

SIEMENS

Ingenuity for life

Konsistente Datensynchronisation mit S7- Kommunikation zwischen mehreren Substationen und einer Kopfstation

S7-1500 / S7-300 / S7-1200 / STEP 7 (TIA Portal)

<https://support.industry.siemens.com/cs/ww/de/view/40556214>

Siemens
Industry
Online
Support



Rechtliche Hinweise

Nutzung der Anwendungsbeispiele

In den Anwendungsbeispielen wird die Lösung von Automatisierungsaufgaben im Zusammenspiel mehrerer Komponenten in Form von Text, Grafiken und/oder Software-Bausteinen beispielhaft dargestellt. Die Anwendungsbeispiele sind ein kostenloser Service der Siemens AG und/oder einer Tochtergesellschaft der Siemens AG („Siemens“). Sie sind unverbindlich und erheben keinen Anspruch auf Vollständigkeit und Funktionsfähigkeit hinsichtlich Konfiguration und Ausstattung. Die Anwendungsbeispiele stellen keine kundenspezifischen Lösungen dar, sondern bieten lediglich Hilfestellung bei typischen Aufgabenstellungen. Sie sind selbst für den sachgemäßen und sicheren Betrieb der Produkte innerhalb der geltenden Vorschriften verantwortlich und müssen dazu die Funktion des jeweiligen Anwendungsbeispiels überprüfen und auf Ihre Anlage individuell anpassen.

Sie erhalten von Siemens das nicht ausschließliche, nicht unterlizenzierbare und nicht übertragbare Recht, die Anwendungsbeispiele durch fachlich geschultes Personal zu nutzen. Jede Änderung an den Anwendungsbeispielen erfolgt auf Ihre Verantwortung. Die Weitergabe an Dritte oder Vervielfältigung der Anwendungsbeispiele oder von Auszügen daraus ist nur in Kombination mit Ihren eigenen Produkten gestattet. Die Anwendungsbeispiele unterliegen nicht zwingend den üblichen Tests und Qualitätsprüfungen eines kostenpflichtigen Produkts, können Funktions- und Leistungsmängel enthalten und mit Fehlern behaftet sein. Sie sind verpflichtet, die Nutzung so zu gestalten, dass eventuelle Fehlfunktionen nicht zu Sachschäden oder der Verletzung von Personen führen.

Haftungsausschluss

Siemens schließt seine Haftung, gleich aus welchem Rechtsgrund, insbesondere für die Verwendbarkeit, Verfügbarkeit, Vollständigkeit und Mangelfreiheit der Anwendungsbeispiele, sowie dazugehöriger Hinweise, Projektierungs- und Leistungsdaten und dadurch verursachte Schäden aus. Dies gilt nicht, soweit Siemens zwingend haftet, z.B. nach dem Produkthaftungsgesetz, in Fällen des Vorsatzes, der groben Fahrlässigkeit, wegen der schuldhafte Verletzung des Lebens, des Körpers oder der Gesundheit, bei Nichteinhaltung einer übernommenen Garantie, wegen des arglistigen Verschweigens eines Mangels oder wegen der schuldhafte Verletzung wesentlicher Vertragspflichten. Der Schadensersatzanspruch für die Verletzung wesentlicher Vertragspflichten ist jedoch auf den vertragstypischen, vorhersehbaren Schaden begrenzt, soweit nicht Vorsatz oder grobe Fahrlässigkeit vorliegen oder wegen der Verletzung des Lebens, des Körpers oder der Gesundheit gehaftet wird. Eine Änderung der Beweislast zu Ihrem Nachteil ist mit den vorstehenden Regelungen nicht verbunden. Von in diesem Zusammenhang bestehenden oder entstehenden Ansprüchen Dritter stellen Sie Siemens frei, soweit Siemens nicht gesetzlich zwingend haftet.

Durch Nutzung der Anwendungsbeispiele erkennen Sie an, dass Siemens über die beschriebene Haftungsregelung hinaus nicht für etwaige Schäden haftbar gemacht werden kann.

Weitere Hinweise

Siemens behält sich das Recht vor, Änderungen an den Anwendungsbeispielen jederzeit ohne Ankündigung durchzuführen. Bei Abweichungen zwischen den Vorschlägen in den Anwendungsbeispielen und anderen Siemens Publikationen, wie z. B. Katalogen, hat der Inhalt der anderen Dokumentation Vorrang.

Ergänzend gelten die Siemens Nutzungsbedingungen (<https://support.industry.siemens.com>).

Securityhinweise

Siemens bietet Produkte und Lösungen mit Industrial Security-Funktionen an, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen.

Um Anlagen, Systeme, Maschinen und Netzwerke gegen Cyber-Bedrohungen zu sichern, ist es erforderlich, ein ganzheitliches Industrial Security-Konzept zu implementieren (und kontinuierlich aufrechtzuerhalten), das dem aktuellen Stand der Technik entspricht. Die Produkte und Lösungen von Siemens formen nur einen Bestandteil eines solchen Konzepts.

Der Kunde ist dafür verantwortlich, unbefugten Zugriff auf seine Anlagen, Systeme, Maschinen und Netzwerke zu verhindern. Systeme, Maschinen und Komponenten sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn und soweit dies notwendig ist und entsprechende Schutzmaßnahmen (z.B. Nutzung von Firewalls und Netzwerksegmentierung) ergriffen wurden.

Zusätzlich sollten die Empfehlungen von Siemens zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Industrial Security finden Sie unter: <https://www.siemens.com/industrialsecurity>.

Die Produkte und Lösungen von Siemens werden ständig weiterentwickelt, um sie noch sicherer zu machen. Siemens empfiehlt ausdrücklich, Aktualisierungen durchzuführen, sobald die entsprechenden Updates zur Verfügung stehen und immer nur die aktuellen Produktversionen zu verwenden. Die Verwendung veralteter oder nicht mehr unterstützter Versionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Produkt-Updates informiert zu sein, abonnieren Sie den Siemens Industrial Security RSS Feed unter: <https://www.siemens.com/industrialsecurity>.

Inhaltsverzeichnis

Rechtliche Hinweise	2
1 Aufgabe	4
2 Lösung	6
2.1 Übersicht.....	6
2.2 Verwendete Hard- und Software-Komponenten	8
2.3 Leistungseckdaten.....	9
2.3.1 Synchronisationsgeschwindigkeit.....	9
2.3.2 Anzahl der Substationen.....	11
2.4 Alternative	11
3 Funktionsmechanismen der Applikation	12
3.1 Übersicht.....	12
3.2 Beschreibung des Programms der Kopfstation.....	15
3.2.1 Programmübersicht	15
3.2.2 FB Sync_Client2Serv (FB111): schreibende Synchronisation.....	17
3.2.3 FB Sync_Serv2Client (FB222): lesende Synchronisation	20
3.3 Beschreibung des Programms einer Substation.....	23
3.3.1 Programmübersicht	23
3.3.2 FB Sync_Server (FB111).....	24
3.3.3 Datenkonsistenz im Server.....	27
3.4 Die PLC-Datentypen.....	27
4 Inbetriebnahme der Applikation	31
4.1 Aufbau der Hardware.....	31
4.2 Konfiguration der Hardware	32
4.3 Öffnen und Laden des STEP 7-Projekts.....	34
5 Bedienung der Applikation	35
5.1 Beobachtung der Applikation	35
5.2 Szenario A: schreibende Datensynchronisation	36
5.3 Szenario B: lesende Synchronisation.....	37
5.4 Szenario C: Fehlermeldung bei gezogenem Kabel.....	39
6 Änderungen am Projekt	40
6.1 Ändern der zu synchronisierenden Datenbereiche	40
6.2 Hinzufügen eines weiteren Servers.....	44
6.3 Sicherstellen der Datenkonsistenz im Server	46
6.4 Erweitern der Funktion des Koordinations-Bytes	47
7 Anhang	49
7.1 Service und Support	49
7.2 Links und Literatur	50
7.3 Änderungsdokumentation	50

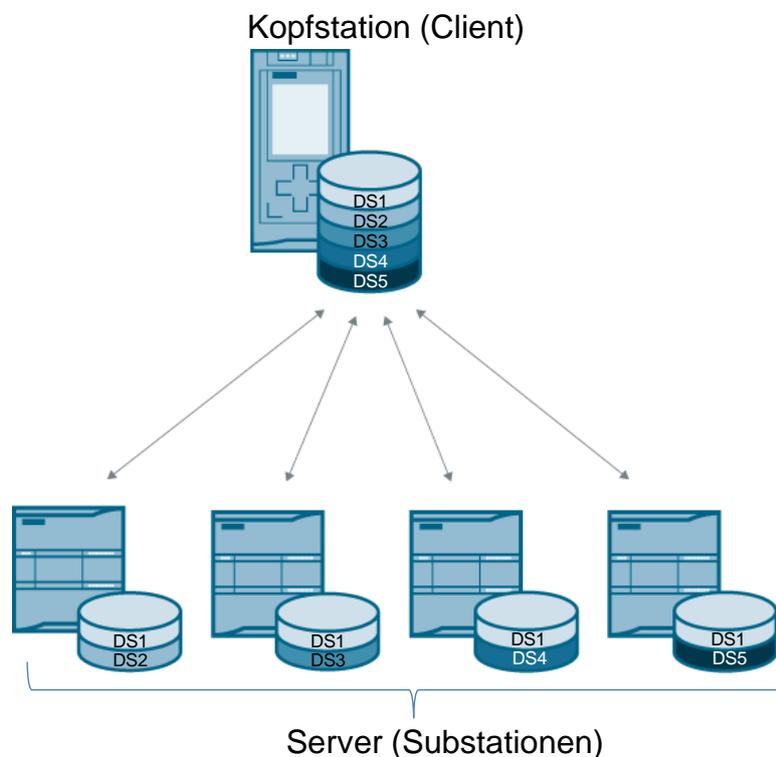
1 Aufgabe

Einleitung

Für die zentrale Haltung von Daten in einem Automatisierungssystem mit mehreren Substationen ist es oft eine Standardaufgabe die Datenbereiche zwischen einer zentralen Kopfstation und vielen dezentralen Substationen zyklisch zu synchronisieren. Eine Anwendung einer Kopfstation als Datenkonzentrator ist zum Beispiel das Beschränken von Zugriffen von Seiten eines SCADA-Systems auf nur eine PLC-Station.

Folgendes Bild gibt einen Überblick über die Automatisierungsaufgabe.

Abbildung 1-1



Beschreibung der Automatisierungsaufgabe

Zur konsistenten Synchronisation zwischen einem zentralen Client (Kopfstation) und mehreren Servern (Substationen) werden Bausteine bereitgestellt, die mit Hilfe der S7-Kommunikation folgende Aufgaben übernehmen:

- Versorgung der Server (Substationen) mit vom Anwender definierbaren Datensätzen (schreibende Synchronisation).
- Aktualisieren von definierten Datensätzen aus den Servern (Substationen) mit dem Client (Kopfstation) (lesende Synchronisation).
- Verwenden der schreibenden Synchronisation zur Synchronisierung der Systemzeit der Substationen mit der Systemzeit der Kopfstation.

Anforderungen an die Automatisierungsaufgabe

Folgende Anforderungen werden an die Applikation gestellt:

- Die Synchronisation muss konsistent erfolgen.
- Die schreibende Synchronisation erfolgt Event getriggert aus dem Anwenderprogramm.
- Die lesende Synchronisation aus den Servern (Substationen) zum Client (Kopfstation) erfolgt zyklisch.
- Die Kommunikation erfolgt über Ethernet mit Hilfe der S7-Kommunikation.
- Die Änderung der Mengengerüste des Programms (Anzahl der Server, Größe und Lage der Sende- bzw. Empfangsdatenmengen) soll einfach möglich sein.
- Als Clients (Kopfstationen) sollen S7-CPU's der S7-300/400 und S7-1500 Familie einsetzbar sein.
- In der Applikation werden als Server (Substationen) S7-1200 CPU's verwendet.
- Die Programmierung erfolgt vorrangig in SCL.

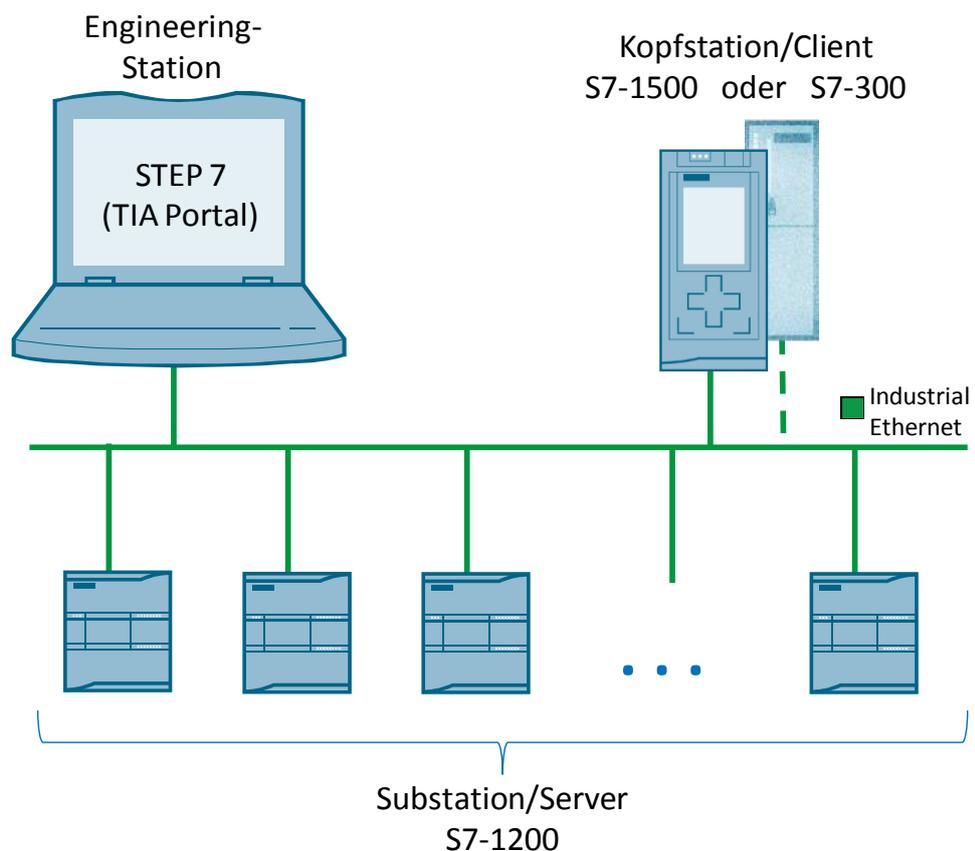
2 Lösung

2.1 Übersicht

Schema

Die folgende Abbildung zeigt schematisch die wichtigsten Komponenten der Lösung:

Abbildung 2-1



Aufbau

Der Aufbau bildet ein Automatisierungssystem mit einer Kopf- und mehreren Substationen ab, die in diesem Applikationsbeispiel als Client (Kopfstation) und Server (Substation) beschrieben werden.

Das Beispiel besteht aus

- einer Kopfstation (S7-1500 oder alternativ eine S7-300)
- vier Substationen (S7-1200).

Funktionen

Das Applikationsbeispiel realisiert die folgenden Funktionen:

Tabelle 2-1

Funktion	Beschreibung	Mengengerüst
schreibende Synchronisation (Kopfstation → Substation)	Die Applikation synchronisiert auf Anforderung parametrierbare Speicherbereiche der Kopfstation (Client) mit den Substationen (Servern).	Maximal können Datenbereiche von 160 Byte synchronisiert werden.
Zyklische, lesende Synchronisation (Substation → Kopfstation)	Die Kopfstation (Client) liest zyklisch, parametrierbare Speicherbereiche aus den Substationen (Servern) aus (wenn neue Daten anstehen).	
Uhrzeitsynchronisation	Auf Anforderung überträgt die Kopfstation (Client) die lokale Systemuhrzeit an die Substationen (Server), die diese dann übernehmen.	

