



S7-Kompakt-SPS Handbuch



S7-Panel-SPS



S7-Kompakt-SPS



S7-Panels/HMI



Peripherie



Software



Energiemanagement

Inhaltsverzeichnis

Allgemeine Hinweise.....	8
Über INSEVIS.....	9
Produktfamilie S7-Kompakt-SPS.....	10
Kommunikationsübersicht.....	10
S7-CPU's in den Kompakt-SPS.....	11
CPU-V.....	11
CPU-T.....	12
Abgesetzte Bedieneinheiten für die Kompakt-SPS.....	13
Peripherie für die Kompakt-SPS.....	14
Kommunikation SPS zu Peripherie.....	14
Zubehör S7-Kompakt-SPS.....	16
Beschaltung der CPU-V.....	17
Beschaltung der CPU-T.....	18
Allgemeines.....	19
Speicherorganisation.....	19
Kommunikation.....	20
Ethernet.....	20
IP-Adresse im Simatic®-Manager einstellen.....	21
Anlegen der CPU und der Ethernet-Verbindung im Simatic®-Manager.....	23
IP Konfiguration der SPS mit dem SFB 129 "IP_CFG".....	24
S7 Aktive Verbindungen zu Partner-SPSen (RFC1006).....	25
Daten lesen von einer Partner-SPS mit SFB 120 "GET".....	25
Daten schreiben an eine Partner-SPS mit SFB 121 "PUT".....	27
Lesen der IP-Konfiguration mit dem SFB 131 "IP_GET" (CPU-T).....	29
Schreiben der IP-Konfiguration mit dem SFB 132 "IP_SET" (CPU-T).....	30
TCP Send/Receive-Verbindung (RFC 793).....	31
Daten senden über TCP mit dem SFB 122 "TSEND".....	31
Daten empfangen über TCP mit dem SFB 123 "TREC".....	33
Beenden der TCP-Verbindung mit dem SFB 124 "TDISCON".....	35
Konfiguration der Partnerparameter bei TCP mit dem SFB 125 "TCONFIG".....	36
Statusabfrage bei der TCP-Verbindung mit dem SFB 126 "TSTATUS".....	37
ISO on TCP.....	38
Modbus-TCP (Client).....	39
Modbus TCP (Server).....	41
UDP Send/Receive-Verbindung (RFC 768).....	41
Daten senden über UDP mit dem SFB 127 "USEND".....	42
Daten empfangen über UDP mit dem SFB 128 "URECV".....	43
Profinet IO Controller.....	46
Konfiguration im SimaticManager.....	46
Konfiguration im TIA-Portal.....	48
Status-LEDs.....	50
Profibus DP.....	51
Profibus DP-Master.....	51
Konfiguration des DP-Master.....	51
Profibus DP-Slave.....	52
Konfiguration des DP-Slaves.....	52
DP Master Konfiguration für INSEVIS Profibus DP-Slave.....	54
Struktur der DP-Slave Diagnosetelegramme.....	55
DP-Slave Adressierung mit SFC 75 „SET_ADDR“.....	58
DP-Slave-Adress- und Baudrateneinstellung mit SFC 76 „SET_DPS“.....	59
Profibus DP.....	60
Profibus DP-Master.....	60
Konfiguration des DP-Master.....	60
Profibus DP-Slave.....	61
Konfiguration des DP-Slaves.....	61
DP Master Konfiguration für INSEVIS Profibus DP-Slave.....	63
Struktur der DP-Slave Diagnosetelegramme.....	64
DP-Slave Adressierung mit SFC 75 „SET_ADDR“.....	67
DP-Slave-Adress- und Baudrateneinstellung mit SFC 76 „SET_DPS“.....	68
UART.....	69
Senden von Daten mit SFB 60 "SEND_PTP", freies ASCII-Protokoll.....	69
Empfang von Daten mit SFB 61 "RCV_PTP", freies ASCII-Protokoll.....	70
Senden von Daten mit SFB 60 "SEND_PTP", Modbus RTU (Client).....	71
Empfang von Daten mit SFB 61 "RCV_PTP", Modbus RTU (Client).....	72
Modbus RTU (Server).....	73
Empfang über RS232 mit Endeerkennung.....	74
Bearbeiten der UART-Konfiguration mit SFB 130 "CFG_PTP".....	74

CAN.....	75
CANopen®-Konfiguration mit der ConfigStage.....	75
Freie CAN-Programmierung (Expertenlevel).....	77
Senden von CAN-Telegrammen SFB105.....	77
Empfang von CAN-Telegrammen SFB106.....	77
Empfangsereignis OB106.....	77
CANopen - Hilfsfunktionen (utilities).....	78
Lesen und Schreiben von CANopen-Variablen (Objekten) per SDO-Transfer SFB107.....	78
FB107 SDO upload.....	78
FB108 SDO download.....	78
FB109 SDO upload.....	78
FB110 SDO download.....	79
FB111 SDO upload.....	79
FB112 SDO download.....	79
FB113 SDO Transfer String.....	79
Alternative Funktion des SFB107.....	79
CAN Statusabfrage SFB114.....	80
Fehlerereignis OB186.....	80
Automatische Wiederkehr.....	80
Konfiguration mit der Software „ConfigStage“.....	82
Grundlegende Einstellungen.....	82
Änderung der Geräte-IP-Adresse.....	83
Wechsel des Zielgerätes.....	83
Adressierung der Onboard-Peripherie.....	83
Standardadressierung in den INSEVIS- SPSen.....	83
Abweichende Adressierungen in SPSen und Adressierungen von dezentraler Peripherie.....	84
CPU-Einstellungen.....	84
WebServer.....	86
Uhrzeiteinstellungen (Zeitzone, Sommerzeit, NTP-Server).....	86
Kommunikationseinstellungen.....	87
RS232 und RS485.....	87
Ethernet.....	87
Profinet.....	89
CAN-Einstellungen.....	89
Dezentrale INSEVIS Peripherie.....	89
Dezentrale Fremdperipherie manuell konfigurieren.....	90
Bibliothekselemente aus eigenen CAN-Slaves erstellen.....	91
Dezentrale Fremdperipherie per EDS konfigurieren.....	92
Dezentrale Fremdperipherie - vordefiniert.....	93
Allgemeine Beispiele für CANopen Fremdgeräte.....	94
Allgemeine Beispiele für CANopen Fremdgeräte.....	97
Systemfunktionen.....	99
Organisationsbausteine (OB).....	99
Organisationsbausteine (OB) - kompatibel zu STEP®7 von Siemens.....	99
Organisationsbausteine (OB) - in Erweiterung zu STEP®7 von Siemens.....	99
Systemfunktionsbausteine (SFB).....	99
Systemfunktionsbausteine (SFB) - kompatibel zu STEP®7 von Siemens.....	99
Systemfunktionsbausteine (SFB) -.....	99
die bei der INSEVIS-CPU noch NICHT vorhanden und nur auf Anfrage erhältlich sind.....	99
Systemfunktionsbausteine (SFB) - in Erweiterung zu STEP®7 von Siemens.....	100
Systemfunktionen (SFC).....	100
Systemfunktionen (SFC) - kompatibel zu STEP®7 von Siemens.....	100
Systemfunktionen (SFC) -.....	101
die bei der INSEVIS-CPU noch NICHT vorhanden und nur auf Anfrage erhältlich sind.....	101
Systemfunktionen (SFC) - in Erweiterung zu STEP®7 von Siemens.....	101
Import der INSEVIS-Zusatzbausteine in das S7-Projekt.....	102
Installation der INSEVIS-S7-Bibliothek im Simatic®-Manager.....	102
Installation der INSEVIS-S7-Bibliothek im TIA-Portal®.....	103
Übersicht über SFCs zur Datensicherung und -wiederherstellung.....	104
Datenbausteine nach ROM/ ins Flash sichern.....	104
Life-Backup für Geräte austausch → vom Gerät auf Micro-SD-Karte.....	104
Manuelles Programm-Backup für Gerätehersteller → von ServiceStage auf PC.....	104
Manuelles Programm-Restore auf einer SD-Karte erstellen.....	104
S7-Programm-Restore für beide Backup-Methoden.....	104
Daten archivieren /dearchivieren (zurücklesen).....	105
Daten zurücklesen aus am PC editierter CSV-Archivdatei.....	105
Archivieren von Daten mit dem SFC 207 "ARCHIVE".....	105
Backup mit dem SFC 208 "BACKUP".....	110
Rücklesen von Daten von Micro-SD-Karte mit SFC 209 "READFILE".....	112

DB kopieren vom Arbeits- in den Ladespeicher mit SFC 210 "COPY DBL"	113
DB kopieren vom Lade- in den Arbeitsspeicher mit SFC 211 "COPY DBW"	115
Daten lesen aus csv-formatierter Archivdatei (SD) mit SFC 213 "READ_CSV"	117
Übersicht über SFCs für ein DateiManagementSystem	119
Erstellen von Ordnern auf Micro-SD mit SFC 223 "CREATE"	120
Kopieren von Dateien auf Micro-SD mit SFC 222 "COPY"	121
Umbenennen von Ordnern/Dateien auf Micro-SD mit SFC 220 "RENAME"	122
Informationen zu Ordnern/ Dateien auf Micro-SD mit SFC 224 "INFO"	123
Abfrage von Dateigröße und -name auf Micro-SD mit SFC 225 "FILEINFO"	124
Löschen von Ordnern/Dateien auf Micro-SD mit SFC 221 "DELETE"	125
LCD-Hinterleuchtungseinstellung mit dem SFC 214 "LCD BRIG"	126
Von RAM nach ROM kopieren mit dem SFC 254 "RAM2ROM"	128
Betriebssystem in der SPS updaten	129
Hardwareseitiges Urlöschen	130
Remotезugang mit der Software „RemoteStage“	131
Allgemeines	131
Auswahl der Remote-Steuerung	131
Eingabe des Passwortes	132
Benutzen und Einstellen des Remote-Bildschirmes	132
Abspeichern der remote (auf dem PC) archivierten Daten	132
Einlesen, Konvertieren und Abspeichern der Archivdaten	133
Ereignis- oder Alarmarchivdarstellung und -export	133
Trenddarstellung und -export	134
Rezepturhandling	135
Datenarchivhandling	136
Kommandozeilenbefehle für RemoteStage - Visualisierungsmodus	137
Automatisches Verbinden zum Remote-Gerät nach Start	137
Automatisches Wiederverbinden zum Remote-Gerät nach Verbindungsabbruch	137
Einstellen der TCP-Portnummer für die S7-Kommunikation	137
Vollbildschirm nach Start	137
Kommandozeilenbefehle für RemoteStage - Archivierungsmodus	138
Lesen (Upload) der Meldungs- (Alarmer und Ereignisse) und Trendarchive	138
Lesen (Upload) der Rezepturarchive vom Remote-Gerät	139
Zurückschreiben (Download) der Rezepturarchive in das Remote-Gerät	140
Lesen (Upload) der Daten (DB-) Archive vom Remote-Gerät	141
Zurückschreiben (Download) der Daten (DB-) Archive in das Remote-Gerät	141
Rückgabewerte	142
Arbeiten mit der Software „ServiceStage“	143
Funktionsübersicht	143
Allgemeine Informationen	143
Betriebszustand	143
Datum und Uhrzeit	143
Speicher	144
Diagnosepuffer	144
Block Update	144
Backupdateien erstellen (Online Backup)	145
Know-how-Schutz	145
Beobachtungstabelle	146

Änderungen zu vorhergehenden Versionen des Handbuches

Rev. 02 / 2012:

neu: SFB130 (Modbus), SFC75 (Profibus DP Slave), SFC254 (RAM nach ROM)
geändert: SFB125/126 (TCP)-Beschreibung, SFB127/128 (UDP)-Beschreibung, Profibus-Beschreibung, SFB/SFC-Übersicht

Rev. 03 / 2012:

geändert: SFB60/61 (UART) Parameterbezeichnung, Kapitel Alarmhandling, Beschreibung SFBs 122/127 korrigiert, ConfigStage komplett neu

Rev. 04 / 2012:

geändert: Beschreibung der Profibus-Signale an Siemens-Beschreibung angepasst, Zykluszeitenbeschreibung erweitert (Kapitel „ConfigStage“)

Rev. 01 / 2013:

neu: IP-Adresse ändern mit ConfigStage 1.0.14.5 und BS 2.0.35
geändert: Beschreibung Ethernet UPD-Empfangen (SFB128), Profibus-Slave Subnetze

Rev. 02 / 2013:

neu: ServiceStage beschrieben

Rev. 03 / 2013:

geändert: Speicheraufteilung beschrieben, CAN überarbeitet

Rev. 04 / 2013:

geändert: Beschreibung von ConfigStage: PDO-/ SDO-Mapping, Adressübersichtsliste (ab ConfigStage 1.0.14.7)

Rev. 05 / 2013:

neu: neue Remote-Funktionen; (DB-) Archivierungen auslesen/ speichern (Ab RemoteStage 1.0.3.7), neue SPS-Funktionen: SFC207: Archivierung von DBs , (ab SPS-BS 2.0.41 und HMI-BS 1.0.11)

Rev. 06 / 2013:

neu: ServiceStage 1.0.1.1 mit Know-how-Schutz, ConfigStage 1.14.9 mit EDS-Import, SFC208 Backup&Restore ab Betriebssystem 2.1.0 und S7-Lib 2_1_0 und ab HMI-Firmware 1.1.0
geändert: Beschreibung CAN SFB114 ergänzt

Rev. 07 / 2013:

geändert: Beschreibung S7-Lib in Projekt einbetten in Systemfunktionen verschoben

Rev. 08 / 2013:

neu: ConfigStage 1.0.14.10: Wechsel des Zielgerätes eingebracht
geändert: Beschreibung UART SFB60/61 verbessert

Rev. 01 / 2014:

neu: SPS-Firmware 2.1.5: SFC209 READFILE, SFC2010 LOAD_DBL, SFC2011 LOAD_DBW neu
geändert: Backup&Restore- Zusammenfassung bei Systemfunktionen erstellt

Rev. 02 / 2014:

geändert: S7-Programmierbeispiele eingefügt für alle SFCs und die Ethernet-SFBs, diese Programmzeilen können direkt in den SimaticManager reinkopiert werden, Hinweise auf VIDEOS eingefügt

Rev. 03 / 2014:

geändert: Verbesserung der Beschreibung zu SFB 120 Get und SFB 121 Put in einer Verbindung, Verbesserung der Beschreibung zu SFB 208 (Ablage der Passwortdatei)

Rev. 01 / 2015:

geändert: ConfigStage: Beschreibung TSAPs verbessert, änderbar ab Firmware 2.2.1 und ConfigStage 1.0.14.15

Rev. 02 / 2015:

geändert: Systemfunktionen: Erweiterung der Übersicht OBs, SFBs, SFCs, Erläuterung zur Erstellung WLD-Datei ausführlicher, UART: Empfang RS232 mit Ende-Erkennung beschrieben

Rev. 03 / 2015:

neu: CPU-T-Geräte mit aufgenommen, Anpassungen (Erweiterungen) für CPU-T-Funktionen vorgenommen,
geändert: CAN: SFB107 + automatische Wiederkehr beschrieben, neue ServiceStage-Funktion Firmwareupdate mit ETH, WLD-Erstellen beschrieben
gilt ab Firmware 2.3.0 (CPU-V/P/T), ConfigStage 1.0.14.21, ServiceStage 1.0.1.5, RemoteStage 1.0.4.13

Rev. 01 / 2016:

neu: Profinet-Handling (bei CPU-T-Geräten) eingefügt
geändert: Systemfunktionen: SFC208-Beschreibung verbessert, Bild eingefügt in manuelles Backup , allgemeine Fehlerbereinigung

Rev. 02 / 2016:

neu: SFC214 „LCD_BRIG“ für Regelung der Helligkeit durch S7-Programm
geändert: Systemfunktionen: manuelles Backup Beschreibung geändert

Änderungen zu vorhergehenden Versionen des Handbuches

Rev. 03 / 2016:

neu: Datensätze ändern am PC, zurückschreiben und löschen möglich, neue Up- und Downloads im Kommandozeilenmodul der RemoteStage, Online-Backup bei ServiceStage, ab SPS-Firmware 2.3.9 (alle CPUs)

Rev. 04 / 2016:

geändert: RemoteStage: Beschreibung bei Download Rezepturen und Datenarchive in Remote-Gerät verbessert,
ServiceStage: Beschreibung Online Backup erweitert

Rev. 05 / 2016:

geändert: Systemfunktionen: Beschreibung Hardware-Reset erweitert, ServiceStage: Beschreibung Online Backup nochmal erweitert
Grafikdesign geändert

Rev. 06 / 2016:

neu: ConfigStage 1.0.14.32: WebServer für alle CPU-T-SPSen, ab Firmware 2.4.6 (Kompakt-/Panel-SPS mit CPU-T

Rev. 01 / 2017:

neu: SFC215 zum Wechseln des Benutzer-Levels, ab Firmware 2.4.8 für Kompakt-SPS mit CPU-T für Nutzung
in Panels mit CPU-T ab Firmware 1.3.5,
SFC75 Einstellen von Profibus-Slave-Adresse UND Baudrate für Kompakt-SPSen mit CPU-V/-P ab BS 2.4.5

Rev. 02 / 2017:

neu: Ethernet; Info zu Kommunikationseinstellungen mit S7-1200/1500 - Einstellungen für TIA-Portal erklärt
geändert: CAN-Kapitel: Tabelle mit Leitungslängen / Baudraten eingebracht

Rev. 01 / 2018:

geändert: RemoteStage-Beispiele für Batchfiles erweitert

Rev. 02 / 2018:

neu: SFC 220...224 für DateiManagementSystem auf Micro-SD-Karte ergänzt (Systemfunktionen)

Rev. 01 / 2019:

neu: SFBs 131 und SFB132 zum getrennten Einstellen der Ethernet-Schnittstellen bei CPU-T
geändert: Screenshots angepasst auf neues Design der 2019er Stages und Windows 10

Rev. 02 / 2019:

neu: Fester und variablen lokaler Port bei Ethernet-TCP, Firmware: CPU-V/P 2.4.9 / CPU-T 2.6.2, ConfigStage ab 1.0.14.39

Rev. 03 / 2019:

neu: SFC225: „FILEINFO“ für SPSen mit CPU-T., Firmware: CPU-T 2.6.6

Rev. 04 / 2019:

geändert: Profinet-Konfiguration mit TIA-Portal hinzugefügt

Rev. 01 / 2020:

neu: SFB52, 53, 54 hinzugefügt, Zeiteinstellungen, Sommerzeit, NTP-Server hinzugefügt, ConfigStage 1.0.14.41, Firmware 2.7.2

Rev. 01 / 2021:

geändert: Profinet-LED-Bedeutung präzisiert, Modbus Server Konfiguration angepasst

Rev. 01 / 2022:

geändert: Lösehebelstecker Stecker als abgekündigt deklariert, Produktbilder Panel HMI erneuert, Fehlerbehebungen

Rev. 01 / 2023

neu: VisuStage 2.2.0.1 NEU: Bildbausteine, erfordert RemoteStage 1.0.4.51

Rev. 01 / 2024:

geändert: Ergänzungen zu Backup und Restorefunktionen, sowie dem Restore von SD-Karte nach dem Utlöschen.

Rev. 01 / 2024:

neu: ServiceStage 1.0.4.0 Beobachtungstabelle für Variablen, Systemfunktionen SFC207 Informationen hinzugefügt

Rev. 03/ 2024:

INSEVIS Vertriebs GmbH

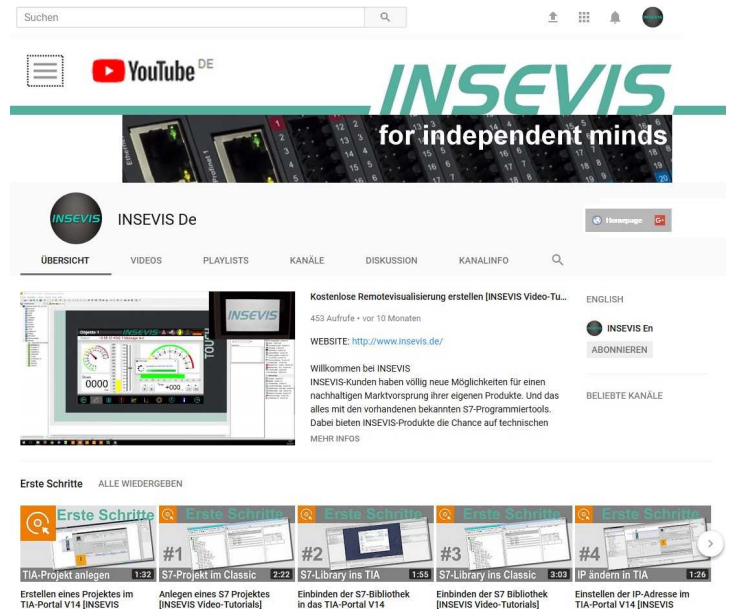
Rev. 01/ 2025:

INSEVIS Modbus TCP Client und ISO on TCP

Hinweis zum besseren Verständnis durch Videoerklärungen

Auf dem deutschen **INSEVIS De** Youtube-Kanal stehen für sie Tutorial Videos für einzelne Themen dieses Handbuchs in Playlists geordnet bereit.

Bitte nutzen Sie diese als Ergänzung zum vorliegenden Handbuch. So können Sie sich noch leichter mit den INSEVIS-Funktionen vertraut machen.



Allgemeine Hinweise

Hinweise zur Sicherheit

Dieses Handbuch beinhaltet Hinweise, die zur Vermeidung von Sachschäden und zu Ihrer persönlichen Sicherheit beachten werden müssen. Diese Hinweise sind durch ein Warndreieck mit Ausrufezeichen sowie einem Signalwort gekennzeichnet.



- Gefahr** Tod, schwere Körperverletzungen oder Sachschäden werden eintreten, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.
- Warnung** Tod oder schwere Körperverletzungen oder Sachschäden können eintreten, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.
- Vorsicht** Körperverletzungen oder Sachschäden können eintreten, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.
- Achtung** bedeutet, dass ein unerwünschtes Ergebnis oder Zustand eintreten kann, wenn der entsprechende Hinweis nicht beachtet wird.
- Wichtig** bedeutet die Verpflichtung zu einem besonderen Verhalten oder einer Tätigkeit für den sicherheitsgerechten Umgang mit der Steuerung / Maschine.

Qualifiziertes Personal

Die in diesem Handbuch beschriebenen Geräte dürfen nur in Verbindung mit dieser Dokumentation eingerichtet und betrieben werden. Installation, Inbetriebnahme und Betrieb der Geräte dürfen nur von qualifiziertem Personal (fachlich ausgebildete Personen, die die Berechtigung nachgewiesen haben, Geräte, Systeme und Stromkreise nach allgemeinen gültigen Standards in Betrieb zu nehmen, zu erden und zu kennzeichnen) vorgenommen werden.

Bestimmungsgemäßer Gebrauch

Das Gerät darf nur für die in der Dokumentation vorgesehenen Einsatzfälle und nur in Verbindung mit zugelassenen Fremdgeräten und -komponenten verwendet werden. Der einwandfreie und sichere Betrieb des Produktes setzt sachgemäßen Transport, Lagerung, sowie Aufstellung, Montage und Wartung voraus. Die im jeweiligen Einsatzgebiet geltenden Sicherheits- und Unfallverhütungsvorschriften sind zu einzuhalten. Das Netzteil ist an einem zentralen Massepunkt (sternförmig) anzuschließen.



Instandhaltung

Modifikationen/ Reparaturen an von INSEVIS gelieferten Geräten dürfen nur von durch INSEVIS-Personal geschulten Fachkräften oder von INSEVIS-Personal selbst in EMV-gerechter Umgebung durchgeführt werden. Jegliche unautorisierten Änderungen können zu Schäden führen. Bei unautorisierter Öffnung der Geräte erlischt die Gewährleistungsverpflichtung von INSEVIS.



Hinweise zur Datensicherheit

Jeder Kunde muss die technischen Mittel zur Absicherung gegen illegales Eindringen in sein Ethernet-Netzwerk selbst bestimmen, installieren und pflegen. INSEVIS ist nicht verantwortlich für Schäden, die aufgrund ungenügender Absicherung gegen Eindringen unberechtigter Dritter in das Ethernet-Netzwerk entstehen. Im Zweifelsfall fragen Sie einen externen vertrauenswürdigen IT-Spezialisten nach den erforderlichen Schutzmaßnahmen für Ihr IT-System.

Copyright

Diese Dokumentation sowie sämtliche gelieferte oder auf den INSEVIS-Webseiten zum Download bereitgehaltene Dokumentation und Software sind urheberrechtlich geschützt. Die Vervielfältigung dieser Dokumentation in irgendeiner Art und Weise ohne ausdrückliche Genehmigung der Firma INSEVIS GmbH ist nicht erlaubt. Die Eigentums- und Urheberrechte an der Dokumentation und Software und jeder der von Ihnen erstellten Kopie bleiben der INSEVIS GmbH vorbehalten.

Marken

INSEVIS weist darauf hin, dass die in der Dokumentation verwendeten Markennamen der jeweiligen Firmen wie - STEP®, SIMATIC®, TiA-Portal® und andere als eingetragene Warenzeichen der Siemens AG. - Windows® und andere als eingetragene Warenzeichen der Microsoft AG - CANopen® und andere als eingetragene Warenzeichen der CAN in Automation eG und weitere eingetragene Warenzeichen den jeweiligen Inhabern gehören und als solche dem allgemeinen markenrechtlichen Schutz unterliegen.

Haftungsausschluss

Alle technischen Angaben in dieser Dokumentation wurden von der INSEVIS GmbH mit größter Sorgfalt erstellt. Dennoch können Fehler nicht ganz ausgeschlossen werden, so dass INSEVIS keine Gewähr für die vollständige Richtigkeit übernimmt. Die Dokumentation wird regelmäßig überprüft, nötige Korrekturen werden in nachfolgenden Revisionen berücksichtigt.

Erforderliche Grundkenntnisse und Geltungsbereich des Handbuchs

Für das Verständnis dieser Dokumentation sind grundlegende Kenntnisse der Automatisierungstechnik allgemein sowie der Programmierung unter STEP7 nötig. Diese Dokumentation gilt für alle S7-Kompakt-SPSen der INSEVIS Serie „CCxxxx“ mit CPU-V ab Hardware-AS 3.0 und Firmware 2.0.0. und mit CPU-T ab Hardware-AS 2.0 und Firmware 2.3.0

Über INSEVIS

S7-Systemkomponenten für die industrielle Automatisierungstechnik

Die INSEVIS- Produktpalette ermöglicht eine durchgehende, einfach zu realisierende Steuerungslösung für kleine und mittlere Anwendungen in aktuellster Technologie, äußerst hoher Qualität und mit leicht zu konfigurierenden zusätzlichen Anbindungen wie z.B. CANopen® und Modbus.

Die einfache Integration der INSEVIS-Produkte in die S7-Welt gilt mittlerweile als vorbildlich. Komplexe Kommunikationseinstellungen werden einfach und intuitiv grafisch vorgenommen, so dass damit die S7-Welt problemlos erweitert werden kann. Eine umfangreiche, multilinguale Visualisierung in einem zeitgemäßen Design ist mit wenigen Handgriffen und bekannten Abläufen erstellt, simuliert und auf der Anlage jederzeit remote erreichbar.

Die S7-CPU's -V und -P bilden die Basis für die erfolgreichen INSEVIS Produktfamilien mit Profibus DP Master/Slave. Mit der S7-CPU-T stehen Panel-SPS und Kompakt-SPSen mit VNC- und Webserver und mit Profinet IO Controller zur Verfügung.

Step®7-Programmierbarkeit

Die INSEVIS-S7-CPU's sind mit STEP 7® - AWL, KOP, FUP, S7-SCL, S7-Graph von Siemens zu programmieren und im wesentlichen befehlskompatibel zur CPU S7-315-2PNDP. Eigene Bausteine erweitern den Funktionsumfang und erlauben dadurch besondere Lösungen. Daher erfolgt die S7-Programmierung immer mit dem bekannten Siemens-Tools SIMATIC®-Manager oder mit dem TIA-Portal® (oder kompatiblen).

Unabhängigkeit

INSEVIS-Produkte basieren nicht auf Windows oder Linux, sondern haben eine eigene Firmware. Damit kann die Hard- und Software exakt auf minimale Leistungsaufnahme und optimalen Ablauf von Steuer- und Regelungsalgorithmen abgestimmt werden. Bootzeiten von weniger als 4 Sekunden, der völlige Verzicht auf Run-Time-Lizenzen und Stromaufnahmen von <100mA @ 24V sind das Ergebnis.

Lizenzfreiheit der Software

INSEVIS steht für klare, ehrliche Lizenzpolitik, die Kunden nachhaltige Kostenvorteile verschafft. Die Hoheit über das BIOS, die eigene Firmware und eine eigene Zusatzsoftware erlauben INSEVIS die klarste Form der Lizenzpolitik: den vollkommen Verzicht auf Lizenzen. Daher bietet INSEVIS seinen Kunden das komplette Softwarepaket kostenlos an. Lediglich für die Firmenlizenz der umfangreichen Vollversion der Visualisierungssoftware wird einmalig ein kleiner Einmalbetrag erhoben.

Quality-Made in Germany

Entwicklung, Leiterplattenfertigung, Bestückung, Test und Montage aller INSEVIS-Produkte - alles erfolgt in Deutschland. Teils im Stammhaus, teils bei zertifizierten und sorgfältig ausgewählten Fertigungspartnern. Damit wird jedes Produkt eine Verbindung deutscher Ingenieurskunst mit Wirtschaftlichkeit und mit einem Ursprungszeugnis mit Präferenzursprung Deutschland versehen.



