



S7-SoftSPS PLC-416
PLC-315

Benutzerhandbuch

Version 4.x

Die in diesem Handbuch enthaltene Information kann ohne Vorankündigung geändert werden und stellt keine Verpflichtung seitens der *IBHsoftec* GmbH dar. Die Software, die in diesem Handbuch beschrieben ist, wird unter einer Lizenzvereinbarung und einer Geheimhaltungsvereinbarung zur Verfügung gestellt. Die Software darf nur nach Maßgabe der Bedingungen der Vereinbarung benutzt und kopiert werden. Es ist rechtswidrig, die Software auf andere Medien zu kopieren, soweit das nicht ausdrücklich in der Lizenzvereinbarung und Geheimhaltungsvereinbarung erlaubt wird. Der Käufer darf eine Kopie zu Sicherungszwecken machen. Ohne ausdrückliche schriftliche Erlaubnis der *IBHsoftec* GmbH darf der Inhalt des Handbuches für andere Zwecke als dem persönlichen Gebrauch durch den Käufer dieses Handbuchs nicht in irgendeiner Form mit irgendwelchen Mitteln, elektronisch oder mechanisch, mittels Fotokopie, durch Aufzeichnung oder mit Informationsspeicherungsgeräten reproduziert und übertragen werden.

© Copyright 1993 - 2007 *IBHsoftec* GmbH. Alle Rechte vorbehalten.

IBHsoftec

Gesellschaft für Automatisierungstechnik mbH

Turmstraße 77

D - 64743 Beerfelden / Odw.

Germany

Telefon 0 60 68 / 30 01

Telefax 0 60 68 / 30 74

Internet: www.IBHsoftec.de

E-Mail: info@IBHsoftec.de

Simatic, Step5, Step7, MicroWin, Graph5, S7-200, S7-300, S7-400 und MPI sind eingetragene Warenzeichen der Siemens Aktiengesellschaft, Berlin und München.

Windows, ActiveX sind eingetragene Marken der Microsoft Corporation.

Excel ist ein eingetragenes Warenzeichen der Microsoft Corporation in den Vereinigten Staaten und anderen Ländern.

Inhaltsverzeichnis

1	IBHsoftec S7-SoftSPS	5
2	Installation	6
2.1	Systemanforderungen.....	6
2.2	Installation der S7-SoftSPS	6
2.3	Deinstallation der S7-SoftSPS.....	6
3	Online	7
3.1	Die S7-SoftSPS Online - Funktionen	7
3.2	Online mit STEP®7	8
3.3	Online mit S7 für Windows®.....	9
3.4	Visualisierungen.....	10
3.5	Datenschnittstelle zur S7-SoftSPS	11
4	Oberfläche & Konfiguration	20
4.1	Klassische Ansicht	20
4.2	Konfigurationsansicht.....	24
4.3	Menü	24
4.3.1	Menü Datei	25
4.3.2	Menü Bearbeiten	26
4.3.3	Menü Ansicht	27
4.3.4	Menü Hilfe	27
4.4	Werkzeugleiste.....	28
4.5	I/O Monitor	29
5	Konfiguration.....	31
5.1	CPU Parameter.....	31
5.1.1	Konfiguration mit Hilfe der Oberfläche	31
5.1.2	Konfiguration in der PLC43.INI	34
5.2	Treiber hinzufügen/entfernen.....	35
5.2.1	Treiber hinzufügen/entfernen mit Hilfe der Oberfläche	35
5.2.2	Treiber hinzufügen/entfernen in der PLC43.INI	36
5.3	Addidata	37
5.3.1	ADDI-DATA Konfiguration.....	37
5.3.2	Einstellen des ADDI-DATA-Treibers	42
5.3.3	ADDI-DATA Treiber für IBHsoftec S7-SoftSPS	43
5.4	Beckhoff TwinCAT I/O	44
5.4.1	Beckhoff TwinCAT I/O Konfiguration	44
5.4.2	TwinCAT I/O Treiber konfigurieren	53
5.4.3	TwinCAT I/O mit PLC43.INI konfigurieren	56
5.5	CIF 30/50/60 Konfiguration	57
5.5.1	Grundlegende Konfiguration	57
5.5.2	Konfiguration der CIF Karte mit dem Systemkonfigurator	70
5.5.3	CIF 30/50/60 Einstellungen.....	71
5.5.4	CIF 30/50/60 Einstellungen der IBHsoftec S7-SoftSPS	74
5.5.5	Profibus Konfiguration via S7 für Windows®/Step® 7	75
5.5.6	Profibus Konfiguration über S7 für Windows®/ Step® 7	75
5.5.7	Profibus Konfiguration über S7 für Windows®/Step® 7	76
5.6	CP343/CP443 Emulation	77
5.6.1	CP343/CP443 Emulation konfigurieren	77
5.6.2	CP343/CP443 Emulation über die Oberfläche konfigurieren	79
5.6.3	CP343/CP443 Emulation in PLC43.INI konfigurieren.....	81
5.7	I/O Port Link	82
5.7.1	PC Hardwareports konfigurieren.....	82
5.7.2	PC Hardwareports mit Oberfläche einstellen.....	82
5.7.3	PC Hardwareports mit PLC43.INI einstellen.....	83

5.8	Modbus on TCP	86
5.8.1	Modbus on TCP Konfiguration	86
5.8.2	Modbus on TCP konfigurieren mit der Oberfläche	88
5.8.3	Modbus on TCP konfigurieren mit PLC43.INI	89
5.9	NVRAM Konfiguration	91
5.9.1	NVRAM Konfiguration	91
5.9.2	NVRAM Konfiguration mit der Oberfläche	93
5.9.3	NVRAM Konfiguration mit der PLC43.INI	94
5.10	Softnet PN IO	95
5.10.1	Softnet PN IO Konfiguration	95
5.10.2	Konfigurieren von Softnet mit der Oberfläche	98
5.10.3	Konfigurieren von Softnet mit PLC43.INI	99
5.11	User SFC	100
5.11.1	User SFC Konfiguration	100
5.11.2	Einstellen der UserSFCs/FCs in der Oberfläche	100
5.11.3	Einstellen der UserSFCs/FCs in der PLC43.INI	102
6	Technische Daten	103
6.1	Technische Daten	103
6.2	Operationsvorrat	104
6.3	Diagnosepuffereinträge	112
6.4	Unterschiede S7-SoftSPS - Hardware-SPS	115
6.5	AppStarter	117
6.6	Null-Modemkabel	119

1 IBHsoftec S7-SoftSPS

Die IBHsoftec S7-SoftSPS ist eine Software zur Nachbildung einer Hardware-SPS mit Echtzeit-Verhalten. Das SPS Programm wird nach dem Laden, wie bei einer Hardware-SPS abgearbeitet. Dies bietet den Vorteil, dass der Programm-Status direkt angezeigt werden kann. Das Austesten eines SPS Programms wird dadurch sehr erleichtert, da jederzeit festgestellt werden kann, welcher Befehl gerade abgearbeitet wird.

Die S7-SoftSPS 315/416 ist kompatibel zu einer SIEMENS SIMATIC 315/416 Hardware SPS.

Es sind folgende Zusätze zu einer Standard SPS implementiert:

- Emulation eines Ethernet-CP 343/443
- Modbus on TCP Server/Client
- NVRAM Unterstützung für remanente Daten
- Schnittstelle für Benutzerprogrammierte FC/SFC

Treiber für :

Hilscher CIF-Karten	AS-Interface-Master, CANopen-Slave/Master, DeviceNet-Slave/Master, InterBus-Slave/Master, PROFIBUS-DP-Slave-DP/FMS-Master, SERCOS
Beckhoff TwinCAT I/O	EtherCAT®, Lightbus, PROFIBUS DP/MC, Interbus, CANopen, SERCOS interface, DeviceNet, Ethernet, PC-Printer-Port, USB, SMB
SoftNet PN IO	PROFINET
Addi-Data Treiber	ADDIPACK
PC-Hardwareports	ISA

2 Installation

Das Kapitel Installation gliedert sich in folgende Unterkapitel:

- Systemanforderungen
- Installation
- Deinstallation

2.1 Systemanforderungen

Die *S7-SoftSPS* ist lauffähig unter den Betriebssystemen Windows® NT Version 4.0, Windows® 2000, Windows® XP und Windows® Vista von Microsoft. Auf Anfrage sind spezielle OEM Versionen für Windows® CE und Windows® XP embedded verfügbar.

2.2 Installation der *S7-SoftSPS*

Das Installationsprogramm führt für Sie alle notwendigen Schritte zum Installieren der *S7-SoftSPS* auf der Festplatte durch. Folgen Sie den Anweisungen des Installationsprogramms.

Um die *S7-SoftSPS* zu installieren, muss die Software mit Ihrer persönlichen Geheimzahl freigeschaltet werden. Ihre persönliche Geheimzahl und die Seriennummer der *S7-SoftSPS* finden Sie auf der Freischaltkarte, die zusammen mit der Softwareregistrierkarte (Registration Card) der Installations-CD beigelegt ist.

2.3 Deinstallation der *S7-SoftSPS*

Um die *S7-SoftSPS* von Ihrem Rechner zu deinstallieren, führen Sie bitte die folgenden Schritte durch:

- In der Windows® Systemsteuerung starten Sie das Programm Software
- Im Menü „Programme ändern oder entfernen“ wählen Sie aus der angezeigten Liste der installierten Programme die *S7-SoftSPS* aus. Betätigen Sie nun die Schaltfläche „Ändern/Entfernen“.

3 Online

In diesem Kapitel werden die verschiedenen Möglichkeiten beschrieben auf die S7-SoftSPS online zu gelangen.

- Allgemein
- Step® 7
- *S7 für Windows®*
- Visualisierungen
- Datenschnittstelle

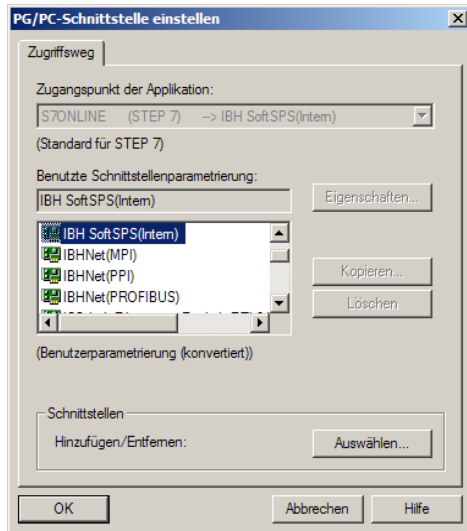
3.1 Die S7-SoftSPS Online - Funktionen

Der Datenaustausch mit der *S7-SoftSPS* kann aus *S7 für Windows®* oder dem Siemens SIMATIC® Manager erfolgen. Ist das Programmiersystem auf dem gleichen Rechner wie die *S7-SoftSPS* installiert, so kann der Datenaustausch mit der *S7-SoftSPS* direkt über den internen PC Speicher erfolgen. Die hierzu erforderlichen Treiber werden mit der *S7-SoftSPS* installiert. Alternativ kann der Datenaustausch über eine serielle Schnittstelle erfolgen. Es kann somit über ein Nullmodemkabel ein Programmiergerät angeschlossen werden. Hierbei ist ein PC Adapter mit 38,4 kBaud am Programmiergerät zu wählen. Weiterhin ist ein Zugriff über ein TCP/IP Ethernet mit Hilfe des *IBHNet* Treibers möglich. Der *IBHNet* Treiber ist unter anderem auf der *IBHsoftec* Homepage zum freien Download bereitgestellt.

Onlinefunktionen sind ebenfalls über das Siemens ISO-On-TCP (RFC1006) Protokoll möglich. Hierzu muss der Ethernet CP-343/443 in der Konfigurationsansicht eingebunden sein.

Wird eine Hilscher CIF30/50/60 Profibuskarte verwendet, so kann die *S7-SoftSPS* auch über den angeschlossenen Profibus programmiert oder von Operator Panels erreicht werden, sofern die Kartentreiber korrekt konfiguriert sind.

3.2 Online mit STEP®7



Mit STEP®7 direkt auf die S7-SoftSPS zugreifen

Über Step®7 ist ein Direktzugriff möglich. Die S7-SoftSPS installiert hierzu einen eigenen Zugangspunkt im Dialog "PG/PC Schnittstelle einstellen".

Mit STEP®7 über IBHNet auf die S7-SoftSPS zugreifen

Ist der IBHNet Treiber auf dem PG installiert, so kann die S7-SoftSPS via Ethernet programmiert werden. Das IBHNet Protokoll ist auf der S7-SoftSPS immer aktiviert, so dass hier keine Konfiguration erforderlich ist. In der IBHNet Station, die auf dem Programmiergerät konfiguriert wird, ist lediglich die IP Adresse des Rechners, der die S7-SoftSPS ausführt einzustellen.

Mit STEP®7 über TCP/IP auf die S7-SoftSPS zugreifen

Der Treiber cpx43.dll muss in der Konfigurationsdatei PLC43.INI eingebunden sein. Im Simatic® Manager muss in der Hardwarekonfiguration ein CP zur Hardwarekonfiguration der SPS hinzugefügt werden, auf dem die IP Adresse des Rechners mit der S7-SoftSPS eingestellt wird.

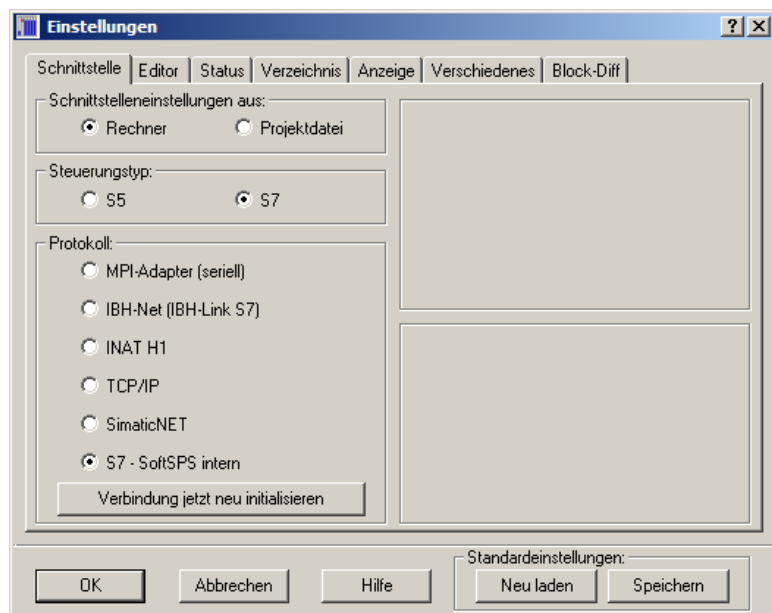
Mit STEP®7 über die serielle Schnittstelle auf die S7-SoftSPS zugreifen

Die MPI®-Schnittstelle muss auf einen COM-Port eingestellt sein. Somit kann man über ein Nullmodemkabel (siehe Technische Unterlagen) auf die SPS zugreifen. Die PG/PC-Schnittstelle muss nun nur noch auf einen PC-Adapter mit 38.400 Baud eingestellt werden.

Mit STEP®7 über Profibus auf die S7-SoftSPS zugreifen

Voraussetzung ist, wie bei einer Hardware SPS, dass der Profibus zuerst über MPI® konfiguriert wurde. Im Falle der S7-SoftSPS kann dies direkt, seriell, oder via TCP/IP geschehen. Ist der Profibus funktionsfähig, so kann in der Treiberkonfiguration die Option „OnlineDP“ eingeschaltet werden, um einen Zugriff auf die SPS via Profibus zu erlauben.

3.3 Online mit *S7 für Windows*®



Mit *S7 für Windows*® direkt auf die *S7-SoftSPS* zugreifen

S7 für Windows® bietet Ihnen die Möglichkeit, direkt auf die *S7-SoftSPS* zuzugreifen, um Programme zu übertragen. Alle *S7 für Windows*® Online-Funktionen stehen bei dem direkten Zugriff auf die *S7-SoftSPS* zur Verfügung. Der Datenaustausch erfolgt extrem schnell.

Falls Sie eine ältere *S7 für Windows*® Version (z.B. < V 4.04) besitzen, bestellen Sie bitte ein Update.

Mit *S7 für Windows*® über *IBHNet* auf die *S7-SoftSPS* zugreifen

S7 für Windows® via *IBHNet* auf die *S7-SoftSPS* verbinden.

Im Dialog *Einstellungen* des *S7 für Windows*® wählt man das Protokoll *IBH-Net (IBH-Link S7)* aus. Nach einem Klick auf „Verbindung auswählen“ sucht man sich im *IBHNet* den gewünschten Verbindungsteilnehmer aus. Mit einem Klick auf „SPS auswählen“ erscheint ein „*S7 CPU-Auswahl*“ Dialog in welchem die Vorhandenen *MPI*®-Adressen angezeigt werden. Durch Bestätigung des Dialoges ist die *SPS* ausgewählt. Nun baut man noch die Verbindung zur *S7-SoftSPS* mit einem Klick auf „Verbindung jetzt neu initialisieren“ auf, und beendet den Einstellungsdialog mit Hilfe eines Klicks auf „OK“. Die Verbindung zur *S7-SoftSPS* ist nun hergestellt.

Mit *S7 für Windows*® über *TCP/IP* auf die *S7-SoftSPS* zugreifen

S7 für Windows® via *RFC1006* auf die *S7-SoftSPS* verbinden.

Im Dialog *Einstellungen* des *S7 für Windows*® wählt man das Protokoll *TCP/IP* aus. Der Dialog „Wahl der Station im Netz über *TCP/IP*“ der nach einem Klick auf „Verbindung auswählen“ erscheint, ermöglicht es die Verbindungsadresse der *S7-SoftSPS* einzutragen und mittels Bestätigung des Dialoges auf „OK“ auszuwählen. Nun baut man noch die Verbindung zur *S7-SoftSPS* mit einem Klick auf „Verbindung jetzt neu initialisieren“ auf, und beendet den Einstellungsdialog mithilfe eines Klicks auf „OK“.

Die Verbindung zur *S7-SoftSPS* ist nun hergestellt.

Mit S7 für Windows® über die serielle Schnittstelle auf die S7-SoftSPS zugreifen

S7 für Windows® via serieller Schnittstelle auf die S7-SoftSPS verbinden.

Im Dialog Einstellungen des S7 für Windows® wählt man das Protokoll MPI®-Adapter (seriell) aus. Im Bereich „Serielle Schnittstelle:“ wählt man sich seinen COM-Port mit dem man sich auf den Rechner der S7-SoftSPS verbunden hat und eine Baudrate von 38400 aus. Mit einem Klick auf „SPS auswählen“ erscheint ein „S7 CPU-Auswahl“ Dialog in welchem die Vorhandenen MPI®-Adressen angezeigt werden. Durch Bestätigung des Dialoges ist die SPS ausgewählt. Nun baut man noch die Verbindung zur S7-SoftSPS mit einem Klick auf „Verbindung jetzt neu initialisieren“ auf, und beendet den Einstellungsdialog mithilfe eines Klicks auf „OK“.

Die Verbindung zur S7-SoftSPS ist nun hergestellt.

3.4 Visualisierungen

Direkte Treiber

Die folgende Liste zeigt eine Auswahl von Visualisierungsherstellern, die direkte Treiber zur IBHsoftec S7 SoftSPS zur Verfügung stellen.

Hersteller	Programm	S7-SoftSPS	IBHNet	RFC 1006	Fetch/Write	Modbus on TCP
Copa-Data GmbH	zenOn	Ja	Nein	Ja	Ja	Ja
EXOR		Nein	Nein	Ja	Ja	Ja
GEFASOFT AG München	GraphPic®	Ja	Ja	Ja	Ja	Ja
GTI-control Gesellschaft für technische Informationsverarbeitung mbH	PROCON-WIN	Nein	Ja	Ja	Nein	Ja
	ISIVIS	Nein	Ja	Ja	Nein	Ja
INOSOFT AG	VisiWinNET®	Ja	Ja	Optional*	Optional*	Optional*
Lenze Digitec Controls GmbH	VisiWinNET® (INOSOFT)	Ja	Ja	Optional*	Optional*	Optional*
Pro-face		Nein	Nein	Ja	Ja	Ja
Siemens	ProTool® / Pro	Ja	Ja	Ja	Ja	Nein
	WinCC® / flexible	Ja	Ja	Ja	Ja	Ja
Sütron	TSwin.net	Nein	Nein	Ja	Nein	Ja
	zenOn (Copa-Data)	Ja	Nein	Ja	Ja	Ja
Wonderware	InTouch	Nein	Nein	Ja	Nein	Ja

* Auf Anfrage bei INOSOFT

IBHNet

Die *S7-SoftSPS* kann mit jeder Visualisierungssoftware, die das *IBHNet* Protokoll unterstützt, mit Hilfe des *IBHNet* Treibers angesprochen werden. Der *IBHNet* Treiber steht unter <http://www.ibhsoftec-sps.de> zum freien Download zur Verfügung.

Hat eine Visualisierung einen eigenen direkten Treiber für das *IBHLink S7* (NetLink) Protokoll selbst implementiert, so kann diese ebenfalls auf die *S7-SoftSPS* zugreifen, da auch dieses Protokoll (NL_MPI) von der *S7-SoftSPS* unterstützt wird.

Zusätzlich besteht die Möglichkeit, dass eigene Visualisierungen (beispielsweise Visual® BASIC oder andere Hochsprachen) über das vom *IBHNet*-Treiber mitgelieferte *IBHNet*-Objekt, eine Kommunikation zur *S7-SoftSPS* aufbauen. Details hierzu sind in der *IBHNet* Dokumentation enthalten.

RFC1006

Ein Visualisierungssystem über RFC1006 an die *S7-SoftSPS* anzubinden ist dann möglich, wenn die Ethernet CP-343/443 Emulation der SPS gestartet wurde.

Mit Operator Panels über Profibus auf die S7-SoftSPS zugreifen

Voraussetzung hierfür ist die Verwendung von Hilscher PROFIBUS® Master Karten. Wie bei einer Hardware SPS muss der Profibus zuerst über MPI® konfiguriert werden. Im Falle der *S7-SoftSPS* kann dies direkt, seriell, oder via TCP/IP geschehen. Ist der Profibus funktionsfähig, so kann in der Treiberkonfiguration die Option „OnlineDP“ eingeschaltet werden, um einen Zugriff auf die SPS via Profibus zu erlauben.

OPC Server

Eine Anbindung der *S7-SoftSPS* an ein Visualisierungssystem mittels des IBHsoftec OPC Servers und anderen OPC Servern, die die genannten Protokolle unterstützen, ist ebenfalls möglich.

3.5 Datenschnittstelle zur S7-SoftSPS

Von verschiedenen Programmiersystemen (z.B. Visual Basic®, Visual C/C++® oder Borland C++Builder®) kann direkt auf die *S7-SoftSPS* zugegriffen werden. Die hierzu benötigte DLL wird mitgeliefert (PLC32.DLL). Es handelt sich hierbei um eine 32 Bit Applikation.

Im Folgenden werden die zur Datenübertragung vorhandenen Funktionen beschrieben.

Funktionen zur Datenübertragung

Wert lesen

Liest einen Wert aus der S7-SoftSPS.

```
unsigned long int DLLexport ReadVal (unsigned long int typ,
                                     unsigned long int nr,
                                     unsigned long int DBNr,
                                     unsigned long int size,
                                     unsigned long int far * val);
```

Parameter	Beschreibung	Kommentar
typ	'E' = Eingang 'A' = Ausgang 'M' = Merker Z' = Zählerwert 'T' = Zeitwert 'D' = Datenwort oder Datendoppelwort 'B' = Datenbit,-byte,-wort,- doppelwort auf Byteadresse 'P' = P_Parameter	nur 16-Bit-Zugriff nur 16-Bit-Zugriff abhängig vom Parameter size abhängig von size S7 kompatibel, nur S7! nur S7 ab Version 3!
nr	Adresse	Byteadresse des einzulesenden Operanden
DBNr	Datenbaustein Nummer	1 – 16383
size	Anzahl der Bits 0..7 = Bit Nr 0 7 8 = 1 Byte 16 = 1 Wort 32 = 1 Doppelwort	nur 'E', 'A', 'M'
val	Gelesener Wert	Pointer auf eine gültige Variable im Adressraum Ihrer Applikation.
Rückgabe	0 bei Fehler	Fehlermeldungen

Länge der Parameterliste (Visual-C): 20 Bytes.

Wert schreiben

Schreibt einen Wert in die S7-SoftSPS.

unsigned long int DLLexport WriteVal (unsigned long int typ,

unsigned long int nr,

unsigned long int DBNr,

unsigned long int size,

unsigned long int far * val);

Parameter	Beschreibung	Kommentar
typ	'E' = Eingang 'A' = Ausgang 'M' = Merker 'Z' = Zählerwert 'T' = Zeitwert 'D' = Datenwort oder Datendoppelwort 'B' = Datenbit,-byte,-wort,-doppelwort auf Byteadresse 'P' = P_Parameter	nur 16-Bit-Zugriff nur 16-Bit-Zugriff abhängig vom Parameter size abhängig von size S7 kompatibel, nur S7! nur S7 ab Version 3!
nr	Adresse	Byteadresse des zu schreibenden Operanden
DBNr	Datenbaustein Nummer	1 – 16383
size	Anzahl der Bits 0..7 = Bit Nr 0 7 8 = 1 Byte 16 = 1 Wort 32 = 1 Doppelwort	nur 'E', 'A', 'M'
val	Zu schreibender Wert	
Rückgabe	0 bei Fehler	Fehlermeldungen

Länge der Parameterliste (Visual-C): 20 Bytes.

Mehrere Werte lesen

Liest mehrere Werte aus der *S7-SoftSPS*.

```
unsigned long int DLLexport ReadVals (unsigned long int typ,
                                     unsigned long int nr,
                                     unsigned long int DBNr,
                                     unsigned long int size,
                                     unsigned char far * vals);
```

Parameter	Beschreibung	Kommentar
typ	'E' = Eingang 'A' = Ausgang 'M' = Merker 'D' = Datenwort 'B' = Datenbyte auf Byteadresse	S5 kompatibel S7 kompatibel, nur S7!
nr	Adresse	
DBNr	Datenbaustein Nummer	1 – 16383
size	Anzahl der Bytes oder Datenworte	Bei typ: 'E', 'A', 'M' und 'B' Angabe in Bytes, bei 'D' Angabe in Datenworten.
vals	Byte-Werte	Pointer auf ein gültiges Array im Adressraum Ihrer Applikation. ACHTUNG: DAS FELD MUSS GROSS GENUG SEIN!
Rückgabe	0 bei Fehler	Fehlermeldungen

Länge der Parameterliste (Visual-C): 20 Bytes.

ACHTUNG: BEI DATENBAUSTEINEN (TYP = 'D') MUSS DAS WERTE-FELD DOPPELT SO GROSS SEIN DA FÜR JEDES DATENWORT 2 BYTES BENÖTIGT WERDEN!

Mehrere Werte schreiben

Schreibt mehrere Werte in die S7-SoftSPS.

unsigned long int DLLexport WriteVals (unsigned long int typ,

unsigned long int nr,

unsigned long int DBNr,

unsigned long int size,

unsigned char far * vals);

Parameter	Beschreibung	Kommentar
typ	'E' = Eingang 'A' = Ausgang 'M' = Merker 'D' = Datenwort 'B' = Datenbyte auf Byteadresse	S5 kompatibel S7 kompatibel, nur S7!
nr	Adresse	
DBNr	Datenbaustein Nummer	1 – 16383
size	Anzahl der Bytes oder Datenworte 'E' 'A' 'M' 'B' 'D'	Angabe in Bytes Angabe in Bytes Angabe in Bytes Angabe in Bytes Angabe in Datenworten
vals	Byte-Werte	Pointer auf ein gültiges Array im Adressraum Ihrer Applikation. ACHTUNG: DAS FELD MUSS GROß GENUG SEIN!
Rückgabe	0 bei Fehler	Fehlermeldungen

Länge der Parameterliste (Visual-C): 20 Bytes.

ACHTUNG: BEI DATENBAUSTEINEN (TYP = 'D') MUSS DAS WERTE-FELD DOPPELT SO GROß SEIN, DA FÜR JEDES DATENWORT 2 BYTES BENÖTIGT WERDEN!

DB lesen

Liest den Inhalt eines Datenbausteines aus der *S7-SoftSPS*.

unsigned long int DLLexport ReadDB (unsigned long int nr,

unsigned short int far * Baustein,

unsigned long int far * Anzahl,

unsigned long int Synchron);

Parameter	Beschreibung	Kommentar
nr	DB-Nummer	1..16383
Baustein	Zeiger auf den DB-Inhalt	Pointer auf ein gültiges Array im Adressraum Ihrer Applikation. ACHTUNG: DAS FELD MUSS GROß GENUG SEIN!
Anzahl	Anzahl der Worte, die aus dem Baustein gelesen werden sollen	Falls der DB kleiner als die Anzahl ist, wird automatisch die Anzahl heruntermgesetzt. Pointer auf eine gültige Variable im Adressraum Ihrer Applikation.
Synchron	Immer 0	Bei S7 ohne Bedeutung
Rückgabe	0 bei Fehler	Fehlermeldungen

Länge der Parameterliste (Visual-C): 16 Bytes.

ACHTUNG: ES WIRD WORTORIENTIERT GELESEN, LOW- UND HIGHBYTE WERDEN VERTAUSCHT.

DB schreiben

Beschreibt einen Datenbaustein in der SPS.

unsigned long int DLLexport WriteDB (unsigned long int nr,

unsigned short int far * Baustein,

unsigned long int far * Anzahl,

unsigned long int Synchron);

Parameter	Beschreibung	Kommentar
nr	DB-Nummer	1..16383
Baustein	Zeiger auf den neuen DB-Inhalt	Pointer auf ein gültiges Array im Adressraum Ihrer Applikation. ACHTUNG: DAS FELD MUSS GROß GENUG SEIN!
Anzahl	Anzahl der Worte, die aus dem Baustein geschrieben werden sollen.	Falls der DB kleiner als die Anzahl ist, wird automatisch die Anzahl heruntermgesetzt. Pointer auf eine gültige Variable im Adressraum Ihrer Applikation.
Synchron	Immer 0	Bei S7 ohne Bedeutung
Rückgabe	0 bei Fehler	Fehlermeldungen

Länge der Parameterliste (Visual-C): 16 Bytes.

ACHTUNG: ES WIRD WORTORIENTIERT GESCHRIEBEN, LOW UND HIGHBYTE WERDEN VERTAUSCHT.

DB lesen S7

Liest den Inhalt eines S7-Datenbausteines in der SPS (Byteorientiert).

```
unsigned long int DLLexport ReadDB_S7 (unsigned long int nr,
                                     unsigned char int far * Baustein,
                                     unsigned long int far * Anzahl);
```

Parameter	Beschreibung	Kommentar
nr	DB-Nummer	1..16383
Baustein	Zeiger auf den DB-Inhalt	Pointer auf ein gültiges Array im Adressraum Ihrer Applikation. ACHTUNG: DAS FELD MUSS GROß GENUG SEIN!
Anzahl	Anzahl der Bytes, die aus dem Baustein gelesen werden sollen.	Falls der DB kleiner als die Anzahl ist, wird automatisch die Anzahl heruntergesetzt. Pointer auf eine gültige Variable im Adressraum Ihrer Applikation.
Rückgabe	0 bei Fehler	Fehlermeldungen

Länge der Parameterliste (Visual-C): 12 Bytes.

DB schreiben S7

Beschreibt den Inhalt eines S7-Datenbausteines in der SPS (Byteorientiert).

```
unsigned long int DLLexport WriteDB_S7 (unsigned long int nr,
                                       unsigned char int far * Baustein,
                                       unsigned long int far * Anzahl);
```

Parameter	Beschreibung	Kommentar
nr	DB-Nummer	1..16383
Baustein	Zeiger auf den neuen DB-Inhalt	Pointer auf ein gültiges Array im Adressraum Ihrer Applikation. ACHTUNG: DAS FELD MUSS GROß GENUG SEIN!
Anzahl	Anzahl der Bytes, die aus dem Baustein gelesen werden sollen.	Falls der DB kleiner als die Anzahl ist, wird automatisch die Anzahl heruntergesetzt. Pointer auf eine gültige Variable im Adressraum Ihrer Applikation.
Rückgabe	0 bei Fehler	Fehlermeldungen

Länge der Parameterliste (Visual-C): 12 Bytes.

