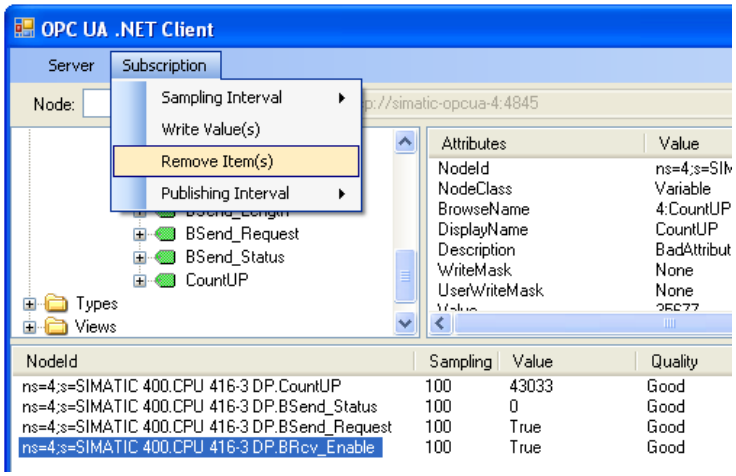
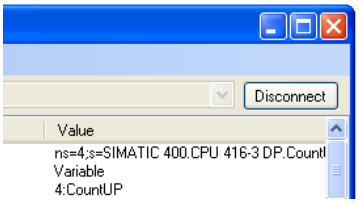
Copyright © Siemens AG 2017 All rights reserved

Nr.	Aktion	Anmerkung
7	Beobachten von Variablen (Monitored Items)	Im Adressraum des Servers vorhandene Variablen können von der Baumansicht aus per Drag&Drop in das untere Beobachtungsfenster gezogen werden. Dort werden dann Wertänderungen angezeigt. 
8	Schreiben von Werten	Sind in dem Beobachtungsfenster eine oder mehrere Variablen selektiert, kann der Benutzer über das Kontextmenü der Liste einen Schreibvorgang starten:  In dem jetzt geöffneten Dialogfenster erscheinen die zuvor in der Liste ausgewählten Variablen mit deren aktuellen Werten:  In der Spalte Write Value können Werte eingetragen werden:

Copyright © Siemens AG 2017 All rights reserved

Nr.	Aktion	Anmerkung
		 Das Betätigen der Schaltfläche **Ok** schreibt die neuen Werte und der Dialog wird geschlossen. Wird dagegen die Schaltfläche **Apply** gedrückt, werden die Werte geschrieben und der Dialog bleibt für das weitere Schreiben von Werten geöffnet.
9	Sampling Intervall ändern	Über das Kontextmenü können drei verschiedene Sampling Intervalle eingestellt werden. Dabei ist die Auswahl mehrerer Einträge des Listenfelds möglich. 
10	Publish Intervall ändern	Das Publish Intervall der Subscription kann ebenfalls über das Kontextmenü des Beobachtungsfensters geändert werden. 

Copyright © Siemens AG 2017 All rights reserved

Nr.	Aktion	Anmerkung
11	Monitored Items entfernen	Die Auswahl des Menüeintrags **Remove Item(s)** entfernt alle in der Liste selektierten Variablen. 
12	Aufruf von Aktionen aus der Menüleiste	Die Aktionen 8 bis 11 können auch über die Menüleiste ausgeführt werden: 
13	Verbindung zum Server abbauen	Über den **Disconnect** Button wird die Verbindung zum OPC UA Server abgebaut. 
14	Verbindungsfehler OPC UA Server zur Steuerung	Ist die Verbindung zwischen OPC UA Server und S7 unterbrochen, dann wechselt der Status der Variablen in der Beobachtungsliste auf Bad.
15	Fehler bei Aufrufen	Treten Fehler bei OPC UA Aufrufen auf, werden diese Fehler in der Statusleiste der Applikation ausgegeben. Reagiert der Server nicht, liefert die Client API Timeout Fehler.

Copyright © Siemens AG 2017 All rights reserved

9 Weitere Hinweise, Tipps und Tricks, etc.

Wiederverwendung und Erweiterung der Client API

Die Client API ist als eigenständige, wiederverwendbare Assembly DLL realisiert. Sie kann direkt in anderen Applikationen angewendet werden. Eine Erweiterung auf zusätzliche OPC UA Features wie Methodenaufrufe ist einfach möglich.

Wiederverwendung der GUI Controls

Die GUI Elemente für Browsen, Auflistung der Attribute und für die Beobachtung von Variablenwerten sind als Controls erstellt. Für eine Wiederverwendung macht es Sinn, diese Controls in eine eigene Assembly DLL auszulagern.