INSEVIS betreibt ein jährlich zertifiziertes Qualitätsmanagementsystem nach DIN EN ISO 9001:2015.

Diesen Qualitätsgedanken verpflichten sich auch alle Lieferanten von INSEVIS und tragen so mit zu dem hohem Qualitätsniveau der INSEVIS-Produkte bei.

Bereits bei der Entwicklung der INSEVIS-Produktfamilien stand vor den technologischen Aspekten das Ziel, höchste Qualität und Ergonomie in die Produkte fest zu integrieren.

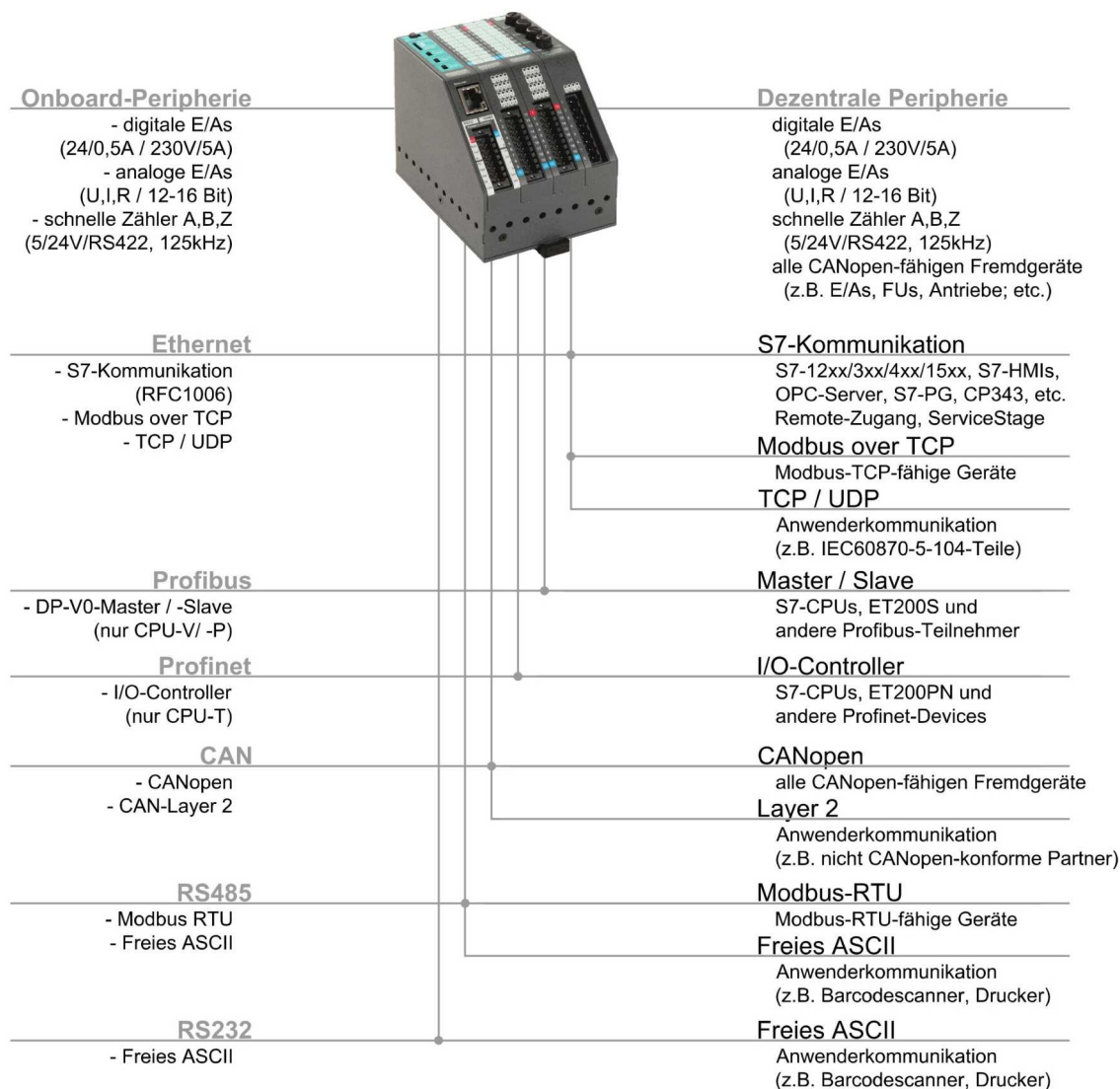
Sämtliche Produkte wurden umfangreichen Tests unterzogen, bevor sie in ausgewählten und zertifizierten Fertigungen in Deutschland hergestellt werden.

INSEVIS - Made in Germany

Produktfamilie S7-Kompakt-SPS

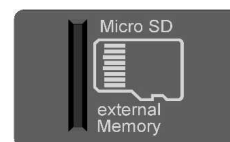
Kommunikationsübersicht

INSEVIS-S7-Kompakt-SPS verfügen über vielseitige Möglichkeiten für die Anbindung von Peripherien und zur Kommunikation mit anderen Geräten. Viele Protokolle sind bereits implementiert, andere können mit den INSEVIS-S7-SFBs und SFCs einfach vom Anwender selbst erstellt werden.



Externe Speicherkarte

Jede Kompakt-SPS besitzt einen Slot für eine optionale Micro-SD-Karte im handelsüblichen FAT32-Format. Es befinden sich keine S7-Programmdaten (im Ladespeicher-Flash = fest) oder Prozessdaten (im batteriegepufferten RAM = flüchtig) auf dieser Karte. Diese Karte ist nur zur Archivierung von Daten aus DBs bzw. zum Restore nötig und kann auch zum S7-Programm- und Firmwareupdate genutzt werden.



Wichtigste Eigenschaften in Stichpunkten

S7-Programmierung

Bestehende Siemens-S7-Programmierungsumgebung weiter nutzen; entweder den SIMATIC®-Manager oder TIA-Portal® in den Programmiersprachen KOP, FUP, AWL, SCL. Auch FB's wie z.B. für Analogverarbeitung weiterverwendbar.

Firmware-Update

Die neue Firmware wird einfach unter das S7-Programm „druntergeschoben“, ohne diese Daten zu beeinflussen. Alles ist in einer Minute erledigt. Natürlich ist ein Firmwareupdate ein Geräteleben lang kostenlos

Datenarchivierung

Prozessdaten auf die Speicherkarte archivieren und nach dem S7-Programmupdate zurücklesen – ganz ohne Programmiergerät – durch INSEVIS- Zusatzbausteine. Damit S7-Kundenprogramme auch noch nach Jahren aktualisieren können.

Gateway-Funktion

Ethernet mit TCP, UDP, RFC1006 oder Modbus TCP auf 1 oder 2 Ports, Profibus-DP V0 Master/Slave oder Profinet IO, CANopen® oder Layer2, freies ASCII auf RS232 und RS485 und Modbus RTU - die INSEVIS-S7-SPS - ein Kommunikationstalent.

S7-CPU in den Kompakt-SPS

CPU-V

Geräte mit **CPU-V** sind ideal geeignet für kleine Anwendungen im LowCost-Bereich von Kompakt-SPSen mit vielfältigen Kommunikationsaufgaben. Profibus steht optional zur Verfügung.



Eigenschaft	Technische Daten
OB, FC, FB, DB Lokaldaten Anzahl Eingänge, Ausgänge Prozessabbild Anzahl Merkerbytes Anzahl Taktmerker Anzahl Zeiten, Zähler Schachtelungstiefe	je 1.024 32kByte (2kByte pro Baustein) je 2.048 Byte (16.384 Bit) adressierbar je 2.048 Byte (128 Byte voreingestellt) 2.048 (Remanenz einstellbar, 0..15 voreingestellt) 8 (1 Merkerbyte) je 256 (jeweils Remanenz einstellbar, 0 voreingestellt) bis zu 16 Codebausteine
Echtzeituhr Betriebsstundenzähler	ja (akkugepufferte Hardware-Uhr) 1 (32Bit, Auflösung 1h)
Programmiersprachen Programmiersystem	STEP 7® - AWL, KOP, FUP, S7-SCL, S7-Graph von Siemens SIMATIC® Manager ab V5.5, TIA-Portal® ab V12 von Siemens
Betriebssystem Referenzbaugruppe	kompatibel zu S7-300® von Siemens CPU 315-2DP/PN (6ES7 315-2EH14-0AB0 ab Firmware V3.1)
Kommunikation	
Serieller CP / onboard (Protokolle)	1x RS 232 (freies ASCII) – Leitung geschirmt 1x RS 485 (freies ASCII, Modbus-RTU)
Ethernet / onboard (Protokolle)	1x ETHERNET: 10/100 MBit mit CP343 Teil-Funktionalität (RFC1006 „S7-Kommunikation“ bzw. „aktives Ethernet“, TCP, UDP, Modbus-TCP)
CAN / onboard (Protokolle)	1x CANopen® kompatibler Master/Slave 10 kBaud ... 1 MBaud CAN-Telegramme (Layer 2),
Schnittstellen / optional (Protokolle)	Profibus DP V0 Master/Slave / 9,6kBaud ... 12 MBaud
Peripherieanbindung	
dezentrale Peripherie	- INSEVIS-Peripherie (mit automatischer Konfiguration) - diverse Fremdperipherie über Modbus RTU/TCP, CAN - alle CANopen® Slaves nach DS401 - alle Profibus DP-V0-Slaves

Speicherausbau	CPU-V
Arbeitsspeicher (davon akkugepuffert)	512kB 256 kByte remanent
Ladespeicher	2MB Flash (nicht flüchtig)
externer Speicher	Micro SD, bis max. 8 GB

Wichtigste Eigenschaften in Stichpunkten

S7-Programmierung

Bestehende Siemens-S7-Programmierungsumgebung weiter nutzen;
entweder den SIMATIC®-Manager oder TIA-Portal® in den
Programmiersprachen KOP, FUP, AWL, SCL. Auch FB's wie z.B. für
Analogverarbeitung weiterverwendbar.

Know-how-Schutz

Sein Know-how effektiv schützen und Neugierige fernhalten.
Mit der kostenlosen ServiceStage unüberwindbare Lese- und
Schreibschutzstufen setzen.
(Siemens-Passwortfunktionen weiterhin aktivierbar.)

Systembootzeit 4 Sekunden

Kein Windows-Betriebssystem heißt, in weniger als 4 Sekunden
booten und vor allem: keine Lizenzen. Aber auch keine Run-Time-
Beschränkungen für Tags. Damit heutige Modelle auch noch in 10
Jahren aktualisiert werden können.

Gateway-Funktion

Ethernet mit TCP, UDP, RFC1006 oder Modbus TCP, Profibus-DP
V0 Master/Slave oder Profinet IO, CANopen® oder Layer2, freies
ASCII auf RS232 und RS485 und Modbus RTU
- die INSEVIS-S7-SPS - ein Kommunikationstalent.

S7-CPU in den Kompakt-SPS

CPU-T

Geräte mit **CPU-T** haben noch mehr Speicher, eine deutlich höhere Abarbeitungsgeschwindigkeit und sind ideal für mittlere bis große Lösungen. Zwei getrennte Ethernet-Ports für verschiedene Netze sind bereits onboard. Profinet steht optional zur Verfügung.



Eigenschaft	Technische Daten
OB, FC, FB, DB Lokaldaten Anzahl Eingänge, Ausgänge Prozessabbild Anzahl Merkerbytes Anzahl Taktmerker Anzahl Zeiten, Zähler Schachtelungstiefe	je 2.048 32kByte (2kByte pro Baustein) je 4.096 Byte (32.768 Bit) adressierbar je 4.096 Byte (128 Byte voreingestellt) 4.096 (Remanenz einstellbar, 0..15 voreingestellt) (wegen TIA-Kompatibilität jedoch nur 2.096 einstellbar,) 8 (1 Merkerbyte) je 512 (jeweils Remanenz einstellbar, 0 voreingestellt) bis zu 16 Codebausteine
Echtzeituhr Betriebsstundenzähler	ja (akkugepufferte Hardware-Uhr) 1 (32Bit, Auflösung 1h)
Programmiersprachen Programmiersystem	STEP 7® - AWL, KOP, FUP, S7-SCL, S7-Graph von Siemens SIMATIC® Manager ab V5.5, TIA-Portal® ab V12 von Siemens
Betriebssystem Referenzbaugruppe	kompatibel zu S7-300® von Siemens CPU 315-2DP/PN (6ES7 315-2EH14-0AB0 ab Firmware V3.1)
Kommunikation	
Serieller CP / onboard (Protokolle)	1x RS 232 (freies ASCII) – Leitung geschirmt 1x RS 485 (freies ASCII, Modbus-RTU)
Ethernet / onboard (Protokolle)	2x ETHERNET (Switch oder getrennte Ports möglich): 10/100 MBit mit CP343 Teil-Funktionalität: (RFC1006 „S7-Kommunikation“ bzw. „aktives Ethernet“, TCP, UDP, Modbus-TCP)
CAN / onboard (Protokolle)	1x CANopen® kompatibler Master/Slave 10 kBaud ... 1 MBaud CAN-Telegramme (Layer 2),
Schnittstellen / optional (Protokolle)	Profinet IO Controller
Peripherieanbindung	
dezentrale Peripherie	- INSEVIS-Peripherie (mit automatischer Konfiguration) - diverse Fremdperipherie über Modbus RTU/TCP, CAN - alle CANopen® Slaves nach DS401 - alle Profinet-IO-Devices

Speicherausbau	CPU-T
Arbeitsspeicher (davon akkugepuffert)	1MB (512 kByte remanent)
Ladespeicher	8MB Flash (nicht flüchtig)
externer Speicher	Micro SD, bis max. 8 GB

Wichtigste Eigenschaften in Stichpunkten

S7-Programmierung
Bestehende Siemens-S7-Programmierungsumgebung weiter nutzen; entweder den SIMATIC®-Manager oder TIA-Portal® in den Programmiersprachen KOP, FUP, AWL, SCL. Auch FB's wie z.B. für Analogverarbeitung weiterverwendbar.
Know-how-Schutz
Sein Know-how effektiv schützen und Neugierige fernhalten. Mit der kostenlosen ServiceStage unüberwindbare Lese- und Schreibschutzstufen setzen. (Siemens-Passwortfunktionen weiterhin aktivierbar.)

2 Ethernet-Ports
Exklusiv bei der CPU-T: Mit den beiden Ethernetschnittstellen als getrennte Ports mit eigenen IP-Adresskreisen die Steuerung als Verbindung zwischen Büronetzwerk und Maschinennetzwerk verwenden.
Gateway-Funktion
Ethernet mit TCP, UDP, RFC1006 oder Modbus TCP, Profibus-DP V0 Master/Slave oder Profinet IO, CANopen® oder Layer2, freies ASCII auf RS232 und RS485 und Modbus RTU - die INSEVIS-S7-SPS - ein Kommunikationstalent.

Abgesetzte Bedieneinheiten für die Kompakt-SPS

Allgemein

INSEVIS-Panel-HMIs sind die idealen abgesetzten Bedieneinheiten für die Kompakt-SPSen. Pro Kompakt-SPS können bis zu 5 Panel-HMI angeschlossen werden.

Durch den Einsatz von TFT-Farb-Displays wird eine hervorragende Brillanz in Bezug auf Farbechtheit, Helligkeit und Blickwinkel gewährt. Die Hinterleuchtung mit energiesparenden LEDs garantiert nicht nur typische Halbwertszeiten von 50.000h, sondern lässt den Betriebstemperaturbereich auf -20°C bis +60°C (ohne Betauung) wachsen. Dabei sind sämtliche Panel-SPSen vertikal und horizontal verwendbar. Der resistive Touch wird durch eine laminierte IP65-dichte Frontfolie zuverlässig geschützt.

Es stehen elektropolierte Edelstahlfronten für den Lebensmittelbereich auf Anfrage zur Verfügung. Kundenspezifische Logos können front- und rückseitig leicht angebracht werden. Es sind auch Betriebssysteme mit Kundenlogo verfügbar. Damit können INSEVIS-Produkte ganz ohne Initialkosten harmonisch in die eigene Produktfamilie integriert werden.

Mit CPU-V und CPU-P

Die S7-Panel-HMIs mit den CPU-V und -P verfügen über folgende Displaydiagonalen:

- 3,5" mit QVGA- Auflösung (320x240Pixel)
→ im 4:3-Format für den Fronttafeleinbau
→ in der Höhe der Normeinbaueinheit von 96mm
- 5,7" mit QVGA- Auflösung (320x240Pixel) und
→ im 4:3-Format für den Fronttafeleinbau
- 7" mit WVGA- Auflösung (800x480Pixel)
→ im 16:9-Format für den Fronttafeleinbau
- 10,2" mit WVGA- Auflösung (800x480Pixel)
→ im 16:9-Format für den Fronttafeleinbau
(abgekündigt)



Mit CPU-T

Die S7-Panel-HMIs mit den CPU-T verfügen über folgende Displaydiagonalen:

- 4,3" mit Auflösung 480x272Pixel
→ im 16:9-Format für den Fronttafeleinbau
→ in der Höhe der Normeinbaueinheit von 96mm
- 7" mit WVGA- Auflösung (800x480Pixel)
→ im 16:9-Format für den Fronttafeleinbau und
→ einbaufähig zu 7" der Generation I
- 10,1" mit Auflösung 1024x600 Pixel
→ im 16:9-Format für den Fronttafeleinbau und
→ einbaufähig zu 10,1" der Generation I
- 15,6" mit Auflösung 1366x768 Pixel
→ im 16:9-Format für den Fronttafeleinbau
(dieses Produkt nur 0...50°C)



Peripherie für die Kompakt-SPS

Kommunikation SPS zu Peripherie

Während die Onboard-Peripherie über den integrierten Rückwandbus direkt auf die INSEVIS-CPU's zugreift, kommunizieren die Kopfstationen der dezentralen Peripherie über ein CANopen®-kompatibles Protokoll mit den SPSen.

Bei den INSEVIS-S7-CPU's wurde diese Ansteuerung kinderleicht gemacht; einfach die Knotennummer eintragen und schon ist die dezentrale Peripherie wie Onboard-Peripherie zu verwenden. Jedes Fehladressieren wird zudem von der kostenlosen Konfigurationssoftware automatisch angezeigt. Und: es ist definitiv kein CAN- Wissen nötig.

Onboard-Peripherie

- digitale E/As (24/0,5A / 230V/5A)
- analoge E/As (U,I,R / 12-16 Bit)
- schnelle Zähler A,B,Z (5/24V/RS422, 125kHz)



1...127x Dezentrale Peripherie DP3xxC

- digitale E/As (24/0,5A / 230V/5A)
- analoge E/As (U,I,R / 12-16 Bit)
- schnelle Zähler A,B,Z (5/24V/RS422, 125kHz)
- alle CANopen-fähigen Fremdgeräte (z.B. E/As, FUs, Antriebe; etc.)

Peripheriebaugruppen, allgemein

Dezentrale Kopfstationen DP303C/DP307C/DP311C mit 3 / 7 / 11 Slots für folgende Peripheriemodule

- DI16 / (16 Digitaleingänge 24V)
- DIO16 / (16 Digitalein- oder ausgänge 24V / 0,5A)
- DO4-R / (4 Relaisausgänge 230V / 3A)
- MIO84 / (8 Digital- und 4 Analogein-/ ausgänge)
- AI404 / (4 Analogein- und 4 Analogausgänge)
- AI802 / (8 Analogein- und 2 Analogausgänge)
- AI8 / (8 Analogeingänge)
- RTD802 / (8 Analogein- und 2 Analogausgänge)
- DIO8-Z / (2 Geberkanäle A,B,Z + 2 Digitalein- oder Ausg.)

Für alle Baugruppen stehen 20polige, farbig pinmarkierte Steckverbinder mit Lösehebeln oder Schraubflansch zur Verfügung (Ausnahme PM-DO4-R).

Für alle Baugruppen geltende Daten:

Eigenschaft	Technische Daten
Betriebstemperaturbereich	-20°C ... +60°C (ohne Betauung)
Lagertemperaturbereich	-30°C ... +80°C
Abmessungen B x H x T	20 x 108 x 70 mm
Gewicht	ca. 150 g
Leitungslänge	
- ungeschirmt (max.)	30 m
- geschirmt (max.)	100 m
Anschluss technik	Steckverbinder mit Zugfederkontakt für Querschnitte max. 1,5mm² - mit Lösehebel oder Schraubflansch

Wichtigste Eigenschaften in Stichpunkten

Hohe Packungsdichte

Durch nur < 20mm Breite pro Modul viele E/As auf geringer Fläche verarbeiten. Zum Kabelkanal geneigte Steckerebene. Pinmarkierte Käfigzugfeder-Steckerbinder erlauben die Verwendung vorab gefertigter Kabelsätze.

Einfache Kopfstationen

Dezentrale Kopfstationen lediglich über eine Knotennummer an zwei Drehschaltern einstellen. Dann die dezentralen Peripheriemodule ansprechen wie die Onboardmodule. Vorbildlich einfach - wie das gesamte INSEVIS-Handling.

Hohe Auflösungen

Mindestens 12Bit haben alle INSEVIS- Analogmodule. Das PM-AI404 kann mit ein wenig mehr Integrationszeit bis zu 16Bit auflösen. Natürlich ohne Aufpreis - So fair wie die gesamte INSEVIS- Philosophie

Intelligente Konfiguration

Per Software für jedes einzelne Bit konfigurieren, ob ein Ein- oder Ausgang verarbeitet werden soll. Möglich durch rücklesbare Ausgänge. Mit dem PM-DIO16 immer genügend Reserven haben - für alle Fälle

Software für die Kompakt-SPS

Die INSEVIS Software Tools sind kostenlos, laufen auf Windows PCs und können wahlweise in deutsch oder englisch ausgeführt werden.

Konfiguration



Mit dem Konfigurationsstool „ConfigStage“ werden die Zusatzfunktionen der INSEVIS-CPU's parametrieren und in die SPS geladen. Die onboard- bzw. dezentrale INSEVIS-Peripherie wird per Drag'n Drop auf die Steckplätze gezogen, parametrieren und Adressbereiche werden vergeben. Alternativ zur Siemens-Programmiersoftware kann man auch hier die S7-Steuerungsparameter der S7-CPU (Anlaufverhalten, Passwortschutz, Zykluszeitüberwachung, Remanenz, Uhrzeit- und Weckalarme) einstellen. Diverse S7-Bausteine für Motion-Control-Funktionen intelligenter CAN-Antriebe und Beispielkonfigurationen für Fremdperipherien stehen dokumentiert auf unserer Webseite zum freien Download zur Verfügung.

Mit der „ConfigStage“-Software können folgende Schnittstellen konfiguriert werden:

- RS232 mit freiem ASCII,
- RS485 mit freiem ASCII und ModbusRTU,
- Ethernet-Verbindung (S7-Verbindung, TCP, UDP, Modbus-TCP, INSEVIS-Panel-HMI),
- CAN (CANopen® über voreingestellte Parametersätze oder Import von EDS-Dateien),
- Ein Webserver ist bei allen SPSen mit CPU-T aktivierbar.

Remotezugang



Mit der Software „RemoteStage“ kann aus der bereits vorhandenen Visualisierung für die INSEVIS-Panel-SPSen eine Remote-Visualisierung auf dem Arbeitsplatzrechner erzeugt werden. Das Programm steht als direkt ausführbare Datei zur Verfügung (auch zum Erstellen von Batch Dateien nutzbar). Die Software kommuniziert via S7 Kommunikation (Put/Get) mit den S7 CPU's. Von dort werden Prozessdaten eingelesen, die Remote visualisiert werden können.

In einer zweiten Instanz können Archivdaten (Alarme, Ereignisse, Trends, DBs) sowie Rezepturdatensätze von der Micro-SD-Karte des Remotegerätes per Ethernet eingelesen, angezeigt und in csv-Dateien abgespeichert werden. Dieser Vorgang lässt sich auch in automatische Abläufe einbinden. Dieses Programms kann mehrfach nebeneinander ausgeführt werden, so dass mehrere Remotegeräte gleichzeitig visualisiert werden können (Leitwartenfunktion).

Servicetool



Die „ServiceStage“ ist in einer Minute installiert, intuitiv bedienbar und stellt die für Service und Wartungsarbeiten benötigten Funktionen zur Verfügung. Das spart im Vergleich zum Einsatz umfangreicher Programmierertools Zeit und reduziert die Komplexität.

- Auslesen gerätespezifische Daten und Aktualisieren der Firmware (CPU-T),
- Änderung des Betriebszustandes RUN ↔ STOP,
- Setzen und Synchronisieren von Datum / Uhrzeit,
- Speicherdiagnose und -komprimierung,
- Auslesen und Speichern des CPU-Diagnosepuffers,
- Download des S7-Programms, der Visualisierungs- und Konfigurationsdaten,
- Setzen der Know-how-Schutzstufen,
- Erstellen von OnlineBackups.

Zubehör für die Kompakt-SPS

Zubehör S7-Kompakt-SPS

Für den Anschluss von INSEVIS-Geräten stehen pinmarkierte Stecker mit seitlichen Schraubflanschen zur Verfügung. Das erlaubt eine eindeutige Zuordnung der Pins zu den Signalen und erleichtert die Verdrahtung. Die Kontaktierung erfolgt mit wartungsfreien Zugfeder-Kontakten für max. 1,5mm² Querschnitte ohne Aderendhülsen. (Die Stecker mit Lösehebeln der Reihe E-CON XX sind abgekündigt und wurden durch die Stecker mit Schraubflanschen der Reihe E-CONS XX ersetzt.)

Jeder Lieferung liegen Befestigungsmaterial und eine Erdungskralle bei. Bei Bestellungen mit Onboard-Peripheriemodulen werden diese werksseitig vormontiert mit zugehöriger Rückfolie und Kennzeichnungsstreifen.

Abbildung Zubehör	Zubehör
Stecker  E-CONS16 (pinmarkierte Stecker für max. 1,5mm² Anschluss)	Für CPU V: Steckverbinder 2x8polig, Schraubflansch Ab 2023 sind die Markierungen seitlich am Stecker aufgebracht. Profibus-Adapter für 12MBAud-Netze
 E-CONS10 (pinmarkierte Stecker für max. 1,5mm² Anschluss)	Für CPU T: Steckverbinder 2x5polig, Schraubflansch Ab 2023 sind die Markierungen seitlich am Stecker aufgebracht.
Externer Speicher  Hinweis: Das S7-Programm läuft auch ohne diese Speicherkarten, die nur für Archivierungen und Rezepturen nötig sind.	Micro SD-Karte 1GB (externer Speicher) Micro SD-Karte 2GB (externer Speicher) Micro SD-Karte 4GB (externer Speicher) Micro SD-Karte 8GB (externer Speicher)
Kundenspezifisches Labeln Hinweis: 1) Verpackungseinheit von 100 Stück 2) Einmalkosten, keine Stückkosten	Einschubstreifen H mit Kundenlogo rückseitig Einschubstreifen V mit Kundenlogo rückseitig
Software Hinweis: Software im Internet downloadbar	ConfigStage RemoteStage ServiceStage
Ersatzteile Hinweis: Eine Erdungskralle ist in jedem Lieferumfang enthalten. 4) Verpackungseinheit von 10 Stück	Zusätzlicher Satz mit 10 ErdungskralLEN

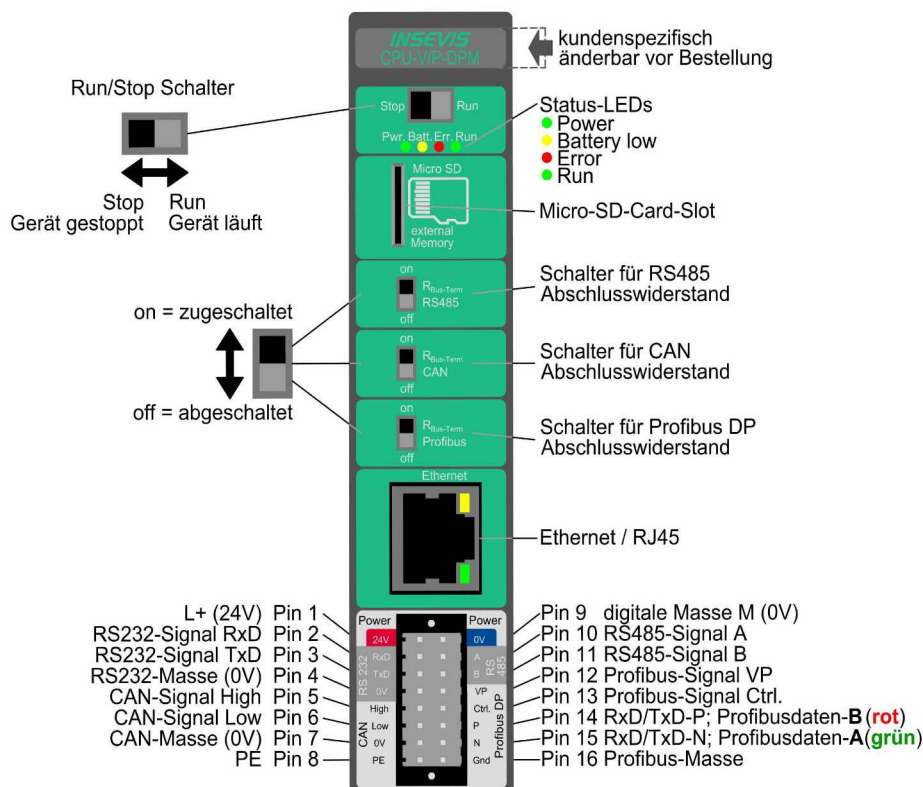
Anbauteile und Sonderkonstruktionen auf Anfrage.