Funktionen zur Zahlenumwandlung

FloatToMC5

Wandelt Gleitpunktzahlen im IBM-PC-Format in S7-Doppelwörter um.

```
void DLLexport FloatToMC5 (float FloatValue,
                           unsigned long far * MC5Val);
```

Parameter	Beschreibung	Kommentar
FloatVal	Gleitpunktzahl im IEEE- bzw. INTEL-Format	
MC5Val	Gleitpunktzahl im Siemens-S7-Format	

MC5ToFloat

Wandelt S7-Doppelwörter in Gleitpunktzahlen im IBM-PC-Format um.

```
void DLLexport MC5ToFloat (unsigned long far * MC5Val,
                           float FloatVal);
```

Parameter	Beschreibung	Kommentar
MC5Val	Gleitpunktzahl im Siemens-S7-Format	
FloatVal	Gleitpunktzahl im IEEE- bzw. INTEL-Format	

Zugriff mit einem Webserver

Sollten Sie beabsichtigen mit Webapplikationen (IIS) auf die S7-SoftSPS zuzugreifen, die Zugriff auf die Prozessdaten haben sollen, so setzen Sie sich bitte mit unserem Support in Verbindung. Im Lieferumfang der S7-SoftSPS ist ebenfalls ein Objekt enthalten, welches einen Datenzugriff aus Scriptsprachen (VB Script, J Script) heraus ermöglicht.

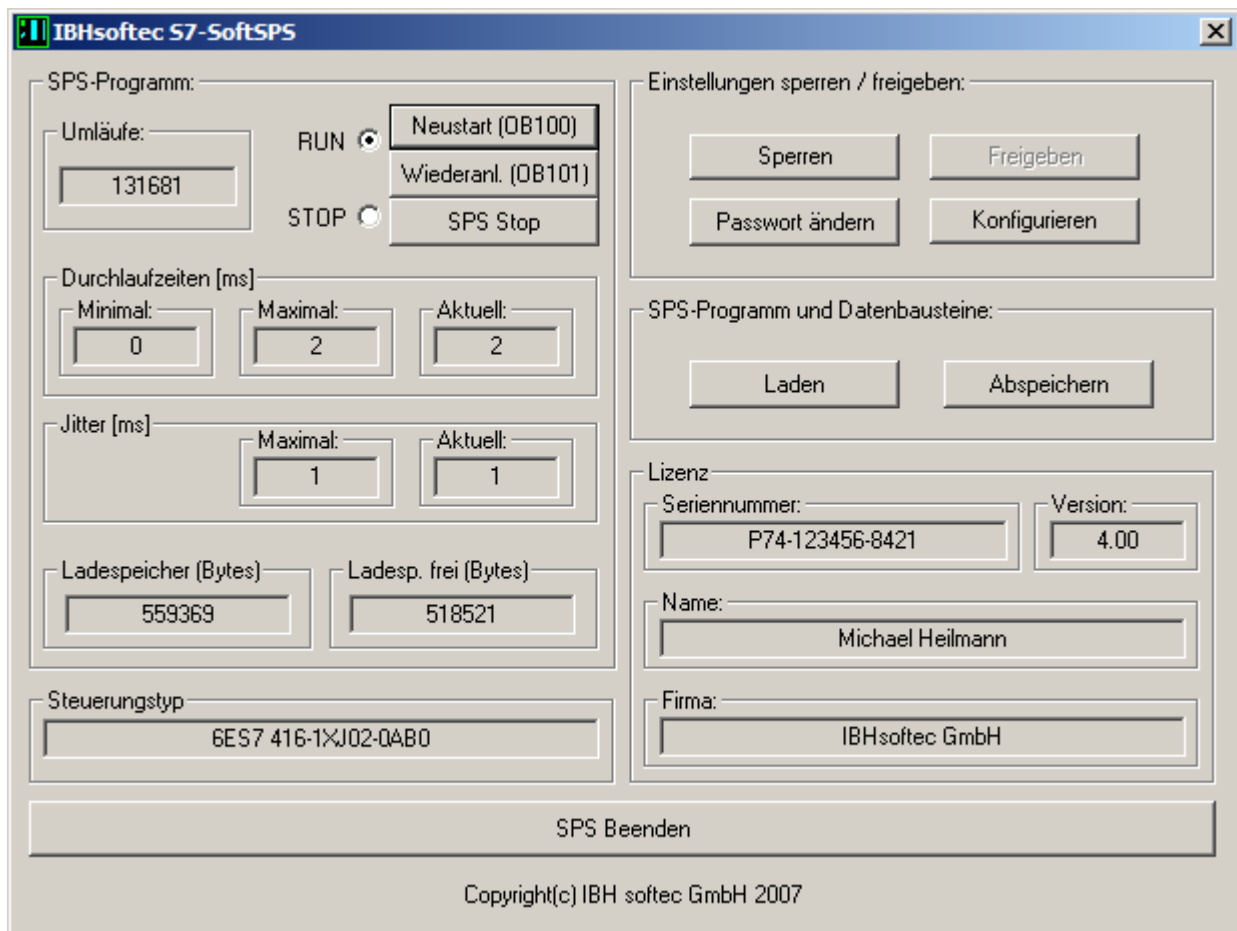
4 Oberfläche & Konfiguration

Es gibt zwei verschiedene Ansichten der *S7-SoftSPS*.

Die eine, im weiteren Text „klassische Ansicht“ genannt, zeigt Zykluszeiten an und ermöglicht es die *S7-SoftSPS* in RUN/STOP zu schalten.

Die andere ist die Konfigurationsoberfläche, die es einem ermöglicht die verwendeten Treiber und *S7-SoftSPS* Einstellungen zu verändern.

4.1 Klassische Ansicht



Dies ist die Standardansicht der IBHsoftec *S7-SoftSPS*. In ihr kann man die Anzahl der Umläufe, sowie Durchlaufzeiten und Jitter des aktuellen SPS-Programmes sehen. Verfügbarer Ladespeicher, sowie freier Ladespeicher, Steuerungstyp, Seriennummer, Version, Name und Firma des Benutzers werden ebenfalls dargestellt. Die „Klassische Ansicht“ bietet die Möglichkeit das Programm der *S7-SoftSPS* (neu) zu starten, stoppen oder die *S7-SoftSPS* zu beenden. Laden und Speichern des SPS-Programmes in das Installationsverzeichnis als auch die Möglichkeit in die Konfigurationsansicht zu wechseln sind ebenfalls vorhanden, sowie das Sperren der Oberfläche um die genannten Konfigurationsmöglichkeiten zu unterbinden.

Neustart (OB100)

Mit Betätigen der Schaltfläche „Neustart“ wird das SPS Programm gestartet. Als Erstes wird der Organisationsbaustein OB 100 abgearbeitet. Erst danach läuft die zyklische Programmbearbeitung mit OB1 an. Dies entspricht dem Anlauf einer Hardware-SPS nach Spannungswiederkehr.

Wiederanl. (OB101)

Mit der Schaltfläche „Wiederanlauf“ wird das SPS Programm gestartet. Als Erstes wird der Organisationsbaustein OB 101 abgearbeitet. Erst danach läuft die zyklische Programmbearbeitung mit OB 1 an. Dies entspricht dem Anlauf einer Hardware-SPS nach manuellem Einschalten

(STOP -> RUN).

SPS Stop

Mit Betätigen der Schaltfläche „Stop“ wird die zyklische Programmbearbeitung unterbrochen. In der Betriebsart „Stop“ wird das SPS Programm nicht abgearbeitet. Der Zustand der Prozessabbilder, der Merker, Zähler und Zeiten, die beim Eintritt des STOP–Zustandes aktuell waren, bleibt erhalten.

SPS Beenden

Mit Betätigen der Schaltfläche "SPS Beenden" wird das Programm beendet. Ein erneutes Starten des S7-SoftSPS Programms ist jederzeit möglich.

Umläufe

Die Anzahl der zyklischen Programmdurchläufe wird hier angezeigt.

Durchlaufzeiten [ms]

Die Zeit, die ein zyklischer Programmdurchlauf benötigt, wird von der S7-SoftSPS kontinuierlich gemessen.

Die minimale, maximale und die aktuelle Zykluszeit des momentan abzuarbeitenden SPS Programms werden angezeigt. Die Auflösung der Zykluszeitmessung ist eine Millisekunde.

Die Zykluszeit einer SPS ist vom Programmaufbau abhängig. Am Ende eines Zyklus speichert die S7-SoftSPS das Ergebnis der Zykluszeitmessung. Dies ist die Zeit, die zwischen einem OB1 Aufruf bis zum nächsten OB 1 Aufruf vergangen ist.

Die Zeit der zyklischen Durchläufe (OB 1 Zykluszeit) kann mit dem SPS Programm ausgelesen werden.

OB 1/LW6:	aktuelle Zykluszeit in ms	(OB1_PREV_CYCLE)
OB 1/LW8:	maximale Zykluszeit in ms	(OB1_MIN_CYCLE)
OB 1/LW10:	minimale Zykluszeit in ms	(OB1_MAX_CYCLE)

Jitter

Mit Jitter werden unregelmäßige zeitlichen Schwankungen von Vorgängen bezeichnet. Der Jitter bei einer SPS Steuerung ist die zeitliche Schwankung der Zykluszeit, die vom System verursacht wird.

Wird bei einer SPS in einem Zeit-OB (z.B. OB 10) alle 10 Millisekunden ein Impuls (Ausgang) ausgegeben, so ist der Jitter die Schwankung der Zeit, die von den 10 Millisekunden abweicht.

Zeitschwankungen sind sowohl bei Hardware-SPS-Steuerungen wie auch bei Software-SPS-Steuerungen vorhanden.

Neben programmtechnischen Zeitverschiebungen (programmierte Programmunterbrechungen wie z.B. Prozessalarmlen) können bei einer Software-SPS-Steuerung Zeitschwankungen durch Festplattenzugriffe, Netzwerkzugriffe usw. verursacht werden.

Der Jitter der *S7-SoftSPS* wird kontinuierlich gemessen. Die maximale und die aktuelle Zeitabweichung von einem imaginären, festen Zeittakt wird angezeigt. Die Auflösung der Zeitmessung ist eine Millisekunde.

Ladespeicher

Der Ladespeicher ist die Größe die ein SPS-Programm belegen kann in Bytes. Die angezeigte Größe des Ladespeichers ist die Gesamtgröße die von der *S7-SoftSPS* bereit gehalten wird. Der freie Ladespeicher ist die Größe die noch frei ist und von zusätzlichen Bausteinen belegt werden kann. Wenn zu wenig freier Ladespeicher zur Verfügung steht, kann das PG keine neuen Bausteine in die *S7-SoftSPS* übertragen.

Einstellungen sperren

Eine Manipulation der *S7-SoftSPS* durch den Bediener kann mit Hilfe eines Passworts verhindert werden.

Um die *S7-SoftSPS* zu sperren betätigt man die Schaltfläche „Einstellungen sperren“.

Die *S7-SoftSPS* -Oberfläche kann nun nicht mehr verwendet werden um das Verhalten der *S7-SoftSPS* zu beeinflussen.

ACHTUNG: AUF DEN PROGRAMMABLAUF DER *S7-SOFTSPS* HAT DAS SPERREN KEINEN EINFLUSS.

Einstellungen freigeben

Die gesperrte *S7-SoftSPS* wird freigegeben.

Um die gesperrte *S7-SoftSPS* freizugeben betätigt man die Schaltfläche „Einstellungen freigeben“.

Die *S7-SoftSPS*-Oberfläche kann nun wieder benutzt werden um Einstellungen an der *S7-SoftSPS* vorzunehmen.

Passwort ändern

Ändert das Passwort mit dem die *S7-SoftSPS* gesperrt werden kann.

Nach Betätigung der Schaltfläche „Passwort ändern“ erscheint ein Dialog in dem man zur Bestätigung der Änderung das alte und zweimal das neue Passwort eingeben muss.

Das Passwort ist nun geändert.

Konfigurieren

Schaltet die Oberfläche in den Konfigurationsmodus.

Nach Betätigung der Schaltfläche „Konfigurieren“ ändert die Oberfläche ihre Darstellung in die Konfigurationsansicht. In dieser kann man Treiber einbinden und parametrieren, Eingänge, Ausgänge,

Merker und Datenbausteine beobachten, sowie die grundlegenden Einstellungen an der *S7-SoftSPS* vornehmen.

Die *S7-SoftSPS* kann nun konfiguriert werden.

Laden

Das momentan im Installationsverzeichnis der *S7-SoftSPS* in der Datei „S7.BIN“ gespeicherte SPS-Programm wird in die *S7-SoftSPS* geladen.

Abspeichern

Das in der *S7-SoftSPS* befindliche SPS-Programm wird als „S7.BIN“ im Installationsverzeichnis der *S7-SoftSPS* gespeichert.

Steuerungstyp

S7-SoftSPS kompatible Bestellnummern:

6ES7 315-1AF01-0AB0	<i>S7-SoftSPS 315</i> ohne Profibus
6ES7 315-2AG10-0AB0	<i>S7-SoftSPS 315</i> mit Profibus
6ES7 416-1XJ02-0AB0	<i>S7-SoftSPS 416</i> ohne Profibus
6ES7 416-2XK02-0AB0	<i>S7-SoftSPS 416</i> mit Profibus

Seriennummer

Die Seriennummer der installierten *S7-SoftSPS*

Version

Die Versionsnummer der installierten *S7-SoftSPS*

Name

Name des Eigentümers der Lizenz

Firma

Firmenname des Lizenzinhabers

4.2 Konfigurationsansicht

In dieser Ansicht gibt es die Möglichkeit die *S7-SoftSPS* Parameter zu konfigurieren, Treiber einzubinden und parametrieren sowie Eingänge, Ausgänge, Merker und Datenbausteine zu beobachten.

Menü

Das Menü bietet das Speichern und Öffnen von Konfigurationsdateien an. Des weiteren kann man die Ansicht wieder auf „Klassische Ansicht“ umschalten oder andere Eigenschaftsfenster anzeigen. Ebenso ist über das Menü diese Hilfe und die IBHsoftec GmbH Kontaktinformationen erreichbar.

Werkzeuggestreife

Die Werkzeuggestreife bietet schnellen Zugriff auf häufig benötigte Menüeinträge.

Konfigurationsbaum

Der Konfigurationsbaum zeigt alle Einstellungsfenster und eingebundenen Treiber an.

Um in die gewünschte Einstellung zu gelangen markiert man ein Baumelement mit der Maus.

Der gewählte Baumeintrag wird nun im Einstellungsfenster dargestellt.

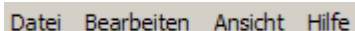
Einstellungsfenster

Hier kann man die Eigenschaften des gewählten Elementes editieren.

Hilfefenster

Hier wird eine kurze Hilfe zum aktuellen Element angezeigt.

4.3 Menü



Das Menü enthält die folgenden Einträge: Datei, Bearbeiten, Ansicht, Hilfe.

Das Menü gliedert sich in folgende vier Bereiche:

- Datei
- Bearbeiten
- Ansicht
- Hilfe

4.3.1 Menü Datei



SoftSPS Konfiguration Öffnen

Zeigt einen „Datei öffnen“-Dialog an. In diesem wählt man nun die gewünschte Konfigurationsdatei (PLC43.INI) aus. Diese wird nun in die Oberfläche geladen.

SoftSPS Konfiguration Speichern

Speichert die aktuelle Konfiguration der SoftSPS im Installationsverzeichnis der *S7-SoftSPS*.

SoftSPS Konfiguration Speichern unter

Öffnet einen „Datei speichern“-Dialog. In diesem wählt man nun den Pfad unter dem man die Konfigurationsdatei (PLC43.INI) speichern will.

SoftSPS neu starten

Beendet die *S7-SoftSPS* und startet diese neu.

SoftSPS beenden

Beendet die *S7-SoftSPS* auf dem Computer.

Konfiguration und SoftSPS beenden

Beendet die *S7-SoftSPS* und die Konfigurationsoberfläche.

Konfiguration beenden

Schließt die Konfigurationsoberfläche.

4.3.2 Menü Bearbeiten



Einstellungen sperren

Sperrt die Oberfläche nach Eingabe des Passwortes.

Einstellungen freigeben

Gibt die Oberfläche nach Eingabe des Passwortes wieder frei.

Passwort ändern

Ändert das Passwort um die Oberfläche zu sperren/freizugeben.

I/O Monitor Auswahl speichern

Speichert die aktuelle Operandenauswahl im I/O Monitor ab, so dass sie nach erneutem Start der Oberfläche wieder zur Verfügung stehen.

I/O Monitor Auswahl löschen

Löscht die aktuelle Operandenauswahl im I/O Monitor, so dass bei erneutem Start der Oberfläche keine Operanden mehr gewählt sind.

Neustart (OB100)

Startet die *S7-SoftSPS* mit OB100 neu.

Wiederanlauf (OB101)

Startet die *S7-SoftSPS* mit OB101 neu.

SoftSPS stoppen

Schaltet die *S7-SoftSPS* in STOP.

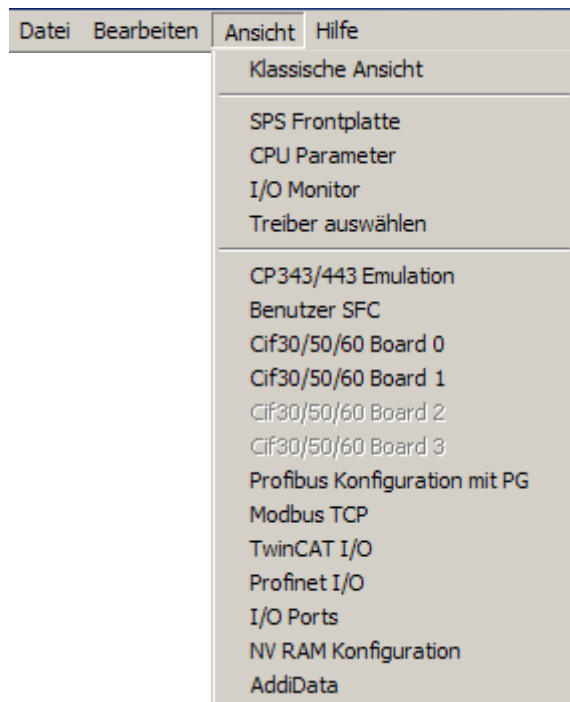
SPS-Programm speichern

Speichert das momentan laufende Programm der *S7-SoftSPS* in die Datei „S7.BIN“ im Installationsverzeichnis der *S7-SoftSPS*.

SPS-Programm laden

Lädt das Programm aus der Datei „S7.BIN“ im Installationsverzeichnis der *S7-SoftSPS* in selbige.

4.3.3 Menü Ansicht

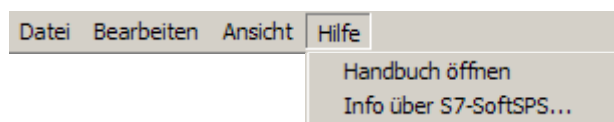


Klassische Ansicht

Schaltet die Oberfläche von „Konfiguration“ auf „Klassisch“ um.

Die anderen Einträge wählen jeweils die Ansicht des gleichnamigen Konfigurationsfensters aus.

4.3.4 Menü Hilfe



Handbuch öffnen

Öffnet das *S7-SoftSPS*-Handbuch.

Info über S7-SoftSPS...

Öffnet ein Fenster in welchem die Versionsnummer der *S7-SoftSPS* und die *IBHsofttec* Kontaktinformationen präsentiert werden.

4.4 Werkzeugleiste



SoftSPS Konfiguration Öffnen

Öffnet einen Dialog in dem die gewünschte Konfigurationsdatei (PLC43.INI) ausgewählt werden kann. Diese wird in die Oberfläche geladen.

SoftSPS Konfiguration Speichern

Speichert die aktuelle Konfiguration der *S7-SoftSPS* im Installationsverzeichnis der *S7-SoftSPS*.

SoftSPS Konfiguration Speichern unter

Öffnet einen Dialog in dem die aktuelle Konfiguration in die Konfigurationsdatei (PLC43.INI) geschrieben werden kann.

Neustart

Startet die *S7-SoftSPS* mit OB100 neu.

Warmstart

Startet die *S7-SoftSPS* mit OB101 neu.

SoftSPS Stop

Schaltet die *S7-SoftSPS* in STOP.

SoftSPS Binärdatei speichern

Speichert das momentan laufende Programm der *S7-SoftSPS* in die Datei „S7.BIN“ im Installationsverzeichnis der *S7-SoftSPS*.

SoftSPS Binärdatei laden

Lädt das in der Binärdatei „S7.BIN“ enthaltene SPS-Programm aus dem Installationsverzeichnis der *S7-SoftSPS*.

Neustart um Konfigurationsänderungen zu übernehmen

Beendet die *S7-SoftSPS* und startet sie neu.

Handbuch öffnen

Öffnet das *S7-SoftSPS*-Handbuch.

Info über

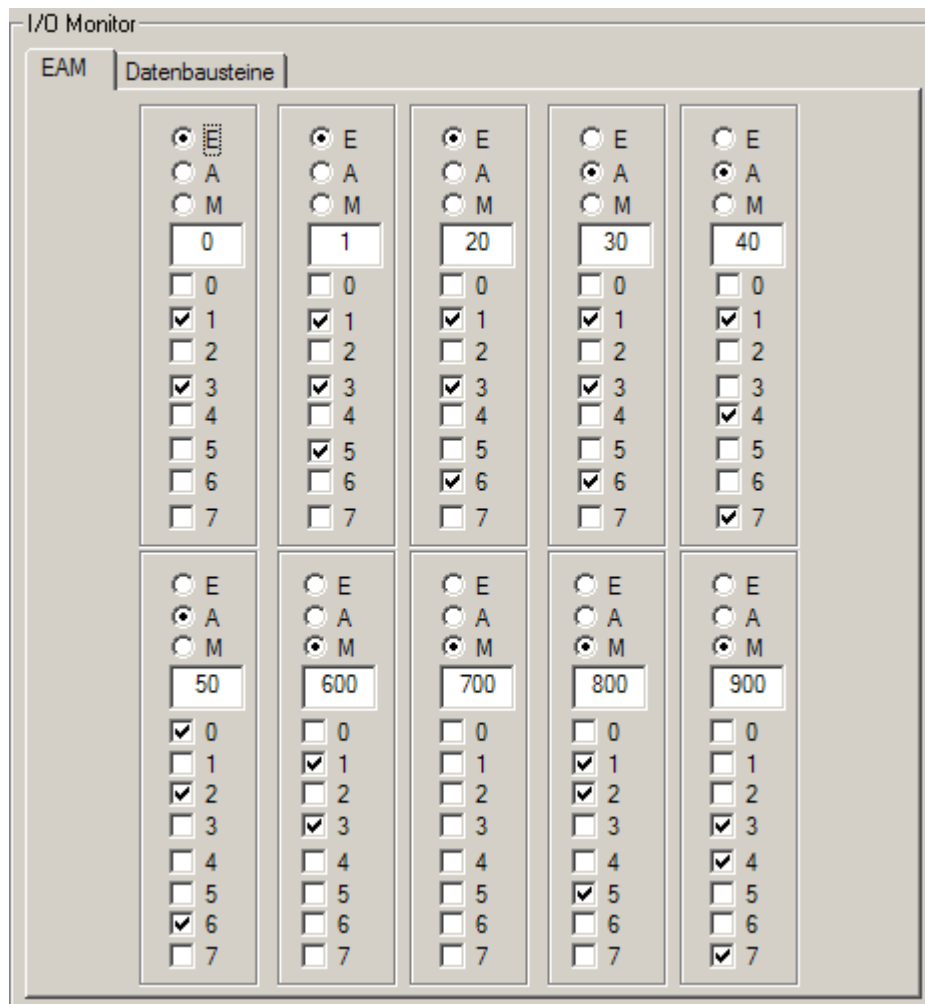
Öffnet ein Fenster in welchem die Versionsnummer der *S7-SoftSPS* und die *IBHsoftec* Kontaktinformationen präsentiert werden.

4.5 I/O Monitor

Dieser Dialog ermöglicht es einem Datenbereiche aus der S7-SoftSPS zu beobachten.

Es gibt zwei (2) Unterkategorien, EAM für Eingänge, Ausgänge und Merker sowie Datenbausteine.

EAM



Die gleichzeitige Darstellung von zehn (10) Byte ist möglich. Durch Anwahl wird ein Feld einem Eingangsbyte, einem Ausgangsbyte oder einem Merkerbyte zugewiesen. Die Schaltflächen sind als Schalter mit Lampen zu verstehen. Ein Haken zeigt an, dass das Bit gesetzt (logisch eins) ist. Die Schaltfläche kann durch den Prozess (Anwenderprogramm) oder durch Markieren (rück)gesetzt werden.

Datenbausteine

The screenshot shows the 'I/O Monitor' window with the 'Datenbausteine' tab selected. The 'Auswahl:' section includes a 'DB Nr.:' field with the value '10' and three radio buttons: 'Byte', 'Wort', and 'Doppelwort'. The 'Doppelwort' radio button is selected. A 'Wert Ändern' button is located to the right. Below this is a table with three columns: 'Adresse', 'Hex', and 'Dezimal'. The table lists data blocks from DB10.DBD0 to DB10.DBD88, showing their hexadecimal and decimal values.

Adresse	Hex	Dezimal
DB10.DBD0	0x1234567	19088743
DB10.DBD4	0x89abcdef	-1985229329
DB10.DBD8	0x11112222	286335522
DB10.DBD12	0x33334444	858997828
DB10.DBD16	0x55556666	1431660134
DB10.DBD20	0x77778888	2004322440
DB10.DBD24	0x9999aaaa	-1717982550
DB10.DBD28	0xbbbbcccc	-1145320244
DB10.DBD32	0xddddeeee	-572657938
DB10.DBD36	0xffff0000	-65536
DB10.DBD40	0x178965	1542501
DB10.DBD44	0x2465	9317
DB10.DBD48	0x3469	13417
DB10.DBD52	0x78967345	2023125829
DB10.DBD56	0x5997896	93943958
DB10.DBD60	0x664569	6702441
DB10.DBD64	0x7789	30601
DB10.DBD68	0x08	8
DB10.DBD72	0x09	9
DB10.DBD76	0x12	18
DB10.DBD80	0x13	19
DB10.DBD84	0x14	20
DB10.DBD88	0x15	21

Dynamische Darstellung eines Datenbausteins und dessen Inhalt. Der Wert des Bytes, Worts oder Doppelworts kann mit der Schaltfläche "Wert ändern" dezimal oder hexadezimal geändert werden.

5 Konfiguration

In diesem Kapitel wird beschrieben wie man die S7-SoftSPS konfigurieren kann.

- CPU Parameter
- Treiber hinzufügen/entfernen
- Addidata
- Beckhoff TwinCAT I/O
- CIF30/50/60
- CP343/CP443 Emulation
- I/O Port
- Modbus on TCP
- NVRAM
- Softnet PN I/O
- User SFC

5.1 CPU Parameter

Hier werden grundlegende Einstellungen an der S7-SoftSPS vorgenommen

- Konfiguration mit Hilfe der Oberfläche
- Konfiguration in der PLC43.INI

5.1.1 Konfiguration mit Hilfe der Oberfläche

The screenshot shows the 'CPU Parameter' configuration window. It contains several sections:

- Einstellungen:** A group box containing five unchecked checkboxes:
 - SPS Programm beim Start laden
 - SPS Programm beim Beenden abspeichern
 - M/T/Z remanent (Bereiche über Hardwarekonfigurator festlegen)
 - Speichern in RUN erlauben
 - Singleprozessormodus (auch bei Hyperthreading CPUs)
- Schlüsselschalterstellung:** A dropdown menu set to 'RUN-P'.
- Betriebssystem:** A dropdown menu set to 'Windows 2000/Windows XP'.
- Sprache:** A dropdown menu set to 'Deutsch'.
- MPI über serielle Schnittstelle:** A dropdown menu set to 'Deaktiviert'.
- SPS Speichergröße (Byte):** A text input field containing '500000'.
- SPS : Windows Rechenzeit [ms]:** A configuration field showing '1' in a box, followed by a colon, '1' in a spinner box, an equals sign, '50' in a box, and '% SPS'.
- Standard:** A button at the bottom of the window.

In diesem Fenster werden die Grundeinstellungen der S7-SoftSPS vorgenommen.

SPS Programm beim Start laden

Das momentan im Installationsverzeichnis der *S7-SoftSPS* in „S7.BIN“ gespeicherte SPS-Programm wird in die *S7-SoftSPS* geladen sobald diese gestartet wird. Die Anzeige des Jitter wird zurückgesetzt.

Ist die Option „SPS Programm beim Start laden“ gewählt, so wird das im Installationsverzeichnis der *S7-SoftSPS* in der Datei „S7.BIN“ gespeicherte SPS-Programm beim Starten der *S7-SoftSPS* in die *S7-SoftSPS* geladen. Das Starten der *S7-SoftSPS* entspricht dem Anlauf einer Hardware-SPS nach Spannungswiederkehr. Somit wird als Erstes der Organisationsbaustein OB 101 abgearbeitet. Erst danach läuft die zyklische Programmbearbeitung mit OB 1 an. Die Signalzustände von Merkern (M), Zeiten (T) und Zählern (Z) werden in die *S7-SoftSPS* übernommen, sofern dies im Hardwarekonfigurator eingestellt ist.

Sollten bei dem nächsten Starten der *S7-SoftSPS* Probleme auftreten, so löschen Sie die Datei „S7.BIN“ im Installationsverzeichnis der *S7-SoftSPS*. Solche Probleme können auftreten, wenn der eigentliche Speicherprozess gestört wurde. In einem solchen Fall muss das SPS-Programm mit Hilfe des Programmiersystems erneut geladen werden. Die Merker (M), Zeiten (T) und Zähler (Z) sind in einem solchen Fall in ihrem Ausgangszustand.

SPS Programm beim Beenden speichern

Das in der *S7-SoftSPS* befindliche SPS-Programm wird im Installationsverzeichnis der *S7-SoftSPS* in die Datei „S7.BIN“ gespeichert, sobald die *S7-SoftSPS* beendet wird.

Ist die Option „SPS Programm beim Beenden speichern“ gewählt, so wird mit dem Herunterfahren der *S7-SoftSPS* das SPS-Programm in die Datei „S7.BIN“ im *S7-SoftSPS* Installationsverzeichnis gespeichert. Die Signalzustände von Merkern (M), Zeiten (T) und Zählern (Z) werden ebenfalls gespeichert.

M/T/Z remanent (Bereiche Über Hardwarekonfigurator festlegen)

Der Speicher wird für M,T,Z remanent gehalten (Beginnend bei Byte 0 bis zu dem über den Hardwarekonfigurator eingestellten Wert)

Speichern in RUN erlauben

Ermöglicht das Speichern ohne die SPS zu stoppen

ACHTUNG: WIRD DAS PROGRAMM IM LAUFENDEN BETRIEB GESPEICHERT, KANN DIE ZYKLUSZEIT STEIGEN.

Singleprozessormodus (auch bei Hyperthreading CPUs)

Stellt fest den Singleprozessormodus ein. Bei echten Doppelprozessoren hat die SPS dann 25% der möglichen Rechenzeit. Bei Hyperthreading Prozessoren sollte diese Option aktiviert werden, da ansonsten die Prozessorlast und die Zykluszeit schwanken können.

Schlüsselschalterstellung

RUN In diesem Modus arbeitet die *S7-SoftSPS* das SPS-Programm ab.

RUN-P In diesem Modus arbeitet die *S7-SoftSPS* das SPS-Programm ab. Das SPS-Programm kann während dem Betrieb bearbeitet werden.

Betriebssystem

Wahl des verwendeten Betriebssystems.

Sprache

Wahl der Sprache in der die Oberfläche der *S7-SoftSPS* dargestellt werden sollen.

MPI über serielle Schnittstelle

Die Verbindung zur *S7-SoftSPS* kann über die gewählte Schnittstelle aufgebaut werden.

SPS Speichergröße

Die Größe des Ladespeichers der *SoftSPS PLC S7-416*. Die Größe des Ladespeichers der *SoftSPS PLC S7-315* ist auf 256 KByte begrenzt.

SPS : Windows Rechenzeit (ms)

Aufteilung der PC-CPU Rechenzeit zwischen *S7-SoftSPS* und Windows®.

Standard

Stellt die Standard Einstellungen wieder her.

