



## SIMATIC NET

### PC-Software Industrielle Kommunikation mit PG/PC Band 1 - Grundlagen

Systemhandbuch

Vorwort

1

SIMATIC NET in der  
industriellen Kommunikation

2

Grundlagen der OPC-  
Schnittstelle

3

Literaturverzeichnis

4

## Rechtliche Hinweise

### Warnhinweiskonzept

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise zu Ihrer persönlichen Sicherheit sind durch ein Warndreieck hervorgehoben, Hinweise zu alleinigen Sachschäden stehen ohne Warndreieck. Je nach Gefährdungsstufe werden die Warnhinweise in abnehmender Reihenfolge wie folgt dargestellt.

#### GEFAHR

bedeutet, dass Tod oder schwere Körperverletzung eintreten **wird**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

#### WARNUNG

bedeutet, dass Tod oder schwere Körperverletzung eintreten **kann**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

#### VORSICHT

bedeutet, dass eine leichte Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

#### ACHTUNG

bedeutet, dass Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Beim Auftreten mehrerer Gefährdungsstufen wird immer der Warnhinweis zur jeweils höchsten Stufe verwendet. Wenn in einem Warnhinweis mit dem Warndreieck vor Personenschäden gewarnt wird, dann kann im selben Warnhinweis zusätzlich eine Warnung vor Sachschäden angefügt sein.

### Qualifiziertes Personal

Das zu dieser Dokumentation zugehörige Produkt/System darf nur von für die jeweilige Aufgabenstellung **qualifiziertem Personal** gehandhabt werden unter Beachtung der für die jeweilige Aufgabenstellung zugehörigen Dokumentation, insbesondere der darin enthaltenen Sicherheits- und Warnhinweise. Qualifiziertes Personal ist auf Grund seiner Ausbildung und Erfahrung befähigt, im Umgang mit diesen Produkten/Systemen Risiken zu erkennen und mögliche Gefährdungen zu vermeiden.

### Bestimmungsgemäßer Gebrauch von Siemens-Produkten

Beachten Sie Folgendes:

#### WARNUNG

Siemens-Produkte dürfen nur für die im Katalog und in der zugehörigen technischen Dokumentation vorgesehenen Einsatzfälle verwendet werden. Falls Fremdprodukte und -komponenten zum Einsatz kommen, müssen diese von Siemens empfohlen bzw. zugelassen sein. Der einwandfreie und sichere Betrieb der Produkte setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung, Montage, Installation, Inbetriebnahme, Bedienung und Instandhaltung voraus. Die zulässigen Umgebungsbedingungen müssen eingehalten werden. Hinweise in den zugehörigen Dokumentationen müssen beachtet werden.

### Marken

Alle mit dem Schutzrechtsvermerk ® gekennzeichneten Bezeichnungen sind eingetragene Marken der Siemens AG. Die übrigen Bezeichnungen in dieser Schrift können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen kann.

### Haftungsausschluss

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden regelmäßig überprüft, notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten.

# Inhaltsverzeichnis

<b>1</b>	<b>Vorwort .....</b>	<b>7</b>
1.1	Produkt - Historie .....	7
1.2	Willkommen bei SIMATIC NET .....	8
<b>2</b>	<b>SIMATIC NET in der industriellen Kommunikation.....</b>	<b>11</b>
2.1	SIMATIC NET und die Protokolle - Ein Überblick.....	13
2.2	Industrielle Kommunikation mit PROFIBUS - Ein Überblick.....	15
2.2.1	PROFIBUS, was ist das? .....	15
2.2.2	PROFIBUS, wie ist die Funktionsweise? .....	16
2.2.3	PROFIBUS, wie sieht er im ISO/OSI-Referenzmodell aus? .....	17
2.3	Industrielle Kommunikation mit Ethernet - Ein Überblick.....	18
2.3.1	Industrial Ethernet, was ist das? .....	18
2.3.2	Switched-Ethernet, was ist das? .....	20
2.3.3	Industrial Ethernet, welche Schichten werden im ISO/OSI-Referenzmodell realisiert? .....	20
2.4	PROFINET - Ein Überblick .....	21
2.4.1	PROFINET, was ist das? .....	21
2.4.2	PROFINET, auf welcher Kommunikation basiert es? .....	23
2.4.3	PROFINET, welche Schichten werden im ISO/OSI Referenzmodell realisiert? .....	25
2.5	SEND/RECEIVE-Protokoll .....	26
2.5.1	SEND/RECEIVE-Protokoll, was ist das? .....	26
2.5.2	SEND/RECEIVE-Protokoll, wie sieht eine typische Anlagenkonfiguration aus? .....	26
2.5.3	SEND/RECEIVE-Protokoll wie funktioniert es? .....	27
2.5.4	SEND/RECEIVE-Protokoll, welche Kommunikationsdienste stehen zur Verfügung? .....	29
2.5.5	SEND/RECEIVE-Protokoll, wie wird es projektiert? .....	30
2.5.6	SEND/RECEIVE-Protokoll, welches sind die Vor- und Nachteile? .....	30
2.6	Das DP-Protokoll.....	32
2.6.1	DP-Protokoll, was ist das? .....	32
2.6.2	DP-Protokoll, wie sieht eine typische Anlagenkonfiguration aus? .....	34
2.6.3	DP-Protokoll, wie funktioniert es? .....	35
2.6.4	DP-Protokoll, wie wird es projektiert? .....	36
2.6.5	DP-Protokoll, welches sind die Vorteile? .....	36
2.6.6	DP-Master Klasse 1, welche Kommunikationsdienste stehen zur Verfügung? .....	37
2.6.7	DP-Master Klasse 2, welche Kommunikationsdienste stehen zur Verfügung? .....	39
2.6.8	DPC1, welche Kommunikationsdienste stehen zur Verfügung? .....	40
2.6.9	DPC2, welche Kommunikationsdienste stehen zur Verfügung? .....	42
2.6.10	DP Slave, welche Kommunikationsdienste stehen zur Verfügung? .....	43
2.7	Das S7-Protokoll .....	44
2.7.1	S7-Protokoll, was ist das? .....	44
2.7.2	S7-Protokoll, wie sieht eine typische Anlagenkonfiguration aus? .....	45
2.7.3	S7-Protokoll, wie funktioniert es? .....	46
2.7.4	S7-Protokoll, welche Kommunikationsdienste stehen zur Verfügung? .....	47
2.7.5	Hochverfügbare S7-Verbindungen, was ist das? .....	52

2.7.5.1	Hochverfügbare S7-Verbindungen, ein Überblick .....	53
2.7.5.2	Projektierung .....	61
2.7.5.3	Diagnose, Inbetriebnahme, Wartung, Betrieb.....	63
2.7.6	S7-Protokoll, wie wird es projektiert? .....	65
2.7.7	S7-Protokoll, welches sind die Vor- und Nachteile? .....	66
2.8	Das SNMP-Protokoll .....	67
2.8.1	SNMP-Protokoll, was ist das?.....	67
2.8.2	SNMP-Protokoll, wie sieht eine typische Anlagenkonfiguration aus?.....	67
2.8.3	SNMP-Protokoll, wie funktioniert es?.....	68
2.8.4	SNMP-Protokoll, welche Kommunikationsdienste stehen zur Verfügung? .....	69
2.8.5	SNMP-Protokoll, wie wird es projektiert? .....	69
2.8.6	SNMP-Protokoll, welches sind die Vor- und Nachteile? .....	70
2.9	Die Kommunikation mit PROFINET IO .....	70
2.9.1	PROFINET IO, was ist das? .....	70
2.9.2	PROFINET IO, wie sieht eine typische Anlagenkonfiguration aus? .....	71
2.9.3	PROFINET IO, wie funktioniert es? .....	73
2.9.4	PROFINET IO mit Isochroner Real Time-Kommunikation (IRT) .....	75
2.9.5	PROFINET IO, welche Kommunikationsdienste stehen zur Verfügung? .....	78
2.9.6	PROFINET IO, wie wird es projektiert? .....	79
2.9.7	PROFINET IO, welches sind die Vorteile? .....	80
2.10	Security bei SIMATIC NET.....	80
<b>3</b>	<b>Grundlagen der OPC-Schnittstelle .....</b>	<b>81</b>
3.1	Einführung in OPC .....	81
3.1.1	OPC, was ist das?.....	81
3.1.2	Vorteile von OPC .....	82
3.1.3	OPC-Schnittstelle, was leistet sie? .....	83
3.1.4	OPC-Server, was ist das? .....	84
3.1.5	OPC-Client, was ist das? .....	86
3.1.6	Server und Client, wie arbeiten sie zusammen?.....	86
3.1.7	Grundbegriffe .....	88
3.1.7.1	COM-Objekte, was ist das? .....	88
3.1.7.2	COM-Objekte, wie werden Sie dargestellt? .....	89
3.1.7.3	COM-Schnittstellen, was leisten sie?.....	90
3.1.7.4	COM-Schnittstellenarten, welche gibt es, wie wird darauf zugegriffen? .....	90
3.2	Data Access .....	92
3.2.1	Einführung in die Data-Access-Schnittstelle .....	92
3.2.1.1	Was kann OPC Data Access? .....	92
3.2.1.2	OPC Data Access, was ist das? .....	93
3.2.1.3	Klassenmodell von OPC Data Access, was leistet es? .....	94
3.2.1.4	Klasse OPC-Server, was leistet sie? .....	95
3.2.1.5	Klasse OPC-Group, was leistet sie? .....	95
3.2.1.6	Klasse OPC-Item, was leistet sie? .....	96
3.2.1.7	OPC Data Access, welche Schnittstellenspezifikationen gibt es? .....	97
3.3	OPC Alarms & Events .....	97
3.3.1	Einführung in OPC Alarms & Events .....	97
3.3.1.1	OPC Alarms & Events, was ist das? .....	97
3.3.1.2	Ereignisse und Ereignismeldungen, was ist das? .....	98
3.3.1.3	Klassenmodell von OPC Alarms & Events, was leistet es? .....	98
3.3.1.4	Klasse OPC-Event-Server, was leistet sie? .....	99

3.3.1.5	Klasse OPC-Event-Subscription, was leistet sie? .....	100
3.3.1.6	Klasse OPC-Event-Area-Browser, was leistet sie? .....	101
3.3.1.7	Meldungsempfang, wie funktioniert er? .....	101
3.3.1.8	Meldungen bei SIMATIC S7, wie sind sie definiert? .....	102
3.3.1.9	Meldungen, wie sieht die Praxis aus (Beispiel)? .....	103
3.3.2	Alarms & Events Schnittstelle .....	105
3.3.2.1	Schnittstellen, welche sind für Alarms & Events spezifiziert? .....	105
3.4	OPC XML .....	105
3.4.1	Einführung XML und SOAP .....	105
3.4.1.1	XML und SOAP, was ist das? .....	105
3.4.1.2	Web-Dienste, wozu dienen sie? .....	108
3.4.2	OPC-XML-Schnittstelle .....	108
3.4.2.1	OPC-XML-Schnittstelle, was leistet sie? .....	108
3.4.2.2	Web-Dienst OPC XML, wie funktioniert er? .....	111
3.4.2.3	Einfache Dienste Lesen / Schreiben, welche Methoden gibt es bei XML? .....	111
3.5	OPC Unified Architecture .....	113
3.5.1	Einführung in OPC UA .....	113
3.5.1.1	Einleitung .....	113
3.5.1.2	Die Sicherheit bei OPC UA .....	114
3.5.1.3	Die Kommunikationsarten von OPC UA .....	114
3.5.1.4	Der Namensraum von OPC UA .....	117
3.5.1.5	Weitere Eigenschaften von OPC UA .....	119
3.5.2	Die OPC UA-Schnittstelle .....	119
3.5.2.1	Welche Schnittstellenspezifikationen der OPC Unified Architecture gibt es? .....	119
3.5.2.2	Wie wird die Verbindung zu einem OPC UA-Server aufgenommen? .....	120
3.5.2.3	Wie kann der OPC UA-Namensraum durchsucht werden? .....	122
3.5.2.4	Wie können Daten gelesen und geschrieben werden? .....	122
3.5.2.5	Wie werden UA-Daten und Ereignisse beobachtet? .....	123
3.5.2.6	Wie kann nach Anmeldung besonders schnell gelesen und geschrieben werden? .....	127
3.5.2.7	Wie funktionieren Events, Conditions und Alarme? .....	127
3.5.2.8	Wie kann Redundanz bei OPC UA verwendet werden? .....	130
3.6	Leistungen von OPC Data Access und OPC Alarms & Events bei SIMATIC NET .....	133
3.6.1	Performance, wie kann sie optimal ausgenutzt werden? .....	133
3.6.2	OPC-Server von SIMATIC NET in der Automatisierungswelt, wie wird er eingesetzt? .....	135
3.6.3	OPC-Server für SIMATIC NET, was sind die Vorteile? .....	135
3.6.4	OPC-Server von SIMATIC NET, was leistet er? .....	137
3.6.5	Prozessdaten, wie wird optimal darauf zugegriffen? .....	138
3.6.6	Mengenoperationen, wie werden sie verwendet? .....	139
3.6.7	OPC-Cache, was ist das? .....	139
3.6.8	MaxAge, was ist das? .....	140
3.6.9	Dienste wenden den Cache an, wie geschieht das (Beispiel)? .....	140
3.6.10	Protokolle, für welche ist eine Optimierung möglich? .....	141
3.6.11	Blockdienste, wozu werden sie verwendet? .....	141
3.6.12	Blockdienste, wie werden Sie verwendet (Beispiel)? .....	142
3.6.13	Methoden, wie werden die geeigneten verwendet? .....	142
3.6.13.1	Synchrone Zugriffe, welche gibt es? .....	142
3.6.13.2	Asynchrone Zugriffe, welche gibt es? .....	143
3.6.13.3	Variablen beobachten, was geschieht da? .....	144
3.6.14	Percent Deadband, wie wird dieser Parameter verwendet? .....	146
3.6.15	Aktualisierungszeit, wie wird sie Item-spezifisch eingesetzt? .....	147

## *Inhaltsverzeichnis*

---

4	Literaturverzeichnis.....	151
	Index .....	153

## Security-Hinweise

### Hinweis

Siemens bietet für sein Automatisierungs- und Antriebsproduktportfolio IT-Security-Mechanismen, um einen sicheren Betrieb der Anlage/Maschine zu unterstützen. Unsere Produkte werden auch unter dem Gesichtspunkt IT-Security ständig weiterentwickelt. Wir empfehlen Ihnen daher, dass Sie sich regelmäßig über Aktualisierungen und Updates unserer Produkte informieren und nur die jeweils aktuellen Versionen bei sich einsetzen. Informationen dazu finden Sie unter:

(<http://support.automation.siemens.com>)

Hier können Sie sich für einen produktsspezifischen Newsletter registrieren.

Für den sicheren Betrieb einer Anlage/Maschine ist es darüber hinaus jedoch notwendig, die Automatisierungskomponenten in ein ganzheitliches IT-Security-Konzept der gesamten Anlage/Maschine zu integrieren, das dem aktuellen Stand der IT-Technik entspricht. Hinweise hierzu finden Sie unter:

(<http://www.siemens.com/industrialsecurity>)

Dabei sind auch eingesetzte Produkte von anderen Herstellern zu berücksichtigen.

## 1.1 Produkt - Historie

### Erweiterungen bei der SIMATIC NET PC Software

Mit der SIMATIC NET PC Software ab V8.1 werden 64-Bit-Betriebssysteme unterstützt.

Mit der SIMATIC NET PC Software ab V8.2 wird VMware ESXi unterstützt. \*\*\*

Mit OPC Unified Architecture Version 1.01 sind die bisherigen Funktionalitäten erweitert worden.

Die SIMATIC NET OPC UA-Server unterstützen zusätzlich:

- Transparente Server-Redundanz für das S7-Protokoll. \*
- Konfigurierbare optionale performante lokale Kommunikation. \*
- Konfigurierbare optionale Benutzer-Authentifizierung. \*

## 1.2 Willkommen bei SIMATIC NET

Allgemeine Neuerungen bei SIMATIC NET OPC:

- Das neue OPC Data Control für die .NET Schnittstellen unterstützt nun OPC UA mit einer einfachen Zertifikatverwaltung. Hiermit ist einfache grafische OPC-UA-Programmierung auch für .NET möglich. \*
- Mit einem neuen Symbol-Editor können nun Symboldateien vom performanten Typ ATI bearbeitet werden. CSV-Import/Export und STI-Import sind nun verfügbar. \*\*

(\*)= Details hierzu finden Sie in "Industrielle Kommunikation mit PG/PC Band 2 - Schnittstellen"

(\*\*)= Details hierzu finden Sie in "PC-Station in Betrieb nehmen"

(\*\*\*)= Details hierzu finden Sie im Installationshandbuch der "SIMATIC NET PC Software" ab V8.2

## 1.2 Willkommen bei SIMATIC NET

### SIMATIC NET - Wegweisende Erfolgskonzepte schwarz auf weiß

Mit Ihrer Entscheidung lassen wir Sie nicht alleine. Diese Dokumentation begleitet Sie auf Ihrem Weg zum erfolgreichen Einsatz von SIMATIC NET. Sie führt verständlich und anschaulich in das Thema ein und zeigt Ihnen, wie Sie einzelne Komponenten installieren sowie projektieren und wie Sie Programme auf der Basis von OPC erstellen. Sie werden sehen, was industrielle Kommunikation mit SIMATIC NET bedeuten kann – für Ihre Automatisierungswelt und vor allem für den Erfolg Ihres Unternehmens.

### SIMATIC NET - Eine gute Entscheidung

Sie kennen die Vorteile verteilter Automatisierungssysteme und wollen die Möglichkeiten industrieller Kommunikation optimal nutzen. Sie bauen auf einen starken Partner, Sie setzen auf innovative und zuverlässige Produkte. Mit SIMATIC NET haben Sie die richtige Wahl getroffen.

Diese Dokumentation baut auf Ihrem Wissen auf und lässt Sie vom Know-how der Spezialisten profitieren.

### Sind Sie Neuling?

Dann können Sie sich systematisch einarbeiten. Beginnen Sie in diesem **Band 1** mit der Einführung in die industrielle Kommunikation. Dort erfahren Sie alles Nötige über Kommunikationsprinzip und Funktionsumfang des SIMATIC NET OPC-Servers. Lesen Sie die Grundlagen zur OPC-Schnittstelle, machen Sie sich mit den Protokollen und deren Vorteile und Funktionen vertraut.

### Sind Sie Profi?

Dann können Sie sofort durchstarten. **Band 2** gibt Ihnen alle Informationen, die Sie zum Bedienen von SIMATIC NET benötigen.

Band 2 – Schnittstellen, Beitrags-ID:

61630140 (<http://support.automation.siemens.com/WW/view/de/61630140>)

**Folgen Sie gerne einem guten Beispiel?**

Dann finden Sie in den mitgelieferten Beispielprogrammen wertvolle Anregungen, die Ihnen helfen, eigene Vorstellungen umzusetzen.



# SIMATIC NET in der industriellen Kommunikation

## Überblick

Die Lektüre dieses Kapitels hilft Ihnen weiter, wenn Sie das Kommunikationsprinzip und den Funktionsumfang der einzelnen Protokolle in der industriellen Kommunikation mit SIMATIC NET® kennen lernen wollen.

Es erklärt Ihnen die Grundlagen der Kommunikationsnetze PROFIBUS und Industrial Ethernet, informiert Sie über die Funktionsweise der für diese Kommunikationsnetze in SIMATIC NET realisierten Protokolle und führt die Vor- und Nachteile dieser Protokolle auf. Zuletzt erhalten Sie einen Einblick in die Technologie und Anwendung von PROFINET und deren Umsetzung in SIMATIC NET.

Nach Abschluss dieses Kapitels sollten Sie die geeignetsten Mittel für die Umsetzung Ihrer Automatisierungsaufgaben erkennen können.

## Was ist eigentlich SIMATIC NET?

Die industrielle Kommunikation bildet das Rückgrat moderner Automatisierungslösungen. Die dort realisierten Kommunikationsnetze und -produkte ermöglichen eine durchgängige Kommunikation zwischen unterschiedlichsten Automatisierungskomponenten und -geräten.

SIMATIC NET ist der Name einer ganzen Familie von Kommunikationsnetzen und -produkten von Siemens. Die einzelnen Netze erfüllen dabei die verschiedensten Leistungs- und Anwendungsanforderungen in der Automatisierungstechnik.

## Was leistet SIMATIC NET?

SIMATIC NET bietet Lösungen für individuelle Kundenanforderungen im Bereich der industriellen Kommunikation. Die Kommunikationsnetze und -produkte von SIMATIC NET sind Teil der Totally Integrated Automation (TIA) von SIEMENS. Auf dieser Basis lassen sich branchenspezifische Automatisierungslösungen mit einem hohen Maß an Vollständigkeit und Durchgängigkeit der Kommunikationsfunktionen realisieren. SIMATIC NET vereinfacht die Inbetriebsetzung von Automatisierungssystemen unabhängig von den verwendeten Kommunikationsnetzen und -produkten.

Die Kommunikationsnetze und -produkte von SIMATIC NET lassen sich bezüglich ihres Leistungsspektrums und ihres Funktionsumfangs in Form einer Automatisierungspyramide darstellen.

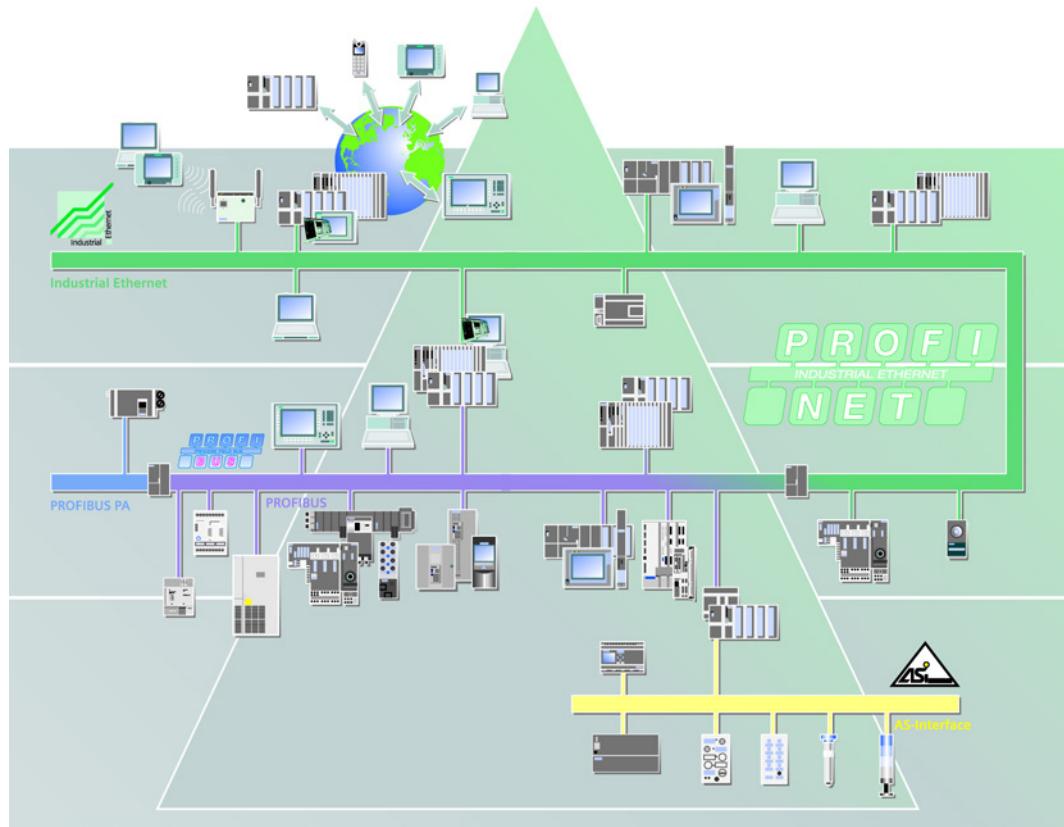


Bild 2-1 Die Automatisierungspyramide von SIMATIC NET

Die Automatisierungspyramide teilt sich in drei Ebenen, die Feldebene, die Zellebene und die Leitebene.

In der Feldebene findet die Prozess- oder Feldkommunikation statt. SIMATIC NET bietet für diese Ebene den PROFIBUS DP und das AS-Interface an.

In der Zellebene werden die erfassten Prozessdaten zum Bedienen und Beobachten auf verschiedene Automatisierungssysteme oder PCs verteilt. Hier werden im Rahmen von SIMATIC NET die Kommunikationsnetze Industrial Ethernet und PROFIBUS eingesetzt.

In der Leitebene finden übergeordnete Managementfunktionen statt. Dort werden Prozessdaten gespeichert, weiterverarbeitet oder Analysen durchgeführt. Für diese Aufgaben eignet sich Industrial Ethernet als Kommunikationsnetz.

## 2.1 SIMATIC NET und die Protokolle - Ein Überblick

### Welche Protokolle sind bei SIMATIC NET definiert?

Bei SIMATIC NET sind im wesentlichen zwei Kommunikationsnetze realisiert, PROFIBUS und Industrial Ethernet. Für beide Netze stehen Protokolle zur Verfügung, die eine durchgängige Kommunikation zwischen Automatisierungskomponenten und -geräten ermöglichen und die mit einem skalierten Funktionsumfang die verschiedenen Anwendungsanforderungen in der Automatisierungstechnik erfüllen.

Die in den nachfolgenden Tabellen aufgeführten Protokolle können mit PROFIBUS bzw. Industrial Ethernet eingesetzt werden:

Tabelle 2- 1 Protokolle für PROFIBUS

Protokoll	Beschreibung
SEND/RECEIVE-Protokoll	Einfache Kommunikationsdienste auf Basis von PROFIBUS FDL zum Datenaustausch mit S5- und S7-Geräten
DP-Master Klasse 1	Zyklisches Lesen der Eingangs- und Setzen der Ausgangsdaten der DP-Slaves.
DP-Master Klasse 2	Zyklischer Zugriff zur Diagnose und Inbetriebnahme eines DP-Systems.
DPC1	Zyklisches Lesen der Eingangs- und Setzen der Ausgangsdaten der DP-Slaves, azyklischer Zugriff auf Datensätze von DP-Slaves mit der DPV1-Erweiterung.
DPC2	Zyklischer Zugriff zur Diagnose und Inbetriebnahme eines DP-Systems, azyklischer Zugriff auf Datensätze von DP-Slaves mit der DPV1-Erweiterung.
DP-Slave	Erfassung, Umwandlung und Übermittlung von Prozesssignalen.
S7-Protokoll	Integrierte und optimierte Kommunikationsfunktion der SIMATIC S7-Systeme für verschiedenste Anwendungen.

Tabelle 2- 2 Protokolle für Industrial Ethernet

Protokoll	Beschreibung
SEND/RECEIVE-Protokoll	Einfache Kommunikationsdienste auf Basis von Transportprotokollen zum Datenaustausch mit S5- und S7-Geräten.
S7-Protokoll	Integrierte und optimierte Kommunikationsfunktion der SIMATIC S7-Systeme für verschiedenste Anwendungen.
S7-Protokoll (hochverfügbar)	Integrierte und optimierte Kommunikationsfunktion der SIMATIC S7-Systeme über redundante und hochverfügbare Verbindungswege
SNMP-Protokoll	Offenes Protokoll für die Administration von Netzwerken.
PROFINET IO	Kommunikation zwischen PROFINET-Geräten

### Das ISO/OSI-Referenzmodell

Für das bessere Verständnis der Funktions- und Wirkungsweise der für PROFIBUS und Ethernet realisierten Protokolle ist es wichtig, eine Spezifikation vorzustellen, die die vielfältigen Anforderungen an die Datenkommunikation standardisiert, nämlich das ISO/OSI-Referenzmodell.

Das nach der "Open Systems Interconnection" (OSI), einer Arbeitsgruppe der internationalen Standardisierungsorganisation ISO, benannte Schichtenmodell ist ein Referenzmodell für die Datenübertragung in Netzwerken. Es beschreibt 7 hierarchisch angeordnete Schichten. Jede Schicht nimmt seine eigene spezielle Aufgabe wahr.

Als Referenzmodell ist OSI kein vorgeschriebener Standard. Dennoch orientieren sich viele Produkte im Telekommunikations- und Netzwerkbereich am ISO/OSI-Referenzmodell.



Bild 2-2 Das ISO/OSI-Referenzmodell

### Welche Schichten sind im ISO/OSI-Referenzmodell definiert?

Die 7 festgelegten Schichten bilden drei Funktionsebenen: Die erste und zweite Schicht stellen die hardwarenahe Ebene dar, die dritte und vierte Schicht bilden die Übertragungsebene und die Schicht fünf bis sieben realisieren die anwendernahe Ebene. Im einzelnen sind die Schichten wie folgt definiert:

- Schicht 1:  
Die Bitübertragungsschicht ist für die physikalische Verbindung zwischen zwei Geräten zuständig. Sie überträgt die Daten von einem Gerät über ein Netzwerk zum Anderen.
- Schicht 2:  
Die Datensicherungsschicht ist für die korrekte Übermittlung der Daten zuständig. Sie fasst einzelne Bits zu Datenblöcken zusammen und fügt Adressierungsinformationen hinzu, die zur Übertragung der Daten von einem Gerät zum Anderen benötigt werden. Außerdem führt die Schicht zur Sicherung der Datenübertragung eine Fehlerkontrolle durch.

- Schicht 3:  
Die Netzwerkschicht sorgt für die Wegewahl und damit die richtige Weiterleitung der Datenblöcke. Sie übernimmt die Adressierung der Pakete und deren Routing im Netz. Ein Beispiel für diese Schicht ist das Internet-Protokoll (IP).
- Schicht 4:  
Die Transportschicht regelt die Übermittlung von Datenpaketen. Sie überprüft, ob alle Pakete vollständig empfangen worden sind. Hierfür werden sogenannte Transportverbindungen zwischen zwei Geräten bereitgestellt. Ein typisches Beispiel für die Schicht 4 ist das Transmission Control Protocol (TCP).
- Schicht 5:  
Die Verbindungsschicht stellt eine andauernde Verbindung zwischen den Geräten her, zwischen denen Daten übertragen werden sollen. Die Schicht sorgt für den Auf- und Abbau der Verbindung und dafür, dass die Verbindung bestehen bleibt.
- Schicht 6:  
Die Darstellungsschicht ist für die Umwandlung der Daten in das für die jeweilige Anwendung erforderliche Format zuständig. Außerdem werden die Daten für den Transport aufbereitet. Hierzu zählen die Komprimierung und die Verschlüsselung von Daten.
- Schicht 7:  
Die Applikationsschicht stellt Anwendungen zur Verfügung, die Daten zur weiteren Verarbeitung entgegennimmt oder Daten für die Übertragung bereitstellt. Klassische Beispiele hierfür wären Mail-Programme oder Internet-Browser.

## 2.2 Industrielle Kommunikation mit PROFIBUS - Ein Überblick

### 2.2.1 PROFIBUS, was ist das?

#### Das ist PROFIBUS

PROFIBUS ist das offene und international genormte (EN50170) Bussystem für die Prozess- und Feldkommunikation mit Feldgeräten und zur Datenkommunikation innerhalb einer Automatisierungszelle. Die Einsatzmöglichkeiten von PROFIBUS erstrecken sich von der Fertigungs- und Prozessautomatisierung bis hin zur Gebäudeautomation.

#### Welche Eigenschaften hat PROFIBUS?

PROFIBUS zeichnet sich durch folgende Eigenschaften aus:

- Datenübertragung über kostengünstige Kommunikationsmedien, wie zum Beispiel Zweidrahtleitung.
- Weites Anwendungsspektrum, da Automatisierungsgeräte, Bedien- und Beobachtungsgeräte über einen einheitlichen Bus kommunizieren.

- Genormte Datenkommunikation nach EN 50170, EC 61158 (Dienste u. Protokoll) und IEC 61784.
- Inbetriebnahme, Projektierung und Fehlersuche können von jeder Stelle des Bussegmentes aus durchgeführt werden.
- Hoher Investitionsschutz, da bestehende PROFIBUS-Anlagen ohne Rückwirkung erweitert werden können.

## 2.2.2 PROFIBUS, wie ist die Funktionsweise?

### So funktioniert PROFIBUS

Die PROFIBUS-Spezifikation lässt genügend Freiraum für die Implementierung verschiedener Protokolle, die je nach Einsatzgebiet für bestimmte Aufgaben optimiert sind. Dabei gewährleistet die Datenverbindungsschicht FDL (Schicht 2 des ISO/OSI-Referenzmodells) einen einheitlichen Buszugriff nach dem Token-Passing-Verfahren.

### Wie funktioniert das Token-Passing-Verfahren am PROFIBUS?

Das Token-Passing-Verfahren steuert den Zugriff auf den Bus, d. h. nur der Teilnehmer am Bus, der das Senderecht (Token) besitzt, darf auch senden. Das Senderecht wird nach einer festgelegten Zeit (Token-Haltezeit) an den nächsten Teilnehmer weitergegeben. Am Ende eines Zyklus erhält wieder der erste Teilnehmer den Token.

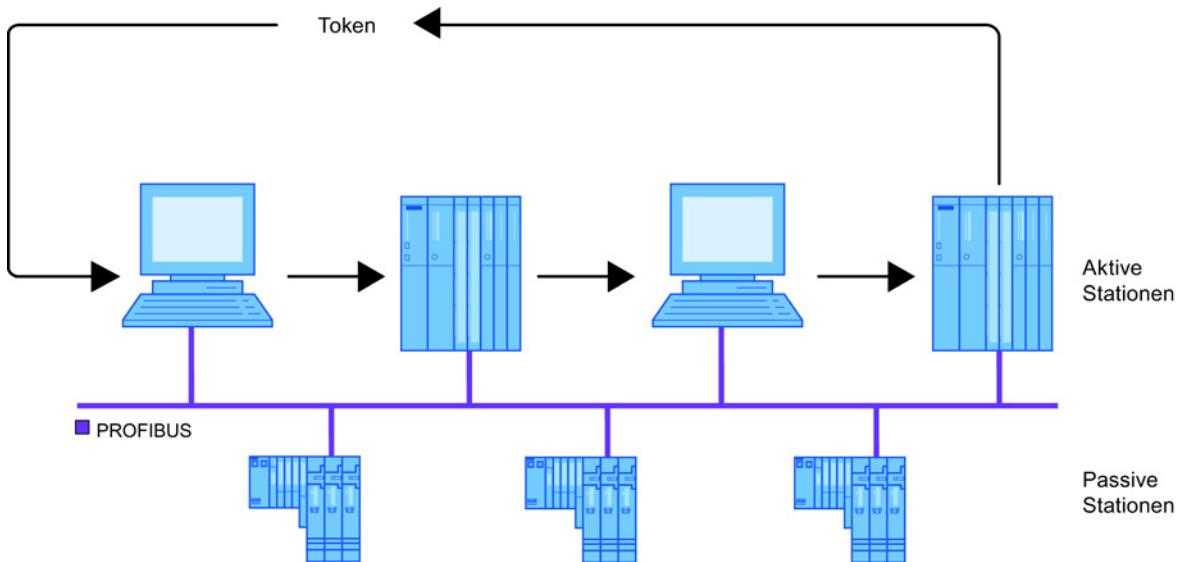


Bild 2-3 Das Token-Passing-Verfahren am PROFIBUS

### Was ist das Master-Slave-Prinzip?

Basiert eine Kommunikation auf dem Master-Slave-Prinzip, so gibt es eine Station, den Master, die selbstständig eine Kommunikation mit einem Slave anstoßen kann. Der Slave antwortet daraufhin dem Master. Mit der Antwort kann der Slave Daten übermitteln. Im Gegensatz zum Master wird ein Slave jedoch nicht von selbst aktiv.

### Wie funktioniert das Master-Slave-Prinzip am PROFIBUS?

Am PROFIBUS-Netz werden zwei Arten von Teilnehmern unterschieden:

- Aktive Teilnehmer (Master) steuern die Kommunikation auf dem Bus. Jeder aktive Teilnehmer erhält einmal pro Zyklus den Token und kann dann mit aktiven und passiven Teilnehmern kommunizieren. Nach Ablauf der Token-Haltezeit gibt er den Token an den nächsten Master weiter (Token-Passing). Ein DP-Master oder ein S7-Server sind beispielsweise aktive Stationen.
- Passive Teilnehmer (Slaves) können nicht selbstständig eine Kommunikation aufnehmen. Sie erhalten den Token nicht und antworten nur auf die an sie gerichteten Anfragen einer aktiven Station. Ein typisches Beispiel für eine passive Station ist ein DP-Slave.

### 2.2.3 PROFIBUS, wie sieht er im ISO/OSI-Referenzmodell aus?

#### So sieht PROFIBUS im ISO/OSI-Referenzmodell aus

PROFIBUS orientiert sich am ISO/OSI-Referenzmodell, realisiert aber nicht alle Schichten. Im folgenden Bild wird verdeutlicht, welche Schichten des ISO/OSI-Referenzmodells in den verschiedenen für PROFIBUS definierten Protokolle berücksichtigen sind. Jedes Protokoll stellt dabei auf der jeweils letzten realisierten Schicht eine Anwenderschnittstelle zur Verfügung, mit Hilfe derer die Dienste für die Datenkommunikation genutzt werden können.

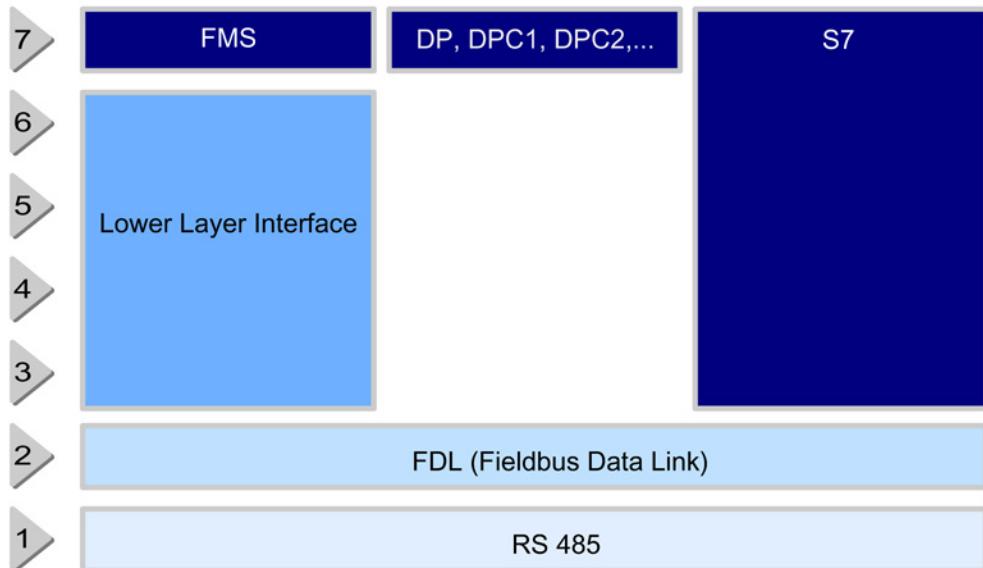


Bild 2-4 PROFIBUS im ISO/OSI-Referenzmodell

## 2.3 Industrielle Kommunikation mit Ethernet - Ein Überblick

### 2.3.1 Industrial Ethernet, was ist das?

#### Das ist Industrial Ethernet

Industrial Ethernet ist ein leistungsfähiges Kommunikationsnetz nach dem internationalen Standard IEEE802.3 (Ethernet), das für die Anforderungen im industriellen Einsatz optimiert wurde.

#### Welche Eigenschaften hat Industrial Ethernet?

Es zeichnet sich durch folgende Eigenschaften aus:

- Vernetzung unterschiedlicher Anwendungsbereiche wie Verwaltung und Fertigung.
- Robuster Aufbau und elektromagnetische Störfestigkeit.
- Hohe Übertragungsleistung (100 Mbit/s und 1 Gbit/s).
- Unterstützung verschiedener Übertragungsmedien, zum Beispiel Twisted Pair oder Lichtwellenleiter.
- Skalierbare Leistung durch Switched-Ethernet-Technologie.
- Hohe Verfügbarkeit durch redundante Netz-Topologien.

- Übertragung von großen Datenmengen durch Anwendung unterschiedlicher Transportprotokolle.
- Echtzeit-Übertragung mittels PROFINET IO ist verfügbar.

### Wie ist ein Industrial Ethernet aufgebaut?

Die Topologie bei Industrial Ethernet entspricht typischerweise einer Sternstruktur.

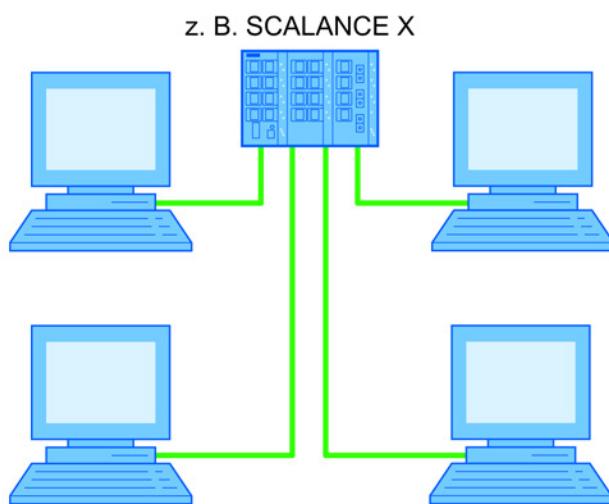


Bild 2-5 Typische Topologie des Industrial Ethernet

### Wie werden große Datenmengen über Industrial Ethernet übertragen?

Ein Merkmal der Datenübertragung über Ethernet ist, dass die maximale Datenpaketgröße begrenzt ist. Müssen größere Daten übertragen werden, müssen sie in mehrere Pakete aufgeteilt werden. Diese Aufgabe übernehmen die verschiedenen Transportprotokolle:

Das ISO-Transportprotokoll unterstützt die Aufteilung großer Datenmengen in Datenpakete und kann somit große Datenmengen übertragen. Es entspricht der Schicht 4 des ISO/OSI-Referenzmodells.

ISO on TCP entspricht dem Standard TCP/IP mit der Erweiterung RFC1006 gemäß der Schicht 4 des ISO/OSI-Referenzmodells. Durch die Erweiterung wird die Aufteilung großer Datenmengen in Datenpakete unterstützt und dadurch die Übertragung großer Datenmengen gewährleistet. RFC1006 ist ein offizieller Standard und wird von vielen Herstellern unterstützt.

TCP/IP native (ohne RFC1006) unterstützt die Aufteilung großer Datenmengen in Datenpakete nicht. Diese Aufgabe muss nun das Anwenderprogramm beider Kommunikationspartner realisieren.

### 2.3.2 Switched-Ethernet, was ist das?

#### Das ist Switched-Ethernet

Switched-Ethernet unterteilt das Netzwerk in Segmente, die mit Switches verbunden sind.

#### Welches sind die Vorteile des Switched-Ethernet?

- Die Unterteilung in Segmente verringert die gesamte Netzlast.
- Jedes Segment hat die volle Datenübertragungsrate zur Verfügung.
- Die Kollisionen der Datenpakete im Vollduplex-Modus ist ausgeschlossen, da für das Senden und Empfangen jeweils eine eigene Leitung zur Verfügung steht.

### 2.3.3 Industrial Ethernet, welche Schichten werden im ISO/OSI-Referenzmodell realisiert?

#### Wie sieht Ethernet im ISO/OSI-Referenzmodell aus?

Basierend auf den unterschiedlichen Schichten des Referenzmodells bietet Industrial Ethernet mehrere Anwenderschnittstellen, mit Hilfe derer die Kommunikationsdienste der verschiedenen Protokolle genutzt werden können.

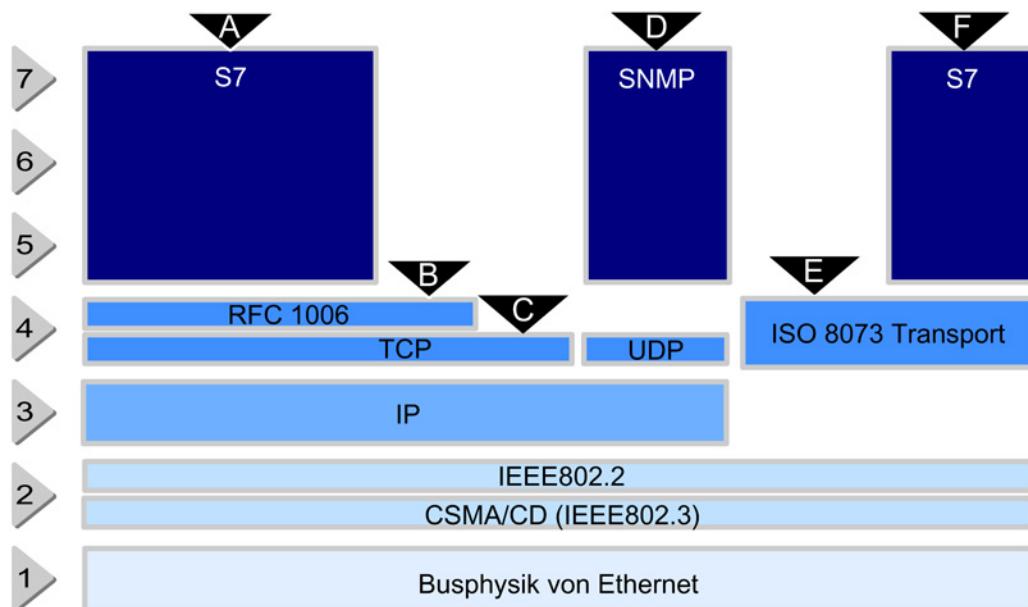


Bild 2-6 Ethernet im ISO/OSI-Referenzmodell

Symbol	Protokoll	Beschreibung
A, F	S7-Kommunikation	Einheitliche Anwenderschnittstelle für TCP/IP (A) und ISO (F) über S7-Funktionen.
B, E	Offene Kommunikationsdienste (SEND/RECEIVE)	Kommunikationsdienste auf Basis der ISO-Transportschnittstelle zum Datenaustausch mit S5- und S7-Geräten, sowie Fremdgeräten. Auf TCP/IP ist ein Adapter (RFC1006) notwendig. Damit ist die offene SEND/RECEIVE- Anwenderschnittstelle für TCP/IP (B) und ISO (E) einheitlich.
C	TCP/IP native	Einfache Kommunikationsdienste auf Basis von TCP/IP zum Datenaustausch mit beliebigen Geräten, welche TCP/IP unterstützen.
D	SNMP-Kommunikation	Kommunikationsdienste auf Basis von UDP/IP zum Datenaustausch mit beliebigen SNMP-fähigen Geräten.

## 2.4 PROFINET - Ein Überblick

### 2.4.1 PROFINET, was ist das?

#### Das ist PROFINET

PROFINET steht für **PROcess Field NET** und ist der innovative und offene Standard für die industrielle Automatisierung auf Basis von Industrial Ethernet. Mit PROFINET können Lösungen in den Bereichen Fertigungsautomatisierung und Motion Control realisiert werden. Im Rahmen von Totally Integrated Automation (TIA) ist PROFINET die konsequente Fortführung des etablierten Feldbusystems PROFIBUS und dem Kommunikationsbus für die Zellebene, Industrial Ethernet. Mit PROFINET können einfache dezentrale Feldgeräte sowie zeitkritische Anwendungen (PROFINET IO) genauso in die Ethernet-Kommunikation eingebunden werden, wie verteilte Automatisierungssysteme auf Basis von Komponenten (Component based Automation).

PROFINET deckt die Anforderungen der Automatisierungstechnik komplett ab. Die Erfahrung von PROFIBUS und Industrial Ethernet werden in PROFINET zusammengeführt. Die Nutzung der offenen Standards, die einfache Handhabung und die Integration von bestehenden Anlagenteilen bestimmten von Beginn an die Definition von PROFINET.

PROFINET ist heute als herstellerübergreifendes Kommunikations-, Automatisierungs- und Engineering-Modell der PROFIBUS Nutzerorganisation e.V. (PNO) definiert und in der IEC 61158 integriert.