Beschaltung der CPU-V

Beschaltung der CPU-V in Geräten mit und ohne Peripherieslots

Beschaltung bei
bei Gerät ohne
Peripherieslots:
CC300V

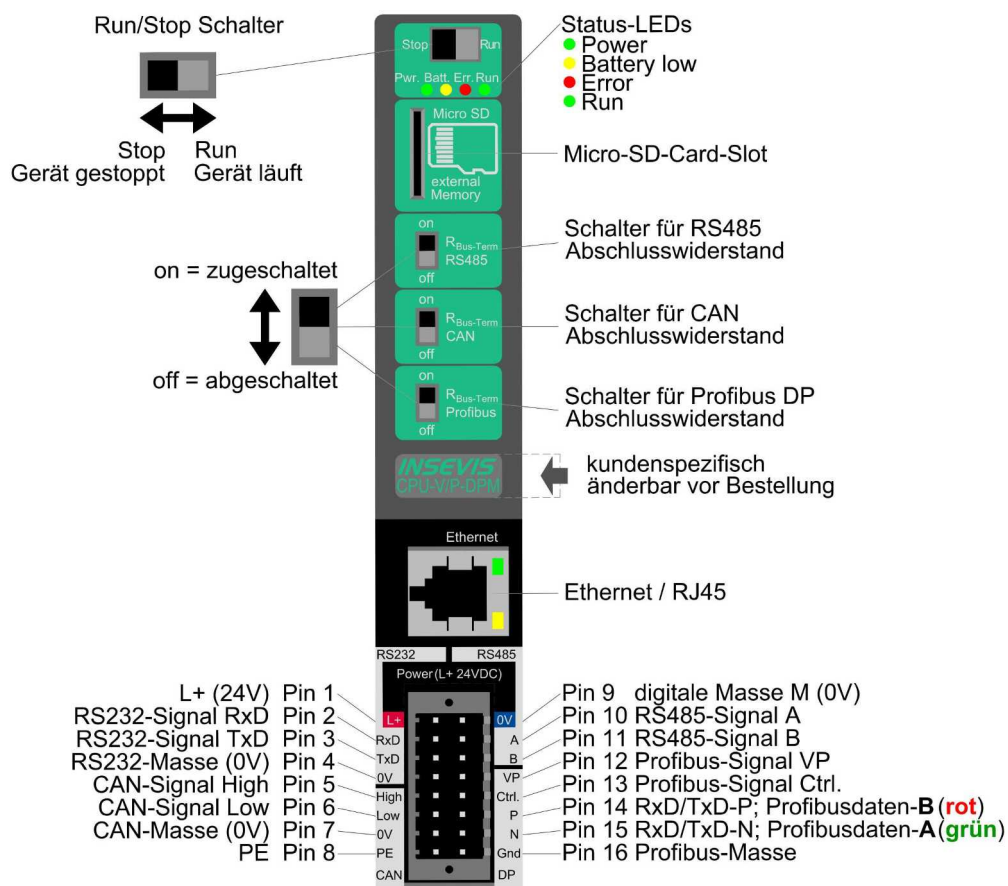
Hinweis:
Bei leerem Akku leuchtet die
Battery LED gelb. Wird der
Akku aufgeladen erlischt die
gelbe LED erst nach Power
OFF/ON.



RS232-Leitung lt. Spec
geschirmt ausführen!

Beschaltung bei
den Geräten mit
Peripherieslots:
CC303V
CC307V
CC311V

Hinweis:
Bei leerem Akku leuchtet die
Battery LED gelb. Wird der
Akku aufgeladen erlischt die
gelbe LED erst nach Power
OFF/ON.



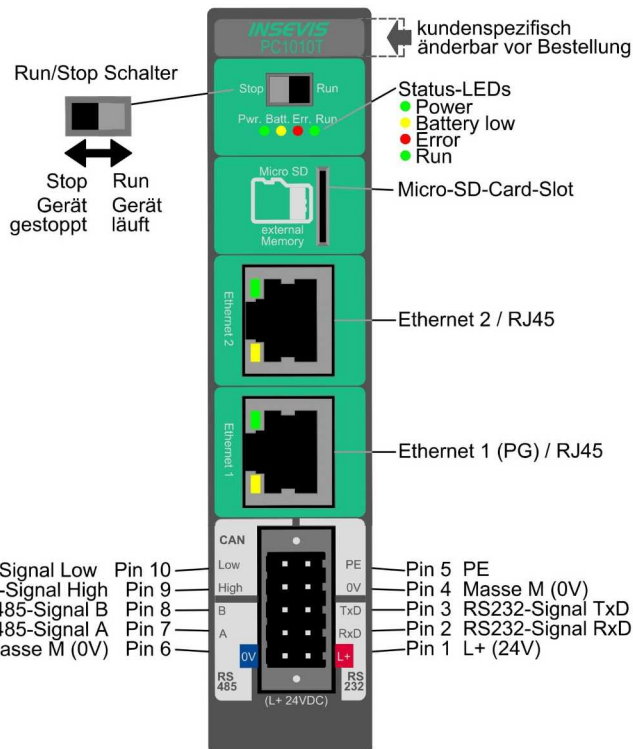
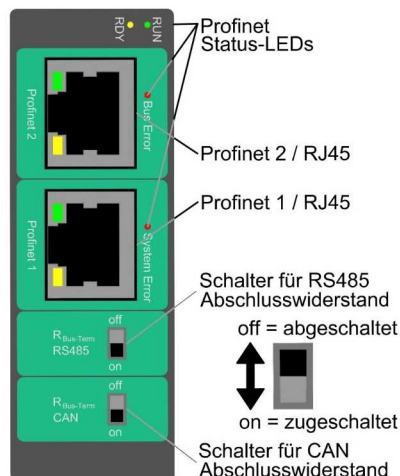
RS232-Leitung lt. Spec
geschirmt ausführen!

Beschaltung der CPU-T

Beschaltung der CPU-T in Geräten mit und ohne Peripherieslots

Beschaltung bei
Gerät ohne
Peripherieslots
CC300T

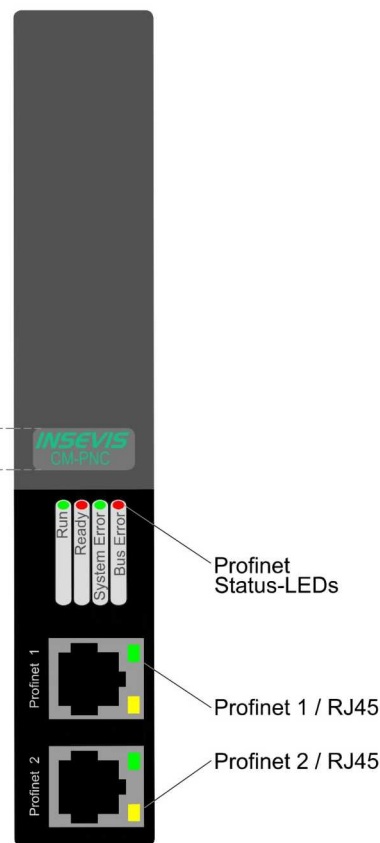
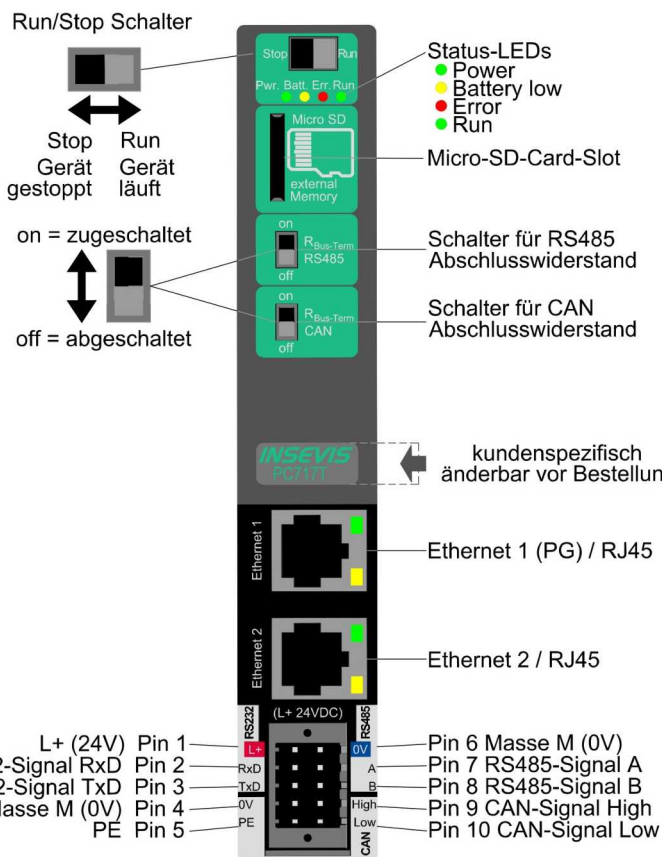
Hinweis:
Bei leerem Akku
leuchtet die Battery LED
gelb. Wird der Akku
aufgeladen erlischt die
gelbe LED erst nach
Power OFF/ON.



RS232-Leitung lt. Spec
geschirmt ausführen!

Beschaltung
bei den Geräten
mit Periphe-
rieslots
CC303T
CC307T
CC311T

Hinweis:
Bei leerem Akku
leuchtet die Battery LED
gelb. Wird der
Akku aufgeladen
erlischt die gelbe
LED erst nach
Power OFF/ON.



RS232-Leitung lt. Spec geschirmt ausführen!

Allgemeines

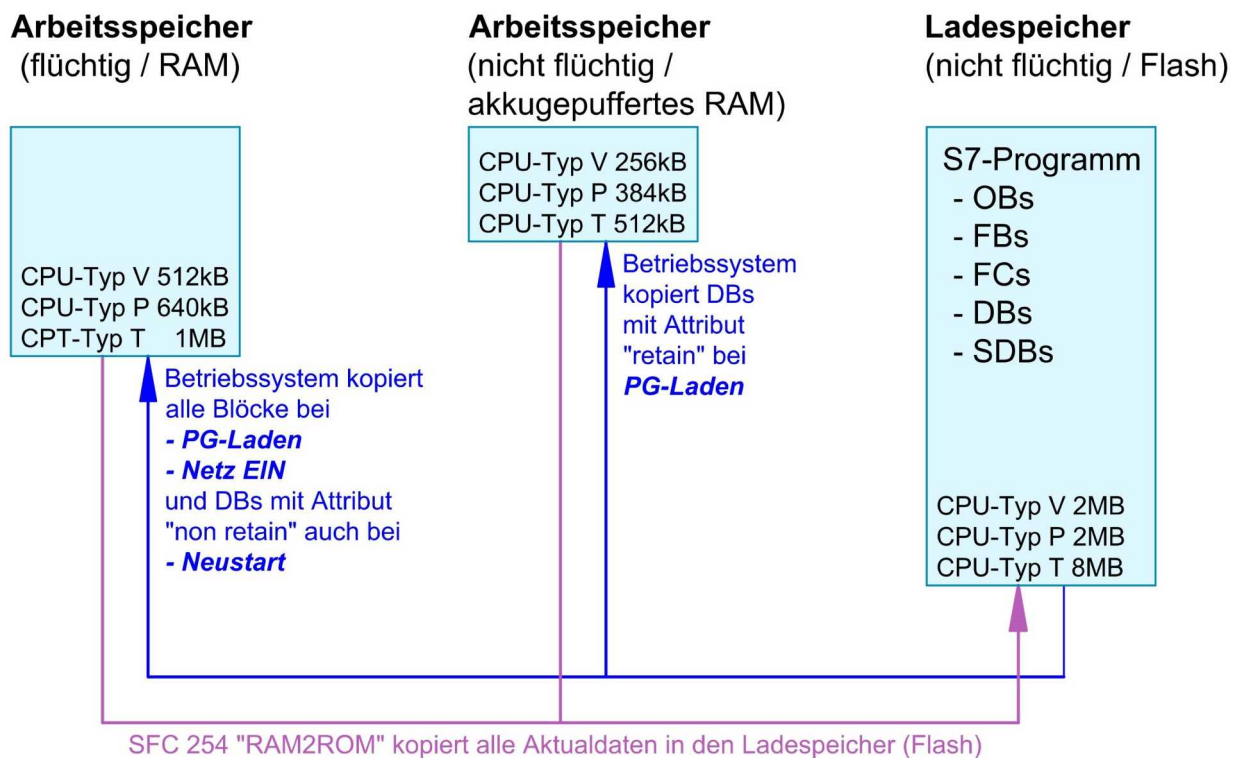
Speicherorganisation

Die Speicheraufteilung entspricht im Wesentlichen denen der Siemens S7-CPU's. Das S7-Programm bleibt im Flash und wird vom INSEVIS- Betriebssystem in das RAM kopiert.

Ein Akku hält die Daten im akkugepufferten RAM ca. 3 Monate. Daher wird vor längerem Ausschalten empfohlen, die Daten vom Arbeitsspeicher (RAM) mit dem SFC 254 „RAM2ROM“ in den Ladespeicher (Flash=ROM) zu kopieren.

Bei leerem Akku wird nach dem Einschalten „Batteriefehler“ in den Diagnosepuffer eingetragen und die gelbe Battery-LED auf der Rückseite der SPS leuchtet.

Das Betriebssystem kopiert alle Blöcke aus dem Lade- in den Arbeitsspeicher und als remanent konfigurierte Merker, Timer und Zähler werden zurückgesetzt. Dann wechselt die CPU in den Run-Modus.



Ethernet

Die Verbindung zum Simatic®-Manager / TIA®-Portal erfolgt über Ethernet TCP/IP. Dabei wird die Panel-SPS über eine eigene IP-Adresse identifiziert.



ACHTUNG:

Die INSEVIS- (Panel- oder Kompakt-) SPSen werden mit den IP-Adressen **192.168.80.50 (CPU-V-P)** oder **192.168.80.55 (CPU-T)** ausgeliefert.

Ändern Sie die IP-Adresse der SPS, bevor Sie die Kommunikation einrichten.

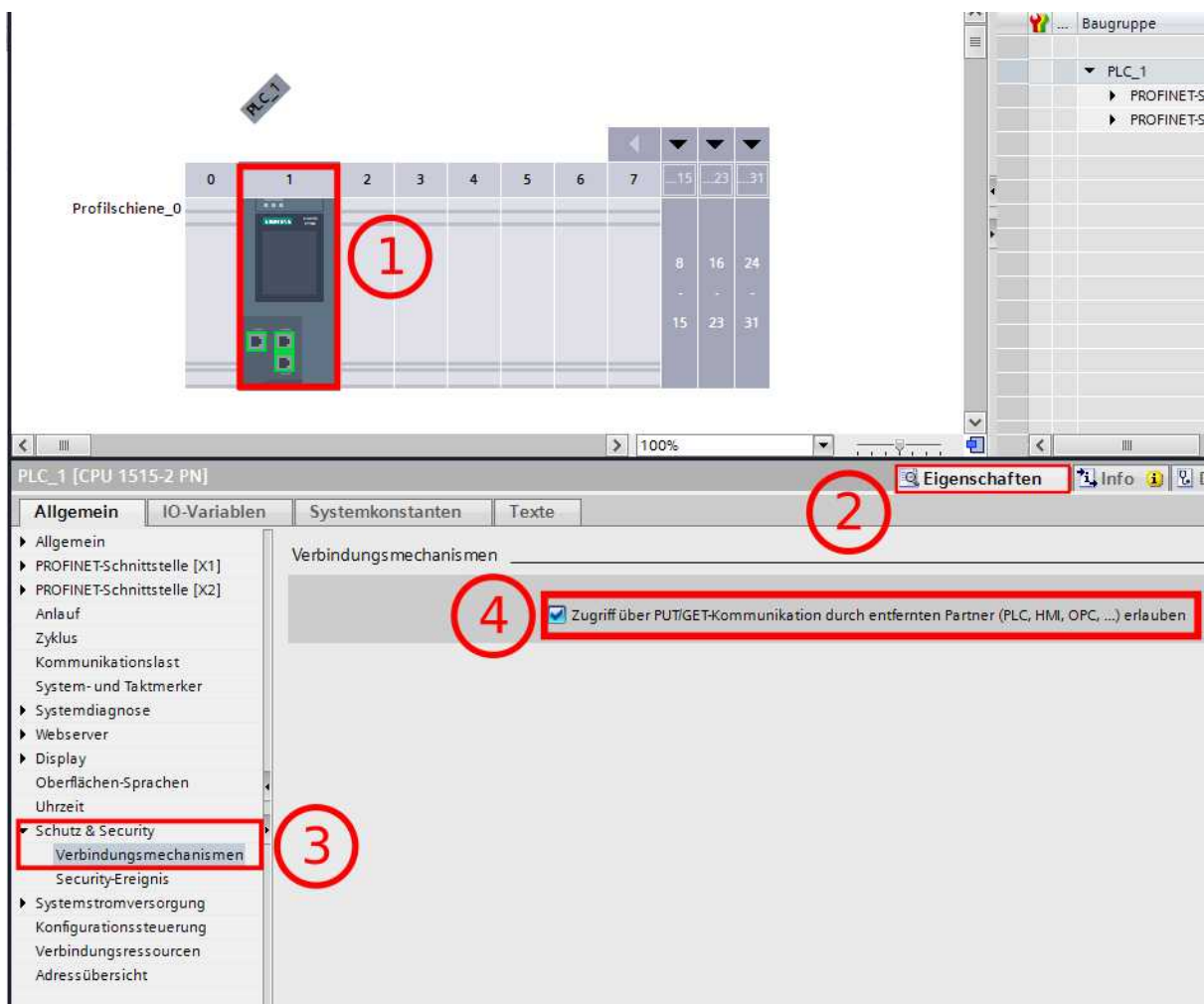
Bei Panel-SPSen ist im Lieferzustand eine Visualisierung mit einem S7-Programm aufgespielt, mit der Sie am Panel die IP-Adresse zur Laufzeit einstellen können.



ACHTUNG bei TIA®-Portal:

Um die Kommunikation mit CPUs der Serien 1200 und 1500 zu gewährleisten muss für diese die PUT/GET Kommunikation aktiviert werden. Führen Sie dazu die folgenden Schritte in Ihrem TIA®-Projekt aus:

1. Markieren Sie Ihre CPU.
2. Wählen Sie "Eigenschaften".
3. Selektieren Sie "Schutz & Security" -> Verbindungsmechanismen
4. Aktivieren Sie die Option:
"Zugriff über PUT/GET-Kommunikation durch entfernte Partner (PLC, HMI, OPC, ...) erlauben".



Kommunikation - Ethernet

IP-Adresse im Simatic®-Manager einstellen

Benötigte Ausrüstung

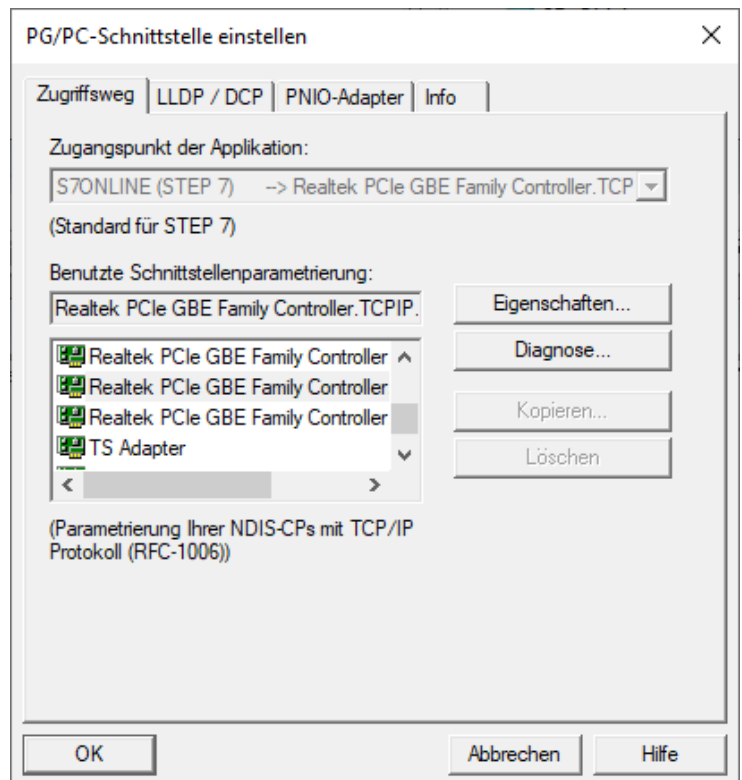
PC mit Simatic®-Manager ab
Version 5.4 SP5 (PG)
INSEVIS-SPS (AG)
Netzteil 24V
Patchkabel oder Crossover-patchkabel

Vorgehensweise

PC mit AG verbinden
AG Power on
Starten Simatic® Manager

PG/PC-Schnittstelle einstellen:

- Extras
- PG/PC-Schnittstelle einstellen
- Reiter „Zugriffsweg“ wählen
- TCP/IP auswählen
- mit „OK“ bestätigen



Steuerung anwählen

- Zielsystem
- Erreichbare Teilnehmer anzeigen
(Meldung als INSEVIS Geräte~~name~~) z.B. **INSEVIS PC430T** auswählen / markieren
- Zielsystem
- Ethernet Teilnehmer bearbeiten



Kommunikation - Ethernet

IP-Adresse ändern

- weiter in Box
- „IP-Konfiguration einstellen“
- „IP-Parameter anwählen“
- IP-Adresse eintragen
- Subnetmaske eintragen
- Button drücken:
- „IP-Konfiguration zuweisen“
- mit „OK“ bestätigen

IP-Adresse aktualisieren

Bevor die IP-Adresse angesprochen werden kann, muss diese neu gesucht werden:

- Zielsystem
- Erreichbare Teilnehmer anzeigen

ODER:

- Mit „F5“ aktualisieren

Kontrolle IP-Adresse

- Zielsystem
- Erreichbare Teilnehmer anzeigen
- rechte Maustaste
- Objekteigenschaften



Hinweis:

Weitere Einstellungen, wie Netzmaske und Router-Adresse werden ausschließlich in der INSEVIS-Konfigurationssoftware ConfigStage bei den Ethernet-Einstellungen der CPU vorgenommen. Aus der Siemens-HW-Config werden keine anderen Daten außer die IP-Adresse der SPS übernommen!



VIDEO-Tutorial verfügbar

Für dieses Beispiel ist auf den INSEVIS-Download-Seiten ein Link auf ein entsprechendes Video bei YouTube® verfügbar für **Simatic®-Manager UND TIA®-Portal**.

Kommunikation - Ethernet

Anlegen der CPU und der Ethernet-Verbindung im Simatic®-Manager

Anlegen der INSEVIS-CPU im Simatic®-Manager:

Baugruppe:
CPU315-2PN/DP

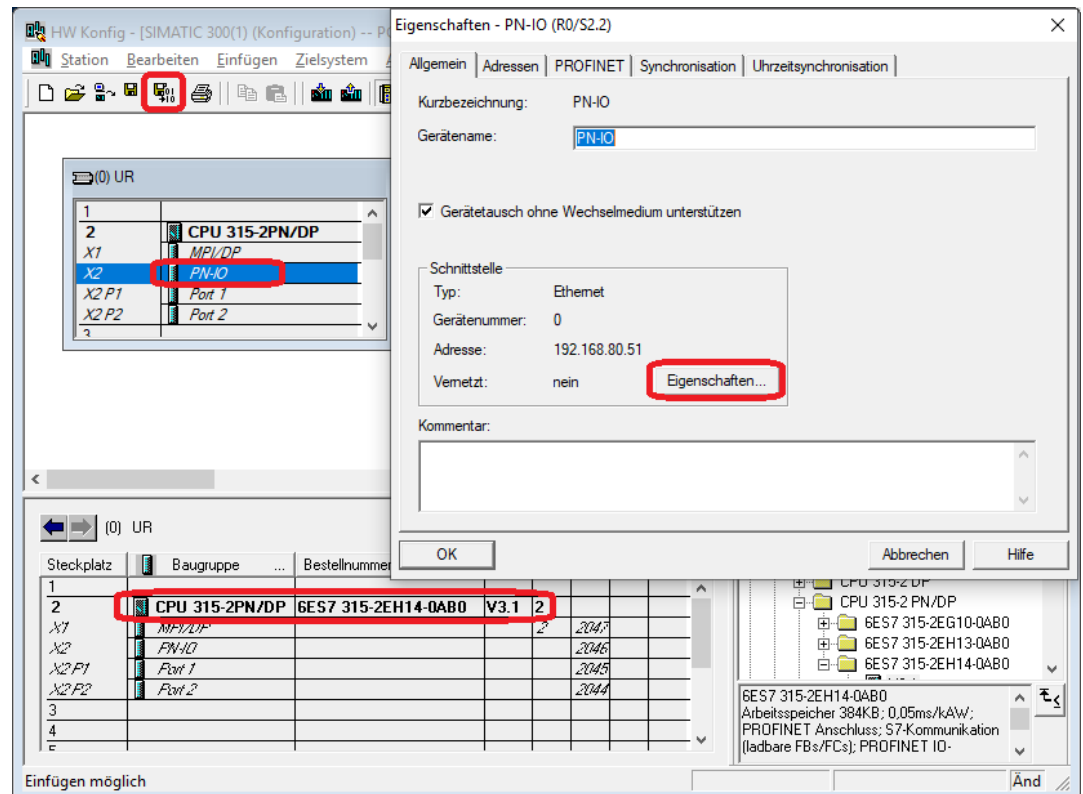
Bestellnummer:
6ES7315-2EH14-0AB0

Firmware:
V 3.1

(diese CPU ist erst ab V5.4 SP5 des Simatic®-Managers verfügbar, ggf. ist ein Update auf V5.5 bzw. V5.6 nötig.

Auswahl der Ethernet-Kommunikation im Simatic®-Manager:

- Doppelklick auf PN-IO
- Auf „Eigenschaften“ gehen
- IP-Adresse einstellen und mit „OK“ bestätigen
- Kompilieren und speichern



ACHTUNG bei Verbindungsproblemen:

Ihre Firewall verhindert die Verbindungsaufnahme → Bitte Firewall(s) rekonfigurieren
IP-Adresse des PC und der SPS müssen in einem gemeinsamen Netz sein (im Beispiel 192.168.80.xxx) → Mit dem Kommandozeilenbefehl "ipconfig" können Sie die aktuelle IP-Konfiguration Ihres PC's abfragen.



VIDEO-Tutorial verfügbar

Für dieses Beispiel ist auf den INSEVIS-Download-Seiten ein Link auf ein entsprechendes Video bei YouTube® verfügbar für **Simatic®-Manager UND TIA®-Portal**.

IP Konfiguration der SPS mit dem SFB 129 "IP_CFG"

INSEVIS-Steuerungen bieten den SFB 129 "IP_CFG", um die IP-Konfigurationsparameter der SPS abzufragen oder zu überschreiben. Damit wird die bisher gültige Konfiguration überschrieben.

Parameter	Bezeichnung	Datentyp	Beschreibung
SET	INPUT	BOOL	TRUE = Überschreiben der IP-Konfiguration, FALSE = Abfragen der IP-Konfiguration
DONE	OUTPUT	BOOL	TRUE = Erfolgreich konfiguriert FALSE = Konfiguration fehlgeschlagen
ERROR	OUTPUT	BOOL	TRUE = Konfiguration fehlgeschlagen
STATUS	OUTPUT	WORD	Statuscode
IP_ADDR	IN_OUT	DWORD	IP-Adresse z.B. 192.168.80.50 => DW#16#C0A85032
NETMASK	IN_OUT	DWORD	Netmask. z.B. 255.255.255.0 => DW#16#FFFFFF00
ROUTER	IN_OUT	DWORD	Router IP-Adresse. DW#16#0 - Kein Router

DONE, ERROR und STATUS Parameterbedeutung

DONE	ERROR	STATUS W#16#..	Beschreibung
1	0	0	Erfolgreich konfiguriert
0	1	8000	Parameter IP_ADDR nicht korrekt



Hinweis: Dieser Funktionsblock ist für SPSen mit den CPUs-V/P und für SPSen mit der CPU-T und als Switch konfigurierten Ethernet-Pots. Für CPU-T-Geräte mit getrennt konfigurierten Ethernet Ports stehen die SFB 131 "IP_GET" und SFB 132 "IP_SET" zur Verfügung.

Kommunikation - Ethernet

S7 Aktive Verbindungen zu Partner-SPSen (RFC1006)

Alle Verbindungen müssen in der ConfigStage konfiguriert und in die lokale SPS geladen werden.

Maximal 16 Verbindungen können konfiguriert werden. Während der Konfiguration werden die Verbindungsparameter wie Verbindungstyp, Partneradresse, Port-Nummer, etc. zugewiesen.

Die konfigurierte Verbindungs-ID-Nummer muss an die ID-Parameter der SFB's 120/121 übergeben werden.

Die INSEVIS S7-SPSen bieten folgende SFBs zum Datenaustausch mit anderen Ethernet-fähigen Kommunikationspartnern durch das Anwenderprogramm an.

- SFB 120 "GET" Daten Lesen von einer Partner-SPS
- SFB 121 "PUT" Daten Schreiben an eine Partner-SPS

Daten lesen von einer Partner-SPS mit SFB 120 "GET"

Durch die Benutzung des SFB 120 "GET" können Sie Daten von einer Partner-SPS (z.B. INSEVIS, SIEMENS, VIPA oder anderer RFC1006-fähiger SPSen) lesen. Dieses Programm steuert das Lesen der Variablen und benötigt keine zusätzlichen Kommunikationsfunktionen im Anwenderprogramm des Kommunikationspartners.