5.1.2 Konfiguration in der PLC43.INI

In dieser Sektion werden die Grundeinstellungen der *S7-SoftSPS* vorgenommen.

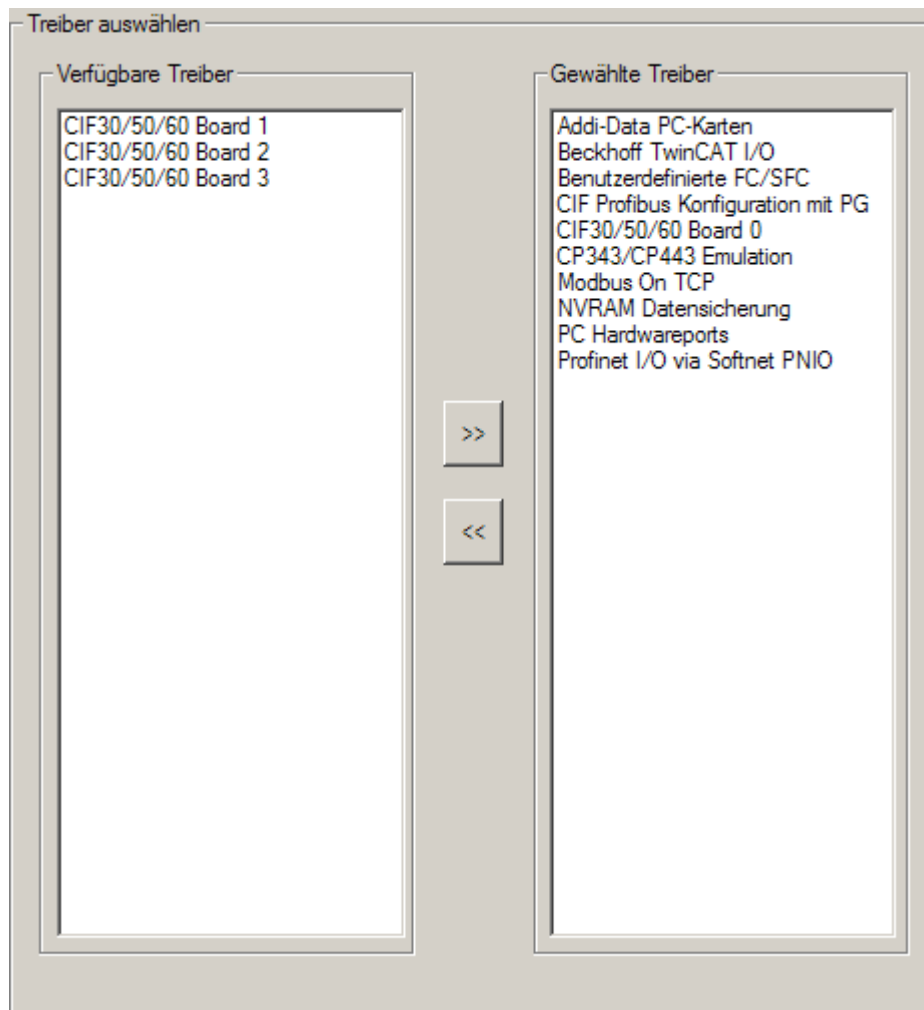
Keyname	Wertebereich	Defaultwert	Beschreibung
OS	NT4 / W2K	W2K	Verwendetes Betriebssystem NT4 = WinNT4, W2K = Win2000/XP
LANGUAGE	DEU / ENG	DEU	Sprache des <i>S7-SoftSPS</i> Dialogfelds DEU = Deutsch, ENG = Englisch
MPI	0 .. 4	0	Verwendete Schnittstelle 0 = Intern, 1 = COM1, 2 = COM2, usw...
LOAD	0 / 1	0	0 = SPS-Programm beim Start nicht laden 1 = SPS-Programm beim Start laden aus S7.BIN laden
STORE	0 / 1	0	0 = SPS-Programm beim Beenden nicht speichern 1 = SPS-Programm beim Beenden in S7.BIN speichern
S7_STORE	50000..... Speicherende	500000	Größe des Ladespeichers in Byte Die <i>SoftSPS PLC S7-315</i> hat einen festgelegten Ladespeicher von 297225 Byte
STORE_NO_STOP	0 / 1	0	1 = Speichern ohne die SPS zu stoppen ACHTUNG: WIRD DAS PROGRAMM IM LAUFENDEN BETRIEB GESPEICHERT, KANN DIE ZYKLUSZEIT STEIGEN.
REM	0 / 1	0	Bei 1 wird der Speicher für M,T,Z remanent gehalten (Beginnend bei Byte 0 bis zu dem über den Hardwarekonfigurator eingestellten Wert).
KEY_SWITCH	1 / 2	2	1 = Meldet dem PG, dass der Schlüsselschalter auf RUN steht, kein Bausteindownload möglich 2 = Meldet dem PG, dass der Schlüsselschalter auf RUN-P steht.
NOMP	0 / 1	0	1 = Stellt fest den Singleprozessormodus ein. Bei echten Doppelprozessoren hat die SPS dann 25% der möglichen Rechenzeit. Bei Hyperthreading Prozessoren muss dieser Wert auf 1 stehen, da ansonsten die Prozessorlast und die Zykluszeit schwanken können.
WINTIME	1 .. 8	1	Reservierte CPU Rechenzeit für Windows in ms 3 = 3 ms Windows und 1 ms SPS, entspricht 75% Rechenzeit für Windows, 25% für die <i>S7-SoftSPS</i> .

5.2 Treiber hinzufügen/entfernen

Hier werden Treiber zur *S7-SoftSPS* hinzugefügt oder entfernt

- Treiber hinzufügen/entfernen mit Hilfe der Oberfläche
- Treiber hinzufügen/entfernen in der PLC43.INI

5.2.1 Treiber hinzufügen/entfernen mit Hilfe der Oberfläche



Hier können die gewünschten Hardwaretreiber der *S7-SoftSPS* hinzugefügt werden.

Auf der linken Seite sind die verfügbaren Treiber aufgelistet, auf der rechten Seite die der *S7-SoftSPS* momentan zugeordneten Treiber

Treiber hinzufügen

Einen Treiber für die *S7-SoftSPS* auswählen.

Mit Betätigen der „>>“ Schaltfläche wird der markierte Treiber in das rechte Feld (Gewählte Treiber) verschoben und somit der *S7-SoftSPS* hinzugefügt.

Der Treiber ist nun nach einem Neustart der *S7-SoftSPS* verfügbar.

Treiber entfernen

Einen Treiber aus der Konfiguration der S7-SoftSPS entfernen.

Mit Betätigen der „<<“ Schaltfläche wird der markierte Treiber in das linke Feld (Verfügbare Treiber) verschoben und somit von der S7-SoftSPS entfernt. Die Treiber können der S7-SoftSPS jederzeit wieder hinzugefügt werden.

Der Treiber ist nach einem Neustart der S7-SoftSPS nicht mehr verfügbar.

ACHTUNG: SIND TREIBER GEWÄHLT, DEREN ENTSPRECHENDE HARDWARE NICHT VORHANDEN IST, KANN ES ZU STÖRUNGEN DER S7-SOFTSPS KOMMEN.

5.2.2 Treiber hinzufügen/entfernen in der PLC43.INI

In der Sektion [OEM] werden die erforderlichen Hardware Treiber DLL's eingetragen. Bis zu zehn Treiber DLL's können hier eingebunden werden. Im Falle von Hilscher CIF Karten ist die Anzahl durch die Hardware auf 4 begrenzt.

Keyname	Wertebereich	Defaultwert	Beschreibung
DLL0	CIF30D0.DLL	---	Bei Verwendung einer CIF-Karte
DLL1	CIFCFG.DLL	---	Der Treiber CIFCFG.DLL wird benötigt, um den Profibus mit dem Siemens SIMATIC® Manager oder S7 für Windows® Hardwarekonfigurator zu konfigurieren. Dieser stellt auch Online Funktionen über Profibus und DPV1 Dienste zur Verfügung. Der Treiber kann nur in Verbindung mit der ersten CIF-Karte verwendet werden.
DLL2 .. DLL9	---	---	Weitere Kartentreiber

Es gibt 2 Möglichkeiten, die Feldbuskomponenten zu konfigurieren:

Soll eine Profibus Masterbaugruppe über den S7 für Windows® Hardwarekonfigurator oder den Step®7 Hardwarekonfigurator konfiguriert werden, so sind in der Sektion [OEM] folgende Treibereinträge zu wählen:

[OEM]

DLL0=CIF30D0.DLL

DLL1=CIFCFG.DLL

Diese Einträge stellen sicher, dass die mit dem S7 für Windows® Hardwarekonfigurator oder dem Step®7 Hardwarekonfigurator erzeugten Systemdaten ausgewertet werden und das CIF Modul (nur Profibus) beim Hochlauf der SPS automatisch konfiguriert wird.

Sollen mehrer Profibuskarten in einem PC betrieben werden, lässt sich nur die erste Karte automatisch konfigurieren. Alle weiteren Karten müssen mit dem Hilscher Systemkonfigurator SyCon parametrieren werden.

Sollen andere CIF Feldbuskarten (z.B. Interbus, CAN) verwendet werden, so kann dies nicht durch den S7 für Windows® Hardwarekonfigurator oder den Step®7 Hardwarekonfigurator realisiert werden, da beide Konfiguratoren die anderen Bussysteme nicht unterstützen.

In diesem Falle muss, bevor Sie die Feldbuskomponenten mit der S7-SoftSPS ansprechen können, die CIF Konfiguration mit dem Systemkonfigurator SyCon der Fa. Hilscher erstellt werden. Falls Sie die voreingestellten Startadressen nicht über die Sondereinstellungen verändert haben, gelten die nachfolgenden Zuordnungen. Die Adressen der digitalen Peripherie legen Sie in den CIF-Adressbereich von 0..511.

Zudem müssen im Systemkonfigurator SyCon unter Setup unbedingt folgende Systemparameter eingestellt werden:

- Gesteuerte Freigabe der Kommunikation durch das Anwenderprogramm
- Byteadressen
- Intel-Format
- Gepuffert, anwendergesteuert

In der Sektion [OEM] ist lediglich folgender Treibereintrag zu wählen:

[OEM]

DLL0=CIF30D0.DLL

Aus Kompatibilitätsgründen zu bestehenden Anwendungen wird der klassische Treiber cif30x0.dll, wie er in der Version 2.x der S7-SoftSPS lange verwendet wurde, weiterhin mitgeliefert und installiert, jedoch nicht mehr standardmäßig eingebunden. Applikationen, die die Diagnosebausteine SFB210 und SFB252 nicht genutzt hatten, können problemlos auf den Treiber Cif30D(n).dll wechseln. Sollte die Umstellung zu aufwendig sein, kann der klassische Treiber noch verwendet werden.

5.3 Addidata

Mithilfe des ADDIPACK Treibers kann die S7-SoftSPS auf Addidata I/O Karten zugreifen

- Zugriff auf Addidata I/O Karten im SPS-Programm
- Konfigurieren des Treibers mit Hilfe der Oberfläche
- Konfigurieren des Treibers in der PLC43.INI

5.3.1 ADDI-DATA Konfiguration

Der ADDI-DATA-Treiber ermöglicht es, Digital I/O und Analog I/O Karten von ADDI-DATA aus dem SPS-Programm der S7-SoftSPS heraus anzusprechen. Dies erfolgt durch Zugriffe auf eine in der ADDIPACK-Software konfigurierte virtuelle Karte über Funktionen (FC-Bausteine) die im SPS-Programm aufgerufen werden müssen. Ein zyklischer Datenaustausch über das Prozessabbild ist aufgrund des Aufbaus der ADDI-DATA-Funktionen nicht sinnvoll und daher als FC-Aufruf realisiert.

Voraussetzungen

- Die Interfacekarte muss im Rechner eingebaut sein
- Die entsprechenden Treiber von ADDI-DATA müssen installiert sein
- Die Software ADDIPACK muss installiert und korrekt lizenziert sein
- Die Interfacekarte muss im ADDIPACK konfiguriert sein (sollte es hierbei Probleme geben wenden Sie sich bitte an die ADDI-DATA Dokumentation und/oder an den ADDI-DATA Support)
- In der S7-SoftSPS muss der "ADDI-DATA PC-Karten" Treiber gewählt sein

Zugriff auf die I/O Karte

Die durch den Treiber in der *S7-SoftSPS* installierten FC-Bausteine greifen auf Funktionen der von ADDI-DATA bereitgestellten ADDIDATA.DLL zu. Diese Funktionen werden transparent von den entsprechenden FC-Bausteinen in das SPS-Programm eingeblendet. Für genauere Informationen lesen sie bitte die entsprechende Dokumentationen von ADDI-DATA. Der in der ADDI-DATA Dokumentation beschriebene Parameter „dw_DriverHandle“ muss nicht beachtet werden, da sich der *S7-SoftSPS* Treiber selbst um das Öffnen und Schließen sowie die Übergabe des „DriverHandle“ an die ADDI-DATA Funktionen kümmert.

Read8DigitalInputs:

entspricht der Funktion „b_ADDIDATA_Read8DigitalInputs (DWORD dw_DriverHandle, BYTE b_Port, PBYTE pb_PortValue)“.

Aufruf:

```
CALL          FC 100
  IN0         := B#16#0           //b_Port
  RET_VAL     := MW 0             //RET_VAL Fehlercode
  OUT1        := MB 2             //pb_PortValue
```

Der Parameter IN0 entspricht dem ADDI-DATA Parameter b_Port, der Parameter OUT1 entspricht dem ADDI-DATA Parameter pb_PortValue. Um den dw_DriverHandle kümmert sich der Treiber selbst, in RET_VAL steht das Funktionsergebnis. Dieses Schema gilt analog für alle Treiberfunktionen.

Read16DigitalInputs:

entspricht der Funktion „b_ADDIDATA_Read16DigitalInputs(DWORD dw_DriverHandle, BYTE b_Port, PWORD pw_PortValue)“.

Aufruf:

```
CALL          FC 101
  IN0         := B#16#0           //b_Port
  RET_VAL     := MW 0             //RET_VAL Fehlercode
  OUT1        := MW 2             //pw_PortValue
```

Read32DigitalInputs:

entspricht der Funktion „b_ADDIDATA_Read32DigitalInputs(DWORD dw_DriverHandle, BYTE b_Port, PDWORD pdw_PortValue)“.

Aufruf:

```
CALL          FC 102
  IN0         := B#16#0           //b_Port
  RET_VAL     := MW 0             //RET_VAL Fehlercode
  OUT1        := MD 2             //pdw_PortValue
```

Set8DigitalOutputs:

entspricht der Funktion „b_ADDIDATA_Set8DigitalOutputsOn(DWORD dw_DriverHandle, BYTE b_Port, BYTE b_PortValue)“.

Aufruf:

```
CALL      FC 103
  IN0     := B#16#0           //b_Port
  IN1     := B#16#0           //b_PortValue
  RET_VAL := MW 0             //RET_VAL Fehlercode
```

Set16DigitalOutputs:

entspricht der Funktion „b_ADDIDATA_Set16DigitalOutputsOn(DWORD dw_DriverHandle, BYTE b_Port, WORD w_PortValue)“.

Aufruf:

```
CALL      FC 104
  IN0     := B#16#0           //b_Port
  IN1     := W#16#0           //w_PortValue
  RET_VAL := MW 0             //RET_VAL Fehlercode
```

Set32DigitalOutputs:

entspricht der Funktion „b_ADDIDATA_Set32DigitalOutputsOn(DWORD dw_DriverHandle, BYTE b_Port, DWORD dw_PortValue)“.

Aufruf:

```
CALL      FC 105
  IN0     := B#16#0           //b_Port
  IN1     := DW#16#0          //dw_PortValue
  RET_VAL := MW 0             //RET_VAL Fehlercode
```

InitAnalogInput:

entspricht der Funktion „b_ADDIDATA_InitAnalogInput(DWORD dw_DriverHandle, WORD w_Channel, pstr_InitAnalogInput ps_InitParameters, DWORD dw_StructSize)“.

Diese Funktion erwartet eine Datenstruktur zum Initialisieren des Eingangs (siehe Dokumentation zur Karte). Diese wird mit Hilfe eines ANY-Pointers vom Typ „BYTE“ an die FC übergeben (ps_InitParameters). Ein separater Parameter für die Länge der Struktur (dw_StructSize) ist nicht erforderlich, dieser wird aus der Länge des ANY-Pointers gebildet.

Aufruf:

```
CALL      FC 106
  IN0     := W#16#0           //w_Channel
  IN1     := P#M 8.0 BYTE 10 //ANY-Pointer auf die Struktur
  RET_VAL := MW 0             //RET_VAL Fehlercode
```

ReleaseAnalogInput:

entspricht der Funktion „b_ADDIDATA_ReleaseAnalogInput(DWORD dw_DriverHandle, WORD w_Channel)“.

Aufruf:

```
CALL      FC 107
  IN0     := W#16#0           //w_Channel
  RET_VAL := MW 0            //RET_VAL Fehlercode
```

InitAnalogOutput:

entspricht der Funktion „b_ADDIDATA_Init1AnalogOutput(DWORD dw_DriverHandle, WORD w_Channel, BYTE b_VoltageMode, BYTE b_Polarity)“.

Aufruf:

```
CALL      FC 108
  IN0     := W#16#0           //w_Channel
  IN1     := B#16#0           //b_VoltageMode
  IN2     := B#16#0           //b_Polarity
  RET_VAL := MW 0            //RET_VAL Fehlercode
```

ReadAnalogInput:

entspricht der Funktion „b_ADDIDATA_Read1AnalogInput(DWORD dw_DriverHandle, WORD w_Channel, DWORD dw_ConvertingTime, BYTE b_ConvertingTimeUnit, BYTE b_InterruptFlag, PDWORD pdw_ChannelValue)“.

Diese Funktion erwartet einen Pointer auf die Rückgabewerte (siehe Dokumentation zur Karte). Diese wird mit Hilfe eines ANY-Pointers vom Typ „DWORD“ an die FC übergeben (pdw_ChannelValue). Beachten Sie, dass der ANY-Pointer immer die Länge DWORD 3 (12 Byte) besitzen muss.

Der Parameter b_InterruptFlag wird nicht verwendet, da der Treiber ausschließlich im Polling-Mode arbeitet.

Aufruf:

```
CALL      FC 109
  IN0     := W#16#0           //w_Channel
  IN1     := DW#16#0          //dw_ConvertingTime
  IN2     := B#16#0           //b_ConvertingTimeUnit
  IN3     := P#M4.0 DWORD 3  //ANY-Pointer auf Ergebnis
  RET_VAL := MW 0            //RET_VAL Fehlercode
```

WriteAnalogOutput:

entspricht der Funktion „b_ADDIDATA_Write1AnalogOutput(DWORD dw_DriverHandle, WORD w_Channel, dw_ValueToWrite)“.

Aufruf:

```

CALL      FC 110
  IN0      := W#16#0           //w_Channel
  IN1      := DW#16#0         //dw_ValueToWrite
  RET_VAL  := MW 0            //RET_VAL Fehlercode

```

Rückgabewerte:

Bei Erfolg wird RET_VAL = 0 zurückgegeben.

Trat ein Fehler auf, wird soweit möglich, der Fehlercode der ADDIDATA.DLL durchgereicht.

Sollte dies nicht möglich sein, werden die folgenden Fehlercodes zurückgegeben:

Fehlercode	Ursache
0x807F	DriverHandle nicht vorhanden
0x8042	Fehler beim Lesen eines Eingangs
0x8043	Fehler beim Schreiben eines Ausgangs
0x8003	Falscher Typ des ANY-Pointers
0x8023	ANY-Pointer zeigt auf einen nicht vorhandenen Bereich oder besitzt nicht die richtige Länge

SPS geht in STOP:

Geht die S7-SoftSPS in STOP, so wird im Diagnosepuffer ein entsprechender Fehlercode eingetragen, der mit dem Programmierwerkzeug (S7 für Windows®, STEP®7) ausgelesen werden kann.

Fehlercode	Ursache
0xA140	Bei "i_ADDIDATA_OpenWin32Driver" ist ein Fehler aufgetreten. Die Fehlernummer der ADDIDATA.DLL steht im nächsten Diagnosepuffer-Eintrag

5.3.2 Einstellen des ADDI-DATA-Treibers

Addi-Data Konfiguration

Addi-Data Treiber nur einmalig initialisieren (nicht bei jedem Start/Stop) Standard

Funktion 'b_ADDIDATA_Read8DigitalInputs' entspricht FC: 100

Funktion 'b_ADDIDATA_Read16DigitalInputs' entspricht FC: 101

Funktion 'b_ADDIDATA_Read32DigitalInputs' entspricht FC: 102

Funktion 'b_ADDIDATA_Set8DigitalOutputsOn' entspricht FC: 103

Funktion 'b_ADDIDATA_Set16DigitalOutputsOn' entspricht FC: 104

Funktion 'b_ADDIDATA_Set32DigitalOutputsOn' entspricht FC: 105

Funktion 'b_ADDIDATA_InitAnalogInput' entspricht FC: 106

Funktion 'b_ADDIDATA_ReleaseAnalogInput' entspricht FC: 107

Funktion 'b_ADDIDATA_Init1AnalogOutput' entspricht FC: 108

Funktion 'b_ADDIDATA_Read1AnalogInput' entspricht FC: 109

Funktion 'b_ADDIDATA_Write1AnalogOutput' entspricht FC: 110

Status:
Addidata.dll nicht gefunden (Addi-Pack nicht installiert)

ADDI-DATA Treiber nur einmalig initialisieren (nicht bei jedem Start/Stop):

Den ADDI-DATA Treiber nur beim ersten Start der *S7-SoftSPS* initialisieren, nicht bei jedem RUN/STOP Übergang.

Nachfolgend können den ADDI-DATA-Funktionen FC Nummern zugewiesen werden. (Zu beachten ist, dass für alle FC-Bausteine eindeutige Nummern vergeben werden!)

Funktion "b_ADDIDATA_Funktionsname" entspricht FC:

Nummer des FCs der der ADDIPACK-Funktion "b_ADDIDATA_Funktionsname" entspricht.

Status:

Zeigt den ADDI-DATA-Treiberstatus des Rechners an.

5.3.3 ADDI-DATA Treiber für *IBHsoftec S7-SoftSPS*

Um den Treiber zu aktivieren wird in der Sektion [OEM] der Treiber in der Form dll0=addi_data.dll eingebunden.

Der Treiber erwartet die Sektion [addi_data], in der weitere Parameter definiert werden können.

Name des Parameters	Defaultwert	Erklärung
Read8FC	100	Nummer der FC Read8DigitalInputs
Read16FC	101	Nummer der FC Read16DigitalInputs
Read32FC	102	Nummer der FC Read32DigitalInputs
Set8FC	103	Nummer der FC Set8DigitalOutputs
Set16FC	104	Nummer der FC Set16DigitalOutputs
Set32FC	105	Nummer der FC Set32DigitalOutputs
InitAnaInpFC	106	Nummer der FC InitAnalogInput
RelAnaInpFC	107	Nummer der FC ReleaseAnalogInput
InitAnaOutFC	108	Nummer der FC InitAnalogOutput
ReadAnaInpFC	109	Nummer der FC ReadAnalogInput
WriteAnaOutFC	110	Nummer der FC WriteAnalogOutput
NoStop	0	0 Treiber wird bei jedem SPS-START initialisiert 1 Treiber wird einmalig initialisiert

Zu beachten ist, dass für alle FC-Bausteine eindeutige Nummern vergeben werden!

Beispiel:

```
[OEM]
dll0=addi_data.dll

[addi_data]
Read8FC=100
Read16FC=101
Read32FC=102
Set8FC=103
Set16FC=104
Set32FC=105
InitAnaInpFC=106
RelAnaInpFC=107
InitAnaOutFC=108
ReadAnaInpFC=109
WriteAnaOutFC=110
NoStop=0
```

5.4 Beckhoff TwinCAT I/O

Mittels des Beckhoff TwinCAT I/O Treibers kann von der *S7-SoftSPS* aus auf die von Beckhoff unterstützten Bussysteme zugegriffen werden.

- Erstellen eines Tasks im TwinCAT System Manager
- Zugriff aus dem SPS-Programm auf den Task (Peripherie)
- Treiber mit Hilfe der Oberfläche konfigurieren
- Treiber in der PLC43.INI konfigurieren

5.4.1 Beckhoff TwinCAT I/O Konfiguration

Der vorliegende Treiber ermöglicht es E/A Daten über den Beckhoff Treiber TwinCAT IO mit der IBHsoftec *S7-SoftSPS* auszutauschen. Über den TwinCAT IO Treiber lassen sich alle E/A Baugruppen der Firma Beckhoff ansprechen, so dass dieser Treiber unabhängig vom verwendeten Bussystem ist.

Um mit der *S7-SoftSPS* die Beckhoff Peripheriebaugruppen ansprechen zu können, muss der TwinCAT IO Treiber von Beckhoff auf dem Rechner installiert und lizenziert sein.

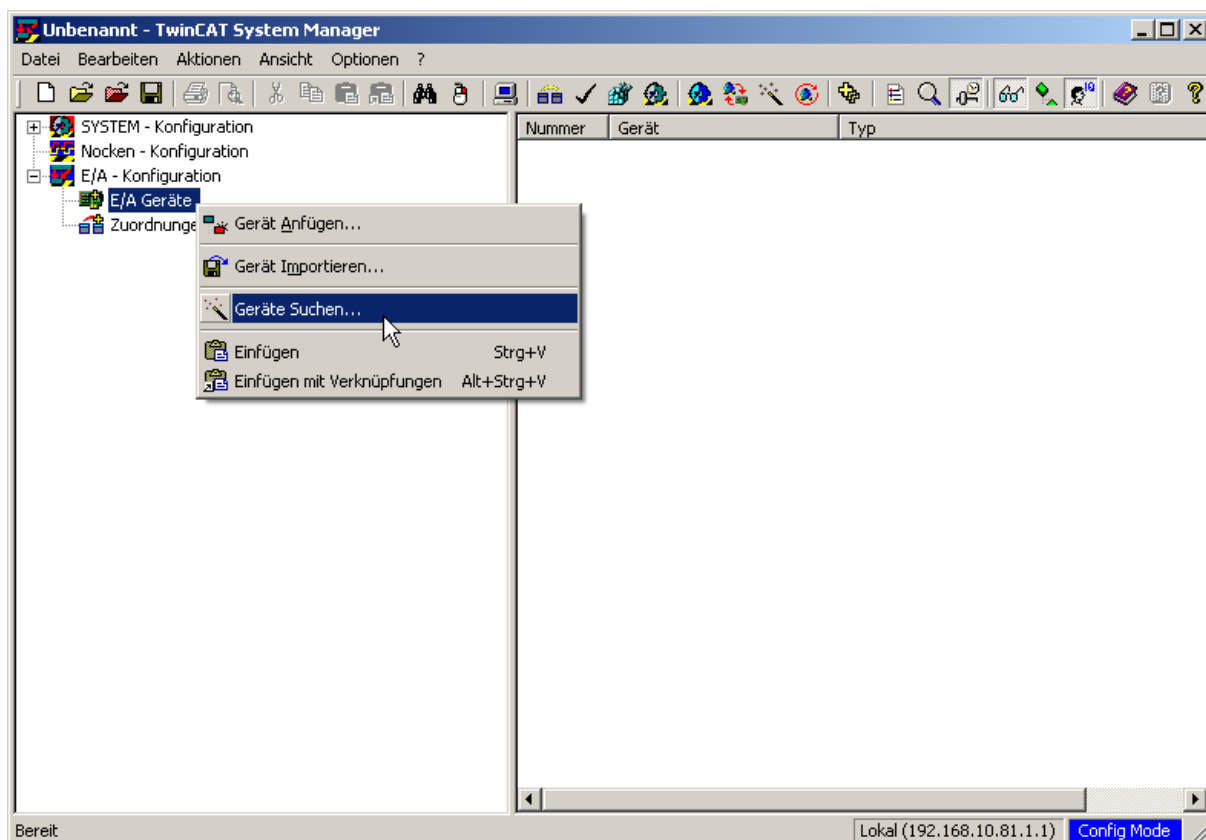
Die IO-Tasks, die mit der *S7-SoftSPS* verwendet werden sollen, müssen mit dem Beckhoff TwinCAT System Manager konfiguriert werden.

Die TwinCatIO.dll kann den E/A-Bereich des PAE/PAA zyklisch auf E/A-Baugruppen lesen/schreiben. Azyklischer Zugriff mittels der Peripheriebefehle ist ebenso möglich wie auch der azyklische Zugriff mit einem SFC/FC.

Um die gewünschten Funktionen in der *S7-SoftSPS* zu aktivieren sind folgende Schritte erforderlich:

- Einfügen einer Station im TwinCAT System Manager
- Anlegen eines Tasks im TwinCAT System Manager
- Aufrufen der Ads-Funktionen aus dem SPS-Programm

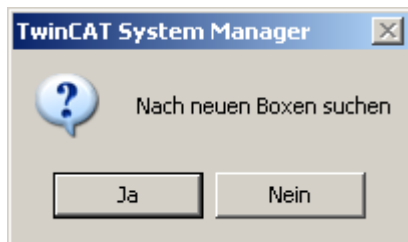
Einfügen einer Station im TwinCAT System Manager



Mittels eines Rechtsklicks auf E/A Geräte >> Geräte Suchen... erscheint ein Dialog mit dem sich eine E/A Baugruppe auswählen lässt (sofern im Netz vorhanden).

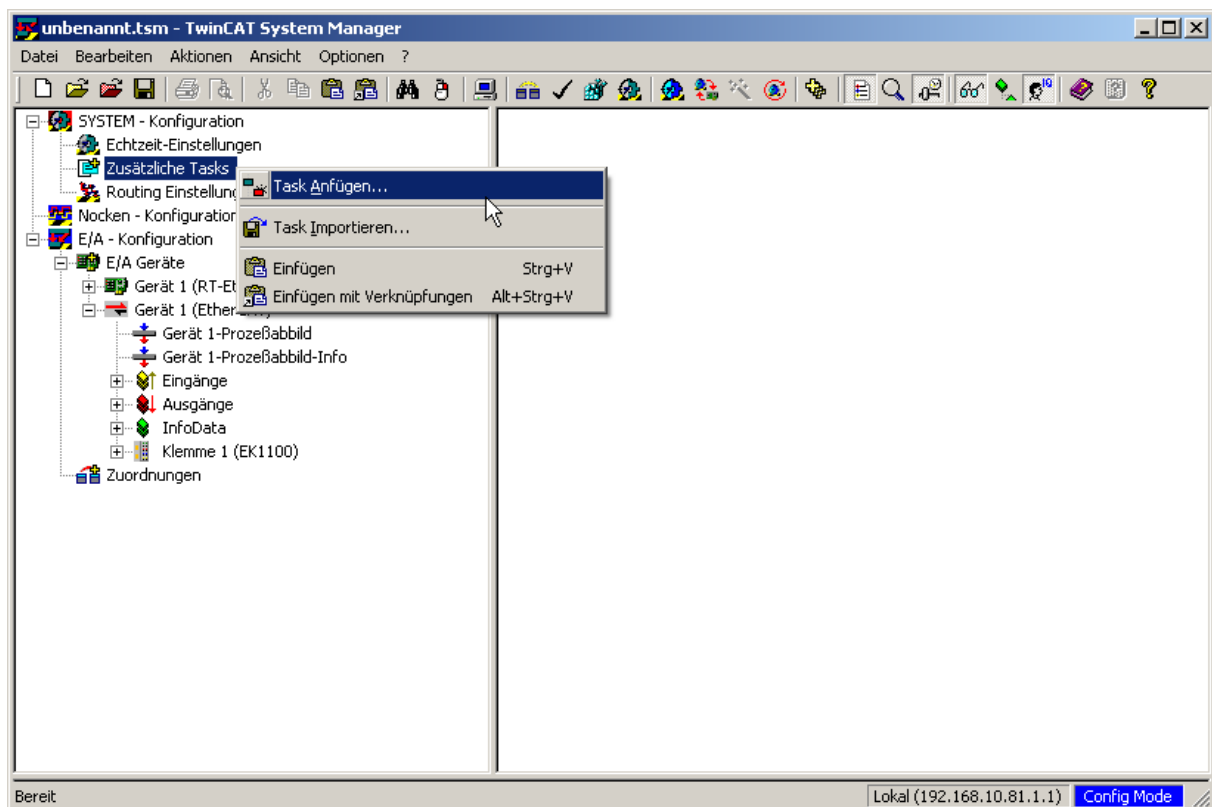


Nachdem man das gewünschte Gerät ausgewählt hat, fragt der TwinCAT System Manager nach ob nach neuen Boxen gesucht werden soll.