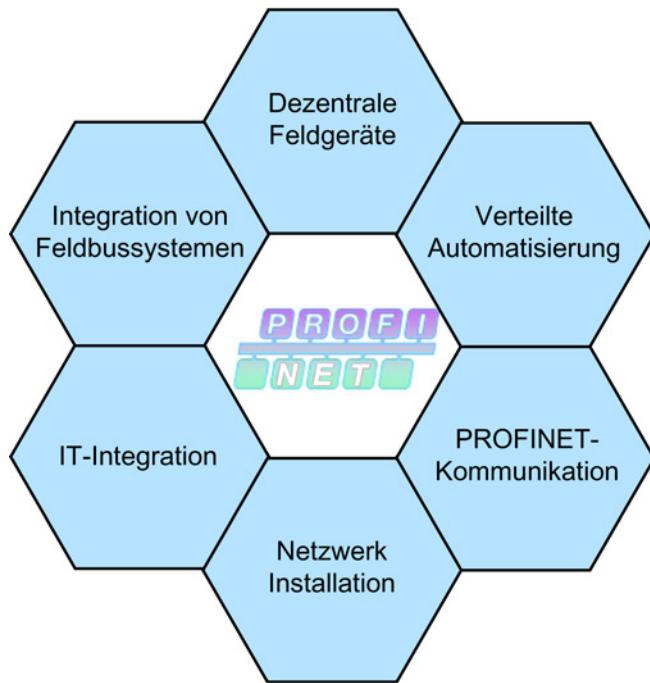


Bild 2-7      Definition PROFINET nach PNO

### Welche Ziele hat PROFINET?

Die Zielsetzung von PROFINET ist:

- eine durchgängige Kommunikation über Feldbus und Ethernet
- eine offene, verteilte Automatisierung
- die Kommunikation in Echtzeit
- die Verwendung von offenen Standards

## 2.4.2 PROFINET, auf welcher Kommunikation basiert es?

### PROFINET basiert auf folgender Kommunikation

PROFINET basiert auf der Ethernet-Kommunikation. Sie ist skalierbar und unterscheidet drei Leistungsstufen:

1. TCP, UDP und IP für zeit-unkritische Daten wie azyklisches Lesen und Schreiben von Datensätzen, Parametrierung und Konfiguration (Non Real Time / NRT)
2. Real Time (RT), performante Kommunikation für zeitkritische Prozessdaten im Bereich der Prozessautomatisierung
3. Isochrones Real Time (IRT), hochperformante, deterministische und taktsynchrone Kommunikation für zeitkritische Prozessdaten im Bereich Motion Control

### Welche Echtzeitkommunikation ist bei PROFINET definiert?

In der Automatisierungstechnik gibt es Anwendungen, die erhöhte Aktualisierungs- und Reaktionszeiten fordern. Im PROFINET-Standard sind deshalb Mechanismen für die Echtzeitkommunikation definiert. Wie eingangs erwähnt, ist die Echtzeitkommunikation skaliert:

- Der **Real Time-Kanal** (RT-Kanal) ist ein Echtzeit-Kommunikationskanal, der direkt auf der Schicht 2 des Ethernet aufsetzt und das RT-Protokoll verwendet. Diese Lösung minimiert die Durchlaufzeiten der Kommunikationsebenen, da einige von ihnen wegfallen. Es wird eine Leistungssteigerung bezüglich der Aktualisierungsrate von Prozessdaten erreicht, weil Daten schneller zur Übertragung bereitstehen und von den Anwenderprogrammen, die die Daten empfangen, schneller verarbeitet werden können. Es werden Aktualisierungs- und Reaktionszeiten von 5- 10 ms erreicht.
- Der **Isochrone Real Time-Kanal** (IRT-Kanal) ist speziell für Motion Control-Applikationen entwickelt worden. Hier gibt es Anforderungen bezüglich der Aktualisierungs- und Reaktionszeit von weniger als 1 ms. Um dies zu erreichen, setzt der IRT-Kanal auf die Schicht 2 des Fast Ethernet (100 Mb/s) auf und verwendet das IRT-Protokoll. Zusätzlich erfolgt die Übertragung von Daten mit einem zeitschlitzgesteuerten Übertragungsverfahren. Durch die zeitliche Synchronisation der Kommunikationspartner am Ethernet kann ein Zeitschlitz festgelegt werden, mit dessen Hilfe die Kommunikation in einen deterministischen und einen offenen Kanal aufgespalten wird. Im deterministischen Kanal werden die zeitkritischen Real Time-Daten übertragen, während die zeitunkritischen Daten im offenen Kanal transportiert werden.

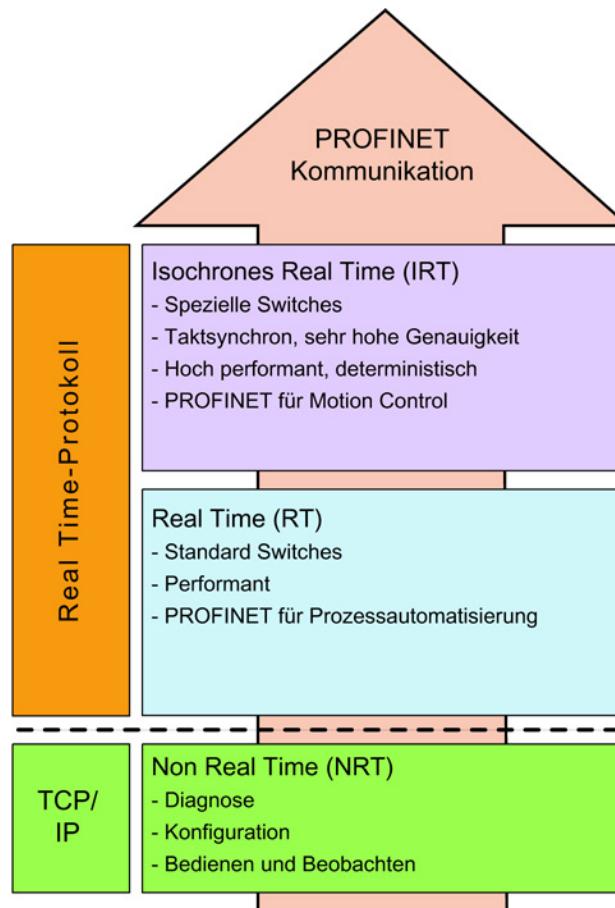


Bild 2-8 PROFINET-Echtzeitkommunikation bei SIMATIC NET

## 2.4.3 PROFINET, welche Schichten werden im ISO/OSI Referenzmodell realisiert?

### PROFINET im ISO/OSI-Referenzmodell

Basierend auf den unterschiedlichen Schichten des Referenzmodells bietet PROFINET im wesentlichen zwei Kommunikationskanäle zur Datenübertragung, den RT-Kanal und den IRT-Kanal.

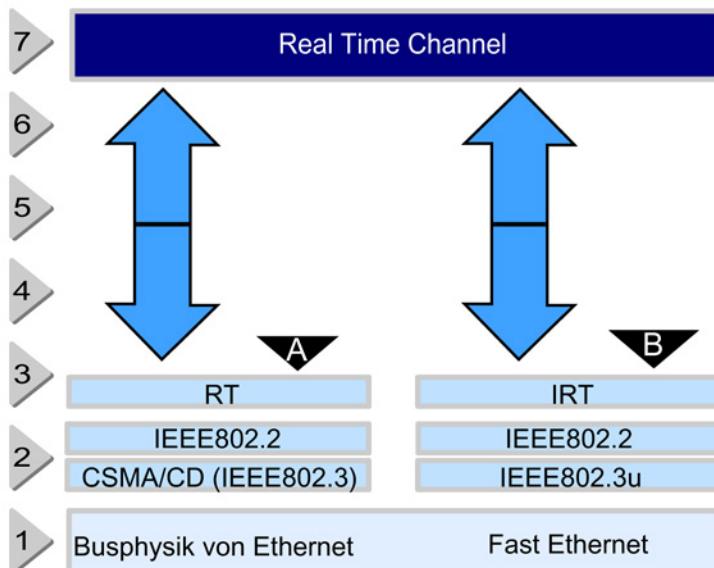


Bild 2-9 PROFINET im ISO/OSI-Referenzmodell

Symbol	Beschreibung
A	RT-Kanal für die Kommunikation mit PROFINET IO
B	IRT-Kanal für die Kommunikation mit PROFINET IO

## **2.5      SEND/RECEIVE-Protokoll**

### **2.5.1      SEND/RECEIVE-Protokoll, was ist das?**

#### **Das SEND/RECEIVE-Protokoll**

Das SEND/RECEIVE-Protokoll ist ein Kommunikationsprotokoll zur Übertragung von Daten über PROFIBUS und Industrial Ethernet. Es ermöglicht einen einfachen Datenaustausch zwischen Automatisierungsgeräten. Mit dem SEND/RECEIVE-Protokoll können SIMATIC S5-Geräte, SIMATIC S7-Geräte, PCs, Workstations und Fremdgeräte miteinander kommunizieren.

#### **Wie unterscheidet sich das SEND/RECEIVE-Protokoll bei PROFIBUS und bei Ethernet?**

- Bei PROFIBUS setzt das SEND/RECEIVE-Protokoll auf den FDL-Diensten auf, während es bei Ethernet die dort verfügbaren Dienste der Transportebene benutzt.
- Die zu übertragende Datenmenge ist bei PROFIBUS auf 246 Bytes, bei Ethernet auf 4096 Bytes begrenzt.
- PROFIBUS verfügt im Gegensatz zu Ethernet nicht über Variabldienste.

### **2.5.2      SEND/RECEIVE-Protokoll, wie sieht eine typische Anlagenkonfiguration aus?**

Dieses Kapitel zeigt, wie typische Anlagenkonfigurationen bei PROFIBUS und Industrial Ethernet aussehen können, in denen die Datenkommunikation zwischen verschiedenen Geräten mittels SEND/RECEIVE-Protokoll realisiert ist.

#### **Beispiel einer Anlagenkonfiguration für das SEND/RECEIVE-Protokoll bei PROFIBUS**

Zur Kommunikation mit dem SEND/RECEIVE-Protokoll über PROFIBUS stehen im SIMATIC NET-Spektrum Kommunikationsbaugruppen für Steuerungen der Familien SIMATIC S5, SIMATIC 505 und SIMATIC S7 sowie für PCs, Workstations und Fremdgeräte zur Verfügung.

Die SIMATIC S7 bietet hierfür die Kommunikationsbaugruppen CP 342-5 und CP 443-5 und für PCs z. B. den CP 5623.

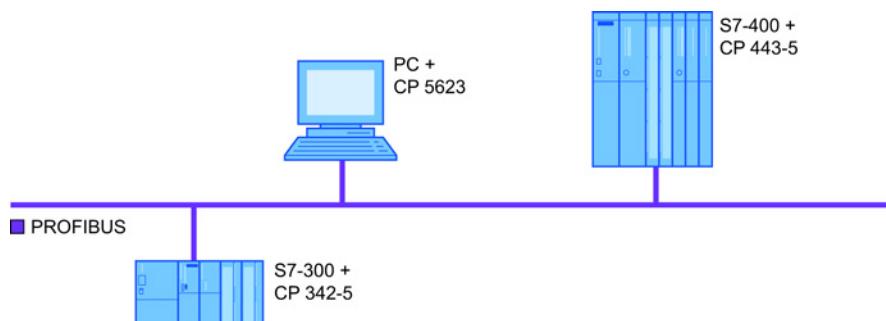


Bild 2-10 Typische Anlagenkonfiguration bei PROFIBUS

### Beispiel einer Anlagenkonfiguration für das SEND/RECEIVE-Protokoll bei Ethernet

Zur Kommunikation mit dem SEND/RECEIVE-Protokoll über Ethernet stehen im SIMATIC NET-Spektrum ebenso Kommunikationsbaugruppen für Steuerungen der Familien SIMATIC S5, SIMATIC 505 und SIMATIC S7 sowie für PCs und Workstations zur Verfügung.

Die SIMATIC S7 bietet hierfür typischerweise die Kommunikationsbaugruppen CP 343-1 und CP 443-1 und für PCs sowie Workstations z. B. den CP 1623.

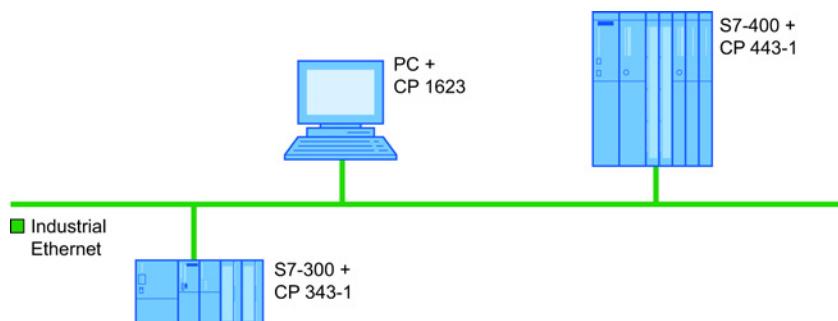


Bild 2-11 Typische Anlagenkonfiguration bei Ethernet

### 2.5.3 SEND/RECEIVE-Protokoll wie funktioniert es?

#### So funktioniert das SEND/RECEIVE-Protokoll bei PROFIBUS

Das SEND/RECEIVE-Protokoll bei PROFIBUS basiert auf der einfachen Übertragung von Daten in einem FDL-Datenblock. Dabei verwendet es direkt die Dienste der Datenübertragungsschicht von PROFIBUS, des Fieldbus Data Link (FDL). Zur Übermittlung der Daten stellt der Empfänger einen Empfangspuffer bereit, in den der Sender seine zu übertragenden Daten hineinschreibt.

Die Datenkommunikation über das SEND/RECEIVE-Protokoll ist ausschließlich zwischen aktiven PROFIBUS Teilnehmern möglich. Durch die Größe der FDL-Datenblöcke ist die Nutzdatenlänge auf maximal 246 Bytes pro Telegramm beschränkt. Zum Datenaustausch werden die Dienste SDA (Send Data with Acknowledge) und SDN (Send Data with No Acknowledge) verwendet.

Für eine Kommunikation über das SEND/RECEIVE-Protokoll ist kein Verbindungsaufbau notwendig.

### **So funktioniert das SEND/RECEIVE-Protokoll bei Ethernet**

Im Gegensatz zur Datenkommunikation bei PROFIBUS setzt das SEND/RECEIVE-Protokoll bei Industrial Ethernet auf der Transportebene des ISO/OSI-Referenzmodells auf. Sie bietet damit dem Anwender die Leistungen der Transportebene wie Verbindungen, Flusskontrolle und Datensegmentierung.

Das SEND/RECEIVE-Protokoll verwendet die auf Industrial Ethernet verfügbaren Transportprotokolle, das ISO- Transportprotokoll, das TCP/IP- Transportprotokoll mit und ohne RFC1006.

Das **ISO-Transportprotokoll** ist in der internationalen Norm ISO 8073 Class 4 festgelegt und stellt Dienste für die Übertragung von Daten bereit.

Aufgrund der möglichen Daten-Segmentierung, d.h. Nutzdaten können auf ISO-Transport in mehrere Datentelegramme segmentiert werden, können mit dem ISO-Transportdienst große Datenmengen übertragen werden. Der ISO-Transportdienst ermöglicht die Kommunikation mit einem beliebigen Kommunikationspartner der Senden bzw. Empfangen von Daten gemäß ISO-Transport unterstützt.

Das **ISO on TCP (RFC1006)** Protokoll entspricht dem Standard TCP/IP (Transmission Control Protocol/Internet Protocol) mit der Erweiterung RFC1006. Die RFC1006-Erweiterung ist erforderlich, da TCP eine Datenkommunikation ohne Segmentierung der Daten in Telegrammen realisiert. RFC1006 beschreibt, wie die Dienste vom ISO-Transportprotokoll, und damit die Segmentierung der Daten auf TCP abgebildet werden. RFC1006 ist ein offizieller Standard und wird von vielen Herstellern eingesetzt.

Das **TCP/IP native Protokoll (ohne RFC1006)** bietet die Möglichkeit, mit jedem beliebigen Kommunikationspartner, der TCP/IP beherrscht, zu kommunizieren. Da die Transportschicht von TCP/IP jedoch einen unstrukturierten Datenstrom liefert, fällt die Aufgabe der Segmentierung dem Anwender zu. Auf einer Kommunikationsverbindung müssen beide Partner über die Größe des zu übertragenden Datenpakets informiert sein, damit das richtige Paket aus dem Datenstrom herausgegriffen werden kann.

Die Datenkommunikation mittels dem SEND/RECEIVE-Protokoll über Ethernet verläuft ausschließlich verbindungsorientiert. Das bedeutet, dass erst eine Transportverbindung zum Partnergerät aufgebaut werden muss, bevor Daten übertragen werden können. Beim Verbindungsaufbau gibt es einen aktiven und einen passiven Kommunikationsteilnehmer. Der aktive Teilnehmer stößt den Verbindungsaufbau zum Partnergerät an. Die Zuständigkeit für den Verbindungsaufbau zwischen zwei Geräten wird in der Verbindungsprojektierung festgelegt.

## 2.5.4 SEND/RECEIVE-Protokoll, welche Kommunikationsdienste stehen zur Verfügung?

### Diese Kommunikationsdienste stellt das SEND/RECEIVE-Protokoll zur Verfügung

Für den Datenaustausch bietet das SEND/RECEIVE-Protokoll Block- und Variabldienste. Die Blockdienste dienen zur Übertragung von unstrukturierten Datenblöcken zwischen zwei Automatisierungsgeräten und sind sowohl bei PROFIBUS als auch bei Ethernet verfügbar. Mit den Variabldiensten werden strukturierte Daten, also Variablen, die in Automatisierungsgeräten definiert sind, übertragen. Variablen sind sogenannte Datenobjekte in Automatisierungsgeräten. Beispiele hierfür sind, Datenbausteine, Ein- und Ausgänge der Peripherie, Merker, Timer, Zähler und Systembereiche. Die Variabldienste können nur auf Ethernet genutzt werden.

Für PCs gibt es bei PROFIBUS einige zusätzliche Dienste, die nicht für Datenkommunikation, sondern für Diagnose- und Informationszwecke gedacht sind:

- Ermittlung der Busparameter und der eigenen Stationsadresse
- Bestimmung der Liste der am Bus befindlichen Teilnehmer
- Teilnehmeridentifikation der lokalen Station und der Partnerstationen

### Wie funktionieren die Blockdienste?

Die Blockdienste des SEND/RECEIVE-Protokolls umfassen zwei Kommunikationsdienste, SEND und RECEIVE.

Der Dienst SEND wird auf dem Gerät benutzt, von dem Daten gesendet werden. Das Senden von Daten muss vom Sender explizit angestoßen werden. Das Gerät, auf dem Daten empfangen werden, muss den Dienst RECEIVE aktivieren, damit eine Empfangsbereitschaft hergestellt ist.

Bei den Kommunikationsdiensten SEND und RECEIVE für die Datenkommunikation auf PROFIBUS handelt es sich um einfache Dienste, ohne Verbindungsüberwachung, sodass der Ausfall des Partnergerätes nicht erkannt wird. Eine solche Überwachung kann nur über ein entsprechendes Anwenderprogramm realisiert werden, indem z.B. eine zyklische Übertragung von Daten angestoßen wird und auf Seiten des empfangenden Gerätes die zyklischen Daten kontrolliert werden.

### Wie funktionieren die Variabldienste?

Die Variabldienste des SEND/RECEIVE-Protokolls umfassen zwei Kommunikationsdienste, FETCH und WRITE. Diese Kommunikationsdienste sind ausschließlich bei Ethernet verfügbar.

Bei Ausführung des Dienstes FETCH wird ein Auftrag vom PC zum Partnergerät übermittelt, in dem die aktuellen Werte von bestimmten Variablen angefordert werden. Das Partnergerät bestätigt den Auftrag mit einem Datenblock, der die aktuellen Werte der angeforderten Variablen enthält.

Mit dem Dienst WRITE kann der PC aktuelle Werte von bestimmten Variablen an das Partnergerät senden. Das Partnergerät wertet diese aus und setzt seinerseits die Variablen mit den übermittelten Werten. Anschließend wird der Dienst vom Partnergerät bestätigt.

## **2.5.5      SEND/RECEIVE-Protokoll, wie wird es projektiert?**

### **So wird das SEND/RECEIVE-Protokoll projektiert**

Für die Kommunikation mit dem SEND/RECEIVE-Protokoll sind vor der Verwendung Verbindungen zu projektiern. Hierfür steht das Projektierungswerkzeug "SIMATIC STEP 7 Professional" zur Verfügung. Die projektierten Verbindungen werden durch einen eindeutig festgelegten Verbindungsnamen identifiziert. Es sind für das SEND/RECEIVE-Protokoll vier Verbindungstypen vordefiniert, die gleichzeitig die Art der Verbindung beschreiben:

- FDL-Verbindung: Verbindung über PROFIBUS
- ISO-Transportverbindung: Verbindung über Ethernet mittels ISO-Transportprotokoll
- ISO-on-TCP-Verbindung: Verbindung über Ethernet mittels ISO-on-TCP-Protokoll
- TCP-Verbindung: Verbindung über Ethernet mittels TCP/IP native Protokoll

Für jede projektierte Verbindung sind Parameter einzustellen, für die das Projektierungswerkzeug beim Anlegen der Verbindung Defaultwerte vorgibt, die der Anwender ohne Änderung übernehmen kann. Die Parameter sind zum Beispiel:

- die Adresse des Kommunikationspartners
- der Dienstzugangspunkt (SAP).

## **2.5.6      SEND/RECEIVE-Protokoll, welches sind die Vor- und Nachteile?**

### **Das sind die Vorteile des SEND/RECEIVE-Protokolls bei PROFIBUS**

Das offene SEND/RECEIVE-Protokoll bei PROFIBUS bietet folgende Vorteile:

- Es können große Datenblöcke bis 246 Bytes übertragen werden.
- Es ist keine Netzlast vorhanden, wenn keine Daten übertragen werden.
- Das Senden von "Broadcast"-Telegramme an mehrere Teilnehmer ist möglich.
- Der strukturierte Zugriff auf Datenblöcke auf dem PC ist möglich.
- Die Kommunikation mit SIMATIC S5 und SIMATIC S7 Geräten ist möglich.
- PC/PGs können miteinander kommunizieren.

### **Das sind die Nachteile des SEND/RECEIVE-Protokolls bei PROFIBUS**

Das SEND/RECEIVE-Protokoll bei PROFIBUS hat folgende Nachteile:

- Der Empfänger kann die Datenübertragung nicht anstoßen. Er muss auf die Übermittlung von Daten durch den Sender warten.
- Es gibt keine Überwachung des Empfängers bei dessen Ausfall oder einer Unterbrechung des Netzes.
- Es gibt keine Routingfähigkeit (Weiterleitung eines Auftrags auf andere Netze).

### **Das sind die Vorteile des SEND/RECEIVE-Protokolls bei Ethernet**

Das SEND/RECEIVE-Protokoll bei Ethernet bietet folgende Vorteile:

- Es können durch Segmentierung größere Datenblöcke bis 64 Kbyte übertragen werden.
- Wenn vom Anwender keine Datenübertragung angestoßen ist, ist keine Netzlast vorhanden.
- Es ist ein strukturierter Zugriff auf Datenblöcke möglich.
- Eine Kommunikation mit S5- und S7-Geräten sowie PCs ist möglich.
- Durch die Variablenleidienste ist ein flexibler Zugriff auf Daten möglich.

### **Das sind die Nachteile des SEND/RECEIVE-Protokolls bei Ethernet**

Das offene (SEND/RECEIVE)-Protokoll bei Ethernet hat folgende Nachteile:

- Der Empfänger kann die Datenübertragung nicht anstoßen. Er muss auf die Übermittlung von Daten durch den Sender warten.
- Die Daten müssen in einem Puffer liegen oder durch ein Anwenderprogramm im Partnergerät in einen Puffer kopiert werden.
- Bei der Verwendung der Variablenleidienste ist im Vergleich zu den Blockdiensten der Datendurchsatz geringer.
- Zur Beobachtung von Variablenveränderungen muss zyklisch auf das Partnergerät zugegriffen werden, das bedeutet höhere Netzlast.

## **2.6 Das DP-Protokoll**

### **2.6.1 DP-Protokoll, was ist das?**

#### **Das DP-Protokoll**

Das DP-Protokoll wird im Bereich Dezentrale Peripherie (DP) eingesetzt und ermöglicht den dezentralen und prozessnahen Einsatz einer Vielzahl von Baugruppen und anderen Feldgeräten. Es basiert auf dem Kommunikationsstandard für den Feldbereich (IEC 61158) und ist in der PROFIBUS Norm (EN 50170) festgeschrieben.

Durch das DP-Protokoll über PROFIBUS können große Entfernungen zwischen den einzelnen Peripheriegeräten überbrückt werden. Dezentrale Peripheriestationen sammeln die Eingabesignale vor Ort und stellen sie zur Abholung bereit. Die zentrale Steuerung im Rechner kann diese dann zyklisch abholen. Zusätzlich sendet die zentrale Steuerung ihre Ausgabedaten zyklisch an die dezentralen Peripheriestationen.

Das DP-Protokoll ist für zeitkritische Anwendungen konzipiert. Durch ein optimiertes und einfaches Übertragungsprotokoll, hohe Übertragungsgeschwindigkeiten und die Verwendung des Master-Slave-Prinzips werden kurze Zykluszeiten erreicht.

#### **Welche Eigenschaften hat des DP-Protokoll?**

Die Eigenschaften sind:

- Zentrale Steuerung durch einen Master.
- Hoher Datendurchsatz durch ein einfaches Übertragungsprotokoll.
- Zyklische Übertragung des Prozessabbildes in Eingabe- und Ausgaberichtung.
- Erkennung von Fehlern durch Online-Diagnose.
- Parallelbetrieb mit anderen Geräten (Master und Slaves) an einem Bus möglich, da es auf PROFIBUS FDL (Schicht 2 des ISO/OSI-Referenzmodells) aufbaut.

#### **Welche Erweiterungen sind im DP-Protokoll definiert?**

Die folgenden Abschnitte zeigen eine Übersicht der verschiedenen DP-Master und deren Erweiterungen.

Für den DP-Master sind die Klassen 1 und 2 für den zyklischen Datenverkehr und Diagnosefunktionen definiert. Zusätzlich sind die Erweiterungen C1 und C2 für die azyklische Kommunikation realisiert.

## Was ist DPV1?

Der DPV1-Standard stellt eine Erweiterung der DP-Kommunikation dar. Slaves, die DPV1 unterstützen, besitzen einen zusätzlichen Speicherbereich, in welchem spezielle, slave-spezifische Datensätze abgelegt sind. DPV1 besteht aus zwei Teilen, zum einen aus der Erweiterung für DPC1 für zyklische Master und zum zweiten aus der Erweiterung für DPC2 für zusätzliche Diagnose- und Parametrierfunktionen. Mit Hilfe der DPV1-Funktionen können die Datensätze für erweiterte Funktionalität gelesen oder geschrieben werden.

## Was ist ein DP-Master Klasse 1?

Ein DP-Master Klasse 1 bietet Dienste zum Parametrieren der Slaves und für den zyklischen Datenverkehr.

## Was ist DPC1?

DPC1 ist eine DPV1-Erweiterung für einen DP-Master Klasse 1. Es ermöglicht dem C1-Master das azyklische Schreiben und Lesen der zusätzlichen Datenbereiche eines DPV1-Slaves.

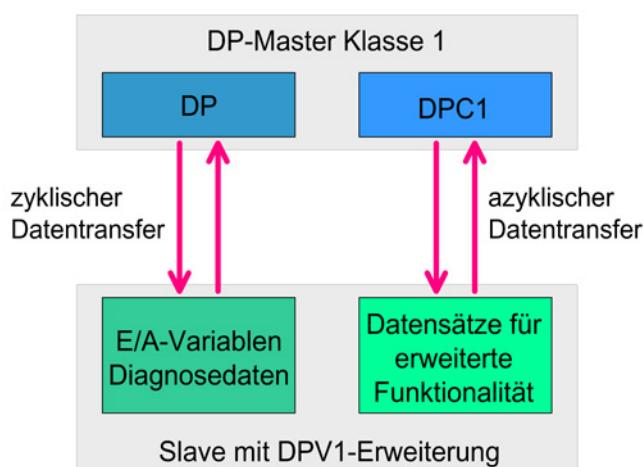


Bild 2-12 DP-Master Klasse 1 und DPC1

## Was ist ein DP-Master Klasse 2?

Ein DP-Master Klasse 2 bietet Diagnosemöglichkeiten und kann den Zustand eines DP-Masters Klasse 1 oder eines DP-Slaves abfragen, ohne den Betrieb eines laufenden Netzes zu beeinträchtigen.

## Was ist DPC2?

DPC2 ist eine DPV1-Erweiterung für einen DP-Master Klasse 2. Sie ermöglicht einem C2-Master das azyklische Schreiben und Lesen der zusätzlichen Datenbereiche eines DPV1-Slaves.

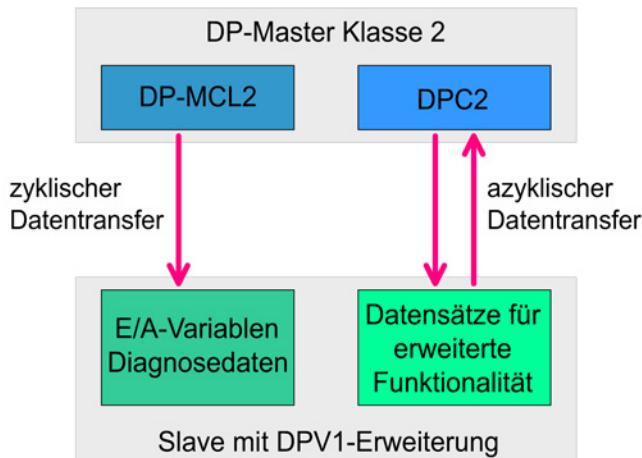


Bild 2-13 DP-Master Klasse 2 und DPC2

## 2.6.2 DP-Protokoll, wie sieht eine typische Anlagenkonfiguration aus?

Im folgenden Kapitel wird gezeigt, wie eine typische Anlagenkonfiguration bei PROFIBUS aussehen kann, in der die Datenkommunikation zwischen verschiedenen Geräten mittels DP-Protokoll realisiert ist.

### Beispiel einer Anlagenkonfiguration für das DP-Protokoll

Zur Kommunikation mit dem DP-Protokoll über PROFIBUS stehen im SIMATIC NET-Spektrum Kommunikationsbaugruppen für Steuerungen der Familien SIMATIC S5 und SIMATIC S7 sowie für PCs und Workstations zur Verfügung.

Die SIMATIC S7 bietet typischerweise die Kommunikationsbaugruppen CP 343-5 und für PCs sowie Workstations z. B. den CP 5623 oder CP 5622. Des weiteren gibt es DP-fähige Baugruppen in der ET 200-Serie.

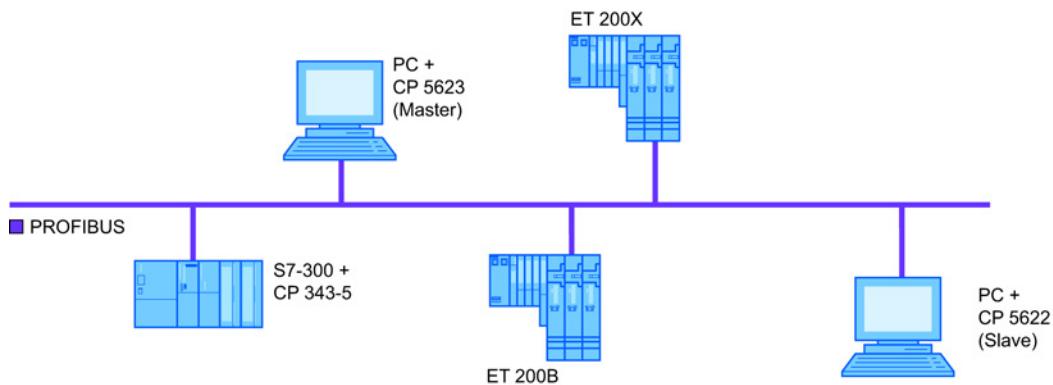


Bild 2-14 Typische Anlagenkonfiguration bei PROFIBUS

### 2.6.3 DP-Protokoll, wie funktioniert es?

#### So funktioniert das DP-Protokoll

Im Dezentralen Peripheriesystem gibt es drei Arten von Kommunikationspartnern:

DP-Kommunikationspartner	Beschreibung
DP-Slaves	Passive Busteilnehmer, üblicherweise die Peripheriegeräte. Aktive Busteilnehmer, z. B. PG-Kommunikations-CPs, die zusätzliche Aufgaben übernehmen.
DP-Master Klasse 1	Aktiver Busteilnehmer, zentrale Komponente zur Steuerung der DP-Slaves.
DP-Master Klasse 2	Aktiver Busteilnehmer, der parallel zum Master Klasse 1 zur Inbetriebnahme und Diagnose eingesetzt wird.

Die übliche Kommunikation zwischen einem DP-Master und den dezentralen Peripheriestationen erfolgt durch Polling. Polling bedeutet, dass der DP-Master zyklisch Aufruftelegramme an jeden ihm zugeordneten DP-Slave sendet.

Das Aufruftelegramm enthält die aktuellen Ausgabedaten, die der DP-Slave an seinen Ausgabeports anlegen soll. Den Empfang bestätigt der DP-Slave, indem er ein Quittungstelegramm zurückschickt. Das Quittungstelegramm enthält die Eingabedaten, die an den Eingangsports der DP-Slaves anliegen.

Besitzt ein DP-Slave keine Ausgabe- oder Eingabeports, so wird statt dessen ein "Leertelegramm" gesendet.

In einem Polling-Zyklus werden alle betriebsbereiten DP-Slaves adressiert. Der Adressierung des letzten Slaves schließt sich sofort ein weiterer Polling-Zyklus an. Durch dieses Verfahren wird die Aktualität der Daten gewährleistet. In jedem Polling-Zyklus versucht der DP-Master, nicht betriebsbereite Slaves in den Zyklus aufzunehmen.

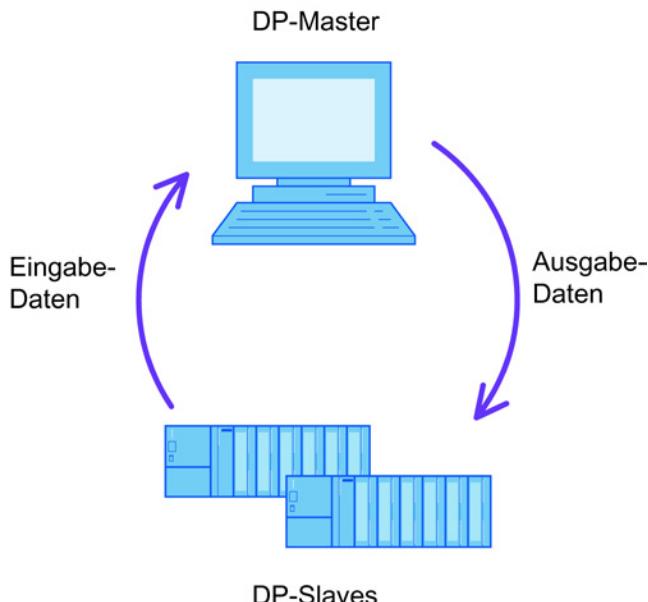


Bild 2-15 Kommunikation zwischen DP-Master und DP-Slave

Das DP-Protokoll ist auf schnellen Datendurchsatz zwischen Master und Slave optimiert und besitzt daher keine Flusskontrolle.

#### 2.6.4 DP-Protokoll, wie wird es projektiert?

##### So wird das DP-Protokoll projektiert

Für die Kommunikation mit dem DP-Protokoll ist ein DP-Gerät zu parametrieren und zu konfigurieren, bevor der Produktivbetrieb gestartet werden kann. Hierfür steht das Projektierungswerkzeug "SIMATIC STEP 7 Professional" zur Verfügung. Mit der Wahl des Protokolls werden außerdem für jedes DP-Gerät globale Betriebsparameter als Parametrierdaten und die Anzahl und die Art der Eingangs- /Ausgangs-Ports als Konfiguriertdaten zusammengestellt.

#### 2.6.5 DP-Protokoll, welches sind die Vorteile?

##### Da sind die Vorteile des DP-Protokolls

Das DP-Protokoll bietet folgende Vorteile:

- Die Kommunikation über PROFIBUS ist effizient und echtzeitfähig.
- Die Anwenderprogramme haben schnellen und unkomplizierten Zugriff auf die Prozessdaten.
- Das DP-Protokoll ist ein offenes, weit verbreitetes und international genormtes Protokoll.

## 2.6.6 DP-Master Klasse 1, welche Kommunikationsdienste stehen zur Verfügung?

Diese Betriebsarten gibt es beim DP-Master Klasse 1

Der DP-Master steuert den Zustand des DP-Systems. Jede Betriebsart des DP-Masters ist durch definierte Aktionen zwischen DP-Master und den DP-Slaves gekennzeichnet:

Betriebsart	Bedeutung
OFFLINE	Es findet keinerlei DP-Kommunikation zwischen DP-Master und den DP-Slaves statt. Dies ist der Grundzustand der DP-Master.
STOP	Auch in dieser Betriebsart findet keinerlei DP-Kommunikation zwischen DP-Master und den DP-Slaves statt. Im Gegensatz zur Betriebsart OFFLINE kann eine DP-Diagnosestation (DP-Master Klasse 2) Diagnoseinformationen des DP-Masters auslesen.
CLEAR	Der DP-Slave wird vom DP-Master mit Daten versorgt, die er zum Anlauf benötigt (Parametrierung und Konfigurierung). Anschließend wird in der Betriebsart CLEAR an alle Slaves mit Prozessausgabe der Wert 0h gesendet, d. h. die Prozessausgabe befindet sich im sicheren Zustand. Die Eingabedaten der Slaves sind bekannt und können ausgelesen werden.
OPERATE	Es findet der zyklische Datentransfer zwischen dem DP-Master und den DP-Slaves statt. Dies ist die Produktivphase. In dieser Betriebsart werden reihum die DP-Slaves vom DP-Master angesprochen.

Ausgehend von der aktuellen Betriebsart müssen die Betriebsarten in der vorgegebenen (aufsteigenden oder absteigenden) Reihenfolge OFFLINE-STOP-CLEAR-OPERATE durchlaufen werden.

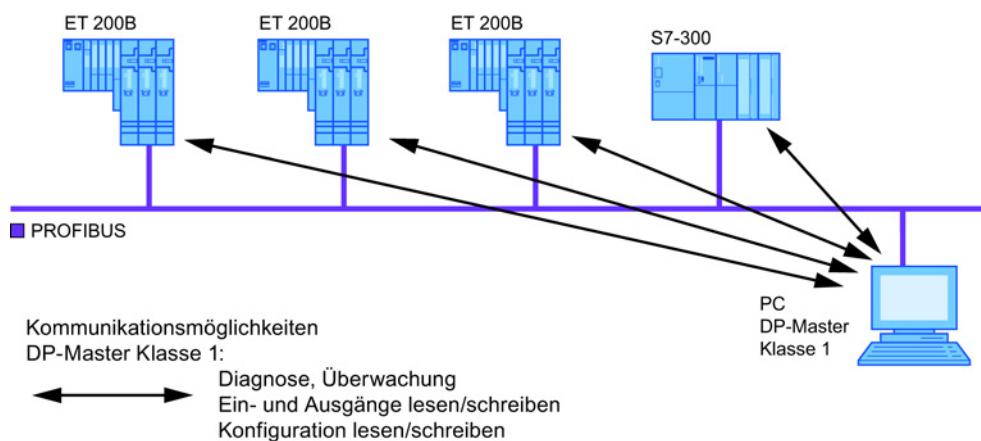


Bild 2-16 Kommunikationsdienste des DP-Master Klasse 1

### Diese Kommunikationsdienste bietet der DP-Master Klasse 1

Der Zugriff auf Prozessvariablen durch eine Applikation des DP-Masters Klasse 1 erfolgt nicht direkt, sondern über ein Prozessabbild auf der Kommunikationsbaugruppe.

Das Prozessabbild enthält für jeden DP-Slave drei Datenbereiche:

- Eingangsdaten vom DP-Slave
- Ausgangsdaten an den DP-Slave
- Diagnosedaten vom DP-Slave

Ein Anwenderprogramm auf einem PC kann folgenden Dienst über einen DP-Master Klasse 1 nutzen:

- Variabeldienste für das Prozessabbild des DP-Masters

Zusätzlich stehen unter anderem folgende Informationsdienste zur Verfügung:

- Betriebsart des DP-Masters und der DP-Slaves
- Ereignismeldungen vom DP-Master
- Aktivitätsüberwachung durch die DP-Baugruppe
- Typ eines DP-Slaves

### Welche Vor- und Nachteile bietet der DP-Master Klasse 1?

Die Verwendung eines DP-Masters Klasse 1 bietet folgende Vorteile:

- Schneller Zugriff auf zyklische Daten.
- Aufträge der Applikationen können sehr schnell bearbeitet werden, weil die Daten direkt aus dem Prozessabbild gewonnen werden und keine explizite Kommunikation bewirken.

Die Verwendung eines DP-Masters Klasse 1 hat folgenden Nachteil:

- Hohe Buslast durch den zyklischen Austausch der Ein- und Ausgabedaten.

## 2.6.7 DP-Master Klasse 2, welche Kommunikationsdienste stehen zur Verfügung?

### So funktioniert der DP-Master Klasse 2

Neben Geräten der DP-Master Klasse 1 können in einem DP-System auch Geräte der DP-Master Klasse 2 existieren. Diese werden für die Inbetriebnahme, zur Konfiguration oder zur Diagnose eingesetzt.

Es ist beispielsweise möglich, einen DP-Master Klasse 2 zu Diagnosezwecken an den PROFIBUS anzuschließen. Dieser kann den Zustand der Slaves und von MASTERN der Klasse 1 jederzeit abfragen, ohne dass der Betrieb eines laufenden Netzes beeinträchtigt wird. Außerdem hat ein DP-Master Klasse 2 die Möglichkeit, die Slave-Adresse zu verändern, sofern der Slave dies erlaubt.

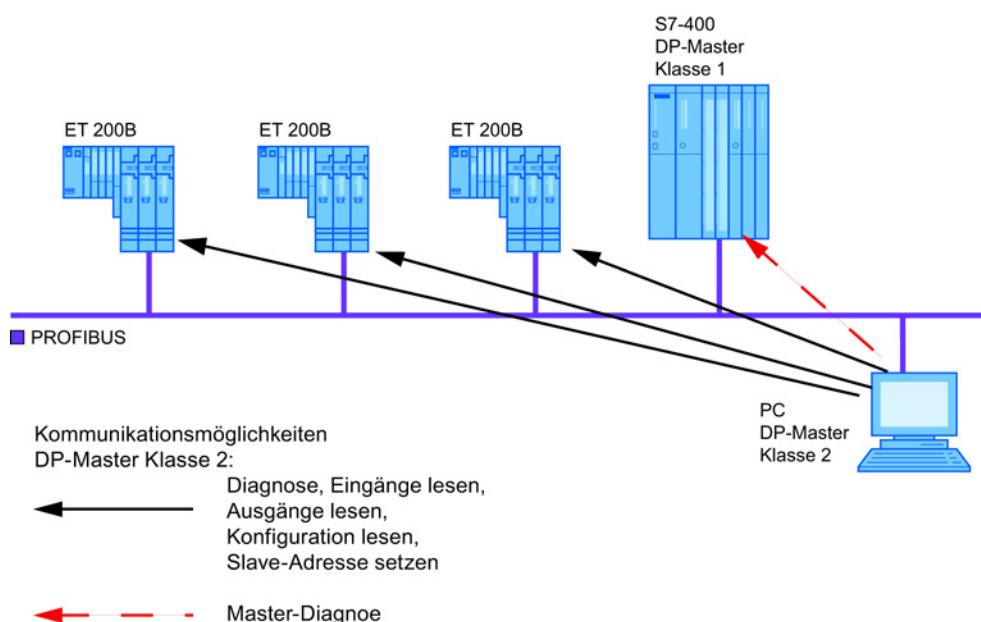


Bild 2-17 Kommunikationsdienste des DP-Master Klasse 2

### Diese Kommunikationsdienste bietet der DP-Master Klasse 2

Vergleichbar mit einem DP-Master Klasse 1 kann auch der DP-Master Klasse 2 auf die zyklischen Ein- und Ausgangsdaten sowie die Diagnosedaten zugreifen und die Informationen auf Variablen abbilden. Die Daten können jedoch nur gelesen, nicht geschrieben werden.

Die wichtigsten Master Klasse 2-Funktionen sind:

- Lesen der Daten vom Slave
- Lesen der Daten vom Master Klasse 1

### **Welche Vor- und Nachteile bietet der DP-Master Klasse 2?**

Die Verwendung eines DP-Master Klasse 2 bietet folgende Vorteile:

- Der Betrieb eines laufenden Netzes wird nur wenig beeinträchtigt.
- Slave-Adresse kann verändert werden.

Die Verwendung eines DP-Master Klasse 2 hat folgende Nachteile:

- Ein- und Ausgänge sowie Diagnosedaten eines Slaves können nur gelesen werden.
- Über OPC ist eine Synchronisation des Zugriffs auf Prozessvariablen mit dem DP-Zyklus nicht möglich.

### **2.6.8 DPC1, welche Kommunikationsdienste stehen zur Verfügung?**

#### **So funktioniert das Kommunikationsprinzip bei DPC1-Diensten**

Mit Hilfe der DPC1-Dienste ist es möglich, neben dem zyklischen Zugriff auf der DP-Master-Schnittstelle auch azyklisch Daten der Slaves abzufragen. Jeder DP-Slave mit DPV1-Erweiterung besitzt einen zusätzlichen Datenbereich, der vom DPC1-Master gelesen und beschrieben werden kann. Dieser Datenbereich ist slave-spezifisch und kann beispielsweise Parametrierdaten oder Alarmmeldungen enthalten. Die einzelnen Datensätze des zusätzlichen Datenbereichs werden durch die Angabe von Slot und Index adressiert.

Für die Nutzung der DPC1-Dienste sind keine Kommunikationsverbindungen zu den Slaves erforderlich, weil der Polling-Zyklus des Masters als implizite Verbindung dazu bereits initiiert ist. Sobald ein Slave mit DPC1-Funktionalität parametriert und konfiguriert worden ist, kann er mit DPC1-Diensten angesprochen werden.

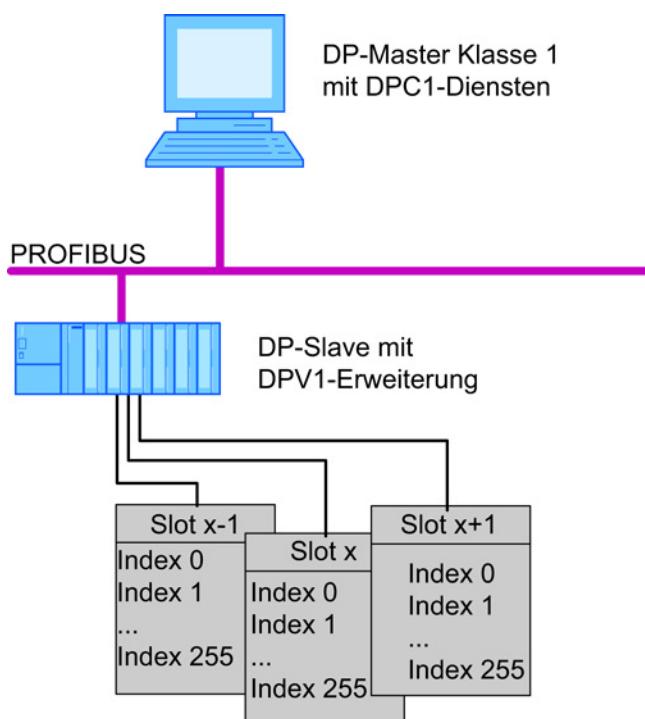


Bild 2-18 Kommunikationsprinzip bei DPC1-Diensten

### Diese DPC1-Kommunikationsdienste gibt es?

Die DPC1-Dienste beinhalten:

- Azyklisches Lesen und Schreiben von Datensätzen.
- Alarmbehandlung.

### Welche Vorteile bieten die DPC1-Dienste?

Die Verwendung der DPC1-Dienste bietet folgende Vorteile:

- Durch azyklischen Zugriff verringert sich die Buslast.
- Datenblöcke bis zu 240 Bytes können übertragen werden.
- Strukturierter Zugriff auf Datenblöcke ist möglich.
- Parallele Verwendung mit Variabldiensten des DP-Master Klasse 1 ist möglich.

## 2.6.9 DPC2, welche Kommunikationsdienste stehen zur Verfügung?

### So funktioniert das Kommunikationsprinzip bei DPC2-Diensten

Mit Hilfe der DPC2-Dienste ist es von einem DP-Master Klasse 2 möglich, neben dem zyklischen Zugriff auch auf azyklische Daten der Slaves zuzugreifen. DPC2-fähige Slaves besitzen einen zusätzlichen Datenbereich, der mit den DPC2-Diensten gelesen und geschrieben werden kann. Dieser Datenbereich ist slave-spezifisch und kann beispielsweise Parametrierdaten oder Alarmmeldungen enthalten. Die einzelnen Datensätze des zusätzlichen Datenbereichs werden durch die Angabe von Slot und Index adressiert.

Wesentlicher Unterschied zur üblichen Master-Slave-Kommunikation ist, dass zunächst eine Verbindung aufgebaut werden muss und anschließend solange aufrecht erhalten bleibt, bis sie entweder durch äußere Einflüsse unterbrochen oder vom Master abgebaut wird. Solange diese Verbindung aufgebaut ist, kann der Master mit dem Slave kommunizieren.

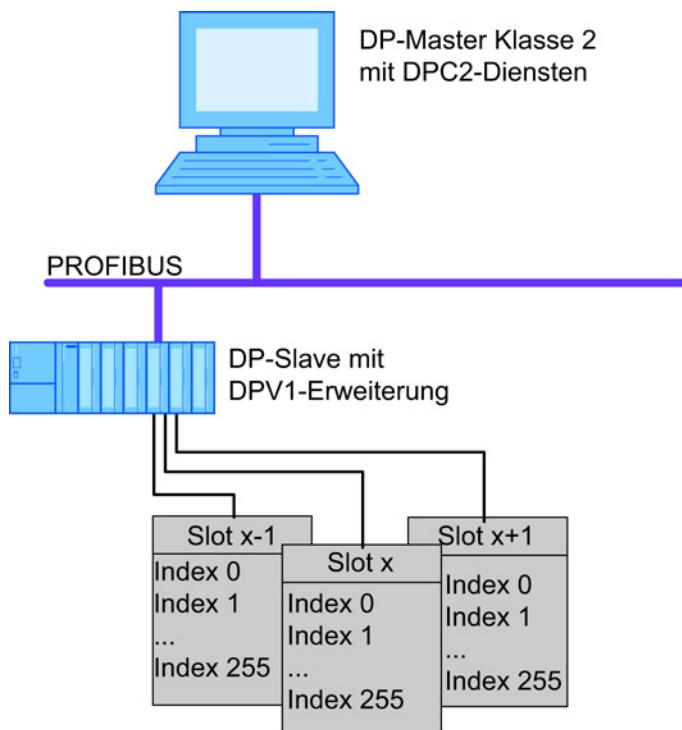


Bild 2-19 Kommunikationsprinzip bei DPC2-Diensten

### Diese DPC2-Kommunikationsdienste gibt es

Die wichtigsten DPC2-Dienste sind:

- Auf- und Abbau einer Kommunikationsbeziehung
- Lesen der Slave-Datensätze

Die Verwendung von DPC2-Diensten bietet dem Anwender sogenannte Blockdienste.