Die Parameter ID, ADDR, RD werden übertragen und der Datenaustausch startet bei der positiven Flanke des Parameters REQ.

Der neue Auftrag kann nur übertragen werden, wenn der alte Auftrag abgearbeitet wurde und nur an einer steigenden Flanke bei REQ.

Die Beendigung des Auftrages wird angezeigt durch den NDR- oder ERROR-Parameter.

Versichern Sie sich, dass die mit den ADDR- und RD-Parametern definierten Bereiche in der Festlegung der Nummern, Länge und Datentyp zusammenpassen.



Hinweis zu Get und Put in einer Verbindung

Jede Verbindung hat einen Auftragspuffer. Wenn ein Auftrag (SFB 120 „Get“ oder SFB 121 „Put“) in Bearbeitung ist, wird kein weiterer Auftrag akzeptiert.

Wenn man z.B. hintereinander bei einer Verbindung den SFB 120 „Get“ und SFB 121 „Put“ aufruft, wird SFB 121 „Put“ nicht akzeptiert, mit Fehlermeldung W#16#8003 zurückgegeben und beendet.

Man muss hier die Beendigung des ersten Auftrages abwarten, bevor man den zweiten Auftrag startet oder mit 2 parallelen Verbindungen arbeiten, wenn der Partner das ermöglicht.

Parameter	Bezeichnung	Datentyp	Beschreibung
REQ	INPUT	BOOL	Anforderung des Datenaustausches bei steigender Flanke
ID	INPUT	INT	Verbindungs-ID-Nummer, welche in in der ConfigStage konfiguriert wurde
NDR	OUTPUT	BOOL	TRUE = Auftrag erfolgreich abgeschlossen, FALSE = Auftrag ist in Abarbeitung oder fehlgeschlagen
ERROR	OUTPUT	BOOL	TRUE = Auftrag fehlgeschlagen
STATUS	OUTPUT	WORD	Code des Auftragsstatus
ADDR	IN_OUT	ANY	ANY-Pointer zum Bereich, wo die Partner-SPS lesbar ist.
RD	IN_OUT	ANY	ANY-Pointer zum Bereich der lokalen SPS, wo die gelesenen Daten eingetragen werden.

Kommunikation - Ethernet

NDR, ERROR und STATUS Parameterbedeutung

NDR	ERROR	STATUS W#16#..	Beschreibung
1	0	0	Auftrag erfolgreich abgeschlossen.
0	0	7000	Auftrag wurde nicht aktiviert, weil: • es der erste Aufruf mit REQ=0 ist oder weil • ein anderer Auftrag (Put) noch aktiv ist
0	0	7001	Erster Aufruf mit REQ=1. Auftrag wurde aktiviert und ist in Bearbeitung.
0	0	7002	Zwischenzeitlicher Aufruf (REQ irrelevant). Auftrag in Bearbeitung.
0	0	7003	Auftragsaktivierung wird ausgeführt in einer Klasse mit niedriger Priorität (Verschachtelungsaufwurf).
0	1	8001	Parameter-ID ist nicht korrekt.
0	1	8002	Verbindung mit ID ist nicht konfiguriert oder inkorrekt Verbindungstyp.
0	1	8003	Auftrag kann nicht aktiviert werden, solange vorhergehender Auftrag nicht abgearbeitet wurde.
0	1	8004	Parameter RD ist nicht korrekt.
0	1	8005	Partner-SPS konnte nicht verbunden werden.
0	1	8006	Auftrag kann nicht aktiviert wegen des Verbindungsstatus'.
0	1	8007	Auftrag fehlgeschlagen wegen Verbindungsproblem (Kabel abgezogen, Kommunikation durch Partner zurückgewiesen).
0	1	8008	Auftrag fehlgeschlagen wegen einer negativen Antwort vom Partner.
0	1	8009	Auftrag fehlgeschlagen wegen wegen eines Zugriffsfehlers zum Partnerbereich.
0	1	800A	Auftrag fehlgeschlagen wegen eines Zugriffsfehlers zum lokalen Bereich.
0	1	800B	Parameter ADDR ist nicht korrekt.

S7-Programmbeispiel für die Verwendung des SFB120

Ziel: Lesen von 20Byte Daten, beginnend mit MB500, von einer Partner-CPU und Schreiben in den MB200 in lokaler CPU

```

CALL SFB120, DB120
REQ    :=M600.0           // TRUE = Lese Daten von Partner S7-CPU
ID     :=1                // In ConfigStage konfigurierte Verbindungs-ID-Nummer
NDR    :=M600.1           // TRUE = Neue Daten verfügbar
ERROR  :=M600.2           // TRUE = Fehler aufgetreten
STATUS:=MW602             // Ergebnis / Rückgabewert
ADDR   :=P#M 500.0 BYTE 20 // 20 Byte lesen von Partner S7-CPU
RD     :=P#M 200.0 BYTE 20 // Schreiben in Merkerbereich,
                          // beginnend bei MB 200 in lokaler CPU

O      M      600.1        // Wenn erfolgreich gelesen wurde
O      M      600.2        // oder Fehler aufgetreten sind, dann
R      M      600.0        // setze Anforderungssignal zurück.
```

Kommunikation - Ethernet

Daten schreiben an eine Partner-SPS mit SFB 121 "PUT"

Mit dem SFB 121 "PUT" können Sie Daten an eine Partner-SPS (z.B. INSEVIS, SIEMENS, VIPA oder andere RFC1006-fähige SPSen) senden. Dieses Programm steuert das Schreiben der Variablen und benötigt keine zusätzlichen Kommunikationsfunktionen im Anwenderprogramm des Kommunikationspartners.

Die Parameter ID, ADDR, SD werden übertragen und der Datenaustausch startet bei der positiven Flanke des Parameters REQ.

Der neue Auftrag kann nur übertragen werden, wenn der alte Auftrag abgearbeitet wurde und nur an einer steigenden Flanke bei REQ.

Die Beendigung des Auftrages wird angezeigt durch den DONE- oder ERROR-Parameter.

Versichern Sie sich, dass die mit den ADDR- und SD-Parametern definierten Bereiche in der Festlegung der Nummern, Länge und Datentyp zusammenpassen.



Hinweis zu Get und Put in einer Verbindung

Jede Verbindung hat einen Auftragspuffer. Wenn ein Auftrag (SFB 120 „Get“ oder SFB 121 „Put“) in Bearbeitung ist, wird kein weiterer Auftrag akzeptiert.

Wenn man z.B. hintereinander bei einer Verbindung den SFB 120 „Get“ und SFB 121 „Put“ aufruft, wird SFB 121 „Put“ nicht akzeptiert, mit Fehlermeldung W#16#8003 zurückgegeben und beendet.

Man muss hier die Beendigung des ersten Auftrages abwarten, bevor man den zweiten Auftrag startet oder mit 2 parallelen Verbindungen arbeiten, wenn der Partner das ermöglicht.

Parameter	Bezeichnung	Datentyp	Beschreibung
REQ	INPUT	BOOL	Anforderung des Datenaustausches bei steigender Flanke
ID	INPUT	INT	Verbindungs-ID-Nummer, welche in in der ConfigStage konfiguriert wurde
DONE	OUTPUT	BOOL	TRUE = Auftrag erfolgreich abgeschlossen, FALSE = Auftrag ist in Abarbeitung oder fehlgeschlagen
ERROR	OUTPUT	BOOL	TRUE = Auftrag fehlgeschlagen
STATUS	OUTPUT	WORD	Code des Auftragsstatus
ADDR	IN_OUT	ANY	ANY-Pointer zum Bereich, in den die Daten geschrieben werden sollen.
SD	IN_OUT	ANY	ANY-Pointer zum Bereich der lokalen SPS, in dem die zu sendenden Daten bereitgehalten werden.

Kommunikation - Ethernet

DONE, ERROR und STATUS Parameterbedeutung

DONE	ERROR	STATUS W#16#..	Beschreibung
1	0	0	Auftrag erfolgreich abgeschlossen.
0	0	7000	Auftrag wurde nicht aktiviert, weil: • es der erste Aufruf mit REQ=0 ist oder weil • ein anderer Auftrag (Get) noch aktiv ist
0	0	7001	Erster Aufruf mit REQ=1. Auftrag wurde aktiviert und ist in Bearbeitung.
0	0	7002	Zwischenzeitlicher Aufruf (REQ irrelevant). Auftrag in Bearbeitung.
0	0	7003	Auftragsaktivierung wird ausgeführt in einer Klasse mit niedriger Priorität (Verschachtelungsaufwurf).
0	1	8001	Parameter-ID ist nicht korrekt.
0	1	8002	Verbindung mit ID ist nicht konfiguriert oder inkorrekt Verbindungstyp.
0	1	8003	Auftrag kann nicht zugewiesen werden solange vorhergehender Auftrag nicht abgearbeitet wurde.
0	1	8004	Parameter RD ist nicht korrekt.
0	1	8005	Partner-SPS konnte nicht verbunden werden.
0	1	8006	Auftrag kann nicht aktiviert werden wegen des Verbindungsstatus´.
0	1	8007	Auftrag fehlgeschlagen wegen Verbindungsproblem (Kabel abgezogen, Kommunikation durch Partner zurückgewiesen).
0	1	8008	Auftrag fehlgeschlagen wegen einer negativen Antwort vom Partner.
0	1	8009	Auftrag fehlgeschlagen wegen eines Zugriffsfehlers zum Partnerbereich.
0	1	800A	Auftrag fehlgeschlagen wegen eines Zugriffsfehlers zum lokalen Bereich.
0	1	800B	Parameter ADDR ist nicht korrekt.

S7-Programmbeispiel für die Verwendung des SFB121

Ziel: Schreiben von 20Byte Daten, beginnend mit MB200, aus einer lokalen CPU in eine Partner-CPU, beginnend bei MB500

```

CALL SFB121, DB121
REQ   :=M600.0           // TRUE = Schreibe Daten an Partner S7-CPU
ID     :=1                // In ConfigStage konfigurierte Verbindungs-ID-Nummer
DONE   :=M600.1           // TRUE = Erfolgreich geschrieben
ERROR  :=M600.2           // TRUE = Fehler aufgetreten
STATUS:=MW602             // Ergebnis / Rückgabewert
ADDR   :=P#M 500.0 BYTE 20 // 20 Byte schreiben an Partner S7-CPU
SD     :=P#M 200.0 BYTE 20 // 20 Byte Daten beginnend von MB200 in lokaler S7-CPU

O      M      600.1        // Wenn erfolgreich geschrieben wurde
O      M      600.2        // oder Fehler aufgetreten sind, dann
R      M      600.0        // setze Anforderungssignal zurück.
```

Kommunikation - Ethernet

Lesen der IP-Konfiguration mit dem SFB 131 "IP_GET" (CPU-T)

Der SFB131 "IP_GET" wird benutzt, um die Einstellungen und den Modus der Ethernet-Ports einzulesen.



Hinweis:

Dieser FB ist nur für SPSen mit der CPU-T verfügbar.
Für CPU-V/P-SPSen steht der SFB 129 "IP_CFG" zur Verfügung.

Parameter	Bezeichnung	Datentyp	Beschreibung
REQ	INPUT	BOOL	TRUE: Erhalte die Ethernet-Einstellungen
PORT	INPUT	INT	Portnummer. 1=Ethernet 1; 2=Ethernet 2
DONE	OUTPUT	BOOL	TRUE: Auftrag erfolgreich ausgeführt
ERROR	OUTPUT	BOOL	TRUE: Auftrag fehlgeschlagen
STATUS	OUTPUT	WORD	Statuscode
IP_ADDR	OUTPUT	DWORD	IP-Adresse des Ethernet-Ports z.B. 192.168.80.50 => DW#16#C0A85032
NETMASK	OUTPUT	DWORD	Netzmaske des Ethernet-Ports z.B. 255.255.255.0 => DW#16#FFFFFF00
ROUTER	OUTPUT	DWORD	IP-Adresse des Routers DW#16#0 - Kein Router
MAC	OUTPUT	ARRAY[1..6] OF BYTE	MAC-Adresse des Ethernet-Ports
PHYMODE	OUTPUT	BYTE	Ethernet-Port-Modus Bit 7: 1 = Modus ist gültig, 0 = Modus ist ungültig Bit 3: 1 = Autonegotiation ist aktiviert, 0 = Manueller Modus Bit 2: Link-Status 1 = Link, 0 = Kein Link Bit 1: Geschwindigkeit 1 = 100M, 0 = 10M Bit 0: Duplex-Modus 1 = Voll, 0 = Halb

Parameterbedeutung:

STATUS	DONE	ERROR	Beschreibung
W#16#0000	1	0	Einstellungen erfolgreich erhalten
W#16#7000	0	0	REQ ist FALSE
W#16#8000	0	1	Ungültige Portnummer. Port muss 1 oder 2 sein.

S7-Programmbeispiel: Lesen der IP-Einstellungen und des Modus' des Ethernet-Ports 1

```

CALL SFB 131 , DB131
REQ      :=M1502.5
PORT     :=1
DONE     :=M1002.0
ERROR    :=M1002.1
STATUS   :=MW1000
IP_ADDR  :=MD2030
NETMASK  :=MD2034
ROUTER   :=MD2038
MAC      :=#MacAddr
PHYMODE  :=MB1800

CLR
=      M      1502.5
U      M      1002.0
BEB
L      MW      1000
// error evaluation
// ...
BEA
// if successful done
// end the function
// error occurred
// end of function

```

Schreiben der IP-Konfiguration mit dem SFB 132 "IP_SET" (CPU-T)

Der SFB132 "IP_GET" wird benutzt, um die Einstellungen und den Modus der Ethernet-Ports zu überschreiben.



Hinweis:

Dieser FB ist nur für SPSen mit der CPU-T verfügbar.
Für CPU-V/P-SPSen steht der SFB 129 "IP_CFG" zur Verfügung.

Parameter	Bezeichnung	Datentyp	Beschreibung
REQ	INPUT	BOOL	TRUE: Anfrage um die IP-Einstellungen zu schreiben
PORT	INPUT	INT	Portnummer. 1=Ethernet 1; 2=Ethernet 2
IP_ADDR	OUTPUT	DWORD	IP-Adresse des Ethernet-Ports z.B. 192.168.80.50 => DW#16#C0A85032
NETMASK	OUTPUT	DWORD	Netzmaske des Ethernet-Ports e.g 255.255.255.0 => DW#16#FFFFFF00
ROUTER	OUTPUT	DWORD	IP-Adresse des Routers DW#16#0 - Kein Router
DONE	OUTPUT	BOOL	TRUE: Auftrag erfolgreich ausgeführt
ERROR	OUTPUT	BOOL	TRUE: Auftrag fehlgeschlagen
STATUS	OUTPUT	WORD	Statuscode

Parameterbedeutung:

STATUS	DONE	ERROR	Beschreibung
W#16#0000	1	0	Einstellungen erfolgreich geschrieben
W#16#7000	0	0	REQ ist FALSE
W#16#8000	0	1	Ungültige Portnummer. Port muss 1 oder 2 sein.

S7-Programmbeispiel: Schreiben der IP-Einstellungen des Ethernet-Ports 1

```

CALL SFB 132 , DB132
REQ      :=M1502.6
PORT     :=1
IP_ADDR:=MD2030
NETMASK:=MD2034
ROUTER  :=MD2038
DONE     :=M1002.0
ERROR    :=M1002.1
STATUS  :=MW1000

CLR
=      M   1502.6
U      M   1002.0
BEB
L      MW  1000
// error evaluation
// ...
BEA
// if successful done
// end the function
// error occurred
// end of function

```


Kommunikation - Ethernet

TCP Send/Receive-Verbindung (RFC 793)

INSEVIS-Steuerungen bieten folgende Systemfunktionsblöcke (SFB) für den programmierbaren Datenaustausch über TCP (RFC 793) mit anderen ethernetfähigen Kommunikationspartnern;

- SFB 122 "TSEND" Daten senden über die TCP Send/Receive-Verbindung
- SFB 123 "TREC" Daten erhalten über die TCP Send/Receive-Verbindung
- SFB 124 "TDISCON" Beenden der TCP Send/Receive-Verbindung
- SFB 125 "TCONFIG" Konfiguration der Partnerparameter der TCP Send/Receive-Verbindung
- SFB 126 "TSTATUS" Statusabfrage der TCP Send/Receive-Verbindung

Die Verbindung muss in der ConfigStage konfiguriert und anschließend in die SPS geladen werden. In der Konfiguration muss der Verbindungsaufbautyp (aktiv oder passiv) zugewiesen werden.

Aktive Verbindungen initiieren den Verbindungsaufbau selbst, die mit passiven warten auf einen Aufbau seitens des Verbindungspartners.

Bei der aktiven Verbindung sollten die Partnerparameter zugewiesen sein, sie können aber auch zur Laufzeit mit dem SFB 125 "TCONFIG" geändert werden.



ACHTUNG:

Bei passiven Verbindungen muss die lokale Port-Nummer zugewiesen und eindeutig (einmalig vergeben) sein.

Daten senden über TCP mit dem SFB 122 "TSEND"

Mit dem SFB 122 "TSEND" können Sie über TCP (RFC 793) zu einem Remote-Partner Daten senden. Dieses funktioniert asynchron, d.h., die Abarbeitung erfolgt über mehrere SFB-Aufrufe.

Die Parameter ID, LEN, DATA werden übertragen, die Übertragung beginnt mit der steigenden Flanke bei REQ. Der neue Auftrag kann nur akzeptiert werden, wenn der vorherige Auftrag beendet wurde und bei der steigenden Flanke bei REQ.

Die Abarbeitung des Auftrages wird signalisiert durch DONE oder ERROR Parameter.

Parameter	Bezeichnung	Datentyp	Beschreibung
REQ	INPUT	BOOL	Anforderung des Datenaustausches bei steigender Flanke
ID	INPUT	INT	Verbindungs-ID-Nummer, welche in in der ConfigStage konfiguriert wurde
LEN	INPUT	INT	Anzahl der zu sendenden Bytes
DONE	OUTPUT	BOOL	TRUE = Auftrag erfolgreich abgeschlossen, FALSE = Auftrag ist in Abarbeitung oder fehlgeschlagen
BUSY	OUTPUT	BOOL	TRUE = Auftrag in Abarbeitung
ERROR	OUTPUT	BOOL	TRUE = Auftrag fehlgeschlagen
STATUS	OUTPUT	WORD	Code des Auftragsstatus
DATA	IN_OUT	ANY	ANY-Pointer auf den Sendebereich, Adresse verweist auf: Eingangsabbild (E), Ausgangsabbild (A), Merker (M), Datenbaustein (DB)

Kommunikation - Ethernet

DONE, BUSY, ERROR und STATUS Parameterbedeutung

DONE	BUSY	ERROR	STATUS W#16#..	Beschreibung
1	0	0	0	Auftrag erfolgreich abgeschlossen.
0	0	0	7000	Erster Aufruf mit REQ=0. Auftrag wurde nicht aktiviert.
0	1	0	7001	Erster Aufruf mit REQ=1. Auftrag wurde aktiviert und ist in Bearbeitung.
0	1	0	7002	Zwischenzeitlicher Aufruf (REQ irrelevant). Auftrag in Bearbeitung.
0	0	0	7003	Auftragsaktivierung wird ausgeführt in einer Klasse mit niedriger Priorität (Verschachtelungsaufruf).
0	0	1	8001	Parameter-ID ist nicht korrekt.
0	0	1	8002	Verbindung mit ID ist nicht konfiguriert oder inkorrektart Verbindungstyp.
0	0	1	8003	Parameter DATA ist nicht korrekt. Nur E, A, M, DB Bereiche erlaubt
0	0	1	8004	Parameter DATA ist nicht korrekt. z.B. DB nicht geladen
0	0	1	8005	Parameter LEN ist 0 oder größer als angegeben unter Parameter DATA
0	0	1	8006	Keine Verbindung zu Partner aufgebaut
0	0	1	8007	Auftrag fehlgeschlagen wegen Verbindungsproblem (Kabel abgezogen, Kommunikation durch Partner zurückgewiesen).

S7-Programmbeispiel für die Verwendung des SFB122

Ziel: Senden von 20Byte Daten, beginnend mit MB200, an einen Partner

```

CALL SFB122, DB122
REQ :=M600.0           // TRUE = Senden der Daten an den Partner
ID :=1                 // In ConfigStage konfigurierte Verbindungs-ID-Nummer
LEN :=20               // 20 Byte Daten zu senden
DONE :=M600.1          // TRUE = Erfolgreich gesendet
BUSY :=M600.2          // TRUE = In Abarbeitung
ERROR :=M600.3         // TRUE = Fehler aufgetreten
STATUS:=MW602          // Ergebnis / Rückgabewert
DATA :=P#M 200.0 BYTE 20 // 20 Byte Daten, beginnend von MB200 in lokaler CPU

O M 600.1              // Wenn erfolgreich gesendet wurde
O M 600.3              // oder Fehler aufgetreten sind,
R M 600.0              // setze Anforderungssignal zurück.
```

Kommunikation - Ethernet

Daten empfangen über TCP mit dem SFB 123 "TREC"V"

Mit dem SFB 123 "TREC"V" können Sie über TCP (RFC 793) von einem Remote-Partner Daten empfangen. Dieses funktioniert asynchron, d.h., die Abarbeitung erfolgt über mehrere SFB-Aufrufe.

Die Parameter ID, LEN, DATA werden ausgewertet und starten den Empfang bei EN_R = TRUE.

Der neue Auftrag kann nur akzeptiert werden, wenn der vorherige Auftrag beendet wurde und bei EN_R = TRUE.

Die Abarbeitung des Auftrages wird signalisiert durch NDR oder ERROR Parameter.

Wenn die Länge der zu empfangenden Daten (LEN-Parameter) größer ist als die gerade empfangenen, wird der SFB die empfangenen Daten in den Empfangsbereich (DATA-Parameter) kopieren und NDR wird erst TRUE, wenn die durch den LEN-Parameter vorgegebene Länge erreicht wurde.

Wenn die Länge der zu empfangenden Daten (LEN-Parameter) kleiner ist als die gerade empfangenen, wird der SFB so viele Bytes der empfangenen Daten in den Empfangsbereich (DATA-Parameter) kopieren, wie im LEN-Parameter voreingestellt wurde. NDR wird auf TRUE gesetzt und in den Parameter LEN_R wird die Länge der empfangenen Daten geschrieben. Mit jeden weiteren Aufruf wird der Rest der Daten empfangen.

Wenn der LEN-Parameter = 0, wird der SFB die empfangenen Daten in den Empfangsbereich (DATA parameter) kopieren und sofort NDR auf TRUE setzen.

Parameter	Bezeichnung	Datentyp	Beschreibung
EN_R	INPUT	BOOL	Datenempfang aktivieren
ID	INPUT	INT	Verbindungs-ID-Nummer, welche in in der ConfigStage konfiguriert wurde
LEN	INPUT	INT	Erwartete Empfangsdatengröße in Bytes
NDR	OUTPUT	BOOL	TRUE = Auftrag erfolgreich abgeschlossen, FALSE = Auftrag ist in Abarbeitung oder fehlgeschlagen
BUSY	OUTPUT	BOOL	TRUE = Auftrag in Abarbeitung
ERROR	OUTPUT	BOOL	TRUE = Auftrag fehlgeschlagen
STATUS	OUTPUT	WORD	Code des Auftragsstatus
LEN_R	OUTPUT	INT	Anzahl der aktuell empfangenen Bytes
DATA	IN_OUT	ANY	ANY-Pointer auf den Empfangsbereich, Adresse verweist auf: Eingangsabbild (E), Ausgangsabbild (A), Merker (M), Datenbaustein (DB)

DONE, BUSY, ERROR und STATUS Parameterbedeutung

DONE	BUSY	ERROR	STATUS W#16#..	Beschreibung
1	0	0	0	Auftrag erfolgreich abgeschlossen.
0	0	0	7000	Erster Aufruf mit EN_R=0. Auftrag wurde nicht aktiviert.
0	1	0	7001	Erster Aufruf mit EN_R=1. Auftrag wurde aktiviert und ist in Bearbeitung.
0	1	0	7002	Zwischenaufruf (EN_R irrelevant). Auftrag ist in Abarbeitung.
0	0	0	7003	Auftragsaktivierung wird ausgeführt in einer Klasse mit niedriger Priorität (Verschachtelungsaufruf).
0	0	1	8001	Parameter ID ist nicht korrekt.
0	0	1	8002	Verbindung mit ID ist nicht konfiguriert oder inkorrekt Verbindungstyp.
0	0	1	8003	Parameter DATA ist nicht korrekt. Nur E, A, M, DB Bereiche erlaubt
0	0	1	8004	Parameter DATA ist nicht korrekt. z.B. DB nicht geladen
0	0	1	8005	Parameter LEN größer als angegeben unter Parameter DATA
0	0	1	8006	Keine Verbindung zu Partner aufgebaut
0	0	1	8007	Auftrag fehlgeschlagen wegen Verbindungsproblem (Kabel abgezogen, Kommunikation durch Partner zurückgewiesen).

Kommunikation - Ethernet

DONE, BUSY, ERROR und STATUS Parameterbedeutung (Erweiterung für Modbus-TCP-Client und ISO-on-TCP ab FW 2.9.4)

DONE	BUSY	ERROR	STATUS W#16#..	Beschreibung
0	0	1	8008	Modbus-TCP Client - Invalid mapping configuration ISOonTCP Send/Receive - Too large data packet
0	0	1	8009	Modbus-TCP Client - Invalid response from Modbus-TCP server ISOonTCP Send/Receive - Invalid response
0	0	1	800A	Modbus-TCP Client: System controlled transfer is active

S7-Programmbeispiel für die Verwendung des SFB123

Ziel: Empfangen von Daten von Partner und Schreiben in Puffer, beginnend bei MB200 und 200 Bytes lang.

```

CALL SFB123, DB123
EN_R   :=M600.0           // TRUE = Aktivieren, um Daten zu empfangen
ID     :=1                // In ConfigStage konfigurierte Verbindungs-ID-Nummer
LEN    :=0                // 0 = Wenn Daten empfangen, dann sofort NDR setzen und
                           // nach DATA kopieren
NDR    :=M600.1           // TRUE = Neue Daten verfügbar
BUSY   :=M600.2           // TRUE = In Abarbeitung
ERROR  :=M600.3           // TRUE = Fehler aufgetreten
STATUS:=MW602             // Ergebnis / Rückgabewert
LEN_R  :=MW604            // Anzahl der Bytes, welche nach DATA kopiert wurden
DATA   :=P#M 200.0 BYTE 200 // Empfangspufferbereich

O      M      600.1       // Wenn erfolgreich empfangen wurde
O      M      600.3       // oder Fehler aufgetreten sind,
R      M      600.0       // setze Anforderungssignal zurück.
```

Kommunikation - Ethernet

Beenden der TCP-Verbindung mit dem SFB 124 "TDISCON"

Der SFB 124 "TDISCON" beendet eine Kommunikation der Verbindungsart TCP oder S7 Client Verbindung. Dieses funktioniert asynchron, d.h., die Abarbeitung erfolgt über mehrere SFB-Aufrufe.

Parameter	Bezeichnung	Datentyp	Beschreibung
REQ	INPUT	BOOL	Anforderung, die Beendigung der Verbindung bei TRUE zu starten
ID	INPUT	INT	Verbindungs-ID-Nummer, welche in in der ConfigStage konfiguriert wurde
DONE	OUTPUT	BOOL	TRUE = Auftrag erfolgreich abgeschlossen, FALSE = Auftrag ist in Abarbeitung oder fehlgeschlagen
BUSY	OUTPUT	BOOL	TRUE = Auftrag in Abarbeitung
ERROR	OUTPUT	BOOL	TRUE = Auftrag fehlgeschlagen
STATUS	OUTPUT	WORD	Code des Auftragsstatus

DONE, BUSY, ERROR und STATUS Parameterbedeutung

DONE	BUSY	ERROR	STATUS W#16#..	Beschreibung
1	0	0	0	Auftrag erfolgreich abgeschlossen.
0	0	0	7000	Erster Aufruf mit REQ=0. Auftrag wurde nicht aktiviert.
0	1	0	7001	Erster Aufruf mit REQ=1. Auftrag wurde aktiviert und ist in Bearbeitung.
0	1	0	7002	Zwischenzeitlicher Aufruf (REQ irrelevant). Auftrag in Bearbeitung.
0	0	1	8001	Parameter ID ist nicht korrekt.
0	0	1	8002	Verbindung mit ID ist nicht konfiguriert oder inkorrekt Verbindungstyp.

S7-Programmbeispiel für die Verwendung des SFB124

Ziel: Verbindung unterbrechen

```

CALL SFB124, DB124
REQ  :=M600.0           // TRUE = Unterbreche die Verbindung
ID   :=1               // In ConfigStage konfigurierte Verbindungs-ID-Nummer
DONE :=M600.1           // TRUE = Erfolgreich
BUSY :=M600.2           // TRUE = In Abarbeitung
ERROR:=M600.3           // TRUE = Fehler aufgetreten
STATUS:=MW602           // Ergebnis / Rückgabewert

O     M     600.1        // Wenn erfolgreich unterbrochen wurde
O     M     600.3        // oder Fehler aufgetreten sind,
R     M     600.0        // setze Anforderungssignal zurück.
```

Konfiguration der Partnerparameter bei TCP mit dem SFB 125 "TCONFIG"

Mit dem SFB 125 "TCONFIG" können Sie die Partnerparameter (Port-Nummer und IP Adresse, bei der S7 Client Verbindung auch die TSAP-Nummer) spezifizieren.