Zur Synchronisation wird die S7-Kommunikation über Industrial Ethernet verwendet. Die S7-Kommunikation über PUT- und GET-Bausteine basiert auf einem Server-Client-Konzept. Der Server hält passiv Daten bereit, die der Client dann über PUT- und GET-Bausteine bearbeiten kann.

Vorteile

Die hier realisierte Lösung bietet die folgenden Vorteile:

- Parametrierbare Größe und Ablageort der zu synchronisierenden Datenbereiche.
- Skalierbares Projekt zum Einsatz von weniger oder mehr als vier Substationen (nur abhängig von der maximal in der Kopfstation verfügbaren S7-Kommunikationsverbindungen).
- Modularer Aufbau des Programms. Die beiden Synchronisationsrichtungen können unabhängig voneinander in Ihren Anwendungen verwendet werden.

Abgrenzung

Diese Applikation enthält keine:

- Beschreibung des S7-Protokolls.
- Einführung in die Programmierung mit STEP 7 (siehe [131](#)).

Grundlegende Kenntnisse über diese Themen werden vorausgesetzt. Das Kapitel [7.2](#) gibt Hinweise zu weiterführender Literatur.

Gültigkeit

Diese Applikation ist gültig für

- STEP 7 ab V15.1
- SIMATIC S7-300
- SIMATIC S7-1200
- SIMATIC S7-1500

2.2 Verwendete Hard- und Software-Komponenten

Die Applikation wurde mit den nachfolgenden Komponenten erstellt:

Hardware-Komponenten

Tabelle 2-2

Komponente	Anz.	Bestellnummer	Hinweis
CPU315-2 PN/DP	1	6ES7315-2EH14-0AB0	Alternativ kann auch eine andere CPU mit Ethernet-Schnittstelle aus dem SIMATIC S7-300 Spektrum verwendet werden.
CPU1516-3PN/DP	1	6ES7516-3AN01-0AB0	Alternativ kann auch eine andere CPU aus dem SIMATIC S7-1500 Spektrum verwendet werden.
CPU 1212C DC/DC/DC	4	6ES7212-1AE31-0XB0	Alternativ kann auch eine andere CPU mit Ethernet-Schnittstelle aus dem SIMATIC S7-1200 Spektrum verwendet werden.
SCALANCE X208	1	6GK5208-0BA10-2AA3	Es können auch andere Switches zur Vernetzung der CPUs verwendet werden.
PM 190W 120/230VAC	1	6EP1333-4BA00	Es können auch andere Power Supplies zur Anbindung der CPUs verwendet werden.

Hinweis Wenn Sie andere Hardware als im Beispielprojekt verwenden, dann müssen Sie entsprechende Änderungen in der Hardwarekonfiguration vornehmen!

Software-Komponenten

Tabelle 2-3

Komponente	Anz.	Bestellnummer	Hinweis
STEP 7 Professional V15.1	1	6ES7822-1..05-..	mit Update 2

Beispieldateien und Projekte

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

Tabelle 2-4

Komponente	Hinweis
40556214_S7- Comm_Sync_TiaV15.1_PROJ_V2.1.zip	Diese gepackte Datei enthält das STEP 7 Projekt.
40556214_S7- Comm_Sync_DOC_V2.1_de.pdf	Dieses Dokument.

2.3 Leistungseckdaten

2.3.1 Synchronisationsgeschwindigkeit

Parameter

Für einen Überblick über die Leistungsfähigkeit des Applikationsbeispiels wurden Leistungsmessungen durchgeführt.

Die gemessenen Szenarien unterscheiden sich in den folgenden Punkten:

- Art der Kopfstation
- Anzahl der Substationen
- Programmlast

Tabelle 2-5 zeigt die Wertebereiche der performance-relevanten Parameter.

Tabelle 2-5

Parameter	Einstellungen
Art der Kopfstation	CPU1516-3 PN/DP; CPU315-2 PN/DP
Anzahl der Substationen	1; 4; 8; 14
Programmlast in Kopf- und Substationen	6ms; 50ms
Größe des zu synchronisierenden Bereichs	160 Byte

Messung

Gemessen wurde die schreibende Synchronisation von einer Kopfstation (Client) zu mehreren Substationen (Servern) bei paralleler lesender Synchronisation. Startzeitpunkt der Messung ist der eingehende Trigger für die Synchronisation. Die Messung wird beendet, wenn alle Substationen einmal synchronisiert wurden.

Messungen

Jeder Messpunkt in den folgenden Diagrammen wird 100-mal gemessen und anschließend gemittelt.

Abbildung 2-2: 160 Byte, 6ms Programmlast

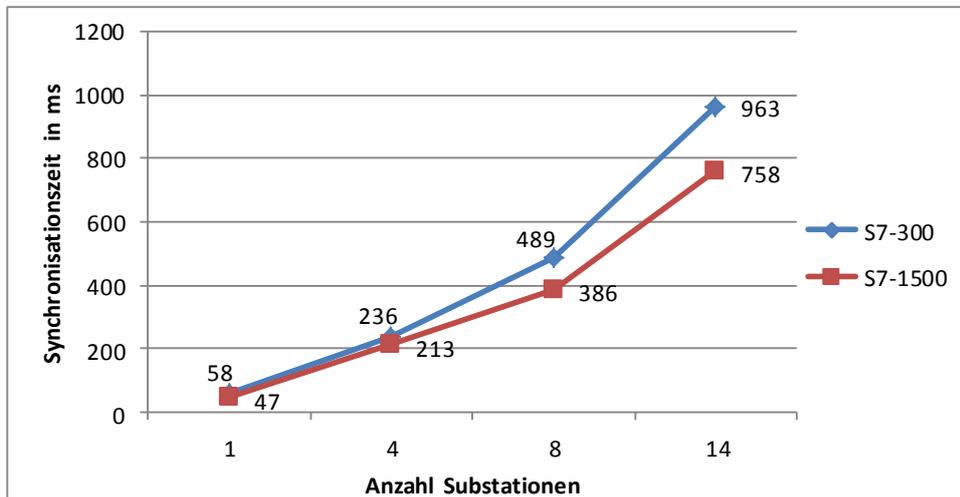
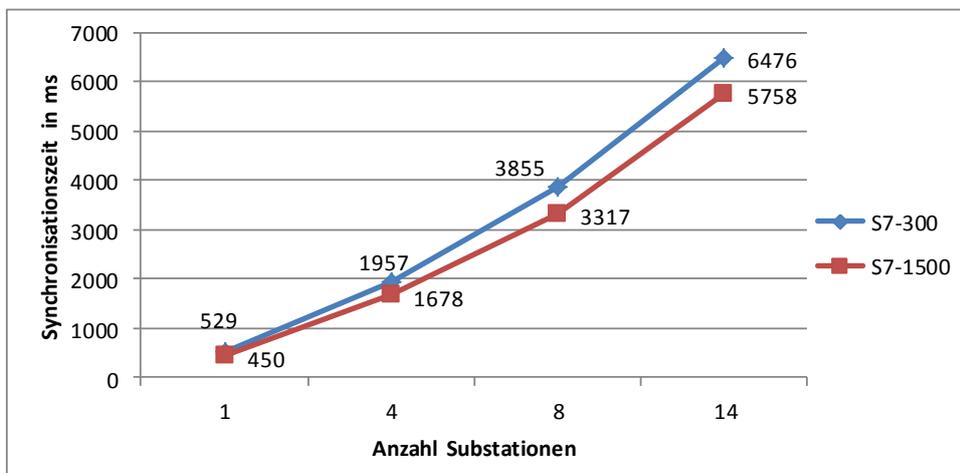


Abbildung 2-3: 160 Byte, 50 ms Programmlast



Beobachtungen

Folgende Beobachtungen können aus der Messung gezogen werden:

- Die S7-1500 synchronisiert die Server schneller als die S7-300 (bei einer Programmlast von 50ms und 14 zu synchronisierende Substationen macht der Unterschied etwa eine Sekunde aus).
- Mit der Anzahl der Substationen erhöht sich die Zykluszeit näherungsweise linear.
- Bei geringer Programmlast wird die Synchronisationszeit selbst bei 14 Substationen nicht größer als eine Sekunde.

2.3.2 Anzahl der Substationen

Die maximal mögliche Anzahl an Substationen hängt nur vom Typ der verwendeten Kopfsteuerung ab und ist den technischen Daten zu entnehmen.

Tabelle 2-6

Kopfstation	Anzahl Verbindungen
CPU 315-2PN/DP (6ES7 315-2EH14-0AB0)	maximal 14 Verbindungen für S7-Kommunikation Siehe \8\
CPU 1516-3 PN/DP(6ES7 516-3AN00-0AB0)	maximal 256 Verbindungen, davon 10 reserviert für ES/HMI/Web Siehe \9\

2.4 Alternative

Offene Kommunikation

Die Synchronisation von Datenbereichen kann auch an Stelle von S7-Kommunikation über die offene Kommunikation erfolgen.

Unterscheidung

Bei der Verwendung der S7-Kommunikation wird auf projektierte Verbindungen zugegriffen, die statisch Kommunikationsreserven der Kommunikationspartner belegen.

Bei Verwendung der offenen Kommunikation (TCP/IP, ISOonTCP), können die Verbindungen programmiert werden und im Anwenderprogramm dynamisch auf- und abgebaut.

Wenn Sie mehr Teilnehmer synchronisieren wollen, als eine CPU parallele Verbindungen aufbauen kann, dann müssen Sie Verbindungen über die offene Kommunikation auf- und abbauen ("multiplexen"). Die offene Kommunikation bietet, im Gegensatz zur S7-Kommunikation, diese Möglichkeit.

Applikationsbeispiel

Das Anwendungsbeispiel "Datensynchronisation über offene Kommunikation zwischen mehreren Feldsteuerungen (S7-1200) und einer Zentralstation (S7-1200/ S7-300/ WinAC)" realisiert eine Datensynchronisation mit offener Kommunikation (siehe [\10\](#)).

3 Funktionsmechanismen der Applikation

Dieses Kapitel beschreibt Ihnen die Funktion der Applikation.

In Kapitel 3.1 erfahren Sie, was unter schreibender und unter lesender Synchronisation zu verstehen ist.

In Kapitel 3.2 und Kapitel 3.3 erfahren Sie, wie die Anwenderprogramme von Client und Server aufgebaut sind.

3.1 Übersicht

Die Applikation synchronisiert auf eine Anforderung hin Speicherbereiche des Clients (entweder eine S7-300 oder eine S7-1500) mit vier Servern.

Zyklisch werden Daten aus vier Servern in Speicherbereiche des Clients geschrieben.

Die folgenden Abschnitte geben einen Überblick über den Vorgang der schreibenden und lesenden Synchronisation.

Ablauf schreibende Synchronisation (Kopfstation → Substation)

Die folgende Abbildung zeigt schematisch den Ablauf der schreibenden Synchronisation von Nutzdaten eines Clients mit nur einem Server.

Abbildung 3-1

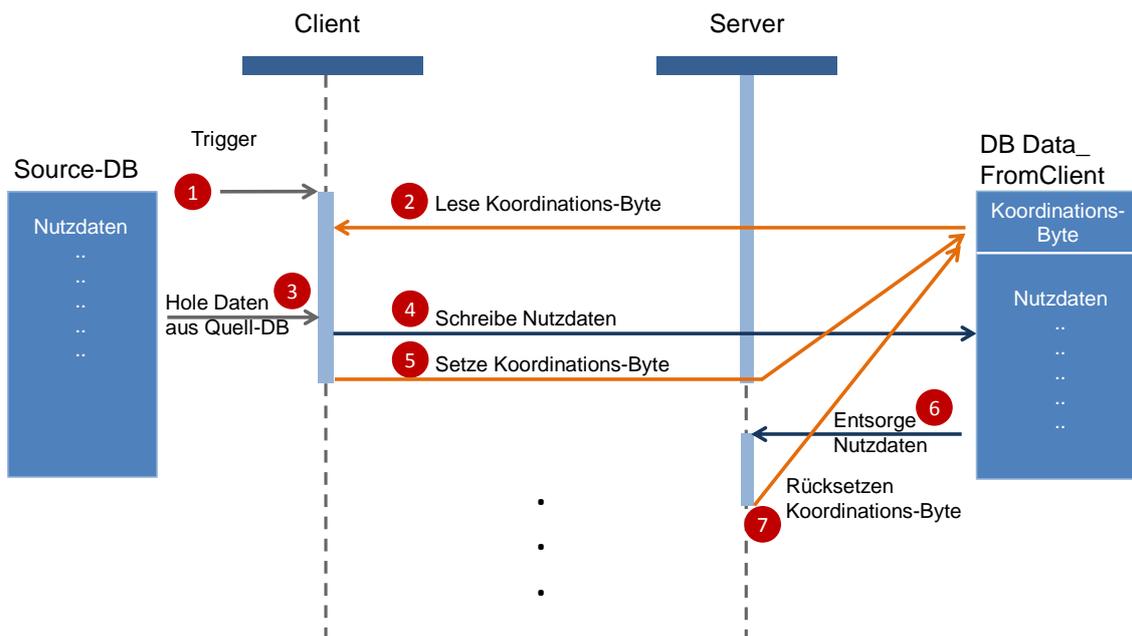


Tabelle 3-1

Nr.	Vorgang
1	Schreibende Synchronisation wird vom Anwenderprogramm aus getriggert.
2	Lesen des Synchronisations-Bytes aus dem Server.
3	Wenn der Server anzeigt, dass neue Daten vorhanden sind, dann werden in Schritt 3 die Daten aus dem vom Nutzer angegebenen Quell-Datenbaustein geholt.
4	Die Nutzdaten werden in den angegebenen Zieldatenbereich geschrieben.

3 Funktionsmechanismen der Applikation

3.1 Übersicht

Nr.	Vorgang
5	Das Koordinations-Byte gesetzt um das Anliegen neuer Daten anzuzeigen.
6	Die angekommenen Nutzdaten werden vom Server weiterverarbeitet.
7	Das Koordinations-Byte wird vom Server dann zurückgesetzt, wenn wieder ein neuer Datensatz vorhanden ist. Der Client beginnt dann wieder mit Schritt 1.

Ablauf lesende Synchronisation (Substation → Kopfstation)

Die folgende Abbildung zeigt schematisch den Ablauf der lesenden Synchronisation mit nur einem Server.