Dieser Frage stimmt man zu um die vorhandene Peripherie aus der E/A Baugruppe zu lesen. Die E/A Baugruppe ist nun für den TwinCAT IO Treiber bekannt und kann von diesem angesprochen werden.

Anlegen eines Tasks im TwinCAT System Manager



Mit einem Rechtsklicks auf **Zusätzliche Tasks** >> **Task Anfügen...** erstellt man einen neuen Task, mit dem die TwinCatIO.dll der S7-SoftSPS Daten austauscht.

Wichtige Eigenschaften des Tasks

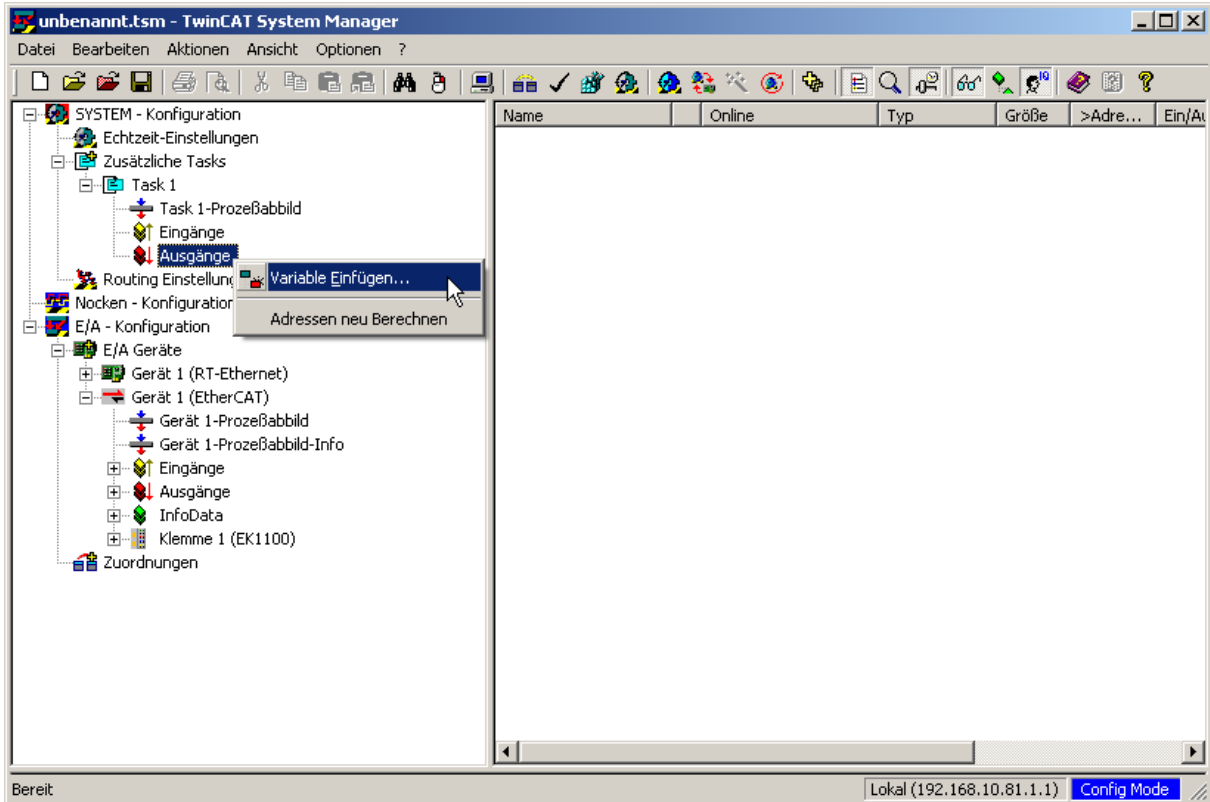
Port: Der Port Nummer 301 dient normalerweise als Prozessabbild für die S7-SoftSPS. Dieser ist auch über einen SFC/FC ansprechbar.

AutoStart: Die S7-SoftSPS kümmert sich selbstständig um das Triggern des Feldbusses und meldet so dem TwinCAT-Treiber das sie noch funktionsbereit ist, so das die Option AutoStart normalerweise nicht gewählt werden sollte.

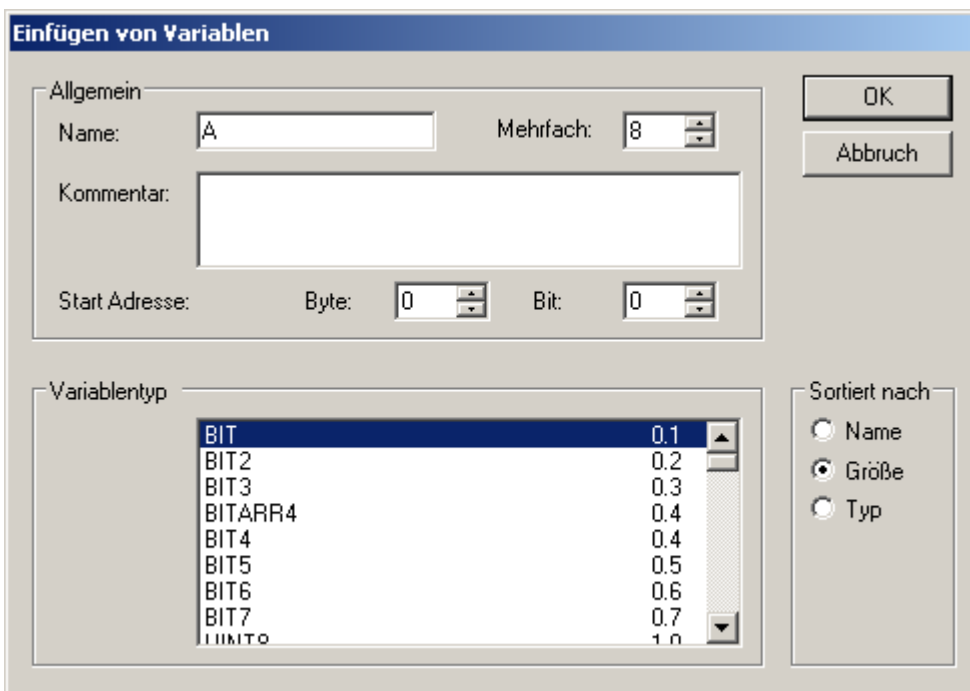
Wenn diese Option gewählt wird muss der TwinCAT IO-Treiber so konfiguriert werden, dass er selbstständig feststellen kann ob die SPS noch arbeitet, um gegebenenfalls die Ausgänge der E/A-Baugruppe in einen definierten Zustand zu schalten.

Konfigurieren des Tasks

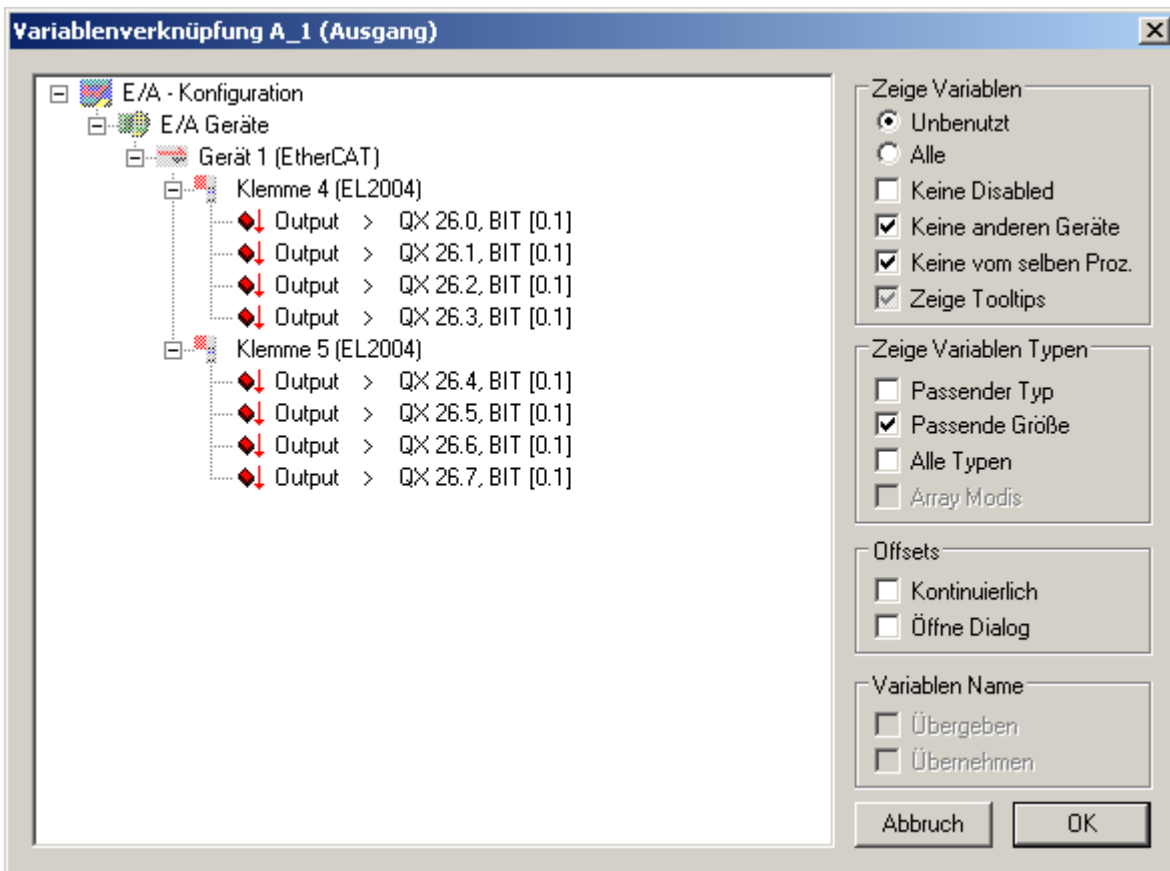
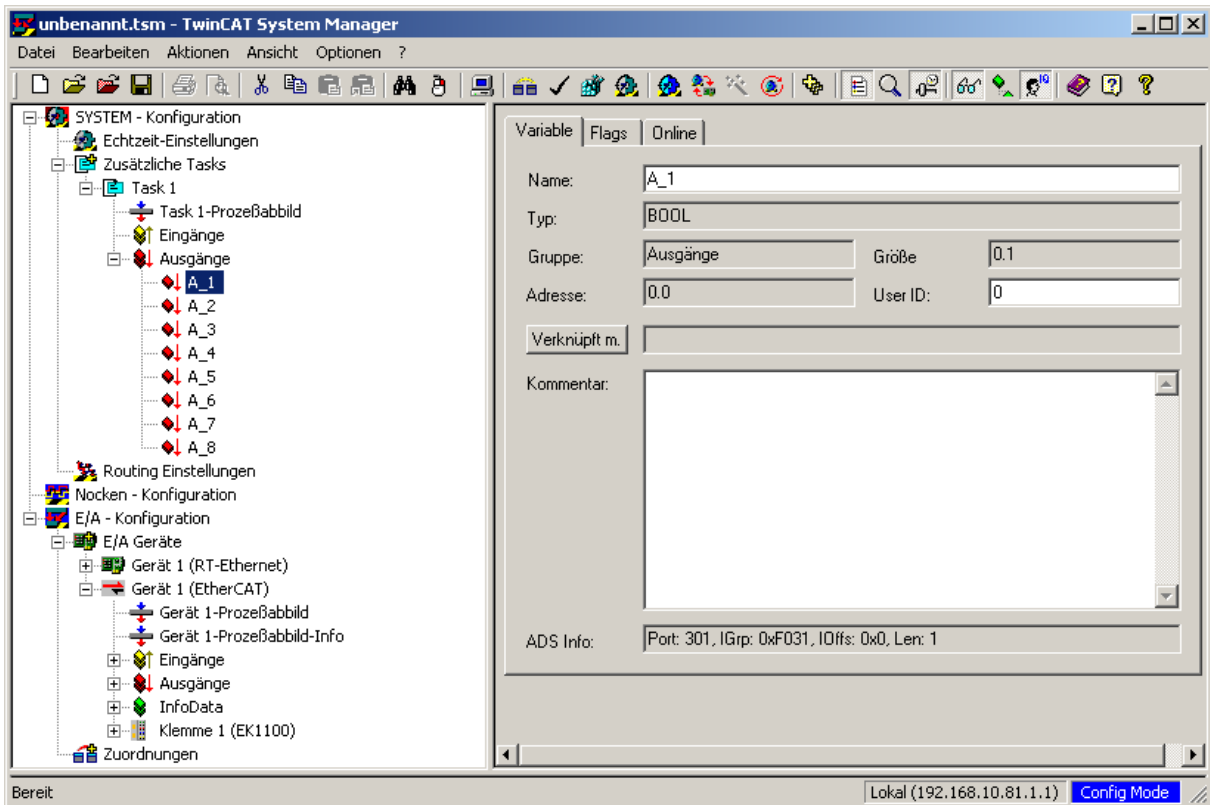
Um die E/A-Baugruppen ansprechen zu können, müssen die E/A-Punkte mit dem Task verknüpft werden. Mittels eines Rechtsklicks auf Eingänge oder Ausgänge >> Variable Einfügen... können die E/A-Punkte dem Task zugeordnet werden.



Je nach vorhandener E/A-Punkt muss der passende Variablentyp ausgewählt werden.



Ist die Variable deklariert muss diese mit der E/A-Baugruppe verknüpft werden. Hierzu betätigt man den Button Verknüpft m. um die Zuordnung zu treffen.



Ist nun die gewünschte Anzahl an Variablen mit dem Task verknüpft, muss die Konfiguration aktiviert werden. Danach ist der TwinCAT IO-Task einsatzbereit, sobald in den Run-Modus gewechselt wurde.

Aufrufen von Sonderfunktionen und ADS-Funktionen aus dem SPS-Programm

SFC 14

Mittels des Bausteins DPRD_DAT können Daten aus dem TwinCAT I/O Task gelesen werden

```
CALL          SFC/FC 14
  LADDR      := W#16#100           //INPUT WORD
  RET_VAL    := MW 0              //RESULT INT
  RECORD     := P#M 100.0 BYTE 1  //OUTPUT ANY
```

Parameter	Datentyp	Bedeutung
LADDR	WORD	Startadresse der zu lesenden Daten
RET_VAL	INT	0 bei Erfolg
RECORD	ANY	ANY-Pointer auf gelesenen Daten

SFC 15

Mittels des Bausteins DPWR_DAT können Daten in den TwinCAT I/O Task geschrieben werden

```
CALL          SFC/FC 15
  LADDR      := W#16#100           //INPUT WORD
  RECORD     := P#M 102.0 BYTE 1  //INPUT ANY
  RET_VAL    := MW 0              //RESULT INT
```

Parameter	Datentyp	Bedeutung
LADDR	WORD	Startadresse der zu schreibenden Daten
RECORD	ANY	ANY-Pointer auf zu schreibende Daten
RET_VAL	INT	0 bei Erfolg

ControlSFC

Mittels des Bausteins CONTROL kann auf jeden beliebigen TwinCAT I/O Task zugegriffen werden

```
CALL          SFC/FC 1000
  IN0        := FALSE             //INPUT BOOL
  IN1        := 301              //INPUT INT
  IN2        := P#M 104.0 BYTE 1 //INPUT ANY
  RET_VAL    := MW 0            //RESULT INT
```

Parameter	Datentyp	Bedeutung
IN0	BOOL	false = Daten aus beliebigem TwinCAT I/O Task lesen true = Daten in beliebigem TwinCAT I/O Task schreiben
IN1	INT	Portnummer des TwinCAT I/O Tasks, auf dessen E/A Bereich zugegriffen werden soll
IN2	ANY	ANY-Pointer auf zu lesende/schreibende Daten mit passender Länge (es sind nur ANY-Pointer vom Typ Byte zulässig)
RET_VAL	INT	0 bei Erfolg, ansonsten werden die Rückgabewerte des TwinCAT I/O Treibers direkt durchgereicht 0x0057= ERROR_INVALID_PARAMETER - einer der übergebenen Parameter ist falsch

GetDeviceIdentification

Der Funktionsbaustein GETDEVID erlaubt es die Geräteidentifikationsdaten auszulesen

```
CALL          SFB/FB 1001, DB 1001
EXEC         := M0.0                //INPUT BOOL
TMOUT        := TIME#0D_0H_0M_1S_0MS //INPUT TIME
BUSY         := M 0.1                //OUTPUT BOOL
ERR          := M 0.2                //OUTPUT BOOL
ERRID        := MD 4                 //OUTPUT DWORD
DEVID        := P#M 10.0 BYTE 214   //IN_OUT ANY
```

Parameter	Datentyp	Bedeutung
IN0	BOOL	TRUE startet das Kommando
IN1	TIME	Nicht benutzt, da die darunterliegende ADS Funktion den Timeout liefert
OUT2	BOOL	Die Daten werden aus dem CX ausgelesen. Nach fehlerfreier Ausführung stehen bei bBusy = FALSE die Daten in der Struktur stDevIdent
OUT3	BOOL	Wird TRUE, sobald ein Fehler eintritt
OUT4	BOOL	Liefert bei einem gesetzten bError-Ausgang die Fehlernummer
IO5	ANY	ST_CxDeviceIdentification (Länge 214 Byte)

AdsRead

Der Funktionsbaustein ADSREAD erlaubt die Ausführung eines ADS-Lesebefehls, um Daten von einem ADS-Gerät anzufordern.

```
CALL          SFB/FB 1002, DB 1002
NETID        := #stringvar          //INPUT STRING
PORT         := W#16#0               //INPUT WORD
IDXGRP       := DW#16#0              //INPUT DWORD
IDXOFFS      := DW#16#0              //INPUT DWORD
LEN          := DW#16#0              //INPUT DWORD
DESTADDR     := P#M 10 BYTE 1        //INPUT ANY
READ         := M0.0                //INPUT BOOL
TMOUT        := TIME#0D_0H_0M_1S_0MS //INPUT TIME
BUSY         := M 0.1                //OUTPUT BOOL
ERR          := M 0.2                //OUTPUT BOOL
ERRID        := MD 4                 //OUTPUT DWORD
```

Parameter	Datentyp	Bedeutung
NETID	STRING	Ist ein String, der die AMS-Netzwerkennung des Zielgerätes enthält, an das der ADS-Befehl gerichtet wird.
PORT	WORD	Enthält die Portnummer des ADS-Gerätes.
IDXGRP	DWORD	Enthält die Index-Gruppennummer (32bit, unsigned) des angeforderten ADS-Dienstes. Dieser Wert ist der ADS-Tabelle des angesprochenen Gerätes zu entnehmen.
IDXOFFS	DWORD	Enthält die Index-Offsetnummer (32bit, unsigned) des angeforderten ADS-Dienstes. Dieser Wert ist der ADS-Tabelle des angesprochenen Gerätes zu entnehmen.
LEN	DWORD	Enthält die Anzahl der zu lesenden Daten in Bytes.

DESTADDR	ANY	Enthält die Adresse des Puffers, der die gelesenen Daten aufnehmen soll. Der Programmierer ist selbst dafür verantwortlich den Puffer in der Größe so zu dimensionieren, dass er LEN Bytes aufnehmen kann. Der Puffer kann eine Einzelvariable, ein Array oder eine Struktur sein, dessen Adresse man mit dem ADR - Operator ermitteln kann.
READ	BOOL	TRUE an diesem Eingang löst den ADS-Befehl aus.
TMOUT	TIME	Gibt die Zeit bis zum Abbrechen der Funktion an.
BUSY	BOOL	Dieser Ausgang bleibt solange auf TRUE, bis der Baustein eine Befehlsanforderung ausführt, längstens aber für die Dauer der, an dem 'Timeout'-Eingang angelegten, Zeit. Während Busy = TRUE wird an den Eingängen kein neuer Befehl angenommen. Bitte beachten Sie, dass nicht die Ausführung des Dienstes, sondern nur dessen Annahme zeitlich überwacht wird.
ERR	BOOL	Dieser Ausgang wird auf TRUE geschaltet, wenn bei der Ausführung eines Befehls ein Fehler aufgetreten ist. Der befehlspezifische Fehlercode ist in 'Errorld' enthalten. Wenn der Baustein ein Timeout-Fehler hat, so ist 'Error' = TRUE und 'Errorld' = 1861 (Hexadezimal 0x745). Wird durch das Ausführen eines Befehls an den Eingängen auf FALSE zurückgesetzt.
ERRID	DWORD	Enthält den befehlspezifischen Fehlercode des zuletzt ausgeführten Befehls. Wird durch das Ausführen eines Befehls an den Eingängen auf 0 zurückgesetzt.

AdsWrite

Der Baustein erlaubt die Ausführung eines ADS-Schreibbefehls, um Daten zu einem ADS-Gerät zu übermitteln.

```
CALL      SFB/FB 1003, DB 1003
NETID    := #stringvar           //INPUT STRING
PORT     := W#16#0              //INPUT WORD
IDXGRP   := DW#16#0            //INPUT DWORD
IDXOFFS  := DW#16#0            //INPUT DWORD
LEN      := DW#16#0            //INPUT DWORD
SRCADDR  := P#M 10.0 BYTE 1     //INPUT ANY
WRITE    := M0.0                //INPUT BOOL
TMOUT    := TIME#0D_0H_0M_0S_0MS //INPUT TIME
BUSY     := M 0.1               //OUTPUT BOOL
ERR      := M 0.2               //OUTPUT BOOL
ERRID    := MD 4                //OUTPUT DWORD
```

Parameter	Datentyp	Bedeutung
NETID	STRING	Ist ein String, der die AMS-Netzwerkennung des Zielgerätes enthält, an das der ADS-Befehl gerichtet wird.
PORT	WORD	Enthält die Portnummer des ADS-Gerätes an das der Befehl gerichtet ist.
IDXGRP	DWORD	Enthält die Index-Gruppennummer (32bit, unsigned) des angeforderten ADS-Dienstes. Dieser Wert ist der ADS-Tabelle des angesprochenen Gerätes zu entnehmen.
IDXOFFS	DWORD	Enthält die Index-Offsetnummer (32bit, unsigned) des angeforderten ADS-Dienstes. Dieser Wert ist der ADS-Tabelle des angesprochenen Gerätes zu entnehmen.
LEN	DWORD	Enthält die Anzahl der zu lesenden Daten in Bytes.
SRCADDR	ANY	Enthält die Adresse des Puffers, aus dem die zu schreibenden Daten geholt werden sollen. Der Programmierer ist selbst dafür verantwortlich, den Puffer in der Größe so zu dimensionieren, dass LEN'- Bytes daraus entnommen werden können. Der Puffer kann eine Einzelvariable, ein Array oder eine Struktur sein, dessen Adresse man mit dem ADR - Operator ermitteln kann.
WRITE	BOOL	TRUE an diesem Eingang löst den ADS-Befehl aus.
TMOUT	TIME	Gibt die Zeit bis zum Abbrechen der Funktion an.
BUSY	BOOL	Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.
ERR	BOOL	Sollte ein ADS-Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.
ERRID	DWORD	Liefert bei einem gesetzten ERR-Ausgang die ADS- oder die Gerätespezifische-Fehlernummer.

5.4.2 TwinCAT I/O Treiber konfigurieren

Allgemein:

Status:

Bus bei SPS Stop nicht stoppen (nur Ausgänge auf 0 setzen)

TwinCAT SFCs als FCs anzeigen SFC 14/15 einblenden

Port Nummer (wie im Twincat System Manager definiert):

Control SFC Nummer (weitere Ports):

FB "GetDeviceIdentification" Nummer:

FB "AdsRead" Nummer:

FB "AdsWrite" Nummer:

Konfiguration aus TwinCAT System Manager Laden:

Vorhandene IO-Tasks:

ID-Task	Taskname

Inhalt der Symboldatei:

Adresse	Symbol	Kommentar

Standard

Stellt die Default-Werte wieder her.

Bus bei SPS Stop nicht stoppen (nur Ausgänge auf 0 setzen)

Anstatt den Feldbus bei SPS STOP komplett anzuhalten, werden alle Ausgänge auf 0 gesetzt.

TwinCAT SFCs als FCs anzeigen

Legt fest, ob Bausteine als FC oder SFC im Bausteinverzeichnis der *S7-SoftSPS* angelegt werden.

SFC 14/15 einblenden

Blendet SFC 14/15 ein.

Port Nummer (wie im TwinCAT System Manager definiert)

Port Nummer des TwinCAT I/O Tasks, normalerweise 301.

Control SFC Nummer (weitere Ports)

Bausteinnummer des Control SFCs mit dem Daten in weitere Zusätzliche Tasks gelesen bzw. geschrieben werden können.

FC "GetDeviceldentification" Nummer

Bausteinnummer des FBs mit dem man die GetDeviceldentification Funktion des TwinCAT IO nutzen kann.

FB "AdsRead" Nummer

Bausteinnummer des FBs mit dem man die AdsRead Funktion des TwinCAT IO nutzen kann.

FB "AdsWrite Nummer

Bausteinnummer des FBs mit dem man die AdsWrite Funktion des TwinCAT IO nutzen kann.

Konfiguration aus dem TwinCAT System Manager Projekt laden

Um Doppeldokumentation zu vermeiden und Symbole wiederverwenden zu können, existiert ein Konverter, der es ermöglicht, Step@7 Symbole aus Symboltabellen (*.SEQ) Dateien in IO-Tasks umzusetzen und auch umgekehrt Symbole aus IO-Tasks in Symboltabellen zu wandeln.

Das Anwenden bestehender Symboltabellen auf existierende Tasks kann nach verschiedenen Kriterien erfolgen.

Symboltabellen werden mit Step@7 aus dem Symboleditor heraus erzeugt.

- Symboltabelle in Symboleditor von Step@7 öffnen.
- Den Menüpunkt 'Tabelle -> Exportieren' wählen.
- Im nachfolgenden Dialog das Format "Zuordnungsliste (*.SEQ)" auswählen.

Symboltabellen werden mit dem Step@7 Symboleditor importiert.

- Symboltabelle in Symboleditor von Step@7 öffnen.
- Den Menüpunkt 'Tabelle -> Importieren' wählen.
- Im nachfolgenden Dialog das Format "Zuordnungsliste (*.SEQ)" auswählen.
- S7 für Windows® Projekte arbeiten generell mit SEQ Dateien. Daher muss hier nur eine Symboldatei gleichen Namens erzeugt werden und mit Mitteln des Windows Explorers ausgetauscht werden.

Vorhandene IO-Tasks

In diesem Fenster werden IO-Tasks der geladenen System Manager Datei (.tsm) angezeigt.

Durch einen Klick auf einen der IO-Tasks wird dieser geladen, so dass er in eine Symboltabelle exportiert werden kann.

System Manager Datei öffnen

Lädt eine System Manager Datei (.tsm) in den Dialog um die vorhandenen Symbole in eine Symboltabelle zu exportieren.

Exportiere in Symboltabelle

Exportiert den geladenen IO-Task in eine Symboltabelle (.seq).

Symboltabelle öffnen

Lädt eine Symboltabelle (.seq) in den Dialog um die darin enthaltenen Symbole in einen IO-Task zu exportieren.

Symboltabelle auf Task anwenden

Exportiert die geladenen Symbole in eine System Manager Datei (.tsm).

Inhalt der Symboldatei

Zeigt den geladenen Symbolinhalt einer System Manager Datei (.tsm) oder einer Symboltabelle (.seq) an.

Verwenden eines Beckhoff EtherCAT®->Profibus Koppler

Die Beckhoff EtherCAT®->Profibus Koppler haben einen festen 100ms-KBus-Watchdog. Ohne diese feste Einstellung würde der Profibus nicht mehr richtig takten. Minimalzyklus ist < 100ms, wenn BKs/KLs verwendet werden.

D.h. es muss mindestens einen Task geben, der schneller ist als 100ms und der muss die höchste Priorität haben, da der Task mit der höchsten Priorität den Profibus triggert.

5.4.3 TwinCAT I/O mit PLC43.INI konfigurieren

Um den Treiber zu aktivieren muss er in der Sektion [OEM] eingebunden werden. Der erforderliche Eintrag lautet zum Beispiel:

```
[OEM]
dll0=TwinCatIO.dll
```

Es gibt dazu in der PLC43.INI folgende Parameter die in der Sektion [TwinCatIO] definiert werden müssen:

Parameter	Defaultwert	Erklärung
PortNumber	301	Eingestellter Port des Tasks im TwinCAT System Manager. Bei PortNumber = 0 werden PEA/PAA und Peripherie Datenaustausch ausgeschaltet und der Zugriff ist nur über SFC/FC möglich. Stimmen die PortNumber der PLC43.INI nicht mit dem Port des TwinCAT IO Tasks überein, so geht die S7-SoftSPS in STOP.
SFC14	1	SFC 14 in Bausteinverzeichnis einblenden
SFC15	1	SFC 15 in Bausteinverzeichnis einblenden
ControlSFC	1000	Nummer des Control SFCs
NoStop	0	Bus bei SPS Stop nicht stoppen
TypeFC	0	SFCs als FCs anzeigen
FB_CxGetDeviceIdentification	1001	Nummer des GetDeviceIdentification FBs
FB_AdsRead	1002	Nummer des AdsRead FBs
FB_AdsWrite	1003	Nummer des AdsWrite FBs

Beispiel:

```
[OEM]
dll0=TwinCatIO.dll

[TWINCATIO]
PortNumber=301
SFC14=1
SFC15=1
ControlSFC=1000
NoStop=0
TypeFC=0
FB_CxGetDeviceIdentification=1001
FB_AdsRead=1002
FB_AdsWrite=1003
```

5.5 CIF 30/50/60 Konfiguration

Konfiguration von Hilscher CIF 30/50/60-Karten.

- Grundlegende Konfiguration
- Profibus Konfiguration mit *S7 für Windows®/ Step® 7* vornehmen

5.5.1 Grundlegende Konfiguration

Hier werden die grundlegenden Konfigurationseigenschaften der CIF-Karten eingestellt.

Wenn kein Profibus verwendet wird so muss die Konfiguration mit dem Systemkonfigurator der Firma Hilscher durchgeführt werden:

- Konfiguration der CIF Karte mit dem Systemkonfigurator

Wie der Treiber in die *S7-SoftSPS* eingebunden wird ist Bestandteil dieser Anleitungen:

- Konfiguration via Oberfläche
- Konfiguration in PLC43.INI

Verwenden der Status und Control SFC

Abfrage CIF-Status

Zur Abfrage des CIF Status muss regelmäßig der Status SFC aufgerufen werden (z.B. aus dem OB 1). Der SFC besitzt folgende Parameter :

Name	Art	Typ	Funktion
HostFlags	OUT	BYTE	HostFlags
DevFlags	OUT	BYTE	DeviceFlags

```
CALL    SFC 252
      OUT0 :=MB10           // HostFlags
      OUT1 :=MB11           // DeviceFlags
```

Die HostFlags und DeviceFlags sind in der Sektion Steuer-SFC Funktion 141 und Funktion 142, wie auch im Hilscher Dokument tke.pdf auf Seite 8 genau beschrieben.

Steuer-SFC

Zum Ausführen von speziellen Funktionen der CIF-Karte dient der Steuer-Funktionsbaustein.

Der Steuerbaustein wird folgendermaßen aufgerufen:

```
L      #Wert
T      MD 20
CALL   SFC 210
      IN0 := 142           // INPUT Funktionsnummer.
      IN1 := 0             // INPUT Datenbausteinnummer (bei DB Operationen,
```

```
sonst 0).
```

```
IO2 := MD 20 // IN_OUT Option / Parameter der Funktion / Ergebnis.
OUT3 := MW 24 // OUTPUT Standard 80xx Fehlercodes, 0 bei Erfolg.
```

Bei Erfolg ist ebenfalls das BIE Bit gesetzt.