Bitte beachten, dass vor der Konfiguration die Ethernet-Verbindung zum Partner automatisch getrennt wird.

Die neuen Konfigurationsparameter sind mit dem nächsten Send/Receive-Auftrag gültig.

Parameter	Bezeichnung	Datentyp	Beschreibung
REQ	INPUT	BOOL	Anforderung, die Beendigung der Verbindung bei TRUE zu starten
ID	INPUT	INT	Verbindungs-ID-Nummer, welche in in der ConfigStage konfiguriert wurde
PORT	INPUT	WORD	Partner-Port/TSAP-Nummer z.B. Port 2000 _{dec} , eingeben als W#16#7D0
IPADDR	INPUT	DWORD	Partner-IP-Adresse z.B. Adresse 192.168.80.50, eingeben als DW#16#C0A85032
TCONN	INPUT	WORD	Reserviert, muss 0 sein
TRESP	INPUT	WORD	Reserviert, muss 0 sein
DONE	OUTPUT	BOOL	TRUE = Auftrag erfolgreich abgeschlossen, FALSE = Auftrag ist in Abarbeitung oder fehlgeschlagen
BUSY	OUTPUT	BOOL	TRUE = Auftrag in Abarbeitung
ERROR	OUTPUT	BOOL	TRUE = Auftrag fehlgeschlagen
STATUS	OUTPUT	WORD	Code des Auftragsstatus

DONE, BUSY, ERROR und STATUS Parameterbedeutung

DONE	BUSY	ERROR	STATUS W#16#..	Beschreibung
1	0	0	0	Auftrag (Konfiguration) erfolgreich abgeschlossen.
0	0	0	7000	Erster Aufruf mit REQ=0. Auftrag wurde nicht aktiviert.
0	1	0	7001	Erster Aufruf mit REQ=1. Auftrag wurde aktiviert und ist in Bearbeitung.
0	1	0	7002	Zwischenzeitlicher Aufruf (REQ irrelevant). Auftrag in Bearbeitung.
0	0	1	8001	Parameter ID ist nicht korrekt.
0	0	1	8002	Verbindung mit ID ist nicht konfiguriert oder inkorrekt Verbindungstyp.
0	0	1	8003	Parameter IPADDR ist nicht korrekt (z.B., 0 durch aktiven Verbindungsaufbau)

S7-Programmbeispiel für die Verwendung des SFB125

Ziel: Ändern der Partner-Konfiguration auf IP-Adresse 192.168.80.52 und Port-Nummer 2000

```

CALL SFB125, DB125
REQ   :=M600.0           // TRUE = Ändere Partner-Konfiguration
ID    :=1                // In ConfigStage konfigurierte Verbindungs-ID-Nummer
PORT  :=W#16#7D0         // Partner-Port-Nummer z.B. 2000 (dez)
IPADDR:=DW#16#C0A85034   // Partner-IP-Adresse z.B. 192.168.80.52
TCONN :=W#16#0           // Reserviert, muss „0“ sein
TRESP :=W#16#0           // Reserviert, muss „0“ sein
DONE  :=M600.1           // TRUE = Erfolgreich
BUSY   :=M600.2           // TRUE = In Abarbeitung
ERROR  :=M600.3           // TRUE = Fehler aufgetreten
STATUS:=MW602            // Ergebnis / Rückgabewert

O      M      600.1       // Wenn erfolgreich zugewiesen wurde
O      M      600.3       // oder Fehler aufgetreten sind,
R      M      600.0       // setze Anforderungssignal zurück.

```


Kommunikation - Ethernet

Statusabfrage bei der TCP-Verbindung mit dem SFB 126 "TSTATUS"

Mit dem SFB 126 "TSTATUS" können Sie den aktuellen Status der Verbindung (TCP, S7 Client-Verbindung) abfragen.

Parameter	Bezeichnung	Datentyp	Beschreibung
ID	INPUT	INT	Verbindungs-ID-Nummer, welche in in der ConfigStage konfiguriert wurde
DONE	OUTPUT	BOOL	TRUE = Auftrag erfolgreich abgeschlossen, FALSE = Auftrag ist in Abarbeitung oder fehlgeschlagen
ERROR	OUTPUT	BOOL	TRUE = Auftrag fehlgeschlagen
STATUS	OUTPUT	WORD	Code des Auftragsstatus
PORT	OUTPUT	WORD	Partner-Port/TSAP-Nummer z.B. Port 2000 _{dec} , ausgegeben als W#16#7D0
IPADDR	OUTPUT	DWORD	Partner-IP-Adresse z.B. Adresse 192.168.80.50, ausgegeben als DW#16#C0A85032

DONE, ERROR und STATUS Parameterbedeutung

DONE	ERROR	STATUS W#16#..	Beschreibung
1	0	0XYZ	Auftrag erfolgreich abgeschlossen. Bit 0-3: Status des Sendeauftrages Bit 0: 1=Verbindung akzeptiert den Sendeauftrag Bit 1: 1=Sendeauftrag in Abarbeitung Bit 2: 1=Sendeauftrag erfolgreich abgeschlossen Bit 3: 1=Sendeauftrag fehlgeschlagen Bit 4-7: Status des Empfangsauftrages Bit 4: 1=Verbindung akzeptiert den Empfangsauftrag Bit 5: 1=Empfangsauftrag in Abarbeitung Bit 6: 1=Empfangsauftrag erfolgreich abgeschlossen Bit 7: 1=Empfangsauftrag fehlgeschlagen Bit 8-10: Status der Verbindung 001 = Aktiver Aufbau in Bearbeitung 010 = Verbindungsbeendung 011 = Verbindung hergestellt/ verbunden mit Partner Bit 11: 1=aktiver Verbindungsaufbau Beispiel: W#16#B31 bedeutet: 1 = 0001 = Verbindung akzeptiert den Sendeauftrag 3 = 0011 = Verbindung akzeptiert den Empfangsauftrag + Empfangsauftrag in Abarbeitung B = 1011 = Verbindung hergestellt/ verbunden mit Partner + aktiver Verbindungsaufbau Die Parameter PORT, IPADDR beinhalten Partner-Portnummer und IP-Adresse.
0	0	7000	Erster Aufruf mit REQ=0. Auftrag wurde nicht aktiviert.
0	0	7001	Erster Aufruf mit REQ=1. Auftrag wurde aktiviert und ist in Bearbeitung.
0	0	7002	Zwischenzeitlicher Aufruf (REQ irrelevant). Auftrag in Bearbeitung.
0	1	8001	Parameter ID ist nicht korrekt.
0	1	8002	Verbindung mit ID ist nicht konfiguriert oder inkorrekt Verbindungstyp.
0	1	8003	Verbindung ist noch nicht aufgebaut

S7-Programmbeispiel für die Verwendung des SFB126

Ziel: Abfrage des Verbindungsstatus

```

UN   M   600.0      // TRUE = Anforderung zu Abfrage des Verbindungsstatus
BEB                                     // andernfalls Beenden der Funktion

CALL #TSTATUS
ID   :=1           // In ConfigStage konfigurierte Verbindungs-ID-Nummer
DONE :=M600.1      // TRUE = Erfolgreich
ERROR:=M600.3      // TRUE = Fehler aufgetreten
STATUS:=MW602      // Ergebnis / Rückgabewert
PORT :=MW604       // Partner-Port-Nummer
IPADDR:=MD606      // Partner-IP-Adresse

O    M   600.1      // Wenn Abfrage erfolgreich
O    M   600.3      // oder Fehler aufgetreten sind,
R    M   600.0      // setze Anforderungssignal zurück.
    
```

ISO on TCP

Die Konfiguration einer ISO-on-TCP-Protokolls entspricht im Wesentlichen der Konfiguration einer TCP-Verbindung mit einer anderen TSAP-Einstellung und ist ab Betriebssystem 2.9.4 für alle SPSen mit T-CPUs ab ConfigStage 1.0.19.2 möglich.

In der ConfigStage unter bei Ethernet eine neue Verbindung einfügen und „ISO-on-TCP Send/Receive“ auswählen.

In der Maske „Aktiver Verbindungsaufbau“ oder „Unspezifiziert“ auswählen, den lokalen TSAP (max. 16 Stellen in ASCII oder hexadezimal eingeben und (bei aktivem Verbindungsaufbau) die IP-Adresse des Verbindungspartners eingeben.

Eigenschaften - ISOonTCP Send/Receive Verbindung

☒ Aktiver Verbindungsaufbau
☐ Unspezifiziert

Lokal

TSAP (ASCII): ISOonTCP-01
 TSAP (hex): 49.53.4F.6F.6E.54.43.50.2D.30.31
 TSAP-Länge: 11

Partner

IP-Adresse: 192.168.80.12
 TSAP (ASCII): ISOonTCP-02
 TSAP (hex): 49.53.4F.6F.6E.54.43.50.2D.30.32
 TSAP-Länge: 11

✓ Ok ✗ Abbrechen

Kommunikation - Ethernet

Modbus-TCP (Client)

Die Modbus-TCP-Client-Verbindung wird nur in CPU-T-Geräten SPS-Betriebssystem 2.9.4 möglich ab ConfigStage 1.0.19.2 unterstützt.

SPSen mit der CPU-T unterstützen maximal 16 TCP-basierte Verbindungen (also maximal 16 Modbus-TCP-Clients). Jede Modbus-TCP-Client-Verbindung kann maximal 16 „Datenbereichszuordnungen“ (Mappings) haben.

In der ConfigStage unter bei Ethernet eine neue Verbindung einfügen und „Modbus TCP Client“ auswählen.

ModBus Bereich	Startadresse	Anzahl	S7 Bereich	Blocknummer	Byte-Offset	Länge in Bytes	Byte-Swap
1 Inputregister	0	125	Datenbaustein	1	0	250	Nein
2 Diskrete Eingänge (Bits)	0	2000	Datenbaustein	1	0	250	
3 Halte- (Ausgangs) Register	0	123	Datenbaustein	1	0	246	Nein
4 Coils (Ausgangsbits)	0	1968	Datenbaustein	1	0	246	

Folgende Modbus-Funktionen wurden fest mit den konfigurierbaren Modbusbereichen verknüpft:

Inputregister (Eingänge):	Modbus-Funktion 4 „Eingangsregister lesen“
Diskrete Eingänge (Eingangs-Bits):	Modbus-Funktion 2 „Eingangsbits lesen“
Holding-Register (Ausgänge):	Modbus-Funktion 16 „Mehrere Register schreiben“
Coils (Ausgangs-Bits):	Modbus-Funktion 15 „Mehrere Coils schreiben“

Zur Laufzeit gibt es 2 Wege, die zugewiesenen (gemappten) Daten zwischen dem Modbus-TCP-Server und der SPS als Modbus-TCP-Client zu übertragen.

1. Verwendung von SFB 123 „TRECVC“

Rufen Sie im S7-Programm den SFB 123 „TRECVC“ auf, um Daten vom/zum Modbus-TCP-Server aus dem/ zum zugeordneten S7-Bereich zu übertragen. Es werden alle konfigurierten Zuordnungen in einem Auftrag übertragen.

Der Parameter „DATA“ des SFB 123 „TRECVC“ sollte auf ein BYTE-Array mit der Größe der Anzahl der in ConfigStage konfigurierten zugeordneten Bereiche verweisen.

Nach erfolgreichem Abschluss von SFB 123 „TRECVC“ enthält das Array-Element den Rückgabewert für jede Zuordnung vom Modbus-TCP-Server wie folgt:

Rückgabewert	Bedeutung
0, 1, 2, 3, 4	Modbus-TCP-Server-Antwort-Exception Code
252	Ungültige Zuordnungs-(Mapping)-Konfiguration
253	Der zugewiesene S7-Bereich ist zu klein für die vom Modbus-TCP-Server empfangenen Daten (z. B. ist der konfigurierte Datenbaustein nicht groß genug).
254	Der zugeordnete S7-Bereich wurde nicht gefunden (z. B. Wurde der konfigurierte Datenbaustein nicht in die SPS geladen).
255	Verbindung wird nicht hergestellt

2. Systemgesteuerte Übertragung

In ConfigStage kann mit einem Haken die systemgesteuerte Übertragung durch Zuweisung folgender Parameter konfiguriert werden („Systemsteuerung aktivieren“):

„Verbindungsstatus“	Byte-Adresse
„Trigger-Bits“	boolisches Array mit der Größe der Anzahl der Zuordnungen
„Status-Bytes“	Byte-Array mit der Größe der Anzahl der Zuordnungen

Das Betriebssystem überprüft regelmäßig den Bereich „Trigger-Bits“ und startet bei einer steigenden Flanke des Bits die Übertragung der entsprechenden Zuordnung.

Nach Abschluss der Übertragung setzt das System das Bit zurück und schreibt den Rückgabewert (siehe oben) in das Array-Element „Status-Bytes“.

Das System schreibt den folgenden Wert in das Byte „Verbindungsstatus“:

Wert	Bedeutung
0	Verbindung wurde/ wird nicht hergestellt
1	Verbindungstrennung wird gerade durchgeführt.
2	Verbindung wird hergestellt
3	Verbindung ist hergestellt
4	Verbindung ist hergestellt und die Zuordnung wird übertragen.

Durch Setzen des Bits in „Triggerbits“ kann der Benutzer das Übertragungsintervall und/oder die Zuordnungsreihenfolge steuern.

Kommunikation - Ethernet

Modbus TCP (Server)

Zur Verwendung des Protokolls Modbus TCP als Server (Slave) muss dieser in der Config Stage konfiguriert sein.

- Modbus-TCP-Server aktivieren (Häkchen setzen)
- Mapping auf (vorhandene!) S7-Operanden definieren

Dabei werden die definierten S7-Operandenbereiche auf die Modbus-Register bzw. -Bits abgebildet.

Mit CPUs der T-Baureihe kann gleichzeitig mit bis zu 8 Modbus-TCP-Clients kommuniziert werden, bei CPUs der Baureihe V/P ist lediglich eine Verbindung möglich..

Die Anzahl unterstützter Modbus-Indizes ergibt sich aus der angegebenen Länge.

Folgende Modbus-Funktionen werden unterstützt:

1 (0x01) Coil lesen	Read Coils
2 (0x02) Eingangsbits lesen	Read Discrete Inputs
3 (0x03) Holding Register lesen	Read Holding Registers
4 (0x04) Eingangsregister lesen	Read Input Registers
5 (0x05) Coil schreiben	Write Single Coil
6 (0x06) Register schreiben	Write Single Register
15 (0x0F) Mehrere Coils schreiben	Write Multiple Coils
16 (0x10) Mehrere Register schreiben	Write Multiple Registers

In S7 muss nichts programmiert werden. Die Daten werden im Hintergrund kontrollpunktsynchron vom Betriebssystem bearbeitet.

UDP Send/Receive-Verbindung (RFC 768)

INSEVIS-Steuerungen bieten folgende Systemfunktionsblöcke (SFB) für den programmierbaren Datenaustausch über UDP (RFC 768) mit anderen ethernetfähigen Kommunikationspartnern;

- SFB 127 "USEND" Daten senden über UDP Send/Receive-Verbindung
- SFB 128 "URECV" Daten empfangen über UDP Send/Receive-Verbindung

Die Verbindung muss in der ConfigStage konfiguriert und anschließend in die SPS geladen werden.



ACHTUNG:

Bei passiven Verbindungen muss die lokale Port-Nummer zugewiesen und eindeutig (einmalig vergeben) sein.

Daten senden über UDP mit dem SFB 127 "USEND"

Mit dem SFB 127 "USEND" können Sie über UDP (RFC 768) zu einem Remote-Partner Daten senden. Dieses funktioniert asynchron, d.h., die Abarbeitung erfolgt über mehrere SFB-Aufrufe. Es können bis zu 1472 Bytes gesendet werden.

Die Parameter ID, LEN, R_PORT, R_ADDR, DATA werden übertragen, die Übertragung beginnt mit der steigenden Flanke bei REQ. Der neue Auftrag kann nur akzeptiert werden, wenn der vorherige Auftrag beendet wurde und bei der steigenden Flanke bei REQ.

Die Abarbeitung des Auftrages wird signalisiert durch DONE oder ERROR Parameter.

Parameter	Bezeichnung	Datentyp	Beschreibung
REQ	INPUT	BOOL	Anforderung des Datenaustausches bei steigender Flanke
ID	INPUT	INT	Verbindungs-ID-Nummer, welche in in der ConfigStage konfiguriert wurde
LEN	INPUT	INT	Anzahl der zu sendenden Bytes
R_PORT	INPUT	WORD	Partner-Portnummer z.B. Port 2000 _{dec} , eingeben als W#16#7D0
R_ADDR	INPUT	DWORD	Partner-IP-Adresse, z.B. Adresse 192.168.80.50, eingeben als DW#16#C0A85032
DONE	OUTPUT	BOOL	TRUE = Auftrag erfolgreich abgeschlossen, FALSE = Auftrag ist in Abarbeitung oder fehlgeschlagen
BUSY	OUTPUT	BOOL	TRUE = Auftrag in Abarbeitung
ERROR	OUTPUT	BOOL	TRUE = Auftrag fehlgeschlagen
STATUS	OUTPUT	WORD	Auftragsstatuscode
DATA	IN_OUT	ANY	ANY-Pointer auf den Sendebereich, Adresse verweist auf: Eingangsabbild (E), Ausgangsabbild (A), Merker (M), Datenbaustein (DB)

DONE, BUSY, ERROR und STATUS Parameterbedeutung

DONE	BUSY	ERROR	STATUS W#16#..	Beschreibung
1	0	0	0	Auftrag erfolgreich abgeschlossen.
0	0	0	7000	Erster Aufruf mit REQ=0. Auftrag wurde nicht aktiviert.
0	1	0	7001	Erster Aufruf mit REQ=1. Auftrag wurde aktiviert und ist in Bearbeitung.
0	1	0	7002	Zwischenzeitlicher Aufruf (REQ irrelevant). Auftrag in Bearbeitung.
0	0	0	7003	Auftragsaktivierung wird ausgeführt in einer Klasse mit niedriger Priorität (Verschachtelungsaufruf).
0	0	1	8001	Parameter-ID ist nicht korrekt.
0	0	1	8002	Verbindung mit ID ist nicht konfiguriert oder inkorrekt Verbindungstyp.
0	0	1	8003	Parameter DATA ist nicht korrekt. Nur I, Q, M, DB Bereiche erlaubt
0	0	1	8004	Parameter DATA ist nicht korrekt. z.B. DB nicht geladen
0	0	1	8005	Parameter LEN ist 0 oder größer als angegeben unter Parameter DATA oder ist größer als 1472
0	0	1	8006	Parameter R_PORT oder R_ADDR ist nicht korrekt
0	0	1	8007	Auftrag fehlgeschlagen wegen Verbindungsproblem (Kabel abgezogen)

Kommunikation - Ethernet

S7-Programmbeispiel für die Verwendung des SFB127

Ziel: Senden von 20 Byte Daten mit UDP, beginnend bei MB200, zu Partner mit IP-Adresse 192.168.80.52 und UDP-Port-Nummer 2000

```

CALL SFB127, DB127
REQ  :=M600.0           // TRUE = Sende Daten zum Partner
ID    :=1               // In ConfigStage konfigurierte Verbindungs-ID-Nummer
LEN   :=20              // 20 Byte Daten zum Senden
R_PORT:=W#16#7D0        // Partner-UDP-Port-Nummer z.B. 2000 (dez)
R_ADDR:=DW#16#C0A85034  // Partner-IP-Adresse z.B. 192.168.80.52
DONE  :=M600.1          // TRUE = Erfolgreich
BUSY  :=M600.2          // TRUE = In Abarbeitung
ERROR :=M600.3          // TRUE = Fehler aufgetreten
STATUS:=MW602           // Ergebnis / Rückgabewert
DATA  :=P#M 200.0 BYTE 20 // 20 Byte Daten, beginnend von MB200 in lokaler CPU

O      M      600.1      // Wenn erfolgreich gesendet
O      M      600.3      // oder Fehler aufgetreten sind,
R      M      600.0      // setze Anforderungssignal zurück.

```

Daten empfangen über UDP mit dem SFB 128 "URECV"

Mit dem 128 "URECV" können Sie über UDP (RFC 768) von einem Remote-Partner Daten empfangen. Dieses funktioniert asynchron, d.h., die Abarbeitung erfolgt über mehrere SFB-Aufrufe.

Sie können einen Remote-Partner durch Zuweisen einer spezifischen Adresse und Portnummer für diesen Remote-Partner in R_ADDR und/oder R_PORT Parameter mit EN_R = TRUE spezifizieren. Wenn Sie z.B. Daten nur von einem Partner mit der IP-Adresse 192.168.80.60 und jeder Portnummer erhalten wollen, dann setzen Sie

```

L      DW#16#C0A8503C    // IP-Adresse des Partners
T      MD 54
L      0                 // Portnummer: 0=Alle Ports
T      MW 52
SET
=      M0.0              // Start des Empfangens

...

```

```

CALL SFB128, DB128
EN_R   := M0.0
ID     := 1
LEN    := 0
NDR    := M0.1
ERROR  := M0.2
STATUS := MW2
LEN_R  := MW4
R_PORT := MW52
R_ADDR := MD54
DATA   := P#DB100.DBX0.0 BYTE 100

```

Wenn dem R_ADDR Parameter DW#16#0 zugewiesen wird, erhält der SFB128 UDP-Pakete von jedem Remote-Partner genauso wie, wenn R_PORT 0 zugewiesen wird, von jeder Portnummer.

Bitte beachten Sie, dass die R_PORT- und R_ADDR- Parameter Ein- und Ausgabe-Parameter sind. Nach dem Datenerhalt, überschreibt der SFB128 die Parameter mit der aktuellen Remote-Adresse, welche gesendet wurde und setzt den NDR-Parameter auf TRUE.

Die Parameter ID, LEN, R_PORT, R_ADDR, DATA werden ausgewertet und starten den Empfang bei EN_R = TRUE. Der neue Auftrag kann nur akzeptiert werden, wenn der vorherige Auftrag beendet wurde und bei EN_R = TRUE. Die Abarbeitung des Auftrages wird signalisiert durch NDR oder ERROR Parameter.

Kommunikation - Ethernet

Wenn die Länge der zu empfangenden Daten (LEN-Parameter) größer ist als die gerade empfangenen, wird der SFB die empfangenen Daten in den Empfangsbereich (DATA-Parameter) kopieren und NDR wird erst TRUE, wenn die durch den LEN-Parameter vorgegebene Länge erreicht wurde.

Wenn die Länge der zu empfangenden Daten (LEN-Parameter) kleiner ist als die gerade empfangenen, wird der SFB so viele Bytes der empfangenen Daten in den Empfangsbereich (DATA-Parameter) kopieren, wie im LEN-Parameter voreingestellt wurde. NDR wird auf TRUE gesetzt und in den Parameter LEN_R wird die Länge der empfangenen Daten geschrieben. Mit jeden weiteren Aufruf wird der Rest der Daten empfangen.

Wenn der LEN-Parameter = 0, wird der SFB die empfangenen Daten in den Empfangsbereich (DATA parameter) kopieren und sofort NDR auf TRUE setzen.

Sie können Partner spezifizieren durch Zuweisen von R_PORT und/oder R_ADDR ungleich "0" und dem Aufruf vom SFB mit EN_R = TRUE.

Wenn Sie "0" zu R_PORT und/oder R_ADDR zuweisen, empfangen Sie Daten von Partnern mit allen Portnummern und/oder allen IP-Adressen.

R_PORT und R_ADDR Parameter (nur NDR = TRUE) zeigen die Parameter des Partners, der gesendet hat.

Parameter	Bezeichnung	Datentyp	Beschreibung
EN_R	INPUT	BOOL	Datenempfang aktivieren
ID	INPUT	INT	Verbindungs-ID-Nummer, welche in in der ConfigStage konfiguriert wurde
LEN	INPUT	INT	Erwartete Empfangsdatengröße in Bytes
NDR	OUTPUT	BOOL	TRUE = Auftrag erfolgreich abgeschlossen, FALSE = Auftrag ist in Abarbeitung oder fehlgeschlagen
BUSY	OUTPUT	BOOL	TRUE = Auftrag in Abarbeitung
ERROR	OUTPUT	BOOL	TRUE = Auftrag fehlgeschlagen
STATUS	OUTPUT	WORD	Code des Auftragsstatus
LEN_R	OUTPUT	INT	Anzahl der aktuell empfangenen Bytes
R_PORT	IN_OUT	WORD	Partner-Portnummer z.B.: Port 2000 _{dec} , Eingabe oder Erhalt als W#16#7D0
R_ADDR	IN_OUT	DWORD	Partner-IP-Adresse z.B. Adresse 192.168.80.50, Eingabe oder Erhalt als DW#16#C0A85032
DATA	IN_OUT	ANY	ANY-Pointer auf den Empfangsbereich, Adresse verweist auf: Eingangsabbild (E), Ausgangsabbild (A), Merker (M), Datenbaustein (DB)

DONE, BUSY, ERROR und STATUS Parameterbedeutung

DONE	BUSY	ERROR	STATUS W#16#..	Beschreibung
1	0	0	0	Auftrag erfolgreich abgeschlossen.
0	0	0	7000	Erster Aufruf mit EN_R=0. Auftrag wurde nicht aktiviert.
0	1	0	7001	Erster Aufruf mit EN_R=1. Auftrag wurde aktiviert und ist in Bearbeitung.
0	1	0	7002	Zwischenaufruf (EN_R irrelevant). Auftrag ist in Abarbeitung.
0	0	0	7003	Auftragsaktivierung wird ausgeführt in einer Klasse mit niedriger Priorität (Verschachtelungsaufruf).
0	0	1	8001	Parameter ID ist nicht korrekt.
0	0	1	8002	Verbindung mit ID ist nicht konfiguriert oder inkorrektter Verbindungstyp.
0	0	1	8003	Parameter DATA ist nicht korrekt. Nur E, A, M, DB Bereiche erlaubt
0	0	1	8004	Parameter DATA ist nicht korrekt. z.B. DB nicht geladen
0	0	1	8005	Parameter LEN größer als angegeben unter Parameter DATA oder ist größer als 1472 bytes

Kommunikation - Ethernet

S7-Programmbeispiel für die Verwendung des SFB128

Ziel: Empfangen von Daten über UDP von irgendeinem Host (unspezifizierten Partner) und Schreiben in den DB200 mit 2000 Bytes Länge.

```

L      0                                // Empfange Daten
T      MW    606                        // von beliebigen Partner-UDP-Port
L      0                                // Empfange Daten
T      MD    608                        // von beliebigen Partner

CALL   SFB128, DB128
EN_R   :=M600.0                        // TRUE = Aktivieren, um Daten zu empfangen
ID     :=1                             // In ConfigStage konfigurierte Verbindungs-ID-Nummer
LEN    :=0                             // 0 = Wenn Daten empfangen, dann sofort NDR setzen und
                                         // nach DATA kopieren
NDR    :=M600.1                        // TRUE = Neue Daten verfügbar
BUSY   :=M600.2                        // TRUE = In Abarbeitung
ERROR  :=M600.3                        // TRUE = Fehler aufgetreten
STATUS:=MW602                          // Ergebnis / Rückgabewert
LEN_R  :=MW604                          // Anzahl von Bytes, die nach DATA kopiert wurden
R_PORT:=MW606                          // Partner-UDP-Port-Nummer
R_ADDR:=MD608                          // Partner-IP-Adresse
DATA   :=P#DB200.DBX 0.0 BYTE 2000    // Empfangspufferbereich

O      M      600.1                    // Wenn erfolgreich empfangen
O      M      600.3                    // oder Fehler aufgetreten sind,
R      M      600.0                    // setze Anforderungssignal zurück.

UN     M      600.1                    // Wenn Daten nicht empfangen, dann
BEB                                         // Beenden der Funktion
L      MD    608                        // Daten wurden empfangen von Partner mit IP-Adresse
L      MW    606                        // und UDP-Port-Nummer

```

Profinet IO Controller

Die INSEVIS- Steuerungen mit CPU-T können optional mit Profinet IO Controller ausgerüstet werden.



ACHTUNG:

Die Konfiguration der CPU kann entweder im Simatic®- Manager oder in der ConfigStage vorgenommen werden. Beim Download überschreibt der Simatic®- Manager alle anderen SDBs mit. Bitte **ZUERST** Profinet und evtl. CPU im Simatic®- Manager konfigurieren und in die SPS downloaden, dann Ethernet, RS232/485, Modbus, CAN und (de)-zentrale Peripherie und evtl. alternativ die CPU mit der ConfigStage konfigurieren und **DANACH** downloaden. Dieser Download überschreibt nur die betreffenden SDBs und erhält die Profinet-Konfiguration.

Beschaltung

Wenn die Option Profinet geliefert wurde, sind zwei zusätzliche RJ45-Schnittstellen (Profinet 1 und 2) bestückt. Diese können als Switch mit gemeinsamer IP-Adresse benutzt werden.

Konfiguration im SimaticManager

Profinet ist bei INSEVIS als Option zusätzlich zur CPU verfügbar. Um diese Option auch in der Konfiguration im Simatic®- Manager darzustellen, ist es nötig, auch hier eine zusätzliche Baugruppe dafür zu verwenden.

Diese Baugruppe ist der CP 343-1 (Siemens-Bestellnummer 6GK7 343-1EX30-0XE0 mit Version V2.4).