Abbildung 3-2

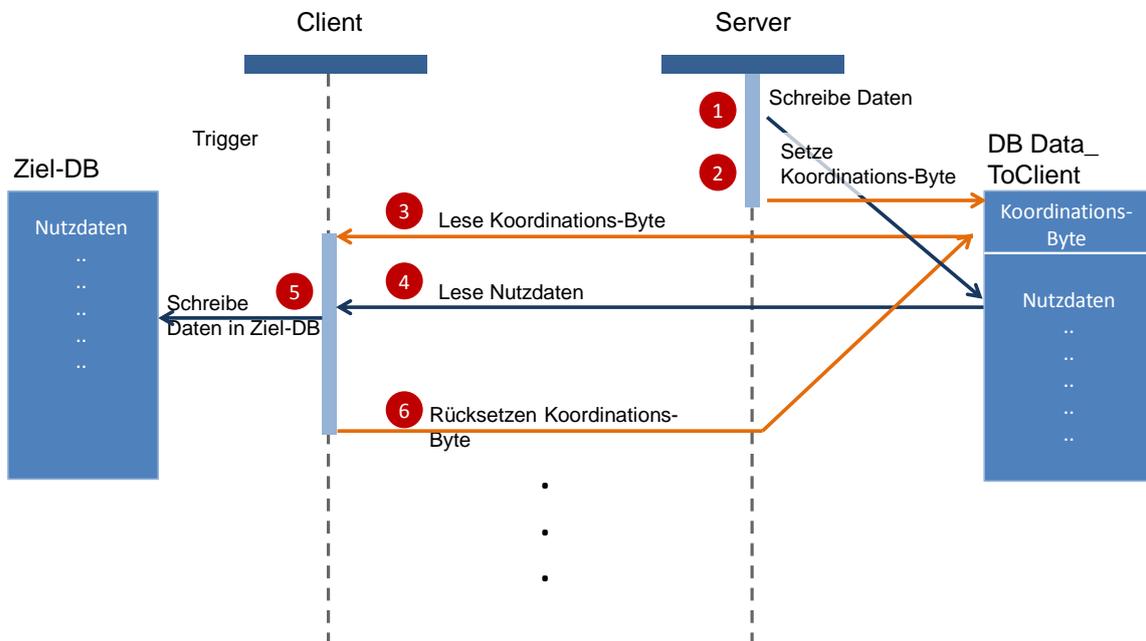


Tabelle 3-2

Nr.	Vorgang
1	Der Server schreibt Daten für den Client in seinen Nutzdatenbereich im DB DataToClient.
2	Der Server setzt das Koordinations-Byte, um dem Client mitzuteilen, dass neue Daten bereit liegen.
3	Der Client liest das Koordinations-Byte und wertet es aus.
4	Der Client liest die Daten aus dem Server und schreibt sie in den Empfangspuffer.
5	Die Daten werden vom Empfangspuffer in den Ziel-Datenbaustein gelegt.
6	Der Server setzt das Koordinations-Byte zurück.

CPUs im Projekt

Im STEP 7 (TIA Portal) Projekt sind folgende S7-CPU's projiziert:

- CPU 1516-3 PN/DP (Client)
- CPU 315-2 PN/DP (Client)
- vier CPU 1212 DC/DC/DC (Server)

Beide Clients besitzen dieselbe Funktionalität und können wahlweise als Kopfstation eingesetzt werden.

Funktionen des Clients

Der zentrale Client (entweder eine S7-1500 oder eine S7-300-Station) realisiert als aktiver Part die Synchronisation von Datenbereichen mit den Servern. Die Synchronisation teilt sich in schreibende (siehe [Abbildung 3-1](#)) und lesende (siehe [Abbildung 3-2](#)) Synchronisation auf.

Für beide Synchronisationsarten kann der Anwender verschiedene Datenbereiche individuell für jeden Server angeben.

Die schreibende Synchronisation wird zudem genutzt, um eine Uhrzeitsynchronisation mit den Servern durchzuführen.

Funktionen des Servers

Der/die Server (eine oder mehrere S7-1200-Stationen) reagiert als passiver Part auf die Synchronisationsaufträge des Clients.

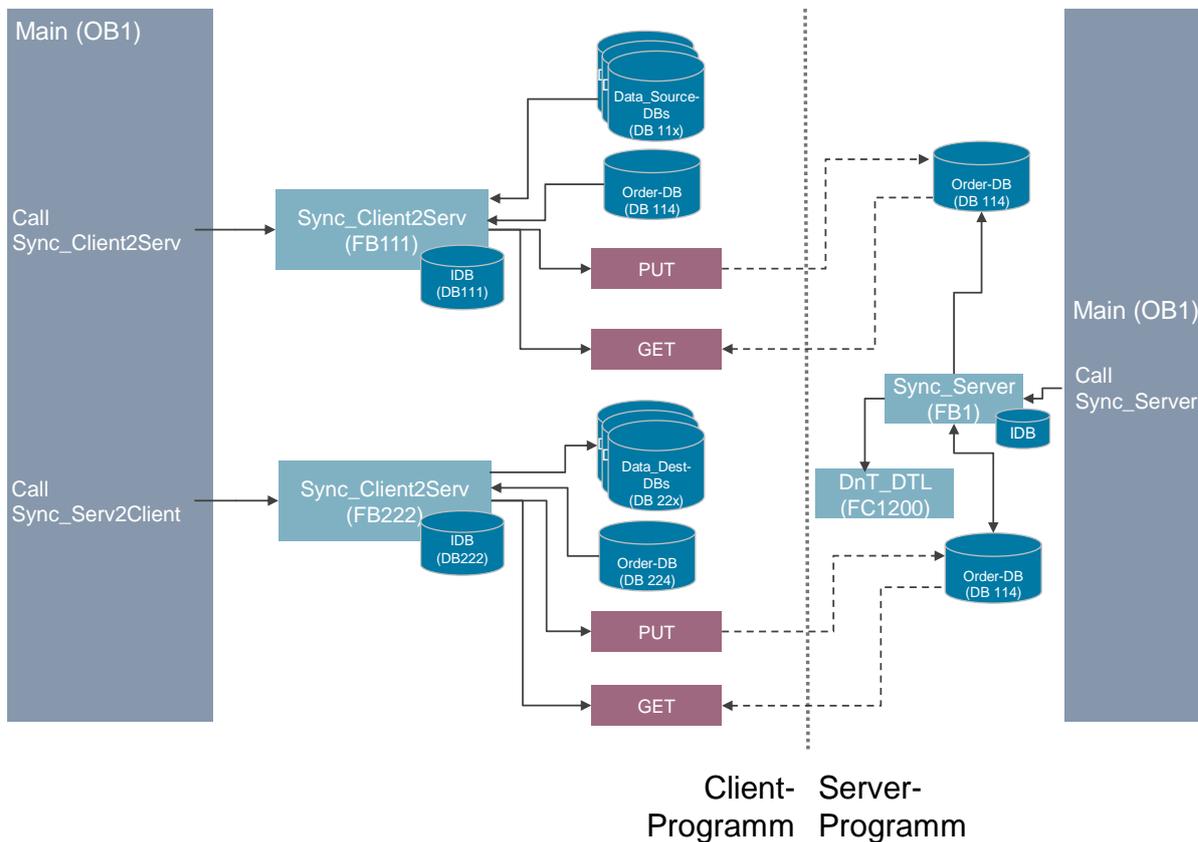
Damit das Serverprogramm auf die Aktionen des Clients reagieren kann und um Datenkonsistenz sicherzustellen, muss bekannt sein, welche Datenbereiche synchronisiert werden sollen.

3.2 Beschreibung des Programms der Kopfstation

3.2.1 Programmübersicht

Überblick

Abbildung 3-3



Copyright © Siemens AG 2019 All rights reserved

Funktionen

Die folgenden Funktionen werden im Client realisiert:

- auf einen Request: Einmalige schreibende Synchronisation mit den Datenbereichen aller Server (siehe Kap [3.2.2](#)).
- Zyklisch lesende Synchronisation von Datenbereichen (siehe Kap [3.2.3](#)).
- auf einen Request: Uhrzeitsynchronisation des Clients mit den Servern.

Die lesende Synchronisation wird zyklisch durchgeführt, da im Anwenderprogramm des Clients (Kopfstation) nicht bekannt ist, wann auf den Servern (Substationen) neue Daten anliegen.

Es werden die Systembausteine PUT und GET verwendet, um Daten zu lesen und zu schreiben.

Bausteine und Anweisungen des Clients

Tabelle 3-3

Element	symbolischer Name	Beschreibung
OB1	Main	Beinhaltet das Hauptprogramm. <ul style="list-style-type: none"> • Ruft den FB Sync_Client2Serv (FB111) und den FB Sync_Serv2Client (FB222) auf. • Realisiert durch die Aufrufe eine Synchronisation von Datenbereichen des Clients mit einem/mehreren Servern.
FB111	Sync_Client2Serv	Synchronisiert auf Anforderung Bereiche des Clients mit einer Größe bis zu 160 Byte entsprechend des INPUTs "orders" mit einem/mehreren Servern. Möglichkeit der Zeitsynchronisation mit allen Servern.
FB222	Sync_Serv2Client	Der INPUT "orders" definiert zu synchronisierende Datenbereiche. Wenn neue Daten in den Servern anliegen, werden diese Datenblöcke mit einer Größe von bis zu 160 Byte mit dem im Client angegebenen Bereich synchronisiert.
DB1	Param_Main_Call	Datenbaustein zur Versorgung der Parameter der Funktionsaufrufe im OB Main (OB1)
DB111	Sync_Client2_Serv_DB	Instanz-DB des FB Sync_Client2Serv (FB111)
DB114	Client2Serv_orders	Definition der zu synchronisierenden Datenbereiche für die Client zu Server-Synchronisation.
DB115	Client2Serv_Buffer	Puffer-Bereich der Daten.
DB11x	src_blk_x	Ein Datenbaustein mit zu synchronisierenden Daten für einen/mehrere Server.
DB222	Sync_Serv2Client_DB	Instanz-DB des FB Sync_Serv2Client (FB222)
DB224	Serv2Client_orders	Definition der zu synchronisierenden Datenbereiche für Server zu Client-Synchronisation.
DB225	Serv2Client_Buffer	Puffer-Bereich der Daten.
DB2xx	dest_blk_x	Ein Datenbaustein mit synchronisierten Daten aus einem Server.
Anweisung	PUT	Schreibt Daten mit der S7-Kommunikation in eine remote CPU.
Anweisung	GET	Liest Daten mit der S7-Kommunikation aus einer remoten CPU.

Die Datenbausteine DB Client2Serv_orders und DB Serv2Client_orders bestehen aus PLC-Datentypen, die in Kapitel [3.4](#) näher beschrieben sind.

Datenkonsistenz

Bei richtiger Implementation der Funktionsbausteine des Anwenderprogramms ist die Konsistenz der zu synchronisierenden Datenbereiche sichergestellt.

Beachten Sie dazu die Anweisungen und Hinweise im Kapitel [6.3](#).

3.2.2 FB Sync_Client2Serv (FB111): schreibende Synchronisation

Überblick

Der FB Sync_Client2Serv (FB111) realisiert die schreibende Synchronisation der angegebenen Datenbereiche auf den S7-Servern (Substationen).

Auf Anforderung führt der FB das

- Lesen und Auswerten des Koordinations-Byte des Ziel-Datenbereiches aus.
- Schreiben neuer Daten in den Server aus.

Das Beispielprogramm synchronisiert hier die maximal möglichen 160 Byte Nutzdaten.

Um Datenmengen kleiner als 160 Byte zu synchronisieren, folgen Sie den Anweisungen und Hinweisen in Kapitel [6.1](#).

Parameter des FB Sync_Client2Serv (FB111)

Die [Abbildung 3-4](#) zeigt die Parameter des FB Sync_Client2Serv (FB111).

Die [Tabelle 3-4](#) beschreibt die einzelnen Parameter.

Abbildung 3-4

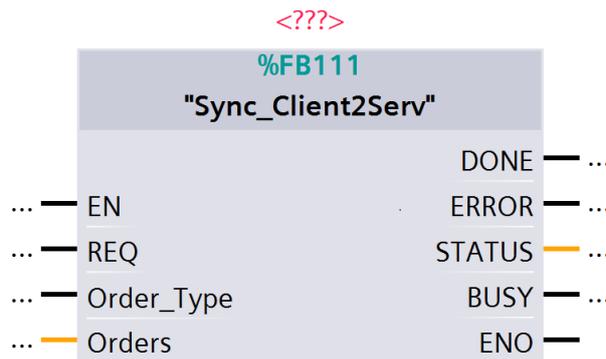


Tabelle 3-4

Parameter	Typ	Anmerkung
REQ	IN: BOOL	Request-Eingang: Eine positive Flanke stößt den Synchronisationsvorgang an.
Order_type	IN: BOOL	Spezifiziert den Synchronisationsauftrag. Order_type=TRUE: Systemzeit wird versendet. Order_type=FALSE: bis zu 160 Byte Nutzdaten werden an den durch "Orders" spezifizierten Datenbereich versendet.
Orders	INOUT: Client2Serv (siehe Kapitel 3.4)	Spezifiziert die Synchronisationsaufträge mit den Servern. Übergibt die Anzahl der zu synchronisierenden Server, die ID der Verbindung, DB-Nummer und Offset von Quelle und Ziel sowie die Länge der Nutzdaten
DONE	OUT: BOOL	Wenn ein Synchronisationsdurchgang erfolgreich durchlaufen ist, wird für einen Zyklus DONE=TRUE
ERROR	OUT: BOOL	Wenn ein Synchronisationsdurchgang mit Fehlern durchlaufen wurde, dann ist für einen Zyklus ERROR=TRUE.

3 Funktionsmechanismen der Applikation

3.2 Beschreibung des Programms der Kopfstation

Parameter	Typ	Anmerkung
STATUS	OUT: DWORD	Gibt weitere Informationen zur Art des Fehlers. STATUS bleibt einen Zyklus anstehen.
BUSY	OUT: BOOL	Zeigt an, dass der Baustein einen Synchronisationsvorgang abarbeitet

ACHTUNG Bei der schreibenden Synchronisation muss die Angabe der Länge des zu synchronisierenden Datenbereiches bei allen Servern identisch sein!

Der Ausgangsparameter STATUS

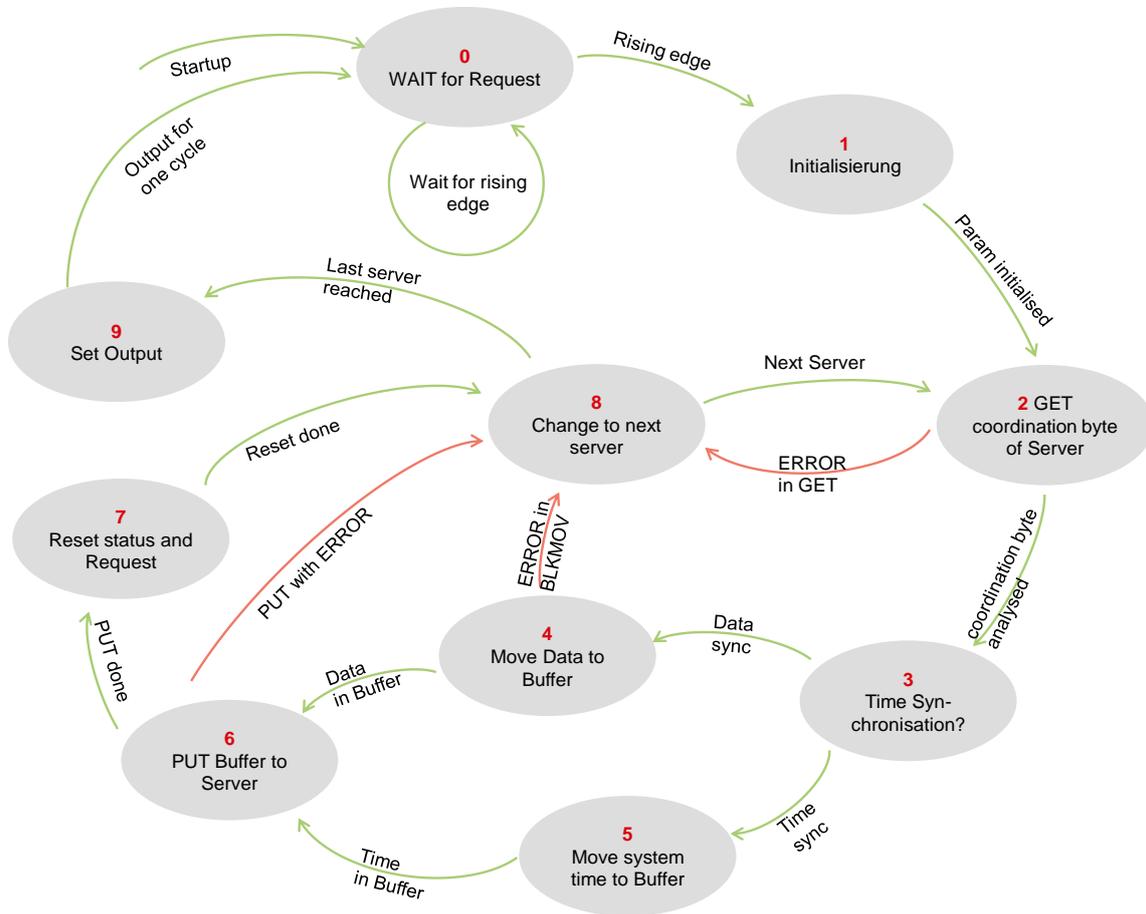
Tabelle 3-5

Byte	Bedeutung
Byte 4	Zeigt an, mit wie vielen Servern die Synchronisation fehlgeschlagen ist.
Byte 3	Anweisung, bei der der Error aufgetreten ist. <ul style="list-style-type: none">• 16#01: GET• 16#02: BLKMOV• 16#03: PUT
Byte 1+2	Zeigt den Status der zuletzt fehlgeschlagenen Anweisung an.