Da die DPC2-Dienste mit niedrigerer Priorität behandelt werden als die zyklischen Datendienste, ist der Datendurchsatz geringer. Zudem erhöht sich im allgemeinen die Tokenumlaufzeit, da das Netz zusätzlich belastet wird.

Analog dem Zugriff eines DP-Masters der Klasse 1 (DPC1) kann auch ein Master Klasse 2 nur blockweise auf die Daten eines DPV1-Slots zugreifen. Ein Leseauftrag liefert den gesamten Inhalt eines durch Slot und Index adressierten Datensatzes, ein Schreibauftrag überschreibt den vollständigen Datensatz.

### Welche Vorteile bieten die DPC2-Dienste?

Die Verwendung der DPC2-Dienste bietet folgende Vorteile:

- Der asynchrone Zugriff auf die Slaves ist möglich.
- Größere Datenblöcke können übertragen werden.
- Strukturierter Zugriff auf Datenblöcke ist möglich.
- Parallele Verwendung mit Master Klasse 1 ist möglich.

## 2.6.10 DP Slave, welche Kommunikationsdienste stehen zur Verfügung?

### Diese Kommunikationsdienste stellt ein DP-Slave zur Verfügung

Ein DP-Slave stellt Datenkommunikationsdienste zur Verfügung, die es einem DP-Master erlauben während eines Aufrufzyklus am PROFIBUS Eingabedaten abzuholen und vom DP-Master gesendeten Ausgangsdaten zu empfangen und weiterzuverarbeiten. Weiterhin kann der DP-Slave Diagnosedaten setzen, die ebenfalls vom DP-Master gelesen werden können.

DP-Slaves werden in der DP-Kommunikation als modular betrachtet. Jeder Slave kann aus mehreren Modulen zusammengesetzt sein, die jeweils eigene Ein- und Ausgabebereiche besitzen.

Ein für die DP-Erweiterung DPV1 tauglicher Slave kann für jedes Modul zusätzliche Datensätze enthalten. Diese Datensätze enthalten Slave-spezifische Daten, die von einem DPC1-Master gelesen und geschrieben werden können. Pro Datensatz stehen bis zu 240 Bytes Nutzdaten zur Verfügung.

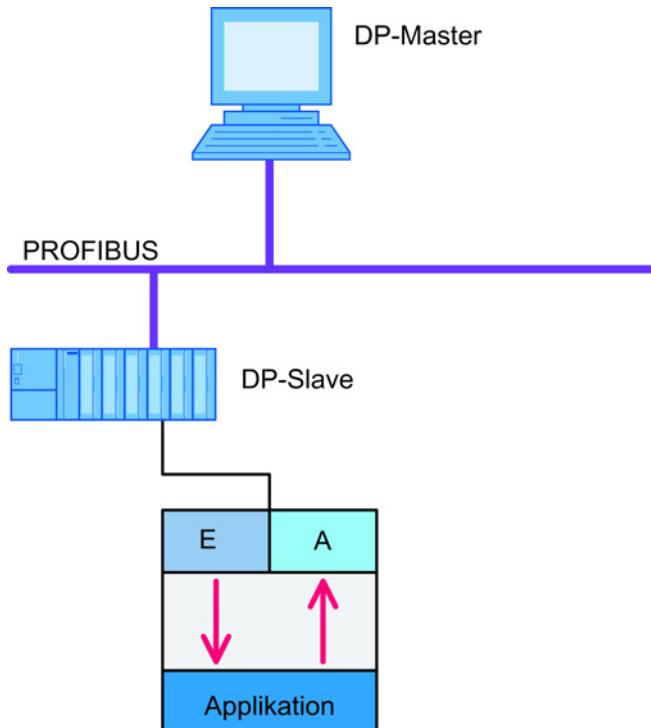


Bild 2-20 Kommunikationsprinzip bei DP-Slaves

### Welche Vorteile bietet die DP-Slave-Erweiterung DPV1?

Die DPV1-Erweiterung bietet folgende Vorteile:

- Azyklischer Zugriff ist möglich.
- Große Datenblöcke können übertragen werden.
- DP-Slave enthält zusätzliche Diagnose-Datensätze.

## 2.7 Das S7-Protokoll

### 2.7.1 S7-Protokoll, was ist das?

#### Das S7-Protokoll

Das S7-Protokoll dient der Kommunikation mit SIMATIC S7-Automatisierungssystemen. Es unterstützt sowohl die Kommunikation zwischen PG/PC und Automatisierungsgeräten, als auch den Datenaustausch zwischen Automatisierungsgeräten des SIMATIC S7-Systems.

### **Welche Eigenschaften hat das S7-Protokoll?**

Das S7-Protokoll zeichnet sich durch folgende Eigenschaften aus:

- Optimiert für die SIMATIC Kommunikation.
- Schnelligkeit im Vergleich zu anderen Automatisierungsprotokollen für die Datenkommunikation.
- Verfügbarkeit für Bussysteme der Leit- und Zellebene mit Industrial Ethernet und der Feldebene mit PROFIBUS.
- Einsetzbar auch bei hochverfügbaren Verbindungen.

### **Welche Unterschiede gibt es für das S7-Protokoll bei PROFIBUS und Ethernet?**

Bei PROFIBUS setzt das S7-Protokoll auf den FDL-Diensten auf, während es bei Ethernet die dort verfügbaren Dienste der Transportebene benutzt. Dieser Unterschied wird durch das S7-Protokoll verdeckt, sodass es keine Kommunikationsunterschiede für Anwendungsprogramme gibt.

### **Was hat das S7-Protokoll bei PROFIBUS und Ethernet gemeinsam?**

Auf beiden Kommunikationssystemen bietet das S7-Protokoll die Vorteile eines verbindungsorientierten Protokolls, wie zum Beispiel die Verbindungsüberwachung. Außerdem stehen alle im S7-Protokoll realisierten Kommunikationsdienste uneingeschränkt zur Verfügung.

## **2.7.2 S7-Protokoll, wie sieht eine typische Anlagenkonfiguration aus?**

Dieses Kapitel zeigt, wie typische Anlagenkonfigurationen bei PROFIBUS und Industrial Ethernet aussehen können, in denen die Datenkommunikation zwischen verschiedenen Geräten mittels S7-Protokoll realisiert ist.

### **Beispiel einer Anlagenkonfiguration für das S7-Protokoll bei PROFIBUS**

Zur Kommunikation mit dem S7-Protokoll über PROFIBUS stehen im SIMATIC NET-Spektrum Kommunikationsbaugruppen für Steuerungen der SIMATIC S7-Familie sowie für PCs und Workstations zur Verfügung.

Die SIMATIC S7 bietet hierfür typischerweise die Kommunikationsbaugruppen CP 342-5, CP 343-5 und CP 443-5 und für PCs sowie Workstations z. B. den CP 5623, CP 5624 oder CP 5622. Für das ET 200-System sind ebenfalls verschiedenen S7-Protokoll-fähige Baugruppentypen verfügbar.

## 2.7 Das S7-Protokoll

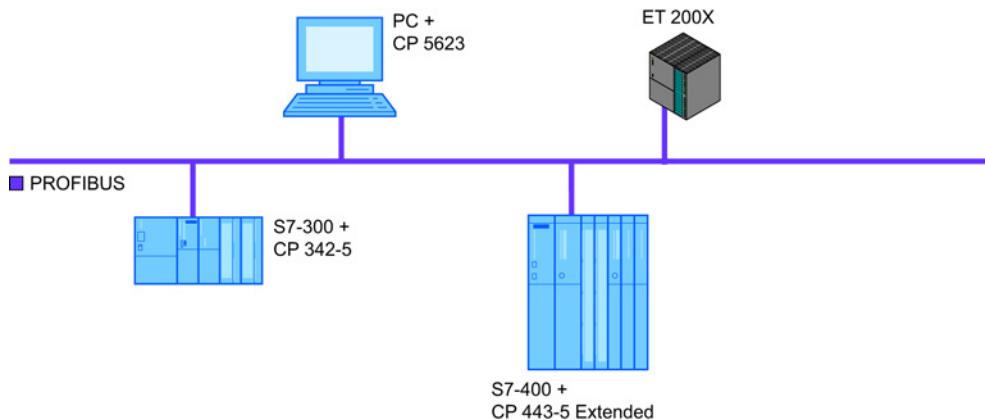


Bild 2-21 Typische Anlagenkonfiguration bei PROFIBUS

### Beispiel einer Anlagenkonfiguration für das S7-Protokoll bei Ethernet

Zur Kommunikation mit dem S7-Protokoll über Ethernet stehen im SIMATIC NET-Spektrum Kommunikationsbaugruppen für Steuerungen der SIMATIC S7-Familie sowie für PCs und Workstations zur Verfügung.

Die SIMATIC S7 bietet hierfür typischerweise die Kommunikationsbaugruppen CP 343-1 und CP 443-1, die PCs und Workstations den CP 1623 oder CP 1628.

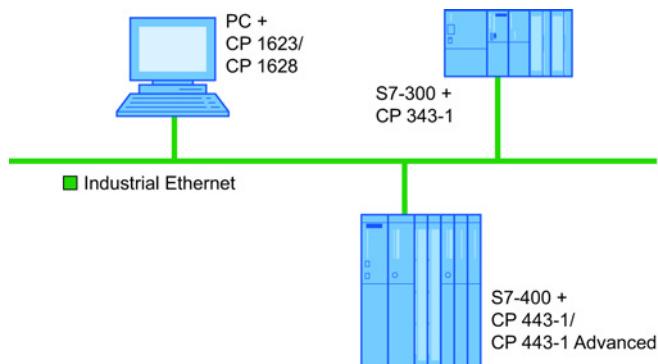


Bild 2-22 Typische Anlagenkonfiguration auf Ethernet

### 2.7.3 S7-Protokoll, wie funktioniert es?

#### So funktioniert das S7-Protokoll

Das S7-Protokoll bietet einfache und leistungsfähige Kommunikationsdienste. Die Datenübertragung erfolgt zwischen einer Automatisierungsanwendung auf einer SIMATIC PC-Station und einem anderen Automatisierungsgerät nach dem Client-Server-Modell. Dabei werden die vom Client angeforderten Daten vom Server bereitgestellt.

Zusätzlich können auch zwei Automatisierungsgeräte Daten austauschen. Diese Kommunikation funktioniert ebenfalls nach dem Client-Server-Modell.

Im Verlauf eines Verbindungsaufbaus stimmen die Kommunikationspartner automatisch wichtige Kenngrößen des Kommunikationsweges aufeinander ab. Es werden dabei die Einstellungen ausgehandelt, die von beiden Kommunikationspartnern noch geleistet werden können.

Im Verlauf dieser Aushandlung werden festgelegt:

- Größe der zu übertragenden Datenpakete
- Anzahl der gleichzeitig benutzbaren Sende- und Empfangsressourcen

## 2.7.4 S7-Protokoll, welche Kommunikationsdienste stehen zur Verfügung?

### Diese Kommunikationsdienste stellt das S7-Protokoll zur Verfügung

Das S7-Protokoll unterstützt folgende Kommunikationsdienste, welche uneingeschränkt bei PROFIBUS und Ethernet verfügbar sind.

Kommunikationsdienst	Beschreibung
Informationsdienste	Informationen zum Verbindungszustand. Anzeige des Geräte- und Anwenderstatus vom Kommunikationspartner.
Variabldienste	Funktionen zum Lesen und Schreiben einer oder mehrerer Variablen.
Blockdienste	Programmgesteuerte Übertragung großer Datenblöcke.
Bausteindienste	Diese Dienste ermöglichen das Laden, Hochladen, Löschen und Einketten von Bausteinen in den Programmablauf eines Automatisierungsgerätes während des laufenden Betriebs. Dadurch wird eine dynamische Änderung von Programmabläufen und -parametern erreicht.
Ereignisdienste	Mit diesen Diensten können Meldungen vom SIMATIC S7-Automatisierungssystem empfangen und weiterverarbeitet werden, z.B. Alarne.
Sicherheitsdienste	Zugriffskontrolle durch das Setzen von Passwörtern auf SIMATIC S7-Datenobjekte.
Serverdienste	Die PC-Station wird zum S7-Server und stellt einen Datenbaustein zur Verfügung, der sowohl lokal als auch remote gelesen und geschrieben werden kann (PUT / GET). Die Nummer des Datenbausteins kann von lokal und von remote abgefragt und angezeigt werden.

### Was können die S7-Informationsdienste?

Das S7-Protokoll bietet Informationsdienste, über die

- Attribute eines Partnergeräts abgefragt werden können.
- der Status eines Partnergeräts gelesen werden kann.

### **Was können die S7-Variablen-Dienste?**

Über das S7-Protokoll kann in einfacher Weise auf S7-Variablen zugegriffen werden.

Auf den meisten S7-Geräten sind folgende S7-Variablen verfügbar:

- Datenbausteine
- Instanzdatenbausteine
- Ein- /Ausgänge
- Peripherie Ein- /Ausgänge
- Merker
- Timer
- Zähler

### **Welche Vor- und Nachteile haben die S7-Variablen-Dienste?**

Die Verwendung der Variablen-Dienste bietet folgende Vorteile:

- Sehr einfacher Zugriff auf das Partnergerät ohne Programmierung des Partners.
- Das Lesen und Schreiben mehrerer Variablen sowie von langen Feldern von Variablen erfolgt optimiert.
- Bei Verwendung des OPC-Servers können Zugriffsrechte zum Schutz sicherheitsrelevanter Variablen vergeben werden.
- Mit OPC ist die Verwendung von Symbolen aus STEP 7 möglich.
- Bei Verwendung von OPC ist die Größe von Variablen und die Größe eines Datenblockes auf max. 64 Kbytes beschränkt.

Die Verwendung der Variablen-Dienste hat folgende Nachteile:

- Zur Beobachtung von Variablenänderungen muss zyklisch auf das Partnergerät zugegriffen werden.
- Zugriffe in kurzen Abständen erzeugen eine hohe Netzlast.

### **Was können die S7-Blockdienste?**

Die S7-Blockdienste ermöglichen eine programmgesteuerte Übertragung größerer Datenblöcke. Die Datenmenge beträgt bei dem Datentransfer bis zu 65534 Byte.

Für den Datenaustausch ist eine Verbindungsprojektierung notwendig. Diese funktioniert sowohl zwischen PC und Automatisierungsgerät als auch zwischen Automatisierungsgeräten.

## Welche Vor- und Nachteile haben die S7-Blockdienste?

Die Verwendung der S7-Blockdienste bietet folgende Vorteile:

- Auch große (max. 65534 Byte) Datenblöcke können übertragen werden.
- Ein SIMATIC PC kann sowohl Client wie auch Server sein, d.h. mit Blockdiensten können auch Daten von PC zu PC über das S7 Protokoll übertragen werden.
- Eine Strukturierung der Datenblöcke in OPC-Items ist möglich.
- Alle OPC-Variablen, die innerhalb eines Empfangspuffers definiert sind, bekommen eine Änderungsmeldung, wenn ein Datenblock eintrifft und sich die entsprechenden Daten geändert haben.
- Es tritt keine Netzbelastung durch Polling auf, wenn keine Daten gesendet werden.

Die Verwendung der blockorientierten Dienste hat folgende Nachteile:

- Eine Programmierung von Sende- und Empfangs-Bausteinen auf dem Automatisierungsgerät bzw. auf dem PC ist notwendig.
- Der Empfänger kann keine Daten anfordern, sondern muss warten, bis die Daten gesendet werden.
- Blockdienste stehen nicht für alle S7-Automatisierungsgeräte zur Verfügung.

## Was können die S7-Bausteindienste?

Der S7-Bausteindienst über das S7-Protokoll bietet folgende Anwendungen:

- Laden von Daten aus dem PG/PC in die SIMATIC-CPU.
- Hochladen von Daten aus der SIMATIC-CPU in das PG/PC.
- Einketten von Bausteinen in den Programmablauf der SIMATIC-CPU.
- Löschen von Bausteinen.
- Komprimieren des Speichers auf dem Automatisierungsgerät.

Ein Baustein repräsentiert einen ladbaren Bereich in einem Automatisierungssystem. Die Bausteindienste können mit Organisationsbausteinen (OB), Funktionsbausteinen (FB), Funktionen (FC), Datenbausteinen (DB) und Systemdatenbausteine (SDB) durchgeführt werden.

Ein Baustein kann beispielsweise über eine S7-OPC-Applikation von einer S7-CPU in einen PC geladen werden oder umgekehrt. Auf dem PC werden die Bausteine in Dateien gespeichert.

Der Bausteinname ist innerhalb der S7-CPU eindeutig. Die maximale Datenmenge ist CPU-spezifisch begrenzt. Die Bausteine werden deshalb in einzelne Segmente unterteilt, die sequentiell übertragen werden.

Ein auf ein Automatisierungsgerät übertragener Baustein wird in einem Zwischenspeicher abgelegt. Damit ist der Baustein für ein S7-Programm noch nicht verfügbar. In der Liste der Datenbausteine, die über die Online-Funktionen von STEP 7 betrachtet werden können, ist der Baustein zwar sichtbar, jedoch kann er nicht geöffnet werden. Dies ist erst möglich, nachdem der Baustein in die Liste der aktiven Bausteine eingekettet wurde.

### Anwendungsbeispiel für S7-Bausteindienste

Mit STEP 7 programmierte bzw. erstellte Bausteine werden bei der Inbetriebnahme mittels eines Programmiergeräts auf ein Automatisierungssystem übertragen. Diese Bausteine, im Beispiel DB\_rotes\_Auto.dbf, DB\_grünes\_Auto.dbf,... werden als DB1, DB2, ... im Programmspeicher abgelegt. Eine S7-OPC-Applikation kann im laufenden Betrieb diese Bausteine hochladen und lokal als Datei DB\_rotes\_Auto.dbf, DB\_grünes\_Auto.dbf,... abspeichern. Eine PC Steuerung kann damit Bausteine laden bzw. löschen und auf einen Programmablauf dynamisch Einfluss nehmen, um beispielsweise einen DB1, entsprechend DB\_rotes\_Auto.dbf, durch einen Datenbaustein DB\_blaues\_Auto.dbf als DB1 auszutauschen.

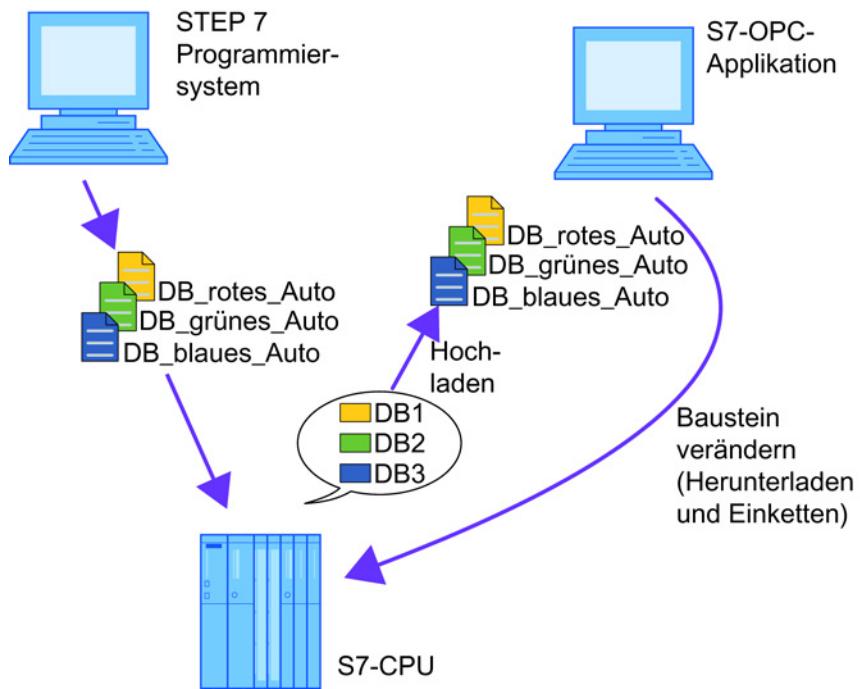


Bild 2-23 Beispiel für S7-Bausteindienste

### Welche Vor- und Nachteile haben die S7-Bausteindienste?

Die Verwendung der S7-Bausteindienste bietet folgende Vorteile:

- Zugriff (Lesen/Schreiben/Löschen) auf ladbaren Bereich eines Automatisierungssystems.

Die Verwendung der S7-Bausteindienste hat folgende Nachteile:

- Die Datenmenge ist CPU-spezifisch begrenzt und die Daten müssen deshalb sequentiell übertragen werden.

### Was kann der S7-Meldungsdienst?

Der S7-Meldungsdienst bietet die Möglichkeit, Meldungen eines Automatisierungsgerätes zu empfangen. So können zum Beispiel Störungen gemeldet werden.

Zur detaillierten Anzeige von Meldungen im PC können diese mit bis zu 10 Begleitwerten gesendet werden. Meldungen außergewöhnliche Ereignisse, die vom Automatisierungsgerät ausgelöst werden. Sie werden gepuffert und können nicht verloren gehen.

### Welche Vor- und Nachteile hat der S7-Meldungsdienst?

Die Verwendung der S7-Meldungsdienste bietet folgende Vorteile:

- Meldungen werden gepuffert und können nicht verloren gehen.
- Mit einer Meldung können bis zu 10 Begleitwerte übermittelt werden.

Die Verwendung der S7-Meldungsdienste hat folgende Nachteile:

- Zur Meldungserzeugung muss ein Programm in der Steuerung erstellt werden.
- Für die Begleitwerte werden nur eine begrenzte Anzahl Datentypen unterstützt.

### Was kann der S7-Sicherheitsdienst?

Der S7-Sicherheitsdienst regelt den Zugriff auf S7-Verbindungen. So kann für eine Verbindung das Passwort zur Legitimation und damit zur Aufhebung einer Schutzstufe übermittelt werden.

Für ein S7-Automatisierungssystem können 3 Schutzstufen für die Bausteindienste mit Hilfe des Projektierungswerkzeugs STEP 7 aktiviert werden:

- Schutz durch die Stellung des Schlüsselschalters
- Schreibschutz
- Schreib- und Leseschutz

Durch die Übermittlung des richtigen Passwortes werden alle obigen Schutzstufen für die aktuelle Verbindung aufgehoben.

### Welche Vorteile hat der S7-Sicherheitsdienst?

Die Verwendung der S7-Sicherheitsdienste bietet folgende Vorteile:

- Zugriffskontrolle für Verbindungen
- Zugriffskontrolle durch Schlüsselschalter kann aufgehoben werden

### Was können die S7-Serverdienste der PC-Station?

Die PC-Station wird zum S7-Server:

- ein Datenbaustein DB1 der Größe 65535 Byte steht zur Verfügung.
- über die S7-Dienste PUT und GET kann der Partner (z.B. S7-Station, PC-Station) die Werte des Datenbausteins lesen oder beschreiben.
- über eine S7-Verbindung "@LOCALSERVER" kann ein Client auf der PC-Station die Werte des Datenbausteins lesen oder beschreiben.
- Datenkonsistenz wird auch bei Parallelzugriffen gewährleistet.

- auch nach Neuanlauf der PC-Station bleiben die Werte im Datenbaustein erhalten (Datenpermanenz).
- die Nummer des Datenbausteins kann von lokal und von remote abgefragt und angezeigt werden.

### **Anwendungsbeispiel für S7-Serverdienste**

Ein S7-Client könnte z.B. eine S7-200 Station sein, welche Statusdaten an die PC-Station melden möchte, ohne dass diese die Statusdaten ständig abpollt. Er schreibt dann (selten) Statuswerte in den Datenbaustein. Ein lokaler Client auf der PC-Station kann über Datenänderungen der Statuswerte informiert werden. Ein ständiges Pollen der Statuswerte auf der S7-Station wird dadurch vermieden.

### **Welche Vor- und Nachteile haben die S7-Serverdienste?**

Die Verwendung der S7-Serverdienste bietet folgende Vorteile:

- Entlastung der S7-Station von zyklischen Zugriffen bei Beobachtung von Variablenänderungen.
- Datenkonsistenz und -permanenz.

Die Verwendung der S7-Serverdienste hat folgende Nachteile:

- Es wird nur ein Datenbaustein zur Verfügung gestellt, keine Merker, Ein-, Ausgänge, Zähler oder Timer.
- Für die Kommunikation ist eine S7-Applikation erforderlich, die die Daten bei Bedarf überträgt.

Es muss ein lokaler Client auf der PC-Station aktiv sein, um die S7-Serverdienste zu aktivieren.

### **2.7.5 Hochverfügbare S7-Verbindungen, was ist das?**

Hochverfügbare S7-Verbindungen sind spezielle projektierte S7-Verbindungen für den Anschluss einer PC-Station an ein hochverfügbares Automatisierungssystem S7-400H über Industrial Ethernet oder für die Kopplung solcher Automatisierungssysteme (siehe folgende Grafik). Eine Kopplung von PC-Stationen untereinander ist über hochverfügbare S7-Verbindungen nicht möglich.

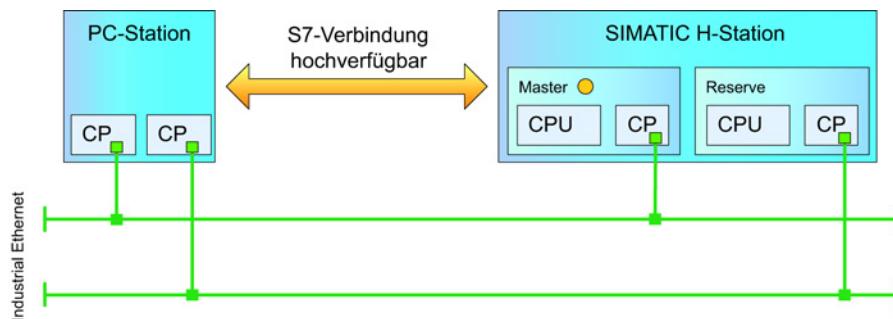


Bild 2-24 Kopplung einer PC-Station mit einer SIMATIC H-Station

In diesem Kapitel erfahren Sie, wie Sie hochverfügbare S7-Verbindungen einsetzen, projektiert, in Betrieb nehmen und diagnostizieren.

### 2.7.5.1 Hochverfügbare S7-Verbindungen, ein Überblick

Die Kommunikation einer PC-Station mit einem Automatisierungssystem S7-400H erfolgt über redundant ausgelegte Kommunikationswege. Während eine Standard S7-Verbindung über einen nichtredundanten Verbindungsweg aufgebaut wird, ermöglicht eine hochverfügbare S7-Verbindung die hochverfügbare Kommunikation über diese redundanten Kommunikationswege.

Eine hochverfügbare S7-Verbindung verhält sich aus Sicht der Anwendung wie eine Standard-S7-Verbindung. Es können also alle weiter oben beschriebenen Dienste des S7-Protokolls genutzt werden. Ebenso können bestehende Applikationen ohne Änderungen verwendet werden.

Gegenüber einer Standard-S7-Verbindung kann eine hochverfügbare S7-Verbindung jedoch gleichzeitig zwei Verbindungswege nutzen (von optional bis zu vier Verbindungs wegen), so dass der Ausfall eines Verbindungsweges keinen Verbindungsabbruch zur Folge hat. Überwachungs- und Synchronisationsmechanismen sorgen dafür, dass bei Ausfall des aktiven Verbindungswegs dieser Ausfall erkannt wird und automatisch der passive (redundante) Verbindungsweg die Kommunikation übernimmt. Die Verbindung selbst bleibt bestehen. Die Umschaltung ist für die Applikation transparent, kann jedoch über eine Diagnoseschnittstelle erkannt werden (siehe auch Kapitel "Diagnose, Inbetriebnahme, Wartung, Betrieb (Seite 63)")

#### Kann ich hochverfügbare S7-Verbindungen zwischen zwei PC-Stationen nutzen?

Nein, anders als Standard-S7-Verbindungen können hochverfügbare S7-Verbindungen ausschließlich zur Kopplung von PC-Stationen mit SIMATIC H-Stationen genutzt werden.

#### Welche Varianten hochverfügbarer S7-Verbindungen gibt es?

Die Redundanz der hochverfügbaren S7-Verbindung ist skalierbar und kann durch Erhöhung der Anzahl der CPs und der eingesetzten Netze erhöht werden.

Es sind folgende hochverfügbare S7-Verbindungen projektierbar:

- Hochverfügbare Verbindungen über 2 Wege
- Hochverfügbare Verbindungen über 4 Wege (erhöhte Redundanz)

### **Welche Vor- und Nachteile haben hochverfügbare S7-Verbindungen gegenüber Standard-S7-Verbindungen?**

Der Vorteil der hochverfügbaren S7-Verbindungen ist, dass der Ausfall von einzelnen Komponenten kompensiert wird.

Grundsätzlich ist eine Erhöhung der Verfügbarkeit von Systemen durch Redundanz mit zusätzlichen Kosten und zusätzlich benötigten Ressourcen verbunden. Das gilt auch für die hochverfügbaren S7-Verbindungen, die Komponenten des S7-400 Automatisierungssystems und beteiligte Netzwerkkomponenten.

### **Welche Transportprotokolle / Medien können für hochverfügbare S7-Verbindungen genutzt werden?**

Hochverfügbare S7-Verbindungen können ausschließlich auf Industrial Ethernet-Netzen betrieben werden. Folgende Transportprotokolle können genutzt werden:

- ISO (nur für den Betrieb über HARDNET-Baugruppen)
- ISO-on-TCP (nach RFC 1006)

### **Kann ich hochverfügbare S7-Verbindungen auch in virtuellen Umgebungen nutzen?**

Ja, Sie können hochverfügbare S7-Verbindungen in virtuellen Maschinen auf der Plattform vSphere von VMware nutzen. Lesen Sie dazu unbedingt die Voraussetzungen, Hinweise und Einschränkungen, die im Kapitel "Installation und Konfiguration unter VMware vSphere" in der Installationsanleitung der "SIMATIC NET PC Software" beschrieben sind.

Beachten Sie, dass der virtuellen Maschine jederzeit ausreichend Ressourcen, insbesondere Rechenleistung, zur Verfügung stehen (besonders bei Nutzung des ISO-on-TCP-Protokolls).

### **Welche Baugruppen können für hochverfügbare S7-Verbindungen verwendet werden?**

In einer PC-Station können alle Industrial Ethernet -Baugruppen wie folgt verwendet werden:

- Der CP 1613 A2 für das Transportprotokoll ISO (nur Windows 7 und Windows Server 2008 R2)
- Alle HARDNET-IE-CPs ab CP 1623 für ISO und ISO-on-TCP (TCP ab SIMATIC NET PC Software V8.1.2)
- Ethernet-Adapter, die als "IE-Allgemein" projektiert werden (ab SIMATIC NET PC Software V8.2, nur ISO-on-TCP, maximal 2 CPs in einer PC-Station, keine erhöhte Redundanz möglich)
- CP 1612 A2 (nur Windows 7, Server 2008 R2, ab SIMATIC NET PC Software V8.2, nur ISO-on-TCP, maximal 2 CPs in einer PC-Station, keine erhöhte Redundanz möglich)

Zur nutzbaren Anzahl von CPs des jeweiligen Typs und auch zur Anzahl der nutzbaren Verbindungen beachten Sie bitte die entsprechenden Mengengerüstangaben. Die Mengengerüstangaben finden Sie auf den Support-Seiten unter der Beitrags-ID: 15227599 (<http://support.automation.siemens.com/WW/view/de/15227599>)

Es wird empfohlen, für eine hochverfügbare S7-Verbindung jeweils nur CPs gleichen Typs zu verwenden.

Zur Nutzung von Anschaltungen in einem S7-H-System beachten Sie bitte auch die entsprechende Dokumentation zu diesen Systemen.

### Welche Eigenschaften hat eine hochverfügbare S7-Verbindung über 2 Wege?

Diese hochverfügbare S7-Verbindung verfügt über 2 Kommunikationswege. Der Ausfall einer Komponente führt zum automatischen Umschalten auf den anderen, redundanten Kommunikationsweg.

Die Abbildung zeigt in einem Zeitdiagramm den Ablauf bei Auftreten einer Störung mitten in einer laufenden Datenübertragung.

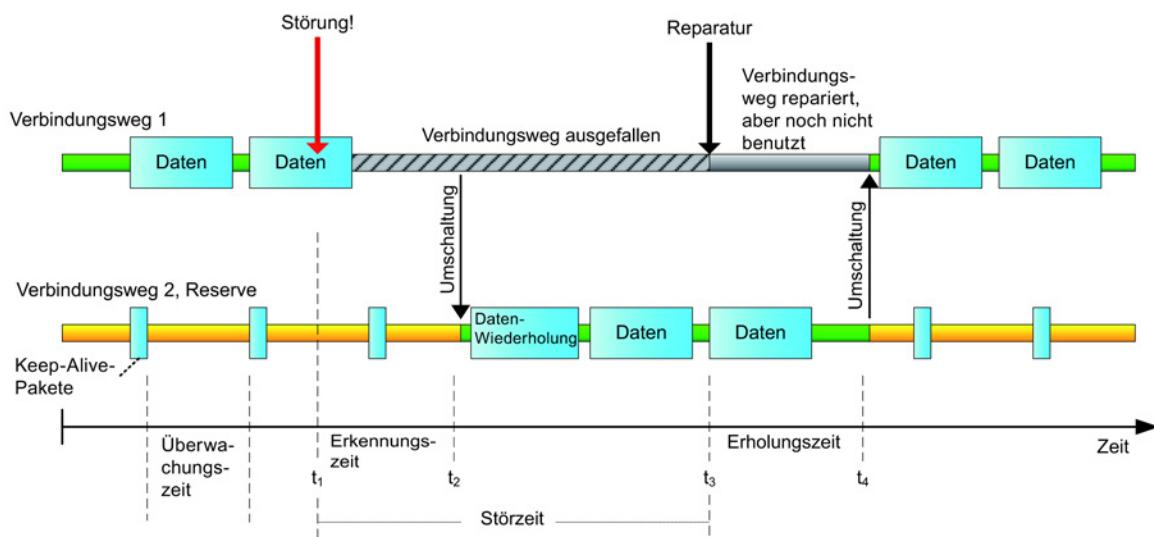


Bild 2-25 Umschaltung auf Reserveweg

Die hochverfügbare S7-Verbindung hat zwei Verbindungswege zur Verfügung, den Verbindungsweg 1 und als Reserve den Verbindungsweg 2. Zunächst wird der Weg 1 für den Transport von Nutzdaten benutzt und die Reserve durch die zyklische Übertragung von Keep-Alive-Paketen aufrechterhalten (Zyklus ist kleiner als die Überwachungszeit). Zum Zeitpunkt  $t_1$  tritt eine Störung (rot) auf.

Nach zweimaligem Verstreichen der Überwachungszeit (in der Abbildung als Erkennungszeit bezeichnet) ist die Störung erkannt und der Verbindungsweg wird als nicht nutzbar gekennzeichnet. Sofort wird umgeschaltet und die Reserveverbindung für die Übertragung der Nutzdaten verwendet. Ein nicht benutzter Verbindungsweg ist grau dargestellt. Wenn der Verbindungsweg wegen einer Störung nicht nutzbar ist, dann ist er zusätzlich schraffiert dargestellt.

Da ein Überwachungsmechanismus die Verbindungswege zyklisch prüft, kann nach erfolgter Reparatur frühestens nach Ablauf eines solchen Zyklus wieder auf Weg 1 zurückgeschaltet werden. Danach läuft wieder der Normalbetrieb.

Während der ganzen Zeit ist die S7-Verbindung aufgebaut. Erst, wenn vor der Reparatur von Weg 1 und erfolgter Umschaltung nach Reparatur (also zwischen den Zeitpunkten  $t_1$  und  $t_4$ ) auch der Reserveweg ausfällt, bricht die S7-Verbindung ab und muss nach Reparatur neu aufgebaut werden.

Die Zeit, innerhalb der eine Leitungsunterbrechung erkannt wird, liegt bei aktivem Datenverkehr auf diesem Verbindungsweg bei Betrieb über das ISO-Protokoll unter einer Sekunde. Die Umschaltzeiten für ISO-on-TCP hängen von der projektierten Überwachungszeit ab (siehe Projektierung).

### Welche Anlagenkonfigurationen mit einer hochverfügbaren S7-Verbindung über 2 Wege sind möglich?

Eine 2-Wege-Kommunikation kann beispielsweise mit folgenden Komponenten aufgebaut werden (siehe auch folgende Abbildung):

- SIMATIC H-Station, 2 Racks mit jeweils einem CP
- 2 Netzwerke
- PC-Station mit 2 CPs, z. B. CP 1623

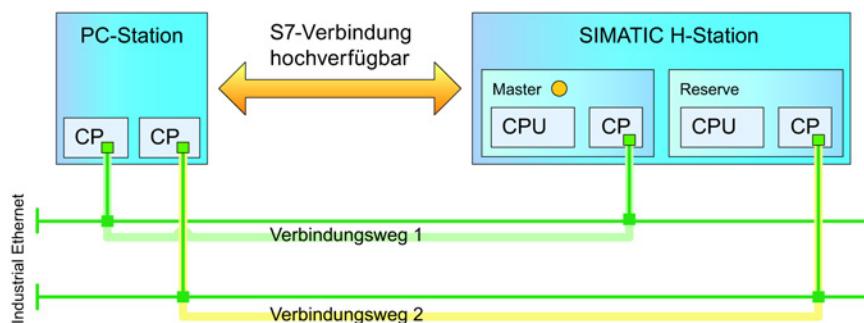


Bild 2-26 Beispiel für eine 2-Wege-Redundanz über 2 Netze

#### Hinweis

Falls Sie hochverfügbare S7-Verbindungen zu einer CPU 417-5H der H-Station über ISO-on-TCP nutzen möchten, können Sie dies auch direkt über die Netzwerkschnittstelle der CPU tun. Wählen Sie in diesem Fall als Verbindungspartner die CPU-Schnittstelle (siehe auch Kapitel "Projektierung (Seite 61)"). Für Verbindungen über ISO-Protokoll ist dies nicht möglich. Die Abbildung "Beispiel für eine 2-Wege-Redundanz über 2 Netze" zeigt die Kommunikation über die CPs der H-Station.

#### Hinweis

Es gibt auch die Möglichkeit die hochverfügbare S7-Verbindung über nur ein Netzwerk und einen CP in der PC-Station zu nutzen. Dann werden beide Verbindungswege über dasselbe Netz geführt. Eine solche Konfiguration kann sinnvoll sein, wenn man an anderer Stelle mit geeigneten Maßnahmen für eine erhöhte Verfügbarkeit der Netz-Infrastruktur sorgt. Empfohlen wird jedoch die oben gezeigte Konfiguration.

## Welche Eigenschaften hat eine hochverfügbare S7-Verbindung über 4 Wege?

Eine hochverfügbare S7-Verbindung über 4 Wege kann gegenüber einer 2-Wege-Verbindung bei einer Störung bis zu zwei zusätzliche Verbindungswege nutzen.

Wenn Sie Ihre hochverfügbare S7-Verbindung mit maximaler CP-Redundanz (4 Wege) projektiert haben, dann wird nach Ausfall des Produktiv- oder Reservepfads ein anderer Verbindungs weg aufgebaut (sofern verfügbar). Der Umschaltvorgang kann je nach Konfiguration etwas dauern. Die Verbindung ist dann wieder im Zustand "redundant" (über einen neuen Weg).

### Hinweis

Prüfen Sie bei der Inbetriebnahme unbedingt, ob die zusätzlichen Verbindungswege auch tatsächlich verfügbar sind, indem Sie z. B. auf Weg 1 und Weg 2 nacheinander Störungen verursachen und die Umschaltung auf Weg 3 und Weg 4 beobachten (siehe auch Kapitel "Diagnose, Inbetriebnahme, Wartung, Betrieb (Seite 63)").

Die folgende Abbildung stellt beispielhaft das Verhalten beim Auftreten einer Störung auf Verbindungs weg 1 dar:

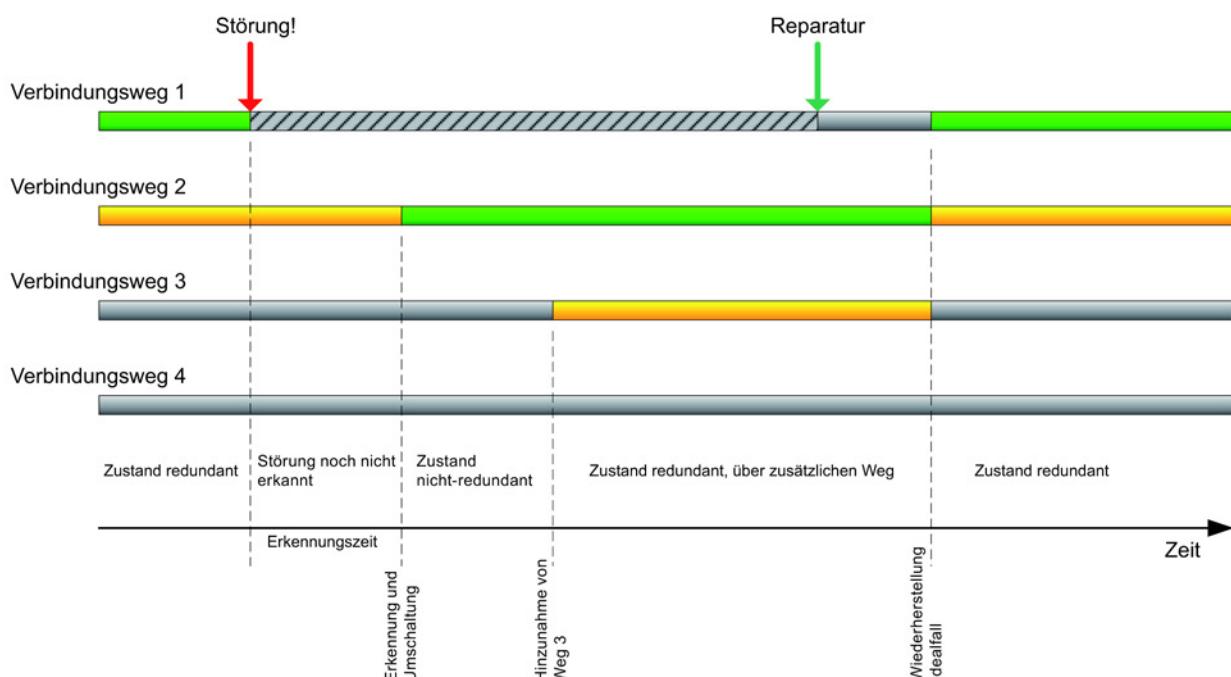


Bild 2-27 Umschaltung auf Reserve und Weg 3 einer hochverfügbaren S7-Verbindung mit erhöhter Redundanz

Gezeigt sind alle vier Verbindungswege, von denen in diesem Beispiel nur die ersten drei benutzt werden. Nach Auftreten der Störung und der Erkennungszeit wird Weg 2 produktiv genutzt (grün). Nachdem erkannt wird, dass der Weg 1 nicht mehr nutzbar ist, wird der bisher nicht genutzte Weg 3 hinzugenommen und dient dann als Reserveweg (gelb). Nicht benutzte Wege sind grau dargestellt (schraffiert, falls der Weg wegen einer Störung nicht nutzbar ist).

Nach erfolgter Reparatur wird erkannt, dass der Weg 1 wieder nutzbar ist und es wird wieder in den Idealfall zurückgeschaltet.

### Hinweis

Nach Störung und erfolgreicher Umschaltung auf zusätzliche Verbindungswege ist die Verbindung wieder im Zustand „Redundant“, da jetzt wieder ein Reserve-Verbindungs weg vorhanden ist. Trotzdem ist dieser Zustand nicht ideal, weil nun je nach Konfiguration nicht mehr garantiert werden kann, dass nach dem Ausfall einer weiteren Komponente die Verbindung weiter aufrechterhalten werden kann.

## Welche Anlagenkonfigurationen mit einer hochverfügbaren S7-Verbindung über 4 Wege sind möglich?

Für 4-Wege-Kommunikation können ausschließlich HARDNET-Baugruppen verwendet werden. Die Verwendung der CPU-Schnittstelle ist ebenfalls nicht möglich.

### 4-Wege-Kommunikation über 2 Netze

Eine hochverfügbare Kommunikation über 4 Wege mit erhöhter Redundanz kann beispielsweise mit folgenden Komponenten aufgebaut werden:

- SIMATIC H-Station, 2 Racks mit jeweils zwei CPs
- 2 Netzwerke
- PC-Station mit 2 CPs (z. B. CP 1623)

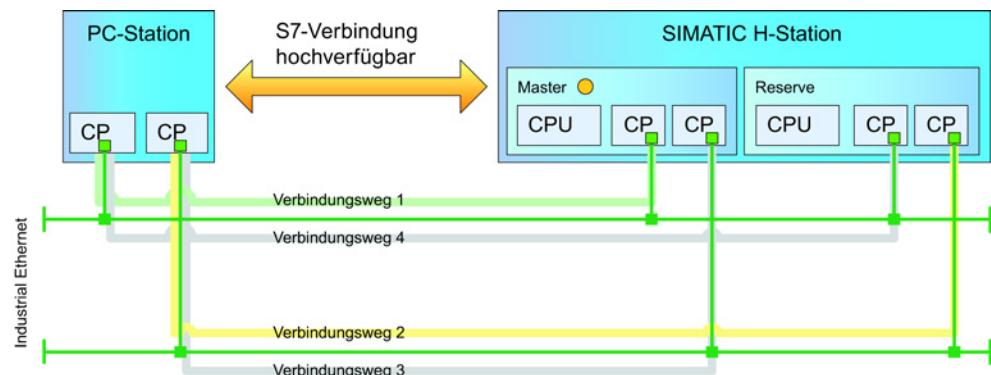


Bild 2-28 Beispiel für eine 4-Wege-Konfiguration über zwei Netze

### Hinweis

Die Netzwerkschnittstellen einer CPU 417-5H sind für 4-Wege-Verbindungen nicht geeignet. Grund: ein Mischbetrieb mit CPs innerhalb einer hochverfügbaren S7-Verbindung ist nicht möglich. Jedoch können beispielsweise zusätzlich zu den 4-Wege-Verbindungen auch 2-Wege-Verbindungen zu den CPU-Schnittstellen betrieben werden.

Beachten Sie, dass nach Ausfall z. B. des Verbindungswegs 2 und anschließender Hinzunahme von Weg 3 keine ideale Redundanz mehr vorliegt, da Weg 1 und Weg 3 in denselben Anlagenteil der SIMATIC H-Station führt, so dass eine Störung dieses Anlagenteils sofort zu einem Verbindungsabbruch führen würde. Dies gilt auch für das nachfolgende Beispiel.

#### 4-Wege-Kommunikation über 4 Netze

Eine andere Möglichkeit ist die 4-Wege-Kommunikation über vier Netze wie in folgendem Bild dargestellt. Sie benötigen dafür den Vollausbau von vier Ethernet-CPs in der PC-Station.

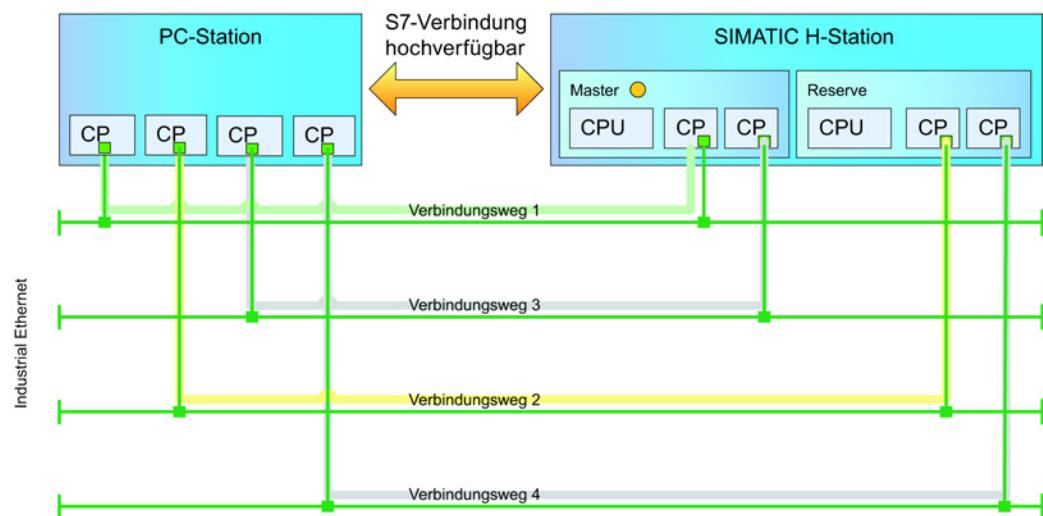


Bild 2-29 Beispiel für eine 4-Wege-Konfiguration über vier Netze

#### Hinweis

Schließlich kann auch 4-Wege-Kommunikation über nur ein Netz genutzt werden. Dann ist in der PC-Station nur ein CP angeschaltet und alle vier Verbindungswege werden über dasselbe Netz geführt. Dies kann sinnvoll sein, wenn die Verfügbarkeit von Netzen und PC-Stationen auf andere Weise sichergestellt ist (z. B. durch redundante Server und redundante Ringe im Netzwerk).

#### Was muss ich bei der Nutzung von hochverfügbaren S7-Verbindungen über ISO-on-TCP beachten?

Bei Nutzung von hochverfügbaren S7-Verbindungen über ISO-on-TCP ist zu beachten, dass in der PC-Station die Verbindungsüberwachung im Prozessraum der Applikation abläuft und nur korrekt funktionieren kann, wenn dem Anwendungsprozess jederzeit genügend Systemressourcen (insbesondere CPU-Leistung) zur Verfügung stehen.