Übersicht der Steuer-SFC Funktionen

Fkt-Nr.	Funktionsbeschreibung
102	Anzahl der digitalen Ein/Ausgänge einstellen
108	Anzahl der analogen Ein/Ausgänge einstellen
125	Prozesseingänge lesen
126	Prozessausgänge schreiben
130	Lese globale Diagnose-Bits
131	Lese Busstatus
132	Lese fehlerhafte Busadresse
133	Lese Fehlernummer
134	Lese Status-Bits
135	Lese Diagnose-Bits
136	Dualportmemory lesen
137	Dualportmemory schreiben
140	Watchdog triggern
141	HostFlags lesen
142	DevFlags lesen
150	DevGetTaskState aufrufen und Ergebnis im DB liefern
151	DevGetInfo aufrufen und Ergebnis im DB liefern
152	DevGetTaskParameter aufrufen und Ergebnis im DB liefern

Fkt 102 Anzahl der digitalen Ein-/Ausgänge einstellen

Mit dieser Funktion können Sie die Anzahl der digitalen Ein-/Ausgänge einstellen. Voreingestellt sind bei der SoftSPS 512 Byte digitale Ein-/Ausgänge. Sie können mit dieser Funktion die Anzahl der digitalen Ein- und Ausgänge auf die erforderliche Anzahl reduzieren. Dies bewirkt eine Beschleunigung der SoftSPS.

Falls die Funktion nicht erfolgreich abgeschlossen wurde, ist das BIE nicht gesetzt. Der Rückgabewert enthält die Standard 80xx Fehlercodes.

Fehlercode:

0x8222 = Anzahl zu groß

Beispiel:

```
L      50 // Es sind nur 50 digitale E/As erforderlich
T      MD20 // Anzahl der digitalen E/As einstellen
CALL   SFC 210
      IN0 := 102 // Funktion 102
      IN1 := 0 // Kein DB
      IO2 := MD 20 // Anzahl
      OUT3 := MW 24 // Fehlercode
```

Fkt 108 Anzahl der analogen Ein-/Ausgänge einstellen

Mit dieser Funktion können Sie die Anzahl der analogen Ein-/Ausgänge einstellen. Voreingestellt sind bei der SoftSPS die beim Initialisieren der Karte gefundene maximale Größe des Dualport RAMs. Sie können mit dieser Funktion die Anzahl der analogen Ein- und Ausgänge auf die erforderliche Anzahl reduzieren. Gezählt wird ab dem Byte 0 des Dualports. Falls die Funktion nicht erfolgreich abgeschlossen wurde, ist das BIE nicht gesetzt. Der Rückgabewert enthält die Standard 80xx Fehlercodes.

Fehlercode:

0x8222 = Anzahl zu groß

Beispiel:

```
L      100          // Es sind nur 100 digitale E/As erforderlich
T      MD20        // Anzahl der digitalen E/As einstellen
CALL   SFC 210
      IN0 := 108    // Funktion 108
      IN1 := 0      // Kein DB
      IO2 := MD 20  // Anzahl
      OUT3 := MW 24 // Fehlercode
```

Fkt 125 Prozesseingänge lesen

Mit dieser Funktion können die Prozesseingänge in einen Datenbaustein geschrieben werden. Hiermit können alle verfügbaren Peripheriebytes verwaltet werden.

Diese Funktion schreibt asynchron. Bei wortorientierten Eingängen besteht keine Datenkonsistenz und darf daher nur mit byteorientierten Eingängen benutzt werden.

Beispiel:

```
L      W#16#0100   // Anfangsadresse der Prozesseingänge
T      MD20
CALL   SFC 210
      IN0 := 125    // Funktion 125
      IN1 := 300    // DB 300
      IO2 := MD 20  // Anfangsadresse der Prozesseingänge
      OUT3 := MW 24 // Fehlercode
```

Nach erfolgreichem Abschluss der Funktion steht im Fehlercode eine 0 und das BIE ist gesetzt. Falls die Funktion nicht erfolgreich abgeschlossen wurde, ist das BIE nicht gesetzt.

Fehlercodes:

0x813A = Datenbaustein nicht vorhanden

0x8222 = Datenbausteingröße größer als die Quellbereichsgröße

0x8042 = READY oder COM fehlt

0x807F = Interner Treiberzugriffsfehler

Fkt 126 Prozessausgänge schreiben

Mit dieser Funktion kann ein Datenbaustein in die Prozessausgänge geschrieben werden. Hiermit können alle verfügbaren Peripheriebytes verwaltet werden.

Diese Funktion schreibt asynchron: Bei wortorientierten Ausgängen besteht keine Datenkonsistenz und die Funktion darf daher nur byteorientiert benutzt werden.

Beispiel:

```
L      W#16#0100      // Anfangsadresse im Prozessausgangsbereich
T      MD20
CALL   SFC 210
      IN0 := 126      // Funktion 126
      IN1 := 300      // DB 300
      IO2 := MD 20    // Anfangsadresse im Prozessausgangsbereich
      OUT3 := MW 24   // Fehlercode
```

Nach erfolgreichem Abschluss der Funktion steht im Fehlercode eine 0 und das BIE ist gesetzt. Falls die Funktion nicht erfolgreich abgeschlossen wurde, ist das BIE nicht gesetzt.

Fehlercodes:

0x813A = Datenbaustein nicht vorhanden

0x8223 = Datenbausteingröße größer als die Quellbereichsgröße

0x8043 = READY oder COM fehlt

0x807F = Interner Treiberzugriffsfehler

Fkt 130 Lese globale Diagnosebits

Mit dieser Funktion können die globalen Diagnosebits ausgelesen werden.

Bit	Profibusmeldung
0	Parameterfehler
1	Baugruppe wegen Busfehler in Auto_Clear_Modus verzweigt
2	Mindestens ein Slave ist nicht in der Datentransferphase oder meldet schweren Fehler
3	Schwerer Busfehler, es ist kein weiterer Datentransfer möglich
4	Kurzschluss auf dem Bus
5	Host ist nicht bereit
6	reserviert
7	reserviert

Bit	Interbusmeldung
0	Parameterfehler
1	Systemfehler aufgetreten
2	mindestens ein Slave meldet Modulfehler
3	mindestens eine defekte W1-Schnittstelle
4	mindestens eine defekte W2-Schnittstelle
5	reserviert
6	reserviert
7	reserviert

Beispiel:

```
CALL    SFC 210
      IN0 := 130          // Funktion 130
      IN1 := 0           // Kein DB
      IO2 := MD 20       // Diagnosebits
      OUT3 := MW 24      // Fehlercode
```

Nach erfolgreichem Abschluss der Funktion steht im Fehlercode eine 0 und das BIE ist gesetzt. Falls die Funktion nicht erfolgreich abgeschlossen wurde, ist das BIE nicht gesetzt.

Fehlercodes:

0x807F = Interner Treiberzugriffsfehler

Fkt 131 Lese Busstatus

Mit dieser Funktion kann der Busstatus ausgelesen werden.

Aufbau für CIF-Interbusbus-Karte:

Bit	Interbusmeldung
0	Bussegmente sind abgeschaltet
1	Reserviert
2	Reserviert
3	Reserviert
4	Reserviert
5	Reserviert
6	Reserviert
7	Reserviert

Beispiel:

```
CALL    SFC 210
      IN0 := 131          // Funktion 131
      IN1 := 0           // Kein DB
      IO2 := MD 20       // Busstatus
      OUT3 := MW 24      // Fehlercode
```

Nach erfolgreichem Abschluss der Funktion steht im Fehlercode eine 0 und das BIE ist gesetzt. Falls die Funktion nicht erfolgreich abgeschlossen wurde, ist das BIE nicht gesetzt.

Fehlercodes:

0x807F = Interner Treiberzugriffsfehler

Fkt 132 Lese fehlerhafte Busadresse

Mit dieser Funktion kann die fehlerhafte Busadresse ausgelesen werden.

Beispiel:

```
CALL    SFC 210
      IN0  := 132          // Funktion 132
      IN1  := 0           // Kein DB
      IO2  := MD 20       // Fehlerhafte Busadresse
      OUT3 := MW 24       // Fehlercode
```

Nach erfolgreichem Abschluss der Funktion steht im Fehlercode eine 0 und das BIE ist gesetzt. Falls die Funktion nicht erfolgreich abgeschlossen wurde, ist das BIE nicht gesetzt.

Fehlercodes:

0x807F = Interner Treiberzugriffsfehler.

Fkt 133 Lese Fehlernummer

Mit dieser Funktion kann die Fehlernummer ausgelesen werden.

Beispiel:

```
CALL    SFC 210
      IN0  := 133          // Funktion 133
      IN1  := 0           // Kein DB
      IO2  := MD 20       // Fehlernummer
      OUT3 := MW 24       // Fehlercode
```

Nach erfolgreichem Abschluss der Funktion steht im Fehlercode eine 0 und das BIE ist gesetzt. Falls die Funktion nicht erfolgreich abgeschlossen wurde, ist das BIE nicht gesetzt.

Fehlercodes:

0x807F = Interner Treiberzugriffsfehler

Fkt 134 Lese Statusbits

Mit dieser Funktion können die Statusbits ausgelesen werden. Vor dem Aufruf wird die Bytenummer (0..15) eingetragen. Nach dem Aufruf der Funktion wird das angewählten Statusbyte zurückgeliefert. Wenn ein Bit im Statusbyte 1 ist, ist der entsprechende Slave aktiv, bei 0 inaktiv.

Beispiel:

```
L      W#16#03      // Byte Nummer 3
T      MD20
CALL   SFC 210
      IN0 := 134      // Funktion 134
      IN1 := 0        // Kein DB
      IO2 := MD 20    // Vor Aufruf Statusbytenummer, nach Aufruf der
Inhalt des Statusbytes.
      OUT3 := MW 24   // Fehlercode
```

Nach erfolgreichem Abschluss der Funktion steht im Fehlercode eine 0 und das BIE ist gesetzt. Falls die Funktion nicht erfolgreich abgeschlossen wurde, ist das BIE nicht gesetzt.

Fehlercodes:

0x807F = Interner Treiberzugriffsfehler

Fkt 135 Lese Diagnosebits

Mit dieser Funktion können die Diagnosebits ausgelesen werden. Vor dem Aufruf wird die Bytenummer (0..15) eingetragen. Nach dem Aufruf der Funktion wird das angewählten Diagnosebyte zurückgeliefert. Wenn ein Bit im Diagnosebyte 1 ist, hat sich der Zustand des entsprechenden Slaves geändert (von aktiv auf inaktiv oder umgekehrt).

Beispiel:

```
L      W#16#03      // Byte Nummer 3
T      MD 20
CALL   SFC 210
      IN0 := 134      // Funktion 134
      IN1 := 0        // Kein DB
      IO2 := MD 20    // Vor Aufruf Diagnosebytenummer, nach Aufruf der
Inhalt des Diagnosebytes.
      OUT3 := MW 24   // Fehlercode
```

Nach erfolgreichem Abschluss der Funktion steht im Fehlercode eine 0 und das BIE ist gesetzt. Falls die Funktion nicht erfolgreich abgeschlossen wurde, ist das BIE nicht gesetzt.

Fehlercodes:

0x807F = Interner Treiberzugriffsfehler

Fkt 136 Dualportmemory lesen

Mit dieser Funktion kann ein Bereich aus dem Dualportmemory der CIF-Karte in einen Datenbaustein geschrieben werden. Es wird grundsätzlich mit dem 1. Byte im Datenbaustein begonnen. Die Länge des Bereiches steht im IO-Parameter-L und die relative Adresse im Dualportmemory im IO-Parameter-H.

Beispiel:

```

L      DW#16#0100000A // 10 Bytes von der relativen Adresse 100H kopieren
T      MD20
CALL   SFC 210
      IN0 := 136          // Funktion 136
      IN1 := 60           // DB 60
      IO2 := MD 20        // Es werden 10 Bytes von der relativen Adresse
0x100H gelesen.
      OUT3 := MW 24       // Fehlercode

```

Nach erfolgreichem Abschluss der Funktion steht im Fehlercode eine 0 und das BIE ist gesetzt. Falls die Funktion nicht erfolgreich abgeschlossen wurde, ist das BIE nicht gesetzt.

Fehlercodes:

0x813A = Datenbaustein nicht vorhanden

0x8222 = Datenbaustein zu klein.

0x8224 = Bereichsüberschreitung

0x8042 = READY oder COM fehlt

0x807F = Interner Treiberzugriffsfehler

Fkt 137 Dualportmemory schreiben

Mit dieser Funktion kann ein Bereich aus einem Datenbaustein in das Dualportmemory der CIF-Karte geschrieben werden. Es wird grundsätzlich mit dem 1. Byte im Datenbaustein begonnen. Die Länge des Bereiches steht im IO-Parameter-L und die relative Adresse im Dualportmemory im IO-Parameter-H.

Beispiel:

```

L      D#16#0100000A // 10 Bytes auf die relative Adresse 100H kopieren
T      MD20
CALL   SFC 210
      IN0 := 137          // Funktion 137
      IN1 := 60           // DB 60
      IO2 := MD 20        // Es werden 10 Bytes auf die relative Adresse
0x100H geschrieben.
      OUT3 := MW 24       // Fehlercode

```

Nach erfolgreichem Abschluss der Funktion steht im Fehlercode eine 0 und das BIE ist gesetzt. Falls die Funktion nicht erfolgreich abgeschlossen wurde, ist das BIE nicht gesetzt.

Fehlercodes:

0x813A = Datenbaustein nicht vorhanden

0x8223 = Datenbaustein zu klein

0x8225 = Bereichsüberschreitung

0x8043 = READY oder COM fehlt

0x807F = Interner Treiberzugriffsfehler

Fkt 140 Watchdog triggern

Diese Funktion ist erforderlich, wenn sehr lange SPS Zyklen vorkommen können.

Beispiel:

```
CALL SFC 210
  IN0 := 140      // Funktion 140
  IN1 := 0        // Kein DB
  IO2 := MD 20   // Kein Funktionscode
  OUT3 := MW 24  // Fehlercode
```

Nach Abschluss der Funktion steht im Fehlercode immer eine 0 und das BIE ist gesetzt.

Fehlercodes:

keine

Fkt 141 Hostflags lesen

Mit dieser Funktion können die Hostflags ausgelesen werden.

Aufbau :

Bit	Hostflag - Meldung
0	HostCom: Auftragsbit für HostMailbox
1	DevAck: Quittungsbit für DevMailbox
2	PdAck: E/A-Datensynchronisationsbit Host
3	reserviert
4	reserviert
5	Com: Nutzdatentransferphase für mindestens ein Modul aktiv
6	Run: Parameterdaten gültig, CIF zur Kommunikation bereit
7	Ready: Gerät läuft, kein Basis-Initialisierungsfehler

Beispiel:

```
CALL SFC 210
  IN0 := 141      // Funktion 141
  IN1 := 0        // Kein DB
  IO2 := MD 20   // Gibt die Host-Flags zurück
  OUT3 := MW 24  // Fehlercode
```

Nach erfolgreichem Abschluss der Funktion steht im Fehlercode eine 0 und das BIE ist gesetzt. Falls die Funktion nicht erfolgreich abgeschlossen wurde, ist das BIE nicht gesetzt.

Fehlercodes:

0x807F = Interner Treiberzugriffsfehler.

Fkt 142 DevFlags lesen

Mit dieser Funktion können die DevFlags ausgelesen werden.

Aufbau :

Bit	Deviceflag - Meldung
0	HostAck: Quittungsbit für HostMailbox
1	DevCom: Auftragsbit für DevMailbox
2	PdCom: E/A-Datensynchronisationsbit Dev
3	reserviert
4	reserviert
5	NotReady: Busbetrieb anhalten (Module im Reset) oder freigeben
6	Init: Zurücksetzen des CIF mit Übernahme der Parameter aus dem DPM
7	Reset: Zurücksetzen des CIF

Beispiel:

```
CALL SFC 210
  IN0 := 142      // Funktion 142
  IN1 := 0        // Kein DB
  IO2 := MD 20   // Gibt die Dev-Flags zurück
  OUT3 := MW 24  // Fehlercode
```

Nach erfolgreichem Abschluss der Funktion steht im Fehlercode eine 0 und das BIE ist gesetzt. Falls die Funktion nicht erfolgreich abgeschlossen wurde, ist das BIE nicht gesetzt.

Fehlercodes:

0x807F = Interner Treiberzugriffsfehler

Fkt 150 DevGetTaskState ausführen

Mit dieser Funktion können die globalen Diagnoseinformationen, die vom CIF Modul gemeldet werden, ausgelesen werden. Die Struktur beinhaltet Informationen über den Buszustand und den Zustand einzelner Slaves. Die Informationen sind im Wesentlichen in Bitfeldern abgelegt. So stehen Informationen über den Master und den Zustand des Feldbusses zur Verfügung. Des Weiteren kann abgelesen werden, welche Slaves konfiguriert sind und bedient werden, welche Slaves aktiv und inaktiv sind und welche Slaves Diagnoseinformationen beinhalten.

Intern wird die Treiberfunktion DevGetTaskState angestoßen. Das Resultat kann in einem DB hinterlegt werden. Um die Funktion verwenden zu können, sollte unbedingt die Dokumentation zur Funktion dem Hilscher Manual dpm_pie.pdf entnommen werden. Der DB sollte der dort beschriebenen DPM_DIAGNOSTICS Struktur entsprechen.

Beispiel:

```
L      DW#16#2
T      MD20 // Task 2
```

```
CALL    SFC 210
      IN0  := 150           // Funktion 150
      IN1  := 300           // DB 300
      IO2  := MD 20         // Task Nummer.
      OUT3 := MW 24         // Fehlercode
```

Nach erfolgreichem Abschluss der Funktion steht im Fehlercode eine 0 und das BIE ist gesetzt. Falls die Funktion nicht erfolgreich abgeschlossen wurde, ist das BIE nicht gesetzt.

Fehlercodes:

0x813A = Datenbaustein nicht vorhanden

0x8222 = Datenbaustein zu klein.

0x8225 = Bereichsüberschreitung.

0x807F = Interner Treiberzugriffsfehler.

Passender Beispiel DB:

```
DATA_BLOCK DB 300
TITLE=DIAG
VERSION : 0.1
STRUCT
  bGlobalBits : BYTE;
  bDPM_state : BYTE;
  tError : STRUCT
    bErr_Rem_Adr : BYTE;
    bErr_Event : BYTE;
  END_STRUCT;
  usBus_Error_Cnt : WORD;
  usTime_Out_Cnt : WORD;
  abReserved : ARRAY[0..7] OF BYTE;
  abSl_cfg : ARRAY[0..15] OF BYTE;
  abSl_state : ARRAY[0..15] OF BYTE;
  abSl_diag : ARRAY[0..15] OF BYTE;
END_STRUCT;
BEGIN
END_DATA_BLOCK
```

Fkt 151 DevGetInfo ausführen

Mit DevGetInfo können verschiedene Informationen aus dem Dualport der CIF Baugruppe ausgelesen werden. Dies sind:

DW#16#1 GET_DRIVER_INFO Treiber Statusinformationen

DW#16#2 GET_VERSION_INFO Versionsinformation der CIF

DW#16#3 GET_FIRMWARE_INFO Version der CIF Firmware

DW#16#4 GET_TASK_INFO Task Informationen

DW#16#5 GET_RCS_INFO CIF spezifische Betriebssysteminformationen

DW#16#6 GET_DEV_INFO Geräteinformationen

DW#16#7 GET_IO_INFO Geräte E/A Informationen

DW#16#8 GET_IO_SEND_DATA Geräte E/A Sendedaten

Intern wird die Treiberfunktion DevGetInfo angestossen. Das Resultat kann in einem DB hinterlegt werden. Um die Funktion verwenden zu können, sollte unbedingt die Dokumentation zur Funktion dem Hilscher Manual devdrv.pdf entnommen werden. Der DB sollte den dort beschriebenen Strukturen entsprechen.

Beispiel:

```
L    DW#16#1          // Funktion GET_DRIVER_INFO
T    MD20
CALL SFC 210
    IN0 := 151        // Funktion 151
    IN1 := 300        // DB 300
    IO2 := MD 20      // Funktionsnummer
    OUT3 := MW 24     // Fehlercode
```

Nach erfolgreichem Abschluss der Funktion steht im Fehlercode eine 0 und das BIE ist gesetzt. Falls die Funktion nicht erfolgreich abgeschlossen wurde, ist das BIE nicht gesetzt.

Fehlercodes:

0x813A = Datenbaustein nicht vorhanden

0x8222 = Datenbaustein zu klein.

0x8225 = Funktionsnummer nicht zulässig

0x807F = Interner Treiberzugriffsfehler

Fkt 152 DevGetTaskParameter ausführen

Mit dieser Funktion können die globalen Einstellungen wie beispielsweise die Übertragungsrate des Busses, die Watchdogzeit usw. aus der CIF ausgelesen werden.

Intern wird die Treiberfunktion DevGetTaskParameter angestoßen. Das Resultat kann in einem DB hinterlegt werden. Um die Funktion verwenden zu können, sollte unbedingt die Dokumentation zur Funktion dem Hilscher Manual devdrv.pdf entnommen werden. Der DB sollte der dort beschriebenen Struktur entsprechen.

Beispiel:

```

L      DW#16#2      // Task 2
T      MD20
CALL   SFC 210
      IN0 := 152      // Funktion 152
      IN1 := 300      // DB 300
      IO2 := MD 20    // Funktionsnummer
      OUT3 := MW 24   // Fehlercode

```

Fkt 153 Slavediagnose eines spezifischen Slaves lesen

Mit dieser Funktion können die Normdiagnosedaten eines DP-Slaves ausgelesen werden. Die Informationen werden im Format DPM_SLAVE_SINGLE_DIAGNOSTICS zurückgeliefert. Diese Funktion darf nicht verwendet werden, wenn gleichzeitig die Datei cifcfg.dll zur Konfiguration und Diagnose des Profibusses verwendet wird!

In diesem Falle ist die über den Treiber cifcfg.dll bereitgestellte SFC 13 (wie Hardware SPS) zu verwenden. Details können dem Hilscher Manual dpm_pie.pdf, Sektion DDLM_Slave_Diag entnommen werden. Der DB sollte der dort beschriebenen Struktur entsprechen.

Beispiel:

```

L DW#16#2B          // Profibusadresse des Slaves (hier : 11dez)
T MD20
CALL   SFC 210
      IN0 := 153      // Funktion 153
      IN1 := 300      // DB 300
      IO2 := MD 20    // Funktionsnummer
      OUT3 := MW 24   // Fehlercode

```

Passender Beispiel DB

```

DATA_BLOCK DB 300
TITLE=
VERSION : 0.1
STRUCT
  DPM_SLAVE_DIAG : STRUCT
    Stationsstatus_1 : BYTE;
    Stationstatus_2 : BYTE;
    Stationstatus_3 : BYTE;
    bMaster_Add : BYTE; // corresponding master address
    usIdent_Number : WORD; // ident number, motorola format
    abExt_Diag_Data : ARRAY[0..99] OF BYTE; // extended diagnostic field
  END_STRUCT;
END_STRUCT;
BEGIN
END_DATA_BLOCK

```

Nach erfolgreichem Abschluss der Funktion steht im Fehlercode eine 0 und das BIE ist gesetzt. Falls die Funktion nicht erfolgreich abgeschlossen wurde, ist das BIE nicht gesetzt.

Fehlercodes:

0x813A = Datenbaustein nicht vorhanden

0x8222 = Datenbaustein zu klein

0x8225 = Bereichsüberschreitung

0x807F = Interner Treiberzugriffsfehler

5.5.2 Konfiguration der CIF Karte mit dem Systemkonfigurator

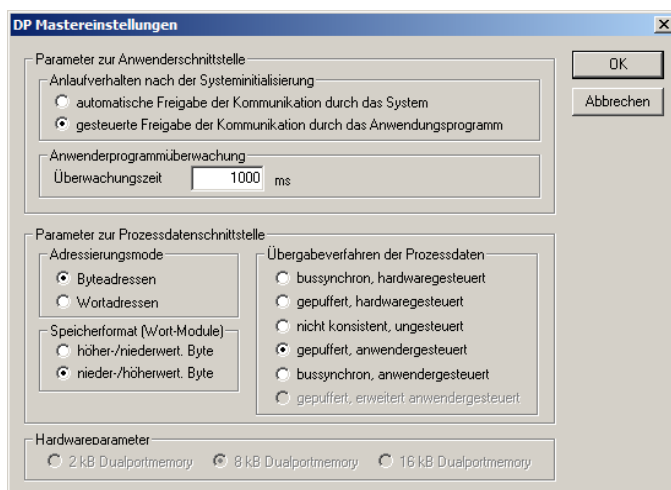
Bevor die Feldbuskomponenten mit der SoftSPS angesprochen werden können, müssen diese konfiguriert werden. Wenn kein Profibus verwendet oder aus anderen Gründen die Konfiguration nicht mit dem S7 für Windows® Hardwarekonfigurator oder dem Step@7 Hardwarekonfigurator erzeugen werden kann, so muss die CIF Konfiguration mit dem Systemkonfigurator der Firma Hilscher erstellt werden. Falls die voreingestellten Startadressen nicht über die Sondereinstellungen verändert wurden, gelten die nachfolgenden Zuordnungen.

Die Adressen der digitalen Peripherie werden in den CIF-Adressbereich von 0..511 gelegt.

Außerdem müssen im Systemkonfigurator unter Setup unbedingt folgende Systemparameter eingestellt

werden:

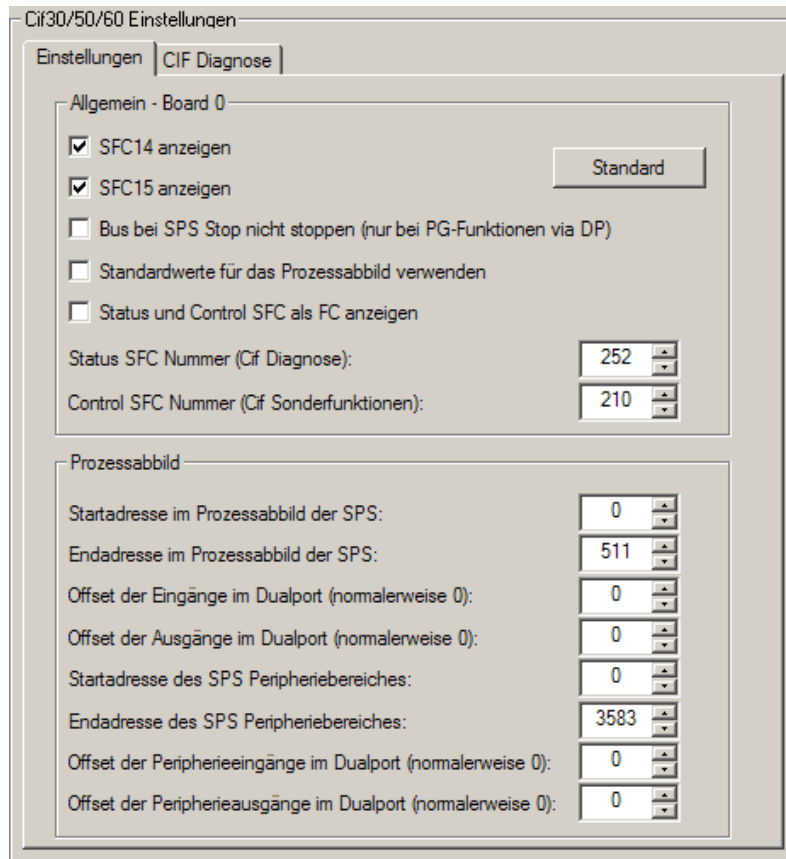
- Gesteuerte Freigabe der Kommunikation durch das Anwenderprogramm
- Byteadressen
- Intel-Format
- Gepuffert, anwendergesteuert



5.5.3 CIF 30/50/60 Einstellungen

Der Dialog besteht aus zwei Reitern die den Treiber für die Hilscher CIF Karten konfigurieren.

Einstellungen



In diesem Fenster werden die spezifischen Einstellungen der Hilscher CIF-Karten (ISA, PCI, PCMCIA) vorgenommen. Sind in dem PC mehr als eine Hilscher Karte eingebaut, so müssen die entsprechenden Treiber zusätzlich in die S7-SoftSPS aufgenommen werden. Zusätzlich muss darauf geachtet werden, dass sich die Bereiche im Prozessabbild der SPS nicht überschneiden. Dies bedeutet, die Einstellungen für BitStart, BitEnd usw. des vorherigen CIF Treibers müssen mit äußerster Sorgfalt getroffen werden.

Standard

Setzt alle Einstellungen auf Standardwerte zurück.

SFC14 anzeigen

Der SFC14 wird im Bausteinverzeichnis eingeblendet.

SFC15 anzeigen

Der SFC15 wird im Bausteinverzeichnis eingeblendet.

ACHTUNG: BEIM BETRIEB MEHRERER CIF-KARTEN DÜRFEN NUR DIE SFCs 14 UND 15 EINER CIF-KARTE EINGEBLENDET WERDEN!

Bus bei SPS Stop nicht stoppen (nur bei PG-Funktionen via DP)

Verhindert, dass der Bus bei SPS Stop durch den Watchdog gestoppt wird. Die Ausgänge werden programmiert auf 0 gesetzt. Die Hilscher Karte wird zudem nur beim Start der SPS oder bei Änderungen der Systemdaten initialisiert. Bei Run/Stop wird keine Initialisierung mehr vorgenommen. Werden Onlinefunktionen über Profibus mit "S7 für Windows®" oder Step® 7 vorgenommen, muss diese Option aktiviert werden.

Standardwerte für das Prozessabbild verwenden

Endadresse im Prozessorabbild der SPS: 511

Endadresse des SPS Peripheriebereiches: 8192

Alle anderen Werte: 0

Status und Control SFC als FC anzeigen

Legt fest, ob der StatusSFC und der ControlSFC als SFC oder als FC im Bausteinverzeichnis der SPS erscheinen. Dies hat den Vorteil, dass man im SPS Programm leere Bausteine gleichen Namens erstellen kann und hier den Parametern sinnvolle Namen geben kann. Beim Erstellen der Bausteinköpfe ist darauf zu achten, dass die Datentypen und die IN/OUT Bereiche identisch mit den Bausteinen der SPS sind. Werden die Bausteine (FC) mit dem PG zur SPS übertragen, behält die SPS die internen Bausteine bei.

Status SFC Nummer (CIF Diagnose)

Legt die Nummer des Status SFC fest.

Control SFC Nummer (CIF Sonderfunktionen)

Legt die Nummer des Control SFC fest.

Prozessabbild

Um die Werte für das Prozessabbild manuell festlegen zu können, muss die Option Standardwerte für das Prozessabbild verwenden deaktiviert werden.

Diagnose

Cif30/50/60 Einstellungen

Einstellungen Cif Diagnose

CIF Informationen für Board 0

Typ:

Firmware:

Status:

Detailinformationen:

HostFlags	DevFlags:	Globale Statusbits:
<input type="checkbox"/> HostCom	<input type="checkbox"/> HostAck	<input type="checkbox"/> Ctrl
<input type="checkbox"/> DevAck	<input type="checkbox"/> DevCom	<input type="checkbox"/> ACIr
<input type="checkbox"/> PdAck	<input type="checkbox"/> PdCom	<input type="checkbox"/> NonExch
<input type="checkbox"/> ERR	<input type="checkbox"/> Reserviert	<input type="checkbox"/> Fatal
<input type="checkbox"/> PdExtAck	<input type="checkbox"/> PdExtCom	<input type="checkbox"/> Event
<input type="checkbox"/> COM	<input type="checkbox"/> NotRdy	<input type="checkbox"/> NRdy
<input type="checkbox"/> RUN	<input type="checkbox"/> Init	<input type="checkbox"/> Tout
<input type="checkbox"/> READY	<input type="checkbox"/> Reset	<input type="checkbox"/> Reserviert

CIF Informationen für Board n

Zeigt Typ, Firmwareversion und Status der gewählten Hilscher Karte an.

Detailinformationen

Siehe Dokumentation der entsprechenden Hilscher Karte.

5.5.4 CIF 30/50/60 Einstellungen der *IBHsoftec S7-SoftSPS*

In der Sektion [CIF30D0] werden die spezifischen Einstellungen der ersten CIF-Karte (ISA, PCI, PCMCIA) vorgenommen.