Die Konfiguration des Profinet in INSEVIS-Projekten ist ganz einfach:

- Auswahl der CPU 315-2 PN/DP (Siemens-Bestellnummer 6ES7 315-2EH14-0AB0) und Platzieren auf Slot 2
- Auswahl des CP 343-1 (Siemens-Bestellnummer 6GK7 343-1EX30-0XE0 mit Version V2.4) und Platzieren auf Slot 4.

Steckplatz	Baugruppe	Bestellnummer	Firmware	MPI-Adresse	E-Adresse	A-Adresse	K...
1							
2	CPU 315-2PN/DP	6ES7 315-2EH14-0AB0	V3.1				
X1	MPI/DP				2047		
X2	PN-IO				2046		
X2 P1	Port 1				2045		
X2 P2	Port 2				2044		
3							
4	CP 343-1	6GK7 343-1EX30-0XE0	V2.4		256...271	256...271	
X1	PN-IO				1023		
X1 P1 R	Port 1				1022		
X1 P2 R	Port 2				1021		
5							
6							

Kommunikation - Profinet IO Controller

Danach können die Profinet-Teilnehmer (-Devices) auf den Profinet-Bus gezogen werden, wie hier im Beispiel gezeigt.

HW Config - [SIMATIC 300(1) (Konfiguration) -- PC710T]

Station Bearbeiten Einfügen Zielsystem Ansicht Extras Fenster Hilfe

Suchen: Profi: Standard

PROFINET IO

- Gateway
- HMI
- I/O
 - ET 200AL
 - ET 200eco PN
 - ET 200M
 - GSD
 - IM153-4 PN HF V3.0
 - IM153-4 PN HF V3.0 She
 - IM153-4 PN HF V4.0
 - IM153-4 PN HF V4.0 She
 - IM153-4 PN ST V3.0
 - IM153-4 PN ST V3.0 She
 - AI-300
 - AI/AO-300
 - AO-300
 - CP-300
 - DI-300
 - DI/DO-300
 - DO-300
 - SM 322 DO16xAI
 - SM 322 DO16xD
 - SM 322 DO16xR
 - SM 322 DO16xU
 - SM 322 DI16xAI

6ES7 322-1BL00-0AA0
Digitalausgabebaugr. DO32 24V/0.5A,
Wurzelung 8; auch als SIPLUS-
Baugruppe mit Bestellnummer 6AG1

Drücken Sie F1, um Hilfe zu erhalten.

Obwohl die INSEVIS-Profinet-Schnittstelle wie eine CP343-1 konfiguriert wird, verhält sich das Handling bei der S7-Programmierung wie bei dem des Onboard-Profinets auf der S7-315-2 PN/DP.

Das heißt im besonderen:

- Es ist kein Prozessdatenaustausch zwischen der CPU und dem CP nötig
- Die Alarm-OBs werden von der CPU selbstständig aufgerufen

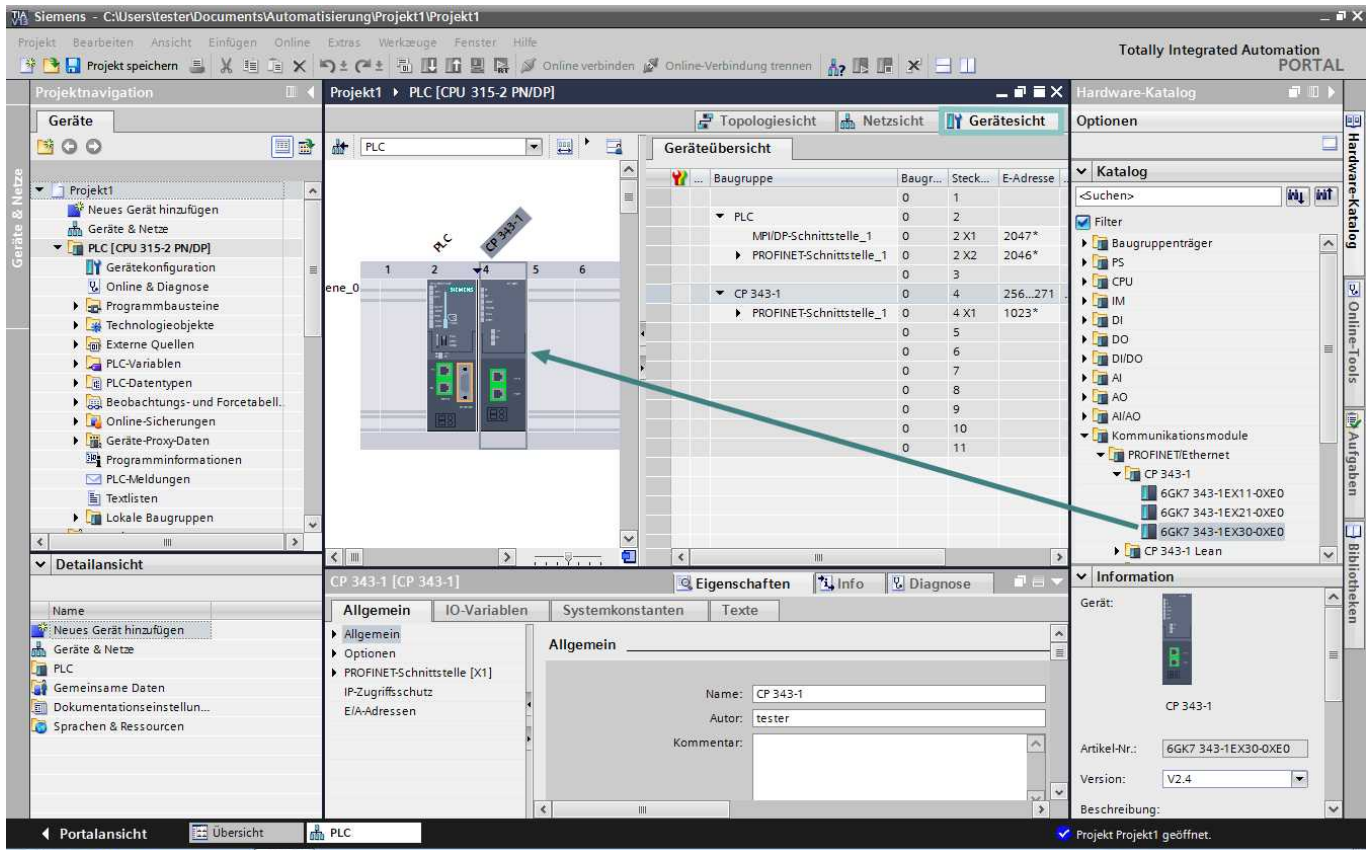
Kommunikation - Profinet IO Controller

Konfiguration im TIA-Portal

Profinet ist bei INSEVIS als Option zusätzlich zur CPU verfügbar. Um diese Option auch in der Konfiguration im TIA®-Portal darzustellen, ist es auch nötig, auch hier eine zusätzliche Baugruppe dafür zu verwenden. Diese Baugruppe ist der CP 343-1 (Siemens-Bestellnummer 6GK7 343-1EX30-0XE0 mit Version V2.4).

Die Konfiguration des Profinet in INSEVIS-Projekten in der **Gerätesicht** ist ganz einfach:

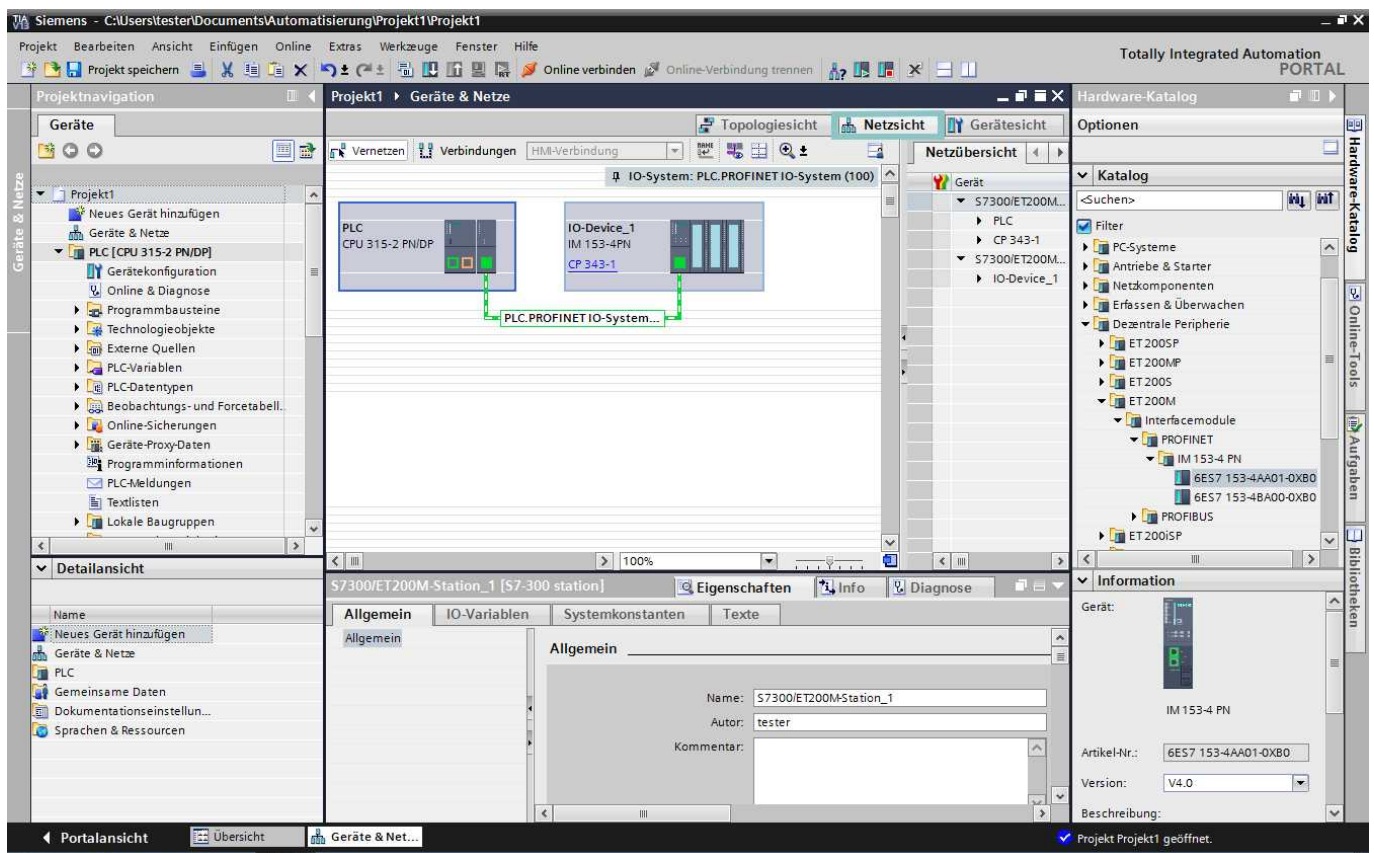
- Auswahl der CPU 315-2 PN/DP (Siemens-Bestellnummer 6ES7 315-2EH14-0AB0) und Platzieren auf Slot 2
- Auswahl des CP 343-1 (Siemens-Bestellnummer 6GK7 343-1EX30-0XE0 mit Version V2.4) und Platzieren auf Slot 4.



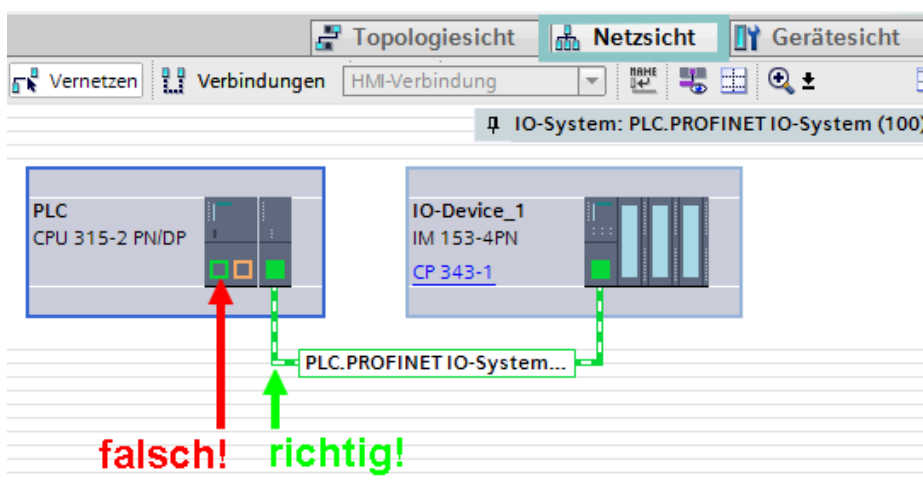
- Auswahl des CP 343-1 (Siemens-Bestellnummer 6GK7 343-1EX30-0XE0 mit Version V2.4) und Platzieren auf Slot 4.

Kommunikation - Profinet IO Controller

Danach können in der Netzsicht die Profinet-Teilnehmer (-Devices) per PLC.PROFINET IO-System verbunden werden, wie hier im Beispiel gezeigt.



Bitte die richtige Verbindung einrichten!



Obwohl die INSEVIS-Profinet-Schnittstelle wie eine CP343-1 konfiguriert wird, verhält sich das Handling bei der S7-Programmierung wie bei dem des Onboard-Profinets auf der S7-315-2 PN/DP.

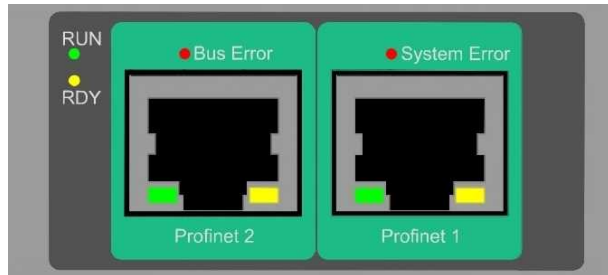
Das heißt im besonderen:

- Es ist kein Prozessdatenaustausch zwischen der CPU und dem CP nötig
- Die Alarm-OBs werden von der CPU selbstständig aufgerufen

Kommunikation - Profinet IO Controller

Status-LEDs

Unten angeführt die Bedeutung der Status-LEDs bei Betrieb eines Profinet IO-Controllers:



LED	Farbe	LED-Status	Bedeutung
RUN	grün	dauerhaft an	Und RDY (gelb) aus: Profinetsystem läuft
		aus	Und RDY (gelb) aus: Profinet-Systemfehler: Spannungsausfall / Defekt
		blinkend	Profinet-Systemfehler: Boot Error
		Und RDY (gelb) an:	Profinet-Systemfehler: Boot Loader Waiting
RDY	gelb	blinkend	Und RUN (grün) aus: Profinet-Systemfehler: Boot Error
		dauerhaft an	Und RUN (grün) aus: Profinet-Systemfehler: Boot Loader Waiting
		aus	Und RUN (grün) aus: Profinet-Systemfehler: Spannungsausfall / Defekt
Bus Error	rot	blinkend	Profinet-Konfigurationsfehler (z.B. nicht alle IO-Devices verbunden)
		dauerhaft an	Profinet-Verbindungsfehler No Link (z. B. Kein Kabel eingesteckt)
		aus	Keine Busfehler
System Error	rot	blinkend	Profinet-Systemfehler (z. B. Ungültige Konfiguration, Watchdog)
		dauerhaft an	Profinet Diagnosealarm berichtet (von mindestens einem IO-Device)
		aus	Keine Busfehler
Bus Error + System Error	rot + rot	dauerhaft an	Keine oder fehlerhafte Profinet-Masterlizenz im Gerät

Kommunikation - Profibus DP-Master

Profibus DP

Die INSEVIS- Steuerungen mit CPU-V und -P können optional mit Profibus ausgerüstet werden. Dabei wird nur die Version DP-V0 unterstützt.



ACHTUNG:

Die Konfiguration der CPU kann entweder im Simatic®- Manager oder in der ConfigStage vorgenommen werden. Beim Download überschreibt der Simatic®-Manager alle anderen SDBs mit. Bitte **ZUERST** Profibus und evtl. CPU im Simatic®- Manager konfigurieren und in die SPS downloaden, dann Ethernet, RS232/485, Modbus, CAN und (de)-zentrale Peripherie und evtl. alternativ die CPU mit der ConfigStage konfigurieren und **DANACH** downloaden. Dieser Download überschreibt nur die betreffenden SDBs und erhält die Profibus-Konfiguration.

Beschaltung

INSEVIS

Beschreibung, andere Bezeichnungen

VP	5V-Versorgung für Repeater (bei Bedarf anschließen)
Ctrl	Steuersignal für Repeater (bei Bedarf anschließen)
P	Signal RxD/TxD-P; PROFIBUS Datenleitung-B; Siemens-Farbcodierung: rot
N	Signal RxD/TxD-N; PROFIBUS Datenleitung-A; Siemens-Farbcodierung: grün
GND	potentialfreies Profibus-Bezugspotential

Profibus DP-Master

Der von INSEVIS als Option angebotene Profibus DPV0 Master wird mit dem Simatic®-Manager konfiguriert.

Konfiguration des DP-Master

Der Simatic®-Hardwarekonfigurator wird für die Konfiguration des DP-Master genutzt. Diese Konfiguration entspricht der der integrierten DP Schnittstelle der CPU315-2PN/DP als DP-Master.

Besonderheiten

Die einzigen Abweichungen zur S7-315-2PN/DP- Zielsteuerung bei der Verwendung des Profibusses sind folgende:

- Der Profibus DP-Master startet die Parametrierung und Konfiguration der Profibus-DP-Slaves erst bei Neustart nach der Übertragung der Konfiguration oder PowerOn.
- Die Diagnoseadresse des Profibus DP-Masters ist fest auf 2047 (W#16#7FF) eingestellt.
- Die Besonderheiten bei OB 86 und SFC 11 sind im Abschnitt „Systemfunktionen“ beschrieben.



VIDEO-Tutorial verfügbar

Für dieses Beispiel ist auf den INSEVIS-Download-Seiten ein Link auf ein entsprechendes Video bei YouTube® verfügbar für Simatic®-Manager.



Hinweis:

- Bei dem INSEVIS Profibus-Anschaltungen steht Ihnen die „Hardware-Online-Diagnose“ aus dem SimaticManager nicht zur Verfügung, die konfigurierten Slaves werden ausgegraut angezeigt.

Kommunikation - Profibus DP Slave

Profibus DP-Slave

Der von INSEVIS als Option angebotene Profibus DP V0 Slave wird mit dem Siemens-Simatic®-Manager konfiguriert.

Funktionen: max. 244 Byte zyklischer Eingangsdaten
 max. 244 Byte zyklischer Ausgangsdaten
 max. 24 konfigurierbare Steckplätze
 GSD-(Gerätestammdaten)-Datei "HIL_0C0F.GSD" für DP-Master Konfiguration
 DP-V0 Funktionsumfang

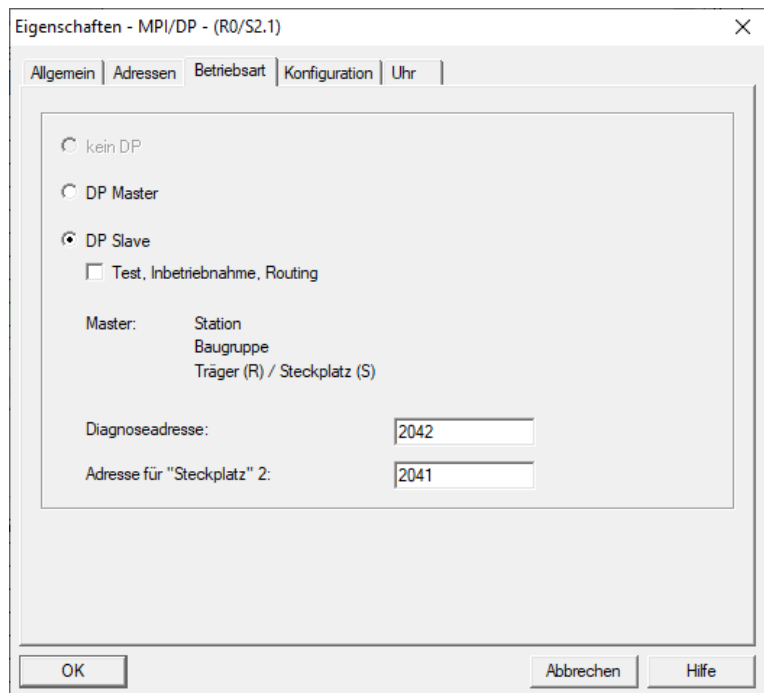
Unterschiede im Vergleich zur CPU315-2PN/DP:

- Kein aktiver Service (Testing, Routing)
- Kein direkter Datenaustausch(DX)-Service

Konfiguration des DP-Slaves

Der Simatic®-Hardware-Konfigurator wird für die Konfiguration des DP-Slaves genutzt. Diese Konfiguration entspricht der der integrierten DP- Schnittstelle der CPU315-2PN/DP als DP-Slave.

- unter „Eigenschaften“
 → Reiter „Allgemein“ den integrierten MPI/DP -Schnittstellentyp als **PROFIBUS** setzen
- unter „Eigenschaften Profibus Schnittstelle“ (wird automatisch geöffnet, wenn nicht, auf „Eigenschaften“ drücken)
 → Reiter „Parameter“:
 - Setzen der DP-Slave-Adresse (Hinweis: Adresse 2 ist meistens für das PG reserviert!)
 - Subnetz nicht vernetzen (Hinweis: sonst Adresskonflikte möglich!)
- mit „OK“ bestätigen
- unter „Eigenschaften“
 → Reiter „Betriebsart“ Setzen der Schnittstelle als „DP-Slave“



Hinweise:

- Wählen Sie **nicht** „Test,
- Wählen Sie „nicht vernetzt“ bei Subnetz an, da ein Subnetz zu Adresskonflikten führen kann!

- Über „Diagnoseadresse“ diagnostiziert der DP-Slave den Ausfall oder die Wiederkehr vom DP-Master. Im Falle eines Ausfalles/ Wiederkehr des DP-Masters, (oder Stecker gezogen / gesteckt), ruft das Betriebssystem den OB86 als Stationsausfall (Ereignis W#16#39C4) / Wiederkehr (Ereignis W#16#38C4). Wenn der OB86 nicht geladen ist, geht die SPS in den STOP-Zustand.
- Wenn kein DP-Slave-Modul bestückt ist, aber eine DP-Schnittstelle mit dem Hardwarekonfigurator konfiguriert wurde, erzeugt das Betriebssystem im Diagnosepuffer keine Einträge.
- Wenn ein DP-Slave-Modul bestückt ist, aber die DP-Schnittstelle als DP-Master im Hardwarekonfigurator konfiguriert wurde, erzeugt das Betriebssystem im Diagnosepuffer folgende Einträge:
 Ereignis: 16# BF02
 OB: 16# 00
 PK: 16# FF
 DatID 1/2: 16# 00 01
 Zusatzinfo 1 / 2 / 3: 16# 0000 0000 0000
- Wenn sich der CPU-Zustand des DP-Slaves ändert (RUN ↔ STOP), sendet der DP-Slave ein Diagnosediagramm zum DP-Master (Sie Struktur wird anschließend beschrieben)



VIDEO-Tutorial verfügbar

Für dieses Beispiel ist auf den INSEVIS-Download-Seiten ein Link auf ein entsprechendes Video bei YouTube® verfügbar für Simatic®-Manager.

Kommunikation - Profibus DP-Slave

Konfiguration der Ein- und Ausgangssteckplätze des DP-Slave

Der DP-Master greift nicht auf die angeschlossenen Ein-/Ausgänge der INSEVIS-SPS mit DP-Slave zu, sondern auf Adressbereiche (Steckplätze) der INSEVIS-SPS. Das Anwenderprogramm der INSEVIS-SPS muss den Datenaustausch zwischen den Adressbereichen und den Ein-/Ausgängen berücksichtigen

Hinweis:

- Der konfigurierte Ein-/Ausgangsbereich für den Datenaustausch zwischen Master und Slaves darf nicht von E/A-Modulen belegt werden.

Die virtuellen Ein-/Ausgangs-Steckplätze unter „Eigenschaften“ → Reiter „Konfiguration“ im Hardwarekonfigurator.

HW Konfig - [SIMATIC 300(1) (Konfiguration) -- PC710T]

Station Bearbeiten Einfügen Zielsystem Ansicht Extras Fenster Hilfe

(0) UR

1	
2	CPU 315-2PN/DP
X1	MPI/DP
X2	PN-IQ
X2 P1	Port 1
X2 P2	Port 2
3	
4	
5	
6	
7	

Eigenschaften - MPI/DP - (R0/S2.1)

Allgemein Adressen Betriebsart Konfiguration Uhr

Zeile	Mode	Partner-DP-Adr	Partner-Adr	lokale Adr	Länge	Konsistenz
1	MS	--	--	E 12	1 Wort	Einheit
2	MS	--	--	A 12	1 Wort	Einheit

Neu ... Bearbeiten ... Löschen

MS Master-Slave-Konfiguration

Master:
Station:
Kommentar:

OK Abbrechen Hilfe

Drücken Sie F1, um Hilfe zu erhalten.



Hinweise:

- Maximal 21 virtuelle Steckplätze (Reihen) können konfiguriert werden.
- Der Modus "Direkter Datenaustausch" (DX) wird nicht unterstützt, nur Master- Slave- Konfiguration (MS).
- Bei Ausfall des DP-Masters (oder Abziehen des Kabels) und direktem Peripheriezugriff (z.B. L PEW 2) auf die konfigurierten Slots wird der Peripheriezugriffsfehler OB122 aufgerufen. Wenn der OB122 nicht programmiert ist, geht die SPS in den STOP- Zustand.

Kommunikation - Profibus DP-Slave

DP Master Konfiguration für INSEVIS Profibus DP-Slave

Bitte beachten Sie, dass bei der Konfiguration des DP- Masters die GSD-(Gerätestammdaten)-Datei "HIL_0C0F.GSD" verwendet wird, um die INSEVIS-SPS als DP-Slave zu konfigurieren.

Benutzen Sie nicht die "CONFIGURED STATIONS → CPU 31X" im Hardware Konfigurator.

- Ziehen des "COMX DP/DPS" Module vom "Hardware Katalog" → PROFIBUS DP → "Weitere Feldgeräte" → "Allgemein"
- Ablegen im DP-Master-System

Steckplatz	DP-Kennun...	Bestellnummer / Bezeichnung	E-Adresse	A-Adresse	Kom...
1	0	blank space			
2	0	blank space			
3	0	blank space			
4	160	1 Byte Out		256	
5	224	1 Word Out		257...258	
6	208	1 Word In	256...257		
7	144	1 Byte In	258		
8					
9					

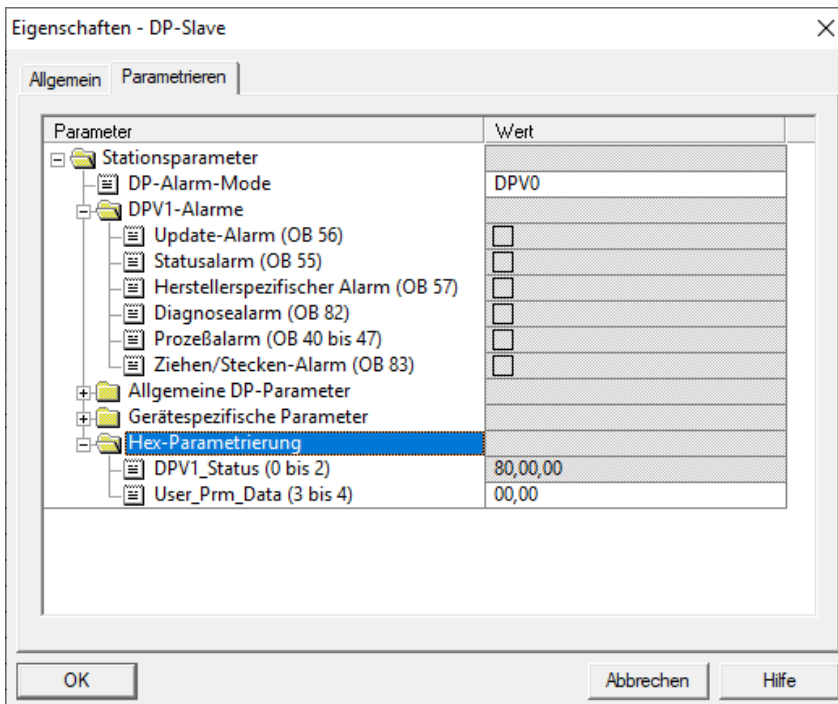


Hinweise:

- Die ersten 3 Steckplätze müssen leer bleiben (Platzhaltermodule)
- Ab dem 4. Steckplatz können die virtuellen Ein-/ Ausgänge konfiguriert werden (Drag 'n Drop, Bild) (Alternativ kann das Universalmodul auf einen freien Slot gezogen und manuell komplett parametrisiert werden, welches einen höheren Aufwand darstellt)
- Maximal 21 Steckplätze können konfiguriert werden (ohne die ersten 3 Platzhaltermodule)

Kommunikation - Profibus DP-Slave

DP-Slave Parameter einstellen in „Eigenschaften“ → Reiter ”Parametrieren“



Hinweise:

- Abwählen aller DP V1 Alarme
- Setzen des "DP Alarm Mode" als DP V0
- Setzen "User_Prm_Data (3 to 4)" als "00, 00"

Struktur der DP-Slave Diagnosetelegramme

DP-Slave-Diagnosetelegramme bestehen aus

- Standard Diagnosedaten (die ersten 6 Byte),
- Moduldiagnose (identifikationsbezogene Diagnose) und
- Modulstatus (gerätebezogene Diagnose)

Byte 0	Stationstatusbyte 1
Byte 1	Stationstatusbyte 2
Byte 2	Stationstatusbyte 3
Byte 3	Master ProfiBus Adresse
Byte 4	Hersteller ID (High byte) 0C
Byte 5	Hersteller ID (Low byte) 0F
Byte 6	Moduldiagnose (identifikationsbezogene Diagnose)
...	
Byte x - 1	(Länge hängt ab von Anzahl konfigurierter Steckplätze)
Byte x	Modulstatus (gerätebezogene Diagnose)
...	
Byte y	(Länge hängt ab von Anzahl konfigurierter Steckplätze))

Kommunikation - Profibus DP-Slave

Stationstatusbyte 1

Bit	Beschreibung	Maßnahme
0	1: Die Station existiert nicht	Überprüfung der Slaveadresse Überprüfung der Busstecker
1	1: Nicht bereit für Datenaustausch	
2	1: Die vom DP-Master gesendeten Konfigurationsdaten sind inkonsistent	Überprüfung der Konfiguration
3	Diagnosealarm 0: Die DP-Slave-CPU ist in RUN 1: Die DP-Slave-CPU ist in STOP	Analyse des Diagnosepuffers der DP-Slave-CPU
4	1: Die Funktion wird nicht unterstützt	Überprüfung der Konfiguration
5	Immer 0	
6	1: Der DP-Slave-Typ ist inkonsistent	Überprüfung der Parametereinstellungen
7	1: Der DP-Slave wurde durch anderem DP-Master konfiguriert.	