### Was muss ich bei der Nutzung von hochverfügbaren S7-Verbindungen über SOFTNET-Baugruppen und ISO-on-TCP beachten?

Beim Betrieb von SOFTNET-Baugruppen und dem ISO-on-TCP-Protokoll ist folgendes unbedingt zu beachten:

Der Transport bei ISO-on-TCP erfolgt letztlich über das TCP/IP-Protokoll. Das IP-Routing wird bei SOFTNET vom Betriebssystem des PCs übernommen. Daher müssen Sie bei der Wahl der Adressen darauf achten, dass das IP-Routing eindeutig erfolgen kann, damit die Kommunikation der verschiedenen Verbindungswege über die richtigen Schnittstellen geleitet wird.

Das heißt: Die IP-Adressen aller angeschlossenen CPs in einer PC-Station und in den Teilstationen der S7-400H müssen in unterschiedlichen Subnetzen liegen. Die folgende Abbildung zeigt beispielhaft die Verwendung der beiden Subnetze 140.1.\* und 140.2.\*. Achten Sie auch auf die Verwendung der korrekten Netzmaske (hier 255.255.0.0) bei allen betroffenen CPs. Das gilt auch für alle in der PC-Station eingebauten nicht-projektierten SOFTNET-Baugruppen.

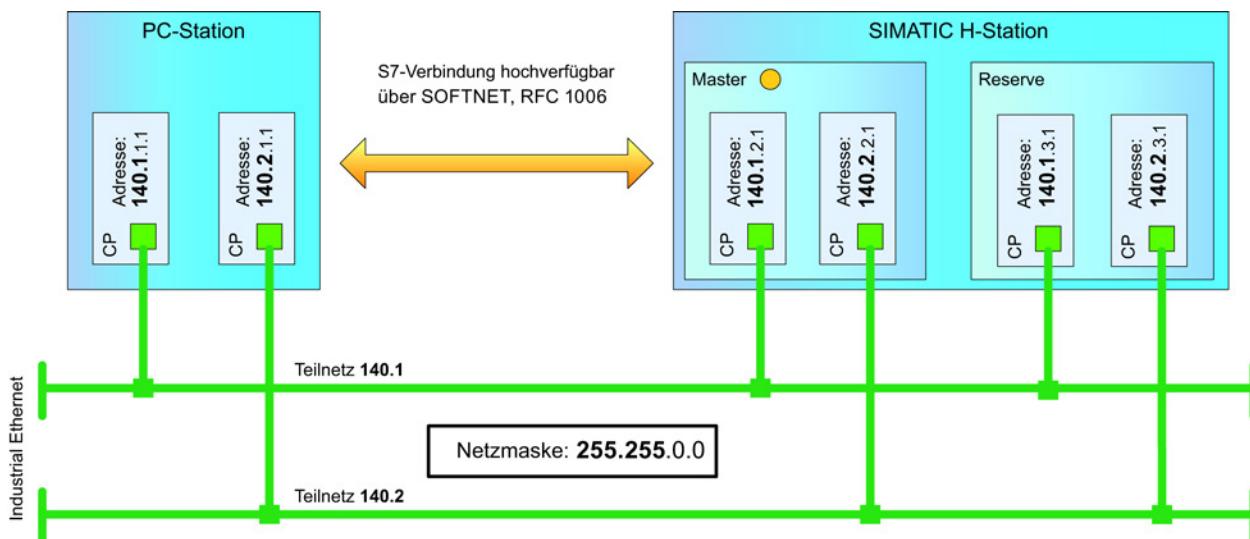


Bild 2-30 Hochverfügbare S7-Verbindungen über SOFTNET, ISO-on-TCP

Sonst können folgende Fehlerfälle auftreten:

- Angenommen, die benutzten Teilnetze sind verbunden (z. B. über einen redundanten Ring) und der Subnetzanteil der IP-Adressen ist nicht eindeutig. Dann werden unter Umständen beide Verbindungswege über denselben physikalischen Verbindungsweg geroutet. Die Verbindung wird dann nur scheinbar redundant aufgebaut und bricht bei Ausfall dieses Verbindungsweges ab.  
Eine solche scheinbare Redundanz ist besonders gefährlich, weil sie möglicherweise erst bei Auftreten einer Störung bemerkt wird. Sie können und sollten dies prüfen, indem Sie wechselweise einzelne beteiligte CPs vom Kommunikationsnetz trennen und nachweisen, dass die Verbindung in einen nicht-redundanten Zustand übergeht (siehe auch Kapitel "Diagnose, Inbetriebnahme, Wartung, Betrieb (Seite 63)").
- Bei physikalisch getrennten Teilnetzen und nicht-eindeutigem Subnetzanteil der IP-Adressen kann die Verbindung unter Umständen nicht redundant aufgebaut werden, wenn alle Verbindungswege über dasselbe Netz geroutet werden. Dann ist der entsprechende Verbindungspartner nicht erreichbar.

---

#### Hinweis

Bei Betrieb über HARDNET-Baugruppen gibt es dieses Problem nicht, da hier der TCP/IP-Verkehr von den Baugruppen selbst übernommen wird. Achten Sie trotzdem auch hier unbedingt auf korrekte IP-Adressierung im gesamten Netz.

---

### 2.7.5.2 Projektierung

Die Nutzung von hochverfügbaren S7-Verbindungen setzt eine Verbindungsprojektierung unbedingt voraus. In diesem Kapitel erfahren Sie, wie Sie die Projektierung vornehmen und worauf Sie dabei achten müssen.

#### Welches sind die Voraussetzungen für eine Verbindungsprojektierung von hochverfügbaren S7-Verbindungen?

Für die Verbindungsprojektierung benötigen Sie das Projektierwerkzeug Step7. Hochverfügbare S7-Verbindungen können Sie nur zwischen PC-Stationen und S7-400-H-Stationen projektieren.

Die PC-Station muss wie bei jeder Verbindungsprojektierung einen OPC Server und/oder eine Applikation enthalten sowie die benötigten Baugruppen.

Vernetzen Sie nun die PC-Station mit der H-Station mit den für die Verbindung erforderlichen Industrial Ethernet-Netzen nach einer der oben beschriebenen Konfigurationen.

---

#### Hinweis

Beachten Sie, wie weiter oben beschrieben, bei Verwendung des Protokolls ISO-on-TCP unbedingt auf korrekte IP-Adressen aller beteiligten Anschlüsse.

---

#### Versionen:

Für die Projektierung von hochverfügbaren S7-Verbindungen benötigen Sie STEP 7-Version 5.1 SP1 oder höher.

Für die Nutzung des RFC 1006-Protokolls für hochverfügbare S7-Verbindungen benötigen Sie STEP 7-Version 5.5 SP3 oder höher. Für die einzusetzenden Komponenten aus dem Hardware-Katalog gilt für dieses Protokoll folgendes:

- OPC Server und Applikationen: V8.1.2 oder neuer
- CP 1612 A2 und "IE-Allgemein": V8.1.2 oder neuer (CP 1612 A2: nur Windows 7 und Windows Server 2008 R2)

### **Wie werden hochverfügbare S7-Verbindungen projektiert?**

Die Verbindungsprojektierung ähnelt der von Standardverbindungen. Achten Sie aber darauf, dass Sie als Verbindungstyp „S7-Verbindung hochverfügbar“ angeben.

### **Welche Besonderheit gibt es bei der Projektierung von hochverfügbaren S7-Verbindungen?**

Gegenüber den Standard-S7-Verbindungen gibt es bei den hochverfügbaren S7-Verbindungen einige zusätzliche Parameter und Besonderheiten, die Sie beachten müssen.

#### **Verbindungsendpunkt:**

Die Initiative beim Verbindungsaufbau geht immer von der PC-Station aus. Die PC-Station ist also immer der aktive, die H-Station der passive Kommunikationspartner. Das entsprechende Auswahlkästchen ist deswegen nicht bedienbar.

#### **Hochverfügbarkeit:**

Wenn Sie die Verbindung zwischen PC-Station und H-Station anlegen, werden Ihnen bereits zwei geeignete Verbindungswege vorgeschlagen. Ebenfalls je nach Vernetzung haben Sie auch die Möglichkeit der Redundanzerhöhung auf 4 Wege. Sie erhalten dann eine Liste von vier Verbindungs wegen.

Sie können die Liste der Verbindungswege - je nach Vernetzung - jeweils durch Wahl anderer CP-Schnittstellen noch beeinflussen.

Beachten Sie jedoch folgendes: Falls Sie eine H-CPU mit Industrial Ethernet-Schnittstelle einsetzen, ist die Projektierung einer 4-Wege-Verbindung, die auch diese Schnittstellen miteinbezieht, nicht möglich.

#### **Protokoll, Überwachungszeit:**

Wie bei Standardverbindungen über Industrial Ethernet haben Sie die Möglichkeit sich für das ISO- oder das TCP-Protokoll (RFC 1006) zu entscheiden, falls bei den Kommunikationspartnern beide Protokolle aktiviert sind. Falls Sie das TCP-Protokoll (RFC 1006) auswählen (oder kein ISO-Protokoll möglich ist, weil nicht aktiviert oder nicht vorhanden), müssen Sie die für Ihre Verbindungsprojektierung passende Überwachungszeit eintragen.

Passende Werte der Überwachungszeiten in Abhängigkeit von der Anzahl der hochverfügbaren S7-Verbindungen finden Sie bei den Mengengerüstangaben zum Produkt S7-Redconnect. Stellen Sie diese Überwachungszeiten bei der Projektierung der hochverfügbaren S7-Verbindungen über TCP/IP in den Verbindungseigenschaften ein.

Beachten Sie, dass der Multiplikator 100 ms beträgt, so dass Sie z. B. für eine Überwachungszeit von 10 Sekunden den Wert 100 eintragen müssen. Die Einstellung von Werten über 25 Sekunden ist nicht sinnvoll, da die TCP-Keep Alive-Überwachung (30 Sekunden) nach Ablauf dieser Zeit bereits zum Abbau des Verbindungswegs führt.

**CP-Optionen:**

Aktivieren Sie für beteiligte S7-400-CPs die "Schnelle Umschaltung der Verbindung" (Optionen des CPs).

**Keepalive-Zeit:**

Stellen Sie sicher, dass für alle beteiligten CPs (sowohl PC-Station als auch H-Station) eine Keepalive-Zeit von 30 Sekunden eingeschaltet ist. Dies ist die Voreinstellung.

**OPC Server:**

In manchen Fällen reicht die Überwachungszeit für den Verbindungsauflaufbau des OPC Server nicht aus. Falls der OPC Server so projektiert ist, dass er die Verbindung nur bei Bedarf aufbaut, hat dies zur Folge, dass der OPC Server nach Störungen unter Umständen die Verbindung nicht mehr aufbaut. Erhöhen Sie in diesem Fall die Überwachungszeit für den Verbindungsauflaufbau.

### 2.7.5.3 Diagnose, Inbetriebnahme, Wartung, Betrieb

Für die Diagnose von hochverfügbaren S7-Verbindungen gibt es folgende Möglichkeiten:

- Ermittlung des Gesamtzustands
- Ermittlung der Zustände der einzelnen Verbindungswege
- Erkennung von Störungen
- Feststellung von Störungsursachen

Bei der Inbetriebnahme können Sie damit feststellen, ob alle Verbindungswege funktionieren und durch manuelles Herbeiführen von Störungen das Verhalten der Verbindung nachvollziehen.

Nach Durchführung von Wartungsarbeiten können Sie mit diesen Mechanismen den ordnungsgemäßen Zustand des Systems und aller hochverfügbaren S7-Verbindungen wieder herstellen.

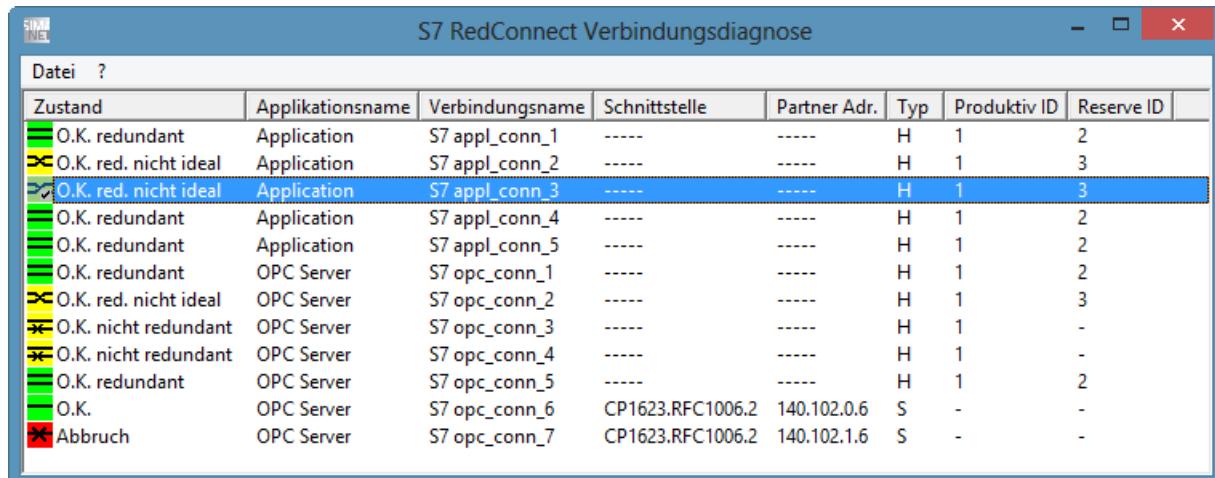
Während des Betriebs kann der Verbindungszustand beobachtet werden. Ereignisse, die den Zustand der Verbindungen betreffen, werden sofort gemeldet.

Es gibt folgende Möglichkeiten, die Diagnose von hochverfügbaren S7-Verbindungen zu nutzen:

- Mit der S7-Verbindungsdiagnose
- Mit OPC
- Mit einem eigenen Anwendungsprogramm der SAPI-S7-Programmierschnittstelle (siehe Kapitel "Diagnosediensste für hochverfügbare Verbindungen" im Handbuch "S7 Programmierschnittstelle", welches sich in der Manual Collection der DVD "SIMATIC NET PC Software" befindet)

## Die S7-Verbindungsdiagnose, was ist das?

Die S7-Verbindungsdiagnose ist eine Applikation der "SIMATIC NET PC Software", die alle projektierten S7-Verbindungen zusammen mit ihren Zuständen und anderen Informationen in einer Listenansicht zeigt.



The screenshot shows a Windows application window titled "S7 RedConnect Verbindungsdiagnose". The window has a standard title bar with minimize, maximize, and close buttons. Below the title bar is a menu bar with "Datei" and a help icon. The main area is a table with the following columns: Zustand, Applikationsname, Verbindungsname, Schnittstelle, Partner Adr., Typ, Produktiv ID, and Reserve ID. The table lists 14 network connections, each with a small colored icon indicating its status. The connections are as follows:

Zustand	Applikationsname	Verbindungsname	Schnittstelle	Partner Adr.	Typ	Produktiv ID	Reserve ID
O.K. redundant	Application	S7 appl_conn_1	----	----	H	1	2
O.K. red. nicht ideal	Application	S7 appl_conn_2	----	----	H	1	3
<b>O.K. red. nicht ideal</b>	<b>Application</b>	<b>S7 appl_conn_3</b>	<b>----</b>	<b>----</b>	<b>H</b>	<b>1</b>	<b>3</b>
O.K. redundant	Application	S7 appl_conn_4	----	----	H	1	2
O.K. redundant	Application	S7 appl_conn_5	----	----	H	1	2
O.K. redundant	OPC Server	S7 opc_conn_1	----	----	H	1	2
O.K. red. nicht ideal	OPC Server	S7 opc_conn_2	----	----	H	1	3
O.K. nicht redundant	OPC Server	S7 opc_conn_3	----	----	H	1	-
O.K. nicht redundant	OPC Server	S7 opc_conn_4	----	----	H	1	-
O.K. redundant	OPC Server	S7 opc_conn_5	----	----	H	1	2
O.K.	OPC Server	S7 opc_conn_6	CP1623.RFC1006.2	140.102.0.6	S	-	-
Abbruch	OPC Server	S7 opc_conn_7	CP1623.RFC1006.2	140.102.1.6	S	-	-

Bild 2-31 S7-Verbindungsdiagnose

Sobald nun eine Störung auftritt, die sich auf einen Verbindungszustand auswirkt, spiegelt sich dies in der Verbindungsdiagnose wider. Die Abbildung zeigt beispielhaft einige projektierte S7-Verbindungen. Hochverfügbare S7-Verbindungen erkennen Sie am Typ „H“, Standard-Verbindungen am Typ „S“.

Aufgebaute Standardverbindungen und hochverfügbare S7-Verbindungen, die den Zustand "redundant-ideal" haben, sind mit einem grünen Symbol gekennzeichnet.

Die Abbildung "S7-Verbindungsdiagnose" zeigt eine Situation nach Auftreten einer Störung auf einem der Verbindungswege, der von einigen der Verbindungen benutzt wird. Die Folge ist deutlich sichtbar: die Standardverbindung, die diesen Weg benutzt hat, ist abgebrochen. 2-Wege-Verbindungen funktionieren weiter, sind aber nicht mehr redundant. Jeder weitere Ausfall auf dem anderen Verbindungsweg führt zum Verbindungsabbruch. Betroffene 4-Wege-Verbindungen haben einen weiteren Weg hinzugeschaltet und fallen nur aus, wenn eine Störung an einer Komponente auftritt, die Bestandteil der beiden nun benutzten Wege ist.

Nach Durchführung der Reparaturarbeiten erholen sich alle hochverfügbaren S7-Verbindungen automatisch wieder. Die Standardverbindung muss jedoch wieder neu aufgebaut werden.

### Weitere Eigenschaften der S7-Verbindungsdiagnose

Die Verbindungsdiagnose unterstützt auch die Überwachung von projektierten Standard-S7-Verbindungen.

Weitere Details zu den Verbindungen werden nach Doppelklick auf die Verbindung oder Drücken der Enter-Taste nach Auswahl der Verbindung angezeigt.

Nach Projektierungsänderungen aktualisiert sich die Anzeige der Verbindungsliste automatisch.

Zur Bedienung beachten Sie bitte auch die Hilfe zu dieser Applikation.

## Wie kann ich hochverfügbare S7-Verbindungen mit OPC diagnostizieren?

Sie diagnostizieren hochverfügbare S7-Verbindungen wie Standard-S7-Verbindungen über die S7-spezifischen Diagnosevariablen des SIMATIC NET OPC Server. Für hochverfügbare S7-Verbindungen gibt es im Vergleich zu Standardverbindungen erweiterte Zustände und Zustände der Verbindungswege. Sie können diese Variablen mit einer OPC-Client-Applikation lesen und beobachten.

Mit dem mitgelieferten Werkzeug „OPC Scout V10“ als OPC-Client können alle S7-Verbindungen diagnostiziert werden. Dieser nutzt ebenfalls die Diagnosevariablen des SIMATIC NET OPC Server. Beachten Sie dazu auch die Hilfe zur Diagnose-Ansicht dieser Applikation.

## 2.7.6 S7-Protokoll, wie wird es projektiert?

### So wird das S7-Protokoll projektiert

Für die Kommunikation mit dem S7-Protokoll sind vor der Verwendung Verbindungen zu projektieren. Hierfür steht das Projektierungswerkzeug "SIMATIC STEP 7 Professional" zur Verfügung. Die projektierten Verbindungen werden durch einen eindeutig festgelegten Verbindungsnamen identifiziert. Es sind für das S7-Protokoll zwei Verbindungstypen vordefiniert:

- S7-Verbindung: Verbindung über PROFIBUS oder Ethernet
- S7-Verbindung hochverfügbar: Verbindung über redundante Verbindungswege

Für jede projektierte Verbindung sind Parameter einzustellen, für die das Projektierungswerkzeug beim Anlegen der Verbindung Defaultwerte vorgibt, die der Anwender ohne Änderung übernehmen kann. Die wesentlichen Parameter sind:

- der Dienstzugangspunkt der Transportschicht (TSAP).
- Art der Verbindung:
  - S7-Verbindung über PROFIBUS
  - S7-Verbindung über Ethernet mittels TCP/IP-Protokoll
  - S7-Verbindung über Ethernet mittels ISO-Transportprotokoll

### Was sind unprojektierte S7-Verbindungen?

Üblicherweise werden Verbindungen zu Partnergeräten in einer Projektierung festgelegt. Dafür steht das Projektierungswerkzeug "SIMATIC STEP 7 Professional" zur Verfügung. Allerdings gibt es Anwendungsfälle, bei denen beispielsweise Daten von einem Partnergerät gelesen bzw. Variablen geschrieben oder beobachtet werden sollen. Es besteht die Möglichkeit, diese Aufgabe auch ohne Projektierung durchzuführen, damit auch Fremdsoftware einfach auf Kommunikationsvariablen zugreifen kann.

**Welche Voraussetzungen gelten für die Nutzung von unprojektierten S7-Verbindungen?**

Für einen Gerätezugriff ohne Projektierung müssen alle kommunikationsrelevanten Daten des Partnergerätes bekannt sein. Dazu gehören unter anderem der Verbindungsname, der Zugangspunkt (Auswahl des CPs), der remote TSAP und die Stationsadresse.

**Wie können unprojektete S7-Verbindungen erzeugt werden?**

Unprojektete S7-Verbindungen können mit dem OPC Server oder mit dem Konfigurationswerkzeug "Kommunikations-Einstellungen" über "COMLS7" erzeugt werden.

**Welche Vor- und Nachteile haben unprojektete S7-Verbindungen?**

Die Verwendung von unprojekteten S7-Verbindungen bietet den Vorteil, dass ein schnellerer Zugriff auf Partnergeräte ermöglicht wird.

Die Verwendung von unprojekteten S7-Verbindungen hat den Nachteil, dass alle kommunikationsrelevanten Informationen vom Partnergerät bekannt sein müssen. Außerdem stehen für unprojektete Verbindungen keine Blockdienste zur Verfügung.

**2.7.7 S7-Protokoll, welches sind die Vor- und Nachteile?**

**Das sind die Vorteile des S7-Protokolls**

Die Verwendung des S7-Protokolls bietet folgende Vorteile:

- Alle Dienste sind uneingeschränkt über PROFIBUS und Ethernet verfügbar.
- Zugriff auf Partnergeräte ohne Programmierung der Partner.
- Zugriffskontrolle durch Passwort.
- Zugriff (Lesen/Schreiben/Löschen) auf ladbaren Bereich eines Automatisierungssystems.
- Alarne werden gepuffert und können nicht verloren gehen.
- Alle Vorteile eines verbindungsorientierten Protokolls

**Das sind die Nachteile des S7-Protokolls**

Die Verwendung des S7-Protokolls hat folgende Nachteile:

- Herstellerabhängig, das S7-Protokoll ist nur im SIMATIC S7 Spektrum realisiert.
- Nicht kompatibel zur S5-Kommunikation.

## **2.8      Das SNMP-Protokoll**

### **2.8.1      SNMP-Protokoll, was ist das?**

#### **Das SNMP-Protokoll**

Das SNMP-Protokoll (Simple Network Management Protocol) ist ein UDP-basiertes, offenes Protokoll für die Administration von Netzwerken. Es erlaubt ein zentrales Netzwerkmanagement für viele Netzwerkkomponenten, wie z.B. Router, Bridges, Hubs sowie Drucker, Server und Workstations. Die primären Ziele von SNMP sind die Verringerung der Komplexität der Managementfunktionen und der transparente Austausch von Informationen bzw. Daten zwischen verschiedenen Netzkomponenten. Dadurch unterstützt das SNMP-Protokoll das Überwachen, Steuern und Verwalten von beliebigen SNMP-fähigen Netzkomponenten.

### **2.8.2      SNMP-Protokoll, wie sieht eine typische Anlagenkonfiguration aus?**

Das folgende Kapitel zeigt, wie eine typische Anlagenkonfiguration bei Ethernet aussehen kann, in der die Datenkommunikation zwischen verschiedenen Geräten mittels SNMP-Protokoll realisiert ist.

### Beispiel einer Anlagenkonfiguration für das SNMP-Protokoll

Zur Kommunikation mit dem SNMP-Protokoll über Ethernet stehen im SIMATIC NET-Spektrum ausschließlich Kommunikationsbaugruppen für PCs und Workstations zur Verfügung. Zur Anwendung kommen hierbei die Kommunikationsbaugruppen wie z. B. CP 1623, CP 1628 oder Baugruppen anderer Hersteller. Weitere SNMP-fähige Baugruppen im SIMATIC NET-Spektrum sind Switches wie z. B. SCALANCE X300 oder SCALANCE X400. Die Konfiguration kann mit beliebigen SNMP-fähigen Netzkomponenten auch von verschiedenen Herstellern erweitert werden.

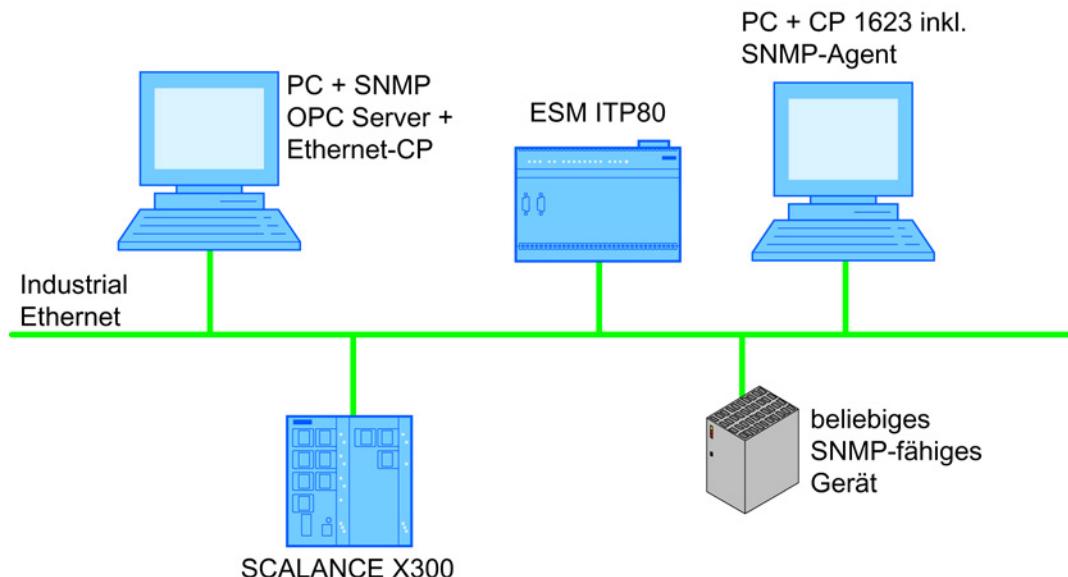


Bild 2-32 Typische Anlagenkonfiguration bei Ethernet

### 2.8.3 SNMP-Protokoll, wie funktioniert es?

#### So funktioniert das SNMP-Protokoll

Das SNMP-Protokoll arbeitet nach dem Client-Server-Modell. Der sogenannte SNMP-Agent läuft als Server auf einer administrierten Netzkomponente, verwaltet die zur Verfügung stehenden Daten und steuert die Netzkomponente. Der SNMP-Manager funktioniert als Client und kann über verschiedene SNMP-Dienste Daten mit dem SNMP-Agent austauschen, ihn überwachen oder auch konfigurieren.

### Wie greift das SNMP-Protokoll auf Daten zu?

Der SNMP-Agent verwaltet die verfügbaren Daten in einer MIB (Management Information Base). Die MIB ist eine Art Tabelle, in der alle Daten strukturiert abgelegt werden. Der SNMP-Manager kann über SNMP-Dienste die MIB des Agenten auslesen und somit gezielt auf die Daten zugreifen, die im SNMP-Manager benötigt werden oder die im SNMP-Agent überschrieben werden sollen.

## 2.8.4 SNMP-Protokoll, welche Kommunikationsdienste stehen zur Verfügung?

### Diese Kommunikationsdienste stellt das SNMP-Protokoll zur Verfügung

Für die Kommunikation zwischen einem SNMP-Manager und einem SNMP-Agenten stellt das SNMP-Protokoll im wesentlichen fünf Kommunikationsdienste bereit.

- **Get-Request:** Mit dem Dienst "Get-Request" fordert der SNMP-Manager beim SNMP-Agenten Daten an, die dieser in seiner MIB verwaltet.
- **Get-Next-Request:** Der Dienst "Get-Next-Request" ermöglicht den Zugriff des SNMP-Managers auf die nächst folgenden Daten im SNMP-Agenten.
- **Get-Response:** Der Dienst "Get-Response" ist die Antwort auf "Get-Request" oder "Get-Next-Request" und wird immer vom SNMP-Agenten zum SNMP-Manager gesendet.
- **Set-Request:** Zum Schreiben von Daten auf dem SNMP-Agenten verwendet der SNMP-Manager den Dienst "Set-Request".
- **TRAP:** Besondere Daten können vom SNMP-Agenten unaufgefordert und ereignisgesteuert zum SNMP-Manager gesendet werden. Hierzu verwendet der SNMP-Agent den Dienst "Trap".

## 2.8.5 SNMP-Protokoll, wie wird es projektiert?

### So wird das SNMP-Protokoll projektiert

Für die Kommunikation mit dem SNMP-Protokoll ist vor der Verwendung die Anlagenkonfiguration aller SNMP-fähigen Partnergeräte zu projektiern. Hierfür steht das Projektierungswerkzeug "SIMATIC STEP 7 Professional" zur Verfügung. Mit dem Anwendungsprogramm HWKonfig werden für das SNMP-fähige Partnergerät mehrere Parameter festgelegt, die das Gerät eindeutig identifizieren. Die zu projektierenden Parameter sind:

- Name des Gerätes: eindeutige, technologisch sinnvolle Bezeichnung
- Adresse des Gerätes: IP-Adresse am Ethernet
- Gerätprofil: Beschreibt die Struktur der über das SNMP-Protokoll verfügbaren Geräteinformationen

## **2.8.6 SNMP-Protokoll, welches sind die Vor- und Nachteile?**

### **Das sind die Vorteile des SNMP-Protokolls**

Das SNMP-Protokoll bietet folgende Vorteile:

- Offenes Protokoll, das von vielen Herstellern unterstützt wird.
- Ist in Ethernet-Netzwerken weit verbreitet.
- Viele unterschiedliche Netzkomponenten werden unterstützt, z.B. Switches, Drucker, PCs, Netzwerkarten.
- Kommunikation kann ereignisgesteuert sein, dadurch wenig Netzlast.

### **Das sind die Nachteile des SNMP-Protokolls**

Das SNMP-Protokoll hat folgende Nachteile:

- Kein Diagnoseprotokoll und damit keine Netzdiagnose möglich.
- Keine Statistiken verfügbar.
- Keine Parametrierung möglich.

## **2.9 Die Kommunikation mit PROFINET IO**

### **2.9.1 PROFINET IO, was ist das?**

#### **Das ist PROFINET IO**

PROFINET IO ist ein Automatisierungskonzept für die Realisierung von modularen und dezentralen Applikationen am Industrial Ethernet. Durch PROFINET IO werden dezentrale Peripherie und Feldgeräte in die Ethernet-Kommunikation eingebunden. Dabei wird die gewohnte IO-Sicht von PROFIBUS DP verwendet, bei der die Nutzdaten der Feldgeräte zyklisch und zeitunkritisch oder in einem Echtzeitkanal in das Prozessabbild eines Automatisierungssystems übertragen werden.

PROFINET IO beschreibt ein Gerätemodell, das sich an den Grundzügen von PROFIBUS DP orientiert und aus Steckplätzen (Slots) und Kanälen (Subslots) besteht. Auch das Engineering von PROFINET IO erfolgt so wie es Systemintegratoren von PROFIBUS DP seit langem gewohnt sind. Dabei werden die dezentralen Feldgeräte in der Projektierung einem Automatisierungsgerät zugeordnet und als sogenanntes PROFINET-Gerät bezeichnet.

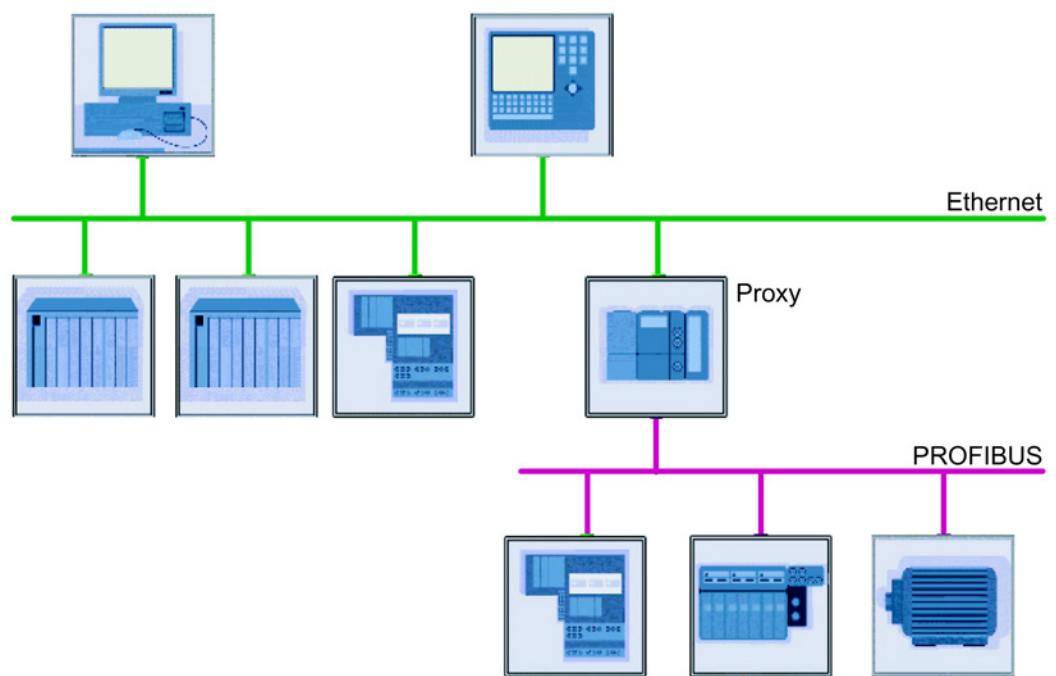


Bild 2-33 Integration von PROFIBUS in die Ethernet-Kommunikation

## 2.9.2 PROFINET IO, wie sieht eine typische Anlagenkonfiguration aus?

### Beispiel einer Anlagenkonfiguration für PROFINET IO

Im folgenden Kapitel wird gezeigt, wie eine typische Anlagenkonfiguration bei PROFINET IO aussehen kann. Dabei wird deutlich, welche Möglichkeiten und Flexibilität PROFINET IO bietet.

Gezeigt wird ein Industrial Ethernet, an dem PROFINET-Geräte vom Typ IO-Controller und IO-Device angeschlossen sind. Zusätzlich wird über ein IE/PB-Link ein PROFIBUS-Segment als unterlagertes Bussystem transparent eingebunden.

## 2.9 Die Kommunikation mit PROFINET IO

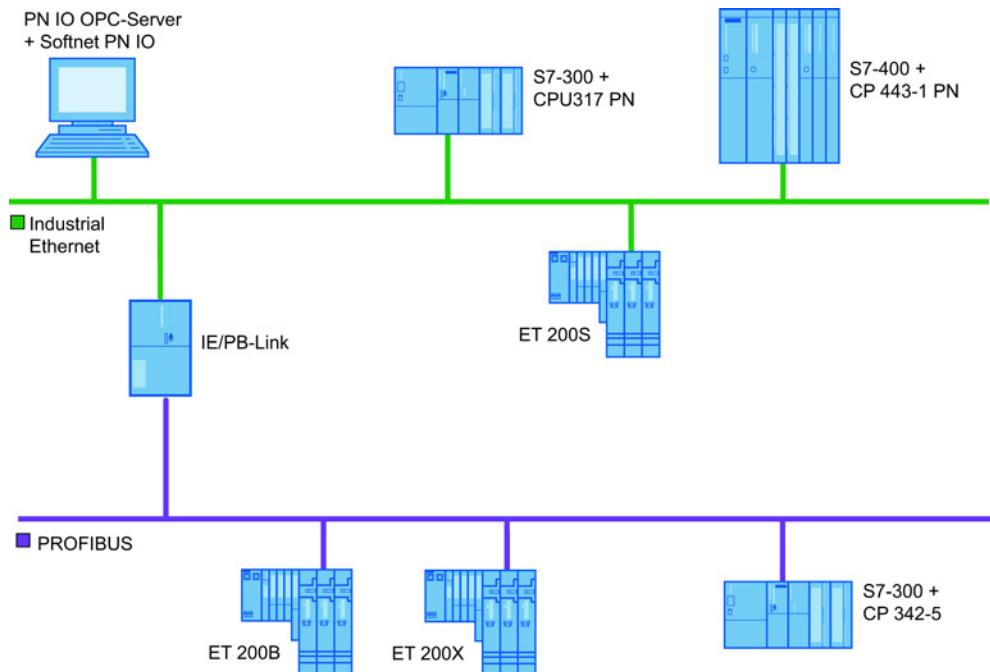


Bild 2-34 Typische Anlagenkonfiguration bei PROFINET IO

Tabelle 2-3 Beschreibung der einzelnen PROFINET-Geräte

PROFINET-Geräte	Gerätetyp	Beschreibung
1	IO-Controller	Der CP, z.B. der Ethernet CP 1616, im PC ist ein PROFINET IO-Controller und kommuniziert mit verschiedenen IO-Devices. Auf dem PC läuft beispielsweise ein PN IO OPC-Server oder eine PN IO-Applikation.
2	IO-Controller	Das S7-300-Gerät funktioniert als IO-Controller und kommuniziert mit verschiedenen IO-Devices.
3	IO-Controller	Das S7-400-Gerät funktioniert als IO-Controller und kommuniziert mit verschiedenen IO-Devices.
4	IO-Device	Der IE/PB-Link erfüllt im Sinne von PROFINET IO die Proxy-Funktionalität und stellt jedes unterlagerte PROFIBUS-Gerät transparent als PROFINET IO-Device am Ethernet dar.
5	IO-Device	Das ET 200S-Gerät funktioniert als IO-Device und ist einem IO-Controller zugeordnet.

### 2.9.3 PROFINET IO, wie funktioniert es?

#### So funktioniert PROFINET IO

Mit PROFINET IO erfolgt die Einbindung der dezentralen Peripherie am Industrial Ethernet. Controller und Device arbeiten nach dem **Provider-Consumer-Modell**, in dem der Provider Daten erzeugt und versendet, die der Consumer empfängt und verarbeitet. Dabei ist das Controller-Device-Prinzip mit dem von PROFIBUS DP bekannten Master-Slave-Prinzip vergleichbar.

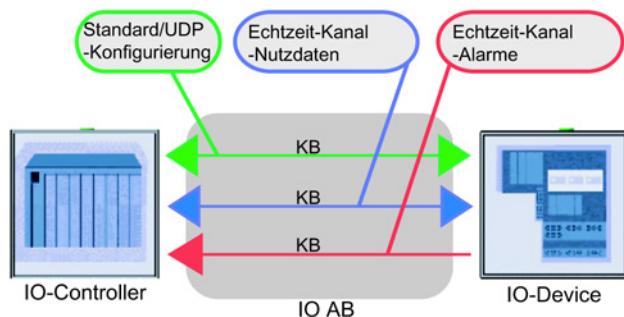
Aus Kommunikationssicht sind alle PROFINET-Geräte am Ethernet gleichberechtigt. Erst über die Projektierung wird jedem Gerät ein Typ zugeordnet, der die Art und Weise der Kommunikation nach dem Provider-Consumer-Modell festlegt.

Bei PROFINET IO unterscheidet man folgende drei Gerätetypen:

- Den **IO-Controller**  
Der IO-Controller ist ein Automatisierungsgerät, in dem ein Automatisierungsprogramm abläuft oder ein CP in einem PC, in dem z.B. ein OPC-Server realisiert ist.
- Das **IO-Device**  
Das IO-Device ist ein dezentrales Feldgerät, das einem IO-Controller zugeordnet ist.
- Den **IO-Supervisor**  
Der IO-Supervisor ist ein PC/PG mit Inbetriebnahme- und Diagnosefunktionen.

Zwischen IO-Controller und IO-Device können Daten über folgende Kanäle übertragen werden:

- Zyklische Nutzdaten über den Echtzeitkanal
- Ereignisgesteuerte Alarme über den Echtzeitkanal
- Azyklisches Lesen und Schreiben von Datensätzen, die Parametrierung und Konfiguration sowie Lesen von Diagnoseinformationen über den Standardkanal (NRT-Kanal) auf Basis von UDP/IP



KB: Kommunikationsbeziehung  
AB: Applikationsbeziehung

Bild 2-35 Kommunikationsprinzip zwischen IO-Controller und IO-Device

## 2.9 Die Kommunikation mit PROFINET IO

Zu Beginn der Kommunikation zwischen IO-Controller und IO-Device wird eine Applikationsbeziehung auf dem UDP/IP-Kanal eingerichtet. Diese enthält mehrere Kommunikationsbeziehungen entsprechend den schon erwähnten Kanälen zur Übertragung von Konfigurationsdaten, Nutzdaten und Alarmen.

Für die Kommunikation zwischen IO-Controller und IO-Supervisor wird ebenfalls eine Applikationsbeziehung aufgebaut. Dabei wird der UDP/IP-Kanal zur Übertragung von Diagnosedaten und für Up- und Downloadfunktionen genutzt.

Die Kommunikation vom IO-Supervisor zum IO-Device basiert ebenso auf einem UDP/IP-Kanal im Rahmen einer Applikationsbeziehung. Neben Diagnosedaten werden hier Statusinformationen und Parameterdaten übertragen.

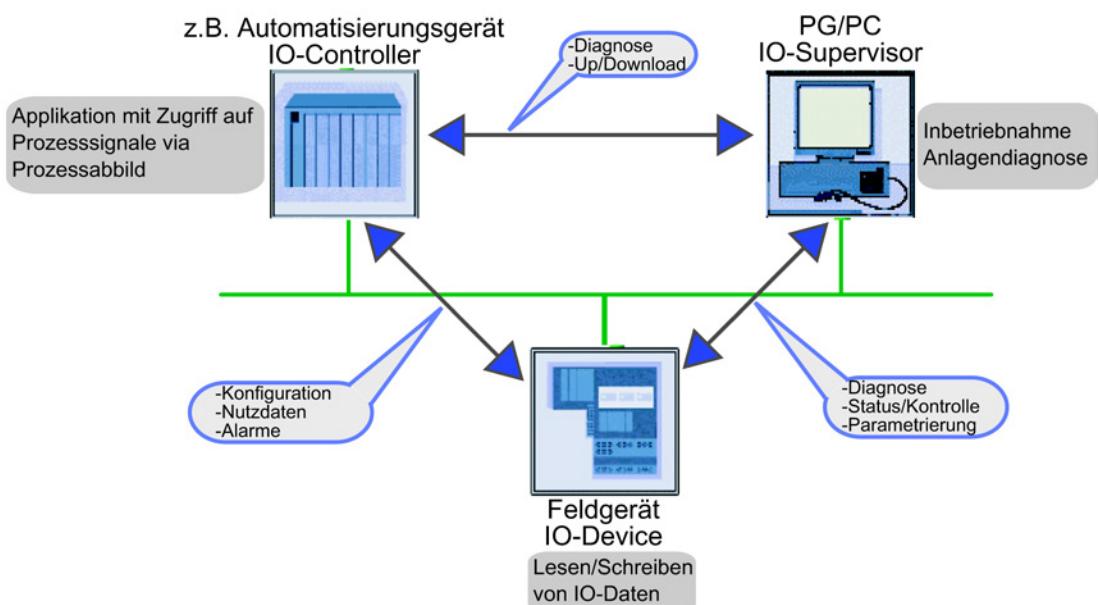


Bild 2-36 Funktionsumfang von PROFINET IO

### Welche Protokolle werden bei PROFINET IO benutzt?

Bei PROFINET IO wird zu Beginn einer Kommunikation zwischen PROFINET IO-Geräten UDP/IP für die Initierung des Datenaustausches, die Parametrierung der dezentralen Peripheriegeräte und die Diagnose eingesetzt. Als Applikationsprotokoll kommt das RPC-Protokoll zur Anwendung. Das RPC-Protokoll (Remote Procedure Call) ist ein Protokoll, das die Implementierung von verteilten Anwendungen in einem Netzwerk erlaubt. Es ermöglicht außerdem den Zugriff von HMI-Stationen oder Engineeringssystemen als IO-Supervisor auf PROFINET IO-Devices. Für die Übertragung der Nutzdaten und Alarme kommt dann der PROFINET-Echtzeitkanal zum Einsatz.

In einer typischen PROFINET IO-Konfiguration gibt es einen IO-Controller, der über Kommunikationsbeziehungen zu mehreren dezentralen Feldgeräten, den IO-Devices, zyklisch Daten austauscht. In jedem Zyklus werden die Eingangsdaten von den zugeordneten Feldgeräten zum IO-Controller gesendet und im Gegenzug die Ausgangsdaten an die entsprechenden Feldgeräte zurückgesendet. Die Überwachung der Kommunikationsbeziehung geschieht dadurch, dass das Eintreffen von zyklischen Daten überwacht wird. Fallen zyklisch erwartete Informationen aus, so erkennt der IO-Controller, dass das entsprechende IO-Device ausgefallen ist.

## 2.9.4 PROFINET IO mit Isochroner Real Time-Kommunikation (IRT)

### Performance der drei Leistungsstufen von PROFINET IO

Gegenüber der Kommunikation über TCP/UDP und IP werden die Aktualisierungszeiten bei der RT-Kommunikation durch Wegfall mehrerer Protokollebenen (Schichten 4-6 des ISO/OSI-Referenzmodells) verkürzt.

Zusätzlich werden Echtzeit-Trogramme bei der RT-Kommunikation durch Nutzung von VLAN-Prioritäten priorisiert übertragen. Die höhere Priorität der Echtzeit-Trogramme gegenüber den TCP-/UDP-Daten und kürzere Speicherzeiten in den Switches führen zu einer weiteren Beschleunigung der Übertragung. Die in PROFINET IO realisierte echtzeitfähige RT-Kommunikation mit Aktualisierungszeiten <10 ms wird für anspruchsvolle Motion Control-Anwendungen durch eine weitere Stufe ergänzt.

Speziell für den Bereich Motion Control wurde die Isochrone Real Time-Kommunikation (IRT) definiert. Bei der IRT-Kommunikation können Aktualisierungszeiten <1 ms realisiert werden, indem zusätzlich zu der Priorisierung der Trogramme und den kürzeren Speicherzeiten in den Switches weitere Merkmale realisiert sind, die im Folgenden erläutert werden.

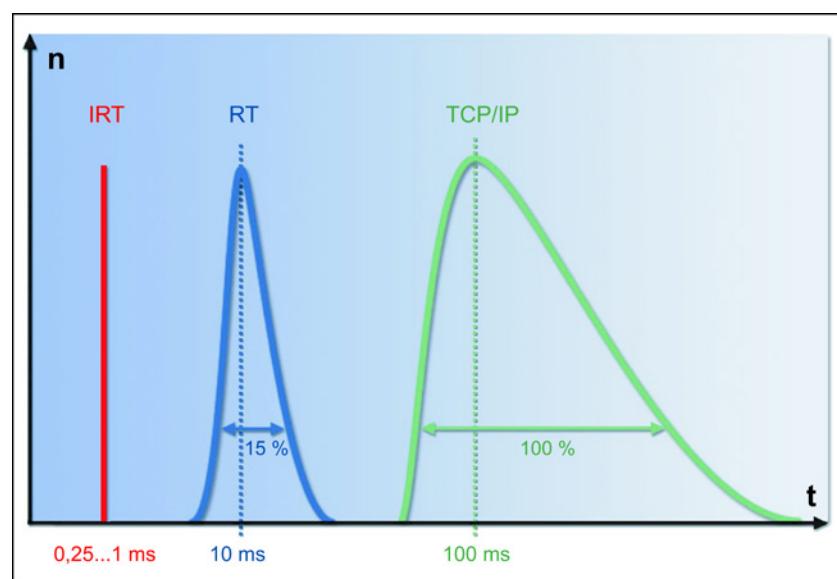


Bild 2-37 Vergleich der Aktualisierungszeiten der IRT-, RT- und TCP/IP-Kommunikation

## Was ist Isochrone Real Time-Kommunikation?

Die hohe Performance bei IRT wird über drei Hauptmerkmale erreicht:

- Die Aufteilung des Übertragungszyklus in zwei Intervalle
- Die taktsynchrone Übertragung durch Synchronisierung der Teilnehmer
- Die zeit- und wege-bezogene Planung der Kommunikation

## Die zwei Intervalle des IRT-Übertragungszyklus

Für eine bevorzugte Übertragung der Echtzeit-Telegramme innerhalb festgelegter Zeitschlüsse wird der Übertragungszyklus in zwei Intervalle aufgeteilt:

- Einen deterministischen IRT-Kanal für die Echtzeit-Telegramme
- Einen zeitunkritischen offenen Kanal für TCP/UDP und RT-Kommunikation

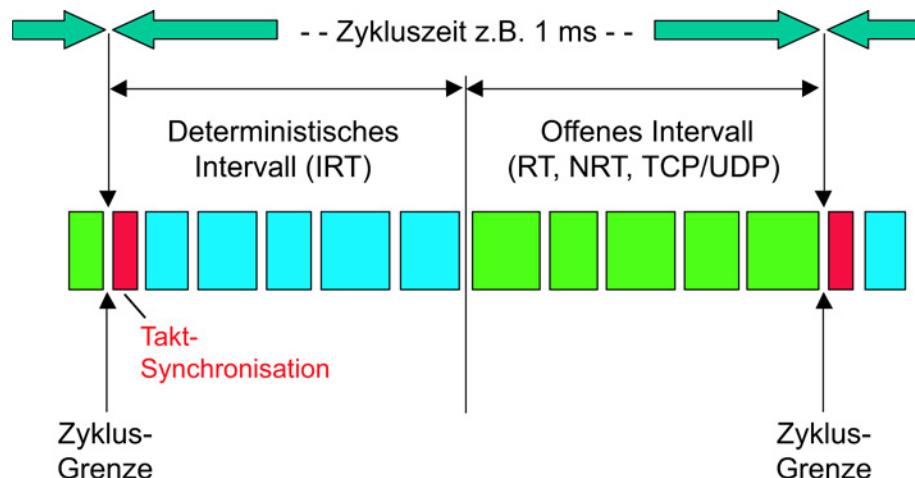


Bild 2-38 Aufbau eines Übertragungszyklus bei Nutzung des IRT-Kanals

Die echtzeitfähigen IRT-Telgramme werden dem deterministischen Intervall zugeordnet, das nur für die Übertragung dieser Daten reserviert ist.