Keyname	Wertebereich	Defaultwert	Beschreibung
BoardNumber	0 .. 3	0	Vom Hilscher „CIF Device Driver Setup“ gemeldete Boardnummer; 0 = Board 0, 1 = Board 1, usw...
StatusSFC	100 .. 255	252	Legt die Nummer des Status SFC fest
ControlSFC	100 .. 255	210	Legt die Nummer des Steuer SFC fest
TypeFC	0 / 1	0 (SFC)	Legt fest, ob der StatusSFC und der SteuerSFC als SFC oder als FC im Bausteinverzeichnis der SPS erscheinen. Mit TypeFC=1 erscheinen die Bausteine als FC. Dies hat den Vorteil, dass man im SPS Programm leere Bausteine gleichen Namens erstellen kann und hier den Parametern sinnvolle Namen geben kann. Beim Erstellen der Bausteinköpfe ist darauf zu achten, dass die Datentypen und die IN/OUT Bereiche identisch mit den Bausteinen der SPS sind. Werden die Bausteine (FC) mit dem PG zur SPS übertragen, behält die SPS die internen Bausteine bei.
BitStart	0 .. 511	0	Startadresse des zu kopierenden Prozessabbilds
BitEnd	0 .. 511	511	Endadresse des zu kopierenden Prozessabbilds
BitIn	0 .. maximale Dualportgröße	0	Offset der Eingänge im Dualport.
BitOut	0 .. maximale Dualportgröße	0	Offset der Ausgänge im Dualport.
PhStart	0 .. maximale Dualportgröße	BitStart	Startadresse des Peripheriebereichs
PhEnd	0 .. maximale Dualportgröße	Festgestellte maximale Dualportgröße	Endadresse des Peripheriebereichs
PhIn	0 .. maximale Dualportgröße	BitIn	Offset der Peripherieeingänge im Dualport
PhOut	0 .. maximale Dualportgröße	BitOut	Offset der Peripherieausgänge im Dualport
SFC14	0 / 1	1	1 = SFC14 über diesen Treiber einblenden Diagnosefähigkeit = Allgemeine Fehler
SFC15	0 / 1	1	1 = SFC15 über diesen Treiber einblenden Diagnosefähigkeit = Allgemeine Fehler
NoStop	0 / 1	0	1 = Verhindert, dass der Bus bei SPS Stop durch den Watchdog gestoppt wird. Die Ausgänge werden programmiert auf 0 gesetzt. Initialisiert die Hilscher Karte zudem nur beim Start der SPS oder bei Änderungen der Systemdaten. Bei Run/Stop wird keine Initialisierung mehr vorgenommen. Muss bei Verwendung der Treibers Cifcfig.dll in dessen Sektion ebenfalls eingeschaltet werden.

Betrieb mehrer Karten

Sind in dem PC mehr als eine Hilscher Karte eingebaut, so müssen die entsprechenden Treiber zusätzlich

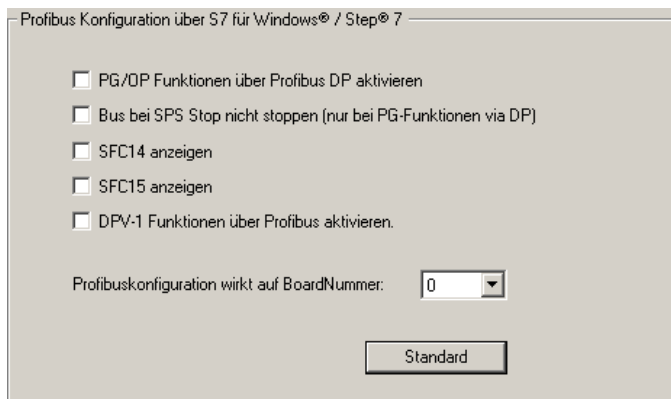
in der Datei PLC43.ini aufgenommen werden. Zusätzlich muss darauf geachtet werden, dass sich die Bereiche im Prozessabbild der SPS nicht überschneiden. Dies bedeutet, die Einstellungen für BitStart, BitEnd usw. der vorherigen Sektion müssen mit äußerster Sorgfalt getroffen werden. Auch dürfen die SFCs 14 und 15 nur aus einer DLL eingeblendet werden. Es können maximal vier Hilscher Karten eingebunden werden.

5.5.5 Profibus Konfiguration via S7 für Windows®/Step® 7

Hier werden Profibus spezifische Einstellungen vorgenommen. Die Einstellungen sind erforderlich, wenn die Buskonfiguration mit dem *S7 für Windows®* Hardwarekonfigurator oder dem Step® 7 Hardwarekonfigurator erzeugt werden soll.

- Konfiguration via Oberfläche
- Konfiguration in PLC43.INI

5.5.6 Profibus Konfiguration über S7 für Windows® / Step® 7



PG/OP Funktionen über Profibus DP aktivieren

Ermöglicht eine PG-Kommunikation über den Profibus zur *S7-SoftSPS*. Einige OPs (wie z.B. OP3) verwenden auch PG-Kommunikation zum Datenaustausch mit der SPS, so dass diese Panels ebenfalls verwendet werden können. *

Bus bei SPS Stop nicht stoppen (nur bei PG-Funktionen via DP)

Verhindert, dass der Bus bei SPS Stop durch den Watchdog gestoppt wird. Die Ausgänge werden programmiert auf 0 gesetzt. Die Hilscher Karte wird zudem nur beim Start der SPS oder bei Änderungen der Systemdaten initialisiert. Bei Run/Stop wird keine Initialisierung mehr vorgenommen. Werden Onlinefunktionen über Profibus mit "S7 für Windows®" oder Step® 7 vorgenommen, muss diese Option aktiviert werden.

SFC14 anzeigen

Der SFC14 wird im Bausteinverzeichnis eingeblendet.

SFC15 anzeigen

Der SFC15 wird im Bausteinverzeichnis eingeblendet.

DPV-1 Funktionen über Profibus aktivieren

Ermöglicht das Senden und Empfangen von DVP1 Diensten zu DPV1 fähigen Slaves. Es stehen die SFB 52 und SFB 53 für DPV1 Dienste zur Verfügung. *

Profibuskonfiguration wirkt auf BoardNumber:

Die mit dem PG erstellten Profibus-Systemdaten werden auf die eingestellte CIF-Karte angewendet.

Standard

Stellt die Einstellungen auf Standardwerte zurück.

* Die Buskonfiguration muss mit dem S7 für Windows® Hardwarekonfigurator oder dem Step®7 Hardwarekonfigurator erzeugt werden.

5.5.7 Profibus Konfiguration über S7 für Windows®/Step® 7

In der Sektion [CIFCFG] werden Profibus spezifische Einstellungen vorgenommen. Die Einstellungen sind erforderlich, wenn die Buskonfiguration mit dem S7 für Windows® Hardwarekonfigurator oder den Step®7 Hardwarekonfigurator erzeugt werden soll.

Keyname	Wertebereich	Defaultwert	Beschreibung
BoardNumber	0 .. 3	0	Vom Hilscher Treiber gemeldete Boardnummer 0 = Board 0, 1 = Board 1, usw...
OnlineDP	0 / 1	0	1 = Ermöglicht eine PG-Kommunikation über den Profibus zur S7-SoftSPS. Einige OP's (wie z.B. OP3) verwenden auch PG-Kommunikation zum Datenaustausch mit der SPS, so dass diese Panels ebenfalls verwendet werden können. *
SFC14	0 / 1	0	1 = SFC14 über diesen Treiber einblenden Diagnosefähigkeit = Jeder Slave (Detaildiagnose). *
SFC15	0 / 1	0	1 = SFC15 über diesen Treiber einblenden iagnosefähigkeit = Jeder Slave (Detaildiagnose). *
DPV1	0 / 1	0	1 = Ermöglicht das Senden und Empfangen von DVP1 Diensten zu DPV1 fähigen Slaves. Es stehen die SFB 52 und SFB 53 für DPV1 Dienste zur Verfügung. *
NoStop	0 / 1	0	1 = Verhindert, dass der Bus bei SPS Stop durch den Watchdog gestoppt wird. Die Ausgänge werden programmiert auf 0 gesetzt. Initialisiert die Hilscher Karte zudem nur beim Start der SPS oder bei Änderungen der Systemdaten. Bei Run/Stop wird keine Initialisierung mehr vorgenommen. Die Aktivierung dieses Parameters verhindert zudem, dass eine laufende PG-Verbindung beim Stoppen der SPS unterbrochen wird. Muss bei Verwendung der Treibers Cif30d(n).dll in dessen Sektion ebenfalls eingeschaltet werden.

* Die Buskonfiguration muss mit dem S7 für Windows® Hardwarekonfigurator oder dem Step®7 Hardwarekonfigurator erzeugt werden.

5.6 CP343/CP443 Emulation

Hier werden die Einstellungen für die CP343/CP443 Emulation vorgenommen

- Allgemein
- Konfiguration über die Oberfläche
- Konfiguration in der PLC43.INI

5.6.1 CP343/CP443 Emulation konfigurieren

Die CP-Funktionalität der S7-SoftSPS unterstützt das ISO on TCP (RFC1006; Port 102) Protokoll für eingehende PG Verbindungen. Send/Receive Funktionen werden über die Protokolle ISO on TCP, TCP/IP und UDP unterstützt. Über ISO on TCP und Standard TCP/IP wird auch Fetch/Write passiv unterstützt.

In der Hardwarekonfiguration sollte bei einer SoftSPS PLC S7-416 auf Seiten der CPU der CP443 mit der Bestellnummer 6GK7 443-1EX11-0XE0 gewählt werden, da dieser ausschließlich das TCP/IP Protokoll unterstützt.

Das ISO Protokoll, das beispielsweise beim 6GK7 443-1EX02-0XE0 verwendet wird, würde spezielle ISO Netzwerkkarten Treiber erforderlich machen. Daher kann eine Verbindung über das ISO Protokoll nicht aufgebaut werden, sondern ausschließlich über ISO on TCP.

Die Anwahl der Netzwerkkarte der PG Station muss mit „TCP/IP -> Ihre Netzwerkkarte“ beginnen.

Funktionsweise

Beim Start der S7-SoftSPS wird nach einer Konfiguration für die CP343/CP443 Emulation gesucht. Hier können die zur Kommunikation relevanten Bausteine ausgewählt und in das Bausteinverzeichnis der SPS eingeblendet werden.

Folgende Protokolle werden unterstützt:

Iso-On-TCP	PG-Kanal	
Iso-On-TCP	Send/Receive	Passiv
Iso-On-TCP	Send/Receive	Aktiv
Iso-On-TCP	Fetch/Write	Passiv
TCP	Send/Receive	Passiv
TCP	Send/Receive	Aktiv
TCP	Fetch/Write	Passiv
UDP	Send/Receive	

Unterschiede zur Hardware SPS

- Die lokale IP-Adresse wird aus der Konfiguration des Windows-Rechners übernommen und wird nicht durch die Systemdaten überschrieben.
- FC5 und FC50 verhalten sich beide wie FC50 (AG_LSEND)
- FC6 und FC60 verhalten sich beide wie FC60 (AG_LRECV)
- FC10 unterstützt nur die Kommandos NOP, CN_STATUS, CN_STATUS_ALL und CN_RESET_ALL.
- FB55 zur Konfiguration via DB darf nur im OB100/OB101 aufgerufen werden.

Modbus on TCP Server

Die CP-Emulation unterstützt per Standardeinstellungen das Modbus on TCP Protokoll.

Folgende Modbus Funktionen werden unterstützt:

Funktion	Datenbereich	Erklärung
0x01	Out	Bits lesen
0x02	In	Bits lesen
0x03	Out	Worte lesen
0x04	In	Worte lesen
0x05	Out	Bit schreiben
0x06	Out	Wort schreiben
0x0F	Out	Bits schreiben
0x10	Out	Worte schreiben

Die Modbus Telegrammtypen die auf den Eingangsbereich zugreifen, werden von einem vom Benutzer frei definierbaren Datenbaustein gelesen.

Die Modbus Telegrammtypen die auf den Ausgangsbereich zugreifen, werden von einem weiteren, vom Benutzer frei definierbaren Datenbaustein gelesen bzw. geschrieben.

Damit besteht die Möglichkeit, mit allen Visualisierungen oder Control Panels die das Modbus on TCP Protokoll unterstützen, auf Prozessdaten der *S7-SoftSPS* zuzugreifen.

5.6.2 CP343/CP443 Emulation über die Oberfläche konfigurieren

CP343/CP443 Emulation konfigurieren:

CP343/443 Emulation mit SoftSPS starten FB55 (IP_CONFIG) verwenden

FC5 (AG_SEND) verwenden FC6 (AG_RECV) verwenden

FC7 (AG_LOCK) verwenden FC8 (AG_UNLOCK) verwenden

FC50 (AG_LSEND) verwenden FC60 (AG_LRECV) verwenden

FC10 (AG_CNTRL) verwenden FC62 (C_CNTRL) verwenden

TCP Acknowledges schicken (nur manche CPs)

Modbus Server

Modbus Server starten Port: 502

DBIn ist Nummer: 501 DBOut ist Nummer: 502

Standard

Hier werden die grundlegenden Eigenschaften der CP343/CP443 Emulation vorgenommen.

CP343/443 Emulation mit S7-SoftSPS starten

Startet die CP-Emulation automatisch wenn die *S7-SoftSPS* startet.

FB55 (IP_CONFIG) verwenden

Blendet den FB55 ins Bausteinverzeichnis der *S7-SoftSPS* ein.

FC5 (AG_SEND) verwenden

Blendet den FC5 ins Bausteinverzeichnis der *S7-SoftSPS* ein.

FC6 (AG_RECV) verwenden

Blendet den FC6 ins Bausteinverzeichnis der *S7-SoftSPS* ein.

FC7 (AG_LOCK) verwenden

Blendet den FC7 ins Bausteinverzeichnis der *S7-SoftSPS* ein.

FC8 (AG_UNLOCK) verwenden

Blendet den FC8 ins Bausteinverzeichnis der *S7-SoftSPS* ein.

FC10 (AG_CNTRL) verwenden

Blendet den FC10 ins Bausteinverzeichnis der *S7-SoftSPS* ein.

FC50 (AG_LSEND) verwenden

Blendet den FC50 ins Bausteinverzeichnis der *S7-SoftSPS* ein.

FC60 (AG_LRECV) verwenden

Blendet den FC60 ins Bausteinverzeichnis der *S7-SoftSPS* ein.

FC62 (C_CNTRL) verwenden

Blendet den FC62 ins Bausteinverzeichnis der *S7-SoftSPS* ein.

TCP Acknowledges schicken (nur manche CPs)

Schickt TCP Acknowledges.

Modbus Server starten

Startet den Modbus Server beim Start des CP343/443.

Port

Weist dem Modbus Server den angegebenen Port zu. Per Default hören Modbus Server auf Port 502.

Modbus DBIn ist Nummer

Datenbausteinnummer des In-Bereiches.

Modbus DBOut ist Nummer

Datenbausteinnummer des Out-Bereiches.

Standard

Setzt die Einstellungswerte zurück auf die Defaultwerte

5.6.3 CP343/CP443 Emulation in PLC43.INI konfigurieren

Hier werden die Grundlegenden Eigenschaften der CP343/CP443 Emulation vorgenommen. Die S7-SoftSPS erwartet eine Sektion mit Namen [CPX43] in der die folgenden Schlüssel Verwendung finden:

KeyName	Defaultwert	Beschreibung
StartWithPlc	1	CP343/443 Emulation mit S7-SoftSPS starten
FB55	0	FB55(IP_CONFIG) verwenden
FC5	0	FC5 (AG_SEND) verwenden
FC6	0	FC6 (AG_RECV) verwenden
FC7	0	FC7 (AG_LOCK) verwenden
FC8	0	FC8 (AG_UNLOCK) verwenden
FC10	0	FC10 (AG_CNTRL) verwenden
FC50	0	FC50 (AG_LSEND) verwenden
FC60	0	FC60 (AG_LRECV) verwenden
FC62	0	FC62 (C_CNTRL) verwenden
SendAcknowledge	0	TCP Acknowledges schicken (nur manche CPs)
ModbusPort	502	Port des Modbus TCP Servers Wenn =0 wird der Modbus Server nicht gestartet.
ModbusDBIn	501	Datenbausteinnummer des In-Bereiches
ModbusDBOut	502	Datenbausteinnummer des Out-Bereiches

Beispiel:

Startet die Emulation mit der SPS und nutzt FCs 5,6,7 und 8 sowie den Modbus Server.

```
[CPX43]
StartWithPlc=1
SendAcknowledge=0
FB55=0
FC5=1
FC6=1
FC7=1
FC8=1
FC10=0
FC50=0
FC60=0
FC62=0
ModbusPort=502
ModbusDBIn=501
ModbusDBOut=502
```

5.7 I/O Port Link

Hier werden Hardwareports des Rechners die in das PAE/PAA abgebildet werden sollen, eingestellt

- Grundlegendes
- Konfiguration der I/O Ports über Oberfläche
- Konfiguration der I/O Ports in PLC43.INI

5.7.1 PC Hardwareports konfigurieren

Hier können Bytes, Worte und Doppelworte der Prozessabbilder Hardware I/O Adressen zugeordnet werden. Es können nur Hardwareadressen im ersten 64k Segment des PCs angesprochen werden (ISA-Bus). Das Prozessabbild der Eingänge wird unmittelbar vor dem OB1 mit den Werten aus den Eingabeports aktualisiert. Am Ende des OB1, werden die Werte des PAA in die dafür konfigurierten Ausgabeports geschrieben. Zusätzlich können auch Peripherieeingänge und Peripherieausgänge zu Ports zugeordnet werden. Diese werden dann nur bei Verwendung dieser Adressen im SPS Programm aktualisiert.

5.7.2 PC Hardwareports mit Oberfläche einstellen

Konfiguration

Ist die I/O Port Konfiguration aktiviert, wird in der Konfigurationsdatei PLC43.INI der S7-SoftSPS nach allen theoretisch möglichen I/O Adressen gesucht. Da dies im maximalen Falle bis zu 45 Sekunden dauern kann, sollte man die Bereiche auf das Nötige einschränken. Die Ausgänge können sowohl mit Initialwerten vorbelegt werden, als auch mit sicheren Werten, die bei SPS Stop vom Treiber eingestellt werden. Es können sowohl Bytes, Worte als auch Doppelworte konfiguriert werden. Um Doppelbelegungen zu vermeiden, hat der zuerst angelegte (bzw. der zuerst in der Konfigurationsdatei PLC43.ini gefundene) Eintrag Vorrang. Wird also beispielsweise ein EW4 gefunden, so wird danach nicht mehr nach dem EB5 gesucht, sondern nach dem EB6. Dies gilt auch für Doppelworte. Überlappungen zwischen Eingängen, Peripherieeingängen, Ausgängen und Peripherieausgängen sind jedoch möglich, da je nach verwendeter Hardware Eingänge und

Ausgänge auf der gleichen I/O Adresse liegen können.

ACHTUNG: BEACHTEN SIE DASS BEI FALSCHER KONFIGURATION WERTE IN BAUGRUPPEN GESCHRIEBEN WERDEN KÖNNEN, DIE DAFÜR NICHT GEEIGNET SIND UND HIERDURCH SCHÄDEN AN DER PC HARDWARE ENTSTEHEN KÖNNEN. BEACHTEN SIE BITTE EBENFALLS, DASS BEIM START DER S7-SOFTSPS DER HARDWAREPORTZUGRIFF BEIM BETRIEBSSYSTEM ANGEMELDET WIRD. DAHER MÜSSEN DIE PORTS ZWINGEND IN EINEM ZUSAMMENHÄNGENDEN, FREIEN BEREICH LIEGEN UND NICHT MIT EXISTIERENDER HARDWARE IN KONFLIKT STEHEN.

5.7.3 PC Hardwareports mit PLC43.INI einstellen

Installation

Die Treiber DLL ioport.dll muss im Installationsverzeichnis der *S7-SoftSPS* vorhanden sein. In der Datei PLC43.INI, Sektion [OEM], in der die Treiber der *S7-SoftSPS* angegeben werden, muss dieser aufgeführt werden.

```
[OEM]
dll10=ioport.dll
```

Konfiguration

Die Datei PLC43.INI wird nach allen theoretisch möglichen I/O Adressen der *S7-SoftSPS* durchsucht. Da dies maximal bis zu 45 Sekunden dauern kann, sollte man die Bereiche auf das Nötige einschränken. Die Ausgänge können sowohl mit Initialwerten vorbelegt werden, als auch mit sicheren Werten, die bei SPS Stop vom Treiber eingestellt werden. Es können sowohl Bytes, Worte als auch Doppelworte konfiguriert werden. Um Doppelbelegungen zu vermeiden, hat der zuerst gefundene Eintrag Vorrang. Wird also beispielsweise ein EW4 gefunden, so wird danach nicht mehr nach dem EB5 gesucht, sondern nach dem EB6. Dies gilt auch für Doppelworte. Überlappungen zwischen Eingängen, Peripherieeingängen, Ausgängen und Peripherieausgängen sind jedoch möglich, da je nach verwendeter Hardware Eingänge und Ausgänge auf der gleichen I/O Adresse liegen können. Alle Eingaben sind dezimal vorzunehmen. Beachten Sie dass bei falscher Konfiguration Werte in Baugruppen geschrieben werden können, die dafür nicht geeignet sind und hierdurch Schäden an der PC Hardware entstehen können. Beachten Sie bitte ebenfalls, dass beim Start der *S7-SoftSPS* der Hardwareportzugriff beim Betriebssystem angemeldet wird. Daher müssen die Ports zwingend in einem zusammenhängenden, freien Bereich liegen und nicht mit existierender Hardware in Konflikt stehen.

Einträge

In der Konfigurationsdatei plc43.ini muss die folgende Sektion enthalten sein:

```
[ioportdll]
;Folgende Einträge sind möglich (n = laufende Nummer):
;Ausgänge mit Initialwerten vorbelegen. Default: 0=Nein
InitOutputs=0

;Ausgänge bei Stop mit sicheren Werten belegen. Default: 0=Nein
SafeOutputs=0

;Sollen Ausgänge nur bei Änderung in die Hardware geschrieben werden.
Default: 0=Nein
BufferOutputs=0

;Zum Verkürzen der Einlesezeit und des Umkopierens in den Prozessabbildern
den Bereich
;der Einträge die für Ports gesucht werden, einschränken.
;Bereich Eingang/Ausgangsbyte von ... Default=0
MinByte=0
;... bis Eingang/Ausgangsbyte. Default=4095
MaxByte=10
```

```
;Konfiguration der Eingänge (Syntax: Eingabeadresse=I/O Portnummer):
EBn=Portnummer(Dez)
EWn=Portnummer(Dez)
EDn=Portnummer(Dez)

;Konfiguration der Peripherieeingänge (Syntax: Eingabeadresse=I/O
Portnummer):
PEBn=Portnummer(Dez)
PEWn=Portnummer(Dez)
PEDn=Portnummer(Dez)

;Konfiguration der Ausgänge (Syntax: Ausgabeadresse=I/O Portnummer):
ABn=Portnummer(Dez)
AWn=Portnummer(Dez)
ADn=Portnummer(Dez)

;I/O Ports mit Werten initialisieren (Bei InitOutputs=1)
InitABn=Wert(Dez)
InitAWn=Wert(Dez)
InitADn=Wert(Dez)

;I/O Ports mit Werten bei SPS Stop belegen (Bei SafeOutputs=1)
SafeABn=Wert(Dez)
SafeAWn=Wert(Dez)
SafeADn=Wert(Dez)

;Konfiguration der Peripherieausgänge (Syntax: Peripherieausgabeadresse=I/O
Portnummer):
PABn=Portnummer(Dez)
PAWn=Portnummer(Dez)
PADn=Portnummer(Dez)

;I/O Ports mit Werten initialisieren (Bei InitOutputs=1)
InitPABn=Wert(Dez)
InitPAWn=Wert(Dez)
InitPADn=Wert(Dez)
```

```
;I/O Ports mit Werten bei SPS Stop belegen (Bei SafeOutputs=1)
SafePABn=Wert(Dez)
SafePAWn=Wert(Dez)
SafePADn=Wert(Dez)
```

ACHTUNG: BEACHTEN SIE DASS BEI FALSCHER KONFIGURATION WERTE IN BAUGRUPPEN GESCHRIEBEN WERDEN KÖNNEN, DIE DAFÜR NICHT GEEIGNET SIND UND HIERDURCH SCHÄDEN AN DER PC HARDWARE ENTSTEHEN KÖNNEN. BEACHTEN SIE BITTE EBENFALLS, DASS BEIM START DER *S7-SOFTSPS* DER HARDWAREPORTZUGRIFF BEIM BETRIEBSSYSTEM ANGEMELDET WIRD. DAHER MÜSSEN DIE PORTS ZWINGEND IN EINEM ZUSAMMENHÄNGENDEN, FREIEN BEREICH LIEGEN UND NICHT MIT EXISTIERENDER HARDWARE IN KONFLIKT STEHEN.

Beispiel

Um eine mögliche Konfiguration darzustellen, wird der Druckerport eines PC dem Prozessabbild der SPS zugeordnet. Der Ausgabeport A (0x378H = 888d) wird dem AB0 zugewiesen, der Eingabeport B (0x379H = 889d) dem EB0. Der Ausgabeport C (0x37AH = 890d), von dem nur einige Bits auf die Schnittstelle geführt sind, wird den Peripherieeingängen PEB10 und den Peripherieausgängen PAB10 zugewiesen. Eine Initialisierung wird nicht vorgenommen.

```
[OEM]
dll0=ioport.dll

[ioportdll]
InitOutputs=0
SafeOutputs=0
BufferOutputs=0
MinByte=0
MaxByte=10
EB0=889
PEB10=890
AB0=888
InitAB0=0
SafeAB0=0
PAB10=890
InitPAB0=0
SafePAB0=0
```

Änderungen

Der Sourcecode dieses Treibers ist von unserem Support bei Bedarf erhältlich. Da neuere Baugruppen in aller Regel auf Basis des PCI Busses aufgebaut sind, können diese Baugruppen nicht über diesen Treiber angesprochen werden. Üblicherweise liefern die Hersteller dieser Baugruppen jedoch Treiber und Zugriffs-DLLs mit. Hier kann der Sourcecode des vorliegenden Treibers als Gerüst dienen, um selbst eine Anpassung an solche, speziellen Baugruppen vorzunehmen.

5.8 Modbus on TCP

Hier wird ein Modbus on TCP Slave konfiguriert, der Daten zyklisch ins PAE/PAA abbilden kann.

- Aufruf des Control SFCs
- Konfigurieren des Treibers über die Oberfläche
- Konfigurieren des Treibers mit PLC43.INI

5.8.1 Modbus on TCP Konfiguration

Dieser Treiber ermöglicht es, dezentrale I/O über Ethernet/Modbus on TCP an die *S7-SoftSPS* anzuschließen. Der Treiber unterstützt die in der Tabelle aufgelisteten Modbus on TCP Telegrammtypen. Soll ein Slave mit der *S7-SoftSPS* betrieben werden, sollte sicher gestellt werden, dass dieser die aufgeführten Modbus-Telegramme unterstützt. Wenn möglich sollte die Modbus Funktion Read/Write Registers (0x17) verwendet werden, da diese Lesen und Schreiben in einem Telegramm durchführt. Unterstützt ein Slave dieses Telegramm nicht, so muss dieser als zwei Slaves angelegt werden. Die erste Konfiguration liest die Eingänge, die zweite schreibt die Ausgänge. Da dies nicht sehr effizient ist, sollte die Funktion Read/Write vorgezogen werden. Zusätzlich installiert der Treiber eine Status SFC/FC mit der der Zustand der Modbus on TCP Verbindung im SPS-Programm abgefragt werden kann.

Funktionsweise der Status SFC/FC

```
CALL          SFC/FC 193
  IN0         := 1          //INPUT INT
  RET_VAL     := MW 0       //RESULT INT
  OUT1        := MW 2       //OUTPUT INT
  OUT2        := MD 4       //OUTPUT DWORD
  OUT3        := MW 8       //OUTPUT INT
  OUT4        := MD 10      //OUTPUT DWORD
```

Parameter	Datentyp	Bedeutung
IN0	INT	Slavenummer Format: 0,1,2,3,... dies entspricht der Slave Parametersatznummer wie in der Datei PLC43.INI konfiguriert
RET_VAL	INT	Gibt bei vorhandener Parametrierung 0 zurück Sonst 0x8090
OUT1	INT	Bit 0 = 1 vorhandene Verbindung Bit 1 = 1 vorhandener Fehler Bit 2...7 reserviert Bit 8...15 verpasste Telegramme seit letztem Aufruf
OUT2	DWORD	Parametrierte IP-Adresse
OUT3	INT	Parametrierter Port
OUT4	DWORD	Windows-Fehler-Code

Die wichtigsten Windows (32-Bit) Netzwerkfehlermeldungen:

0x271D 10013d	/	Permission denied - Deutet auf eingeschränkte Rechte oder eine Firewall hin
0x2745 10053d	/	Software caused connection abort - Wahrscheinlich fehlerhafte Netzwerkverbindung oder kein Slave unter der Adresse vorhanden.
0x2746 10054d	/	Connection reset by peer. Der Slave hat die Verbindung getrennt.
0x274C 10060d	/	Connection timed out. Timeout auf der Netzwerkverbindung.
0x274D 10061d	/	Connection refused. Der Slave (oder ein Router dazwischen) hat die Verbindung verweigert.
0x2751 10065d	/	No route to host. Die Verbindung zum Zielnetz kann nicht hergestellt werden. Netzwerkeinstellungen und Gateway/Routereinstellungen prüfen.

Statuswort

Bei Erfolg ist das BIE des Statuswortes gesetzt, OR, STA und \ER gelöscht.

Bei Misserfolg sind BIE, OR, STA und \ER des Statuswortes gelöscht.

5.8.2 Modbus on TCP konfigurieren mit der Oberfläche

Modbus On TCP Konfiguration

Global:

Status SFC als FC anzeigen

Status SFC / FC Nummer:

Connect Priorität:

Konfigurierte Modbus Teilnehmer:

Adresse	Port	FC	E SPS	E Modul	E Länge	A SPS	A Modul	A Länge	TOut
192.16...	502	0x17	1	1	1	1	0	1	2000
192.16...	502	0x04	0	0	10	-	-	-	2000
192.16...	502	0x06	-	-	-	10	0	1	2000

Konfiguration erstellen / ändern:

IP Adresse: Timeout:

Port Nr.: FC:

Eingänge lesen:

Adresse in SPS:

Adresse in I/O:

Anzahl an Worten:

Ausgänge senden:

Adresse in SPS:

Adresse in I/O:

Anzahl an Worten:

Status SFC als FC anzeigen

Legt fest, ob zur Zustandsabfrage der Modbus Teilnehmer ein FC oder ein SFC im Bausteinverzeichnis der S7-SoftSPS angelegt wird.

Status SFC / FC Nummer

Legt fest welche SFC bzw. FC Nummer vergeben wird. Es sollte darauf geachtet werden, dass die angegebene Nummer nicht bereits in der SPS verwendet wird und in keiner Siemens Bibliothek vergeben ist.

Connect Priorität

Priorität des Connect-Threads.

Konfigurierte Modbus Teilnehmer

Unter Konfigurierte Modbus Teilnehmer können Modbusteilnehmer angelegt, konfiguriert und wieder entfernt werden. Die Anzahl der Parameter, die für einen Slave gemacht werden können, hängt von dem verwendeten Modbus-Telegramm ab.

5.8.3 Modbus on TCP konfigurieren mit PLC43.INI

Einstellungen der Slaves

Die Syntax zum Einstellen der Slaves lautet:

01_Parametersatz für den ersten ModbusSlave

02_Parametersatz für den zweiten ModbusSlave

...

10_Parametersatz für den zehnten und letzten ModbusSlave

Die Parametersätze der Slaves müssen fortlaufend nummeriert werden, ansonsten wird das Einlesen an der Unterbrechungsstelle angehalten. Die Reihenfolge in der Datei PLC43.INI ist beliebig.

Bsp:

```
01_IPAdr=192.168.0.80
```

```
02_IPAdr=192.168.0.81
```

```
06_IPAdr=192.168.0.84
```

Die Folge ist an Stelle 03_ unterbrochen, somit werden nur die Einstellungen der Slaves von 01_ bis 02_ eingelesen.