Interne Abläufe im FB Sync_Client2Serv (FB111)

Die Synchronisationsaufgabe ist im FB Sync_Client2Serv (FB111) als Zustandsautomat (Tabelle 3-6) realisiert.

Abbildung 3-5



Copyright © Siemens AG 2019 All rights reserved

Tabelle 3-6

Schritt	Aktion
0	Wartet auf eine steigende Flanke am Eingang REQ.
1	Setzt Parameter, wie <ul style="list-style-type: none"> den Request Eingang von PUT = 0. den Request Eingang von GET = 1 um das Koordinations-Byte abzuholen. ...
2	Das erhaltene Koordinations-Byte wird ausgewertet. Folgende Aktion wird je nach Status des Koordinations-Bytes durchgeführt: <ul style="list-style-type: none"> 16#F0 = Daten noch nicht verarbeitet: Abbruch der Synchronisation mit diesem Server. 16#FF = Uhrzeitsynchronisation noch nicht beendet: Abbruch der Synchronisation mit diesem Server. 16#00 = Server ist bereit Daten zu empfangen → Schritt 3 Zur Bedeutung der Koordinations-Bytes siehe Tabelle 3-7 .

3 Funktionsmechanismen der Applikation

3.2 Beschreibung des Programms der Kopfstation

Schritt	Aktion
3	Ist der Eingang Order_type <ul style="list-style-type: none"> TRUE: Uhrzeitsynchronisation → Schritt 5 FALSE: Übertragung von Nutzdaten → Schritt 4
4	Holt aus dem im DB Client2Serv_orders (DB 114) definierten Datenbereich per BLKMOV Daten ab und schreibt die Daten in den Puffer-DB Client2Serv_Buffer (DB115). Setzt zuletzt den Request Eingang der PUT-Anweisung.
5	Liest die Systemzeit und schreibt sie in den Puffer-DB Client2Serv_Buffer (DB115). Setzt zuletzt den Request Eingang der PUT-Anweisung.
6	<ul style="list-style-type: none"> PUT mit Fehler → Schritt 8 PUT ohne Fehler → Schritt 7 Die Anweisung PUT arbeitet die Aufträge an den Eingängen ADDR_x und SD_x sequentiell aufsteigend ab. Dadurch ist das gesicherte Senden zuerst der Nutzdaten und als zweites des Koordinations-Bytes in einem einzigen PUT-Aufruf möglich. → Datenkonsistenz
7	Rücksetzen des Status für den aktiven Server → Schritt 8
8	Wechsel des zu synchronisierenden Servers durch das Inkrementieren von control.number → Schritt 2 Wurden alle Server einmal versucht zu synchronisieren → Schritt 9
9	Setzt die Ausgangsparameter, siehe Tabelle 3-5 . → Schritt 0

Tabelle 3-7 Bedeutung der Koordinations-Bytes

Richtung \ Wert	16#00	16#F0	16#FF
	Client2Serv	Daten können geschrieben werden.	Daten wurden noch nicht verarbeitet.
Serv2Client	Es liegen keine neuen Daten an.	Neue Daten → Daten können gelesen werden.	---

3.2.3 FB Sync_Serv2Client (FB222): lesende Synchronisation

Überblick

Der FB Sync_Serv2Client realisiert die zyklische, lesende Synchronisierung der angegebenen Datenbereiche mit den S7-Servern.

Folgende Einzelfunktionen realisiert der Client zyklisch:

- Lesen und Auswerten des Koordinations-Bytes im Server.
- Abholen der Daten vom Server bei neuen Daten.
- Anzeige für den Server, dass die Daten abgeholt wurden.

Hier werden immer Datenbereiche mit einer Größe von 160 Byte abgeholt. Um die Größe des zu synchronisierenden Datenbereichs zu ändern, beachten Sie bitte Kapitel [6.1](#).

Parameter des FB Sync_Serv2Client (FB222)

Abbildung 3-6

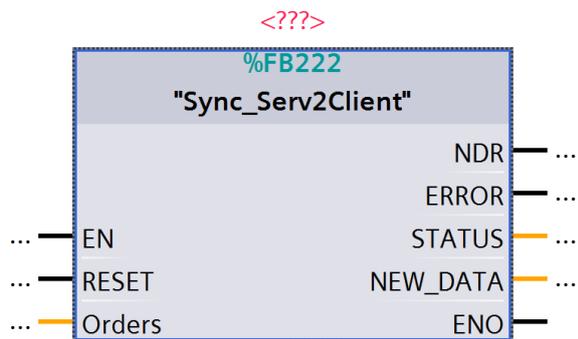


Tabelle 3-8

Parameter	Typ	Anmerkung
RESET	IN: BOOL	Ist RESET=TRUE, dann springt der Baustein in den Schritt 0: Initialisierung (Abbildung 3-7). Bei RESET=FALSE wird der Zustandsautomat des Bausteins normal durchlaufen.
Orders	INOUT: Serv2Client (siehe Kapitel 3.4)	Spezifiziert die Synchronisationsaufträge mit den Servern. Übergibt die Anzahl der zu synchronisierenden Server, die ID der Verbindung, DB-Nummer und Offset von Quelle und Ziel sowie die Länge der Nutzdaten.
NDR	OUT: BOOL	Zeigt nach einem erfolgreichen Synchronisationsdurchgang für einen Zyklus NDR=TRUE an.
ERROR	OUT: BOOL	Zeigt nach einem Synchronisationsdurchgang mit Fehler/n für einen Zyklus ERROR=TRUE an.
STATUS	OUT: DWORD	Gibt weitere Informationen zur Art des Fehlers. Der Ausgang STATUS bleibt ebenfalls für einen Zyklus anstehen.
NEW_DATA	OUT: INT	Zeigt bei NDR=TRUE oder ERROR=TRUE an, wie viele Datenbereiche in einem Durchgang synchronisiert wurden.

Der Ausgangsparameter STATUS

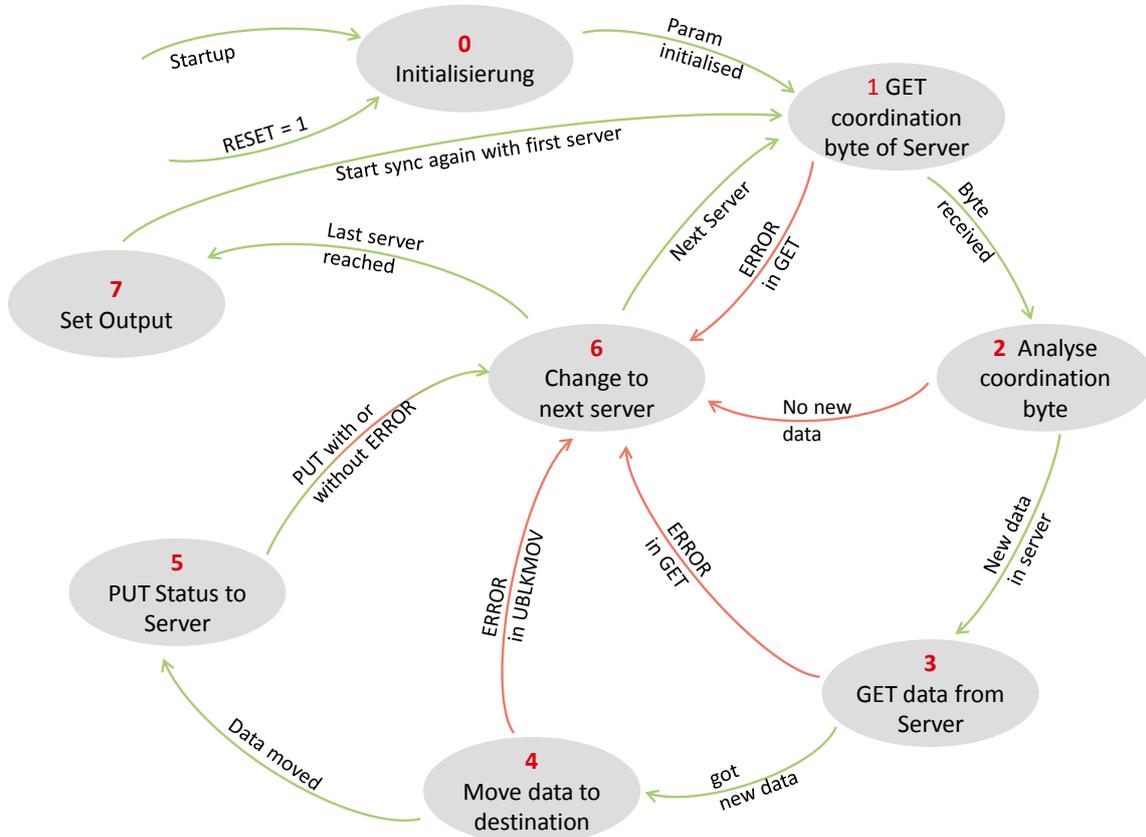
Tabelle 3-9

Wort	Bedeutung
Byte 4	Zeigt an, mit wie vielen Servern die Synchronisation fehlgeschlagen ist.
Byte 3	Anweisung, bei der der Error aufgetreten ist. <ul style="list-style-type: none"> • 16#01: GET Koordinations-Byte • 16#02: GET Daten • 16#03: PUT Koordinations-Byte • 16#04: UBLKMOV Daten
Byte 1+2	Zeigt den Status der zuletzt fehlgeschlagenen Anweisung an.

Interne Abläufe im FB Sync_Serv2Client (FB222)

Die Synchronisationsaufgabe ist im FB Sync_Serv2Client (FB222) als Zustandsautomat ([Abbildung 3-7](#)) realisiert.

Abbildung 3-7



Copyright © Siemens AG 2019 All rights reserved

Tabelle 3-10

Schritt	Aktion
0	Setzt Parameter wie <ul style="list-style-type: none"> den Request Eingang von GET = 0, um sicher eine positive Flanke zu erzeugen. control.number=1, um den ersten Server als aktiv zu markieren. ...
1	Setzt den Request-Eingang der ersten GET-Anweisung um das Koordinations-Byte auszulesen.
2	Analysiert das Koordinations-Byte des Servers: <ul style="list-style-type: none"> 16#F0: neue Daten des Servers liegen an → Schritt 4 16#00: keine neuen Daten vorhanden → Wechsel zum nächsten Server (Schritt 6) Zur Bedeutung der Koordinations-Bytes siehe Tabelle 3-7 .
3	Abholen der Daten aus dem Server durch Setzen des Request-Eingangs der zweiten GET-Anweisung.
4	Definieren des Koordinations-Bytes (16#00) und setzen des Request-Eingangs der PUT-Anweisung, um das Koordinations-Byte in den Server zu schreiben.

Schritt	Aktion
5	Holen der Daten aus dem Puffer und Ablage im Ziel-DB mit der Anweisung UBLKMOV.
6	Wechsel des zu synchronisierenden Servers durch das Inkrementieren von control.number → Schritt 1 Wurden alle Server einmal versucht zu synchronisieren wird der erste Server als aktiv markiert → dann Sprung zu Schritt 7
7	Setzen der Ausgangsparameter → Schritt 1 für neuen Synchronisationsdurchlauf.

3.3 Beschreibung des Programms einer Substation

3.3.1 Programmübersicht

Funktionen

Der Server reagiert nur passiv auf die Synchronisation durch den Client. Die Bausteine PUT/GET werden nur im Client (Kopfstation) aufgerufen und nur der Client kann das Ende eines Schreib-/Lesevorgangs feststellen.

Der passive Server enthält keine aktiven Kommunikationsbausteine und muss die Koordination der Zugriffe durch das Anwenderprogramm steuern.

Folgende Funktionen sind im Anwenderprogramm des Servers (Substation) realisiert:

- Koordinieren der Entsorgung der Daten vom Client (schreibende Synchronisation zur Substation).
- Koordinieren der Versorgung der Daten zum Client (lesende Synchronisation von der Substation).
- Stellen der Systemzeit bei einem Zeit-Synchronisationsauftrag.

Eine grafische Programmübersicht ist in [Abbildung 3-3](#) beschrieben.

Bausteine und Anweisungen in den Substationen

Tabelle 3-11

Element	symbolischer Name	Beschreibung
OB1	Main	Beinhaltet das Hauptprogramm. <ul style="list-style-type: none"> • Ruft den FB Sync_Server (FB2) auf. • Realisiert durch den Aufruf eine Empfangslogik zur Synchronisation mit dem Client.
FB2	Sync_Server	Reagiert auf die Synchronisation vom Client. Dem FB muss übergeben werden, welche DBs synchronisiert werden. Bei einer Abweichung vom Applikationsbeispiel muss das Anwenderprogramm angepasst werden (siehe Kapitel 6).
FC1200	DnT_DTL	Wandelt den Datentyp Date_and_Time in den Datentyp DTL um.
DB2	Sync_Server_DB	Instanz-DB des FB Sync_Server (FB2)
DB6	Data_FromClient	Ablageort der Daten vom Client.
DB7	Data_ToClient	Ablageort der Daten für den Client.

3.3.2 FB Sync_Server (FB111)

Überblick

Der FB Sync_Server (FB111) reagiert auf Änderungen des Koordinations-Bytes und stellt so eine reibungslose Synchronisation mit dem Client sicher.

Für Änderungen und Anpassungen im Anwenderprogramm beachten Sie Kapitel [6](#).

Parameter des FB Sync_Server (FB111)

Abbildung 3-8

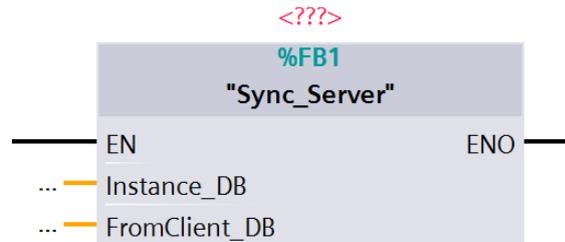


Tabelle 3-12

Parameter	Typ	Anmerkung
Instance_DB	IN: INT	Gibt die Nummer des Instanz-Datenbausteins des FBs an, um bei einer Uhrzeitsynchronisation die Uhrzeit im Instanz-Datenbaustein zwischenspeichern.
FromClient_DB	IN: INT	Gibt die Nummer des DBs Data_FromClient an, um bei einer Uhrzeitsynchronisation die Uhrzeit aus dem Datenbaustein herauszulesen.