## Die zeitliche Synchronisation bei IRT

Die einzelnen Zyklen bei der IRT-Kommunikation werden zeitlich synchronisiert, um die isochrone Übertragung der IRT-Telgramme und damit die extrem kurzen Zykluszeiten zu erreichen.

Für die zeitliche Synchronisation der beteiligten Teilnehmer einer IRT-Sync-Domäne wird ein Sync-Master projektiert, der Sync-Telgramme verteilt. Geräte, die sich auf die Zeitbasis des Sync-Masters synchronisieren, werden als Sync-Slaves bezeichnet. Der Sync-Master und die Sync-Slaves bilden zusammen eine IRT-Sync-Domäne. Gegenüber einer Domäne enthält eine IRT-Sync-Domäne nur PROFINET-Geräte mit IRT.

### Wie greift ein Anwenderprogramm auf taktsynchrone Prozessdaten zu?

Innerhalb des offenen Intervalls greift das Anwenderprogramm auf die Prozessdaten zu, und die für die IRT-Kommunikation projektierten Daten werden innerhalb des nächsten deterministischen Intervalls über den IRT-Kanal übertragen. Hierdurch wird eine konsistente Datenübertragung erreicht.

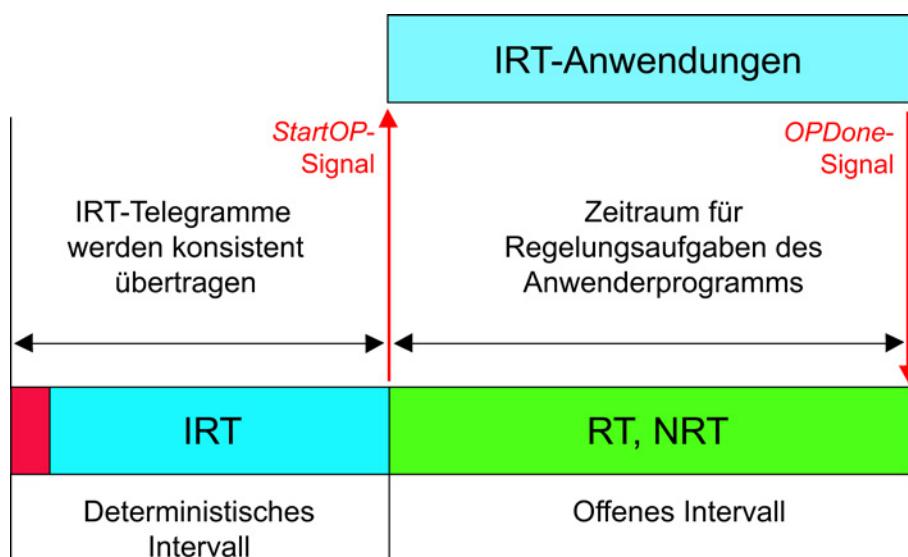


Bild 2-39 Zeitliche Synchronisierung der Übertragungszyklen bei der IRT-Kommunikation

Nach dem Senden der IRT-Télégramme gibt die Anwenderschnittstelle am Ende des IRT-Intervalls eine "StartOP"-Meldung (Start Operation) aus.

Nach der "StartOP"-Meldung kann das Anwenderprogramm im offenen Intervall seine zyklischen Regelungsaufgaben durchführen. Im offenen Intervall werden die zeitunkritischen NRT-Télégramme und die RT-Télégramme über den offenen Kanal übertragen.

Am Ende des Programm-Zyklus gibt das Anwenderprogramm eine "OPDone"-Rückmeldung aus. Die "OPDone"-Rückmeldung muss vor dem neuen Zyklusbeginn gemeldet werden.

Die Dauer des deterministischen Intervalls wird innerhalb der Projektierung festgelegt.

### Die zeit- und wege-bezogene Planung der IRT Kommunikation

Die IRT-Kommunikation wird mit den Siemens Projektierungswerkzeugen aufgebaut:

Kürzeste Übertragungszyklen können dadurch realisiert werden, dass die Kommunikationswege zwischen den einzelnen Partnern geplant werden. Hierzu werden die Verbindungen zwischen den einzelnen IO-Geräten und den zwischengeschalteten IRT-Switches in einer Topologieplanung projektiert, wobei die Kabellänge berücksichtigt werden muss. Der kürzestmögliche Zyklus wird durch Kalkulation der Laufzeiten zwischen den Teilnehmern vom Projektierungswerkzeug berechnet.

## **Hardware-Voraussetzungen für die IRT-Kommunikation**

Für die isochrone Datenübertragung mit Zykluszeiten < 1 ms bei einem Jitter der aufeinanderfolgenden Zyklen von 1 µs werden spezielle IRT-ASICs benötigt, sowohl in den beteiligten Controllern und Devices als auch bei den zwischengeschalteten Switches.

Siemens stellt verschiedene Komponenten zur Verfügung, die auf Basis der Ethernet-ASICs ERTEC 200 und ERTEC 400 eine hochperformante IRT-Kommunikation ermöglichen:

- Die Kommunikationsprozessoren CP 1604 und CP 1616 mit IO-Base-Software
- Die Switches SCALANCE X204IRT und X202IRT
- Weitere Komponenten sind in Vorbereitung.

## **Hinweise zur Verwendung von IRT bei SOFTNET PNIO**

---

### **Hinweis**

SOFTNET PNIO unterstützt kein IRT.

---

## **2.9.5 PROFINET IO, welche Kommunikationsdienste stehen zur Verfügung?**

### **Diese Kommunikationsdienste stellt PROFINET IO zur Verfügung**

Für die Kommunikation zwischen einem PROFINET IO-Controller und einem PROFINET IO-Device stellt PROFINET IO einige Kommunikationsdienste bereit. Hierbei werden Initialisierungsdienste und Produktivdienste unterschieden. Die Initialisierungsdienste sind:

- **IO-Controller Status:** Mit dem Dienst "IO-Controller Status" kann der IO-Controller seinen eigenen Status abfragen und verändern. Es sind die Statuszustände CLEAR, OPERATE und OFFLINE für IO-Geräte definiert.
- **IO-Device aktivieren:** Der Dienst "IO-Device aktivieren" ermöglicht dem IO-Controller, das IO-Device in den aktiven Zustand zu setzen.
- **IO-Device deaktivieren:** Der Dienst "IO-Device deaktivieren" ermöglicht es dem IO-Controller, das IO-Device in den deaktiven Zustand zu setzen.

Die Produktivdienste sind:

- **I/O-Daten lesen:** Mit dem Dienst "I/O-Daten lesen" liest der IO-Controller die zyklischen Eingangsdaten des IO-Device. Gleichzeitig werden zusätzlich remote Statusinformationen (Provider Status) vom IO-Device gelesen und die lokalen Statusinformationen (Consumer Status) zum IO-Device übertragen.
- **I/O-Daten schreiben:** Mit dem Dienst "I/O-Daten schreiben" modifiziert der IO-Controller die zyklisch zum IO-Device gesendeten Ausgangsdaten. Gleichzeitig werden zusätzlich lokale Statusinformationen (Provider Status) zum IO-Device übertragen.

- **Alarne empfangen und quittieren:** Der IO-Controller empfängt mittels des Dienstes "Alarne empfangen und quittieren" Alarminformationen vom IO-Device und kann diese zum IO-Device hin quittieren.
- **Datensatz lesen/schreiben:** Mit diesem Dienst kann der IO-Controller azyklisch mit dem IO-Device kommunizieren. Der IO-Controller liest Datensätze vom IO-Device oder schreibt Datensätze auf das IO-Device.

## 2.9.6 PROFINET IO, wie wird es projektiert?

### So wird PROFINET IO projektiert

Für die Kommunikation mit PROFINET IO ist jedes PROFINET IO-Gerät zu projektieren. Hierfür steht das Projektierungswerkzeug "SIMATIC STEP 7 Professional" zur Verfügung.

Für jedes projektierte PROFINET IO-Gerät sind Parameter einzustellen, für die das Projektierungswerkzeug beim Anlegen des Gerätes Defaultwerte vorgibt, die der Anwender ohne Änderung übernehmen kann. Wesentliche Parameter sind:

- die Aktualisierungszeit
- Adressen, über die die Geräte angesprochen werden

### Besonderheiten bei der Projektierung der IRT-Kommunikation

Bei der Projektierung der zeitbasierten IRT-Kommunikation müssen zusätzlich zu den Teilnehmern, die über den IRT-Kanal kommunizieren, auch die Adressen der zwischengeschalteten Switches projektiert werden. Den Switches werden die Planungsdaten zum Aufbau der Transferlisten automatisch beim Hochlaufen der Switches vom Controller übergeben.

Die PROFINET-Geräte mit IRT innerhalb einer IRT-Sync-Domäne können einem einzelnen IO-System oder mehreren IO-Systemen angehören. Grundsätzlich ist zu beachten, dass alle PROFINET-Geräte mit IRT, die innerhalb eines IO-Systems projektiert sind, nur einer IRT-Sync-Domäne angehören. IRT-Sync-Domänen dürfen sich nicht überlappen. Eine Verbindung zu PROFINET-Geräten einer anderen IRT-Sync-Domäne darf nur über PROFINET-Geräte oder Ports ohne IRT-Unterstützung erfolgen.

Eine IRT-Sync-Domäne darf nur Switches mit IRT-Unterstützung enthalten, keine Standard-Switches.

Die Zuordnung einzelner Datenpakete zu dem offenen NRT-Kanal, zum RT- oder IRT-Kanal erfolgt ebenfalls über die genannten Projektierungswerzeuge.

## **2.9.7 PROFINET IO, welches sind die Vorteile?**

### **Das sind die Vorteile von PROFINET IO**

Die Verwendung von PROFINET IO bietet folgende Vorteile:

- Investitionsschutz
- Leichte Anlagenerweiterung
- Minimierung der Kosten von Installation, Engineering und Inbetriebnahme
- Vertikale Integration der Ebenen der Automatisierungspyramide durch Integration von PROFIBUS
- Die Mengengerüste von PROFIBUS wurden bei gleichzeitiger höherer Performance erweitert
- Koexistente Nutzung von Echtzeit- und TCP-basierter Kommunikation auf einer Leitung
- Skalierbare Echtzeit-Kommunikation von performant bis hochperformant und taktsynchron
- Standardisierte Kommunikation zwischen PROFINET-Geräten

## **2.10 Security bei SIMATIC NET**

Alle Informationen zu Security bei SIMATIC NET finden Sie im Handbuch "Industrial Ethernet Security - Security einrichten". Dieses Dokument befindet sich auf der Manual Collection der "SIMATIC NET PC Software" im Ordner "doc" oder auf den Support-Seiten unter folgender Beitrags-ID:

60166939 (<http://support.automation.siemens.com/WW/view/de/60166939>)

# Grundlagen der OPC-Schnittstelle

## Überblick

Das folgende Kapitel gibt Ihnen einen Überblick über die Grundlagen von OPC und die Anwendungen von OPC bei SIMATIC NET.

Es beantwortet Ihnen Fragen rund um die Grundlagen von OPC, hilft Ihnen beim Einstieg in die Terminologie bei OPC und liefert Ihnen Erklärungen zu den einzelnen Begriffen. Es umreißt kurz Ihre Vorteile beim Einsatz von OPC mit SIMATIC NET, nämlich den OPC-Scout, Symbolik und Data OCX. Es gewährt einen Einblick in die Spezifikationen des Zugriffs auf Prozessdaten über Variablen (OPC Data Access), der Übertragung von Prozessalarmen und Ereignissen (Alarms & Events), den Zugriff auf das Internet (OPC XML).

Weiterhin werden die Eigenschaften der Spezifikation OPC Unified Architecture (OPC UA) beschrieben. Nicht zuletzt erhalten Sie Informationen über die Leistungsfähigkeit von OPC.

Nach der Lektüre dieses Kapitels sollten Sie mit Hilfe der detaillierten Angaben zum Einsatz von OPC in Band 2 dieses Handbuchs keine Schwierigkeiten haben.

## 3.1 Einführung in OPC

### 3.1.1 OPC, was ist das?

OPC ist eine herstellerunabhängige Software-Schnittstelle, die einen Datenaustausch zwischen Hardware und Software auch von verschiedenen Herstellern ermöglicht.

#### Was kann OPC?

Einfach ausgedrückt "OPC bringt alles unter einen Hut".

Vor OPC war es ziemlich aufwändig, die Hardware verschiedener Hersteller mittels Software-Applikationen zu steuern. Es gab eine Vielzahl verschiedener Systeme und Protokolle. Für jeden Hersteller und jedes Protokoll musste ein Anwender eine spezielle Software zum Zugriff auf die spezifischen Schnittstellen und Treiber einsetzen. Damit waren die Anwenderprogramme abhängig vom Hersteller, Protokoll oder System. OPC auf Basis von COM oder DCOM hat als einheitliche und herstellerunabhängige Software-Schnittstelle den Datenaustausch in der Automatisierungstechnik revolutioniert.

Das folgende Bild gibt Ihnen einen Überblick über die Leistungsfähigkeit und die Flexibilität von OPC. Die einzelnen Komponenten werden Ihnen im weiteren immer wieder begegnen und auch dort erklärt werden.

### 3.1 Einführung in OPC

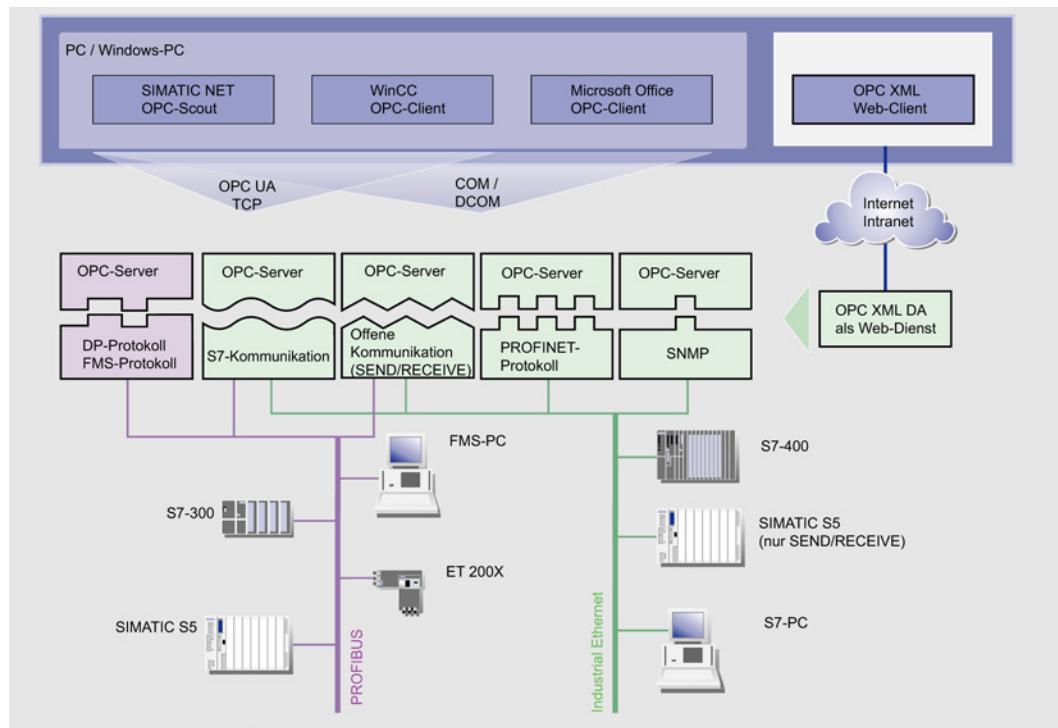


Bild 3-1 Systemintegration mit OPC-Server

### 3.1.2 Vorteile von OPC

Besondere Vorteile von OPC:

- Die Symbolik
- Der OPC-Scout
- Das SIMATIC NET OPC Data Control

#### Was bringt der Einsatz von Symbolik?

Auf alle Fälle einen einfachen symbolischen Zugriff auf Prozessvariablen von SIEMENS S7 Steuerungen.

Bei der STEP 7 Projektierung wird mit Symbolik gearbeitet. Diese dort bereits investierte Arbeit, zahlt sich hier aus. OPC benutzt dieselbe Symbolik wie STEP 7.

Sie können die bei der STEP 7-Projektierung festgelegten Symboldefinitionen auch auf OPC-Seite weiterverwenden.

Der Einsatz von Symbolen erlaubt eine flexible Programmierung von OPC-Clients.

Sie sparen Zeit, Geld, Arbeit und vor allem werden Fehlerquellen vermieden.

### Wie erleichtert der OPC-Scout die Arbeit?

Mit dem OPC-Scout von SIMATIC NET steht Ihnen ein leistungsfähiges Werkzeug zum einfachen Zugriff auf Prozessvariablen zur Verfügung.

Mit dem OPC-Scout können Sie eine OPC-Anwendung testen oder den OPC-Server in Betrieb nehmen.

Der OPC-Scout zeigt Ihnen den Namensraum der Variablen, bestehend aus Kommunikationsverbindungen und symbolischen Namen.

Mit dem OPC-Scout haben Sie über den OPC-Server Zugriff auf alle Prozessvariablen, die Sie über die konfigurierten Protokolle und Verbindungen erreichen können.

Mit dem OPC-Scout können Sie die Werte von Prozessvariablen beobachten, Werte lesen und Werte schreiben oder generieren.

Der OPC-Scout zeigt Ihnen unter anderem den Zustand der Kommunikationsverbindungen an. Dies kann mit Hilfe von Eigenschaften der Prozessvariablen oder über Informationsvariablen erfolgen. Auf diese Weise können Sie erkennen, wenn ein Partnergerät nicht erreichbar ist.

### Wie erleichtert SIMATIC NET OPC Data Control den Zugriff auf Daten?

Mit Data Control können Sie überall, wo Sie auf ActiveX-Controls zugreifen können, beispielsweise in Visual Basic, schnell einfache OPC-Clients erstellen. Ein komfortabler und einfacher Zugriff auf Prozessdaten ist möglich.

Das Data Control greift auf Prozessdaten zu, die vom OPC-Server ermittelt werden.

Die Anzeige-Controls sind Elemente zur Visualisierung von Prozessdaten. Sie erhalten Ihre Daten über das SIMATIC NET OPC Data Control und nicht direkt durch Zugriff auf OPC oder eine andere Schnittstelle.

Damit Sie Prozessdaten anzeigen oder eingeben können, müssen Sie über das SIMATIC NET OPC Data Control die Eigenschaft eines ActiveX-Controls mit einer Eigenschaft einer OPC-Variablen verknüpfen.

Eigenschaften eines OPC-Items sind Wert, Qualität und Zeitstempel. Zum Beispiel kann in einfacher Weise eine Hintergrundfarbe mit der Qualität einer OPC-Variablen verschaltet werden. Den Qualitätszuständen GOOD, BAD und UNCERTAIN werden beispielsweise die Farbcodes Grün, Rot und Gelb zugewiesen.

Die Hintergrundfarbe des Number Controls ändert sich, wenn sich die Qualität der Prozessvariablen z.B. durch Leitungsbruch ändert.

### 3.1.3

### OPC-Schnittstelle, was leistet sie?

Die OPC-Schnittstelle ist ein Teil der Software, die auf einem PC als Plattform für Bedien- und Beobachtungssysteme oder andere Anwendungen läuft. Sie liegt unterhalb des jeweiligen Anwendungsprogramms.

Als industrieller Standard definiert OPC den Informationsaustausch für verschiedene Anwendungsfälle im industriellen Umfeld.

### **Auf welchem Prinzip basiert die OPC-Schnittstelle?**

Die Anwendungen der OPC-Schnittstelle basieren auf dem Client-Server-Modell. Eine Komponente stellt über Schnittstellen als Server ihre Dienste Anderen zur Verfügung. Eine andere Komponente nimmt die Dienste als Client in Anspruch. Eine Anwendung kann feststellen, welche OPC-Server in einem System eingerichtet sind. Sie kann einen oder mehrere dieser Server ansprechen und überprüfen, welche Dienste von dem Server bereitgestellt werden. Da mehrere unterschiedliche OPC-Clients gleichzeitig auf denselben OPC-Server zugreifen können, ist dieselbe Datenquelle für beliebige OPC-konforme Anwendungen nutzbar.

Hersteller von Baugruppen, die Prozessdaten liefern (Kommunikationssysteme, Messgeräte etc.), stellen für ihre Baugruppe einen OPC-Server zur Verfügung, der die Anbindung an die jeweilige Datenquelle übernimmt.

### **Diese Aufgaben übernimmt OPC**

Sie können über die OPC-Schnittstelle vom PC aus Systemdaten und Ereignisse der Automatisierungssysteme überwachen, abrufen und verarbeiten.

### **Was umfasst die OPC-Schnittstelle?**

Seit 1996 erstellt die OPC-Foundation Spezifikationen für die OPC-Schnittstelle. Aktuell gibt es für die Automatisierungstechnik folgende Spezifikationen:

- für den Datenaustausch auf Basis von Prozessvariablen: Data Access
- für die Behandlung von Alarmen und Ereignissen: Alarms & Events
- für den Datenaustausch auch über das Internet: Data Access XML
- für den horizontalen Datenaustausch zwischen OPC-Servern: Data Exchange
- für die Behandlung von Rezepturen: Batch
- für den Zugriff auf archivierte Daten: Historical Data Access
- für die Zusammenfassung vieler OPC-Spezifikationen: OPC Unified Architecture

Als Schnittstelle zu den Systemen der industriellen Kommunikation bietet der SIMATIC NET OPC-Server die Funktionalität von Data Access, Alarms & Events, Data Access XML und Unified Architecture.

#### **3.1.4 OPC-Server, was ist das?**

Die OPC-Schnittstelle basiert auf dem Prinzip der Zusammenarbeit zwischen initierendem Prozess (stellt Anfragen, erteilt Aufträge) und reagierendem Prozess (bearbeitet Anfragen und Aufträge) – Client und Server.

## OPC-Server

OPC-Komponenten, die Daten liefern, heißen OPC-Server. Sie realisieren die Anbindung an bestehende Kommunikationssysteme. Neben Diensten stellen sie für den OPC-Client Informationen aus beliebigen Datenquellen bereit; das können Hardware-getriebene Datenquellen sein oder Software-Komponenten. Die jeweiligen Daten werden z. B. von Schnittstellen, Feldbuskarten, Messgeräten oder Reglern erfasst. Jeder OPC-Server hat einen eindeutigen Namen zur Identifizierung.

### Woher kommen die Server-Namen (ProgID)?

Jeder OPC-Server erhält vom Hersteller einen eindeutigen Namen zur Identifizierung. Gemäß dem COM-Standard heißen diese Namen ProgIDs. Durch Angabe der ProgIDs können Sie einzelne OPC-Server gezielt ansprechen.

### Welche Server-Typen gibt es?

Beim OPC-Server werden drei Server-Typen unterschieden. Abhängig davon, wie sie in das Kommunikationssystem eingebunden sind, werden sie folgendermaßen bezeichnet:

- Local Server (Out-Process-Server)  
(dieser Server befindet sich auf dem lokalen Rechner)
- Remote Server (Out-Process-Server)  
(dieser Server befindet sich auf einem anderen Rechner im Netzwerk)
- In-Process-Server  
(dieser Server ermöglicht eine höhere Performance)

Der Anbieter eines OPC-Servers legt fest, ob der Server ein In-Process-Server oder ein Local Server ist. Der Betrieb als Remote Server wird vom Anwender konfiguriert.

Die Syntax der Methodenaufrufe ist für alle drei Server-Typen gleich.

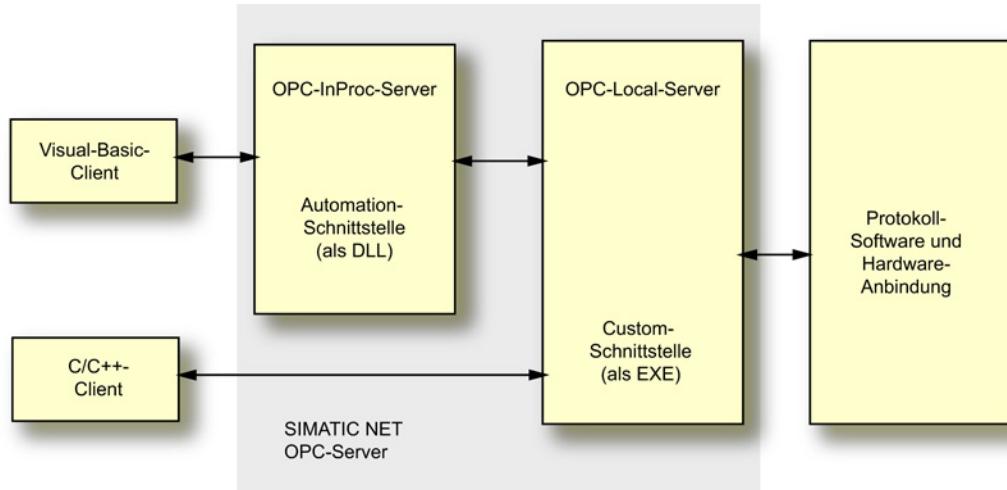


Bild 3-2 OPC-Server

### **3.1.5        OPC-Client, was ist das?**

Die OPC-Schnittstelle basiert auf dem Prinzip der Zusammenarbeit zwischen initierendem Prozess (stellt Anfragen, erteilt Aufträge) und reagierendem Prozess (bearbeitet Anfragen und Aufträge) – Client und Server.

#### **Das ist ein OPC-Client**

OPC-Komponenten, die einen OPC-Server als Datenquelle nutzen, heißen OPC-Clients.

#### **OPC-Clients kaufen?**

OPC-Clients sind als Standard-Software erhältlich. Auch Software-Module werden angeboten, die Sie nach eigenem Bedarf zu einem funktionsfähigen Client zusammenbauen können.

#### **OPC-Clients selber machen?**

Um die individuellen Anforderungen Ihres Systems bestmöglichst zu erfüllen und eine größtmögliche Performance zu erzielen, können Sie eigene OPC-Clients in verschiedenen Programmiersprachen (z. B. Visual Basic, C, C++ und C#) schreiben.

#### **Welche Eigenschaften sind zu beachten?**

Einige Eigenschaften von OPC-Servern (z.B. Variablennamen) sind durch den OPC-Standard nicht definiert, sondern hängen z.B. von den Eigenschaften des Automatisierungssystems oder der Anlage ab und werden durch den Hersteller festgelegt. Damit OPC-Clients mühelos auf verschiedene OPC-Server anwendbar sind, sollte bei der Programmierung eine flexible Auswahl der Variablen oder Symbolik vorgesehen werden. So ist eine Applikation vielfach nutzbar und wiederverwendbar.

### **3.1.6        Server und Client, wie arbeiten sie zusammen?**

#### **Server und Client arbeiten zusammen**

Server und Client kommunizieren auf Basis von COM oder DCOM. Dabei greift der Client nicht direkt auf einen Server zu, sondern mit Hilfe der COM-Bibliothek. Durch Angabe der ProgID kann der OPC-Client jeden gewünschten OPC-Server direkt ansprechen.

Das Client-Programm merkt keinen Unterschied, ob der Zugriff über COM oder DCOM stattfindet.

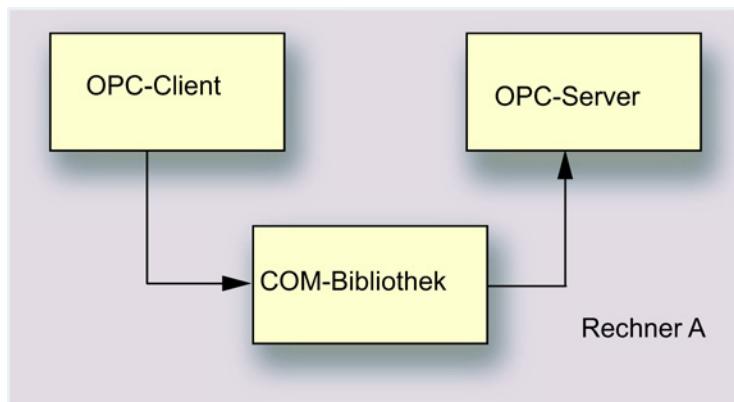


Bild 3-3 COM auf dem lokalen Rechner

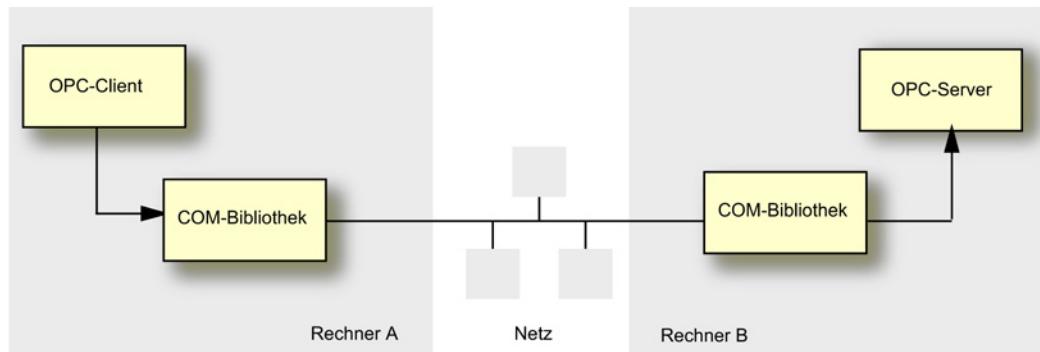


Bild 3-4 COM auf dem remoten Rechner

### Welche Eigenschaften und Methoden werden genutzt?

Der Leistungsumfang von OPC-Servern ist durch ihre Schnittstellen festgelegt. Deshalb kennt der OPC-Client die zu erwartenden Server-Leistungen und kann die angebotenen Dienste gezielt nutzen. Im Sinne der Objektorientierung werden die Dienste der OPC-Server durch Eigenschaften und Methoden repräsentiert. Alle OPC-Server verfügen über einen Stamm von gleichen Eigenschaften und Methoden. Darüber hinaus sind einige Schnittstellen in den OPC-Spezifikationen als Optional gekennzeichnet. Falls ein Server diese optionalen Funktionalitäten nicht anbietet, kann ein Client dies erkennen und angemessen darauf reagieren. Dadurch können Komponenten unterschiedlicher Hersteller problemlos zusammenarbeiten.

Über die OPC-Schnittstellen kann ein Client Objekte im Server erzeugen, verwenden und löschen. Der OPC-Client greift auf Server-Funktionen zurück und verwendet die Methoden des Servers z.B. zum Lesen und Schreiben von Daten. Jede Server-Funktion korrespondiert mit einem Aufruf im Client.

### **3.1.7      Grundbegriffe**

#### **3.1.7.1    COM-Objekte, was ist das?**

Für eine effektivere Zusammenarbeit von Client und Server besteht die Möglichkeit, gleichartige Aufgaben zusammenzufassen oder zu spezifizieren. COM-Objekte machen dies möglich.

#### **Was ist ein COM-Objekt?**

COM-Objekte sind unter Windows ablauffähige Komponenten, die anderen Komponenten über ihre Schnittstellen eine definierte Funktionalität anbieten. Ein COM-Objekt kann gleichzeitig von mehreren Anwendungen verwendet werden.

#### **Was ist COM?**

COM ist zentraler Bestandteil der Windows Betriebssysteme und regelt die Zusammenarbeit mehrerer Software-Komponenten.

Durch die Nutzung von COM wird der OPC-Server wie ein Teil des Windows-Betriebssystems und damit unabhängig von Dateinamen, Ablageorten und Versionen.

Die Grundlage der OPC-Mechanismen ist COM, das Component Object Model von Microsoft.

COM definiert einen Standard, der es ermöglicht, Objekte als abgeschlossene Einheiten in Windows zu definieren und über Prozessengrenzen hinweg auf diese zuzugreifen.

COM-Objekte können als Erweiterungen des Betriebssystems verstanden werden. Sie sind unabhängig von Programmiersprachen und stehen prinzipiell allen Applikationen zur Verfügung.

Die Daten und der Code des Objekts sind für den Anwender des COM-Objekts nicht direkt zugänglich.

#### **Was ist DCOM?**

DCOM bedeutet Distributed Component Object Model. Als Weiterentwicklung von COM unterstützt DCOM verteilte Anwendungen und ermöglicht die rechnerübergreifende Zusammenarbeit von Software-Komponenten innerhalb eines Netzes.

## Struktur von COM-Objekten

Das nachfolgende Bild veranschaulicht die Struktur eines COM-Objekts mit 4 Schnittstellen. Der Zugriff auf das Objekt erfolgt nur über die Schnittstellen. Er ist über verschiedene Methoden geregelt. Auf das eigentliche Objekt als Ganzes, die darin enthaltenen Daten oder den Code, gibt es keine Zugriffsmöglichkeit.

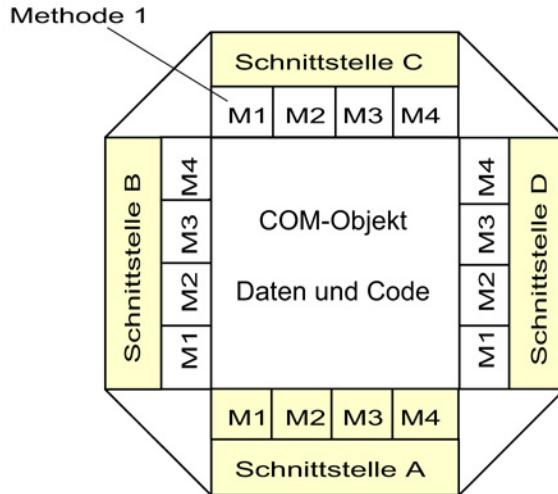


Bild 3-5 Struktur eines COM-Objekts

### 3.1.7.2 COM-Objekte, wie werden Sie dargestellt?

#### Darstellung von COM-Objekten

In der Dokumentation werden COM-Objekte meistens grafisch dargestellt. Die objektspezifischen Schnittstellen werden seitlich des Objekts dargestellt, die mit allen Objekten gelieferte Schnittstelle `IUnknown` an der Oberkante des Objekts.

Die Methoden, die hinter den Schnittstellen stehen, werden durch die Schnittstellen verborgen.

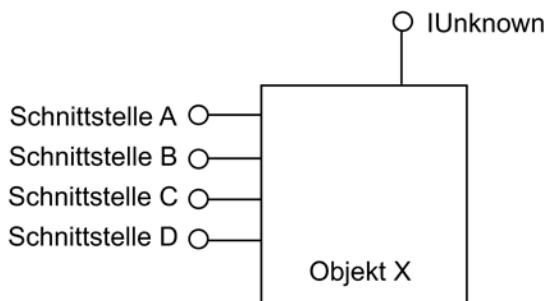


Bild 3-6 Darstellung eines COM-Objekts

### 3.1.7.3 COM-Schnittstellen, was leisten sie?

#### Das leisten COM-Schnittstellen

Eine COM-Schnittstelle ist ein definierter, üblicherweise zusammengehörender Satz von Methoden zum Aufruf der Funktionalität des COM-Objekts. Sie besteht aus einer Tabelle von Zeigern, die auf die Methoden verweisen. Eine COM-Schnittstelle kapselt die Funktionalität des COM-Objekts und stellt sicher, dass nur in definierter Weise auf das Objekt zugegriffen werden kann. COM-Schnittstellen sind mit einer eindeutigen Kennung versehen, so dass eine Applikation, die auf das COM-Objekt zugreifen will, vor dem Zugriff prüfen kann, ob das Objekt die Schnittstelle unterstützt.

#### So sind Schnittstellen aufgebaut

Das Bild zeigt den grundsätzlichen Aufbau einer Schnittstelle.

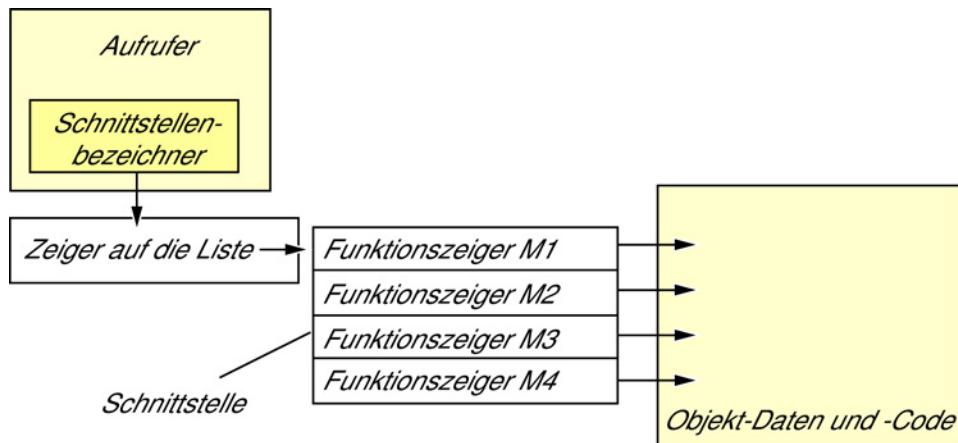


Bild 3-7 Struktur einer Schnittstelle

### 3.1.7.4 COM-Schnittstellenarten, welche gibt es, wie wird darauf zugegriffen?

#### Diese Schnittstellenarten gibt es

COM unterscheidet zwei Schnittstellenarten:

- Automation-Schnittstelle
- Custom-Schnittstelle

Die Schnittstellen unterscheiden sich im internen Methodenaufruf. Für jede Schnittstelle gibt es eigene Schnittstellenspezifikationen. Unabhängig davon eignen sie sich jedoch gleichermaßen für die verschiedensten Anwendungsfälle wie z.B. Variablenzugriff oder Meldungsempfang.

Die Automation-Schnittstelle unterstützt Client-Anwendungen, die auf einer Skriptsprache, wie Visual Basic oder VBA basieren.

Die Custom-Schnittstelle erhöht die Performance der Anwendungen, die auf C oder C++ basieren.

Die Custom-Schnittstelle ist für das Leistungsspektrum von Entwicklungswerkzeugen, die auf Skriptsprachen basieren, nicht geeignet. Durch die Erweiterung der COM-Objekte durch die Automation-Schnittstelle werden die Methoden der Objekte auch für einfache Skriptsprachen zugänglich. Die Automation-Schnittstelle macht die Aufrufe nach außen hinsichtbar, die das Objekt versteht.

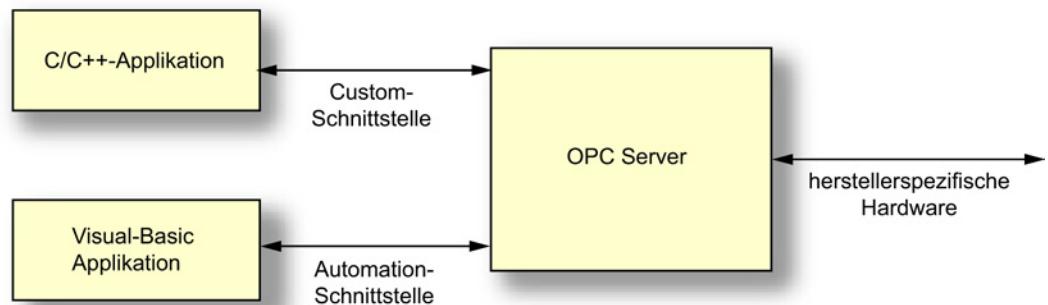


Bild 3-8 Beispiel für die Zuordnung zwischen Schnittstellen und Applikation

### Wie kann ein .NET-Client auf die COM-Schnittstelle zugreifen?

Im nachfolgenden werden die Abläufe bei Verwendung der Custom-Schnittstelle und der Automation-Schnittstelle beschrieben.

#### Ablauf bei Verwendung der OPC Custom-Schnittstelle

Ein .NET-Client kann aus verwaltetem Code heraus auf ein allgemeines COM-Objekt der Custom-Schnittstelle zugreifen. Wegen der unterschiedlichen Eigenschaften von COM und dem .NET-Programmiermodell (in .NET gibt es z.B. keine Zeigerzugriffe) ist allerdings kein direkter Aufruf möglich.

Beim Übergang von verwaltetem Code zu unverwaltetem Code muss ein RCW (Runtime callable Wrapper) benutzt werden. RCWs vermitteln zwischen verwalteten .NET-Objekten und nicht verwalteten COM-Objekten.

### 3.2 Data Access

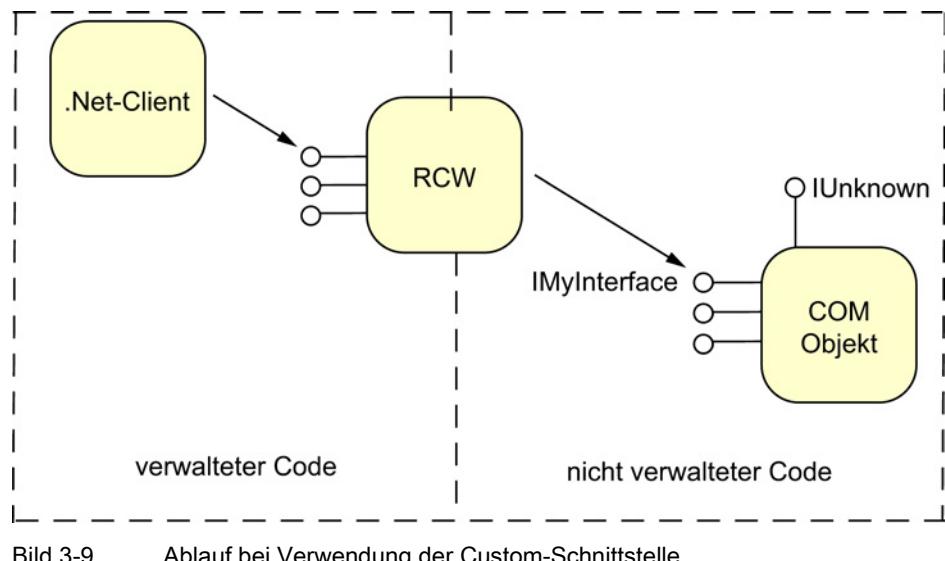


Bild 3-9 Ablauf bei Verwendung der Custom-Schnittstelle

#### Ablauf bei Verwendung der OPC Automation-Schnittstelle

Mit Hilfe einer .NET Framework-Importanwendung wird eine sogenannte Interop-Assembly, eine .NET-Komponente, erstellt. .NET-Clients können damit Instanzen von COM-Objekten erstellen und Methoden von COM-Objekten aufrufen, als würde es sich um eine .NET-Instanz handeln. Nicht verwalteter Code, die Automation Schnittstelle, wird also in eine .NET-Komponente umgesetzt.

## 3.2 Data Access

### 3.2.1 Einführung in die Data-Access-Schnittstelle

#### 3.2.1.1 Was kann OPC Data Access?

Die Data Access Schnittstelle ist ein weltweiter herstellerunabhängiger Standard zum Lesen, Schreiben und Beobachten von Prozessdaten. Die Kommunikation basiert auf dem Microsoft COM Protokoll. Bei Anwendern wie auch Herstellern hat sich dieser Standard gleichermaßen durchgesetzt. Die Anwenderprogramme sind einfache Office-Applikationen bis hin zu anspruchsvollen HMI- (Human Machine Interface) oder SCADA- (Supervisory Control and Data Acquisition) Systemen.

## Das kann OPC Data Access

Die OPC-Data-Access-Spezifikation definiert die Schnittstelle zwischen Client- und Server-Programmen zur Prozessdatenkommunikation. Dabei ermöglichen Data-Access-Server einem oder mehreren Data-Access-Clients den transparenten Zugriff auf die verschiedensten Datenquellen (z.B. Temperatursensor) und Datensenken (z.B. Regler). Diese Datenquellen und -senken können sich auf direkt im PC gesteckten I/O Karten befinden, sie können aber auch auf beliebigen Geräten wie Reglern, Ein-/Ausgabemodulen u. a. liegen, die über serielle Verbindungen oder über Feldbusse angeschlossen sind. Selbstverständlich kann ein Data-Access-Client auch gleichzeitig auf mehrere Data-Access-Server zugreifen.

## Was sind Data-Access-Clients?

Data-Access-Clients können sehr simple Excel-Sheets oder umfangreiche Programme (z.B. Visual Basic) sein. Data-Access-Clients können aber auch wieder Bestandteil größerer Programme sein.

## Was ist ein Data-Access-Server?

Data-Access-Servers können einfache Programme sein, die z.B. den Zugriff auf die Register einer SPS über eine serielle Schnittstelle zur Verfügung stellen. Es sind auch komplexere Programme möglich, die den Zugriff auf eine Vielzahl von Variablen in einer großen Anzahl von Geräten über umfangreiche Kommunikationsmechanismen ermöglichen. Data-Access-Servers können auch Bestandteil größerer Programme sein und Daten dieser Programme verfügbar machen.

### 3.2.1.2 OPC Data Access, was ist das?

#### So können Sie mit OPC Data Access auf Prozessvariablen zugreifen

Data Access ist eine OPC-Spezifikation zum Zugriff auf Prozessdaten über Variablen. Ein OPC-Server für Data Access verwaltet die Prozessvariablen und die verschiedenen Zugriffsmöglichkeiten auf diese Variablen. Dadurch kann er:

- den Wert einer oder mehrerer Prozessvariablen lesen
- den Wert einer oder mehrerer Prozessvariablen ändern, indem er einen neuen Wert schreibt
- den Wert einer oder mehrerer Prozessvariablen überwachen
- Werteänderungen melden.

Prozessvariablen sind Platzhalter für Werte, die aktuell ermittelt werden müssen.

### 3.2.1.3 Klassenmodell von OPC Data Access, was leistet es?

#### Das leistet das OPC-Data-Access-Klassenmodell

Das hierarchische Klassenmodell von Data Access hilft beim Datenzugriff durch den Client, Zeitaufwand und inhaltliches Ergebnis den aktuellen Anforderungen einer Applikation anzupassen. Data Access unterscheidet drei Klassen:

- OPC-Server
- OPC-Group
- OPC-Item

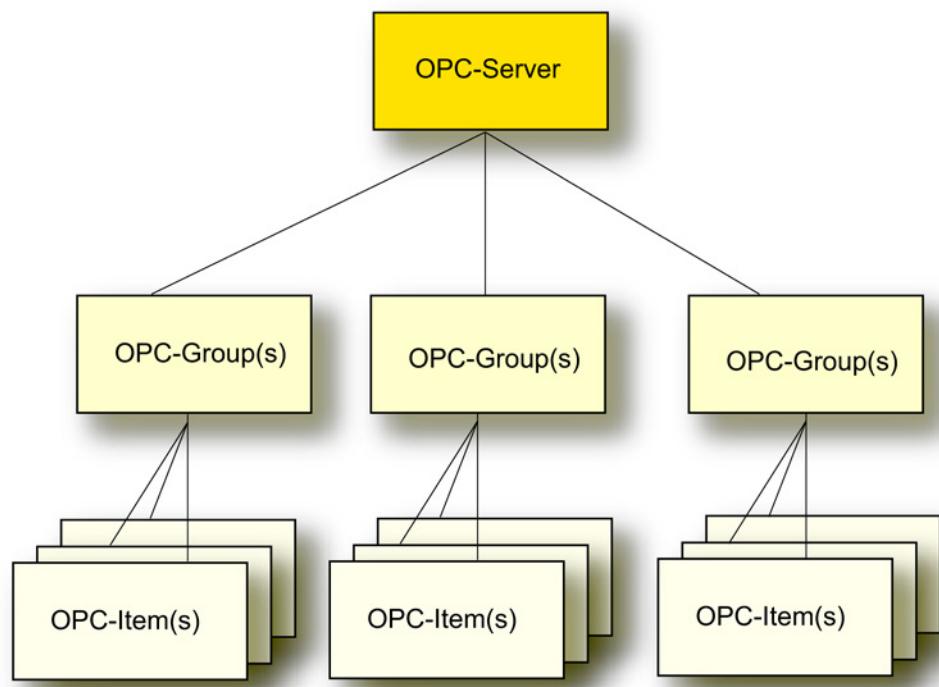


Bild 3-10 Klassenmodell der Data-Access-Schnittstelle

Nur zur Erzeugung eines Objekts der Klasse OPC-Server verwendet die Client-Applikation COM-Aufrufe des Betriebssystems. Die anderen Objekte werden durch entsprechende OPC-Methoden der Klasse OPC-Server oder untergeordneten Klassen erzeugt.

#### Für wen gilt das Klassenmodell?

Das Klassenmodell gilt sowohl für die Automation-Schnittstelle, als auch für die Custom-Schnittstelle.

### 3.2.1.4 Klasse OPC-Server, was leistet sie?

#### Das leistet die Klasse OPC-Server

An oberster Stelle steht die Klasse OPC-Server. Jeder OPC-Server gehört dieser Klasse an. Diese Klasse stellt den Zugang für alle weiteren Dienste des Data-Access-Servers dar.