Parametrierung PLC43.INI

Die Parametrierung der PLC43.ini kann manuell oder mit Hilfe der Konfigurationsoberfläche vorgenommen werden.

```
[OEM]
```

```
;Name des E/A Treibers
```

```
DLL0=ilbeth24.dll
```

```
[ILBETH24]
```

```
;allgemeine Einstellungen
```

```
TypeFC=0 ;als FC erstellen 0 = Nein (SFC), 1 = Ja (FC) (default = 0)
```

```
StatusSFC=193 ;SFC/FC Nummer (default = 193)
```

```
Prio=251 ;Priorität des Connect-Threads (default = 251)
```

```
;erster Slave
```

```
01_IPAdr=192.168.0.21 ;IP Adresse der Verbindung 01
```

```
01_Port=502 ;Port der Verbindung 01 (default = 502)
```

```
01_ModbusFC=0x17 ;ModbusFunctionCode der Verbindung 01 (default = 0x17)
```

```
01_Timeout=2000 ;Timeout der Verbindung in Millisekunden
```

```
01_AdrPLC_AW=1 ;Ausgangswort im Prozessabbild der SPS der Verbindung 01
```

```
01_AdrI/O_AW=2 ;Ausgangswort des I/O-Moduls der Verbindung 01
```

```
01_Count_AW=1 ;Anzahl der Ausgangsworte der Verbindung 01
```

```
01_AdrPLC_EW=1 ;Eingangswort im Prozessabbild der SPS der Verbindung 01
01_AdrI/O_EW=0 ;Eingangswort des I/O-Moduls der Verbindung 01
01_Count_EW=1 ;Anzahl der Eingangsworte der Verbindung 01
```

```
;zweiter Slave
```

```
02_IPAdr=192.168.0.22
02_ModbusFC=0x04
02_AdrPLC_EW=0
02_AdrI/O_EW=0
02_Count_EW=10
```

```
;dritter Slave
```

```
03_IPAdr=192.168.0.23
03_ModbusFC=0x06
03_AdrPLC_AW=10
03_AdrI/O=0
```

```
...
```

Die Anzahl der Parameter, die für einen Slave gemacht werden müssen, hängen von dem verwendeten Modbus-Telegramm ab (siehe Tabelle).

Um aus dem SPS-Programm heraus den aktuellen Verbindungsstatus und damit den Zustand, des Modbus-Slaves abfragen zu können, installiert der Treiber die SFC 193 (wie in der PLC43.INI parametrisiert) in das Bausteinverzeichnis der *S7-SoftSPS*. Dieser Baustein kann im SPS-Programm aufgerufen werden, um auf eventuelle Hardwarefehler zu reagieren.

5.9 NVRAM Konfiguration

Treiber um remanente Bereiche in der *S7-SoftSPS* anzulegen, entweder mit Hilfe einer ADS-TEC PCI-Karte, durch Netzwerkdateisystem, oder Hardware die ein Dateisystem simuliert.

- Allgemeines und überprüfen via SFC
- Konfiguration über Oberfläche
- Konfiguration in PLC43.INI

5.9.1 NVRAM Konfiguration

Mit dem NVRam Konfigurator wird ein remanentes Verhalten in der *S7-SoftSPS* ermöglicht.

Der NVRam Konfigurator legt eine Datei an, in der die remanent zu haltenden Werte gespeichert werden. Je nach Leistungsvermögen ihres Dateisystems und Größe der parametrisierten Werte kann der Speichervorgang Zeit in Anspruch nehmen, so dass der Prozess nicht in Echtzeit erfolgt.

Es ist nicht möglich über die NVRamDII Remanenzdaten auf lokale Festplatten zu speichern, da Microsoft Windows® nur puffernd schreibt und aufgrund des hohen Zugriffes keine Daten auf der Festplatte ablegt. Daher kann das Speichern von Remanenzdaten nur auf Netzwerklaufwerken erfolgen.

Werden PC-Einsteckkarten verwendet, die über batteriegepuffertes RAM eine Festplatte emulieren, muss sicher gestellt werden, dass der verwendete Treiber die Dateisystempufferung des Microsoft Windows® Betriebssystems außer Kraft setzt und die Daten sofort auf die emulierte Festplatte geschrieben werden.

Überprüfen der Remanenz mit STATUS SFC/FC

Der STATUS-SFC/FC gibt zwei Informationen zurück:

OUT0 (DWORD) liefert den Intervall mit dem der letzte Remanenzvorgang von statten ging.

RET_VAL (INT) siehe Tabelle:

RET_VAL(INT)	Bedeutung
0	Alles funktionsbereit
-1	NVRamDll nicht mehr im Speicher
-2	Datei öffnen nicht möglich
-3	Schreibzugriff nicht möglich
-4	Datei schließen nicht möglich

Aufruf des STATUS SFC/FC:

```
CALL      SFC 212
      OUT0   := MD 0 //OUTPUT DWORD
      RET_VAL := MW 4 //RESULT INT
```

Anmerkungen zu Dateisystemen

Es ist nicht möglich über die NVRamDll Remanenzdaten auf lokale Festplatten zu speichern, da Microsoft Windows® nur puffernd schreibt und aufgrund des hohen Zugriffes keine Daten auf der Festplatte ablegt. Daher kann das Speichern von Remanenzdaten nur auf Netzwerklaufwerken erfolgen.

Werden PC-Einsteckkarten verwendet, die über batteriegepuffertes RAM eine Festplatte emulieren, muss sicher gestellt werden, dass der verwendete Treiber die Dateisystempufferung des Microsoft Windows® Betriebssystems außer Kraft setzt und die Daten sofort auf die emulierte Festplatte geschrieben werden.

5.9.2 NVRAM Konfiguration mit der Oberfläche

NVRam Konfiguration

Typ des remanenten Speichers:

Remanente Daten in Dateisystem speichern
 Remanente Daten auf ADS-TEC 128/256k PCI-Karte speichern

Ordner für Sicherungsdatei (keine lokalen Festplatten):
 F:\SoftPLC-NVRAM\

Schreibintervall [ms]: 10

Priorität: Sehr niedrig

Status SFC Nummer: 216

Status SFC als FC anzeigen

Remanente Bereiche:

Bereich	DB-Nummer	Länge
Merker	-	1000
Zähler	-	4
Timer	-	8
Datenbaustein	10	16

Remanente Bereiche definieren:

Typ:
 Merker
 Timer
 Zähler
 Datenbaustein

Länge / DB Nummer:
 Remanente Länge (ab Byte 0): 1000
 Datenbausteinnummer: 1

Hinzufügen Ändern Entfernen

Typ des remanenten Speichers:

Bestimmt ob eine ADS-TEC PCI-Karte oder das Dateisystem verwendet werden soll.

Ordner für Sicherungsdatei (keine lokalen Festplatten):

Pfad für remanente Daten.

Schreibintervall (ms):

Intervall in dem die Remanentendaten gespeichert werden sollen.

Priorität:

Priorität des Threads der die Daten speichert.

Status SFC Nummer:

Bausteinnummer des Status SFC mit dem man den aktuellen Status der Sicherung im SPS-Programm abfragen kann.

Status SFC als FC anzeigen:

Status SFC wird als FC in der *S7-SoftSPS* eingeblendet.

Remanente Bereiche:

Bereich der remanent gehalten wird

5.9.3 NVRAM Konfiguration mit der PLC43.INI

Mit dem vorliegenden Treiber wird ein remanentes Verhalten in der *S7-SoftSPS* ermöglicht. Die *NVRamDII* legt eine Datei an, in der sie die remanent zu haltenden Werte speichert. Je nach Leistungsvermögen ihres Dateisystems und Größe der parametrisierten Werte kann der Speichervorgang Zeit in Anspruch nehmen, so dass der Prozess nicht in Echtzeit erfolgt.

Siehe hierzu auch Anmerkungen zum Dateisystem

Einstellungen in der Konfigurationsdatei PLC43.INI

Um den Treiber zu aktivieren wird in der Sektion [OEM] der Treiber in der Form `dll0=NVRamDII.dll` eingeschaltet. Der Treiber erwartet die Sektion [IBHNVRAM], in der die weiteren Parameter definiert sind.

Name des Parameters	Defaultwert	Erklärung
RemFolder(benötigt)		Pfad in dem die Remanentendaten hinterlegt werden.
WriteInterval(optional)	10	Zeitintervall in Millisekunden in denen der Thread die Daten abspeichert.
WritePrio(optional)	THREAD_PRIORITY_IDLE	Windows Thread Priorität mit der NVRamDII läuft.
ShowFC(optional)	0	Statusbaustein als FC einblenden 0 = SFC, <>1 = FC.
SFCNo(optional)	212	Statusbausteinnummer
Das ‚#‘-Zeichen ist ein Platzhalter für eine Zahl zwischen 0 und 99		
TYP#		Remanententyp; M = Merker, D = Datenbaustein, T = Zeiten, Z = Zähler.
LEN#		Länge remanent zu haltender Daten ab Adresse 0.
DBNR#		Datenbausteinnummer – muss nur bei TYP = D angegeben werden.

Beispiel mit 1000 Byte Merkern, 4 Zählern, 8 Zeiten und 16 Byte aus DB10:

```
[OEM]
dll0=NVRamDII.dll

[IBHNVRAM]
RemFolder=F:\SoftPLC-NVRAM\
TYP0=M
LEN0=1000
TYP1=Z
LEN1=4
TYP2=T
LEN2=8
TYP3=D
LEN3=16
DBNR3=10
```

5.10 Softnet PN IO

Treiber um Softnet PN IO von der *S7-SoftSPS* zu verwenden

- Allgemeine Konfiguration
- Konfiguration über die Oberfläche
- Konfiguration in der PLC43.INI

5.10.1 Softnet PN IO Konfiguration

Um die *S7-SoftSPS* mit Profinet IO über Simatic SoftNet PN IO 2006 nutzen zu können müssen folgende Schritte durchgeführt werden.

- SoftNet muss installiert und lizenziert auf dem Rechner vorliegen.
- Der Rechner muss über eine statische IP Adresse verfügen

Komponenten Konfigurator

- Im Komponenten Konfigurator wird eine Applikation auf den Index 1 eingefügt

ACHTUNG: DER STATIONSNAME MUSS IDENTISCH MIT DEM RECHNERNAMEN SEIN.

- Auf Index 3 wird eine Komponente vom Typ "IE Allgemein" eingefügt. In der Parametrierung wählt man die Netzwerkkarte aus, die für PROFINET® verwendet werden soll.
- Im erscheinenden Fenster "Komponenteneigenschaften" überprüft man die Netzwerkeinstellungen (Der Haken bei "Nur ISO-Protokoll aktivieren darf nicht gesetzt sein)

Simatic Manager (STEP@7 v5.4 SP1 oder größer)

- Die PG/PC Schnittstelle wird auf TCP/IP eingestellt.
- Im Projekt wird eine SIMATIC PC-Station eingefügt.

ACHTUNG: DER STATIONSNAME MUSS IDENTISCH MIT DEM RECHNERNAMEN SEIN.

- Die Hardwarekonfiguration starten.
- Auf Index 1 wird ein Objekt eingefügt. Rechtsklick auf Index 1 >> Objekt einfügen >> Benutzer Applikation >> Applikation >> SW V6.3
- Auf Index 3 wird ein Objekt eingefügt. Rechtsklick auf Index 3 >> Objekt einfügen >> CP Industrial Ethernet >> IE Allgemein >> SW V6.2 SP1...
- Im erscheinenden Fenster "Eigenschaften - Ethernet Schnittstelle IE Allgemein" wird bei IP-Adresse die statische IP des Rechners eingegeben und die Ethernet Schnittstelle mit einem neuen Subnetz mit Standard Parametern vernetzt.
- Mit einem Rechtsklick auf IE Allgemein >> PROFINET IO System einfügen wird ein PROFINET® Netz erzeugt.

I/O Klemmen dem PROFINET® I/O Netz hinzufügen

- Mit einem Rechtsklick auf IE Allgemein >> Ethernet Teilnehmer bearbeiten, erscheint folgendes Fenster:
- Durch den Button "Durchsuchen" können vorhandene PROFINET®-Knoten gesucht und in die Oberfläche geladen werden.
- Jedem PROFINET®-Knoten muss ein eindeutiger Name vergeben werden. Nachdem via Durchsuchen ein Knoten in den Dialog geladen wurde, kann unter "Gerätename vergeben" in dem Textfeld "Gerätename" ein Namen für den Knoten eingegeben und durch einen Klick auf "Name zuweisen" dem Knoten zugeteilt werden.

ACHTUNG: DIESER NAME MUSS EINZIGARTIG IM PROFINET®-NETZ SEIN. DIESER NAME WIRD SPÄTER BEIM KONFIGURIEREN WIEDER BENÖTIGT.

- Jetzt kann ein PROFINET®-Knoten dem PROFINET®-Netz zugeteilt werden. Der Knoten wird aus dem Katalog mittels Drag & Drop dem Netz hinzugefügt.
- Rechtsklick auf den Knoten >> Objekteigenschaften erscheint folgender Dialog:
- Unter Geräte-name muss nun der gleiche Namen eingestellt werden, wie vorher dem Knoten bei Ethernet Teilnehmer bearbeiten zugeteilt wurde.

ACHTUNG: DER GERÄTENAME MUSS IDENTISCH SEIN MIT DEM FÜR DEN KNOTEN VERGEBENEN NAMEN WEITER OBEN

- Die einzelnen Klemmen des Knotens müssen in dem Knoten entsprechend konfiguriert werden, sonst funktioniert die Kommunikation zu dem Knoten nicht.

Abschließen der Hardware-Konfiguration

- Die eingegebene Hardware-Konfiguration muss nun noch übersetzt und in die Baugruppe geladen werden.

ACHTUNG: DER BUS LÄUFT ERST NACH CA. 30S AN, WENN VON DER APPLIKATION DIE I/Os GETRIGGERT WERDEN.

Unnötige CPU Last Vermeiden:

SoftNet PN IO ist ein Software-Treiber der Systemressourcen braucht. Damit das System performanter wird sollte folgendes beachtet werden.

- Nicht benötigte Dienste abschalten.
- Simatic NET IEPG Help Service auf „Deaktiviert“ setzen.
- Den Stationsmanager aus den Autostart (Startmenü) entfernen.

Der Dienst „Simatic Net Station Manager (s7wnsmk.exe)“ hat den Port 102 (iso-tsap) für sich reserviert, so dass im Zusammenhang mit SoftNet PN IO die CP343/443 Emulation (CPx43) nicht betrieben werden kann, da dieser den Port 102 ebenfalls benötigen würde. Deaktiviert man den Dienst „Simatic Net Station Manager (s7wnsmk.exe)“, so funktioniert Profinet IO nicht mehr.

Typische Fehler:

- Der Siemens „Komponenten Konfigurator“ verliert die Netzwerkkarte.

Abhilfe:

- Den Komponenten Konfigurator starten
- Die Netzwerkkarte einfügen
- Den Komponenten Konfigurator beenden
- Den Komponenten Konfigurator wieder starten
- Die Netzwerkkarte entfernen (Jetzt weiß dieser, dass die Karte nicht mehr vorhanden ist)
- Die Netzwerkkarte wieder einfügen (Jetzt ist die Karte bereit)
- Den Komponenten Konfigurator beenden
- Die Konfiguration mit Simatic NCM PC Manager neu laden

SoftNet PN IO in der S7-SoftSPS verwenden

SoftNet PN IO kann aus dem SPS-Programm mittels SFC14/15 SFB52/53 angesprochen werden.

Der Parameter LADDR beziehungsweise ID und INDEX entsprechen der PROFINET®-Adresse aus der Hardware-Konfiguration.

5.10.2 Konfigurieren von Softnet mit der Oberfläche

Profinet Konfiguration

Prozessdatenaustausch über SFC14/15, Records über SFB52/53

Prozessdatenaustausch erfolgt über benutzerdefinierte Bausteine:

Baustein DPRD_DAT ist FC: 14

Baustein DPWR_DAT ist FC: 15

Baustein RD_REC ist FB: 52

Baustein WR_REC ist FB: 53

Status:

Standard

Prozessdatenaustausch über SFC14/15, Records über SFB52/53

Nutzt zum Datenaustausch die Standard SFC/SFB.

Baustein "Funktionsname" ist FC:

Bausteinnummer der Funktion "Funktionsname".

Standard

Setzt die Einstellungen auf Standardwerte zurück.

5.10.3 Konfigurieren von Softnet mit PLC43.INI

Der vorliegende Treiber ermöglicht dem Anwender aus dem SPS-Programm mittels SFC 14/15 Daten mit Profinet IO Slaves über PROFINET IO SoftNet® auszutauschen.

Des weiteren stehen zur Diagnose die SFB 52/53 zur Verfügung welche es ermöglichen die PROFINET IO SoftNet® Funktionen Datensatz lesen/schreiben zu nutzen.

Parametrieren der PnioSn.dll in der PLC43.INI

Die PnioSn.dll muss in der Sektion [OEM] eingebunden werden.

```
[OEM]
```

```
dll10=pniosn.dll
```

Des weiteren erwartet die PnioSn.dll eine Sektion [PNIOSN] in der die folgenden Parameter eingestellt werden:

Parameter	Defaultwert, Mögliche Werte	Erklärung
SlotNumber	3	in SoftNet eingestellte Port Nummer
TypeFC	0, 1	Standart SFC/SFB verwenden
SFC14	14, 1 ... 255	Baustein DPRD_DAT ist FC Nr
SFC15	15, 1 ... 255	Baustein DPWR_DAT ist FC Nr
SFB52	52, 1 ... 255	Baustein RDREC ist FC Nr
SFB53	53, 1 ... 255	Baustein WRREC ist FC Nr

Folgendes Beispiel stellt PnioSn.dll auf Slotnummer 3, Funktionsbausteine als FC/FB und nutzt die FC/FB Nummern äquivalent zu den SFC/SFB Nummern:

```
[PNIOSN]
```

```
SlotNumber=3
```

```
TypeFC=1
```

```
SFC14=14
```

```
SFC15=15
```

```
SFB52=52
```

```
SFB53=53
```

5.11 User SFC

Hier werden selbsterstellte SFCs/FCs in die *S7-SoftSPS* eingeblendet.

- Allgemeines zu User SFCs/FCs
- Konfiguration der User SFCs/FCs mit Hilfe der Oberfläche
- Konfiguration der User SFCs/FCs in der PLC43.INI

5.11.1 User SFC Konfiguration

Es besteht die Möglichkeit, selbst erstellte SFCs/FCs der *S7-SoftSPS* hinzugefügt werden. Die *S7-SoftSPS* erlaubt es dem Anwender bis zu 100 DLLs mit selbst erstellten Systemfunktionen (SFC) oder Funktionen (FC) zu verwenden. Jede DLL kann wiederum mehrere SFC bzw. FC enthalten. Diese SFC/FC können nach vorgegeben Regeln in C++ vom Anwender erstellt werden.

Durch Einbinden eigener SFC/FC, kann der Funktionsvorrat der SoftSPS um eigene, applikationsspezifische Codeteile erweitert werden (Technologiefunktionen, Know-How Schutz, spezielle Hardwarebaugruppen). Werden die Bausteine (FC) mit dem PG zur SPS übertragen, behält die SPS die in C++ erstellten internen Bausteine bei.

Beispiele zum Erstellen eigener SFC/FC-DLLs sind beim IBHsoftec Support erhältlich.

5.11.2 Einstellen der UserSFCs/FCs in der Oberfläche

User SFC Konfiguration

Verfügbare SFCs/FCs

- C:\UserSFC\sfc253.dll
- C:\UserSFC\sfc254.dll
- C:\UserSFC\sfc255.dll
- C:\UserSFC\sfc256.dll
- C:\UserSFC\sfc257.dll
- C:\UserSFC\sfc258.dll
- C:\UserSFC\sfc259.dll

Gewählte SFCs/FCs

- C:\UserSFC\sfc256.dll
- C:\UserSFC\sfc258.dll

SFC in anderem Ordner oder spezielle SFC suchen:

I/O Ports unter NT 4 / Windows 2000 / Windows XP freigeben:

I/O Ports freigeben

Startadresse der freizugebenden Ports (Hex):

Anzahl der freizugebenden Ports:

Verfügbare SFCs/FCs

Hier sind die verfügbaren SFC/FC-DLLs aufgelistet. Sollten hier keine SFC/FC-DLLs aufgeführt werden, können diese mit der Schaltfläche Suchen hinzugefügt werden. Mit Betätigen der „>>“ Schaltfläche wird die markierte SFC/FC-DLL in das rechte Feld (Gewählte SFCs/FCs) verschoben und somit der *S7-SoftSPS* hinzugefügt.

Gewählte SFCs/FCs

Hier sind die der *SoftSPS* momentan zugeordneten SFC/FC-DLLs aufgelistet. Mit Betätigen der „<<“ Schaltfläche wird die markierte SFC/FC-DLL in das linke Feld (Verfügbare SFC/FC) verschoben und somit von der *S7-SoftSPS* entfernt. Die SFC/FC-DLLs können der *SoftSPS* jeder Zeit wieder hinzugefügt werden.

Mit der Schaltfläche Suchen können dem Feld "Verfügbare SFCs/FCs" selbsterstellte SFC/FC-DLLs hinzugefügt werden.

I/O Ports unter Windows® NT4 / Windows® 2000 / Windows® XP freigeben

Greift eine Anwender SFC/FC direkt auf Hardwareports des PCs zu, so ist zu beachten, dass unter Windows® NT4, Windows® 2000 und Windows® XP kein direkter Zugriff auf die PC Hardware erlaubt ist. Um solche Zugriffe trotzdem zu erlauben, können die Treiber der *S7-SoftSPS* die Zugriffe auf die PC Hardware durch das Betriebssystem freigeben lassen.

5.11.3 Einstellen der UserSFCs/FCs in der PLC43.INI

Die Sektion [SF] erlaubt es dem Anwender bis zu 100 DLLs mit selbst erstellten Systemfunktionen (SFC) oder Funktionen (FC) zu verwenden. Jede DLL kann wiederum mehrere SFC bzw. FC enthalten. Diese SFC/FC können nach vorgegeben Regeln in C++ vom Anwender erstellt werden.

Durch Einbinden eigener SFC/FC, kann der Funktionsvorrat der *S7-SoftSPS* um eigene, applikationsspezifische Codeteile erweitert werden (Technologiefunktionen, Know-How Schutz, spezielle Hardwarebaugruppen). Werden die Bausteine (FC) mit dem PG zur SPS übertragen, behält die SPS die in C++ erstellten internen Bausteine bei.

Beispiele zum Erstellen eigener SFC/FC sind beim *IBHsoftec* Support erhältlich.

Keyname	Wertebereich	Defaultwert	Beschreibung
SF00	SFCnnn.DLL	---	Dateiname der selbst erstellten SFC/FC
SF01	SFCnnn.DLL	---	Dateiname der selbst erstellten SFC/FC
SF02	SFCnnn.DLL	---	Dateiname der selbst erstellten SFC/FC
....			
SF99	SFCnnn.DLL	---	Dateiname der selbst erstellten SFC/FC

Hardwareports freigeben

Greift eine Anwender SFC/FC direkt auf Hardwareports des PCs zu, so ist zu beachten, dass unter Windows NT4, Windows 2000 und Windows XP kein direkter Zugriff auf die PC Hardware erlaubt ist.

Um solche Zugriffe trotzdem zu erlauben, können die Treiber der *S7-SoftSPS* die Zugriffe auf die PC Hardware durch das Betriebssystem freigeben lassen. Hierzu muss in der Datei PLC43.INI die Sektion [IO-PORTS] angelegt werden.

Nachstehend werden beginnend bei der Adresse 0x0378, auf der sich normalerweise der Druckerport des PCs befindet, 4 Ports zur Bearbeitung freigeschaltet.

```
[ IO-PORTS ]
```

```
378=4
```

Für die serielle Schnittstelle COM1, die sich normalerweise auf der Adresse 0x03F8 befindet und 8 Bytes belegt, würde der Eintrag wie folgt aussehen:

```
[ IO-PORTS ]
```

```
3F8=8
```

Alternativ kann auch der mitgelieferte IO-Port Treiber (ioport.dll) verwendet werden.

6 Technische Daten

Dieses Kapitel beinhaltet die technischen Daten der S7-SoftSPS

- Ausbaustand
- Operationsvorrat
- Diagnosepuffereinträge
- Unterschiede
- AppStarter
- Null-Modemkabel

6.1 Technische Daten

	Windows NT 4.x/2000/XP	
	PLC S7 315	PLC S7 416
Ladespeicher	256 kByte	einstellbar
Arbeitsspeicher	256 kByte	einstellbar
Bausteine	16.384	16.384
Merker (Bit)	16.384	131.072
Zeiten	256	2048
Zähler	256	2048
Digital E/A	1.024 Bit	32.768 Bit
Analog E/A	2.048 Byte	8.192 Byte
	Bearbeitungszeiten *)	
Pentium III 600 MHz	100 µs	100 µs
Pentium IV 1,1 GHz	50 µs	50 µs
Pentium IV 2,6 GHz	28 µs	28 µs

*) Zeiteinstellung 50% SPS-Zeit : 50% Windows-Zeit; 1024 gemischte Anweisungen (50% Binär, 50% Digital)

6.2 Operationsvorrat

Die *SoftSPS PLC S7-315* besitzt den kompletten Befehlsvorrat der Simatic® S7-315 CPU.

Die *SoftSPS PLC S7-416* besitzt den kompletten Befehlsvorrat der Simatic® S7-416 CPU.

Binäre Verknüpfungsoperationen

Operation	Operand	Funktionsbeschreibung
U		UND mit Abfrage auf „1“
UN		UND mit Abfrage auf „0“
O		ODER mit Abfrage auf „1“
ON		ODER mit Abfrage nach „0“
X		Exklusiv-ODER mit Abfrage auf „1“
XN		Exklusiv-ODER mit Abfrage auf „0“
	E	eines Eingangs
	A	eines Ausgangs
	M	eines Merkers
	L	eines Lokaldatenbits
	T	einer Zeitfunktion
	Z	einer Zählfunktion
	DBX	eines Globaldatenbits
	DIX	eines Instanzdatenbits
==0		Ergebnis gleich Null
<>0		Ergebnis ungleich Null
>0		Ergebnis größer Null
>=0		Ergebnis größer-gleich Null
<0		Ergebnis kleiner Null
<=0		Ergebnis kleiner-gleich Null
UO		ungültiges Ergebnis
OV		Overflow
OS		Überlauf (speichernd)
BIE		Binärerergebnis

Sonstige Verknüpfungen

Operation	Operand	Funktionsbeschreibung
U(UND Klammer auf
UN(UND-NICHT Klammer auf
O(ODER Klammer auf
ON(ODER-NICHT Klammer auf
X(Exklusiv-ODER Klammer auf
XN(Exklusiv-ODER-NICHT Klammer auf
)		Klammer zu
O		ODER-Verknüpfung von UND
NOT		VKE negieren
SET		VKE setzen
CLR		VKE rücksetzen
SAVE		VKE ins BIE retten

Operation	Operand	Funktionsbeschreibung
=		Zuweisung
S		Setze
R		Rücksetzen
FP		Positive Flanke
FN		Negative Flanke
	E	eines Eingangsbits
	A	eines Ausgangsbits
	M	eines Merkerbits
	L	eines Lokaldatenbits
	DBX	eines Datenbits
	DIX	eines Instanz-Datenbits

Master Control Relay

Operation	Operand	Funktionsbeschreibung
MCRA		MCR-Bereich aktivieren
MCRD		MCR-Bereich deaktivieren
MCR(MCR-Zone öffnen
)MCR		MCR-Zone schließen

Lade / Transferoperationen

Operation	Operand	Funktionsbeschreibung
L		Operand laden
T		Operand transferieren
	EB	Eingangsbyte
	EW	Eingangswort
	ED	Eingangsdoppelwort
	AB	Ausgangsbyte
	AW	Ausgangswort
	AD	Ausgangsdoppelwort
	MB	Merkerbyte
	MW	Merkerwort
	MD	Merkerdoppelwort
	LB	Lokaldatenbyte
	LW	Lokaldatenwort
	LD	Lokaldatendoppelwort
	DBB	Datenbausteinbyte
	DBW	Datenbausteinwort
	DBD	Datenbausteindoppelwort
	DIB	Instanz-Datenbausteinbyte
	DIW	Instanz-Datenbausteinwort
	DID	Instanz-Datenbausteindoppelwort
	STW	Statuswort
L	konst	Laden einer Konstanten
L	#Pointer	Laden eines Zeigers
L	T	Zeitwert
LC	T	Zeitwert (BCD codiert)
L	Z	Zähler
LC	Z	Zähler (BCD codiert)
L	DBNO	Lade Datenbausteinnummer
L	DBLG	Lade Datenbausteinlänge
L	DINO	Lade Datenbausteinnummer (Instanz-DB)
L	DILG	Lade Datenbausteinlänge (Instanz-DB)

Peripheriezugriffe

Operation	Operand	Funktionsbeschreibung
L	PEB	Peripherieeingangsbyte
L	PEW	Peripherieeingangswort
L	PED	Peripherieeingangsdoppelwort
T	PAB	Peripherieausgangsbyte
T	PAW	Peripherieausgangswort
T	PAD	Peripherieausgangsdoppelwort

Akkumulatorbefehle

Operation	Operand	Funktionsbeschreibung
PUSH		Akkus „nach oben“ schieben
POP		Akkus „nach unten“ schieben
ENT		Akkus schieben (ohne AI) A2->A3, A3->A4
LEAVE		Akkus schieben (ohne AI) A4->A3, A3->A2
TAK		Akku 1 und Akku 2 tauschen
TAW		Akku 1 Bytes 0 und 1 tauschen
TAD		Akku 1 alle Bytes tauschen

Schiebe- und Rotieroperationen

Operation	Operand	Funktionsbeschreibung
SLW		Schieben links wortweise
SLD		Schieben links doppelwortweise
SRW		Schieben rechts wortweise
SRD		Schieben rechts doppelwortweise
SSI		Schieben mit Vorzeichen wortweise
SSD		Schieben mit Vorzeichen doppelwortweise
RLD		Rotieren links doppelwortweise
RLDA		Rotieren links durch AI
RRDA		Rotieren rechts durch AI
RRD		Rotieren rechts doppelwortweise
	konst	um feste Anzahl von Bits
	-	Schiebezahl im Akku 2

Zeitoperationen

Operation	Operand	Funktionsbeschreibung
SI	T	Starten als Impuls
SV	T	Starten als verlängerter Impuls
SE	T	Starten als Einschaltverzögerung
SS	T	Starten als speichernde Einschaltverzögerung
SA	T	Starten als Ausschaltverzögerung
R	T	Zeitfunktion rücksetzen
FR	T	Zeitfunktion freigeben

Zähloperationen

Operation	Operand	Funktionsbeschreibung
ZV	Z	Zählfunktion vorwärtszählen
ZR	Z	Zählfunktion rückwärtszählen
S	Z	Zählfunktion setzen
R	Z	Zählfunktion rücksetzen
FR	Z	Zählfunktion freigeben

Wortoperationen

Operation	Operand	Funktionsbeschreibung
UW		UND wortweise
UD		UND doppelwortweise
OW		ODER wortweise
OD		ODER doppelwortweise
XOW		Exklusiv-ODER wortweise
XOD		Exklusiv-ODER doppelwortweise
	konst	Mit einer Wort- bzw. Doppelwortkonstanten
	-	mit dem Inhalt von Akku 2

Arithmetische Operationen

Operation	Operand	Funktionsbeschreibung
==I		Integervergleich GLEICH
<>I		Integervergleich UNGLEICH
>I		Integervergleich GRÖSSER
>=I		Integervergleich GRÖSSER GLEICH
<I		Integervergleich KLEINER
<=I		Integervergleich KLEINER GLEICH
==D		Doppel-Integervergleich GLEICH
<>D		Doppel-Integervergleich UNGLEICH
>D		Doppel-Integervergleich GRÖSSER
>=D		Doppel-Integervergleich GRÖSSER GLEICH
<D		Doppel-Integervergleich KLEINER
<=D		Doppel-Integervergleich KLEINER GLEICH
==R		REAL-Zahl Vergleich GLEICH
<>R		REAL-Zahl Vergleich UNGLEICH
>R		REAL-Zahl Vergleich GRÖSSER
>=R		REAL-Zahl Vergleich GRÖSSER GLEICH
<R		REAL-Zahl Vergleich KLEINER
<=R		REAL-Zahl Vergleich KLEINER GLEICH
SIN		Sinus
COS		Cosinus
TAN		Tangens
ASIN		Arcussinus
ACOS		Arcuscosinus
ATAN		Arcustangens
SQR		Quadrieren
SQRT		Wurzel (Radizieren)
EXP		Exponent zur Basis e
LN		Natürlicher Logarithmus
+I		Integer Addition
-I		Integer Subtraktion
*I		Integer Multiplikation
/I		Integer Division
+D		Doppel-Integer Addition
-D		Doppel-Integer Subtraktion
*D		Doppel-Integer Multiplikation
/D		Doppel-Integer Division
+R		Realzahl Addition
-R		Realzahl Subtraktion
*R		Realzahl Multiplikation
/R		Realzahl Division
MOD		Doppel-Integer Division (Rest)
+	konst	Addieren einer Konstante

+P#	konst	Addieren eines Zeigers
DEC		Dekrementieren
INC		Inkrementieren

Umwandlungsfunktionen

Operation	Operand	Funktionsbeschreibung
ITD		Wandlung NT nach DNT
ITB		Wandlung NT nach BCD
DTB		Wandlung DNT nach BCD
DTR		Wandlung DNT nach REAL
BTI		Wandlung BCD nach NT
BTD		Wandlung BCD nach DINT
RND		Rundung zur nächsten ganzen Zahl
RND+		Rundung zur nächstgrößeren Zahl
RND-		Rundung zur nächstkleineren Zahl
TRUNC		ohne Rundung
INV1		INT-Einerkomplement
INVD		DINT-Einerkomplement
NEGI		INT-Negation
NEGD		DINT-Negation
NEOR		REAL-Negation
ABS		REAL-Betragsbildung

Sprungoperationen

Operation	Operand	Funktionsbeschreibung
SPA	Ziel	Sprung absolut (unbedingt)
SPB	Ziel	Sprung wenn VKE = 1
SPBB	Ziel	Sprung wenn VKE = 1 (VKE speichern)
SPBN	Ziel	Sprung wenn VKE = 0
SPBNB	Ziel	Sprung wenn VKE = 0 (VKE speichern)
SPBI	Ziel	Sprung wenn BIE = 1
SPBIN	Ziel	Sprung wenn BIE = 0
SPZ	Ziel	Sprung bei Vergleichsergebnis = 0
SPN	Ziel	Sprung bei Vergleichsergebnis != 0
SPP	Ziel	Sprung bei Vergleichsergebnis > 0
SPPZ	Ziel	Sprung bei Vergleichsergebnis >= 0
SPM	Ziel	Sprung bei Vergleichsergebnis < 0
SPMZ	Ziel	Sprung bei Vergleichsergebnis <= 0
SPU	Ziel	Sprung bei Vergleichsergebnis ungültig
SPO	Ziel	Sprung bei Overflow
SPS	Ziel	Sprung bei speicherndem Overflow
SPL	Ziel	Sprungverteiler
LOOP	Ziel	Schleife

Aufruf Operationen

Operation	Operand	Funktionsbeschreibung
CALL FB		Sprung zum Funktionsbaustein
CALL FC		Sprung zur Funktion
CALL SFB		Sprung zum System-Funktionsbaustein
CALL SFC		Sprung zur System-Funktion
UC FB		Funktionsbaustein absolut aufrufen
CC FB		Funktionsbaustein bedingt aufrufen
UC FC		Funktion absolut aufrufen
CC FC		Funktion bedingt aufrufen
BEA		Bausteinende absolut
BEB		Bausteinende bedingt

BE		Bausteinende
AUF	DB DI	Datenbaustein öffnen Instanz-Datenbaustein öffnen
TDB		Datenbausteinregister tauschen

Indirekte Adressierung

Operation	Operand	Funktionsbeschreibung
LAR1/LAR2	MD LD DBD DID	AR1/AR2 laden mit Merkerdoppelwort mit Lokaldatendoppelwort mit Datenbausteindoppelwort mit Instanz-Datenbausteindoppelwort
LAR1	-	AR1 laden mit Inhalt aus Akku1
LAR2	-	AR2 laden mit Inhalt aus Akku1
LAR1	AR2	AR1 laden mit Inhalt aus A2
LAR1	P#	AR1 laden mit Zeiger
LAR2	P#	AR2 laden mit Zeiger
TAR1/TAR2	MD LD DBD DID	Schreibe AR1 / AR2 in Merkerdoppelwort in Lokaldatendoppelwort in Datenbausteindoppelwort in Instanz-Datenbausteindoppelwort
TAR1	-	Schreibe AR1 in Akku1
TAR2	-	Schreibe AR2 in Akku1
TAR1	AR2	AR1 nach AR2 transferieren
TAR		AR1 mit AR2 tauschen
+AR1		Addiere Akku1 zu AR1
+AR2		Addiere Akku1 zu AR2
+AR1	P#	Addiere Pointer zu AR1
+AR2	P#	Addiere Pointer zu AR2

Sonstige Operationen

Operation	Operand	Funktionsbeschreibung
NOP	0	
NOP	1	Nulloperation
BLD	konst	Bildaufbau

Aufbau des SPS-Speichers der S7-SoftSPS

Der Speicher der *SoftSPS PLC S7-315* und *SoftSPS PLC S7-416* gleicht im Wesentlichen dem Speicher der Simatic[®]-Steuerung S7-315 bzw. S7-416.