Reaktion auf schreibende Synchronisation

Abbildung 3-9

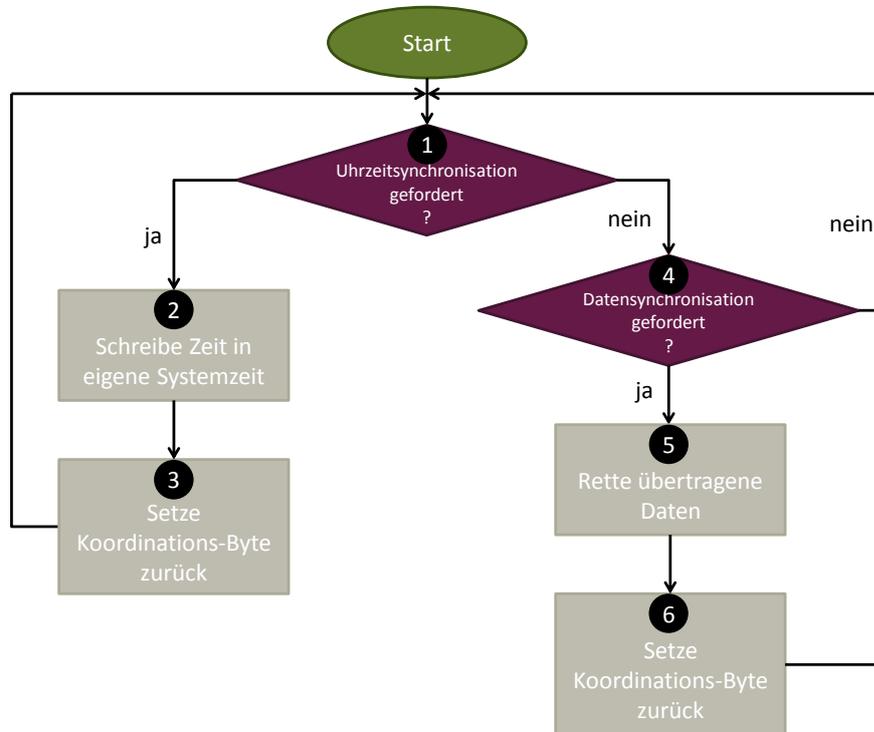


Tabelle 3-13

Nr.	Vorgehen
1	Wenn das Koordinations-Byte den Wert 16#FF besitzt, dann hat der Client die Uhrzeit in den Empfangspuffer abgelegt.
2	<ul style="list-style-type: none"> Die empfangene Uhrzeit wird in den Instanz-DB übertragen. Die Uhrzeit wird vom Datentyp Date_and_Time in den Datentyp DTL gewandelt. Die Uhrzeit wird im Server als Systemzeit übernommen.
3	Das Koordinations-Byte wird zurückgesetzt, der Synchronisationsauftrag ist beendet.
4	Wenn das Koordinations-Byte den Wert 16#F0 besitzt, dann hat der Client Nutzdaten im Zielbereich abgelegt.
5	Die Daten im Empfangspuffer können weiterverarbeitet werden.
6	Das Koordinations-Byte wird zurückgesetzt, damit der Client weiß, dass die Daten entsorgt wurden.

Reaktion auf lesende Synchronisation

Abbildung 3-10

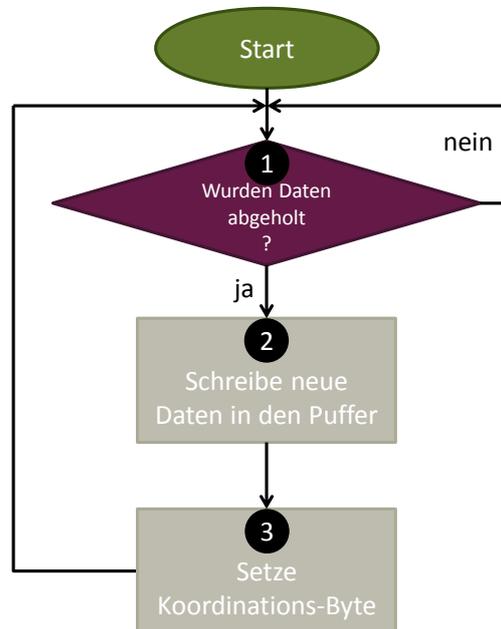


Tabelle 3-14

Nr.	Vorgehen
1	Wenn das Koordinations-Byte den Wert 16#00 besitzt, dann hat der Client die Daten im Sendepuffer bereits abgeholt.
2	Es können nun neue Daten in den Sendepuffer geschrieben werden.
3	Das Koordinations-Byte wird auf 16#F0 gesetzt um dem Client anzuzeigen, dass neue Daten bereit liegen.

3.3.3 Datenkonsistenz im Server

Definition

Der Begriff Datenkonsistenz bezieht sich darauf, dass die Daten, die zwischen Client und Server synchronisiert werden, in einem korrekten Zustand übertragen werden.

Dateninkonsistenz kann entstehen, wenn zwei Prozesse auf denselben Speicherbereich asynchron zugreifen. Ein mögliches Szenario ist, dass der deutlich schnellere Lesezugriff den Schreibzugriff "überholt" und so Daten gelesen werden, die aus den alten und neuen Werten bestehen und damit inkonsistent sind.

Voraussetzung

Bei der Kommunikation mit PUT- und GET-Funktionen handelt es sich um unkoordinierte Kommunikation, da der Server nicht direkt auf die Aufträge des Clients reagieren kann. Im Unterschied zur Kommunikation mit BSEND und BRCV bei der abgefragt werden kann, ob noch Daten gesendet oder empfangen werden, muss bei den PUT- und GET-Funktionen im Anwenderprogramm für eine Koordination der Zugriffe gesorgt werden.

Maßnahmen

Die Zugriffe im Anwenderprogramm werden über das "Koordinations-Byte" gesteuert. Dessen genaue Bedeutung sowie Maßnahmen zum Erhalt der Datenkonsistenz des Programmes erhalten Sie in Kapitel [6.3](#) und Kapitel [6.4](#).

3.4 Die PLC-Datentypen

Übersicht

Die folgenden PLC-Datentypen werden im Anwenderprogramm des Clients (Kopfstation) verwendet:

- "order"
- "Client2Serv"
- "Serv2Client"
- "Any_struct"

Dieses Kapitel beschreibt den Aufbau und die Verwendung der Datentypen.

Der PLC-Datentyp "order"

Die [Abbildung 3-11](#) zeigt den Aufbau des PLC-Datentypen "order". Der PLC-Datentyp wird sowohl vom PLC-Datentyp Serv2Client als auch Client2Serv verwendet und bildet einen Synchronisationsauftrag mit einem Server ab.

Abbildung 3-11

Name	Data type	Def...	Vi
ID	CONN_PRG		
DB_nr_cl	Int	0	
DBB_nr_cl	Int	0	
DB_nr_s	Int	0	
DBB_nr_s	Int	0	
length	Int	160	
Status	DWord	16#0	

Tabelle 3-15

Name	Anmerkung									
ID	<p>ID der projektierten S7-Verbindung. Kann aus der Netzsicht des TIA-Portals unter dem Reiter "Verbindungen" entnommen werden.</p> <table border="1"> <thead> <tr> <th>Local connection name</th> <th>Local end point</th> <th>Local ID (hex)</th> </tr> </thead> <tbody> <tr> <td>S7_Connection_50</td> <td>S7-1500 [CPU...</td> <td>100</td> </tr> <tr> <td>S7_Connection_51</td> <td>S7-1500 [CPU...</td> <td>101</td> </tr> </tbody> </table> <p>Hinweis! Die ID ist in der Netzsicht in hexadezimaler Schreibweise angegeben.</p>	Local connection name	Local end point	Local ID (hex)	S7_Connection_50	S7-1500 [CPU...	100	S7_Connection_51	S7-1500 [CPU...	101
Local connection name	Local end point	Local ID (hex)								
S7_Connection_50	S7-1500 [CPU...	100								
S7_Connection_51	S7-1500 [CPU...	101								
DB_nr_cl	Gibt die Nummer des Datenbausteins im Client an, dessen Daten synchronisiert werden sollen.									
DBB_nr_cl	Gibt den Byte-Offset der zu synchronisierenden Daten an.									
DB_nr_s	Gibt die Nummer des Datenbausteins im Server an, dessen Daten synchronisiert werden sollen.									
DBB_nr_s	Gibt den Byte-Offset der zu synchronisierenden Daten an.									
length	<p>Gibt die Länge (in Byte) der zu synchronisierenden Daten an. Maximale Größe der Nutzdaten: 160 Byte.</p> <p>Hinweis! Bei der schreibenden Synchronisation muss die Längenangabe für alle Server identisch sein.</p>									
Status	Tritt ein Fehler während der Synchronisation mit dem Server auf, so wird der Status der Anweisung hier gespeichert, bis ein erfolgreicher Synchronisationsdurchgang den Wert auf 16#0 zurücksetzt.									

ACHTUNG PUT und GET können nicht auf optimierte Datenbausteine zugreifen!

Der PLC-Datentyp "Client2Serv"

Der PLC Datentyp "Client2Serv" beinhaltet für den Client alle relevanten Informationen zur schreibenden Synchronisation mit den Servern.

Abbildung 3-12

Name	Datentyp
No_of_Servers	Int
Client2Serv	array[1..4] of "order"

Tabelle 3-16

Name	Anmerkung
No_of_Servers	Beinhaltet die Anzahl der schreibend zu synchronisierenden Server.
Client2Serv	Ein Array aus dem PLC-Datentyp "order". Für jeden zu synchronisierenden Server wird ein Array-Element verwendet. Die Größe des Arrays sollte dem Wert der Variablen "No_of_Servers" entsprechen.

In der S7-1500 CPU wird ein Datenbaustein direkt aus dem PLC-Datentypen "Client2Serv" generiert und dann am Eingang "Orders" des FB Sync_Client2Serv verschaltet.

Der PLC-Datentyp "Serv2Client"

Der PLC Datentyp "Serv2Client" beinhaltet für den Client alle relevanten Informationen zur lesenden Synchronisation mit den Servern.

Abbildung 3-13

Name	Datentyp	Def
No_of_Servers	Int	4
Serv2Client	array[1..4] of "order"	

Tabelle 3-17

Name	Anmerkung
No_of_Servers	Beinhaltet die Anzahl der zu lesend zu synchronisierenden Server.
Serv2Client	Ein Array aus dem PLC-Datentyp "order". Für jeden zu synchronisierenden Server wird ein Array-Element verwendet. Die Größe des Arrays sollte dem Wert der Variablen "No_of_Servers" entsprechen.

3.4 Die PLC-Datentypen

In der S7-1500 CPU wird ein Datenbaustein direkt aus dem PLC-Datentypen "Serv2Client" generiert und dann am Eingang "Orders" des FB Sync_Serv2Client verschaltet.

In der S7-300 CPU wird in einen Datenbaustein der PLC-Datentyp eingefügt und dann am Eingang "Orders" des FB Sync_Serv2Client verschaltet. Eine direkte Verschaltung des gesamten DBs wie in der S7-1500 CPU ist bei einer S7-300 CPU nicht möglich.

Der PLC-Datentyp "Any_struct"

Der PLC-Datentyp "Any_struct" bildet die Struktur des Anypointers ab und wird im Projekt als AT-Konstrukt an Any-Variablen verwendet.

Dadurch kann der Anypointer komfortabler belegt oder modifiziert werden. Der Aufbau des Anypointers kann dem Handbuch [\[8\]](#) entnommen werden.

Unterschiede in den PLC-Datentypen bei einer S7-300 CPU als Client

1. Die S7-300 verwendet nicht den Datentyp CONN_PRG sondern INT ([Abbildung 3-11](#))
2. In der S7-300 CPU wird in einen Datenbaustein der PLC-Datentyp eingefügt und dann am Eingang "Orders" des FB Sync_Client2Serv verschaltet. Eine direkte Verschaltung des gesamten DBs wie in der S7-1500 CPU ist bei einer S7-300 CPU nicht möglich. ([Tabelle 3-18](#))

Tabelle 3-18

	S7-1500	S7-300
Client2Serv_orders		
Verschaltung	<p>%DB114</p> <p>"Client2Serv_orders" — Orders</p>	<p>P#DB114.DBX0.0</p> <p>"Client2Serv_orders". Client2serv_orders — Orders</p>

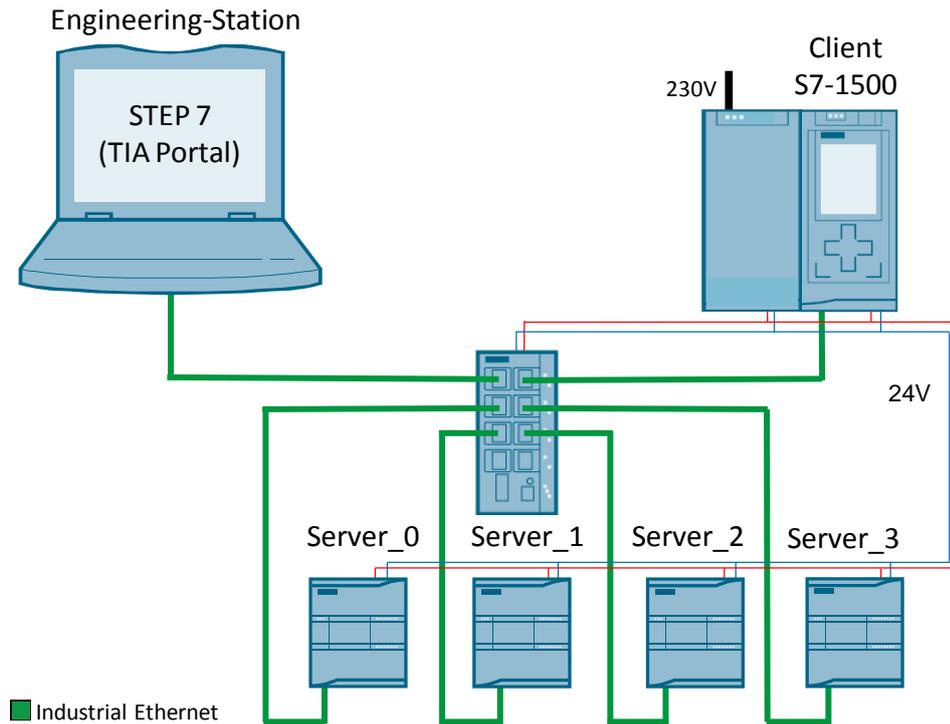
4 Inbetriebnahme der Applikation

4.1 Aufbau der Hardware

Übersicht

Nachfolgendes Bild zeigt den Hardwareaufbau des Projekts mit der S7-1500 als Client.

Abbildung 4-1



Die folgende Tabelle beschreibt das Vorgehen für den Hardwareaufbau des Projektes.

Beachten Sie dabei die Vorschriften für den Aufbau einer S7-Station.

Hardwareaufbau der SIMATIC S7-CPU

Tabelle 4-1

Nr.	Vorgehen	Anmerkung
1.	Stecken Sie das Power Supply und die CPUs auf den entsprechenden Bauträger.	Profilschiene für die <ul style="list-style-type: none"> S7-300: z. B. 6ES7390-1AE80-0AA0 S7-1500: z. B. 6ES7590-1AF30-0AA0 S7-1200: DIN-Hutschiene
2.	Montieren Sie den Switch auf eine Hutschiene.	
3.	Verdrahten Sie die CPUs mit dem Power Supply.	Achten Sie auf die richtige Polung!

4 Inbetriebnahme der Applikation

4.2 Konfiguration der Hardware

Nr.	Vorgehen	Anmerkung
4.	Verbinden Sie Ihre CPUs per Ethernet-Kabel mit dem Switch.	
5.	Verbinden Sie Ihren Power Supply mit dem Stromnetz (230V Wechselstrom)	
6.	Verbinden Sie Ihre Engineering Station ebenfalls per Ethernet Kabel mit dem Switch.	