Stationstatusbyte 2

Bit	Beschreibung
0	1: Der DP-Slave fordert neue Parameter und Konfiguration an
1	1: Eine Diagnosenachricht wurde erhalten. Der DP-Slave die Operation nicht fortsetzen, bis der Fehler beseitigt ist (statische Diagnosenachricht).
2	Immer 1
3	1: Der Watchdog Monitor im DP-Slave ist aktiviert
4	1: Der DP-Slave hat den Steuerbefehl "FREEZE" erhalten.
5	1: Der DP-Slave hat den Steuerbefehl "SYNC" erhalten.
6	Immer 0
7	1: Der DP-Slave ist deaktiviert, und wurde von der zyklischen Verarbeitung ausgeschlossen.

Stationstatusbyte 3

Bit	Beschreibung
0..6	Immer 0
7	1: Die eingehenden Diagnosenachrichten von dem DP-Slave übersteigen die Speicherkapazität des DP-Masters. Der DP-Master kann nicht alle vom DP-Slave gesendeten Diagnosenachrichten in seinen Diagnosepuffer schreiben.

Master PROFIBUS-Adresse

Bit	Beschreibung
0..7	Adresse des DP-Mastes, welcher den DP-Slave konfiguriert hat
	FF: DP-Slave wurde nicht von einem DP-Master konfiguriert

Hersteller ID

Byte 4	Byte 5
0C_H	0F_H

Kommunikation - Profibus DP Slave

Moduldiagnose (identifikationsbezogene Diagnose)

Byte 6	Bit 6..7 = 01 : Code für Moduldiagnose Bit 0..5 = Länge der Moduldiagnose inklusive Byte 6. (abhängig von der Anzahl konfigurierten Steckplätze bis zu 4 Byte)
Byte 7	Bit 0 = Immer 0 Bit 1 = 0 : Die DP-Slave-CPU ist in RUN 1 : Die DP-Slave-CPU ist in STOP Bit 2 = Immer 0 Bit 3 = Eintrag für Steckplatz 4 Bit 4 = Eintrag für Steckplatz 5 Bit 5 = Eintrag für Steckplatz 6 Bit 6 = Eintrag für Steckplatz 7 Bit 7 = Eintrag für Steckplatz 8
Byte 8	Bit 0 .. 7 = Eintrag für Steckplatz 9 to 16
Byte 9	Bit 0 .. 7 = Eintrag für Steckplatz 17 to 24

Modulstatus (gerätebezogene Diagnose)

Der Modulstatus zeigt den Status der konfigurierten Steckplätze an.

Byte x	Bit 6..7 = 00 Code für Modulstatus Bit 0..5 = Länge der Modulstatus inklusive Byte x. (abhängig von der Anzahl konfigurierten Steckplätze bis zu 10 Byte)
Byte x + 1	82_H : Code for Modulstatus
Byte x + 2	Immer 0
Byte x + 3	Immer 0
Byte x + 4	Bit 0..1 = Immer 0 Bit 2..3 = Status der CPU, 00 : OK, 01 : Fehler Bit 4..5 = Immer 0 Bit 6..7 = Status des Steckplatzes 4, 00 : OK, 01 : Fehler
Byte x + 5	Bit 0..1 = Status des Steckplatzes 5 ... Bit 6..7 = Status des Steckplatzes 8
...	...
Byte y	Bit 0..1 = Status des Steckplatzes 21 Bit 2..3 = Status des Steckplatzes 22 Bit 4..5 = Status des Steckplatzes 23 Bit 6..7 = Status des Steckplatzes 24

Kommunikation - Profibus DP Slave

DP-Slave Adressierung mit SFC 75 „SET_ADDR“

Die INSEVIS-SPSen verfügen über einen SFC 75 „SET_ADDR“, mit dem die Profibus DP-Slave-Adresse zur Laufzeit geändert werden kann.

Allgemeine Hinweise:

- Asynchrone Funktion (Ausführung zieht sich über mehrere SFC-Aufrufe hin)
- Kein Update der neu zugewiesenen Adresse in die Systemkonfiguration (in den SDB)
- Durch STOP und NEUSTART wird die neue Adresse beibehalten.
- Mit dem nächsten POWERON wird dem DP-Slave wieder die original konfigurierte DP-Slave-Adresse zugewiesen

Hinweise für den DP-Master:

- DP-Slave mit alter Adresse wird als Stationsausfall diagnostiziert
- DP-Slave mit neuer Adresse wird als Stationswiederkehr diagnostiziert

Hinweise für den DP-Slave-CPU (INSEVIS-SPS):

- DP-Master wird als Ausfall und Wiederkehr mit Diagnoseadresse diagnostiziert (konfiguriert im DP-Slave).
Die DP-Slave-CPU ruft den OB86 auf. Wenn der OB86 nicht programmiert ist, geht die CPU in den STOP-Zustand.

Parameter	Bezeichnung	Datentyp	Beschreibung
REQ	INPUT	BOOL	TRUE = Start der Adressierung
LADDR	INPUT	WORD	DP-Slave Interfaceadresse, fix W#16#7FF
ADDR	INPUT	BYTE	Neue DP-Adresse
RET_VAL	OUTPUT	INT	Fehlercodes 0000 _H kein Fehler 7000 _H Erster Aufruf mit REQ=FALSE, keine Zuweisung 7001 _H Erster Aufruf mit REQ=TRUE, Zuweisung ist in Abarbeitung 7002 _H Zwischenaufruf, Parameter werden ermittelt 8090 _H LADDR-Parameter ungültig, müssen fix sein W#16#7FF 8091 _H ADDR-Parameter ungültig 809b _H Interface ist nicht als DP-Slave konfiguriert 80C3 _H SFC aufgerufen in niedrigerer Prioritätsebene
BUSY	OUTPUT	BOOL	TRUE = Auftrag in Abarbeitung

Kommunikation - Profibus DP Slave

DP-Slave-Adress- und Baudrateneinstellung mit SFC 76 „SET_DPS“

Die INSEVIS-SPSen verfügen über einen SFC 76 „SET_DPS“, mit dem die Profibus DP-Slave- Adresse und Baudrate zur Laufzeit geändert werden kann. (Erweiterung zu SFC75)

Allgemeine Hinweise:

- Asynchrone Funktion (Ausführung zieht sich über mehrere SFC-Aufrufe hin)
- Kein Update der neu zugewiesenen Adresse in die Systemkonfiguration (in den SDB)
- Durch STOP und NEUSTART werden die neue Adresse und Baudrate beibehalten.
- Mit dem nächsten POWERON wird dem DP-Slave wieder die original konfigurierte DP-Slave-Adresse und Baudrate zugewiesen

Hinweise für den DP-Master:

- DP-Slave mit alter Adresse wird als Stationsausfall diagnostiziert
- DP-Slave mit neuer Adresse wird als Stationswiederkehr diagnostiziert

Hinweise für den DP-Slave-CPU (INSEVIS-SPS):

- DP-Master wird als Ausfall und Wiederkehr mit Diagnoseadresse diagnostiziert (konfiguriert im DP-Slave).
Die DP-Slave-CPU ruft den OB86 auf. Wenn der OB86 nicht programmiert ist, geht die CPU in den STOP-Zustand.

Parameter	Bezeichnung	Datentyp	Beschreibung
REQ	INPUT	BOOL	TRUE = Start der Adressierung
LADDR	INPUT	WORD	DP-Slave Interfaceadresse, fix W#16#7FF
ADDR	INPUT	BYTE	Neue DP-Adresse
BAUD	INPUT	BYTE	Neue DP Baudraten B#16#00 9.6 kbit/s B#16#01 19.2 kbit/s B#16#02 93.75 kbit/s B#16#03 187.5 kbit/s B#16#04 500 kbit/s B#16#06 1.5 Mbit/s B#16#07 3 Mbit/s B#16#08 6 Mbit/s B#16#09 12 Mbit/s B#16#0a 31.25 kbit/s B#16#0b 45.45 kbit/s B#16#0f Auto
RET_VAL	OUTPUT	INT	Fehlercodes 0000 _H kein Fehler 7000 _H Erster Aufruf mit REQ=FALSE, keine Zuweisung 7001 _H Erster Aufruf mit REQ=TRUE, Zuweisung ist in Abarbeitung 7002 _H Zwischenaufruf, Parameter werden ermittelt 8090 _H LADDR-Parameter ungültig, müssen fix sein W#16#7FF 8091 _H ADDR-Parameter ungültig 8092 _H BAUD-Parameter ungültig 809b _H Interface ist nicht als DP-Slave konfiguriert 80C3 _H SFC aufgerufen in niedrigerer Prioritätsebene
BUSY	OUTPUT	BOOL	TRUE = Auftrag in Abarbeitung

Kommunikation - Profibus DP-Master

Profibus DP

Die INSEVIS- Steuerungen mit CPU-V und -P können optional mit Profibus ausgerüstet werden. Dabei wird nur die Version DP-V0 unterstützt.



ACHTUNG:

Die Konfiguration der CPU kann entweder im Simatic®- Manager oder in der ConfigStage vorgenommen werden. Beim Download überschreibt der Simatic®-Manager alle anderen SDBs mit. Bitte **ZUERST** Profibus und evtl. CPU im Simatic®- Manager konfigurieren und in die SPS downloaden, dann Ethernet, RS232/485, Modbus, CAN und (de)-zentrale Peripherie und evtl. alternativ die CPU mit der ConfigStage konfigurieren und **DANACH** downloaden. Dieser Download überschreibt nur die betreffenden SDBs und erhält die Profibus-Konfiguration.

Beschaltung

INSEVIS	Beschreibung, andere Bezeichnungen
VP	5V-Versorgung für Repeater (bei Bedarf anschließen)
Ctrl	Steuersignal für Repeater (bei Bedarf anschließen)
P	Signal RxD/TxD-P; PROFIBUS Datenleitung-B; Siemens-Farbcodierung: rot
N	Signal RxD/TxD-N; PROFIBUS Datenleitung-A; Siemens-Farbcodierung: grün
GND	potentialfreies Profibus-Bezugspotential

Profibus DP-Master

Der von INSEVIS als Option angebotene Profibus DPV0 Master wird mit dem Simatic®-Manager konfiguriert.

Konfiguration des DP-Master

Der Simatic®-Hardwarekonfigurator wird für die Konfiguration des DP-Master genutzt. Diese Konfiguration entspricht der der integrierten DP Schnittstelle der CPU315-2PN/DP als DP-Master.

Besonderheiten

Die einzigen Abweichungen zur S7-315-2PN/DP- Zielsteuerung bei der Verwendung des Profibusses sind folgende:

- Der Profibus DP-Master startet die Parametrierung und Konfiguration der Profibus-DP-Slaves erst bei Neustart nach der Übertragung der Konfiguration oder PowerOn.
- Die Diagnoseadresse des Profibus DP-Masters ist fest auf 2047 (W#16#7FF) eingestellt.
- Die Besonderheiten bei OB 86 und SFC 11 sind im Abschnitt „Systemfunktionen“ beschrieben.



VIDEO-Tutorial verfügbar

Für dieses Beispiel ist auf den INSEVIS-Download-Seiten ein Link auf ein entsprechendes Video bei YouTube® verfügbar für Simatic®-Manager.

Kommunikation - Profibus DP Slave

Profibus DP-Slave

Der von INSEVIS als Option angebotene Profibus DP V0 Slave wird mit dem Siemens-Simatic®-Manager konfiguriert.

Funktionen: max. 244 Byte zyklischer Eingangsdaten
 max. 244 Byte zyklischer Ausgangsdaten
 max. 24 konfigurierbare Steckplätze
 GSD-(Gerätestammdaten)-Datei "HIL_0C0F.GSD" für DP-Master Konfiguration
 DP-V0 Funktionsumfang

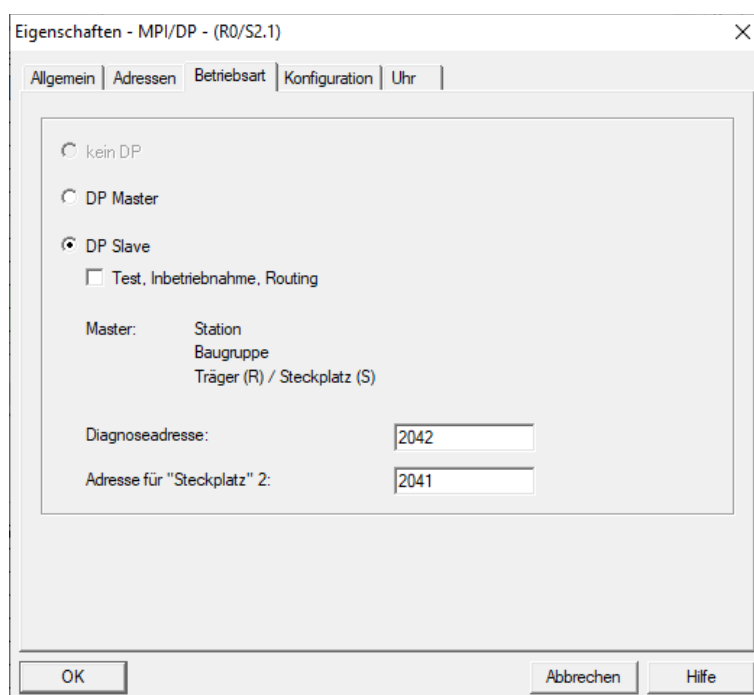
Unterschiede im Vergleich zur CPU315-2PN/DP:

- Kein aktiver Service (Testing, Routing)
- Kein direkter Datenaustausch(DX)-Service

Konfiguration des DP-Slaves

Der Simatic®-Hardware-Konfigurator wird für die Konfiguration des DP-Slaves genutzt. Diese Konfiguration entspricht der der integrierten DP- Schnittstelle der CPU315-2PN/DP als DP-Slave.

- unter „Eigenschaften“
 → Reiter „Allgemein“ den integrierten MPI/DP -Schnittstellentyp als **PROFIBUS** setzen
- unter „Eigenschaften Profibus Schnittstelle“ (wird automatisch geöffnet, wenn nicht, auf „Eigen-schaften“ drücken)
 → Reiter „Parameter“:
 - Setzen der DP-Slave-Adresse (Hinweis: Adresse 2 ist meistens für das PG reserviert!)
- Subnetz nicht vernetzen (Hinweis: sonst Adresskonflikte möglich!)
- mit „OK“ bestätigen
- unter „Eigenschaften“
 → Reiter „Betriebsart“ Setzen der Schnittstelle als „DP-Slave“



Hinweise:

- Wählen Sie nicht „Test,
- Wählen Sie „nicht vernetzt“ bei Subnetz an, da ein Subnetz zu Adresskonflikten führen kann!

- Über „Diagnoseadresse“ diagnostiziert der DP-Slave den Ausfall oder die Wiederkehr vom DP-Master. Im Falle eines Ausfalles/ Wiederkehr des DP-Masters, (oder Stecker gezogen / gesteckt), ruft das Betriebssystem den OB86 als Stationsausfall (Ereignis W#16#39C4) / Wiederkehr (Ereignis W#16#38C4). Wenn der OB86 nicht geladen ist, geht die SPS in den STOP-Zustand.
- Wenn kein DP-Slave-Modul bestückt ist, aber eine DP-Schnittstelle mit dem Hardwarekonfigurator konfiguriert wurde, erzeugt das Betriebssystem im Diagnosepuffer keine Einträge.
- Wenn ein DP-Slave-Modul bestückt ist, aber die DP-Schnittstelle als DP-Master im Hardwarekonfigurator konfiguriert wurde, erzeugt das Betriebssystem im Diagnosepuffer folgende Einträge:

Ereignis:	16# BF02
OB:	16# 00
PK:	16# FF
DatID 1/2:	16# 00 01
Zusatzinfo 1 / 2 / 3:	16# 0000 0000 0000
- Wenn sich der CPU-Zustand des DP-Slaves ändert (RUN ↔ STOP), sendet der DP-Slave ein Diagnosediagramm zum DP-Master (Sie Struktur wird anschließend beschrieben)



VIDEO-Tutorial verfügbar

Für dieses Beispiel ist auf den INSEVIS-Download-Seiten ein Link auf ein entsprechendes Video bei YouTube® verfügbar für Simatic®-Manager.

Kommunikation - Profibus DP-Slave

Konfiguration der Ein- und Ausgangssteckplätze des DP-Slave

Der DP-Master greift nicht auf die angeschlossenen Ein-/Ausgänge der INSEVIS-SPS mit DP-Slave zu, sondern auf Adressbereiche (Steckplätze) der INSEVIS-SPS. Das Anwenderprogramm der INSEVIS-SPS muss den Datenaustausch zwischen den Adressbereichen und den Ein-/Ausgängen berücksichtigen

Hinweis:

- Der konfigurierte Ein-/Ausgangsbereich für den Datenaustausch zwischen Master und Slaves darf nicht von E/A-Modulen belegt werden.

Die virtuellen Ein-/Ausgangs-Steckplätze unter „Eigenschaften“ → Reiter "Konfiguration" im Hardwarekonfigurator.

HW Konfig - [SIMATIC 300(1) (Konfiguration) -- PC710T]

Station Bearbeiten Einfügen Zielsystem Ansicht Extras Fenster Hilfe

(0) UR

1	
2	CPU 315-2PN/DP
X1	MPI/DP
X2	PN-IO
X2 P1	Port 1
X2 P2	Port 2
3	
4	
5	
6	
7	

Eigenschaften - MPI/DP - (R0/S2.1)

Allgemein Adressen Betriebsart Konfiguration Uhr

Zeile	Mode	Partner-DP-Adr	Partner-Adr	lokale Adr	Länge	Konsistenz
1	MS	--	--	E 12	1 Wort	Einheit
2	MS	--	--	A 12	1 Wort	Einheit

Neu ... Bearbeiten ... Löschen

MS Master-Slave-Konfiguration

Master:
Station:
Kommentar:

OK Abbrechen Hilfe

Drücken Sie F1, um Hilfe zu erhalten.



Hinweise:

- Maximal 21 virtuelle Steckplätze (Reihen) können konfiguriert werden.
- Der Modus "Direkter Datenaustausch" (DX) wird nicht unterstützt, nur Master- Slave- Konfiguration (MS).
- Bei Ausfall des DP-Masters (oder Abziehen des Kabels) und direktem Peripheriezugriff (z.B. L PEW 2) auf die konfigurierten Slots wird der Peripheriezugriffsfehler OB122 aufgerufen. Wenn der OB122 nicht programmiert ist, geht die SPS in den STOP- Zustand.

Kommunikation - Profibus DP-Slave

DP Master Konfiguration für INSEVIS Profibus DP-Slave

Bitte beachten Sie, dass bei der Konfiguration des DP- Masters die GSD-(Gerätestammdaten)-Datei "HIL_0C0F.GSD" verwendet wird, um die INSEVIS-SPS als DP-Slave zu konfigurieren.

Benutzen Sie nicht die "CONFIGURED STATIONS → CPU 31X" im Hardware Konfigurator.

- Ziehen des "COMX DP/DPS" Module vom "Hardware Katalog" → PROFIBUS DP → "Weitere Feldgeräte" → "Allgemein"
- Ablegen im DP-Master-System

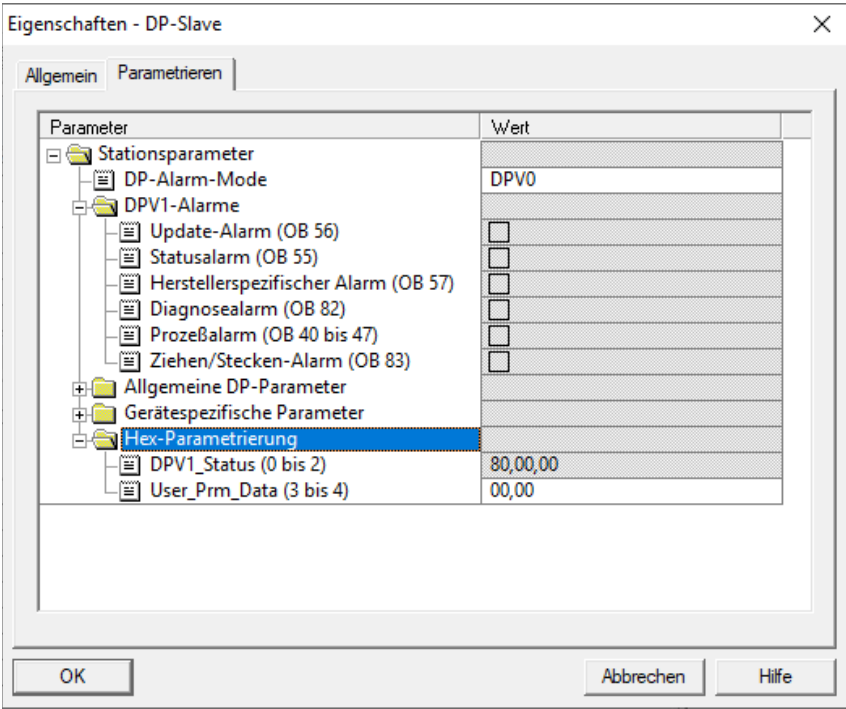
Steckplatz	DP-Kennun...	Bestellnummer / Bezeichnung	E-Adresse	A-Adresse	Kom...
1	0	blank space			
2	0	blank space			
3	0	blank space			
4	160	1 Byte Out		256	
5	224	1 Word Out		257...258	
6	208	1 Word In	256...257		
7	144	1 Byte In	258		
8					
9					



Hinweise:

- Die ersten 3 Steckplätze müssen leer bleiben (Platzhaltermodule)
- Ab dem 4. Steckplatz können die virtuellen Ein-/ Ausgänge konfiguriert werden (Drag 'n Drop, Bild) (Alternativ kann das Universalmodul auf einen freien Slot gezogen und manuell komplett parametrieret werden, welches einen höheren Aufwand darstellt)
- Maximal 21 Steckplätze können konfiguriert werden (ohne die ersten 3 Platzhaltermodule)

DP-Slave Parameter einstellen in „Eigenschaften“ → Reiter "Parametrieren"



- Hinweise:
- Abwählen aller DP V1 Alarme
 - Setzen des "DP Alarm Mode" als DP V0
 - Setzen "User_Prm_Data (3 to 4)" als "00, 00"

Struktur der DP-Slave Diagnosetelegramme

- DP-Slave-Diagnosetelegramme bestehen aus
- Standard Diagnosedaten (die ersten 6 Byte),
 - Moduldiagnose (identifikationsbezogene Diagnose) und
 - Modulstatus (gerätebezogene Diagnose)

Byte 0	Stationstatusbyte 1
Byte 1	Stationstatusbyte 2
Byte 2	Stationstatusbyte 3
Byte 3	Master ProfiBus Adresse
Byte 4	Hersteller ID (High byte) 0C
Byte 5	Hersteller ID (Low byte) 0F
Byte 6	Moduldiagnose (identifikationsbezogene Diagnose)
...	
Byte x - 1	(Länge hängt ab von Anzahl konfigurierter Steckplätze)
Byte x	Modulstatus (gerätebezogene Diagnose)
...	
Byte y	(Länge hängt ab von Anzahl konfigurierter Steckplätze))

Kommunikation - Profibus DP-Slave

Stationstatusbyte 1

Bit	Beschreibung	Maßnahme
0	1: Die Station existiert nicht	Überprüfung der Slaveadresse Überprüfung der Busstecker
1	1: Nicht bereit für Datenaustausch	
2	1: Die vom DP-Master gesendeten Konfigurationsdaten sind inkonsistent	Überprüfung der Konfiguration
3	Diagnosealarm 0: Die DP-Slave-CPU ist in RUN 1: Die DP-Slave-CPU ist in STOP	Analyse des Diagnosepuffers der DP-Slave-CPU
4	1: Die Funktion wird nicht unterstützt	Überprüfung der Konfiguration
5	Immer 0	
6	1: Der DP-Slave-Typ ist inkonsistent	Überprüfung der Parametereinstellungen
7	1: Der DP-Slave wurde durch anderem DP-Master konfiguriert.	

Stationstatusbyte 2

Bit	Beschreibung
0	1: Der DP-Slave fordert neue Parameter und Konfiguration an
1	1: Eine Diagnosenachricht wurde erhalten. Der DP-Slave die Operation nicht fortsetzen, bis der Fehler beseitigt ist (statische Diagnosenachricht).
2	Immer 1
3	1: Der Watchdog Monitor im DP-Slave ist aktiviert
4	1: Der DP-Slave hat den Steuerbefehl "FREEZE" erhalten.
5	1: Der DP-Slave hat den Steuerbefehl "SYNC" erhalten.
6	Immer 0
7	1: Der DP-Slave ist deaktiviert, und wurde von der zyklischen Verarbeitung ausgeschlossen.

Stationstatusbyte 3

Bit	Beschreibung
0..6	Immer 0
7	1: Die eingehenden Diagnosenachrichten von dem DP-Slave übersteigen die Speicherkapazität des DP-Masters. Der DP-Master kann nicht alle vom DP-Slave gesendeten Diagnosenachrichten in seinen Diagnosepuffer schreiben.

Master PROFIBUS-Adresse

Bit	Beschreibung
0..7	Adresse des DP-Mastes, welcher den DP-Slave konfiguriert hat
	FF: DP-Slave wurde nicht von einem DP-Master konfiguriert

Hersteller ID

Byte 4	Byte 5
0C _H	0F _H

Kommunikation - Profibus DP Slave

Moduldiagnose (identifikationsbezogene Diagnose)

Byte 6	Bit 6..7 = 01 : Code für Moduldiagnose Bit 0..5 = Länge der Moduldiagnose inklusive Byte 6. (abhängig von der Anzahl konfigurierten Steckplätze bis zu 4 Byte)
Byte 7	Bit 0 = Immer 0 Bit 1 = 0 : Die DP-Slave-CPU ist in RUN 1 : Die DP-Slave-CPU ist in STOP Bit 2 = Immer 0 Bit 3 = Eintrag für Steckplatz 4 Bit 4 = Eintrag für Steckplatz 5 Bit 5 = Eintrag für Steckplatz 6 Bit 6 = Eintrag für Steckplatz 7 Bit 7 = Eintrag für Steckplatz 8
Byte 8	Bit 0 .. 7 = Eintrag für Steckplatz 9 to 16
Byte 9	Bit 0 .. 7 = Eintrag für Steckplatz 17 to 24

Modulstatus (gerätebezogene Diagnose)

Der Modulstatus zeigt den Status der konfigurierten Steckplätze an.

Byte x	Bit 6..7 = 00 Code für Modulstatus Bit 0..5 = Länge der Modulstatus inklusive Byte x. (abhängig von der Anzahl konfigurierten Steckplätze bis zu 10 Byte)
Byte x + 1	82_H : Code for Modulstatus
Byte x + 2	Immer 0
Byte x + 3	Immer 0
Byte x + 4	Bit 0..1 = Immer 0 Bit 2..3 = Status der CPU, 00 : OK, 01 : Fehler Bit 4..5 = Immer 0 Bit 6..7 = Status des Steckplatzes 4, 00 : OK, 01 : Fehler
Byte x + 5	Bit 0..1 = Status des Steckplatzes 5 ... Bit 6..7 = Status des Steckplatzes 8
...	...
Byte y	Bit 0..1 = Status des Steckplatzes 21 Bit 2..3 = Status des Steckplatzes 22 Bit 4..5 = Status des Steckplatzes 23 Bit 6..7 = Status des Steckplatzes 24

Kommunikation - Profibus DP Slave

DP-Slave Adressierung mit SFC 75 „SET_ADDR“

Die INSEVIS-SPSen verfügen über einen SFC 75 „SET_ADDR“, mit dem die Profibus DP-Slave-Adresse zur Laufzeit geändert werden kann.