Mit Hilfe klassenspezifischer Attribute und Methoden können Sie Informationen über Status, Version und (optional) den Namensraum der verfügbaren Prozessvariablen erhalten. Ein Objekt der Klasse OPC-Server verwaltet die Instanzen der untergeordneten Klasse OPC-Group.

### 3.2.1.5 Klasse OPC-Group, was leistet sie?

#### Das leistet die Klasse OPC-Group

Die Klasse OPC-Group ist der Klasse OPC-Server direkt untergeordnet und strukturiert die vom OPC-Server genutzten Prozessvariablen. Ein OPC-Client kann mehrere Objekte dieser Klasse gleichzeitig benutzen. Mit Hilfe der Objekte von OPC-Group kann ein Client sinnvolle Einheiten von Prozessvariablen bilden und mit diesen Operationen ausführen. So können beispielsweise alle Prozessvariablen einer Bildschirmseite eines Bedien- und Beobachtungssystems in einer Gruppe zusammengefasst werden.

Die Klasse OPC-Group definiert Methoden, über die die Werte der Prozessvariablen gelesen und geschrieben werden können.

Bei einigen Methoden (Lesen und Schreiben von OPC-Items) können mehrere Variablen in einem Auftrag zusammengefasst und gleichzeitig übergeben werden. In vielen Fällen kann der OPC Server zusätzlich eine interne Optimierung durchführen. Insbesondere bei der Benutzung eines OPC Server über das Netz ermöglichen diese Mengenoperationen eine hohe Ausführungsgeschwindigkeit. Umgekehrt ist eine Performanceeinbuße bei vielen Einzelaufträgen in kurzen Zeitabständen zu erwarten.

Ab Data Access Spezifikation 3.00 kann über die Klasse OPC-Group eine zyklische Lebenszeichens-Überwachung des OPC-Server (KeepAliveTime) eingestellt werden. Auch wenn sich die Prozessvariablen nicht ändern, wird eine Rückmeldefunktion (ohne Datenwerte) im OPC-Client vom OPC-Server aufgerufen.

### **3.2.1.6 Klasse OPC-Item, was leistet sie?**

#### **Das leistet die Klasse OPC-Item**

Objekte dieser Klasse repräsentieren die eigentlichen Prozessvariablen und ermöglichen eine gezielte Abfrage einzelner Daten. Jede Variable ist ein Element (Item) im Namensraum des OPC-Servers und wird durch eine Item-ID identifiziert. Die Item-ID wird vom Hersteller des Servers festgelegt und muss innerhalb des Namensraums des Servers eindeutig sein. Mit jedem Item sind folgende Eigenschaften verbunden:

- Wert  
Zuletzt erfasster Wert der Variable.
- Qualität  
Aussagekraft der Wertes. Wenn die Qualität gut ist, konnte der Wert sicher ermittelt werden.
- Zeitstempel  
Zeitpunkt, an dem der aktuelle Wert der Variablen ermittelt wurde. Mit jeder zum Client gemeldeten Werteänderung wird auch der Zeitstempel aktualisiert. Ändert sich der Wert einer Variablen nicht, bleibt auch der Zeitstempel gleich.

#### **Welche Rolle spielen die Variablen?**

Variablen müssen bei den Aufrufen der OPC-Schnittstelle angegeben werden, um Prozesswerte zu erhalten. Durch die Angabe von Variablen kann der Client beim Server die benötigten Werte anfordern. Der Client muss jede gewünschte Variable beim Server anmelden, um festzulegen, welche Variablen gelesen werden sollen. Variablen können sowohl synchron als auch asynchron gelesen und geschrieben werden.

Der Client kann die Beobachtung von Variablen auf den Server übertragen. Wenn sich der Wert einer Variablen ändert, schickt der Server dem Client eine entsprechende Nachricht.

Die vom OPC-Server angebotenen Variablen lassen sich unterteilen in:

- Prozessvariablen  
Repräsentieren Mess- und Steuergrößen von Ein-/Ausgabegeräten oder
- Steuervariablen  
Die Verwendung dieser Variablen löst bestimmte Zusatzdienste aus, z.B. die Übertragung von Passwörtern.  
oder
- Informationsvariablen  
Diese Variablen werden vom Kommunikationssystem und vom OPC-Server bereitgestellt und geben Auskunft über den Zustand von Verbindungen, Geräten usw.

Hier einige Beispiele für Variablen eines OPC Data Access Servers:

- Steuerungsgrößen einer speicherprogrammierbaren Steuerung
- Daten eines Messdatenerfassungssystems
- Statusvariablen des Kommunikationssystems

### 3.2.1.7 OPC Data Access, welche Schnittstellenspezifikationen gibt es?

**Es gibt zwei Schnittstellenspezifikationen für OPC Data Access**

Für Data Access sind Automation- und Custom-Schnittstelle spezifiziert:

- Data Access Automation Interface, Standard, February 4, 1999, Version 2.02 (und Folgeversionen)
- Data Access Custom Interface, Standard, March 4, 2003, Version 3.00

Eine Übersicht der Spezifikationen finden Sie im Literaturverzeichnis von Band 2.

## 3.3 OPC Alarms & Events

### 3.3.1 Einführung in OPC Alarms & Events

#### 3.3.1.1 OPC Alarms & Events, was ist das?

**Das ist Alarms & Events**

Alarms & Events ist eine Spezifikation zur Übertragung von Prozessalarmen und Ereignissen. Sie ist sehr flexibel gestaltet und kann deshalb auf unterschiedlichste Ereignisquellen angewendet werden. Das Spektrum reicht von einfachen Ereignissen bis hin zu komplexen Ereignissen und sogar quittierpflichtigen Ereignissen.

Die OPC-Spezifikation definiert die möglichen Zustandsübergänge für bedingte Ereignisse in einem Zustandsdiagramm.

**Wozu dient Alarms & Events?**

Alarms-&-Events-Server dienen zum Beispiel zum

- Erkennen von Ereignissen – z.B. Kesselfüllung erreicht,
- Feststellen des Zustandes eines Ereignisses – Kessel voll,
- Bestätigen eines Ereignisses – Erreichen der Kesselfüllung erkannt,
- Überwachen der Bestätigung – die Bestätigung wird vom Kessel-Alarmmelder überwacht, der Alarm wurde erkannt, das Warnsignal kann ausgeschaltet werden.

Neue Ereignisse können auch ohne Bestätigung gemeldet werden.

Die standardisierte OPC-Alarms-&-Events-Schnittstelle erlaubt die Hantierung dieser Anforderungen.

### **3.3.1.2 Ereignisse und Ereignismeldungen, was ist das?**

#### **Das sind Ereignisse**

Ereignisse sind besondere Zustände im Prozess, die an einen Empfänger gemeldet werden müssen. Welche Ereignisse an den OPC-Client gemeldet werden, wird vom OPC-Client über Filterkriterien eingestellt.

Alle Ereignisse, die den eingestellten Filterkriterien entsprechen, müssen vom Erzeuger des Ereignisses bis zum Anwender geleitet werden. Damit unterscheidet sich Alarms & Events von Data Access. Bei der Beobachtung von Variablen werden nur die im angegebenen Zeitraster liegenden Werteänderungen mitgeteilt.

#### **Das sind Ereignismeldungen**

Bei der Meldung werden die gemäß OPC-Spezifikation definierten Parameter und gegebenenfalls herstellerseitig festgelegte Begleitwerte geliefert.

Es gibt einfache Ereignismeldungen und komplexere zustandsgebundene Meldungen. Für diese komplexen zustandsgebundenen Meldungen kann der Ereignissender eine Quittierung durch den OPC-Client verlangen.

#### **Ereignisarten**

Die OPC-Spezifikation definiert drei Arten von Ereignissen:

- Bedingte Ereignisse (Condition related Events)  
Sie melden die im OPC-Zustandsmodell definierten Zustandsübergänge und sind an definierte Bedingungen gebunden.
- Protokollierereignisse (Tracking Events)  
Sie melden Veränderungen des Prozesses, wenn beispielsweise ein Anwender den Sollwert eines Reglers ändert.
- Einfache Ereignisse (Simple Events)  
Sie melden alle übrigen zustandslosen Ereignisse, beispielsweise den Ausfall einer Systemkomponente.

Die OPC-Spezifikation definiert die Syntax der Schnittstelle zum Meldungsempfang. Welche Ereignisarten ein Server liefert, ist durch den Hersteller des OPC-Servers festgelegt.

### **3.3.1.3 Klassenmodell von OPC Alarms & Events, was leistet es?**

#### **Das leistet das Klassenmodell von OPC Alarms & Events**

Das Klassenmodell von Alarms & Events ermöglicht die Anpassung des OPC-Clients an die Anforderungen einer Automatisierungslösung. Alarms & Events unterscheidet drei Klassen:

- OPC-Event-Server
- OPC-Event-Subscription
- OPC-Event-Area-Browser

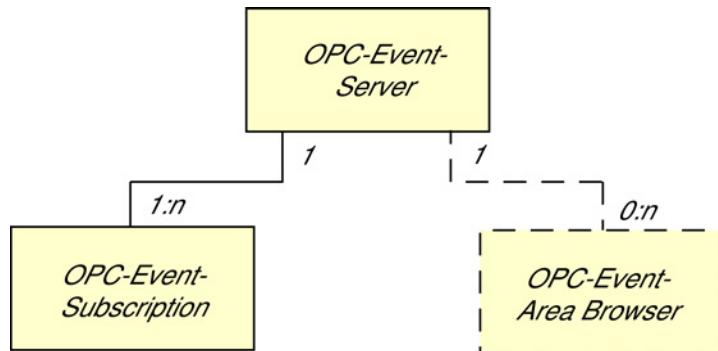


Bild 3-11 Klassenmodell der Alarms &amp; Events-Schnittstelle

### 3.3.1.4 Klasse OPC-Event-Server, was leistet sie?

#### Das leistet die Klasse OPC-Event-Server

Über Objekte der Klasse OPC-Event-Server legt ein Client ein oder mehrere Objekte der Klasse OPC-Event-Subscription an. Ein Objekt dieser Klasse ist ein Abonnement einer Menge von Ereignissen. Objekte dieser Klasse verwalten Client-spezifisch die benötigten Filter und Attribute. Durch die Filterung kann ein Client festlegen, welche Ereignisse er empfangen will. Die Methode SelectReturnedAttributes ermöglicht festzulegen, welche Ereignisattribute mit jeder Ereignismeldung übermittelt werden sollen. Mit Hilfe von Objekten der Klasse OPC-Event-Subscription kann der Client sinnvolle Gruppen bilden und Mengenoperationen durchführen.

#### Quittierung von Ereignissen

Mit der Methode AckCondition der Klasse OPCEventServer quittiert der Client condition-related Events, wenn das im Parameter AckRequired des Ereignisses so festgelegt ist. Sobald die Quittierung eintrifft, führt das zu einer Änderung des Parameters NewState des condition-related Events und somit zu einem neuen Ereignis.

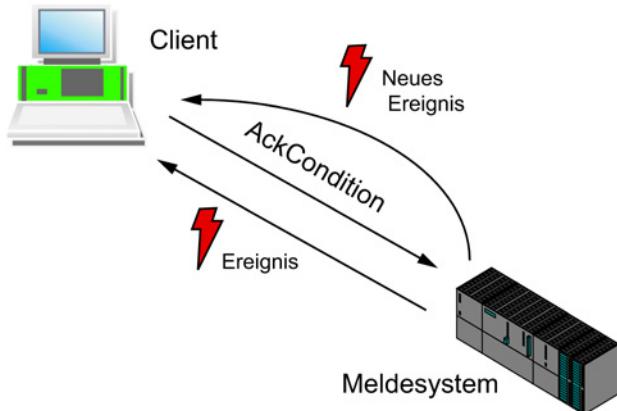


Bild 3-12 Abfolge von Ereignis und Quittierung bei condition-related Events

### 3.3.1.5 Klasse OPC-Event-Subscription, was leistet sie?

#### Das leistet die Klasse OPC-Event-Subscription

Über Objekte der Klasse OPC-Event-Server legt ein Client ein oder mehrere Objekte der Klasse OPCEventSubscription an. Ein Objekt dieser Klasse ist ein Abonnement einer Menge von Ereignissen. Objekte dieser Klasse verwalten client-spezifisch die benötigten Filter und Attribute. Durch die Filterung kann ein Client festlegen, welche Ereignisse er empfangen will. Es kann festgelegt werden, welche Ereignisattribute mit jeder Ereignismeldung übermittelt werden sollen. Mit Hilfe von Objekten der Klasse OPC-Event-Subscription kann der Client sinnvolle Gruppen bilden und Mengenoperationen durchführen.

#### Welche Filtermöglichkeiten von Ereignissen gibt es?

Durch die Filterung kann ein Client bestimmen, welche Ereignisse er empfangen will. Ein Filter ist nichts anderes, als die Definition eines Ereignisses anhand seiner Eigenschaften. Dabei werden folgende Kriterien zugrunde gelegt:

- Eventtyp
- Kategorie
- Priorität
- Ereignisquelle

Ein Ereignis wird nur dann an den Client weitergeleitet, wenn es in allen Kriterien den Filterwerten entspricht.

#### Warum werden Ereignisse gepuffert?

Wenn jedes Event einzeln zum Client übertragen wird, sind höhere Ressourcen erforderlich, als wenn für die Übertragung mehrere Events zusammengefasst werden. Mit dem Parameter BufferTime kann der Client festlegen, dass Ereignisse nur nach einem bestimmten Zeitintervall gesendet werden. Zwischenzeitlich auftretende Ereignisse werden bis zum nächsten Sendezeitpunkt gepuffert.

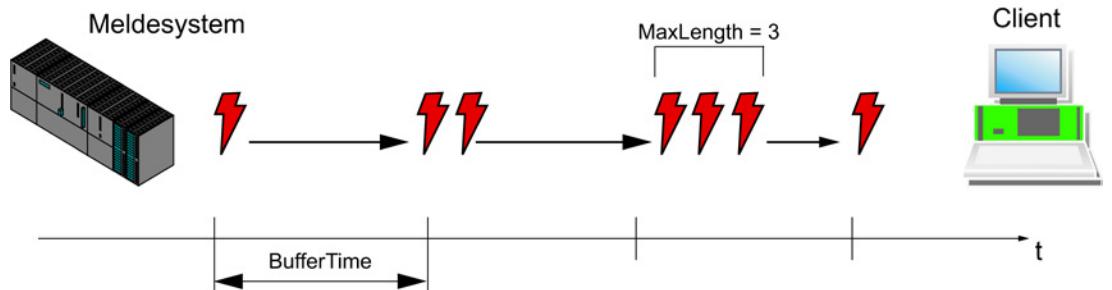


Bild 3-13 Bedeutung der Parameter BufferTime und MaxSize

Die Anzahl der maximal zu puffernden Ereignisse kann mit dem Parameter MaxSize bestimmt werden. Sobald die festgelegte Anzahl erreicht ist, werden alle Ereignisse an den Client gesendet, unabhängig vom gewählten Intervall BufferTime.

BufferTime und MaxSize werden als Parameter in der Methode CreateEventSubscription der Klasse OPCEventServer und in den Methoden GetState und SetState der Klasse OPCEventSubscription verwendet.

### 3.3.1.6 Klasse OPC-Event-Area-Browser, was leistet sie?

#### Das leistet die Klasse OPC-Event-Area-Browser

Mit OPC Alarms & Events können Sie umfassende Anlagen in Anlagenbereiche (Areas) einteilen. Areas können zum Filtern von Ereignissen verwendet werden. Mit Hilfe von Objekten der Klasse OPC-Event-Area-Browser können die Anlagenbereiche untersucht werden.

---

#### Hinweis

Objekte der Klasse OPC-Event-Area-Browser sind optional und werden vom OPC-Alarms-& Events-Server von SIMATIC NET nicht unterstützt.

---

### 3.3.1.7 Meldungsempfang, wie funktioniert er?

#### So funktioniert der Meldungsempfang

Eine Applikation meldet sich in vier Schritten zum Empfang von Meldungen an:

#### Ablauf

- Der Client meldet sich beim Server für den Meldungsempfang an.
- Der Client legt ein oder mehrere Objekte der Klasse OPCEventSubscription an.

### 3.3 OPC Alarms & Events

- Der Client richtet einen Callback über das Interface IConnectionPointContainer ein.
- Der Client stellt eine OnEvent-Methode zur Verfügung, die vom Server bei Events aufgerufen wird.

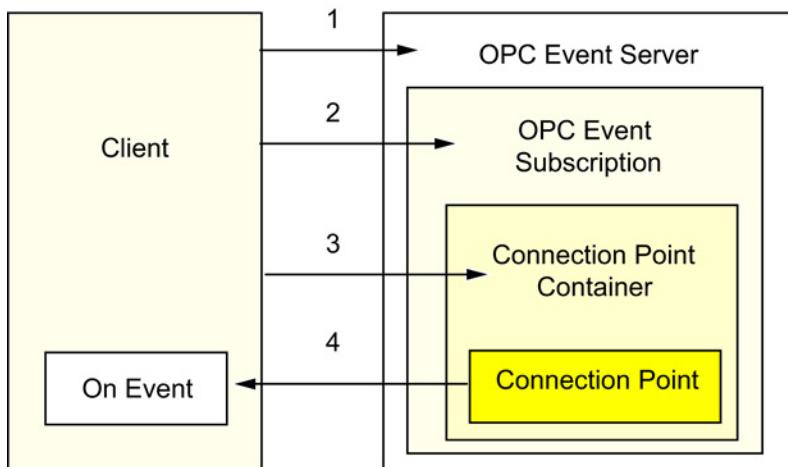


Bild 3-14 Verbindung zwischen Server und Client beim Empfang von Meldungen

#### 3.3.1.8 Meldungen bei SIMATIC S7, wie sind sie definiert?

##### So ist eine Meldung definiert

Eine Meldung ist durch folgende Eigenschaften charakterisiert:

- Eine Meldung wird gekennzeichnet durch eine Änderung eines binären Signals (Flanke).
- Die Signaländerung führt zu einem neuen binären Signalzustand, der die Zeit  $t > 0$  andauert.
- Jede Signaländerung kann von einem Signalempfänger quittiert werden.
- Der Quittierungszustand kann vom Meldungsauslöser überwacht werden.
- Eine erneute Signaländerung kann auch ohne Quittierung der letzten Signaländerung stattfinden.

Eine SIMATIC S7 kann Meldungen mit verschiedenen Bausteinen auslösen

Tabelle 3- 1 Meldungsauslösen mit verschiedenen Bausteinen

Baustein	Bezeichnung	Anzahl überwachter Signale	Quittierung	Begleitwerte	Severity
SFB 36	NOTIFY	1	Nein	1 ... 10	0 ... 127
SFB 31	NOTIFY_8P	8	Nein	1 ... 10	0 ... 127
SFB 33	ALARM	1	Ja (SFB 33)	1 ... 10	0 ... 127
SFB 34	ALARM_8	8	Ja (SFB 34)	Nein	0 ... 127
SFB 35	ALARM_8P	8	Ja (SFB 35)	1 ... 10	0 ... 127
SFC 17	ALARM_SQ	1	Ja (SFC 19)	1	Nein

Baustein	Bezeichnung	Anzahl überwachter Signale	Quittierung	Begleitwerte	Severity
SFC 18	ALARM_S	1	implizit quittiert	1	Nein
SFC 107	ALARM_DQ	1	Ja (SFC 19)	1	Nein
SFC 108	ALARM_D	1	implizit quittiert	1	Nein

Im S7-Anwendungsprogramm ist festgelegt, ob eine Quittierung seitens des Meldungsempfängers notwendig ist. Das S7-Programm unterscheidet zwischen Quittungen für das Eintreten des Meldungszustandes (Meldung gekommen) und Quittungen für das Beenden eines Meldungszustandes (Meldung gegangen).

Die OPC-Schnittstelle bietet keine Möglichkeit für eine solche Unterscheidung, es wird nur die Quittierung für das Auftreten einer Meldung unterstützt. Die Quittierung für das Ende des Meldungszustandes erfolgt implizit durch den Alarms-&-Events-Server.

Zusätzlich zu den bausteinbezogenen Meldungen unterstützt der OPC-Alarm&Event-Server:

- Symbolbezogene Meldungen (SCANs)  
Ermöglichen asynchron zum SPS-Anwenderprogramm die Überwachung von Bits in den Bereichen E, A, M und DB der CPU.
- Diagnosemeldungen  
Systemdiagnose Anwenderdiagnose mit WR\_USMSG (SFC 52)

### 3.3.1.9 Meldungen, wie sieht die Praxis aus (Beispiel)?

#### Beispiele zur Behandlung von Meldungen

Nachfolgend werden Ihnen 2 Beispiele zur Behandlung von Meldungen dargestellt:

- Meldung ohne Quittierung
- Meldung mit Quittierung

### Meldung ohne Quittierung

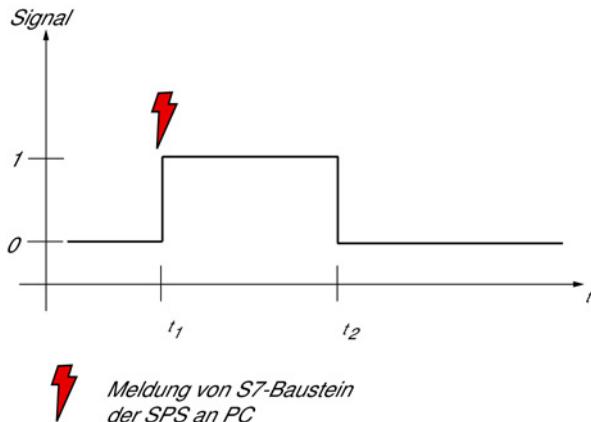


Bild 3-15 Signalzustände bei einer Meldung ohne Quittierung

Ein S7-Baustein überwacht den Füllstand eines Behälters, der im Laufe eines Produktionsprozesses gefüllt wird. Wenn der Behälter voll ist, löst der S7-Baustein eine Meldung aus ( $t_1$ ) und die Produktion wird gestoppt. Die Meldung muss nicht quittiert werden, die Unterbrechung der Produktion erfolgt ohne weitere Maßnahmen durch die Steuerung. Wenn die Steuerung erkennt, dass der Behälter entleert wurde, beendet sie die Meldung ( $t_2$ ) und die Produktion wird fortgesetzt.

### Meldung mit Quittierung

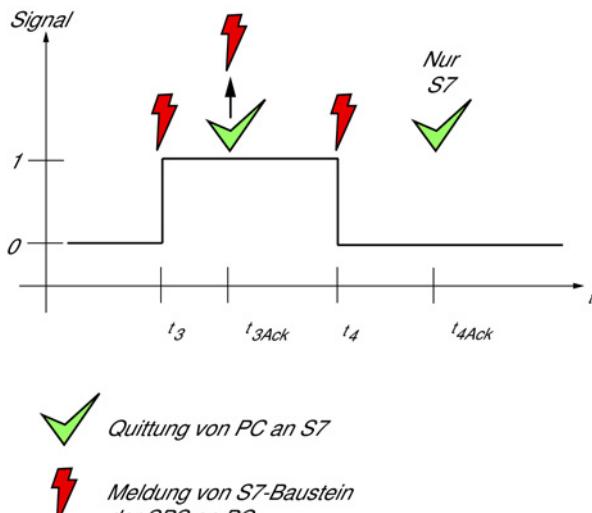


Bild 3-16 Signalzustände bei einer Meldung mit Quittierung

Ein S7-Baustein überwacht einen Kesseldruck. Beim Überschreiten des Grenzwertes löst der S7-Baustein eine Meldung aus ( $t_3$ ), gleichzeitig werden eine Warnlampe am Überdruckventil und eine Alarmhupe aktiviert.

Die Bedienerquittung ( $t_{3Ack}$ ) schaltet die Alarmhupe ab, der Meldungszustand bleibt jedoch bestehen, weil der Kesseldruck über dem Grenzwert liegt. Auch die Warnlampe wird durch die Quittung nicht abgeschaltet. Das Empfangen der Bedienerquittung löst in der S7-Steuerung eine weitere Meldung aus.

Nach dem Druckablassen erkennt der S7-Baustein das Unterschreiten des Grenzdrucks und beendet die Meldung ( $t_4$ ). Auch das Beenden des Meldungszustandes löst eine Meldung aus.

Eine Bedienerquittung für das Beenden des Meldungszustandes schaltet die Warnlampe aus ( $t_{4Ack}$ ). Diese Quittung ist an der OPC-Schnittstelle nicht sichtbar, weil OPC nur Quittungen für das Eintreten von Meldungszuständen unterstützt.

### 3.3.2 Alarms & Events Schnittstelle

#### 3.3.2.1 Schnittstellen, welche sind für Alarms & Events spezifiziert?

##### Für Alarms & Events sind zwei Schnittstellen spezifiziert

Für Alarms & Events sind die Automation- und Custom-Schnittstellen spezifiziert:

- Alarms & Events Automation Interface, Standard, December 15, 1999, Version 1.01  
Beschreibung des OPC-Alarms-&-Events-Servers sowie die Spezifikation der Custom-Schnittstelle dieses Servers
- OPC Alarms & Events Custom Interface, October 2, 2002, Version 1.10  
Spezifikation der Automation-Schnittstelle des OPC-Alarms-&-Events-Servers

Eine Übersicht der Spezifikationen finden Sie im Literaturverzeichnis von Band 2.

## 3.4 OPC XML

#### 3.4.1 Einführung XML und SOAP

##### 3.4.1.1 XML und SOAP, was ist das?

##### OPC goes Internet

Prozessdaten können bei Benutzung der OPC XML-Data Access-Schnittstelle auch über das Internet gelesen, geschrieben und in einfacher Form auch beobachtet werden.

Dazu benutzt OPC SOAP.

### **Was ist SOAP?**

SOAP stellt einen einfachen und durchsichtigen Mechanismus zum Austausch von strukturierter und getypter Information zwischen Rechnern in einer dezentralisierten, verteilten Umgebung zur Verfügung.

Das "Simple Object Access Protokoll" (SOAP) bildet eine Basis für den XML-basierten Informationsaustausch.

### **Was ist XML und OPC XML?**

XML (eXtensible Markup Language) ist ein Standard fürs Internet, der auch in vielen anderen Bereichen von Standardsoftware eine weite Verbreitung gefunden hat. XML bietet ebenso wie HTML die Möglichkeit, Daten mit Metainformationen zu versehen. Allerdings können mit XML eigene Datenstrukturen und eigene Attribute definiert werden.

Für OPC wurde basierend auf XML ebenfalls eine neue Spezifikation, OPC XML, definiert, welche die Prozessdatenschnittstelle mit XML-Datensätzen beschreibt.

### **Wie funktioniert der Zugriff auf OPC über das Internet?**

Die Kommunikation über die DCOM-Schnittstellen von OPC ist meistens auf lokale Netze beschränkt. Auch sind COM-Schnittstellen in der Regel auf Windows-basierte Systeme festgelegt. Firewalls bieten aus Sicherheitsgründen nur eingeschränkte Zugriffsmöglichkeiten aus dem Internet und in das Internet. Mit OPC XML wird ein Standard zur Verfügung gestellt, welcher die Kommunikation über das plattformunabhängige Protokoll SOAP (Simple Object Access Protokoll) erlaubt. Der Datenzugriff mittels OPC XML hat einen an OPC Data Access angelehnten Funktionsumfang.

### **Wie ist die Schnittstellenbeschreibung mit XML?**

Die Datenschnittstellen und Methoden werden dabei durch XML beschrieben. Die exakte Beschreibung der Methoden ist in einer WSDL-Spezifikation (Web Service Definition Language) festgelegt, die von der OPC Foundation zusammen mit der OPC XML DA-Spezifikation geliefert wird. Die Methoden werden mittels SOAP (XML-Protokoll) beschrieben und über das HTTP-Protokoll versandt. Vereinfacht kann man sagen:

SOAP = HTTP + XML

Die folgende Grafik zeigt diesen Zusammenhang:

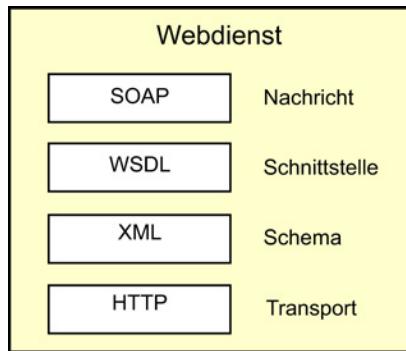


Bild 3-17 Schnittstellenbeschreibung mit XML

### Wie funktioniert die Datenübertragung mit dem HTTP-Protokoll?

Der Zugriff auf Methoden direkt aus dem Internet stellt ein erhebliches Sicherheitsrisiko dar. Deshalb verwendet SOAP für die Datenübertragung ausschließlich den Internet HTTP-Kanal (HTTP = HyperText Transfer Protokoll), der sich durch eine Firewall einfach administrieren lässt.

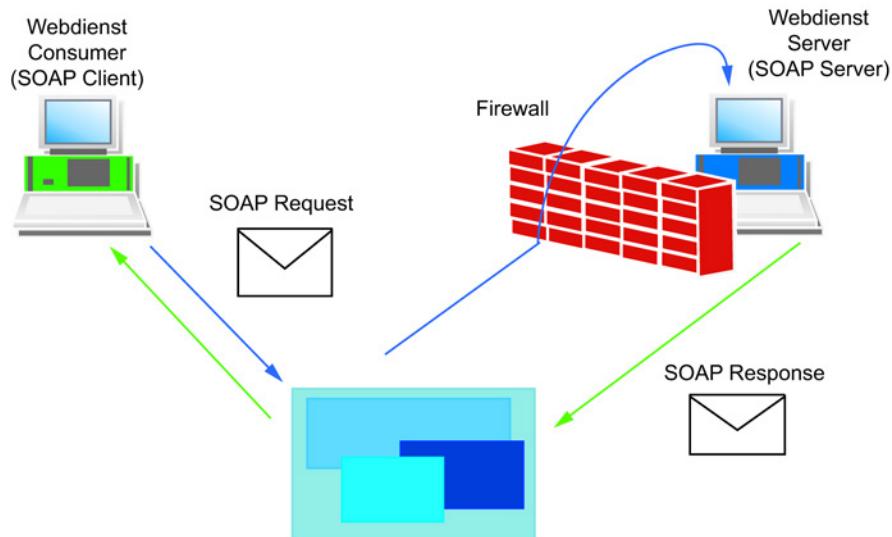


Bild 3-18 Datenübertragung mit dem HTTP-Protokoll

### 3.4.1.2 Web-Dienste, wozu dienen sie?

#### Web-Dienste verbinden Client und Server über das Internet

Ein Web-Dienst bietet die Möglichkeit, Funktionsaufrufe über das Internet an einen Web-Server zu senden. Die Beschreibung der Methoden und Parameter, die ein Web-Dienst zur Verfügung stellt, ist in WSDL-Dateien im XML-Format abgelegt. Diese können vom Client beim Web-Server abgefragt werden. Für die Nutzung von Webdiensten ist damit im wesentlichen nur die Internet-Adresse, die URL des Web-Dienstes, nötig. Die Datenübertragung erfolgt über SOAP mit HTTP-Telegrammen.

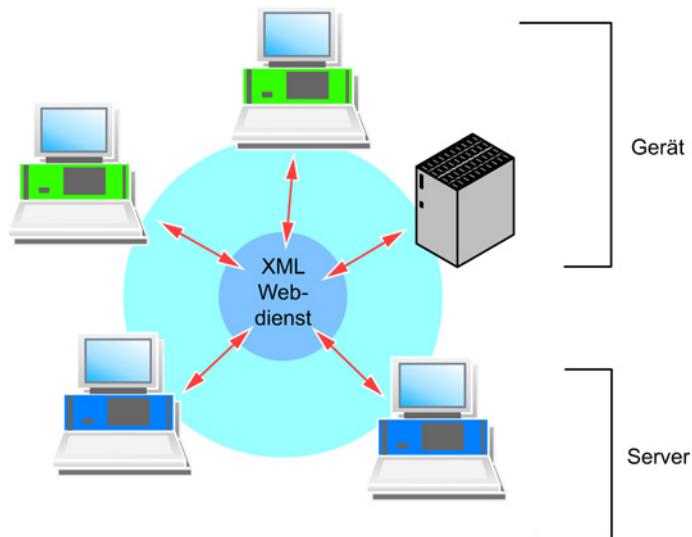


Bild 3-19 Funktionsaufrufe über das Internet

### 3.4.2 OPC-XML-Schnittstelle

#### 3.4.2.1 OPC-XML-Schnittstelle, was leistet sie?

##### OPC kann über die XML-Schnittstelle auf das Internet zugreifen

OPC XML ist ein Standard, der die Kommunikation mit einem plattformunabhängigen Protokoll über das Internet ermöglicht. Ein Client ist nicht mehr auf eine Windows-Umgebung (COM) festgelegt. Auch andere Betriebssysteme, wie zum Beispiel LINUX, können mit dem HTTP-Protokoll und der SOAP-Schnittstelle OPC-Daten über das Internet beobachten und austauschen.

Der Datenzugriff mittels OPC XML hat einen an OPC Data Access angelehnten Funktionsumfang, es stehen allerdings nur einfache Schreib- und Lesedienste zur Verfügung. Änderungsgesteuerte Rückmeldungen über Datenänderungen, wie bei den DCOM OPC DA-Schnittstellen, sind für OPC XML aufgrund der losen Internet-Verbindung nicht vorgesehen.

Ein möglicher Nachteil dieser Schnittstelle ist, dass sie einen Internet-Server erfordert. Außerdem ist eine geringe Performance über das Internet zu erwarten.

### Welche Schnittstellen bedient der SIMATIC NET OPC-Server?

Die folgende Grafik zeigt die interne Struktur des SIMATIC NET OPC-Servers sowie die verfügbaren Schnittstellen:

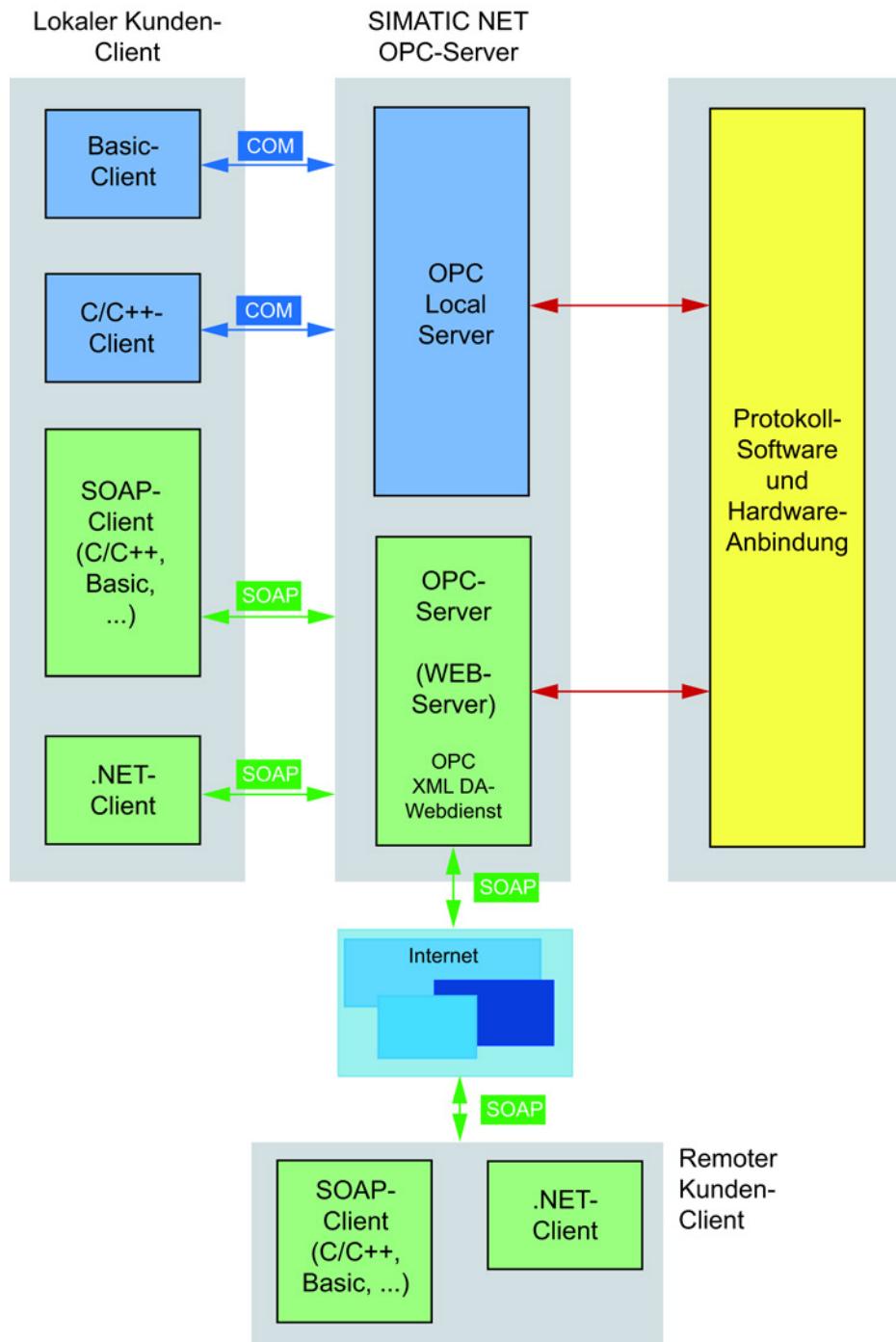


Bild 3-20 Struktur des SIMATIC NET OPC-Servers und verfügbare Schnittstellen

### 3.4.2.2 Web-Dienst OPC XML, wie funktioniert er?

Die OPC-XML-Spezifikation ist bei SIMATIC NET durch einen Web-Dienst des Microsoft-Internet-Information-Servers (IIS) realisiert. Beachten Sie, dass der Internet-Information-Server eine Komponente des Betriebssystems ist, die separat installiert und konfiguriert werden muss.

#### So funktioniert der OPC-XML-Web-Dienst

Die Komponente OPC XML ist weitgehend unsichtbar für den Benutzer. Sie wird automatisch durch den IIS gestartet, wenn ein Web-Client die entsprechenden OPC-XML-Dienste anfordert. Die folgende Grafik stellt diesen Zusammenhang dar:

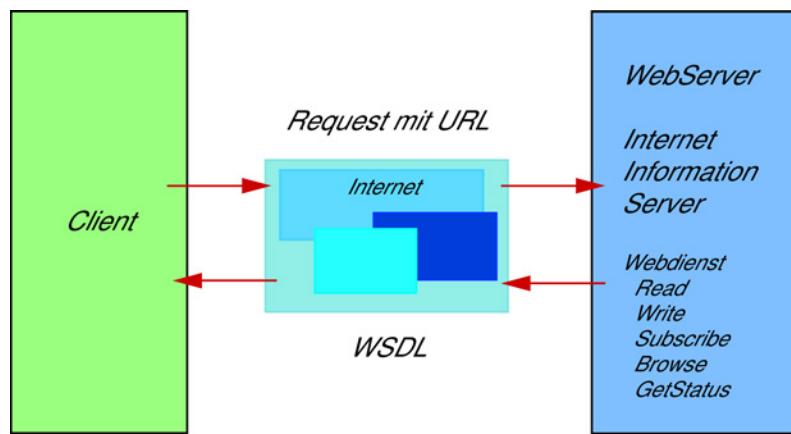


Bild 3-21 Web-Dienst des Internet-Information-Servers (IIS)

### 3.4.2.3 Einfache Dienste Lesen / Schreiben, welche Methoden gibt es bei XML?

#### Diese Methoden stehen zur Verfügung

Beim SIMATIC-NET-OPC-XML-Server sind die folgenden Methoden implementiert:

- **GetStatus**

Mit GetStatus können der allgemeine Zustand sowie herstellerspezifische Information (Version, Produktnamen) abgefragt werden.

- **Read**

Mit dem Dienst Read kann der Wert einer (oder mehrerer) Variablen gelesen werden.

- **Write**

Der Dienst Write schreibt den Wert einer (oder mehrerer) Variablen. Optional kann der OPC-XML-Server anschließend noch die Methode Read ausführen.

- Subscription, SubscriptionPolledRefresh, SubscriptionCancel

Bei Subscriptions werden die Variablen angemeldet und eventuelle Änderungen (zyklisch) mit SubscriptionPolledRefresh gelesen. Eine Subscription kann mit SubscriptionCancel beendet werden.

Eine Besonderheit beim SubscriptionPolledRefresh ist das Zeitfenster, das mit WaitTime und HoldTime spezifiziert wird. Dabei wird der Aufruf im Server bis zur WaitTime angehalten. Die Beantwortung des Aufrufs erfolgt dann sobald eine Werteänderung erkannt wird oder spätestens nach Erreichen der Hold-Time.

- Browse

Der Dienst Browse erlaubt das Navigieren durch einen hierachischen Adressraum. Anders als beim COM-Interface können mit einem Request sowohl Branches als auch Leafs gelesen werden.

Außerdem kann festgelegt werden, welche Eigenschaften eines Elements der Server zurück liefern soll.

- GetProperties

Alternativ zu Browse können Eigenschaften von Elementen auch mit dem Dienst GetProperties gelesen werden

## Synchrone / Asynchrone Verwendung der Methoden

Die genannten Methoden sind gemäß OPC-XML-DA-Spezifikation asynchron ausgelegt. Request und Response sind getrennte Protokollbestandteile. Die Anwendung dieser Methoden durch höhere Programmiersprachen wie C#, Visual Basic usw. erlaubt eine Zusammenfassung von Request und Response zu einer synchronen Methode.

Beim Anlegen einer Proxy-Klasse für das Client-Programm wird für jede Methode eine synchrone und eine asynchrone Variante erzeugt. Beide Varianten nutzen die gleichen OPC-XML-DA-Methoden. Die Verwendung der asynchronen Variante wirkt sich allerdings günstig auf das Laufzeitverhalten des Client-Programms aus.

Detailinformationen finden Sie in /1/

## 3.5 OPC Unified Architecture

### 3.5.1 Einführung in OPC UA

#### 3.5.1.1 Einleitung

**Was vereinigt OPC UA?**

Die bisherigen Funktionalitäten und Möglichkeiten der bestehenden OPC-Normen wie Data Access, Alarm & Events, Security, Historicial-, Complex- und XML Data Access sind in einer neuen, sicheren und leistungsfähigen Spezifikation zusammengefasst: OPC Unified Architecture (OPC UA)

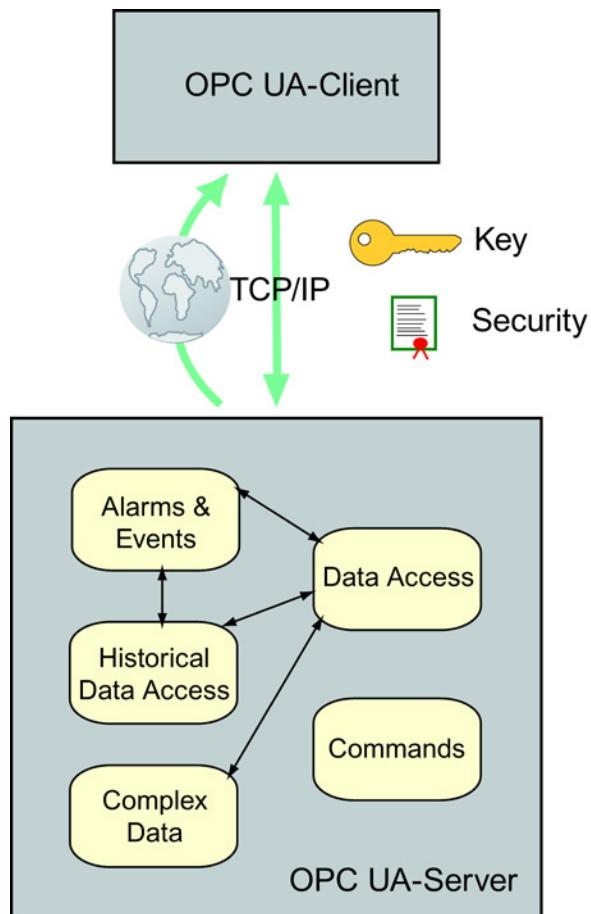


Bild 3-22     Funktionen von OPC Unified Architecture

Für OPC UA wird ein neues TCP-basiertes, sicheres, leistungsfähiges und normiertes Kommunikationsprotokoll eingesetzt.

### **Was sind die Vorteile der OPC UA-Architektur?**

- Vereinheitlichung der bisherigen OPC-Normen zu einer Schnittstelle.  
Damit vereinfacht sich die Entwicklung von Client-Anwendungen.
- Der UA-Namensraum bietet die Möglichkeit der vollständigen Modellierung von beliebig komplexen Systemen. Er hat einen deutlich höheren Funktionsumfang als der bisherige Namensraum.

### **Was sind die Vorteile der OPC UA-Kommunikation gegenüber der COM/DCOM-Schnittstelle?**

- Plattformunabhängige Kommunikation
- Erhöhte Sicherheit durch Unabhängigkeit von DCOM:
  - Keine komplexen Sicherheitseinstellungen
  - Kein offener Port 135
  - Keine weiteren dynamisch zugewiesenen Ports
  - Leichtere Einstellung der Firewall
  - Keine langen Timeouts bei Störungen
- Hohe Datensicherheit durch moderne, auf Zertifikaten basierende Authentifizierungsverfahren
- Auswahl zwischen verschiedenen Kommunikationsprotokollen, passend zum jeweiligen Anwendungsfall

#### **3.5.1.2 Die Sicherheit bei OPC UA**

##### **Wie wird die hohe Sicherheit gewährleistet?**

Ein OPC UA-Client und ein OPC UA-Server müssen sich gegenseitig durch digitale Zertifikate und zugehörige Schlüssel authentisieren und autorisieren. Die Nachrichten werden entsprechend verschlüsselt. Für die Authentisierung werden gängige X.509-Zertifikate verwendet.

Einer OPC UA-Client-Anwendung ist normalerweise ein Zertifikatespeicher zugeordnet. Dort können Schlüssel von autorisierten OPC UA-Servern abgelegt werden.

Für die Verwaltung der Zertifikate verwendet der SIMATIC NET OPC UA Server eine Zielsystem-spezifische Public Key Infrastructure (PKI) auf dem Client.

#### **3.5.1.3 Die Kommunikationsarten von OPC UA**

##### **Was beinhalten die OPC UA-Kommunikationsarten "TCP binär" und "XML"?**

Das Kommunikationsprotokoll von OPC UA ist auf unterster Ebene TCP-basiert und deshalb plattformübergreifend einsetzbar, auch auf Embedded Systemen. Eine sichere, verschlüsselte Übertragung ist in allen Fällen erforderlich.

Als Protokoll an der OPC UA-Anwenderschnittstelle stehen gemäß der Norm folgende Möglichkeiten zur Verfügung:

- Einfaches XML/SOAP mit HTTP/HTTPS über Port 80/443
- Binäres TCP über Port 4840 (und weiteren Ports wie z.B. Port 4845 oder ggf. Port 5000, 6000 etc., wenn weitere Server hinzukommen)

Besser noch: gepacktes binäres TCP

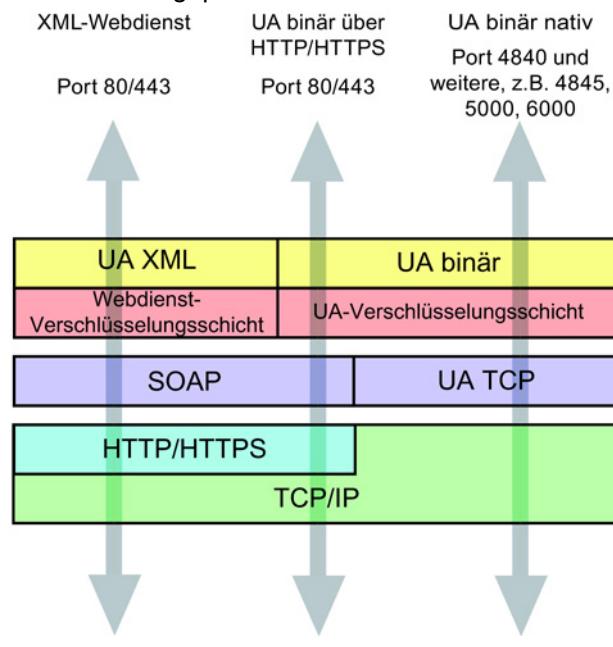


Bild 3-23 Die von OPC UA genutzten Protokolle

Das Protokoll kann über die URL-Adresse des UA-Servers an der OPC UA-Anwenderschnittstelle gewählt werden. Hierzu stehen Ihnen alternativ die beiden folgenden Möglichkeiten zur Verfügung.

Beispiele:

- OPC UA XML-Web-Dienste unter Angabe einer URL, z.B.:
  - `http://<hostname>:80`
  - oder
  - `https://<hostname>:443`
- Reines (natives) binäres TCP-Protokoll unter Angabe von:
  - `opc.tcp://<hostname>:4840`

Auf der Applikationsschicht sind die OPC UA-Funktionsaufrufe gleich.

Nicht jeder OPC UA-Server unterstützt alle Protokolle.

### **Was sind die Vorteile des Protokolls "OPC UA nativ binär"?**

Unter OPC UA hat das Protokoll "OPC UA nativ binär" die höchste Übertragungsgeschwindigkeit, da die Daten komprimiert übertragen werden und somit wenig Verpackungsinformation verwendet werden muss. Es benötigt den geringsten Zusatzaufwand. Beispielsweise wird kein XML-Parser benötigt, der für SOAP und HTTP erforderlich ist.

Das Format ist bis auf Binärebene normiert. Dies stabilisiert den Datenaustausch zwischen OPC UA-Client und Server, da keine Freiheitsgrade wie beispielsweise Leerzeichen oder Kommentare bei XML vorhanden sind.