4.2 Konfiguration der Hardware

Verwendete IP-Adressen

Die verwendete Subnetzmaske ist die 255.255.255.0

Tabelle 4-2

CPU	IP-Adresse
S7-300	192.168.0.252
S7-1500	192.168.0.253
Server_0	192.168.0.6
Server_1	192.168.0.7
Server_2	192.168.0.8
Server_3	192.168.0.9

Anpassen der IP-Adresse der S7-1500 CPU

Tabelle 4-3

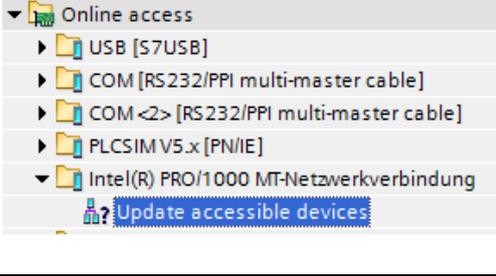
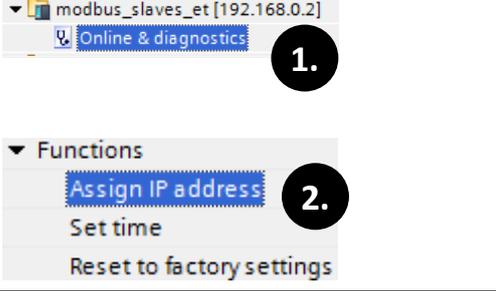
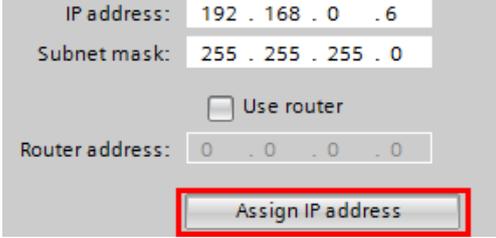
Nr.	Vorgehen	Anmerkung
1.	Navigieren Sie über das Display Ihrer S7-1500 CPU zum Punkt "IP-Adresse einstellen". Einstellungen> Adressen> X1 (IE/PN) ("Settings > Addresses >X1 (IE/PN) > IP address")	Im Display können Sie über das Steuerkreuz und die Tasten "OK" und "ESC" navigieren.
2.	Stellen Sie die im Beispielprojekt verwendete IP-Adresse der S7-1500 ein (192.168.0.253).	

4 Inbetriebnahme der Applikation

4.2 Konfiguration der Hardware

Anpassen der IP-Adresse der S7-300 und S7-1200 CPUs

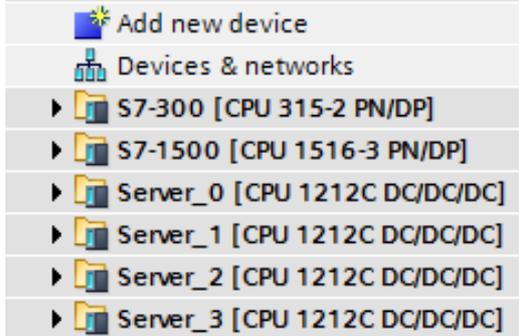
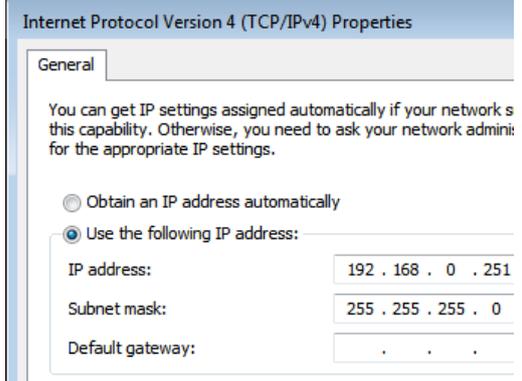
Tabelle 4-4

Nr.	Vorgehen	Anmerkung
1.	<p>Öffnen Sie das TIA Portal in der Projektsicht. Suchen Sie nach "Erreichbaren Teilnehmern". Navigieren Sie dazu in "Projektnavigation> Online-Zugänge> [Ihr_Ethernet_Adapter]> Erreichbare Teilnehmer aktualisieren" ("Project Tree> Online Access> [Your_Ethernet_Adapter]> Update accessible devices") Ihre S7-CPU's werden nun erkannt.</p>	
2.	<p>Das Vorgehen ist für alle CPU's identisch: Navigieren Sie nun zu "[Ihre_S7CPU]> Online&Diagnose" ("[Your_S7CPU]> Online&Diagnostics") Im grafischen Bereich von "Online&Diagnose" wählen Sie nun "Funktionen> IP Adresse zuweisen" ("Functions> Assign IP address")</p>	
3.	<p>Geben Sie für jede CPU die im Projekt verwendete IP Adresse ein (siehe Inspektorfenster in der Gerätesicht). Bestätigen Sie die Aktion mit "IP-Adresse zuweisen" ("Assign IP address") Die einzelnen Stationen erhalten so nacheinander die IP-Adressen von Ihrer Engineering Station zugewiesen.</p>	

4.3 Öffnen und Laden des STEP 7-Projekts

Die folgende Tabelle zeigt Ihnen, wie Sie das STEP 7-Projekt öffnen und in Ihre S7-Station laden.

Tabelle 4-5

Nr.	Vorgehen	Anmerkung
1.	Entzippen Sie das TIA Portal Projekt (siehe Tabelle 2-4) in einen lokalen Ordner ihres PCs und öffnen es.	
2.	Stellen Sie sicher, dass sich Ihre Engineering Station im selben Subnetz wie die S7-CPU's befindet. Beispiel: IP-Adresse: 192.168.0.251 Subnetzmaske: 255.255.255.0	
3.	Laden Sie über den Button "Laden in Gerät" einzeln in jede CPU die Projektierung. <ul style="list-style-type: none"> • S7-300 oder S7-1500 • Server_0 • Server_1 • Server_2 • Server_3 Nach dem Download erscheint die Meldung, dass der Ladevorgang erfolgreich beendet wurde.	
4.	Nun können Sie die Applikation, wie unter Kapitel 5 beschrieben, beobachten.	

5 Bedienung der Applikation

5.1 Beobachtung der Applikation

Beobachten

Das Anwenderprogramm von Client und Server erzeugt Daten zur Übersicht über das Programm und zur Fehlerdiagnose.

Diese Daten können einzeln, zum Beispiel aus den Instanz-DBs, gelesen werden.

Zur besseren Übersichtlichkeit wurde sowohl im Client, als auch in den Servern eine Beobachtungstabelle erstellt, die einen Überblick über den Zustand des Programms erlaubt.

Die Beobachtungstabellen können Sie an Ihre Anforderungen anpassen.

Beobachtungstabelle Overview_Client

Die Beobachtungstabelle macht Aussagen über den Zustand des Anwenderprogramms des Clients. Zusätzlich können Sie über die Beobachtungstabelle Ihr Programm steuern, sofern Sie die Steuerung nicht durch eine zusätzliche Logik im Anwenderprogramm realisiert haben.

Tabelle 5-1

Variable	Datentyp	Anmerkung
*.P_REQ	BOOL	Request-Eingang für den FB Sync_Client2Serv (FB111)
*.P_TIME	BOOL	Time-Eingang für den FB Sync_Client2Serv (FB111)
*.P_ERROR	BOOL	Error-Ausgang des FB Sync_Client2Serv (FB111)
*.P_STATUS	DWORD	Status-Ausgang des FB Sync_Client2Serv (FB111)
*.G_NDR	BOOL	NDR Ausgang des FB Sync_Serv2Client (FB222)
*.G_New_Data	INT	New-Data Ausgang des FB Sync_Serv2Client (FB222)
*.coord	BYTE	Koordinations-Byte Client to Server
*.Client2Serv_Buffer[0]	BYTE	Erstes Buffer-Byte der Nutzdaten
*.coord	BYTE	Koordinations-Byte Server to Client
*.Serv2Client_Buffer[0]	BYTE	Erstes Buffer-Byte der Nutzdaten
*[1].Status	DWORD	Status der Client to Server-Synchronisation mit dem Server_0.
*[2].Status	DWORD	Status der Client to Server-Synchronisation mit dem Server_1.
*[3].Status	DWORD	Status der Client to Server-Synchronisation mit dem Server_2.
*[4].Status	DWORD	Status der Client to Server-Synchronisation mit dem Server_3.
*[1].Status	DWORD	Status der Server to Client-Synchronisation mit dem Server_0.
*[2].Status	DWORD	Status der Server to Client-Synchronisation mit dem Server_1.
*[3].Status	DWORD	Status der Server to Client-Synchronisation mit dem Server_2.
*[4].Status	DWORD	Status der Server to Client-Synchronisation mit dem Server_3.

Beobachtungstabelle Overview_Server_x

Die Beobachtungstabelle macht Aussagen über den Zustand des Anwenderprogramms des Servers.

Sie können die Beobachtungstabelle beliebig anpassen.

Tabelle 5-2

Variable	Datentyp	Anmerkung
*.Coordination_Byte	BYTE	Koordinations-Byte der Client to Server-Synchronisation
*.buffer[0]	BYTE	Erstes Byte der Nutzdaten
*.buffer[1]	BYTE	Zweites Byte der Nutzdaten
*.Coordination_Byte	BYTE	Koordinations-Byte der Server to Client-Synchronisation
*.buffer[0]	BYTE	Erstes Byte der Nutzdaten
*.buffer[1]	BYTE	Zweites Byte der Nutzdaten

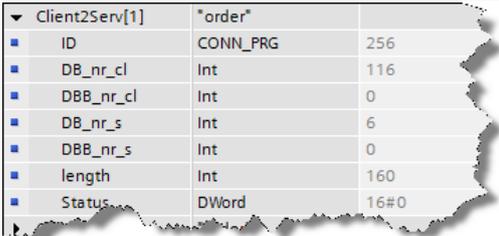
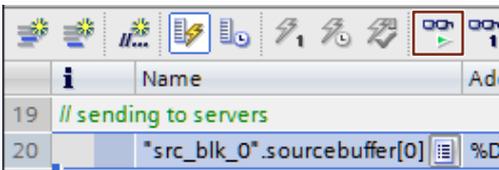
5.2 Szenario A: schreibende Datensynchronisation

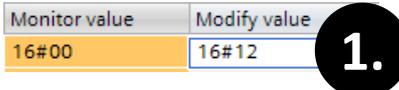
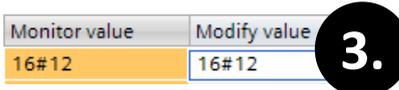
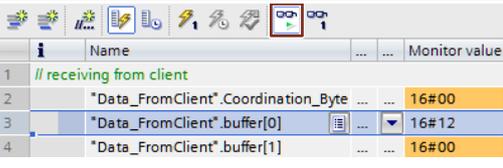
Szenario

Dieses Kapitel beschreibt das Vorgehen zur Synchronisation eines Datenbereichs des Clients mit dem Datenbereich eines Servers. Außerdem wird die Beobachtung des Vorganges beschrieben.

Vorgehen

Tabelle 5-3

Nr.	Vorgehen	Anmerkung
1.	Nehmen Sie das Beispielprojekt, wie unter Kapitel 4 beschrieben, in Betrieb.	
2.	Überprüfen Sie die Parameter des ersten Auftrags im DB Client2Serv_orders (DB114)	
3.	Öffnen Sie die Beobachtungstabelle "Overview_Client" und betätigen Sie die Schaltfläche "Alle beobachten".	

Nr.	Vorgehen	Anmerkung
4.	Steuern Sie den Wert der Variablen "src_blk_0".sourcebuffer[0] (Zeile 20) auf einen beliebigen Wert, zum Beispiel 16#12. Steuern Sie jetzt den Wert, um ihn in die CPU zu übertragen.	  
5.	Erzeugen Sie, ebenfalls über die Beobachtungstabelle, eine positive Flanke der Variablen "Param_Main_Call".P_REQ. Dadurch wird der Synchronisationsvorgang angestoßen und die Aufträge des DB "Client2Serv_orders" (DB114) abgearbeitet.	
6.	Wechseln Sie in den Server_0 des Projekts. Öffnen Sie im Server_0 die Beobachtungstabelle "Overview_Server_0" und betätigen Sie die Schaltfläche "Alle beobachten". Der Inhalt des buffer[0] hat sich bei erfolgreicher Übertragung auf den im Client eingegebenen Wert geändert.	

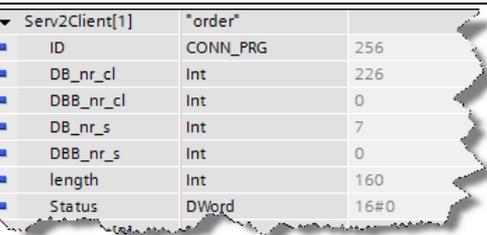
5.3 Szenario B: lesende Synchronisation

Szenario

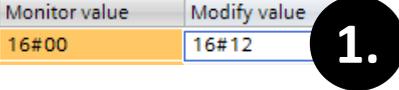
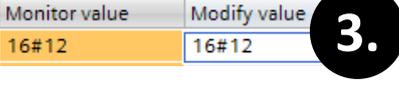
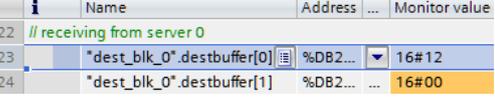
Dieses Kapitel beschreibt das Vorgehen zur Synchronisation eines Datenbereichs eines Servers mit einem Client. Außerdem wird die Beobachtung des Vorgangs beschrieben.

Vorgehen

Tabelle 5-4

Nr.	Vorgehen	Anmerkung
1.	Nehmen Sie das Beispielprojekt, wie unter Kapitel 4 beschrieben, in Betrieb.	
2.	Überprüfen Sie die Parameter des ersten Auftrags im DB "Serv2Client_orders" (DB224)	

5 Bedienung der Applikation

Nr.	Vorgehen	Anmerkung
3.	<p>Öffnen Sie die Beobachtungstabelle "Overview_Server_0".</p> <p>Steuern Sie den Wert der Variablen "Data_ToClient.buffer[0]" (Zeile 7), auf einen beliebigen Wert, zum Beispiel 16#12.</p> <p>Steuern Sie den Wert, um ihn in die CPU zu übertragen.</p>	  
4.	<p>Wechseln Sie wieder zurück in den Client.</p> <p>Öffnen Sie die Beobachtungstabelle "Overview_Client".</p> <p>Beobachten Sie alle Variablen.</p> <p>Der Inhalt der Variablen "dest_blk_0".destbuffer[0] hat sich bei erfolgreicher Übertragung der Daten auf den im Server vorgegebenen Wert geändert.</p>	

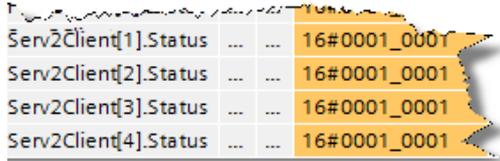
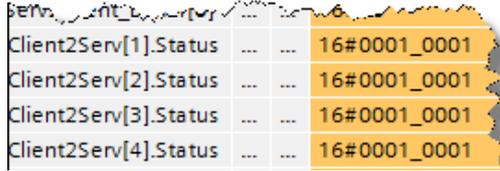
5.4 Szenario C: Fehlermeldung bei gezogenem Kabel

Szenario

Dieses Kapitel beschreibt das Verhalten des Programms, wenn die Verbindung vom Client zu den Servern z.B. durch falsche Verkabelung unterbrochen ist.