Speichern von NodeIds

NodeIds setzen sich aus einem Identifier und dem Namespace Index zusammen. Der Namespace Index ändert sich zwar nicht, solange der OPC UA Server läuft, aber prinzipiell kann sich der Index bei einem Neustart des OPC UA Servers ändern. Das ist zwar beim SIMATIC NET OPC UA Server nicht der Fall, aber ein OPC UA Client sollte darauf vorbereitet sein, wenn er die NodeIds speichert.

Beim Speichern darf nicht einfach der Index gespeichert werden, sondern es muss die Namespace URI gespeichert werden. Diese URI bleibt konstant, auch wenn sich der Index ändert.

Es gibt zwei Strategien, die Namespace URI zu speichern:

- Die URI wird statt des Index mit dem Identifier für die NodeId gespeichert. Das ist die einfachste Variante, hat aber den Nachteil, dass sehr viel redundante Information gespeichert wird, wenn die Namespace URI für alle gespeicherten NodeIds des Server gleich ist.
- Der Index wird mit dem Identifier gespeichert, aber parallel dazu wird die passende Namespace Tabelle mit den Namespace URIs abgelegt. Diese Variante ist effizienter, erfordert aber einen zusätzlichen Speicherort für die Tabelle.

Für beide Varianten muss nach dem Verbindungsaufbau mit dem Server die Namespace Tabelle vom Server gelesen werden und die Namespace URI wieder in den aktuellen Index umgesetzt werden.

Optimierung der NodeIds für Read und Write

NodeIds können lange Texte enthalten und sind deshalb nicht dafür geeignet, in zyklischen Aufrufen von Read und Write verwendet zu werden, da ein unnötiger Overhead auf dem Netzwerk und bei der Verarbeitung im Server erzeugt wird.

OPC UA stellt die speziellen Services RegisterNodes und UnregisterNodes zur Verfügung, um eine Optimierung zu erreichen. RegisterNodes liefert für eine Liste von originalen NodeIds eine Liste von optimierten NodeIds mit numerischen Identifiern, die wie Handles verwendet werden können. Diese NodeIds sind auf dem Netzwerk nur vier Byte lang und können im Server für einen sehr schnellen Zugriff auf die Daten verwendet werden.

Da das Handle auch eine NodeId ist, kann es in allen Services anstatt der originalen NodeId verwendet werden. Die optimierte NodeId ist aber nur innerhalb der Session gültig.

Werden registrierte NodeIds nicht mehr benötigt, sollten sie mit UnregisterNode freigegeben werden, um die Ressourcen im Server freizugeben.

Copyright © Siemens AG 2017 All rights reserved

10 Literaturhinweis

Literaturangaben

Diese Listen sind keinesfalls vollständig und spiegeln nur eine Auswahl an geeigneter Literatur wieder.

Tabelle 10-1

	Themengebiet	Titel
/1/	STEP 7	Automatisieren mit STEP7 in AWL und SCL Hans Berger Publicis MCD Verlag ISBN 3-89578-113-4
/2/	OPC UA	OPC Unified Archticture Mahnke, Leitner, Damm Springer Verlag ISBN 978-3-540-68898-3

Internet-Link-Angaben

Tabelle 10-2

	Themengebiet	Titel
\1\	Referenz auf den Beitrag	https://support.industry.siemens.com/cs/ww/de/view/42014088http://support.automation.siemens.com/WW/view/de/42014088
\2\	Siemens Industry Online Support	http://support.automation.siemens.com
\3\	OPC Data Access Custom Interface Version 3.0	Spezifikation auf der OPC Foundation Webseite zum Download für OPC Mitglieder www.opcfoundation.org
\4\	OPC Unified Architecture	Spezifikation auf der OPC Foundation Webseite zum Download für OPC Mitglieder www.opcfoundation.org
\5\	SIMATIC NET PC-Software Industrielle Kommunikation mit PG/PC Band 1 - Grundlagen Systemhandbuch	Beschreibung bzw. Informationen zu: • Allgemeine Informationen über die PC–Werkzeuge. • Funktionen von NCM PC . https://support.industry.siemens.com/cs/ww/de/view/77376110
\6\	SIMATIC NET PC-Software Industrielle Kommunikation mit PG/PC Band 2 - Schnittstellen Programmierhandbuch	Handbuch zur Industriellen Kommunikation auf PG/PC mit SIMATIC NET. https://support.industry.siemens.com/cs/ww/de/view/77378184
\7\	Unified Automation	http://www.unified-automation.com/downloads/opc-ua-development.html

Copyright © Siemens AG 2017 All rights reserved

11 Historie

Tabelle 11-1

Version	Datum	Änderung
V1.2	12/2017	Überarbeite Version: Migration auf STEP 7 V14, SIMATIC NET V14, OPC UA 1.03
V1.1	06/2014	Überarbeite Version: Migration auf STEP 7 V13
V1.0	04/2010	Erste Ausgabe

Copyright © Siemens AG 2017 All rights reserved