Für die Kommunikation wird beim Protokoll "OPC UA nativ binär" der speziell hierfür spezifizierte TCP-Port 4840 verwendet und beim SIMATIC NET OPC-Server noch Port 4845 und ggf. weitere wie Port 5000 oder 6000. Diese Ports können definiert in einer Firewall freigegeben oder gesperrt werden.

### **Was sind die Vorteile der Protokolle der XML-Web-Dienste?**

XML kann sehr einfach mit gängigen Entwicklungsumgebungen für OPC UA-Anwendungen verwendet werden.

Die Firewall ist meist für den HTTP-Port 80 und den HTTPS-Port 443 freigeschaltet oder kann für diese Ports leicht freigeschaltet werden. Daher ist ein Internetzugang meist ohne weitere Konfiguration für die Nutzung von XML-Web-Diensten möglich.

### **Mit welchen Programmiersprachen kann eine OPC UA-Anwendung auf die OPC UA-Schnittstelle zugreifen?**

Ein OPC UA-Client kann über eine C-, .NET- (C#,VB.NET), JAVA- und eine C++-Schnittstelle auf die OPC UA-Schnittstelle zugreifen. Die zugehörigen Bibliotheken und Assemblies werden von der OPC Foundation inklusive des Kommunikations-Stacks zur Verfügung gestellt.

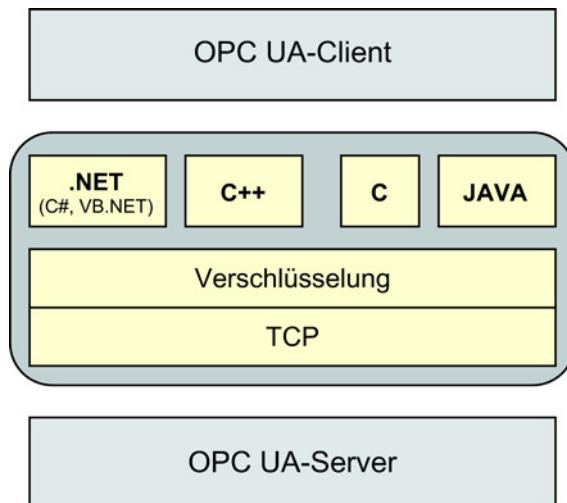


Bild 3-24 Zugang des OPC UA-Client zum OPC UA-Server mit Hilfe verschiedener Programmiersprachen

### 3.5.1.4 Der Namensraum von OPC UA

#### Was zeigt der OPC UA-Namensraum?

Der Namensraum bei OPC UA besteht nicht mehr nur aus Ordnern, Items und Properties. Er ist ein Netzwerk aus Knoten mit Zusatz-Informationen und Verknüpfungen.

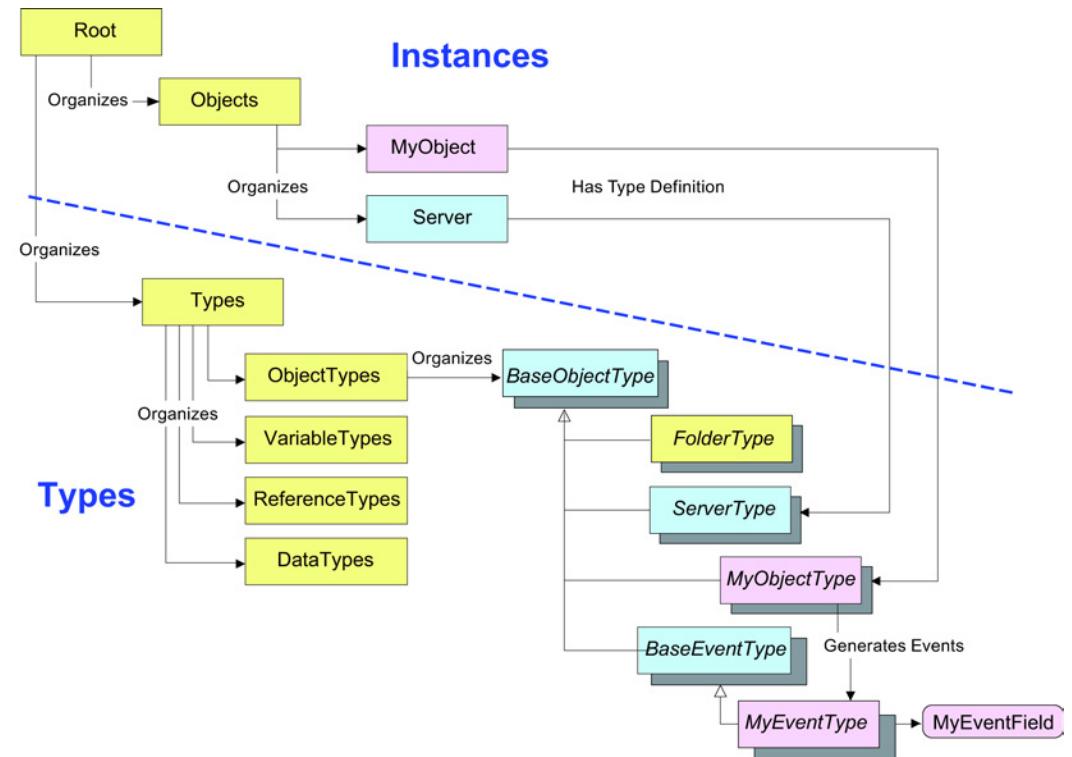


Bild 3-25 Aufbau des Namensraums von OPC UA

Die Kästchen repräsentieren Knoten. Dies sind die Objekte von OPC UA. Die Pfeile bedeuten Referenzen von einem Quellknoten zu einem Zielknoten.

Die Knoten werden sowohl für die Nutzdaten (Instances) als auch für weitere Informationen wie Typbeschreibungen von Daten (Types) verwendet. Die Knoten von OPC UA lassen sich folgendermaßen unterteilen:

- Types

Dies sind die in der OPC UA-Spezifikation und ggf. vom jeweiligen Hersteller spezifizierte Knotentypen, die hinsichtlich ihrer Eigenschaften und Attribute eindeutig definiert sind. Es gibt folgende vier Grundtypen:

- ObjectTypes
- VariableTypes
- ReferenceTypes
- DataTypes

Sie definieren weitere Typen, von denen einige in der Abbildung rechts des Typs "ObjectTypes" abgebildet sind.

Die Types dienen als Typbeschreibung für die Instances.

- Instances

Dies sind die Instanzen der Objekte Ihres realen Projekts. Hinsichtlich ihrer Eigenschaften referenzieren sie je nach Art des Knotens auf verschiedene Typen. In der Abbildung referenzieren die zwei Objekte "MyObject" und "Server" auf die zwei Types "MyObjectType" und "ServerType".

Die Root Ihres OPC UA Servers organisiert sowohl die Typen als auch die Instanzen. Das Organisieren beinhaltet die Definition weiterer Knoten.

Ein Knoten kann folgende Eigenschaften besitzen:

- Attribute, die gelesen werden können
- Methoden die aufgerufen werden können
- Ereignisse, die gemeldet werden können

Viele Standard-Knoten werden von der OPC UA-Spezifikation vorgegeben. Weitere Knotentypen können herstellerspezifisch hinzukommen. Der Namensraum wird im OPC Scout V10 in einer Baumstruktur angezeigt.

### **Auffinden von OPC UA Servern auf einem System mit der "Discovery"**

Der OPC UA-Suchdienst "Discovery" erlaubt das Auffinden von OPC UA Servern, die auf einem System vorhanden sind. Dieser Suchdienst verwendet den für OPC UA reservierten Port 4840, den Host-Namen und/oder die IP-Adresse. Er meldet die Endpunkte aller OPC UA Server sowie deren Protokolle, Ports und Sicherheitsanforderungen.

### **Endpunkte der OPC UA Server**

Ein OPC UA Server stellt Endpunkte für die Kommunikation zur Verfügung.

Ein Endpunkt ist die physikalische Adresse in einem Netzwerk, die es OPC UA-Clients ermöglicht, auf einen oder mehrere Dienste des OPC UA Servers zuzugreifen.

Einzelheiten zu den Endpunkten OPC UA Server finden in Band 2 dieses Handbuchs in der Manual Collection der DVD "SIMATIC NET PC Software".

### 3.5.1.5 Weitere Eigenschaften von OPC UA

#### Was kann OPC UA noch?

OPC UA bietet noch eine Reihe weiterer Funktionen, von denen einige kurz vorgestellt werden:

- Redundanz

Zwischen mehreren OPC UA-Clients und mehreren OPC UA-Servern bietet OPC UA Redundanz-Funktionen, die vom einfachen Übernehmen von Sessions zwischen OPC UA Clients bis zur abgestimmten Redundanz zwischen OPC UA-Servern reichen.

- Verbindungsüberwachung

Ein OPC UA-Server erkennt Verbindungsunterbrechungen zum OPC UA-Client und ein OPC UA-Client erkennt Verbindungsunterbrechungen zum OPC UA-Server. Die Überwachungszeiten sind in der Norm spezifiziert.

- Datenwertspeicherung

Verbindungsunterbrechungen führen in der Regel nicht mehr zu Datenverlust. Die Datenwerte werden gespeichert. Nicht ordnungsgemäß empfangene Daten können erneut angefordert werden. Der Empfang erhaltener Daten kann quittiert werden.

## 3.5.2 Die OPC UA-Schnittstelle

### 3.5.2.1 Welche Schnittstellenspezifikationen der OPC Unified Architecture gibt es?

#### Welche Schnittstellenspezifikationen der OPC Unified Architecture gibt es?

Die OPC UA-Spezifikation besteht aus mehreren Teilen. Es handelt sich um Dienstspezifikationen.

Die Anwendungsschnittstelle ist durch den ebenfalls von der OPC Foundation zur Verfügung gestellten sprachabhängigen OPC UA-Server-Stack bzw. die OPC UA-Anwenderbibliotheken spezifiziert. Die Spezifikation besteht aus den folgenden Teilen:

Teil	Thema	Titel	URL
Part 1	Concepts	OPC UA Specification: Part 1 – Concepts, Version 1.0 or later	<a href="http://www.opcfoundation.org/UA/Part1">http://www.opcfoundation.org/UA/Part1</a>
Part 2	Security Model	OPC UA Specification: Part 2 – Security Model, Version 1.0 or later	<a href="http://www.opcfoundation.org/UA/Part2">http://www.opcfoundation.org/UA/Part2</a>
Part 3	Address Space Model	OPC UA Specification: Part 3 – Address Space Model, Version 1.0 or later	<a href="http://www.opcfoundation.org/UA/Part3">http://www.opcfoundation.org/UA/Part3</a>
Part 4	Services	OPC UA Specification: Part 4 – Services, Version 1.0 or later	<a href="http://www.opcfoundation.org/UA/Part4">http://www.opcfoundation.org/UA/Part4</a>
Part 5	Information Model	OPC UA Specification: Part 5 – Information Model, Version 1.0 or later	<a href="http://www.opcfoundation.org/UA/Part5">http://www.opcfoundation.org/UA/Part5</a>

Teil	Thema	Titel	URL
Part 6	Service Mappings	OPC UA Specification: Part 6 – Mapping, Version 1.0 or later	<a href="http://www.opcfoundation.org/UA/Part6">http://www.opcfoundation.org/UA/Part6</a>
Part 7	Profiles	OPC UA Specification: Part 7 – Profiles, Version 1.0 or later	<a href="http://www.opcfoundation.org/UA/Part7">http://www.opcfoundation.org/UA/Part7</a>
Part 8	Data Access	OPC UA Specification: Part 8 – Data Access, Version 1.0 or later	<a href="http://www.opcfoundation.org/UA/Part8">http://www.opcfoundation.org/UA/Part8</a>
Part 9	Alarms and Conditions	OPC UA Specification: Part 9 – Alarms and Conditions, Version 1.0 or later	<a href="http://www.opcfoundation.org/UA/Part9">http://www.opcfoundation.org/UA/Part9</a>
Part 10	Programs	OPC UA Specification: Part 10 – Programs, Version 1.0 or later	<a href="http://www.opcfoundation.org/UA/Part10">http://www.opcfoundation.org/UA/Part10</a>
Part 11	Historical Access	OPC UA Specification: Part 11 – Historical Access, Version 1.0 or later	<a href="http://www.opcfoundation.org/UA/Part11">http://www.opcfoundation.org/UA/Part11</a>
Part 12	Discovery	OPC UA Specification: Part 12 – Discovery, Version 1.0 or later	

### **3.5.2.2 Wie wird die Verbindung zu einem OPC UA-Server aufgenommen?**

#### **Begriffsdefinition**

Begriff	Bedeutung
Sicherer Kanal	Ein Kommunikationskanal zur sicheren Datenübertragung zwischen den Kommunikations-Stacks von OPC UA-Client und Server. Jeder sichere Kanal hat eine globale Kennung (Identifier) und enthält spezifische Informationen zur Verschlüsselung der Nachrichten, die über diesen Kanal laufen.
Kanal-Kennung	Kennung eines sicheren Kanals, die beim Aufbau des Kanals festgelegt wird.
Kommunikations-Stack	Ein schichtweise aufgebauter Satz von Software-Modulen zwischen Applikation und Hardware, der verschiedene Aufgaben bei der Kommunikation zwischen Teilnehmern abwickelt.
Zertifikat	Hier: Schlüssel (Signatur), der einen Teilnehmer innerhalb eines kryptografischen Systems identifiziert.
Session	Zeitlich begrenzte Sitzung, während der Daten zwischen OPC UA-Client und Server ausgetauscht werden.
Session-ID	Identifikationsnummer einer Session, die der OPC UA-Server nach dem Verbindungsaufbau an den OPC UA-Client vergibt. Bei allen folgenden Anfragen muss der Client diese Session-ID dem Server mitgeben.

## Verbindungsauflaufbau

Die folgende Abbildung gibt einen Überblick über die Komponenten, die am Verbindungsauflaufbau zwischen OPC UA-Client und OPC UA-Server beteiligt sind.

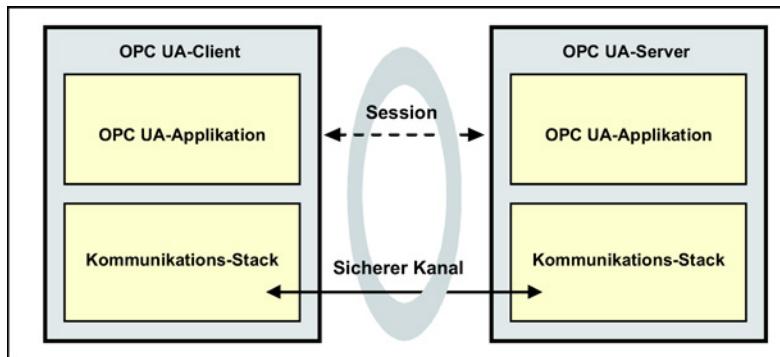


Bild 3-26 Schema des Verbindungsauflaufbaus bei OPC UA

Folgende Schritte werden bei einem Verbindungsauflaufbau durchlaufen:

### 1. Aufbau eines sicheren Kanals

Zum Aufbau eines sicheren Kanals wird der Dienst "OpenSecureChannel" verwendet.

Dieser Dienst öffnet oder erneuert einen sicheren Kanal während einer Session. Der sichere Kanal ermöglicht die vertrauliche Übertragung von Informationen zwischen OPC UA-Client und Server.

Nach dem Aufbau eines sicheren Kanals wendet der Kommunikations-Stack die verschiedenen Sicherheits-Algorithmen für die zu versendenden Telegramme an. Nach Signierung durch das Zertifikat des Absenders werden die Telegramme verschlüsselt übertragen. Nur autorisierte Partner können das Telegramm entschlüsseln.

### 2. Aufbau einer Session

Zum Aufbau einer Session wird der Dienst "CreateSession" verwendet.

Mit Hilfe dieses Dienstes baut ein OPC UA-Client eine Session auf. Der OPC UA-Server gibt die Session-ID zurück.

Bei nachfolgenden Anfragen des Clients akzeptiert der Server die Anfrage nur, wenn der Client sowohl die Kanal-Kennung als auch die Session-ID mitgibt.

### **3.5.2.3 Wie kann der OPC UA-Namensraum durchsucht werden?**

#### **Wie kann der OPC UA-Namensraum durchsucht werden?**

Zum Durchsuchen des OPC UA-Namensraums stehen folgende Dienste zur Verfügung:

- "Browse"

Dieser Dienst wird genutzt, um die Referenzen (Verknüpfungen) eines Knotens festzustellen.

- "Read"

Dieser Dienst wird genutzt, um ein oder mehrere Attribute eines oder mehrerer Knoten festzustellen.

In der Antwort (response) wird der jeweils gesuchte Wert (Referenz, Eigenschaft bzw. Attribut) geliefert.

### **3.5.2.4 Wie können Daten gelesen und geschrieben werden?**

#### **Wie kann einfach gelesen und geschrieben werden?**

Zum Lesen und Schreiben der Attribut-Werte von Knoten stehen die beiden Dienste "Read" und "Write" zur Verfügung.

- "Read"

Dieser Dienst wird genutzt, um ein oder mehrere Attribute eines oder mehrerer Knoten festzustellen. Bei strukturierten Attribut-Werten, dessen Elemente wie bei einem Array indiziert sind, können Clients den ganzen Satz indizierter Werte im Verbund lesen, sie können bestimmte Bereiche des Satzes lesen oder einzelne Elemente.

Die Aktualität der Werte wird mit Hilfe des Parameters "maxAge" festgestellt.

- "Write"

Dieser Dienst wird genutzt, um Werte an ein oder mehrere Attribute eines oder mehrerer Knoten zu schreiben. Bei strukturierten Attribut-Werten, dessen Elemente wie bei einem Array indiziert sind, können Clients den ganzen Satz indizierter Werte im Verbund schreiben, sie können bestimmte Bereiche des Satzes schreiben oder einzelne Elemente.

Der Dienstauftrag steht solange an, bis die Werte geschrieben wurden oder bis festgestellt wurde, dass die Werte nicht geschrieben werden konnten.

Der Zugriff bei "Read" und "Write" erfolgt über die "NodeID" des bzw. der jeweiligen Knoten. Die NodeID ist die Kennung eines Knotens im Namensraum von OPC UA.

### 3.5.2.5 Wie werden UA-Daten und Ereignisse beobachtet?

#### Begriffsdefinition

Begriff	Bedeutung
Subscription	Eine Subscription dient der Datenübertragung vom OPC UA-Server an den Client. Eine Subscription enthält einen Satz von MonitoredItems, die in einer Notification an den Client übertragen werden.
MonitoredItem	Ein Client definiert MonitoredItems zur Erfassung von Daten und Ereignissen. Ein MonitoredItem identifiziert ein zu beobachtendes Item, seine zugehörige Subscription und die Notification zur Übermittlung der Daten durch die Subscription.
Item	Ein Item kann ein beliebiges Knoten-Attribut sein.
Notification	Eine Datenstruktur, die Änderungen in Datenwerten oder Ereignissen beschreibt. Diese Datenstruktur wird mit den Daten der MonitoredItems befüllt.
NotificationMessage	Eine Notification wird zur Übertragung an den Client von der Subscription in eine NotificationMessage gepackt.
Publish Request	Eine Anfrage des Client an den Server zur Übermittlung von Daten
Attribut	Ein einfaches Merkmal eines Knotens, das durch die OPC UA-Spezifikation definiert wird.
Knoten / NodeID	Ein Knoten ist eine fundamentale Komponente des Namensraums. Jeder Knoten wird durch seine NodeID ausgewiesen.

## Das MonitoredItem-Modell

Das MonitoredItem-Modell beschreibt das Beobachten folgender Eigenschaften bzw. Objekte:

- Attribute

Ein Attribut wird hinsichtlich der Änderung seines Werts beobachtet. Jede Änderung eines Attributs führt zur Erzeugung einer Notification (keine Anwendung der Filter, siehe unten).

Attribute sind nicht zu verwechseln mit dem Wert-Attribut einer Variablen.

- Variablen

Eine Variable kann den Wert oder den Status ändern. Im Unterschied zum zuvor genannten "Attribut" wird bei einer Variablen das "Wert-Attribut" der Variablen, der Status, beobachtet.

- Knoten

Knoten können Werte und Ereignisse liefern. Ereignisse können nur von Knoten gebildet werden, bei denen im "EventNotifier-Attribut" das Bit "SubscribeToEvents" gesetzt ist. Zur Beobachtung von Ereignissen können Objekte und Views verwendet werden.

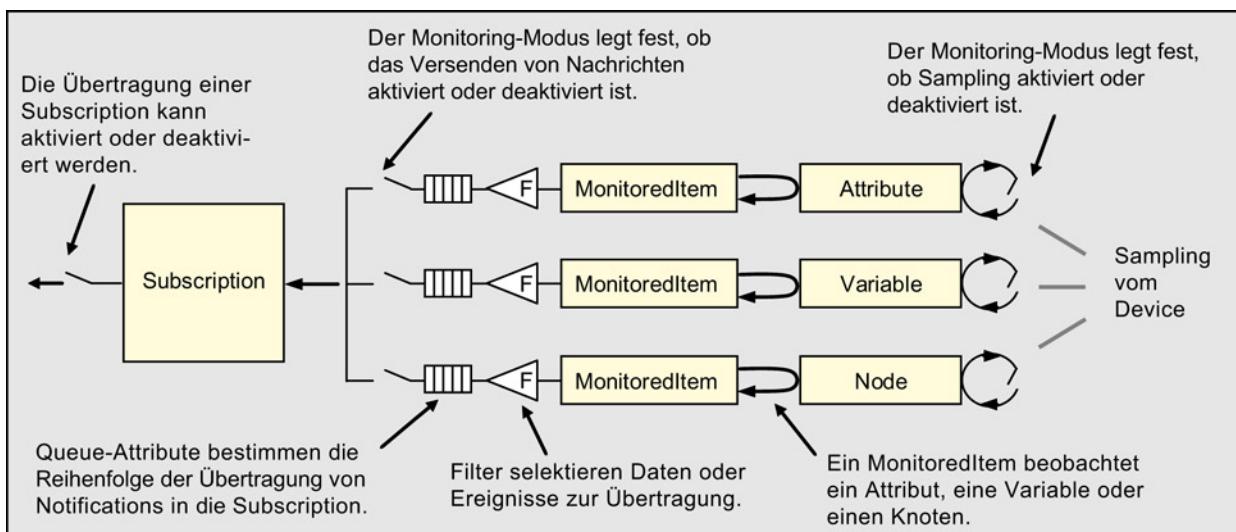


Bild 3-27 Das MonitoredItem-Modell

## Übersicht des Beobachtens von Daten mit dem MonitoredItem-Modell

Beim Beobachten von Daten werden die Informationen in folgenden Schritten vom Device über den OPC UA-Server an den OPC UA-Client übertragen:

1. Für jedes zu beobachtende Item wird durch den UA-Client ein MonitoredItem definiert.
2. Die Items der abgetasteten Devices werden vom OPC UA-Server in Attributen, Variablen oder Knoten beobachtet und die aktuellen Daten in MonitoredItems gespeichert.
3. Jedes MonitoredItem erzeugt eine Notification (wenn dies durch den Monitoring-Modus des MonitoredItem aktiviert wurde, siehe Attribute des MonitoredItem").
4. Die Subscription fasst die Notifications in einer NotificationMessage zusammen.

5. Die Subscription überträgt die NotificationMessage an den Client.
6. Der Client quittiert den Erhalt der NotificationMessage.

## Die Attribute der MonitoredItems

Das MonitoredItem besitzt vier Attribute mit folgenden Funktionen:

- Das Sampling-Intervall

Jedem MonitoredItem, das durch den Client erzeugt wird, wird ein Sampling-Intervall zugewiesen. Mit dem Sampling-Intervall wird festgelegt, in welchem kürzesten Intervall der Server die ihm unterlagerte Datenquelle (das Device) abtastet.

Das Sampling-Intervall wird entweder vom Publishing-Intervall der Subscription (siehe unten) vererbt oder es wird individuell projektiert, um das Publishing-Intervall zu überschreiben. Das Sampling-Intervall hat in der Voreinstellung den Wert des Publishing-Intervalls.

- Der Monitoring-Modus

Mit dem Monitoring-Modus wird festgelegt, ob das Abfragen und die Übertragung von Notifications aktiviert oder deaktiviert ist.

- Der Filter

Über den Filter wird die Größe der Änderung definiert, ab der ein Wert oder Ereignis übertragen werden soll. Ein Filter kann auch die Properties des "EventType" eines Ereignisses filtern, wie z.B. EventID, EventType, SourceNode, Time und Description.

Auf Attribute werden keine Filter angewendet, da jede Änderung eines Attributs eine Notification erzeugt.

Der Filter stellt außerdem fest, ob ein Ereignis, das von einem Knoten abgeleitet wurde, an den Client gesendet wurde.

- Die Queue-Attribute

Über die Queue-Attribute kann die Reihenfolge der Übertragung an den Client festgelegt werden.

## Datenübertragung mit der Subscription

Eine Subscription wird verwendet, um Notifications an den Client zu übertragen.

Eine Subscription besitzt ein oder mehrere MonitoredItems, die ihr vom Client zugewiesen werden. MonitoredItems erzeugen Notifications, die von der Subscription in NotificationMessages gepackt werden. Ein oder mehrere NotificationMessages werden von einer Subscription an den Client übertragen.

Eine Subscription wird mit dem Dienst "CreateSubscription" erzeugt. Sie hat folgende wesentliche Merkmale:

- Publishing-Intervall

Eine Subscription hat ein Publishing-Intervall, das den Zyklus festlegt, in dem die Subscription aktiv wird. In diesem Publishing-Zyklus versucht die Subscription, eine NotificationMessage an den Client zu senden.

NotificationMessages enthalten Notifications, die noch nicht an den Client gesendet wurden.

- Antwort auf einen "Publish Request"

NotificationMessages werden als Antwort auf einen Publish Request an den Client gesendet. Sobald ein Publish Request vom Server empfangen wird, wird der Publish Request in die Warteschlange der Session eingetragen.

- Wenn eine Notification zur Übertragung ansteht, wird der Publish Request aus der Warteschlange entfernt und von der Subscription, die zur aktuellen Session gehört, bearbeitet.
- Wenn keine Notification zur Übertragung ansteht, dann wird der Publish Request nicht aus der Warteschlange der Session entfernt und der Server wartet bis zum nächsten Zyklus und prüft, ob eine Notification vorliegt.

Zu Beginn eines Zyklus, wenn Notifications vorliegen aber noch kein Publish Request, geht der Server in einen Wartezustand für Publish Requests. Sobald ein Publish Request empfangen wird, wird dieser unmittelbar bearbeitet, ohne das nächste Publishing-Intervall abzuwarten.

- Sequenznummer der NotificationMessage (fehlende Nachrichten)

Jede NotificationMessage besitzt eine individuelle Sequenznummer, die es Clients ermöglicht, das Fehlen von Nachrichten festzustellen.

- Keep-alive-Zähler

Subscriptions haben einen Keep-alive-Zähler, der die Anzahl der aufeinander folgenden Zyklen zählt, in denen keine Notification zur Übertragung anstand. Wenn der einstellbare Maximalwert des Zählers erreicht ist, dann wird der Publish Request aus der Warteschlange entfernt und dazu genutzt, eine Keep-alive-Nachricht zu versenden. Die Keep-alive-Nachricht informiert den Client, dass der Server noch aktiv ist.

Eine Keep-alive-Nachricht ist die Antwort auf einen Publish Request, in dem die NotificationMessage keine Notification sondern die Sequenznummer der NotificationMessage enthält, die als nächste gesendet wird.

- Den Dienst "Publishing" aktivieren

Der Dienst "Publishing" einer Subscription kann durch den Client aktiviert oder deaktiviert werden, wenn die Subscription angelegt wird. Alternativ kann "Publishing" auch über den Dienst "SetPublishingMode" aktiviert / deaktiviert werden.

Bei Deaktivierung schickt die Subscription keine NotificationMessages an den Client, sie wird aber weiterhin zyklisch aktiv und sendet Keep-alive-Nachrichten an den Client.

- Lifetime-Zähler

Subscriptions haben einen Lifetime-Zähler, der die Anzahl der aufeinander folgenden Publishing-Zyklen zählt, in denen kein Publish Request des Client vorlag. Wenn der Zähler den Wert erreicht, der für die Lebenszeit einer Subscription auf Basis des Parameters "MaxKeepAliveCount" des Dienstes "CreateSubscription" berechnet wurde, dann wird die Subscription geschlossen.

Durch das Schließen einer Subscription wird deren MonitoredItem gelöscht. Zusätzlich sendet der Server eine NotificationMessage "StatusChangeNotification" mit dem Status "code Bad\_Timeout".

- Quittieren der NotificationMessage und Notification-Puffer

Subscriptions besitzen einen Puffer für wiederholte Übertragung von NotificationMessages. NotificationMessages werden in diesem Puffer zurückgehalten, bis sie vom Client quittiert wurden, mindestens jedoch für 1 Keep-alive-Intervall.

## Der Dienst "Publish"

Der Dienst "Publish" dient zwei Zwecken:

- Aufforderung des Servers, eine NotificationMessage oder eine Keep-alive-Nachricht zu senden
- Quittierung des Empfangs von NotificationMessages für eine oder mehrere Subscriptions

Da Publish Requests nicht an spezifische Subscriptions adressiert werden, können sie von einer beliebigen Subscription verwendet werden.

### 3.5.2.6 Wie kann nach Anmeldung besonders schnell gelesen und geschrieben werden?

#### Wie kann nach Anmeldung besonders schnell gelesen und geschrieben werden?

Zum schnellen Lesen und Schreiben der Attribut-Werte von registrierten Knoten stehen nach der Registrierung der betreffenden Knoten die Methoden "Read(..,handle,..)" und "Write(..,handle,..)" zur Verfügung. Mit diesen Methoden ist ein Kurzaufzug der registrierten Knoten möglich, wodurch eine zeitsparende Datenübertragung ermöglicht wird. Der Zugriff erfolgt über die NodeID der registrierten Knoten.

Das Lesen / Schreiben wird in den folgenden Schritten durchgeführt:

1. RegisterNodes()
2. Read(..,handle,..)
- Write(..,handle,..)
3. UnregisterNodes()

Die Funktion der Methode "RegisterNodes" ist vergleichbar mit der Funktion der Methode "AddItems" bei OPC Data Access.

### 3.5.2.7 Wie funktionieren Events, Conditions und Alarme?

Dieser Abschnitt beschreibt Events, Conditions und Alarme.

## Events

Events beschreiben besondere Zustände im Prozess, die an einen Empfänger gemeldet werden müssen. Welche Ereignisse an den OPC-Client gemeldet werden, wird vom OPC-Client über Filterkriterien eingestellt.

Die OPC UA Events unterscheiden sich von den Events an der bisherigen COM-Schnittstelle OPC Alarne & Events darin, dass für OPC UA Events dieselben Zugriffstechniken und Schnittstellen verwendet wurden, wie bei UA Data Access.

## Conditions

Conditions sind abgeleitet von den allgemeinen Events. Conditions werden für die Darstellung des Status eines Systems oder einer seiner Komponenten verwendet. Einige Beispiele sind:

- eine Temperatur übersteigt einen konfigurierten Grenzwert.
- ein Gerät benötigt Wartung.
- ein Batch-Pozess, der für das Fortsetzen der weiteren Prozessschritte die Zustimmung des Anwenders benötigt.

Die primären Stati von Conditions sind "enabled" und "disabled". Der "disabled"-Status dient dazu, die Conditions über den Server auszuschalten. Der "enabled"-Status ist im Allgemeinen durch zusätzliche Sub-Stati erweitert.

Ein Wechsel in den "disabled"-Status resultiert in ein Condition-Ereignis. Jedoch werden dabei bis zum Wechsel in den "enabled"-Status keine darauffolgenden Ereignis-Meldungen generiert.

Wenn eine Condition in den "enabled"-Status eintritt, führen der Wechsel und alle darauf folgenden Wechsel zu Condition-Ereignissen, die vom Server generiert wurden.

Für den Status "enabled" ist die Condition von Bedeutung. Der OPC-Server und das Gerät bearbeiten die Condition.

Für den Status "disabled" ist Condition nicht von Bedeutung. Der OPC-Server und das Gerät müssen die Condition nicht bearbeiten, es gibt keine Ereignismeldungen "Events" zu dieser Condition.

## Quittierung

Der Typ AcknowledgeableConditionType ist abgeleitet vom Typ ConditionType. Er beinhaltet einen Sub-Status einer Condition, um darzustellen, ob eine Condition quittiert oder bestätigt werden muss.

## Alarne

Alarne sind eine spezielle Form von Conitions. Der AlarmConditionType ist eine besondere Form des AcknowledgeableConditionType, bei denen die Konzepte eines "enabled"-Status, eines zurückgestellten Status und eines unterdrückten Status zu einer Condition hinzukommen.

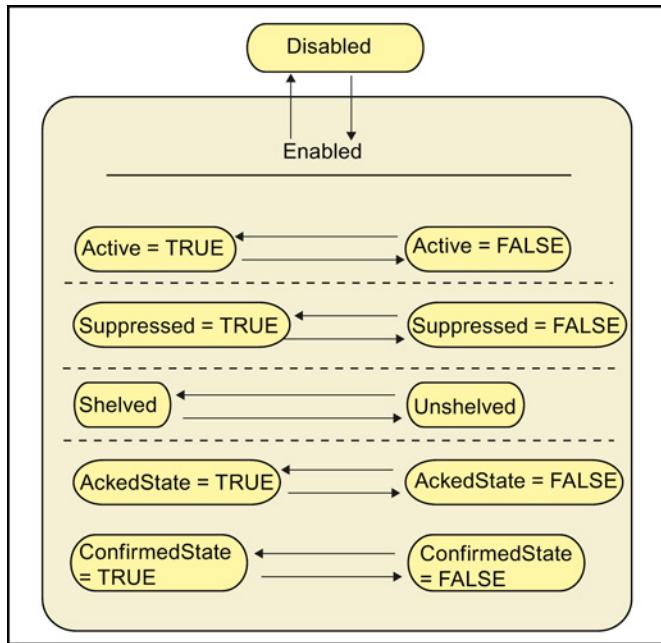


Bild 3-28 Status-Model für Alarne

Ein Alarm im "enabled"-Status zeigt an, dass der Zustand, den die Condition darstellt, derzeit besteht. Wenn ein Alarm "disabled" ist, wird dadurch dargestellt, dass der Zustand in einen normalen Status zurückgekehrt ist.

Einige Unterarten der Alarne führen Sub-Stati des "enabled"-Status ein. Ein Beispiel: Ein Alarm stellt eine Temperatur dar, die von einem High-Level-Status und einem Critically-High-Status bestimmt wird.

Die tatsächliche Quelle von OPC-Alarmen können die Meldungen einer SIMATIC S7 sein, siehe Kapitel "Meldungen bei SIMATIC S7, wie sind sie definiert? (Seite 102)

## Condition-Instanzen im Namensraum

Da Conditions immer einen "enabled"- oder "disabled"-Status und möglicherweise einige Sub-Stati haben, ist es sinnvoll Instanzen von Conditions zu haben, die einen Namensraum aufzeigen. Wenn der Server Condition-Instanzen darstellt, erscheinen diese im Namensraum als Komponenten der Objekte, die sie "besitzen". Zum Beispiel ein Temperatur-Fühler, der eine Hoch-Temperatur-Überwachung eingebaut hat, würde im Namensraum als eine Instanz von einigen Temperatur-Fühler-Objekten mit einer HasTypeDefinition-Referenz zu einem LimitAlarmType erscheinen. Das Temperaturfühler-Objekt hat eine HasCondition-Referenz zur Condition-Instanz. Zusätzlich kann ein Temperaturfühler-Objekt auch eine HasComponent-Referenz zur Condition-Instanz haben.

Clients für den Alarmzugriff richten eine Subscription auf das Attribut "EventNotifier" eines alarmfähigen Objekts ein, z.B. dem Temperaturfühler-Objekt und werden über alle Zustandswechsel informiert. Die Erreichbarkeit von Instanzen ermöglicht es Clients, den momentanen Status der Conditions zu überwachen, indem eine Subscription auf das Attribut "Value" auf die entsprechenden Properties einer Condition-Instanz eingerichtet wird. In diesem Fall ist es aber möglich, dass dieser Client nicht über alle Zustandswchsel informiert wird.

Auch wenn es nicht immer möglich ist Condition-Instanzen im Namensraum anzubieten, erlaubt diese Vorgehensweise einen direkten Zugriff (Lesen, Schreiben und Methoden-Aufruf) für die spezielle Condition-Instanz. Zum Beispiel, wenn eine Condition-Instanz nicht angezeigt wird, gibt es keine Möglichkeit die Aktivierungs- oder Deaktivierungs-Methode für die spezielle Condition-Instanz aufzurufen.

Weitere Information finden Sie im Kapitel OPC Alarms & Events (Seite 97)

### **Wie können Events, Conditions und Alarne empfangen werden?**

Ein UA-Client kann dieselben Techniken für den Empfang von Events einsetzen, wie beim Beobachten von Daten. Der Namensraum kann durchsucht werden, Nodes mit den Referenzen HasNotifier, HasEventSource und HasCondition zeigen an, dass Events oder Conditions zur Verfügung stehen können. Der UA-Client meldet dann eine Subscription auf diese Nodes an. Zum Empfang der Events muss anschließend ein passender Filter gesetzt werden.

#### **3.5.2.8 Wie kann Redundanz bei OPC UA verwendet werden?**

##### **Server-Redundanz**

Redundanz in OPC UA gewährleistet, dass Client und Server redundant sein können.

Bei der Server-Redundanz gibt es zwei Arten:

- transparent
- nicht-transparent

Bei transparenter Server-Redundanz ist die Ausfallsicherung der Server-Funktionen von einem Server zum anderen transparent zum Client: der Client ignoriert oder weiß nicht, dass ein Ausfall aufgetreten ist; der Client muss außer dem normalen Reconnect-Mechanismus auf SecureChannel-Ebene nichts tun, um den Datenfluss zu halten. Im Gegensatz dazu benötigt ein nicht-transparenter Redundanz-Server bei der Ausfallsicherung eine angemessene Reaktion des Clients.

Es gibt zwei spezielle Anforderungen an die Redundanz von Servern:

- Abgleich der Client- und Server-Informationen zwischen den Servern und
- Steuern der Umschaltung des Datenflusses von einem Server zum anderen bei Ausfall eines Servers.

## Transparente Server-Redundanz

Bei transparenter Server-Redundanz erstellt OPC UA eine Datenstruktur, die es dem Client ermöglicht zu identifizieren, welche Server verfügbar sind, auf welcher Service-Ebene sich jeder Server befindet und welcher Server derzeit eine bestimmte Session unterstützt. Jegliche OPC UA-Kommunikation innerhalb einer Session wird von einem Server unterstützt und der Client kann feststellen, welcher Server dies ist. Dies erlaubt es, die Daten vollständig zu protokollieren. Die transparenten Server sind dafür zuständig, dass Informationen zwischen ihnen abgeglichen werden und dass bei Ausfall eines Servers ein anderer Server Session und Subscriptions des Clients übernimmt. Eine Ausfallsicherung benötigt eine Wiederverbindung des Clients auf der Transportschicht. Dabei werden eine neue TCP/IP-Verbindung und ein neuer SecureChannel aufgebaut. Die URL des Endpunktes ändert sich dabei jedoch nicht.

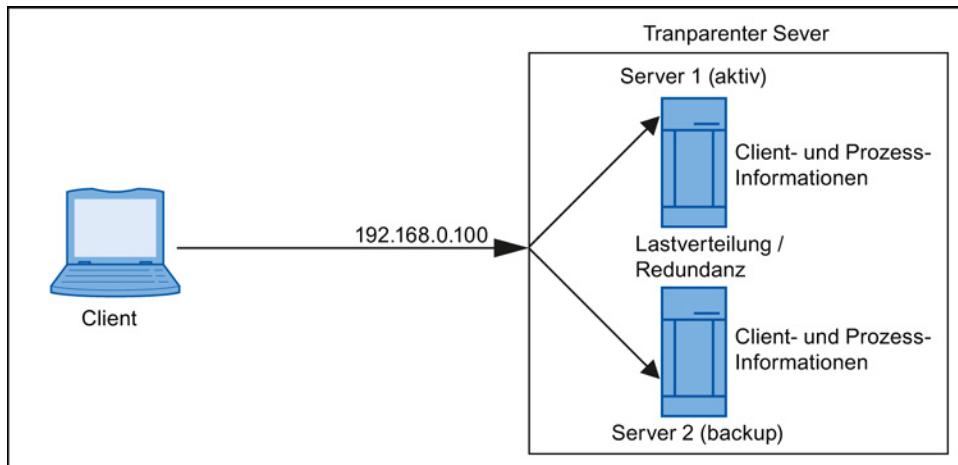


Bild 3-29 Transparente Server-Redundanz

Ein Client sieht nur die IP-Adresse des transparenten Servers (siehe Bild). Die Client-Information (Sessions, Zertifikate) und Prozess-Informationen werden automatisch vom Transparenten Server auf die physikalisch vorhandenen unterlagerten Server (Server 1 + 2) verteilt. Eine Lastverteilung zwischen den Servern 1 + 2 ist möglich. Bei Ausfall eines internen Servers (Server 1 oder 2) ist vom Client die normale Wiederverbindung der Transportschicht (meist im Client UA SDK abgewickelt) nötig.

### Nicht-transparente Server-Redundanz

Bei der nicht-transparenten Server-Redundanz stellt OPC UA die gleichen Daten-Strukturen und auch Server-Informationen zur Verfügung, die dem Client mitteilen, welche Arten der Ausfallsicherung der Server unterstützt. Diese Informationen ermöglichen es dem Client festzustellen, welche Aktivität notwendig ist, um die Ausfallsicherung auszuführen.

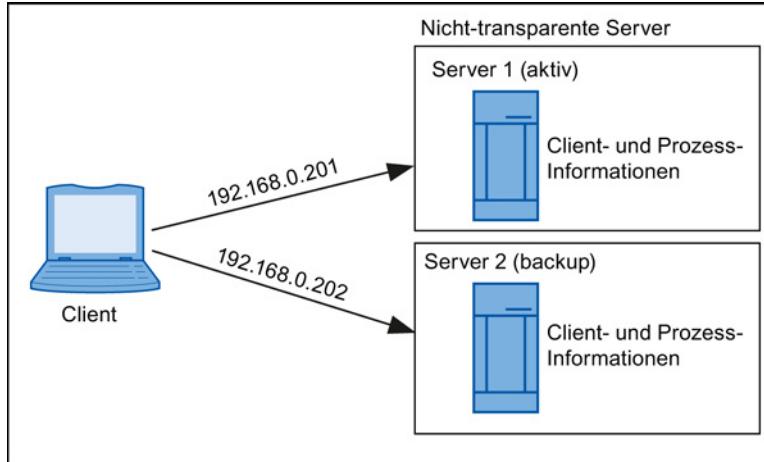


Bild 3-30 Nicht-transparente Server-Redundanz

Ein Client sieht mehrere IP-Adressen des Nicht-transparenten Servers (siehe Bild). Die Client-Informationen (Sessions, Zertifikate) sind mehrfach vorhanden. Die Prozess-Informationen der Server sind gleich. Eine Lastverteilung ist Aufgabe des Clients. Bei Ausfall eines Servers (Server 1 oder 2) muss der Client die komplette UA-Kommunikation von einem Server zum anderen wechseln.

Für nicht-transparente Redundanz bietet der Server zusätzliche Mechanismen für die:

- "cold"-Ausfallsicherung  
für Server, bei denen jeweils nur 1 Server aktiv sein kann.
- "warm"-Ausfallsicherung  
für Server, bei denen der Backup-Server keine Verbindung zu tatsächlichen unterlagerten Daten-Punkten des Prozessabbildes aufbauen kann.
- "hot"-Ausfallsicherung  
für Server, bei denen mehr als 1 Server aktiv und voll betriebsfähig ist.

## Client-Redundanz

Client-Redundanz wird von OPC UA durch den Aufruf TransferSubscriptions und durch die Bereitstellung der Client-Informationen in der Informations-Struktur des Servers unterstützt. Da die Lebenszeit der Subscription nicht an die Session gebunden ist, zu der sie erstellt wurde, können Backup-Clients die aktiven Client-Sessions über den Server überwachen, so als würden sie eine andere Datenvariable überwachen. Wenn der aktive Client in den inaktiven Zustand wechselt, schickt der Server ein Daten-Update zu jedem Client, der diese Variable überwacht hat. Wenn ein Backup-Client eine solche Meldung erhält, beauftragt er den Server die Subscription zu seiner eigenen Session zu übergeben. Wenn die Subscription sorgfältig, mit ausreichenden Ressourcen um Daten während des Übergangs zu puffern, angelegt wurde, entsteht bei der Ausfallsicherung des Clients kein Datenverlust. OPC UA unterstützt keinen standardisierten Mechanismus für die Übertragung der SessionId und SubscriptionIds vom aktiven Client zu den Backup-Clients. Aber solange wie die Backup-Clients den Client-Namen des aktiven Clients kennen, ist diese Information erhältlich, in dem die SessionDiagnostics- und SubscriptionDiagnostics-Anteile der ServerDiagnostic-Daten gelesen wird.

## 3.6 Leistungen von OPC Data Access und OPC Alarms & Events bei SIMATIC NET

### 3.6.1 Performance, wie kann sie optimal ausgenutzt werden?

#### Mit dem COM-inproc-Server können Sie die Performance verbessern

In einigen Anwendungsfällen, z.B. bei Verwendung von PC-basierten Steuerungen, sind extrem kurze Zugriffszeiten auf Prozessdaten erforderlich.

Der Verwendung von OPC in COM-basierter Client-Server-Architektur sind jedoch je nach Ausprägung des OPC-Servers bestimmte interne Laufzeiten vorgegeben.

Diese entstehen vor allem bei Verwendung eines Local Servers (auch "Out-Process-Server" genannt; EXE-Datei mit eigenem Prozessraum) durch Prozesswechsel und Übermittlung der Funktionsparameter vom Client zum Server (sog. Marshalling) und zurück.

Bei Ausprägung des OPC-Servers als In-Process-Server fallen die Laufzeiten für Prozesswechsel und Marshalling weg, da der OPC-Server als dynamisch ladbare Bibliothek (DLL) implementiert ist und im Prozessraum des Clients abläuft.

Die Verwendung eines In-Process-Servers hat jedoch Nachteile, die bei der Auswahl des Servers berücksichtigt werden müssen:

Nur 1 Client kann den Server zu einer Zeit benutzen. Die gleichzeitige Benutzung des In-Process-OPC-Servers durch mehrere Clients würde eine mehrfache Erzeugung des Servers in verschiedenen Prozessräumen bewirken, die jedoch alle gleichzeitig und nicht koordiniert auf die gleiche Hardware zugreifen würden. Die Folge wäre, dass nur der zuerst gestartete Client Zugriff auf die Prozessdaten hat, der Zugriff weiterer Clients jedoch abgewiesen würde.

### *3.6 Leistungen von OPC Data Access und OPC Alarms & Events bei SIMATIC NET*

Die Stabilität des OPC-Servers ist vom Client abhängig. Verhält sich der OPC-Client unkontrolliert z.B. bei Zugriffsverletzungen, ist unweigerlich der OPC-Server mit betroffen. Die Folge wäre, dass das ggf. notwendige Rücksetzen der Kommunikationsbaugruppe durch den OPC-Server nicht mehr erfolgen kann. Auch ein explizites Beenden des OPC-Servers über die Konfigurationsprogramme wäre nicht möglich.

Für das sehr schnelle DP-Protokoll bietet SIMATIC NET einen In-Process-Server, der die Leistungsfähigkeit des DP-Protokolls auch OPC-Clients nahezu ungeschmälert bereitstellt.

#### **Performanceverbesserungen auch bei mehreren Clients?**

Auch das ist möglich. Wie im vorangegangenen Abschnitt beschrieben, kann ein hochperformanter Inproc-Server nur von einem Client genutzt werden. Um bei höheren Performanceanforderungen die gleichzeitige Nutzung von zwei oder mehr Clients zu ermöglichen, wird eine weitere Konfigurationsviante angeboten. Hierfür werden jeweils die unterlagerten DP-, SR- oder S7-Protokollbibliotheken und der COM-Server als Inproc-Server in den Outproc-OPC-Server geladen. Die Protokollbearbeitung läuft im Prozess des OPC-Servers ab, weitere Laufzeiten für Prozesswechsel und Multiprotokollbetrieb fallen weg. Der Prozesswechsel zwischen OPC-Client und OPC-Server ist allerdings noch vorhanden.

#### **Mehr Vorteile als Nachteile**

Die Verwendung des performanten DP-, S7- und SR-OPC-Servers bietet Vorteile:

- Höhere Performance als beim Multiprotokollbetrieb.
- Einfache Konfiguration.
- Zugang über die ProgID OPC.SimaticNET.
- Mehrere Clients können den Server zur gleichen Zeit nutzen.
- Die Stabilität des OPC-Servers ist nicht von den Clients abhängig.

Jedoch auch einen Nachteil:

- Bei Verwendung des performanten DP-, S7- oder SR-OPC-Servers ist jeweils nur der Einzelprotokollbetrieb möglich.

#### **Wie kann die performante Variante aktiviert werden?**

Die Aktivierung dieser performanten Variante erfolgt implizit durch die alleinige Auswahl des DP-, S7- oder SR-Protokolls im Konfigurationsprogramm "PC-Station einstellen".

### 3.6.2 OPC-Server von SIMATIC NET in der Automatisierungswelt, wie wird er eingesetzt?

#### Einsatzmöglichkeiten des OPC-Servers in der Automatisierungswelt

Das Bild zeigt, wie vielseitig der OPC-Server von SIMATIC NET eingesetzt werden kann.