Beschreibung

Tabelle 5-5

Nr.	Vorgehen	Anmerkung
1.	Nehmen Sie das Beispielprojekt, wie unter Kapitel 4 beschrieben, in Betrieb.	
2.	Ziehen Sie das Ethernet-Kabel, das die Client CPU mit den Server-CPU's verbindet.	
3.	Öffnen Sie die Beobachtungstabelle "Overview_Client". Die zyklische lesende Synchronisation sollte bereits den Status 16#0001_0001 geschrieben haben. Bedeutung: <ul style="list-style-type: none"> • Erstes Wort: Die erste Anweisung schlug fehl. • Zweites Wort: Der Status des Fehlers der fehlgeschlagenen Anweisung ist 16#0001 	 <p>In der Hilfe zur Anweisung GET im TIA Portal wird als Möglichkeit der Fehlerbehebung für den Status 16#0001 angegeben, die Verkabelung zu überprüfen.</p>
4.	Erzeugen Sie über die Beobachtungstabelle eine positive Flanke der Variablen "Param_Main_Call".P_REQ. Dadurch wird der schreibende Synchronisationsvorgang angestoßen und die Aufträge des DB "Client2Serv_orders" (DB114) abgearbeitet.	
5.	Dadurch, dass die Aufträge nicht korrekt abgearbeitet werden können, wird als Ergebnis ebenfalls der Status der schreibenden Synchronisation mit dem Wert 16#0001_0001 belegt. Zur Bedeutung siehe Nr. 3.	

6 Änderungen am Projekt

Überblick

Wenn Sie Änderungen am STEP 7 (TIA Portal)-Projekt vornehmen wollen, dann bietet Ihnen dieses Kapitel Unterstützung.

Die folgenden Anpassungen sind dokumentiert:

- Ändern der zu synchronisierenden Datenbereiche.
- Hinzufügen eines weiteren Slaves.
- Sicherstellen der Datenkonsistenz im Server.
- Erweitern der Funktion des Koordination-Bytes.

6.1 Ändern der zu synchronisierenden Datenbereiche

Übersicht

Die Beispielapplikation legt auf Anforderung 160 Byte aus dem DB116 des Clients in den DB6 der Server ab.

Aus dem DB7 der Server werden 160 Byte in die folgenden DBs abgelegt:

- vom Server_0 in den DB226
- vom Server_1 in den DB227
- vom Server_2 in den DB228
- vom Server_3 in den DB229

Wenn Sie diese Quell- und Zielbereiche abändern wollen, dann befolgen Sie die Anweisungen in diesem Kapitel.

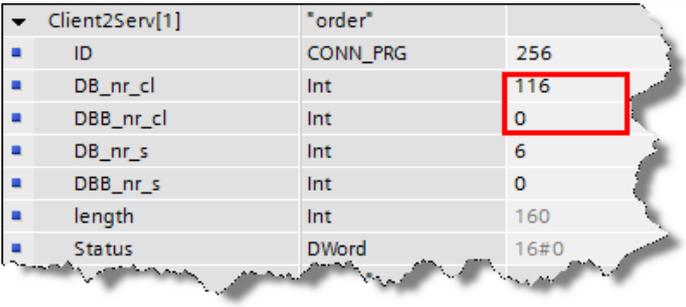
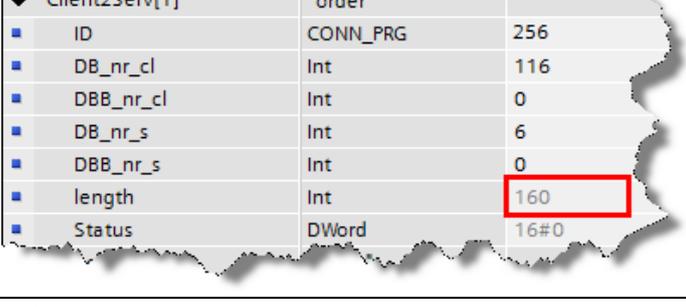
Die Vorgehensweise ist jeweils nacheinander zuerst für die Client- und dann für die Server-Seite beschrieben.

Vorgehen im Client bei schreibender Synchronisation

Tabelle 6-1

Nr.	Vorgang	Anmerkung
1.	Öffnen Sie den PLC-Datentyp "Client2Serv".	
2.	Öffnen Sie den Auftrag, den Sie abändern möchten.	

6 Änderungen am Projekt

Nr.	Vorgang	Anmerkung																								
3.	<p>Passen Sie die Nummer des Client-DB und dessen Offset an.</p> <ul style="list-style-type: none"> DB_nr_cl gibt den DB im Client an, aus dem die 160 Byte gelesen werden. DBB_nr_cl gibt den Byte-Offset an, ab dem die Daten gelesen werden. 	 <table border="1"> <thead> <tr> <th>Client2Serv[1]</th> <th>*order*</th> <th></th> </tr> </thead> <tbody> <tr> <td>ID</td> <td>CONN_PRG</td> <td>256</td> </tr> <tr> <td>DB_nr_cl</td> <td>Int</td> <td>116</td> </tr> <tr> <td>DBB_nr_cl</td> <td>Int</td> <td>0</td> </tr> <tr> <td>DB_nr_s</td> <td>Int</td> <td>6</td> </tr> <tr> <td>DBB_nr_s</td> <td>Int</td> <td>0</td> </tr> <tr> <td>length</td> <td>Int</td> <td>160</td> </tr> <tr> <td>Status</td> <td>DWord</td> <td>16#0</td> </tr> </tbody> </table>	Client2Serv[1]	*order*		ID	CONN_PRG	256	DB_nr_cl	Int	116	DBB_nr_cl	Int	0	DB_nr_s	Int	6	DBB_nr_s	Int	0	length	Int	160	Status	DWord	16#0
Client2Serv[1]	*order*																									
ID	CONN_PRG	256																								
DB_nr_cl	Int	116																								
DBB_nr_cl	Int	0																								
DB_nr_s	Int	6																								
DBB_nr_s	Int	0																								
length	Int	160																								
Status	DWord	16#0																								
4.	<p>Passen Sie die Nummer des Server-DB und dessen Offset an.</p> <ul style="list-style-type: none"> DB_nr_s gibt den DB im Server an, in den die Daten geschrieben werden. DBB_nr_cl gibt den Byte-Offset an, ab dem die Daten geschrieben werden. 	 <table border="1"> <thead> <tr> <th>Client2Serv[1]</th> <th>*order*</th> <th></th> </tr> </thead> <tbody> <tr> <td>ID</td> <td>CONN_PRG</td> <td>256</td> </tr> <tr> <td>DB_nr_cl</td> <td>Int</td> <td>116</td> </tr> <tr> <td>DBB_nr_cl</td> <td>Int</td> <td>0</td> </tr> <tr> <td>DB_nr_s</td> <td>Int</td> <td>6</td> </tr> <tr> <td>DBB_nr_s</td> <td>Int</td> <td>0</td> </tr> <tr> <td>length</td> <td>Int</td> <td>160</td> </tr> <tr> <td>Status</td> <td>DWord</td> <td>16#0</td> </tr> </tbody> </table>	Client2Serv[1]	*order*		ID	CONN_PRG	256	DB_nr_cl	Int	116	DBB_nr_cl	Int	0	DB_nr_s	Int	6	DBB_nr_s	Int	0	length	Int	160	Status	DWord	16#0
Client2Serv[1]	*order*																									
ID	CONN_PRG	256																								
DB_nr_cl	Int	116																								
DBB_nr_cl	Int	0																								
DB_nr_s	Int	6																								
DBB_nr_s	Int	0																								
length	Int	160																								
Status	DWord	16#0																								
5.	<p>Sie können auch die Länge des zu synchronisierenden Datenbereiches in Byte angeben.</p> <p>Beachten Sie, dass maximal 160 Byte synchronisiert werden können.</p>	 <table border="1"> <thead> <tr> <th>Client2Serv[1]</th> <th>*order*</th> <th></th> </tr> </thead> <tbody> <tr> <td>ID</td> <td>CONN_PRG</td> <td>256</td> </tr> <tr> <td>DB_nr_cl</td> <td>Int</td> <td>116</td> </tr> <tr> <td>DBB_nr_cl</td> <td>Int</td> <td>0</td> </tr> <tr> <td>DB_nr_s</td> <td>Int</td> <td>6</td> </tr> <tr> <td>DBB_nr_s</td> <td>Int</td> <td>0</td> </tr> <tr> <td>length</td> <td>Int</td> <td>160</td> </tr> <tr> <td>Status</td> <td>DWord</td> <td>16#0</td> </tr> </tbody> </table>	Client2Serv[1]	*order*		ID	CONN_PRG	256	DB_nr_cl	Int	116	DBB_nr_cl	Int	0	DB_nr_s	Int	6	DBB_nr_s	Int	0	length	Int	160	Status	DWord	16#0
Client2Serv[1]	*order*																									
ID	CONN_PRG	256																								
DB_nr_cl	Int	116																								
DBB_nr_cl	Int	0																								
DB_nr_s	Int	6																								
DBB_nr_s	Int	0																								
length	Int	160																								
Status	DWord	16#0																								
6.	<p>Wenn Sie die Länge des zu synchronisierenden Bereichs verkleinern, dann müssen Sie auch den Puffer im DB Client2Serv_Buffer (DB115) auf diese Größe verringern.</p>	<p>Achtung! Der Puffer im DB Client2Serv_Buffer (DB115) wird von allen Aufträgen verwendet. Der Parameter length muss deshalb bei der schreibenden Synchronisation bei allen Aufträgen denselben Wert besitzen!</p>																								
7.	<p>Stellen Sie sicher, dass der unter 3. angegebene DB auf dem Client vorhanden ist.</p>																									
8.	<p>Laden Sie ihr Anwenderprogramm neu in die CPU und starten Sie die CPU neu.</p>																									

Vorgehen im Server bei schreibender Synchronisation

Tabelle 6-2

Nr.	Vorgang	Anmerkung
1.	<p>Erstellen Sie einen DB mit der in Tabelle 6-1 unter Nr. 4 angegebenen Datenbausteinnummer und dem symbolischen Namen</p>	

6 Änderungen am Projekt

Nr.	Vorgang	Anmerkung
	"Data_fromClient".	
2.	Fügen Sie ein Array mit dem Namen "buffer" und dem Offset "DBB_nr_s" und der Länge "length"(aus Tabelle 6-1 unter Nr. 4) in den Datenbaustein ein.	
3.	Öffnen Sie den OB Main (OB1).	
4.	Passen Sie den Aufruf des FB Sync_Server (FB1) an, indem Sie die Nummer des unter 1. erstellten DBs am Eingang FromClient_DB angeben.	
5.	Laden Sie das Anwenderprogramm in die CPU und starten Sie die CPU. Mit dem nächsten Synchronisations-Request auf Seite der Client-Station werden die neu definierten Datenbereiche synchronisiert	

Vorgehen im Client bei lesender Synchronisation

Tabelle 6-3

Nr.	Vorgang	Anmerkung
1.	Öffnen Sie den DB Client2Serv_orders.	
2.	Öffnen Sie den Auftrag, den Sie abändern möchten.	
3.	Passen Sie die Nummer des Client-DB und dessen Offset an. <ul style="list-style-type: none"> DB_nr_cl gibt den DB an, in den die 160 Byte geschrieben werden. DBB_nr_cl gibt den Byte-Offset an, ab dem die Daten geschrieben werden. 	

6 Änderungen am Projekt

Nr.	Vorgang	Anmerkung																					
4.	<p>Passen Sie die Nummer des Server-DB und dessen Offset an.</p> <ul style="list-style-type: none"> DB_nr_s gibt den DB an, aus dem die Daten gelesen werden. DBB_nr_cl gibt den Byte-Offset an, ab dem die Daten gelesen werden. 	<table border="1"> <tr><td>ID</td><td>CONN_PRG</td><td>256</td></tr> <tr><td>DB_nr_cl</td><td>Int</td><td>227</td></tr> <tr><td>DBB_nr_cl</td><td>Int</td><td>0</td></tr> <tr><td>DB_nr_s</td><td>Int</td><td>7</td></tr> <tr><td>DBB_nr_s</td><td>Int</td><td>0</td></tr> <tr><td>length</td><td>Int</td><td>160</td></tr> <tr><td>Status</td><td>DWord</td><td>16#0</td></tr> </table>	ID	CONN_PRG	256	DB_nr_cl	Int	227	DBB_nr_cl	Int	0	DB_nr_s	Int	7	DBB_nr_s	Int	0	length	Int	160	Status	DWord	16#0
ID	CONN_PRG	256																					
DB_nr_cl	Int	227																					
DBB_nr_cl	Int	0																					
DB_nr_s	Int	7																					
DBB_nr_s	Int	0																					
length	Int	160																					
Status	DWord	16#0																					
5.	<p>Sie können auch die Länge des zu synchronisierenden Datenbereiches in Byte angeben.</p> <p>Beachten Sie, dass maximal 160 Byte synchronisiert werden können.</p>	<table border="1"> <tr><td>ID</td><td>CONN_PRG</td><td>256</td></tr> <tr><td>DB_nr_cl</td><td>Int</td><td>227</td></tr> <tr><td>DBB_nr_cl</td><td>Int</td><td>0</td></tr> <tr><td>DB_nr_s</td><td>Int</td><td>7</td></tr> <tr><td>DBB_nr_s</td><td>Int</td><td>0</td></tr> <tr><td>length</td><td>Int</td><td>160</td></tr> <tr><td>Status</td><td>DWord</td><td>16#0</td></tr> </table>	ID	CONN_PRG	256	DB_nr_cl	Int	227	DBB_nr_cl	Int	0	DB_nr_s	Int	7	DBB_nr_s	Int	0	length	Int	160	Status	DWord	16#0
ID	CONN_PRG	256																					
DB_nr_cl	Int	227																					
DBB_nr_cl	Int	0																					
DB_nr_s	Int	7																					
DBB_nr_s	Int	0																					
length	Int	160																					
Status	DWord	16#0																					
6.	Stellen Sie sicher, dass der unter 3. angegebene DB vorhanden ist.																						
7.	<p>Laden Sie den DB in die CPU und starten Sie die CPU neu.</p> <p>Mit dem nächsten Synchronisations-Request werden die neu definierten Datenbereiche synchronisiert.</p>																						

Vorgehen im Server bei lesender Synchronisation

Tabelle 6-4

Nr.	Vorgang	Anmerkung
1.	Erstellen Sie einen DB mit der in Tabelle 6-3 unter Nr. 4 angegebenen Datenbausteinnummer und dem symbolischen Namen "Data_toClient".	
2.	Fügen Sie ein Array mit dem Namen "buffer" und dem Offset "DBB_nr_s" und der Länge "length" (aus Tabelle 6-3 unter Nr. 4) in den Datenbaustein ein.	
3.	Laden Sie den DB in die CPU und starten Sie die CPU neu.	

6.2 Hinzufügen eines weiteren Servers

Übersicht

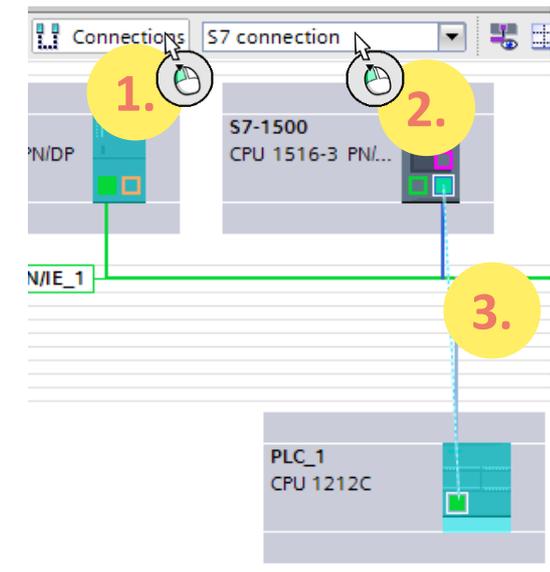
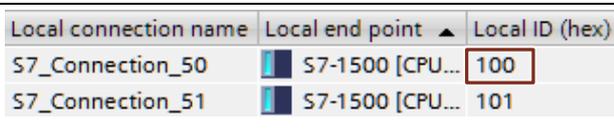
Für das Hinzufügen eines weiteren Servers in das Projekt müssen Sie die folgenden Schritte realisieren:

- Hinzufügen einer neuen CPU in das Projekt.
- Projektieren einer S7-Verbindung.
- Anpassen des Anwenderprogramms des Clients (der Kopfstation).
- Erstellen des Anwenderprogramms des neu eingefügten Servers (der Substation).