Beachten Sie, dass bei einem Zugriff von Windows-Programmen auf diesen Speicher das niederwertige mit dem höherwertigen Byte getauscht werden muss.

Aufbau der Systemdaten SoftSPS

Der Aufbau der Systemdaten in den Systemdatenbausteinen gleicht im Wesentlichen dem Aufbau der Systemdaten in der Simatic[®]-Steuerung S7-315 bzw. S7-416.

Integrierte Organisationsbausteine

Die in der folgenden Tabelle aufgeführten Organisationsbausteine werden ausgeführt, sofern sie programmiert sind:

OB-Nr	Kurzbeschreibung
OB 1	Hauptprogramm
OB 10 – OB 17	Uhrzeitalarm
OB 20 – OB 23	Verzögerungsalarm
OB 30 – OB 38	Zeitbausteine (zyklischer Aufruf)
OB 80 – OB 87	Fehlerbausteine
OB 90	Hintergrundbearbeitung
OB 100/OB 101	Neustart bzw. Wiederanlauf
OB 121/OB 122	Programmierfehler

Integrierte Systemfunktionen

In der SoftSPS sind die für eine Software-SPS relevanten Systemfunktionen (SFC) integriert. S7 für Windows unterstützt die SFC-Aufrufe durch intern vorhandene Bausteinkopf-Deklarationen. Im Editiermodus wird ein integrierter SFC über die Menüpunkte "Einfügen|S7-Baustein aufrufen" in das aufgeschlagene Netzwerk eingebunden. Die Bausteinparameter werden mit Name und Typ angezeigt.

Beispiel für SFC 20:

```
CALL      SFC 20

SRCBLK   := P#M 0.0 BYTE 1 //INPUT ANY
RET_VAL  := MW 0           //RESULT INT
DSTBLK   := P#M 0.0 BYTE 1 //OUTPUT ANY
```

Die folgende Tabelle zeigt (in numerischer Aufzählung) die vorhandenen SFC's.

SFC-Nr	SFC-Name	Kurzbeschreibung
SFC 0	SET_CLK	Setzen der Systemzeit
SFC 1	READ_CLK	Lesen der Systemzeit
SFC 5***	GADR_LGC	Logische Basisadresse einer Baugruppe ermitteln
SFC 6	RD_SINFO	Startinformation des aktuellen OBs auslesen
SFC 13*	DPNRM_DG	Slave-Diagnose eines DP-Slaves
SFC 14**	DPRD_DAT	Konsistente Daten eines DP-Normslaves lesen
SFC 15**	DPWR_DAT	Daten konsistent auf DP-Normslave schreiben
SFC 20	BLKMOV	SPS-Variable in Zielbereich kopieren
SFC 21	FILL	SPS-Variable in Zielbereich vorbesetzen
SFC 22	CREAT_DB	Anlegen von Datenbausteinen
SFC 23	DEL_DB	Datenbaustein löschen
SFC 24	TEST_DB	Attribute eines Datenbausteins testen
SFC 25	COMPRESS	Ladespeicher komprimieren
SFC 28	SET_TINT	Stellen eines Uhrzeitalarms
SFC 29	CAN_TINT	Uhrzeitalarm anhalten (nicht ausführen)
SFC 30	ACT_TINT	Uhrzeitalarm freigeben

SFC 31	QRY_TINT	Status des Uhrzeitalarms
SFC 32	SRT_DINT	Starte Verzögerungsalarm
SFC 33	CAN_DINT	Unterbreche Verzögerungsalarm
SFC 34	QRY_DINT	Statusabfrage des Verzögerungsalarms
SFC 36	MSK_FLT	Maskieren eines synchronen Fehlerereignisses
SFC 37	DMSK_FLT	Demaskieren eines synchronen Fehlerereignisses
SFC 38	READ_ERR	Auslesen des Ereignisstatusregister
SFC 39	DIS_IRT	Sperren der Asynchron- und Alarmfehlerbearbeitung
SFC 40	EN_IRT	Freigeben der Asynchron- und Alarmfehlerbearbeitung
SFC 41	DIS_AIRT	Verzögerung höherpriorer Asynchron- und Alarmfehlerbearbeitung
SFC 42	EN_AIRT	Freigabe höherpriorer Asynchron- und Alarmfehlerbearbeitung
SFC 43	RE_TRIGR	Nachtriggern des Watchdogs
SFC 46	STP	CPU Betriebszustand STOP
SFC 49***	LGC_GADR	Ermitteln des Steckplatzes einer logischen Adresse
SFC 50***	RD_LAGADR	Ermitteln aller logischer Adressen einer Baugruppe
SFC 51	RDSYST	Systemzustandsliste (SZL) auslesen
SFC 64	TIME_TCK	Timer Tick
SFC 81	UBLKMOV	Block Move der nicht unterbrochen werden kann

*Die SFC 13 ist nur vorhanden, wenn der Treiber Cifcfg.dll eingebunden ist. Zudem kann diese nur gültige Ergebnisse liefern, wenn die Profibuskonfiguration über den S7 für Windows® Hardwarekonfigurator oder den Step®7 Hardwarekonfigurator erzeugt wurde. Dies ist nur bei Verwendung des Treibers (im Lieferumfang) CIFCFG.DLL möglich.

**Die SFC 14 und SFC 15 sind nur verfügbar, wenn diese über den Treiber CIF30D0.dll oder den Treiber CIFCFG.DLL eingebunden sind.

*** Die SFCs können nur gültige Ergebnisse liefern, wenn die Profibuskonfiguration über den S7 für Windows® Hardwarekonfigurator oder den Step®7 Hardwarekonfigurator erzeugt wurde. Dies ist nur bei Verwendung des Treibers (im Lieferumfang) CIFCFG.DLL möglich.

Integrierte Systembausteine

In der SoftSPS sind die für eine Software-SPS relevanten Systembausteine (SFB) integriert.

S7 für Windows® unterstützt die SFB-Aufrufe durch intern vorhandene Bausteinkopf-Deklarationen (siehe Kapitel "Integrierte Systemfunktionen").

Folgende Systemfunktionsbausteine werden unterstützt:

SFB-Nr	SFB-Name	Kurzbeschreibung
SFB 0	CTU	Vorwärtszähler (Counter Up)
SFB 1	CTD	Rückwärtszähler (Counter Down)
SFB 2	CTUD	Vorwärts- und Rückwärtszähler
SFB 3	TP	Zeitimpuls
SFB 4	TON	Einschaltverzögerung
SFB 5	TOF	Ausschaltverzögerung
SFB 32	DRUM	Schaltwerk zur Steuerung von Schrittketten
SFB 52****	RDREC_DP	Datensatz lesen
SFB 53****	WRREC_DP	Datensatz schreiben

****Die SFB 52 und SFB 53 sind nur unter Verwendung des Treibers CIFCFG.DLL und entsprechendem Eintrag DPV1=1 in der Sektion [CIFCFG] der Konfigurationsdatei PLC43.ini verfügbar.

6.3 Diagnosepuffereinträge

Die untenstehende Tabelle listet die Unterschiede zu S7-Hardware-SPS Diagnosepuffereinträgen auf.

Diagnosepuffereinträge	Treiber	Erklärung
0xA000	CIF PROFIBUS	CIF-PROFIBUS-Karte konnte nicht initialisiert werden
0xA001	CIF PROFIBUS	Timeout während Kaltstartphase
0xA002	CIF PROFIBUS	Parameter für CIF-Task unpassend
0xA003	CIF PROFIBUS	Timeout während Warmstartphase
0xA004	CIF PROFIBUS	Fehler beim lesen der CIF-Firmware-Info
0xA005	CIF PROFIBUS	Fehler beim lesen des CIF-Treiber-Status
0xA006	CIF PROFIBUS	Fehler beim schreiben einer Message auf die CIF-Karte
0xA007	CIF PROFIBUS	Fehler beim lesen einer Message von der CIF-Karte
0xA008	CIF PROFIBUS	Fehler bei PROFIBUS Diagnose (PG) CIF-Fehlercode
0xA009	CIF PROFIBUS	Fehler bei PROFIBUS Diagnose (SFC/SFB) CIF-Fehlercode
0xA010	CIF PROFIBUS	Kein CIF-Treiber installiert
0xA011	CIF PROFIBUS	CIF-PROFIBUS-Karte konnte nicht initialisiert werden
0xA012	CIF PROFIBUS	CIF-PROFIBUS-Karte nicht vorhanden
0xA013	CIF PROFIBUS	Busparameter für Slaves konnten nicht gesetzt werden
0xA014	CIF PROFIBUS	Parametrieren der CIF-PROFIBUS-Karte mit SDB-Daten fehlgeschlagen
0xA015	MPI®	MPI®-Telegrammfehler
0xA016	CIF PROFIBUS	Fehler bei der PG Kommunikation über PROFIBUS
0xA017	CIF PROFIBUS	Fehler bei der PG Kommunikation über PROFIBUS
0xA018	CIF PROFIBUS	Fehler bei der PG Kommunikation über PROFIBUS
0xA019	CIF PROFIBUS	Timeout beim E-Datenaustausch mit der CIF-Karte während SFC DPRD_DAT
0xA020	CIF PROFIBUS	Fehler beim E-Datenaustausch mit der CIF-Karte während SFC DPRD_DAT
0xA021	CIF PROFIBUS	Timeout beim A-Datenaustausch mit der CIF-Karte während SFC DPWR_DAT
0xA022	CIF PROFIBUS	Fehler beim A-Datenaustausch mit der CIF-Karte während SFC DPWR_DAT
0xA023	CIF PROFIBUS	CIF-PROFIBUS-Karte konnte nicht ermittelt werden
0xA024	CIF PROFIBUS	CIF-PROFIBUS-Karte konnte nicht initialisiert werden
0xA025	MPI®	MPI®-Telegrammfehler
0xA050	CIF 30/50/60	Kein CIF-Treiber installiert
0xA051	CIF 30/50/60	CIF-Karte konnte nicht initialisiert werden
0xA052	CIF 30/50/60	CIF-Karte nicht vorhanden
0xA053	CIF 30/50/60	Timeout während Kaltstartphase
0xA054	CIF 30/50/60	Anwendung konnte nicht in Betriebsmodus "READY" gesetzt werden
0xA055	CIF 30/50/60	Fehler beim E/A-Datenaustausch mit der CIF-Karte während PAE/PAA Aktualisierung
0xA056	CIF 30/50/60	Fehler beim E-Datenaustausch mit der CIF-Karte während T PE
0xA057	CIF 30/50/60	Fehler beim A-Datenaustausch mit der CIF-Karte während T PA
0xA058	CIF 30/50/60	Fehler beim lesen des CIF-Task-Modus

0xA060	CIF 30/50/60	Fehler beim lesen des CIF-Task-Modus
0xA061	CIF 30/50/60	Fehler beim Zugriff auf den WatchDog
0xA062	CIF 30/50/60	CIF-Treiber nicht im Betriebsmodus "COM"
0xA064	CIF 30/50/60	Fehler beim lesen des CIF-Treiber-Status
0xA065	CIF 30/50/60	Fehler beim lesen des CIF-Treiber-Status
0xA069	CIF 30/50/60	Timeout beim E/A-Datenaustausch mit der CIF-Karte während PAE/PAA Aktualisierung
0xA070	CIF 30/50/60	Timeout beim E-Datenaustausch mit der CIF-Karte während L PE
0xA071	CIF 30/50/60	Timeout beim A-Datenaustausch mit der CIF-Karte während T PA
0xA072	CIF 30/50/60	Timeout beim E-Datenaustausch mit der CIF-Karte während SFC DPRD_DAT
0xA073	CIF 30/50/60	Fehler beim E-Datenaustausch mit der CIF-Karte während SFC DPWR_DAT
0xA074	CIF 30/50/60	Timeout beim E-Datenaustausch mit der CIF-Karte während SFC DPRD_DAT
0xA075	CIF 30/50/60	Fehler beim A-Datenaustausch mit der CIF-Karte
0xA080	CIF 30/50/60	CIF-Treiber nicht bereit
0xA081	CIF 30/50/60	CIF-Treiber nicht im Betriebsmodus "RUN"
0xA082	CIF 30/50/60	CIF-Treiber nicht im Betriebsmodus "COM"
0xA100	TwinCAT I/O	Portnummer aus PLC43.INI, in TwinCAT-Treiber oder TwinCAT-Treiber selbst nicht vorhanden. Nachfolgender Wert ist Rückgabewert aus TCatIoOpen
0xA101	TwinCAT I/O	Fehler in PLC43.INI Peripheriestart < Bytestart
0xA102	TwinCAT I/O	Aktualisieren des PAE/PAA fehlgeschlagen
0xA103	TwinCAT I/O	Fehler in PLC43.INI Peripherie < Prozessabbild
0xA104	TwinCAT I/O	Timeout bei verbinden zu TwinCAT-Treiber Nachfolgender Wert ist Rückgabewert der letzten Funktion
0xA105	TwinCAT I/O	TwinCAT-Treiber nicht in RUN Nachfolgender Wert ist Rückgabewert der Funktion TCatGetState()
0xA106	TwinCAT I/O	TwinCAT-Treiber-Update ClearOutputs fehlgeschlagen (Treiber nicht mehr in RUN) Nachfolgender Wert ist Rückgabewert der Funktion TCatIoOutputUpdate
0xA107	TwinCAT I/O	TwinCAT-Treiber-Update PAE fehlgeschlagen (Treiber nicht mehr in RUN) Nachfolgender Wert ist Rückgabewert der Funktion TCatIoInputUpdate
0xA108	TwinCAT I/O	TwinCAT-Treiber-Update PAA fehlgeschlagen (Treiber nicht mehr in RUN) Nachfolgender Wert ist Rückgabewert der Funktion TCatIoOutputUpdate
0xA109	TwinCAT I/O	TwinCAT-Treiber-Update LoadPY fehlgeschlagen (Treiber nicht mehr in RUN) Nachfolgender Wert ist Rückgabewert der Funktion TCatIoInputUpdate
0xA110	TwinCAT I/O	TwinCAT-Treiber-Update StorePY fehlgeschlagen (Treiber nicht mehr in RUN) Nachfolgender Wert ist Rückgabewert der Funktion TCatIoOutputUpdate
0xA111	TwinCAT I/O	TCatIoGetInputPtr fehlgeschlagen
0xA112	TwinCAT I/O	TCatIoGetOutputPtr fehlgeschlagen
0xA113	TwinCAT I/O	Ads Kommunikationsthread konnte keine lokale Adresse beziehen

		und wurde Beendet; Keine ADS Kommunikation möglich Nächster Wert ist Rückgabewert der Funktion __AdsGetLocalAddress, wenn der Wert 0 ist, dann ist __AdsPortOpen fehlgeschlagen
0xA120	Softnet PN IO	Softnet Fehlercode der Funktion PNIO_set_mode folgt Softnet Fehlercode
0xA121	Softnet PN IO	Softnet Fehlercode der Funktion PNIO_controller_open folgt Index des Controllers Softnet Fehlercode
0xA122	Softnet PN IO	Callbackfehlercode für "mode change confirmation" folgt Softnet Fehlercode
0xA123	Softnet PN IO	Callbackfehlercode für "device activate confirmation" folgt Softnet Fehlercode
0xA124	Softnet PN IO	Softnet Fehlercode der Funktion PNIO_close folgt Softnet Fehlercode
0xA126	Softnet PN IO	Fehler beim Initialisieren des Softnet PN IO Treibers
0xA127	Softnet PN IO	Softnet Fehlercode der Funktion PNIO_set_mode operate folgt Softnet Fehlercode
0xA128	Softnet PN IO	Treiber braucht beim initialisieren zu lange um in Operate Betriebsmodus zu gelangen
0xA140	Addidata	Addidata Fehlercode folgt Nachfolgender Wert ist Rückgabewert einer ADDIPACK Funktion
0xA200	I/O Port	Keine I/O Port Zugriffsberechtigungen

6.4 Unterschiede S7-SoftSPS - Hardware-SPS

Im Gegensatz zur konventionellen, hardwarebasierten Steuerung verfügt die *S7-SoftSPS* über keine eigene Hardware. Sie muss immer in ein bestehendes Rechnersystem mit einem Windows® Betriebssystem eingebunden werden.

Da die *S7-SoftSPS* innerhalb eines Windows-Betriebssystems läuft, können andere Windows-Applikationen (z.B. EXCEL®, ACCESS®) ohne Hardwareverkabelung direkt und schnell auf die SPS-Operanden zugreifen.

Die Größe des Ladespeichers der *SoftSPS PLC S7-416* kann im Gegensatz zu einer Hardware-SPS flexibel geändert werden (bis zur max. verfügbaren Speichergröße des PC).

Unterschiede der *SoftSPS PLC S7-315* zur *SoftSPS PLC S7-416*

Der Unterschied der *SoftSPS PLC S7-315* zur *SoftSPS PLC S7-416* besteht in einer unterschiedlichen Anzahl von Merkern, Timern und Zählern entsprechend der Hardware-SPS.

Die Unterschiede entsprechen den Unterschieden der 315er und 416er Siemens Hardware SPS. Der Ladespeicher der *SoftSPS PLC S7-315* ist auf 256 kByte begrenzt, während der Ladespeicher der *SoftSPS PLC S7-416* frei konfigurierbar ist.

Nicht vorhandene Funktionen in der *S7-SoftSPS*

Da es physikalische Unterschiede zwischen einer Hardware-SPS und einem Windows-Rechner gibt, können nicht alle Funktionen der CPU S7-315 bzw. CPU S7-416 in der *S7-SoftSPS* realisiert werden. Es sollten jedoch Ihre vorhandenen Programme mit geringen Änderungen auf der *SoftSPS* lauffähig sein.

Die folgenden Organisationsbausteine werden z.Z. noch nicht unterstützt:

- OB 60: Mehrprozessoralarm
- OB 40 – OB 47: Prozessalarm 0 - 7

Zeit-OB's werden nicht durch Windows unterbrochen, d.h. diese OBs dürfen keine langen Durchlaufzeiten haben (<1ms).

Des Weiteren sind verschiedene SFB und SFC nicht realisiert, die auf einer PC Plattform nicht sinnvoll eingesetzt werden können. Es besteht jedoch die Möglichkeit die *S7-SoftSPS* um eigene, in C++ geschriebene SFCs zu erweitern.

Diagnosepuffer

Zusätzlich zu den normalen Diagnosepuffereinträgen, die beim Starten, bei Programmierfehlern, bei Bereichsüberschreitungen und bei fehlenden Bausteinen in den Diagnosepuffer eingetragen werden, gibt es weitere Einträge.

Diese werden von den Hardwaretreibern der *S7-SoftSPS* erzeugt und geben sehr exakt Aufschluss, über die Ursache eines eventuell auftretenden Problems. Bevor Sie sich mit unserem Support in Verbindung setzen, sollten Sie einen Auszug des Diagnosepuffers mit dem Programmiersystem speichern.

So kann ein eventuelles Problem leicht lokalisiert werden.

S7-SoftSPS als Dienst starten

Um Zugriffe nicht privilegierter Benutzer zu verhindern, kann die SoftSPS als Dienst gestartet werden. Hierzu muss die SoftSPS in die Liste der Dienste eingetragen werden. Öffnen Sie hierzu die Kommandozeile und wechseln Sie in das Installationsverzeichnis der S7-SoftSPS.

Geben Sie an der Kommandozeile folgenden Befehl ein:

```
C:\SoftSPS>plc416.exe -Service
```

Es darf keine Fehlermeldung erscheinen. Die SoftSPS ist nun in die Liste der Dienste eingetragen. Über „Verwaltung – Dienste“ findet sich in der Liste nun der Dienst „CPUS7“. Hier kann nun die Startart auf „Automatisch“ gestellt werden, damit die SoftSPS auch ohne einen am PC angemeldeten Benutzer startet. Auch ist es so möglich, dass Benutzerkonten mit eingeschränkten Rechten die SoftSPS verwenden können.

USV Betrieb

Die meisten USV Programme bieten eine Möglichkeit, bestimmte Tätigkeiten vor dem Herunterfahren des PCs erledigen zu lassen. In aller Regel kann man in deren Konfigurationsdialogen Programme oder Batchdateien angeben, die vor dem Herunterfahren ausgeführt werden sollen.

Wird hier die Anwendung „PlcStop.exe“ (Im *S7-SoftSPS* Installationsverzeichnis) eingetragen, speichert die SPS das Programm und im Anschluss wird die *S7-SoftSPS* ordnungsgemäß beendet.

Unbefugte Zugriffe

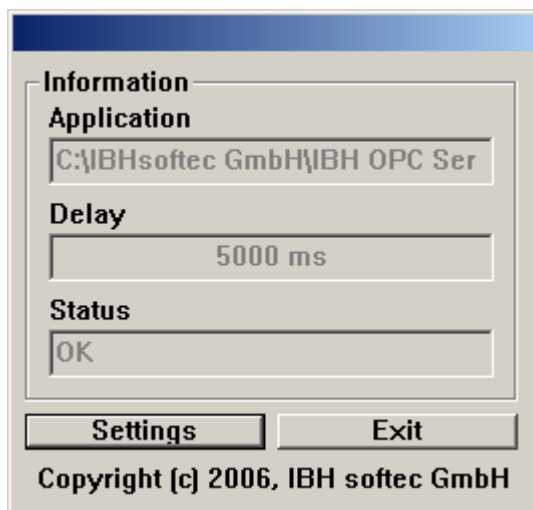
Um unbefugte Zugriffe auf den PC zu verhindern, kann die *S7-SoftSPS* naturgemäß keine Sicherheitsmechanismen zur Verfügung stellen. Hier sind vom Anwender auf die Applikation zugeschnittene, geeignete Maßnahmen zu treffen.

6.5 AppStarter

Mit dem Applikationsstarter ist es möglich Software Applikationen wie z.B. die SoftSPS oder einen OPC Server in einer vom Anwender bestimmten Reihenfolge und Verzögerung zu starten. Der Applikationsstarter wird mit der Datei „AppStarter.exe“ aus dem Installationsverzeichnis der SoftSPS gestartet.

ACHTUNG: DER APPLIKATIONSSTARTER ERFORDERT ZUR KONFIGURATION EINE WINDOWS BENUTZERANMELDUNG MIT ADMINISTRATIONSRECHTEN!

Sobald die AppStarter.exe ausgeführt wird, erscheint der Informationsdialog.



Application

Zeigt welche Applikation als Nächstes gestartet wird.

Delay

Zeigt die verbleibende Zeit in Millisekunden bis zum Start der nächsten Applikation an.

Status

Zeigt den Status der zuletzt gestarteten Applikation an.

Settings

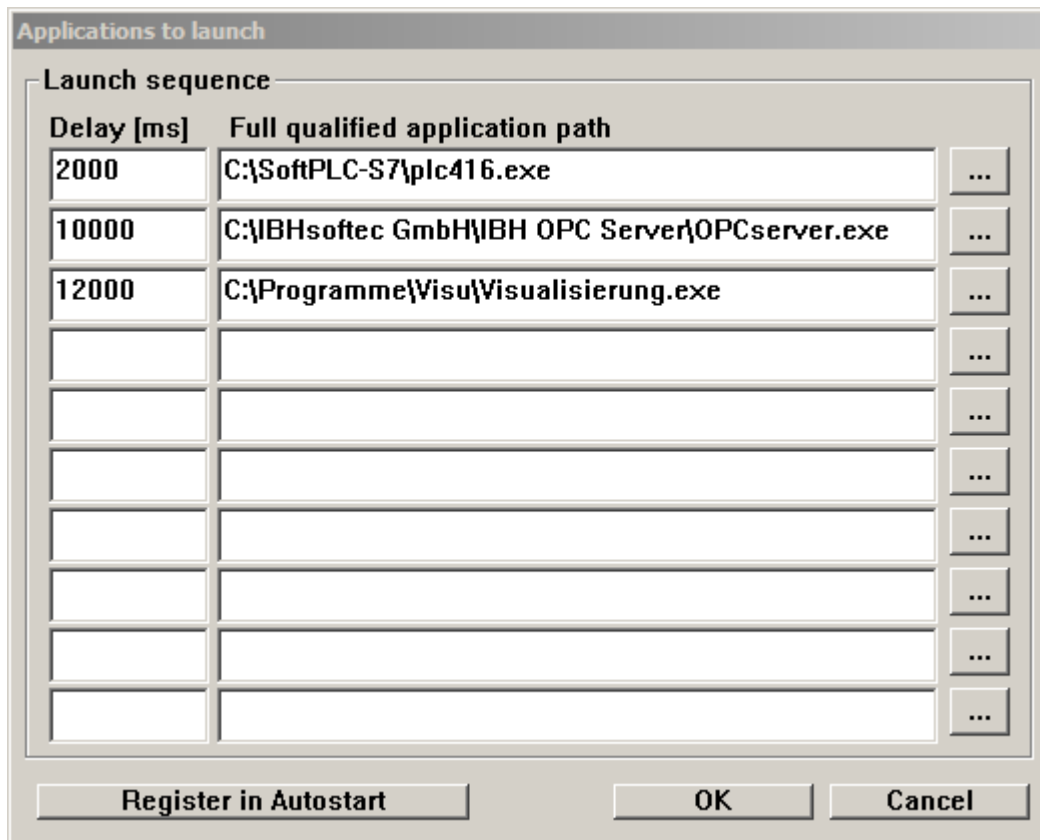
Öffnet den Einstellungsdialog.

Exit

Beendet den Applikationsstarter.

Einstellungen des AppStarters

Im Einstellungsdialog wird der Applikationsstarter konfiguriert.



Launch sequence

Die „Launch sequence“ gibt die Reihenfolge der zu startenden Applikationen an. Die Applikationen werden, oben beginnend, in der Reihenfolge, in der sie angelegt sind, gestartet.

Delay [ms]

Zeit in Millisekunden bis zum Start der Applikation. Die Zeit bis zum Start der ersten Applikation läuft ab, sobald der Applikationsstarter gestartet wird. Die Zeit bis zum Start der nächsten Applikation läuft ab, sobald die vorherige Applikation gestartet wurde.

Full qualified application path

Manuelle Angabe des kompletten Pfades und der Datei, die ausgeführt werden soll, um die gewünschte Applikation zu starten.

...

Browsingfunktion für die Datei, die ausgeführt werden soll, um die gewünschte Applikation zu starten.

OK

Übernimmt die Einstellungen und beendet den Einstellungsdialog.

Cancel

Verwirft die Änderungen und beendet den Einstellungsdialog.

HINWEIS: „REGISTER IN AUTOSTART“ UND „REMOVE FROM AUTOSTART“ WIRD SOFORT, UNABHÄNGIG VON OK/CANCEL, ÜBERNOMMEN!

Register in Autostart

Die Einstellungen werden in der Windows-Registry abgespeichert. Unmittelbar nach dem Start von Windows wird nun auch der Applikationsstarter und somit die angegebenen Applikationen gestartet.

Remove from Autostart

Die Einstellungen werden aus der Windows-Registry entfernt. Der Applikationsstarter wird beim Windows Neustart nicht mehr automatisch ausgeführt.

Windows-Registry Eintrag:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

6.6 Null-Modemkabel

Sub.-D Gehäuse			Sub.-D Gehäuse	
Buchse			Buchse	
9 Pin	25 Pin		9 Pin	25 Pin
1	-----	----- keine Verbindung -----	1	-----
2	3	-----	3	2
3	2	-----	2	3
4	20	-----	6	6
5	7	-----	5	7
6	6	-----	4	20
7	4	-----	8	5
8	5	-----	7	4
9	-----	----- keine Verbindung -----	9	-----
Gehäuse	Gehäuse	----- Schirm -----	Gehäuse	Gehäuse