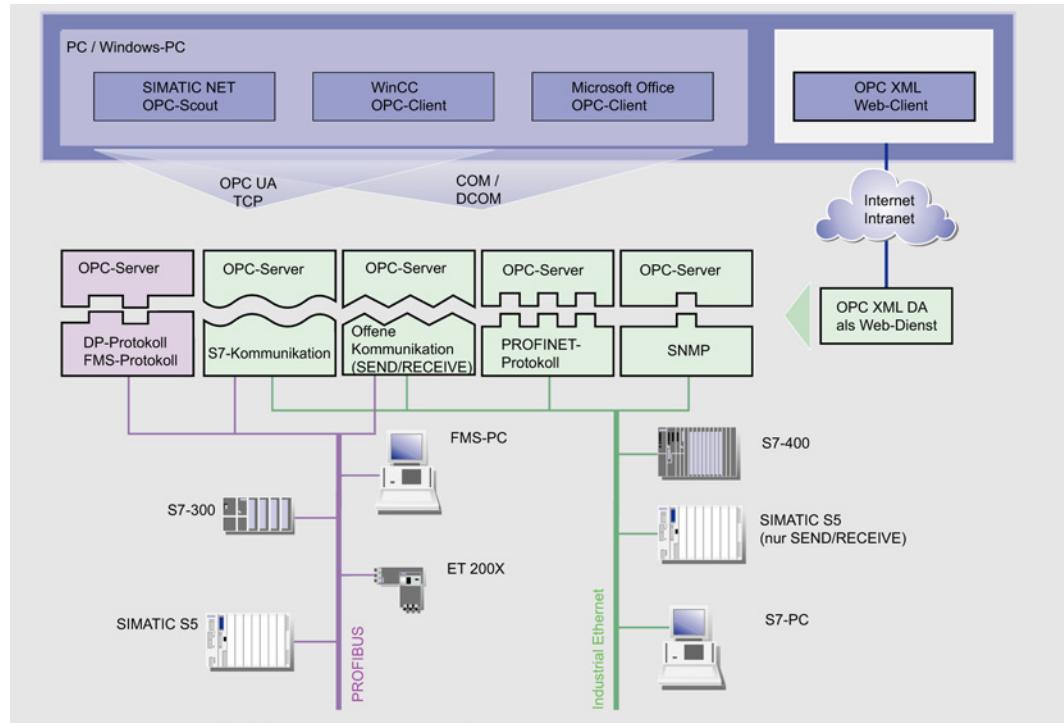


Bild 3-31 Systemintegration mit OPC-Server

### 3.6.3 OPC-Server für SIMATIC NET, was sind die Vorteile?

#### OPC-Server, die Vorteile liegen auf der Hand

Mit den Leistungen von OPC bietet SIMATIC NET zusammenfassend eine Reihe von Vorteilen allgemeiner Art sowie im Hinblick auf Programmierung und Entwicklung individueller Clients. Aber auch das Personal zur Inbetriebnahme kommt nicht zu kurz.

Die offene OPC-Schnittstelle ist die zentrale Schnittstelle der Produkte auf dem PG/PC von SIMATIC NET. Der OPC-Server von SIMATIC NET unterstützt alle Kommunikationsprotokolle und Dienste, die von den Kommunikationsbaugruppen bereitgestellt werden.

Der OPC-Server von SIMATIC NET unterstützt für alle Protokolle die Schnittstellenspezifikation OPC Data Access. Für Protokolle, die über Mechanismen zur Übermittlung von Ereignissen (S7-Kommunikation und SNMP) verfügen, wird auch OPC Alarms & Events unterstützt.

Im Folgenden erfahren Sie, wie der OPC-Server von SIMATIC NET eingesetzt wird, welche Vorteile seine Benutzung hat und welche Eigenschaften gelten. Außerdem sehen Sie, wie Sie mit OPC-Server optimal auf Prozessdaten zugreifen können.

Der OPC-Server von SIMATIC NET ermöglicht den Zugang zu den industriellen Kommunikationsnetzen PROFIBUS und Industrial Ethernet von SIMATIC NET. Er stellt OPC-Clients die Werte von Prozessvariablen zur Verfügung oder meldet Ereignisse vom Partnergerät. Dazu greift er mit Hilfe der Protokoll-Software und des Kommunikationsprozessors von SIMATIC NET über das Kommunikationsnetz auf die Partnergeräte zu.

### **Vorteile für die Inbetriebnahme**

- Sie verwenden eine protokollunabhängige Schnittstelle, d.h. es muss für mehrere Applikationen nur eine Schnittstelle installiert werden.
- Sie haben einen einfachen Zugang zu den Kommunikationsnetzen von SIMATIC NET.
- Über das Kommunikationsnetz von SIMATIC NET können Sie Ihre Automatisierungssysteme mit einer Vielzahl von Applikationen der Automatisierungstechnik verwenden.
- Sie können Microsoft Office Produkte integrieren.
- Über DCOM können auch Applikationen, die auf anderen Rechnern installiert sind, über globale oder lokale Netze auf die Leistungen des OPC-Servers zugreifen.
- Mit der OPC-Client-Applikation "OPC-Scout" von SIMATIC NET steht Ihnen ein leistungsfähiges Werkzeug zum einfachen Zugriff auf Prozessvariablen zur Verfügung.
- Mit Hilfe von SIMATIC Computing können Sie beispielsweise in Visual Basic schnell einfache Hilfsprogramme erstellen.

### **Vorteile für die Programmentwicklung**

- Sie arbeiten mit einer herstellerunabhängigen Schnittstelle. Das bietet Ihnen Zukunftssicherheit. So können Sie einen größeren Markt bedienen und können Ihre Entwicklungen wiederverwenden.
- Die entwickelten Applikationen sind unabhängig vom Kommunikationssystem eines Herstellers und können unverändert mit OPC-Servern verschiedenster Hersteller kommunizieren.
- Mit den OPC-Schnittstellen steht den Applikationen ein leistungsfähiger Zugang zu OPC-Servern und den unterlagerten Kommunikationsnetzen zur Verfügung.
- OPC bietet hochperformante Schnittstellen für die Programmiersprachen C und C++.
- Ein komfortabler und einfacher Zugriff auf Prozessdaten ist in einer Entwicklungsumgebung wie z.B. Visual Basic über das Data Control möglich.
- Sie müssen sich nicht in protokoll- und herstellerspezifische Schnittstellen einarbeiten.
- Durch die Möglichkeit einer Trace-Ausgabe wird die Fehlersuche vereinfacht.
- Dadurch dass ein Partnergerät simuliert werden kann, kann die Programmentwicklung ohne die Installation zusätzlicher Geräte erfolgen.

## Welche Einschränkungen hat der OPC-Server?

Der OPC-Server von SIMATIC NET unterstützt alle geforderten Schnittstellen der Spezifikationen für OPC Data Access und OPC Alarms & Events. Weiterhin stellt er die wichtigsten optionalen Schnittstellen zur Verfügung, wie beispielsweise das Browsing Interface für OPC Data Access.

Für die optionalen Schnittstellen gelten folgende Einschränkungen:

- Der OPC-Server für Data Access bietet keine Unterstützung von "OPC Public Groups".
- Der OPC-Server für Data Access erlaubt nicht das Schreiben von Zeitstempeln und Qualitäten.
- OPC-Server für Alarms & Events
  - Unterteilung der Anlagen in Anlagenbereiche (Areas) ist nicht möglich
  - Untersuchung von Anlagenbereichen mittels Browsing ist nicht möglich

Welche Art und welchen Inhalt eine Meldung hat, ist durch die OPC-Spezifikation nicht festgelegt. Über Projektierungsinformation kann festgelegt werden, ob Alarne als Simple Events oder Conditional Events gemeldet werden sollen.

## 3.6.4 OPC-Server von SIMATIC NET, was leistet er?

### Das leistet der OPC-Server von SIMATIC NET

Die offene OPC-Schnittstelle ist die zentrale Schnittstelle der Produkte auf dem PG/PC von SIMATIC NET. Der OPC-Server von SIMATIC NET unterstützt alle Kommunikationsprotokolle und Dienste, die von den Kommunikationsbaugruppen bereitgestellt werden.

Der OPC-Server von SIMATIC NET unterstützt für alle Protokolle die Schnittstellenspezifikation OPC Data Access. Für Protokolle, die über Mechanismen zur Übermittlung von Ereignissen (S7-Kommunikation) verfügen, wird auch OPC Alarms & Events unterstützt.

Im Folgenden erfahren Sie, wie der OPC-Server von SIMATIC NET eingesetzt wird, welche Vorteile seine Benutzung hat und welche Eigenschaften gelten. Außerdem sehen Sie, wie Sie mit OPC-Server optimal auf Prozessdaten zugreifen können.

Der OPC-Server von SIMATIC NET ermöglicht den Zugang zu den industriellen Kommunikationsnetzen PROFIBUS und Industrial Ethernet von SIMATIC NET. Er stellt OPC-Clients die Werte von Prozessvariablen zur Verfügung oder meldet Ereignisse vom Partnergerät. Dazu greift er mit Hilfe der Protokoll-Software und des Kommunikationsprozessors von SIMATIC NET über das Kommunikationsnetz auf die Partnergeräte zu (siehe Bild).

### 3.6 Leistungen von OPC Data Access und OPC Alarms & Events bei SIMATIC NET

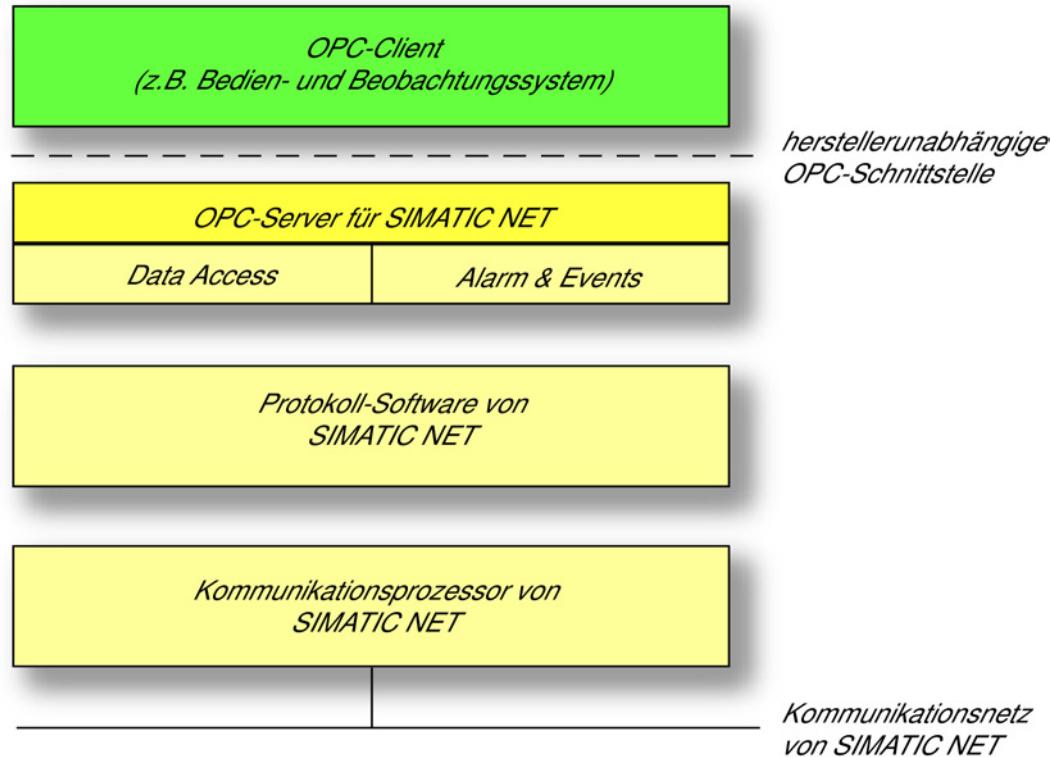


Bild 3-32 OPC-Server für SIMATIC NET mit OPC-Client

#### Welche besondere Leistungen erbringt der OPC-Server?

Weil der OPC-Server in der Lage ist, Aufträge an verschiedene Kommunikationssysteme zu verteilen, kann der OPC-Client über einen einzigen OPC-Server mehrere unterschiedliche Protokolle gleichzeitig benutzen. Wird in der Projektierung des OPC-Servers nur 1 Protokoll verwendet, ist keine Verteilung notwendig. Dadurch wird der Datendurchsatz optimiert.

#### 3.6.5 Prozessdaten, wie wird optimal darauf zugegriffen?

##### Verschiedene Arten um auf Prozessdaten zuzugreifen

Mit OPC Data Access können Sie auf verschiedene Arten auf Prozessdaten zugreifen. Durch die Wahl der geeigneten Methoden können Sie den Datendurchsatz Ihrer Anwendung beeinflussen.

Bei einigen Protokollen haben Sie auch durch die Auswahl eines Dienstes und durch die Strukturierung der Variablen im Namensraum Einfluss auf die Leistungsfähigkeit des OPC-Servers.

### Tipps für einen optimalen Datendurchsatz gefällig?

Die Informationen in den folgenden Unterkapiteln:

- Mengenoperationen verwenden
- Auf OPC-Cache zugreifen
- Items strukturieren
- Blockdienste verwenden

helfen Ihnen den größtmöglichen Datendurchsatz zu erzielen

### Welche Methoden sind die geeignetsten?

Es gibt verschiedene Zugriffsmöglichkeiten auf Variablen. Da sie unterschiedliche Eigenschaften haben, beachten Sie die im Folgenden beschriebenen Vor- und Nachteile und wählen Sie die Zugriffsmöglichkeit, die für Ihren Fall am geeignetsten ist.

Das Unterkapitel

- Methoden, wie werden die geeigneten verwendet?  
hilft Ihnen die richtige Entscheidung zu treffen.

### 3.6.6 Mengenoperationen, wie werden sie verwendet?

#### So werden Mengenoperationen verwendet

Bei vielen Methoden können Sie mit einem Funktionsaufruf mehrere Prozessvariablen in einem Feld als Parameter übergeben. Die Verwendung von Mengenoperationen ist vorteilhaft für den Datendurchsatz, da weniger Funktionsaufrufe und Prozesswechsel zwischen OPC-Client und OPC-Server stattfinden. Der OPC-Server kann so selbst die Kommunikation über das Netz optimieren, indem er beispielsweise einzelne Aufträge zusammenfasst.

Wenn Sie mit einem Remote Server über DCOM arbeiten, ist die Verwendung von Mengenoperationen besonders wirksam, da in diesem Fall ein Funktionsaufruf über das Netz transportiert wird.

### 3.6.7 OPC-Cache, was ist das?

#### Das ist OPC-Cache

Der OPC-Cache ist ein interner Zwischenspeicher des OPC-Servers, in dem die zuletzt erfassten Werte der OPC-Items gespeichert werden.

### **OPC-Cache, wie funktioniert er?**

Der OPC-Server aktualisiert alle aktiven Items, die in aktive Gruppen eingefügt sind und legt die gelesenen Werte im Cache ab. Voraussetzung für gültige Werte im Cache ist, dass das Item erfolgreich gelesen wurde.

Das Lesen einer Variablen aus dem Cache läuft deutlich schneller ab, als der Zugriff auf das Device. Wenn die Werte im Cache für den Anwendungsfall in einem genügend kleinen Zeitraum aktualisiert werden, sollten Sie folglich auf den Cache zugreifen.

Die Aktualisierungsrate im Cache wird durch den Parameter RevisedUpdateRate festgelegt.

### **3.6.8 MaxAge, was ist das?**

#### **MaxAge, was ist das?**

Mit OPC Data Access 3.00 kann zwischen dem Lesen vom Cache und vom Device feiner abgestuft werden.

MaxAge ist das gewünschte maximale Alter eines Werts in Millisekunden bis zur nächsten Aktualisierung vom Device. Wenn diese Zeitdauer beim Leseauftrag nicht überschritten ist, dann wird vom Cache zurückgemeldet. Wenn die Zeit überschritten ist, dann muss neu vom Device gelesen werden.

Ein MaxAge-Wert = 0 entspricht dem Lesen vom Device (OPC\_DS\_DEVICE) und ein MaxAge-Wert = 1 bis 0xFFFFFFFF (49,7 Tage) dem Lesen von Cache (OPC\_DS\_CACHE).

### **3.6.9 Dienste wenden den Cache an, wie geschieht das (Beispiel)?**

#### **Beispiele zur Anwendung von Cache**

Hier sehen Sie Beispiele für Dienste, die auf den Cache angewendet werden können:

#### **IOPCSyncIO::Read(...,OPC\_DS\_CACHE,...)**

Von mehreren OPC-Items werden synchron Wert, Zeitstempel und Qualität aus dem Cache gelesen.

#### **IOPCAsyncIO2::Refresh(...,OPC\_DS\_CACHE,...)**

Für alle aktiven OPC-Items wird unabhängig vom Wert ein Rückruf (Callback) im OPC-Client erzeugt. Der im Cache gespeicherte Wert wird als aktueller Wert an den Client gesendet.

#### **IOPCAsyncIO3::ReadMaxAge(...,MaxAge=500,...)**

Für alle OPC-Items der Gruppe wird unabhängig vom Wert ein Rückruf (Callback) im OPC-Client erzeugt. Der im Cache gespeicherte Wert (sofern vorhanden) wird als aktueller Wert an den Client gesendet, falls die Daten nicht älter als 500 Millisekunden sind.

### 3.6.10 Protokolle, für welche ist eine Optimierung möglich?

#### Für diese Protokolle ist eine Optimierung möglich

Für die Variabeldienste folgender Protokolle bietet der OPC-Server für SIMATIC NET einen Optimierungsalgorithmus:

1. S7-Protokoll
2. Offene Kommunikationsdienste (SEND/RECEIVE) über Industrial Ethernet

Mehrere gleichzeitige Zugriffsaufträge auf einzelne Variablen werden intern in einen einzigen Zugriff auf das Partnergerät umgewandelt. So wird die Anzahl der über das Netz transportierten Datenpakete reduziert, die Auslastung der einzelnen Pakete verbessert und der Anteil der Nutzdaten eines Pakets erhöht.

Diese Optimierung gilt sowohl für Lese- als auch für Schreibzugriffe und ist standardmäßig aktiviert. Für den OPC-Client ist dieser Optimierungsalgorithmus unsichtbar.

#### Welche Regeln gelten für die Anordnung der OPC-Items im Namensraum?

Damit die Optimierung auch stattfinden kann, müssen Sie für die Anordnung der OPC-Items im Namensraum folgende Regeln beachten:

- OPC-Items, die gleichzeitig gelesen oder beobachtet werden, **sollten** im Namensraum des Partnergeräts aufeinanderfolgend angeordnet sein.  
Kleinere Lücken zwischen den relevanten Teilen werden zwar verarbeitet, verschlechtern jedoch den Datendurchsatz.
- OPC-Items, die gleichzeitig geschrieben werden, **müssen** im Namensraum aufeinanderfolgend angeordnet sein.  
Wenn der Schreibzugriff optimal erfolgen soll, dürfen keine Lücken vorhanden sein. Das durch die Optimierung gebildete Feld wird auf jeden Fall vollständig zum Partnergerät übertragen. Der Adressbereich der Lücken würde mit undefinierten Werten überschrieben. Damit das nicht erfolgt, setzt der OPC-Server wieder einzelne Zugriffsaufträge ohne Optimierung ab.

### 3.6.11 Blockdienste, wozu werden sie verwendet?

#### Blockdienste werden verwendet

Für die Übertragung von großen Datenpaketen bieten die S7-Kommunikation und die offenen Kommunikationsdienste (SEND/RECEIVE) über Industrial Ethernet und PROFIBUS Blockdienste an. Dabei werden Datenpakete zwischen den Kommunikationspartnern versendet. Die Übermittlung der Daten belastet das Netz nur dann, wenn ein Partner explizit einen Sendeauftrag absetzt.

Mit dem OPC-Server für SIMATIC NET können Sie die Datenblöcke strukturieren. So können einzelne Teile des Datenpakets OPC-Items zugeordnet werden.

### 3.6.12 Blockdienste, wie werden Sie verwendet (Beispiel)?

#### Beispiel zur Verwendung von Blockdiensten

Die folgende Abbildung zeigt, wie ein S7-400 Gerät ein Datenpaket an eine PC-Station mit S7-OPC-Server sendet.

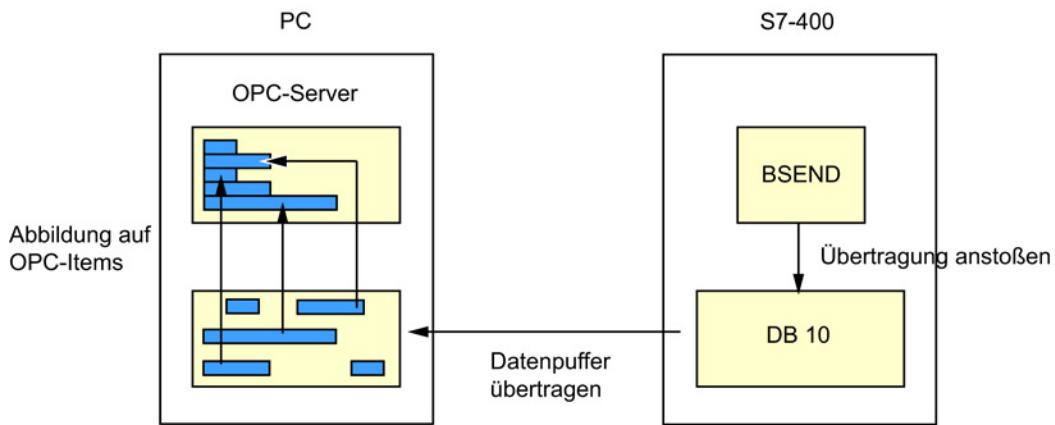


Bild 3-33 Senden eines Datenpakets

Damit der OPC-Server Daten empfangen kann, werden im PC ein oder mehrere OPC-Items des Typs BRCV in eine aktive Gruppe eingefügt.

Im S7-400 Gerät stößt ein Anwendungsprogramm den Funktionsbaustein BSEND an. BSEND startet die Übertragung des gesamten Datenbereichs als Puffer zum PC.

Im PC werden die empfangenen Daten an den OPC-Server übergeben. Der OPC-Server bildet nun die Teilbereiche des Datenblocks auf die entsprechenden OPC-Items ab. Wenn diese OPC-Items beobachtet werden, sendet der OPC-Server bei Änderung der Werte einen Rückruf an den OPC-Client.

### 3.6.13 Methoden, wie werden die geeigneten verwendet?

#### 3.6.13.1 Synchrone Zugriffe, welche gibt es?

##### Möglichkeiten für synchrone Zugriffe

Für OPC Data Access gibt es zwei Möglichkeiten, auf Daten zuzugreifen:

##### Synchrones Lesen und Schreiben

Sowohl der lesende, als auch der schreibende Zugriff kann synchron erfolgen.

Ein Programm setzt einen synchronen Funktionsaufruf zum Zugriff auf Prozessdaten ab. Wenn die Funktion abläuft, wickelt der OPC-Server die vollständige Kommunikation über das Netz ab. Mit den Rückgabeparametern der Funktion werden die gelesenen Werte an das Anwendungsprogramm übergeben, das den Ablauf nun mit den nachfolgenden Anweisungen fortsetzen kann.

### Anwendungsbereich von synchronen Zugriffen

Sie sollten immer dann synchrone Zugriffe verwenden, wenn eine längere Unterbrechung des Anwendungsprogramms eine untergeordnete Rolle spielt. Der synchrone Zugriff bietet immer den schnellstmöglichen Zugriff auf Daten des Partnergerätes.

### Vor- und Nachteile von synchronem Zugriffen

Vorteile:

- einfache Programmierung
- hoher Datendurchsatz, da nur ein Prozesswechsel pro Auftrag zwischen OPC-Client und OPC-Server stattfindet.

Nachteil:

- Die Anwendung wird so lange unterbrochen, bis der synchrone Auftrag verarbeitet ist. Erst wenn alle Daten gelesen sind, kann die Anwendung fortsetzen. Wenn die Funktion nicht in einem eigenen Thread aufgerufen wird, wird z.B. die Benutzeroberfläche einer interaktiven Anwendung während des Funktionsaufrufs blockiert.

### 3.6.13.2 Asynchrone Zugriffe, welche gibt es?

#### Möglichkeiten für asynchrone Zugriffe

Für OPC Data Access gibt es folgende Möglichkeiten, auf Daten zuzugreifen:

#### Asynchrones Lesen und Schreiben

Asynchron kann lesend oder schreibend zugegriffen werden.

Ein Programm setzt einen asynchronen Funktionsaufruf zum Zugriff auf Prozessdaten ab. Das Programm erhält sofort eine Rückmeldung, ob der Auftrag erfolgreich an den OPC-Server abgegeben ist. Dann arbeitet das Programm weiter.

Der OPC-Server hat dem Auftrag eine TransactionID zugeteilt, über die der Client später die Antwort zu diesem Auftrag identifizieren kann.

Zu einem späteren, undefinierten Zeitpunkt ruft der OPC-Server die Funktion (AsyncReadComplete) oder (AsyncWriteComplete) des OPC-Clients auf. Als Aufrufparameter werden dem Client die Ergebnisse des vorherigen Funktionsaufrufs (lesend oder schreibend) und die TransactionID übermittelt.

Die Übermittlung der Daten über das Netz erfolgt zeitlich unabhängig vom Programmablauf des OPC-Clients.

## Anwendungsbereich von asynchronen Zugriffen

Asynchrone Zugriffe sind dann sinnvoll, wenn große Datenmengen gelesen werden müssen und gleichzeitig das Anwendungsprogramm während der Auftragsbearbeitung reaktionsfähig sein soll.

Asynchrone Zugriffe auf den Cache des OPC-Servers sind nicht sinnvoll.  
Hierbei entsteht nur durch Prozessübergänge zwischen OPC-Client und OPC-Server eine hohe Prozessorbelastung.

## Vor- und Nachteile von asynchronem Zugriff

Vorteile:

die eigene Anwendung wird nur kurz unterbrochen, weil die eigentliche Kommunikation parallel zu der Anwendung läuft.

Nachteile:

- die Erstellung der Anwendung ist nicht ganz so leicht. In der Anwendung muss ein Rückrufmechanismus (Callback) implementiert sein, der zu jedem Zeitpunkt das Ergebnis der Auftragsbearbeitung entgegennehmen kann.  
Windows-Programme verfügen standardmäßig über asynchrone Mechanismen, damit sie auf Eingaben des Anwenders reagieren können.
- bei der Übergabe weniger Variablen in einem Auftrag fällt eine große Belastung durch die Prozessübergänge beim Aufruf und beim Rückruf an. Es sind doppelt so viele wie bei synchronen Zugriffen.

### 3.6.13.3 Variablen beobachten, was geschieht da?

#### Variablen beobachten

Beim Beobachten von Variablen prüft der OPC-Server fortlaufend, ob sich der Wert oder die Qualität von Variablen geändert hat.

Zu diesem Zweck fügt der OPC-Client aktive OPC-Items einer Gruppe hinzu und aktiviert die Gruppe. Es werden dann alle aktiven OPC-Items in allen aktiven Gruppen beobachtet.

Der OPC-Client stellt die Funktion OnDataChange() bereit. Der OPC-Server ruft diese Funktion auf, wenn eine Änderung der Werte stattgefunden hat. Der OPC-Server übergibt als Parameter die geänderten Werte, Qualitäten und Zeitstempel der OPC-Items.

Der OPC-Client wird durch die Beobachtung der Variablen nicht belastet. Erst wenn eine Änderung erkannt wird, wird das Programm des Clients ausgeführt.

Damit der OPC-Client bei sich schnell ändernden Prozessvariablen nicht mit Änderungsmeldungen überlastet wird, können Sie über den gruppenspezifischen Parameter RequestedUpdateRate bzw. RevisedUpdateRate vorgeben, mit welcher minimalen Aktualisierungsrate er aufgerufen werden soll.

Rauschende Analogwerte würden zu häufigen Änderungsmeldungen führen, weil sich der Wert laufend geringfügig ändert. Durch den Parameter DefaultGroupDeadBand wird für alle Items einer Gruppe durch eine Prozentangabe ein Bereich definiert, in dem Änderungen nicht gemeldet werden. Die absolute Größe des Bereichs ist der prozentuale Anteil der Differenz zwischen einer projektierten Ober- und Untergrenze.

Voraussetzung dafür ist, dass der Wertebereich der Variablen im Symbol-Editor definiert wurde.

### **Wann sollten Variablen beobachtet werden ?**

Das Beobachten von Variablen ist die optimale Lösung, wenn ein Programm stets einen aktuellen Datenbestand des Prozesses oder eines Teils des Prozesses benötigt.

### **Vor- und Nachteile vom Beobachten von Variablen**

Vorteile:

- die Anwendung wird nur benachrichtigt, wenn sich die Prozessdaten geändert haben. Dadurch ergibt sich eine geringere CPU-Belastung.
- hoher Datendurchsatz, weil wenig Prozesswechsel stattfinden. Je nach Zusammensetzung der Item-Struktur ist eine gute Optimierung möglich.
- die Beobachtung der Variablen kann item- und gruppenweise durch den Client ein- und ausgeschaltet werden.
- Windows-Programme verfügen standardmäßig über asynchrone Mechanismen, damit sie auf Eingaben des Anwenders reagieren können.

Nachteile:

- die Reaktionszeit von der Änderung eines Wertes im Prozess bis zur Übergabe des neuen Wertes an den Client ist größer als die Aktualisierungsrate der Gruppe.
- die Erstellung der Anwendung ist nicht ganz so leicht. Die Anwendung benötigt einen asynchronen Teil zum Empfang von Werteänderungen.

### **Die Aktualisierungszeit (RevisedUpdateRate)**

Der über das Anwendungsprogramm einstellbare Parameter "Aktualisierungszeit" (RequestedUpdateRate/RevisedUpdateRate) legt das kleinstmögliche Zeitintervall für eine Überprüfung der Werte der OPC-Items einer aktiven OPC-Gruppe fest.

Der Server prüft, ob sich der Wert geändert hat. Wenn ein neuer Wert vorliegt, meldet der Server dem Client den neuen Wert. Die Meldungen an den Client erfolgen dabei nicht schneller, als die vom Client vorgesehene "RevisedUpdateRate". Ändert sich ein Wert schneller als in der "Update Rate" angegeben, wird der Client über Zwischenwerte nicht informiert!

### **Die Zykluszeit**

Die Zykluszeit in ms legt fest, wie oft der OPC-Server die Werte der OPC-Items über einen neuen Kommunikationsauftrag aktualisiert.

### **Der Zusammenhang zwischen Zykluszeit und Aktualisierungszeit (UpdateRate)**

Die vom OPC-Server für SIMATIC NET verwendeten Aktualisierungsrationen (RevisedUpdateRate) sind Vielfache der hier bei der Projektierung festgelegten Zykluszeit. Die kleinste zulässige Aktualisierungsrate (minimale UpdateRate) entspricht der Zykluszeit.

## Der Zusammenhang zwischen den protokollspezifisch eingestellten Zykluszeiten

Da der SIMATIC NET OPC-Server Variablen verschiedener Protokolle gleichzeitig verwenden kann, ergibt sich die minimale UpdateRate des OPC-Server als der kleinste Wert der für die aktiven Protokolle projektierten Zykluszeit.

### 3.6.14 Percent Deadband, wie wird dieser Parameter verwendet?

#### So wird Percent Deadband verwendet

Der Parameter "Percent Deadband" definiert für ein Item einen Bereich, in dem Änderungen des Werts nicht gemeldet werden. Die absolute Größe des Bereichs ist der prozentuale Anteil der Differenz zwischen einer projektierten Ober- und Untergrenze.

Für das folgende Beispiel gilt: Percent Deadband = 10% = 1 Einheit (bei Obergrenze = 10 und Untergrenze = 0).

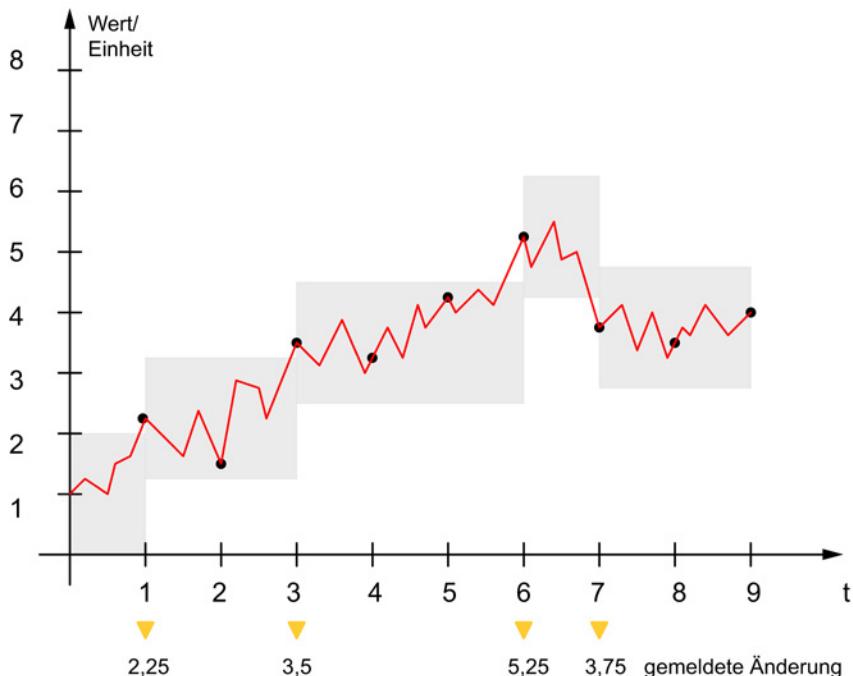


Bild 3-34 Bereich in dem Änderungen nicht gemeldet werden

Nach jeder an den Client gemeldeten Wertänderung wird ein Band um den zuletzt gemeldeten Wert gelegt. Erst wenn bei einem Lesezugriff dieses Band beim Beobachten des Werts über- oder unterschritten wurde, wird eine Änderungsmeldung an den Client erzeugt und ein neues Band festgelegt.

Über die mit OPC Data Access Version 3.00 eingeführte Schnittstellenfunktion kann der Parameter "Percent Deadband" innerhalb einer Gruppe Item-spezifisch vorgegeben werden.

### 3.6.15 Aktualisierungszeit, wie wird sie Item-spezifisch eingesetzt?

#### Item-spezifische Einstellung der Aktualisierungszeit ab Version 3.00 der Data Access-Spezifikation

Die Aktualisierungszeit für die Beobachtung von Variablen wird über die Parameter "RequestedUpdateRate" und "RevisedUpdateRate" eingestellt. Die Aktualisierungszeit gilt jeweils für eine oder mehrere aktive OPC-Gruppen.

Darüber hinaus können Sie ab der Data Access-Spezifikation 3.00 im OPC-Client item-spezifische Aktualisierungszeiten mit der optionalen Schnittstelle "IOPCItemSamplingMgt" einstellen. Diese können größer oder kleiner der Gruppen-Aktualisierungszeit sein (RequestedUpdateRate / RevisedUpdateRate). Damit kann die Aktualisierungszeit einzelner Items innerhalb einer OPC-Gruppe feiner auf die tatsächliche Änderungsgeschwindigkeit eingestellt werden. Die item-spezifischen Werte müssen hierzu im OPC-Server gepuffert werden (item buffering).

#### Beispiel für Item-spezifische Aktualisierungszeiten in einer OPC-Gruppe

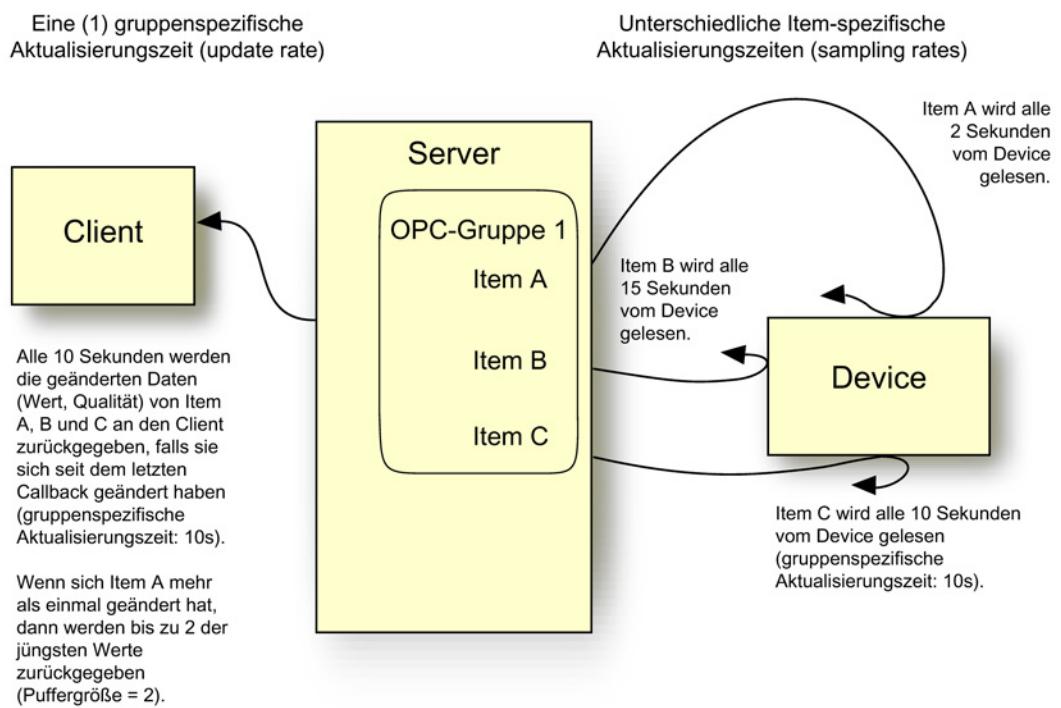


Bild 3-35 Item-spezifische Aktualisierungszeiten in einer OPC-Gruppe

Die OPC-Gruppe 1 im Server hat die drei Items A, B und C. Die Gruppenspezifische Aktualisierungszeit beträgt 10 Sekunden. Die Gruppe und alle Items sind aktiv. Die Items sind folgendermaßen projektiert:

- Item A
  - Item-spezifische Aktualisierungszeit: 2 s
  - Pufferung: Aktiv
  - Puffer-Größe: 2
- Item B
  - Item-spezifische Aktualisierungszeit: 15 s
- Item C
  - Keine projektierte Item-spezifische Aktualisierungszeit

Mit der gruppenspezifischen Aktualisierungszeit von 10 Sekunden werden über die Funktion "Callback" die geänderten Daten (Wert, Qualität) der Items A, B und C mit Zeitstempel an den OPC-Client übertragen.

Wenn von Item A innerhalb der gruppenspezifischen Aktualisierungszeit (10 s) mehr geänderte Daten vorliegen, dann werden bis zu 2 der jüngsten geänderten Daten an den OPC-Client übertragen (Puffergröße = 2).

Die gruppenspezifische Aktualisierungszeit zum OPC-Client wird durch Item-spezifische Werte nicht verändert. Bei Gleichheit der gewünschten Aktualisierungszeit von Item und Gruppe gibt es keine Änderung gegenüber dem Verhalten gemäß der Spezifikation Version 2.xx.

## Pufferung und Übertragung der Items

Folgende Regeln gelten derzeit für die Rückgabe von Items mit einer gruppenspezifischen Aktualisierungszeit:

1. Wenn sich die Qualität seit der letzten Aktualisierung geändert hat, wird das Item an den Client zurückgegeben.
2. Wenn sich der Wert seit der letzten Aktualisierung geändert hat und der gesamte Änderungsbetrag das Deadband (falls vorhanden) übersteigt, wird das Item an den Client zurückgegeben.

Wenn die item-spezifische Aktualisierungszeit kürzer als die gruppenspezifische Aktualisierungszeit ist, muss der Server mithilfe zusätzlicher Logik bestimmen, was bei der nächsten Aktualisierung an den Client zurückgegeben wird.

Wenn die Pufferung nicht aktiviert ist und mindestens eine gelesene Variable den oben stehenden Kriterien 1 und 2 entspricht, dann wird der jüngste Wert an den Client zurückgegeben.

Die jüngste gelesene Variable wird auch dann zurückgegeben, wenn der letzte Wert die Kriterien 1 und 2 nicht erfüllt.

Wenn die Pufferung aktiviert ist, beginnt der Server erst mit der Pufferung von gelesenen Variablen, wenn er eine Variable liest, die Kriterium 1 und 2 erfüllt. Nach begonnener Pufferung von gelesenen Variablen für ein Item fügt der Server dem Puffer eine neue gelesene Variable dann hinzu, wenn die neue gelesene Variable beim Vergleich mit der vorherigen gelesenen Variablen eine abweichende Qualität oder einen abweichenden Wert aufweist.

---

*3.6 Leistungen von OPC Data Access und OPC Alarms & Events bei SIMATIC NET*

Wenn die neue gelesene Variable mit der letzten gelesenen Variablen im Puffer identisch ist, aktualisiert der Server nur den Zeitstempel der letzten gelesenen Variablen im Puffer.

Zusammenfassend ausgedrückt: Die an den Client zurückgegebene Gruppe von gelesenen Variablen ergibt sich als eine Folge von Werten, die sich alle von der vorherigen gelesenen Variablen unterscheiden und einen Zeitstempel aufweisen, aus welchem der letzte Zeitpunkt hervorgeht, an dem für das Item der betreffende Wert bekannt war.

Wenn bei einem bestimmten OnDataChange-Callback für ein Item mehrere Werte/Qualitäten/Zeitstempel zurückzugeben sind, so führt dies je nach Größe der Sammlung zu mehrfachen doppelten ClientHandles, die mit der entsprechenden Dreiergruppe aus Wert, Qualität und Zeitstempel zurückgegeben werden.

Die Funktion des mit OPC Data Access 3.00 eingeführten "item buffering" heißt "SetItemBufferEnable".

Die Meldung der Werte vom OPC-Server an den Client erfolgt über die Funktion OnDataChange().



# 4

## Literaturverzeichnis

### Auffinden der SIMATIC NET-Dokumentation

- **Kataloge**

Die Bestellnummern für die hier relevanten Siemens-Produkte finden Sie in den folgenden Katalogen:

- SIMATIC NET Industrielle Kommunikation / Industrielle Identifikation, Katalog IK PI
- SIMATIC Produkte für Totally Integrated Automation und Micro Automation, Katalog ST 70

Die Kataloge sowie zusätzliche Informationen können Sie bei Ihrer Siemens-Vertretung anfordern.

Die Industry Mall finden Sie unter folgender Adresse im Internet:

Industry Mall

([https://eb.automation.siemens.com/goos/WelcomePage.aspx?regionUrl=/&nodeID=1000000&view=intranet&infoTypeID=\\*&language=de#topAnch](https://eb.automation.siemens.com/goos/WelcomePage.aspx?regionUrl=/&nodeID=1000000&view=intranet&infoTypeID=*&language=de#topAnch))

- **Dokumentation im Internet**

Die SIMATIC NET-Handbücher finden Sie auf den Internet-Seiten des Siemens Automation Customer Support:

Link zum Customer Support (<http://support.automation.siemens.com/WW/view/de>)

Navigieren Sie zur gewünschten Produktgruppe und nehmen Sie folgende Einstellungen vor:

Register "Beitragsliste", Beitragstyp "Handbücher / Betriebsanleitungen"

- **Dokumentation in der STEP 7-Installation**

Handbücher, die in der Online-Dokumentation der STEP 7-Installation auf Ihrem PG/PC vorhanden sind, finden Sie über das Startmenü ("Start" > "SIMATIC" > "Dokumentation").

### Siehe auch

Link zur Dokumentation:

([http://www.automation.siemens.com/simatic/portal/html\\_00/techdoku.htm](http://www.automation.siemens.com/simatic/portal/html_00/techdoku.htm))

Professional ASP.NET Web Services

WROX Press Ltd

ISBN 1-861005-45-8

hier speziell Kapitel 9 "Asynchronous Programming"

SIMATIC NET  
PC-Stationen In Betrieb nehmen - Anleitung und Schnelleinstieg  
Projektierungshandbuch  
Siemens AG  
(SIMATIC NET Manual Collection)  
Im Internet unter folgender Beitrags-ID:  
13542666 (<http://support.automation.siemens.com/WW/view/de/13542666>)

SIMATIC NET  
PROFIBUS Netzhandbuch  
Siemens AG  
(SIMATIC NET Manual Collection)  
Im Internet unter folgender Beitrags-ID:  
35222591 (<http://support.automation.siemens.com/WW/view/de/35222591>)

SIMATIC NET  
Handbuch Twisted Pair- und Fiber Optic Netze  
Siemens AG  
(SIMATIC NET Manual Collection)

# Index

## A

Aktualisierungszeit, 145  
Alarm, 102, 104  
Alarms & Events, 97  
Anlagenkonfiguration  
    DP-Protokoll, 34  
    PROFINET IO, 71  
    S7-Protokoll, 45  
    SEND/RECEIVE-Protokoll, 26  
    SNMP-Protokoll, 68  
Areas, 101  
Asynchrone Zugriffe, 143  
    Asynchrones Lesen und Schreiben, 143  
Attribut, 123  
Authentisierung, 114  
Automation-Schnittstelle, 90

## B

Blockdienste, 29, 141, 142  
Browse, 112

## C

Cache, 140  
Client-Server-Modell, 46, 68, 84  
COM, 88  
COM-inproc-Server, 133  
COM-Objekt, 88  
COM-Objekte, 89  
COM-Schnittstellen, 90  
Custom-Schnittstelle, 90

## D

Data-Access-Clients, 93  
Data-Access-Server, 93  
DCOM, 88  
Discovery, 118  
DPC1, 33  
DPC1-Dienste, 40  
DPC2, 33  
DPC2-Dienste, 42

DP-Master Klasse 1, 33, 37  
DP-Master Klasse 2, 33  
DP-Slave, 43  
DPV1, 33

## E

Endpunkte der OPC UA Server, 118  
Ereignismeldungen, 98  
    Bedingte Ereignisse, 98  
    Einfache Ereignisse, 98  
    Protokollierereignisse, 98  
Ereignisse, 98

## F

Feldebene, 12  
Filter, 125

## G

GetProperties, 112  
GetStatus, 111

## H

HTTP-Protokoll, 107

## I

Industrial Ethernet, 18  
In-Process-Server, 85  
Instances, 118  
Internet, 105  
IO-Controller, 73  
IO-Device, 73  
IO-Supervisor, 73  
IP, 23  
ISO on TCP, 28  
ISO/OSI-Referenzmodell  
    Allgemein, 14  
    Industrial Ethernet, 20  
    PROFIBUS, 17  
    PROFINET, 25  
Isochroner Real Time, 23, 75  
ISO-Transportprotokoll, 28

## K

Kanal  
  sicherer, 120  
Klassenmodell  
  Klasse OPC Item, 96  
  Klasse OPC-Event-Area-Browser, 101  
  Klasse OPC-Event-Subscription, 100  
  Klasse OPC-Group, 95  
  Klasse OPC-Server, 95  
  OPC Alarms & Events, 98  
  OPC Data Access, 94  
  OPC-Event-Server, 99  
Knoten, 123

## L

Leitebene, 12  
Local Server, 85

## M

Management Information Base, 69  
Master-Slave-Prinzip, 17  
Meldungsempfang, 101  
MonitoredItem, 123  
Monitoring-Modus, 125

## N

NodeID, 122, 123  
Notification, 123  
NotificationMessage, 125

## O

OPC Data Access, 93  
OPC Data Control, 82  
OPC-Cache, 139  
OPC-Client, 86  
OPC-Event-Area-Browser, 98  
OPC-Event-Server, 98  
OPC-Event-Subscription, 98  
OPC-Schnittstelle, 84  
OPC-Scout, 82  
OPC-Server, 85  
OPC-XML, 106  
Optimierung, 141

## P

Percent Deadband, 146  
Performance, 133  
Polling, 35  
PROFIBUS, 15  
PROFINET, 21  
PROFINET IO, 70  
PROFINET-Geräte, 70  
ProgID, 85  
Protokolle  
  DP-Protokoll, 32  
  Offenes SEND/RECEIVE-Protokoll,  
  RPC-Protokoll, 74  
  S7-Protokoll, 44  
  SNMP-Protokoll, 67  
  Übersicht Industrial Ethernet, 13  
  Übersicht PROFIBUS, 13  
Provider-Consumer-Modell, 73  
Publish Request, 123, 126  
Publishing-Intervall, 126

## Q

Queue-Attribute, 125

## R

Read, 111  
Real Time, 23  
Remote Server, 85  
RFC1006, 28

## S

S7-Bausteindienste, 49  
S7-Blockdienste, 48  
S7-Ereignisdienst, 50  
S7-Informationsdienste, 47  
S7-Sicherheitsdienst, 51  
S7-Variablen-dienste, 48  
S7-Verbindungen  
  hochverfügbare, 52  
Sampling-Intervall, 125  
Schnittstellenspezifikationen, 97  
Session, 120  
Session-ID, 120  
SIMATIC NET, 11  
SNMP-Agent, 68  
SNMP-Manager, 68  
SOAP, 106

Subscription, 112, 123  
Switched-Ethernet, 20  
Symbolik, 82  
Synchrone Zugriffe, 142  
  Synchrones Lesen und Schreiben, 142

## T

TCP, 23  
TCP/IP native Protokoll, 28  
Token-Passing-Verfahren, 16  
Totally Integrated Automation, 11, 21  
Types, 118

## U

UDP, 23  
UpdateRate, 145

## V

Variabldienste, 29

## W

Web-Dienste, 108  
Write, 111

## X

XML, 106  
XML-Schnittstelle, 108  
XML-Web-Dienst, 111

## Z

Zellebene, 12  
Zertifikat, 120  
Zertifkate, 114