Folgen Sie dazu den Anweisungen der folgenden Tabelle.

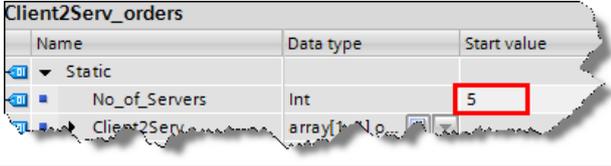
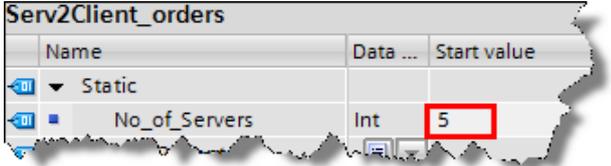
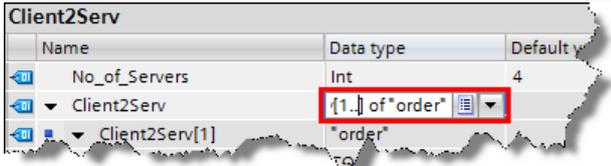
Vorgehen auf Seite des Servers

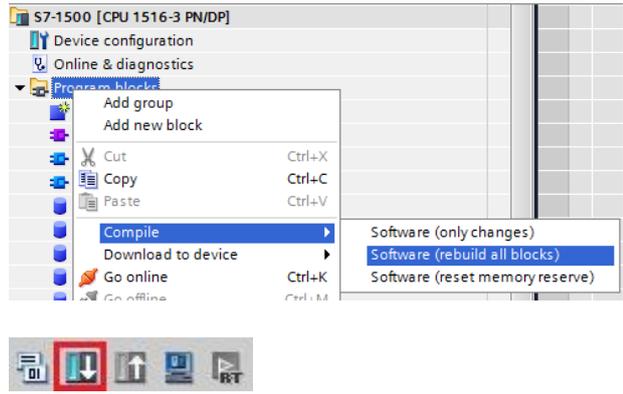
Tabelle 6-5

Nr.	Vorgang	Anmerkung									
1.	Fügen Sie in ihr Projekt über "neues Gerät hinzufügen" ("Add new device") eine weitere S7-1200 ein.										
2.	Wechseln Sie in die Netzansicht ("Network View"). Fügen Sie eine S7-Verbindung zwischen der neu eingefügten CPU und der bereits vorhandenen S7-300 oder S7-1500 ein.										
3.	Merken Sie sich unter "Verbindungen" ("Connections") die lokale ID in der S7-300/S7-1500 der neu erstellten Verbindung.	 <table border="1"> <thead> <tr> <th>Local connection name</th> <th>Local end point</th> <th>Local ID (hex)</th> </tr> </thead> <tbody> <tr> <td>S7_Connection_50</td> <td>S7-1500 [CPU...</td> <td>100</td> </tr> <tr> <td>S7_Connection_51</td> <td>S7-1500 [CPU...</td> <td>101</td> </tr> </tbody> </table>	Local connection name	Local end point	Local ID (hex)	S7_Connection_50	S7-1500 [CPU...	100	S7_Connection_51	S7-1500 [CPU...	101
Local connection name	Local end point	Local ID (hex)									
S7_Connection_50	S7-1500 [CPU...	100									
S7_Connection_51	S7-1500 [CPU...	101									
4.	Kopieren Sie die Bausteine eines anderen Servers in den Programmbaustein-Ordner der neu eingefügten S7-1200.										

6 Änderungen am Projekt

Vorgehen auf Seite des Clients

Nr.	Vorgang	Anmerkung
1.	Wechseln Sie zum Ordner "Programmbausteine" ("program blocks") Ihres Clients und öffnen Sie den DB Client2Serv_orders (DB114)	
2.	Erhöhen Sie den Parameter No_of_Servers um 1.	
3.	Öffnen Sie jetzt den DB Serv2Client_orders (DB224)	
4.	Erhöhen Sie den Parameter No_of_Servers um 1.	
5.	Öffnen Sie im Ordner "PLC-Datentypen" ("PLC data types") den Datentyp "Client2Serv"	
6.	Vergrößern Sie das Array PUT_orders um 1.	
7.	Geben Sie für die Startwerte der Parameter folgende Daten ein: <ul style="list-style-type: none"> ID aus Tabelle 6-5 Nr. 3. DB_nr_cl = 116 DBB_nr_cl :=0 DB_nr_s = 6 DBB_nr_s =0 	
8.	Fügen Sie in Ihr Projekt einen Datenbaustein mit einem Array der Größe 160 Byte ein.	
9.	Wiederholen Sie die Schritte 9.-11. für den Datentyp "Serv2Client" mit den folgenden Daten: <ul style="list-style-type: none"> ID aus Tabelle 6-5 Nr. 3. DB_nr_cl = Nummer des DBs aus Nr. 10 DBB_nr_cl :=0 DB_nr_s = 7 DBB_nr_s =0 	

Nr.	Vorgang	Anmerkung
10.	Übersetzen und laden Sie die Konfigurationen und das Anwenderprogramm in die CPUs. Jetzt synchronisiert der Client Daten mit dem eingefügten Server. Zur Anpassung des zu synchronisierenden Datenbereiches beachten Sie bitte Kapitel 6.1	

6.3 Sicherstellen der Datenkonsistenz im Server

Datenkonsistenz

Dateninkonsistenz kann auftreten, wenn zwei Prozesse auf denselben Datenbereich gleichzeitig zugreifen.

Konkret kann in diesem Beispiel vorkommen, dass der Server einen Datenbereich in dem Moment auswerten will, indem der Client auf diesen Datenbereich schreibend zugreift.

Dieses Kapitel beschreibt, wie die Datenkonsistenz auf Seiten des Servers sichergestellt wird.

Schreibende Synchronisation

Synchronisiert der Client schreibend einen Datenbereich mit einem Server, so überprüft er vor dem Senden neuer Daten das erste Byte des Puffers.

Hat dieses Byte den Wert 16#00, so können neue Daten in den Server geschrieben werden.

Folgendes Vorgehen im Anwenderprogramm des Servers ist empfohlen und in Teilen schon implementiert:

Tabelle 6-6

Nr.	Vorgehen
1.	Fragen Sie regelmäßig das Koordinations-Byte im DB "Data_FromClient" auf den Status 16#F0 ab (vor Aufruf des FB "Sync_Server").
2.	Sind neue Daten angekommen, so retten Sie diese Daten in einen Puffer oder verarbeiten Sie direkt (auch vor Aufruf des FB "Sync_Server").
3.	Der FB "Sync_Server" setzt das Koordinations-Byte auf 16#00 zurück, damit der Client wieder Daten schicken kann.

Lesende Synchronisation

Synchronisiert der Client einen Datenbereich lesend mit dem Server, so überprüft er vor dem Auslesen der Daten das erste Byte des Puffers.

Hat dieses Byte den Wert 16#F0, so liegen neue Daten im Server an.

Folgendes Vorgehen im Anwenderprogramm des Servers ist empfohlen und in Teilen schon implementiert:

Tabelle 6-7

Nr.	Vorgehen
1.	Schreiben Sie in den DB "Data_ToClient" Ihre Nutzdaten.
2.	Der FB "Sync_Server" setzt Sie das Koordinations-Byte auf den Wert 16#F0 und vom Client wird es nach Empfang der Daten wieder auf 16#00 gesetzt (permanenter Transfer).
3.	Fragen Sie regelmäßig das Koordinations-Byte im DB "Data_ToClient" auf den Status 16#00 ab (vor Aufruf des FB "Sync_Server"). Hat das Byte diesen Wert, so hat der Client die Daten abgeholt und es kann wieder mit Nr. 1. begonnen werden.

6.4 Erweitern der Funktion des Koordinations-Bytes

Überblick

Um die Kommunikation zwischen Client und Server zu koordinieren wird das Koordinations-Byte verwendet.

Wenn Sie in Ihrem Programm weitere Koordinationsfunktionen einfügen wollen, dann beschreibt dieses Kapitel, wo im Programm Sie Anpassungen vornehmen müssen.

Werte des Koordinations-Bytes

Das Koordinations-Byte kann in der Beispielapplikation 3 Zustände annehmen. [Tabelle 6-8](#) beschreibt die Werte, die das Koordinations-Byte annehmen kann und deren Bedeutung aus Client-Sicht.

Tabelle 6-8

Wert \ Richtung	16#00	16#F0	16#FF
Client2Serv	Daten können geschrieben werden.	Daten wurden noch nicht verarbeitet.	Zeit wurde noch nicht im Server synchronisiert
Serv2Client	Es liegen keine neuen Daten an.	Neue Daten → Daten können gelesen werden.	---

Erweitern der Werte im Client

Um auf spezielle andere Werte als die in [Tabelle 6-8](#) zu reagieren, müssen Sie in der Auswertung weitere Optionen hinzufügen. Konkret bedeutet das, dass Sie die bereits vorhandene IF-Anweisung um weitere ELSIF Optionen erweitern müssen.

Für den FB Sync_Client2Serv an folgender Stelle des Codes:

6 Änderungen am Projekt

Abbildung 6-1

```
2: //step2: get coordination byte from the server
// depending on the coordination byte: send data or go to next server in order list
#control.g_req:= 1; //start GET
IF #view.G_NDR THEN //data arrived
#control.g_req:= 0;
IF "Client2Serv_Buffer".coord = 16#00 THEN //server is ready to receive data
#control.step := 3; //goto step3
ELSIF "Client2Serv_Buffer".coord = 16#F0 THEN //server is not ready to get data,
//data not yet handled
#Orders.Client2Serv[#control.number].Status := 16#000000F0; //save status
#view.no_new_data := #view.no_new_data +1;//data not accepted from server
#control.step:=8; //go to step8: next server in order list
ELSIF "Client2Serv_Buffer".coord = 16#FF THEN //server is not ready to get data,
//time sync is not yet finished
#Orders.Client2Serv[#control.number].Status := 16#000000FF; //save status
#view.no_new_data := #view.no_new_data +1;//data not accepted from server
#control.step:=8; //go to step8: next server in order list
ELSIF
END_IF;
```

add code

Für den FB Sync_Serv2Client an folgender Stelle des Codes:

Abbildung 6-2

```
2: //step2: analyse coordination byte
IF "Serv2Client_Buffer".coord = 16#F0 THEN //new data in the server available
#control.step := 3; //goto step3
ELSIF "Serv2Client_Buffer".coord = 16#00 THEN //no new data in the server available
#Orders.Serv2Client[#control.number].Status := 16#00010028; //save status
#view.no_new_data := #view.no_new_data +1;//no new data in server available
#control.step := 6; //goto step6: change server
ELSIF
END_IF;
```

add code

7 Anhang

7.1 Service und Support

Industry Online Support

Sie haben Fragen oder brauchen Unterstützung?

Über den Industry Online Support greifen Sie rund um die Uhr auf das gesamte Service und Support Know-how sowie auf unsere Dienstleistungen zu.

Der Industry Online Support ist die zentrale Adresse für Informationen zu unseren Produkten, Lösungen und Services.

Produktinformationen, Handbücher, Downloads, FAQs und Anwendungsbeispiele – alle Informationen sind mit wenigen Mausklicks erreichbar:

support.industry.siemens.com

Technical Support

Der Technical Support von Siemens Industry unterstützt Sie schnell und kompetent bei allen technischen Anfragen mit einer Vielzahl maßgeschneiderter Angebote – von der Basisunterstützung bis hin zu individuellen Supportverträgen.

Anfragen an den Technical Support stellen Sie per Web-Formular:

www.siemens.de/industry/supportrequest

SITRAIN – Training for Industry

Mit unseren weltweit verfügbaren Trainings für unsere Produkte und Lösungen unterstützen wir Sie praxisnah, mit innovativen Lernmethoden und mit einem kundenspezifisch abgestimmten Konzept.

Mehr zu den angebotenen Trainings und Kursen sowie deren Standorte und Termine erfahren Sie unter:

www.siemens.de/sitrain

Serviceangebot

Unser Serviceangebot umfasst folgendes:

- Plant Data Services
- Ersatzteilservices
- Reparaturservices
- Vor-Ort und Instandhaltungsservices
- Retrofit- und Modernisierungsservices
- Serviceprogramme und Verträge

Ausführliche Informationen zu unserem Serviceangebot finden Sie im Servicekatalog:

support.industry.siemens.com/cs/sc

Industry Online Support App

Mit der App "Siemens Industry Online Support" erhalten Sie auch unterwegs die optimale Unterstützung. Die App ist für Apple iOS, Android und Windows Phone verfügbar:

support.industry.siemens.com/cs/ww/de/sc/2067

7.2 Links und Literatur

Tabelle 7-1

Nr.	Thema
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Link auf die Beitragsseite des Anwendungsbeispiels https://support.industry.siemens.com/cs/ww/de/view/40556214
\3\	Handbuch "SIMATIC STEP 7 Basic/Professional V15.1 und SIMATIC WinCC V15.1" https://support.industry.siemens.com/cs/ww/de/view/109755202
\4\	Handbuch "CPU-CPU Kommunikation mit SIMATIC Controllern" https://support.industry.siemens.com/cs/ww/de/view/78028908
\5\	Handbuch "Kommunikation mit SIMATIC" https://support.industry.siemens.com/cs/ww/de/view/25074283
\6\	Handbuch "SIMATIC System- und Standardfunktionen für S7-300/400 - Band 1/2" https://support.industry.siemens.com/cs/ww/de/view/109751826
\7\	Handbuch "SIMATIC Programmieren mit STEP 7 V5.5" https://support.industry.siemens.com/cs/ww/de/view/45531107
\8\	Technische Daten "CPU315-2 PN/DP, 384 KB" https://support.industry.siemens.com/cs/ww/de/pv/6ES7315-2EH14-0AB0/td?dl=de
\9\	Technische Daten "CPU 1516-3 PN/DP, 1MB Prog., 5MB Daten" https://support.industry.siemens.com/cs/ww/de/pv/6ES7516-3AN01-0AB0/td?dl=de
\10\	Anwendungsbeispiel "Datensynchronisation über offene Kommunikation zwischen mehreren Feldsteuerungen (S7-1200) und einer Zentralstation (S7-1200/ S7-300/ WinAC)" https://support.industry.siemens.com/cs/ww/de/view/39040038

7.3 Änderungsdokumentation

Tabelle 7-2

Version	Datum	Änderung
V1.0	01/2009	S7-Kommunikation über die integrierte S7-300 CPU Schnittstelle (Aufgabenstellung A) und über einen S7-300 CP (Aufgabenstellung B)
V1.1	02/2010	Ergänzung im Kapitel 2.3: S7-1200 Datenübertragung
V1.2	08/2010	Änderung der Aufgabenstellung in deterministischen Datenaustausch über S7-Kommunikation (Aufgabenstellung A)
V2.0	06/2013	Streichung von Aufgabenstellung B und flexiblere Gestaltung der Aufgabenstellung A: Parametrierbare Synchronisations-Bereiche
V2.1	08/2019	Fehlerbehebung + Aktualisierung auf TIA Portal V15.1