Allgemeine Hinweise:

- Asynchrone Funktion (Ausführung zieht sich über mehrere SFC-Aufrufe hin)
- Kein Update der neu zugewiesenen Adresse in die Systemkonfiguration (in den SDB)
- Durch STOP und NEUSTART wird die neue Adresse beibehalten.
- Mit dem nächsten POWERON wird dem DP-Slave wieder die original konfigurierte DP-Slave-Adresse zugewiesen

Hinweise für den DP-Master:

- DP-Slave mit alter Adresse wird als Stationsausfall diagnostiziert
- DP-Slave mit neuer Adresse wird als Stationswiederkehr diagnostiziert

Hinweise für den DP-Slave-CPU (INSEVIS-SPS):

- DP-Master wird als Ausfall und Wiederkehr mit Diagnoseadresse diagnostiziert (konfiguriert im DP-Slave).
Die DP-Slave-CPU ruft den OB86 auf. Wenn der OB86 nicht programmiert ist, geht die CPU in den STOP-Zustand.

Parameter	Bezeichnung	Datentyp	Beschreibung
REQ	INPUT	BOOL	TRUE = Start der Adressierung
LADDR	INPUT	WORD	DP-Slave Interfaceadresse, fix W#16#7FF
ADDR	INPUT	BYTE	Neue DP-Adresse
RET_VAL	OUTPUT	INT	Fehlercodes 0000 _H kein Fehler 7000 _H Erster Aufruf mit REQ=FALSE, keine Zuweisung 7001 _H Erster Aufruf mit REQ=TRUE, Zuweisung ist in Abarbeitung 7002 _H Zwischenaufruf, Parameter werden ermittelt 8090 _H LADDR-Parameter ungültig, müssen fix sein W#16#7FF 8091 _H ADDR-Parameter ungültig 809b _H Interface ist nicht als DP-Slave konfiguriert 80C3 _H SFC aufgerufen in niedrigerer Prioritätsebene
BUSY	OUTPUT	BOOL	TRUE = Auftrag in Abarbeitung

Kommunikation - Profibus DP Slave

DP-Slave-Adress- und Baudrateneinstellung mit SFC 76 „SET_DPS“

Die INSEVIS-SPSen verfügen über einen SFC 76 „SET_DPS“, mit dem die Profibus DP-Slave- Adresse und Baudrate zur Laufzeit geändert werden kann. (Erweiterung zu SFC75)

Allgemeine Hinweise:

- Asynchrone Funktion (Ausführung zieht sich über mehrere SFC-Aufrufe hin)
- Kein Update der neu zugewiesenen Adresse in die Systemkonfiguration (in den SDB)
- Durch STOP und NEUSTART werden die neue Adresse und Baudrate beibehalten.
- Mit dem nächsten POWERON wird dem DP-Slave wieder die original konfigurierte DP-Slave-Adresse und Baudrate zugewiesen

Hinweise für den DP-Master:

- DP-Slave mit alter Adresse wird als Stationsausfall diagnostiziert
- DP-Slave mit neuer Adresse wird als Stationswiederkehr diagnostiziert

Hinweise für den DP-Slave-CPU (INSEVIS-SPS):

- DP-Master wird als Ausfall und Wiederkehr mit Diagnoseadresse diagnostiziert (konfiguriert im DP-Slave). Die DP-Slave-CPU ruft den OB86 auf. Wenn der OB86 nicht programmiert ist, geht die CPU in den STOP-Zustand.

Parameter	Bezeichnung	Datentyp	Beschreibung
REQ	INPUT	BOOL	TRUE = Start der Adressierung
LADDR	INPUT	WORD	DP-Slave Interfaceadresse, fix W#16#7FF
ADDR	INPUT	BYTE	Neue DP-Adresse
BAUD	INPUT	BYTE	Neue DP Baudraten B#16#00 9.6 kbit/s B#16#01 19.2 kbit/s B#16#02 93.75 kbit/s B#16#03 187.5 kbit/s B#16#04 500 kbit/s B#16#06 1.5 Mbit/s B#16#07 3 Mbit/s B#16#08 6 Mbit/s B#16#09 12 Mbit/s B#16#0a 31.25 kbit/s B#16#0b 45.45 kbit/s B#16#0f Auto
RET_VAL	OUTPUT	INT	Fehlercodes 0000 _H kein Fehler 7000 _H Erster Aufruf mit REQ=FALSE, keine Zuweisung 7001 _H Erster Aufruf mit REQ=TRUE, Zuweisung ist in Abarbeitung 7002 _H Zwischenaufruf, Parameter werden ermittelt 8090 _H LADDR-Parameter ungültig, müssen fix sein W#16#7FF 8091 _H ADDR-Parameter ungültig 8092 _H BAUD-Parameter ungültig 809b _H Interface ist nicht als DP-Slave konfiguriert 80C3 _H SFC aufgerufen in niedrigerer Prioritätsebene
BUSY	OUTPUT	BOOL	TRUE = Auftrag in Abarbeitung

Kommunikation - UART

UART

Das INSEVIS UART-Interface ermöglicht sowohl eine frei programmierbare serielle Kommunikation unter STEP®7 sowie eine komfortable Kommunikation im Hintergrund mit Standardprotokollen, konfiguriert mit dem Softwaretool „ConfigStage“. Die allgemeinen Parameter (Baudrate und Datenformat) müssen in jedem Fall mit ConfigStage eingestellt werden. Das Betriebssystem puffert die Sende- und Empfangsdaten, die folgenden Bausteinen dienen nur dem Zugriff auf den Puffer. Senden und Empfangen auf Hardware-Ebene erfolgt unabhängig von S7, z.B. auch im Zustand Stop.



VIDEO-Tutorial verfügbar

Für die Einrichtung serieller Verbindungen mit der ConfigStage stehen auf den INSEVIS-Download-Seiten Beispielprogramme zur Verfügung und ein Link auf ein entsprechendes Video bei YouTube®.

Senden von Daten mit SFB 60 "SEND_PTP", freies ASCII-Protokoll

Mit SFB 60 werden Daten aus einem Datenbaustein versendet.

Die Aktivierung des Sendevorgangs erfolgt nach Aufruf des Bausteins und positiver Flanke am Steuereingang REQ. Der Bereich der zu sendenden Daten wird durch DB-Nummer und Anfangsadresse in SD_1 vorgegeben. Die Typ- und Längeninformationen des ANY- Pointers werden ignoriert.

In LEN wird die Länge des zu sendenden Datenblocks beim Erstauftrag vorgegeben und die Länge der aktuell übertragenen Daten zurückgegeben..

Damit der SFB den Auftrag bearbeiten kann, muss er mit R = FALSE aufgerufen werden.

Bei einer positiven Flanke am Steuereingang R wird ein laufender Sendevorgang abgebrochen und der SFB in den Grundzustand versetzt. Ein abgebrochener Auftrag wird mit einer Fehlermeldung (STATUS-Ausgang) beendet. Mit UART wird die Schnittstelle ausgewählt.

Der Rückgabewert signalisiert, ob die Daten vollständig in den Sendepuffer eingetragen sind oder ob der Aufruf wiederholt werden muss. Es wird entweder DONE auf TRUE gesetzt, wenn der Auftrag ohne Fehler beendet wurde oder ERROR auf TRUE gesetzt, wenn der Auftrag mit Fehler beendet wurde.

Wurde der Auftrag mit DONE=TRUE durchlaufen, bedeutet das, dass die Daten an den Kommunikationspartner gesendet wurden. Nicht sichergestellt ist, ob die Daten auch vom Kommunikationspartner empfangen wurden.

Im STATUS zeigt die CPU bei einem Fehler bzw. bei einer Warnung die entsprechende Ereignisnummer an.

DONE bzw. ERROR/STATUS werden auch bei RESET des SFB (R=TRUE) ausgegeben.

Parameter	Deklaration	Datentyp	Offset	Beschreibung
REQ	INPUT	BOOL	0.0	"Request": Aktiviert den Datentransfer bei positiver Flanke.
R	INPUT	BOOL	0.1	"Reset". Auftrag wird abgebrochen. Senden gesperrt.
LADDR	INPUT	WORD	2	UART-Index 0 = RS232, 1 = RS485
DONE	OUTPUT	BOOL	4.0	FALSE: Inaktiv oder Senden wird noch ausgeführt. TRUE: Auftrag wurde fehlerfrei ausgeführt. Der Parameter ist nur einen Aufruf lang gesetzt.
ERROR	OUTPUT	BOOL	4.1	"Error": Fehlerstatus in STATUS Der Parameter ist nur einen Aufruf lang gesetzt.
STATUS	OUTPUT	WORD	6	0303hex = ANY-Pointer auf Datenbereich ungültig 051Dhex = Reset durch Eingang R
SD_1	INOUT	ANY	8	ANY-Pointer in Datenbereich Sendedaten
LEN	INOUT	INT	18	Länge in Bytes

Die Länge der zu sendenden Daten ist nur durch den parametrierten Datenbereich begrenzt, die Daten werden in Konsistenzblöcken variabler Länge, mit max. 256 Byte übertragen. Der erste Block ist 256 Byte lang.

Empfang von Daten mit SFB 61 "RCV_PTP", freies ASCII-Protokoll

Mit SFB 61 werden empfangene Daten in einem Datenbaustein übertragen.

Der Baustein ist nach Aufruf mit dem Wert TRUE am Eingang EN_R empfangsbereit. Eine laufende Übertragung können Sie mit Signalzustand FALSE am Parameter EN_R abbrechen. Ein abgebrochener Auftrag wird mit einer Fehlermeldung (STATUS-Ausgang) beendet. Der Bereich für die zu empfangenden Daten wird durch DB-Nummer und Anfangsadresse in RD_1 vorgegeben. Die Typ- und Längeninformationen des ANY- Pointers werden ignoriert.

In LEN wird die Länge des zu empfangenden Datenblocks beim Erstaufruf vorgegeben und bei jedem Aufruf die Länge der aktuell übertragenen Daten zurückgegeben. Erstaufruf bedeutet Aufruf entweder nach Reset mit 'R'=TRUE oder nach Erreichen von LEN mit Rückgabe des NDR-Flags mit TRUE.

Damit der SFB den Auftrag bearbeiten kann, muss er mit R(Reset)=FALSE aufgerufen werden. Bei einer positiven Flanke am Steuereingang R, wird die laufende Übertragung abgebrochen und der SFB in den Grundzustand versetzt. Ein abgebrochener Empfangsauftrag wird mit einer Fehlermeldung (STATUS Ausgang) beendet.

Mit UART wird die Schnittstelle ausgewählt.

Es wird entweder NDR auf TRUE gesetzt, wenn der Auftrag ohne Fehler beendet wurde oder ERROR auf TRUE gesetzt, wenn der Auftrag mit Fehler beendet wurde.

Im STATUS zeigt die CPU bei einem Fehler bzw. bei einer Warnung die entsprechende Ereignisnummer an.

NDR bzw. ERROR/STATUS werden auch bei RESET des SFB (R=TRUE) ausgegeben

Funktionsweise:

Der Empfangspuffer im Betriebssystem ist ein Ringpuffer und hat eine Größe von 2 kByte und wird in ganzer Länge für Empfangsdaten genutzt. Der SFB61 holt die Daten dort ab und kopiert linear in den S7-Datenbereich um. Die Parameter LEN und NDR setzen voraus, dass der Anwender eine vorgegebene Anzahl (LEN) Bytes empfängt. Mit dem letzten Byte (Byte[LEN-1]) wird NDR gesetzt und der nächste Auftrag wieder ab Position [0] nach RD_1 geschrieben. Der Parameter EN_R ist statisch (im Gegensatz zum Senden nicht flankengetriggert) und enabled/disabled nur die Funktion des SFB, nicht des Datenempfangs und löscht auch keine Puffer. Der Parameter R setzt nur die internen Zustand des SFB zurück und hat ebenfalls keinen Einfluss auf den Puffer. D.h. wenn unbekannte Daten im Puffer zu erwarten sind und diese stören, sollte der Empfangspuffer durch Lesezugriffe geleert werden, bis LEN als 0 zurückgegeben wird.

Beispiel:

Wenn die Anzahl zu empfangender Daten unbekannt ist (z.B. bei CR/LF-gesteuerten ASCII-Protokollen), sollte LEN auf die maximale Telegrammlänge gesetzt werden. Der SFB61 kann "zwischendurch" den aktuellen Status ermitteln und gibt in LEN die Anzahl bereits empfangener Bytes in RD_1 an.

Jetzt kann das Anwenderprogramm den Datenbereich RD_1 ggf. auswerten und bei erkanntem Telegrammende mit einem Aufruf mit Parameter "R"=TRUE den Auftrag für beendet erklären.

Alternativ können mit LEN=1 die Bytes einzeln aus dem Puffer gelesen werden

Hinweis: LEN darf nicht größer vorgeben werden als der reservierte Datenbereich in RD_1.

Parameter	Deklaration	Datentyp	Offset	Beschreibung
EN_R	INPUT	BOOL	0.0	"Enable receive": Empfangsfreigabe
R	INPUT	BOOL	0.1	"Reset": Auftrag wird abgebrochen
LADDR	INPUT	WORD	2	UART-Index 0 = RS232, 1 = RS485
NDR	OUTPUT	BOOL	4.0	TRUE = Auftrag fertig ohne Fehler, Daten übernommen FALSE = Auftrag wurde nicht gestartet oder läuft noch.
ERROR	OUTPUT	BOOL	4.1	"Error": Fehlerstatus in STATUS Der Parameter ist nur einen Aufruf lang gesetzt.
STATUS	OUTPUT	WORD	6	0303hex = ANY- Pointer auf Datenbereich ungültig 051Dhex = Reset durch Eingang R
RD_1	INOUT	ANY	8	ANY- Pointer in Datenbereich Empfangsdaten
LEN	INOUT	INT	18	Länge in Bytes

Kommunikation - UART

Senden von Daten mit SFB 60 "SEND_PTP", Modbus RTU (Client)

Zur Verwendung des Protokolls Modbus RTU als Client (Master) muss dieses in der Config Stage konfiguriert sein:

- Protokoll RS485 auf "RTU"
- kein Modbus-RTU-Server enabled

Mit SFB 60 wird eine Modbus-RTU PDU aus einem Datenbaustein versendet.

Die Aktivierung des Sendevorgangs erfolgt nach Aufruf des Bausteins und TRUE am Steuereingang REQ.
Der Bereich der zu sendenden Daten wird durch DB-Nummer und Anfangsadresse in SD_1 vorgegeben. Die Typ- und Längeninformationen des ANY- Pointers werden ignoriert.

In LEN wird die Länge des zu sendenden Datenblocks vorgegeben und die Länge der aktuell übertragenen Daten zurückgegeben. (das sollte immer übereinstimmen, da Modbus-Telegramme als ununterbrochener Block übertragen werden müssen!) Die Längenangabe bezieht sich auf die Modbus-Nutzdaten (UID + RTU) ohne Checksumme.
Das Betriebssystem schreibt die Checksumme 2 Byte über die angegebene Länge hinaus, d.h. der Datenbereich muss 2 Bytes länger sein als die angegebene Send-Länge.

Damit der SFB den Auftrag bearbeiten kann, muss er mit R = FALSE aufgerufen werden.

Der Rückgabewert signalisiert, ob die Daten in den Sendepuffer eingetragen werden konnten oder ob der Aufruf wiederholt werden muss. Es wird entweder DONE auf TRUE gesetzt, wenn der Auftrag ohne Fehler beendet wurde oder ERROR auf TRUE gesetzt, wenn der Auftrag mit Fehler beendet wurde.

Wurde der Auftrag mit DONE=TRUE durchlaufen, bedeutet das, dass die Daten an den Kommunikationspartner gesendet wurden. Nicht sichergestellt ist, ob die Daten auch vom Kommunikationspartner empfangen wurden.

Im STATUS zeigt die CPU bei einem Fehler bzw. bei einer Warnung die entsprechende Ereignisnummer an.

DONE bzw. ERROR/STATUS werden auch bei RESET des SFB (R=TRUE) ausgegeben.

Parameter	Deklaration	Datentyp	Offset	Beschreibung
REQ	INPUT	BOOL	0.0	"Request": Aktiviert den Datentransfer bei TRUE
R	INPUT	BOOL	0.1	"Reset". Senden gesperrt. Nicht erforderlich, aus Kompatibilitätsgründen
LADDR	INPUT	WORD	2	nur 1 = RS485
DONE	OUTPUT	BOOL	4.0	FALSE: Inaktiv oder Senden wird noch ausgeführt. (Verletzt die Modbus Timing-Regeln) TRUE: Auftrag wurde fehlerfrei ausgeführt. Der Parameter ist nur einen Aufruf lang gesetzt.
ERROR	OUTPUT	BOOL	4.1	"Error": Fehlerstatus in STATUS Der Parameter ist nur einen Aufruf lang gesetzt.
STATUS	OUTPUT	WORD	6	0303hex = ANY-Pointer auf Datenbereich ungültig 051Dhex = Reset durch Eingang R
SD_1	INOUT	ANY	8	ANY-Pointer in Datenbereich Sendedaten - UID (1 Byte Zieladresse) - PDU: Modbus-Command (1 Byte) ... Modbus-Adresse, Anzahl ... (kommandospezifisch) -CRC (2 Byte Platz im Datenfeld reservieren)
LEN	INOUT	INT	18	Länge der Nutzdaten (ohne Checksumme) in Bytes

Kommunikation - UART

Empfang von Daten mit SFB 61 "RCV_PTP", Modbus RTU (Client)

Zur Verwendung des Protokolls Modbus RTU als Client (Master) muss dieses in der Config Stage konfiguriert sein (siehe SFB60). Mit SFB 61 werden empfangene Daten in einem Datenbaustein übertragen.

Der Baustein ist nach Aufruf mit dem Wert TRUE am Eingang EN_R empfangsbereit. Der Bereich für die zu empfangenden Daten wird durch DB-Nummer und Anfangsadresse in RD_1 vorgegeben. Die Typ- und Längeninformatoren des ANY- Pointers werden ignoriert.

In LEN wird die Länge des Empfangsdatenbereiches vorgegeben und die Länge der aktuell übertragenen Daten zurückgegeben. Die Längenangabe bezieht sich jeweils auf die Modbus-Nutzdaten (UID + RTU) ohne Checksumme. Das Betriebssystem schreibt die Checksumme 2 Byte über die angegebene Länge hinaus, d.h. der reservierte Datenbereich muss 2 Bytes länger sein als die angegebene Länge.

Damit der SFB den Auftrag bearbeiten kann, muss er mit R(Reset)=FALSE aufgerufen werden.

Es wird entweder NDR auf TRUE gesetzt, wenn der Auftrag ohne Fehler beendet wurde oder ERROR auf TRUE gesetzt, wenn der Auftrag mit Fehler beendet wurde.

Im STATUS zeigt die CPU bei einem Fehler bzw. bei einer Warnung die entsprechende Ereignisnummer an.

NDR bzw. ERROR/STATUS werden auch bei RESET des SFB (R=TRUE) ausgegeben

Parameter	Deklaration	Datentyp	Offset	Beschreibung
EN_R	INPUT	BOOL	0.0	"Enable receive": Empfangsfreigabe
R	INPUT	BOOL	0.1	"Reset": Auftrag wird abgebrochen Nicht erforderlich, aus Kompatibilitätsgründen
LADDR	INPUT	WORD	2	nur 1 = RS485
NDR	OUTPUT	BOOL	4.0	TRUE = Auftrag fertig ohne Fehler, Daten übernommen FALSE = Empfang läuft noch.
ERROR	OUTPUT	BOOL	4.1	"Error": Fehlerstatus in STATUS Der Parameter ist nur einen Aufruf lang gesetzt.
STATUS	OUTPUT	WORD	6	0303hex = ANY- Pointer auf Datenbereich ungültig 051Dhex = Reset durch Eingang R 0000hex = ok 0001hex = gesperrt durch anderen Auftrag 0002hex = nichts empfangen 0003hex = < 2 Byte empfangen 0004hex = CRC Error, Daten ignoriert 0005hex = Paritätsfehler 0006hex = Empfang läuft
RD_1	INOUT	ANY	8	ANY- Pointer in Datenbereich Empfangsdaten, - UID (1 Byte Zieladresse) - PDU: Modbus-Command (1 Byte) ... Modbus-Adresse, Anzahl ... (kommandospezifisch) -CRC (2 Byte Platz im Datenfeld reservieren)
LEN	INOUT	INT	18	Länge der Nutzdaten (ohne Checksumme) in Bytes

Es ist nicht erlaubt, SFC60/61 gleichzeitig in mehreren Abarbeitungsebenen (OB1, OB35) zu benutzen.

Kommunikation - UART

Modbus RTU (Server)

Zur Verwendung des Protokolls Modbus RTU als Server (Slave) muss dieser in der ConfigStage konfiguriert sein.

- RS485 Baudrate maximal 19200
- Datenformat 8E1 (=Modbus-Empfehlung)
- Protokoll RS485 auf "RTU"
- Modbus-RTU-Server enable
- Mapping auf (vorhandene!) S7-Operanden definieren

folgende Kommandos werden unterstützt:

- 01 (0x01) Read Coils
- 02 (0x02) Read Discrete Inputs
- 03 (0x03) Read Holding Registers
- 04 (0x04) Read Input Registers
- 05 (0x05) Write Single Coil
- 06 (0x06) Write Single Register
- 15 (0x0F) Write Multiple Coils
- 16 (0x10) Write Multiple registers

In S7 muss nichts programmiert werden. Die Daten werden im Hintergrund kontrollpunktsynchron vom Betriebssystem bearbeitet.



VIDEO-Tutorial verfügbar

Für Modbus RTU-Anwendungen stehen auf den auf den INSEVIS-Download-Seiten Beispielpprogramme zur Verfügung und ein Link auf ein entsprechendes Video bei YouTube.

Empfang über RS232 mit Endeerkennung

Zur Verwendung des Protokolls „Empfang mit Endeerkennung“ muss dieses in der Config Stage konfiguriert sein.

Modus „Nach Ablauf der Zeichenverzugszeit“:

Im SFB61 wird die maximale Länge des zu erwartenden Empfangstelegramms angegeben.

NDR wird gesetzt, wenn die maximale Länge erreicht ist oder seit dem letzten SFB61-Aufruf keine neue Daten empfangen wurden und die konfigurierte Zeichenverzugszeit seit dem letzten Empfang per SFB61-Aufruf überschritten wurde.

Da die Protokollverarbeitung im SFB61 stattfindet, müssen 2 aufeinanderfolgende Telegramme durch eine Sendepause von mindestens der 2-fachen SFB61-Zykluszeit getrennt sein. Anderenfalls verschmelzen beide Telegramme.

Modus „Endezeichen“

Im SFB61 wird die maximale Länge des zu erwartenden Empfangstelegramms angegeben.

NDR wird gesetzt, wenn die maximale Länge erreicht oder das/die konfigurierten Endezeichen erkannt wurden. Die Länge und das Empfangstelegramm werden einschließlich des/der Endezeichen zurückgegeben.

Da die Protokollverarbeitung im SFB61 stattfindet, dürfen nach dem (den) Endezeichen bis zum Aufruf des SFB61 keine weiteren Bytes gesendet werden. Anderenfalls gehen diese Daten verloren.

Bearbeiten der UART-Konfiguration mit SFB 130 "CFG_PTP"

INSEVIS-Steuerungen bieten den SFB 130 "CFG_PTP", um die UART-Schnittstellenparameter der SPS abzufragen oder zu überschreiben. Damit wird die bisher gültige Konfiguration bis zum nächsten Anlauf überschrieben.

Parameter	Deklaration	Datentyp	Offset	Beschreibung
REQ	INPUT	BOOL	0.0	TRUE = Überschreiben der UART-Konfiguration, FALSE = Abfragen der UART-Konfiguration
UART	INPUT	WORD	2	0 = RS232, 1 = RS485
DONE	OUTPUT	BOOL	4.0	TRUE = Erfolgreich konfiguriert FALSE = Konfiguration fehlgeschlagen
ERROR	OUTPUT	BOOL	4.1	TRUE = Konfiguration fehlgeschlagen
STATUS	OUTPUT	WORD	6	Statuscode: 10hex = Index UART ungültig 20hex = Baudrate BAUD ungültig 30hex = Datenformat DATA_FMRT ungültig 40hex = Modbus-Server Unit-Identifizier UID ungültig
BAUD	INOUT	DINT	8	Baudrate in bit/s, z.B. "9600" Wert muss > 0 sein
DATA_FMRT	INOUT	BYTE	12	CPU-V/P: 0 = 8 Datenbit, keine Parität 1 = 8 Datenbit, gerade Parität 2 = 8 Datenbit, ungerade Parität CPU-T (ab BS 2.5.1): bit .3 - .0 0 = keine Parität 1 = gerade Parität 2 = ungerade Parität 3 = Mark-Parität (1) 4 = Space-Parität (0) bit .5 - .4 0 = 1 Stopbit 1 = 1,5 Stopbits (nur bei 5 Datenbits) 2 = 2 Stopbits (außer bei 5 Datenbits) bit .7 - .6 0 = 8 Datenbits 1 = 7 Datenbits 2 = 6 Datenbits 3 = 5 Datenbits
UID	INOUT	BYTE	13	0 = Konfiguration ASCII oder Modbus-RTU Client 1..247 = Modbus-RTU Server Unit-Identifizier (nur RS485)

Es ist nicht erlaubt, SFB130 gleichzeitig in mehreren Abarbeitungsebenen (OB1, OB35) zu benutzen.

Kommunikation - CAN

CAN

Alle INSEVIS-Steuern verfügen onboard über einen CANopen®-konformen Master.

Das INSEVIS CAN-Interface ermöglicht sowohl

- eine komfortable **CANopen®-konforme Konfiguration mit der kostenlosen Software „ConfigStage“**
 - a) von dezentraler INSEVIS-Peripherie (vollkommen ohne CAN-Wissen) und
 - b) von dezentraler Fremd-Peripherie (einfach mit automatisch einlesbarer EDS-Datei).
- als auch eine **freie Programmierung der CAN-Kommunikation unter STEP7**

Als globaler Parameter ist eine **Baudrate** festzulegen. Aus physikalischen Gründen ist beim CAN-Bus die maximale Baudrate von der Leitungslänge begrenzt. Bei Verwendung von Repeatern / Potentialtrennungen ist u.U. eine weitere Reduzierung der max. Buslänge bzw. Baudrate zu beachten.

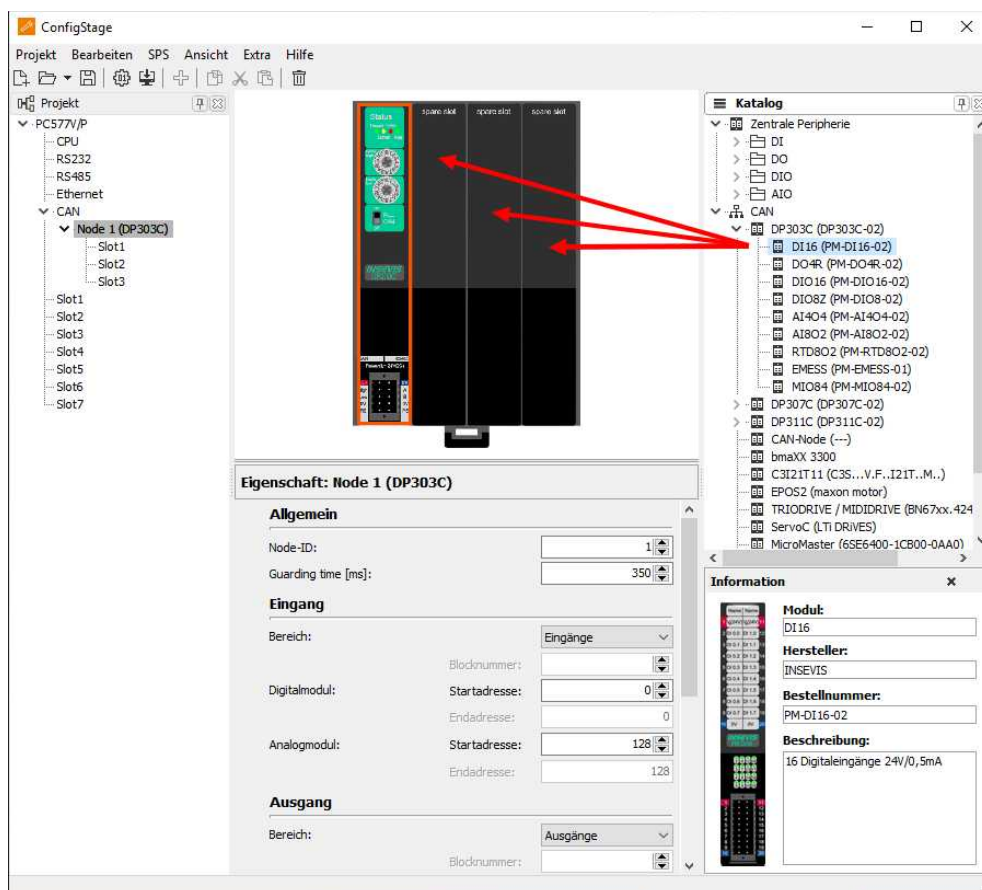
Baudrate (kbit)	maximale Buslänge (m)
1000	20 ... 50
500	100
250	250
125	500
50	1000

CANopen®-Konfiguration mit der ConfigStage

Die Einstellungen in der Konfigurationssoftware ConfigStage sind einfach durchzuführen und → im Kapitel „ConfigStage“ unter dem Punkt „CAN-Einstellungen“ detailliert beschrieben.

a) Bei Verwendung der INSEVIS-Peripherie als dezentrale Peripherie über CANopen® sind keine CAN-spezifischen Kenntnisse nötig. Lediglich folgende Angaben sind erforderlich:

- die Node-ID (Knotennummer) wird eingegeben (die an den hexadezimalen Drehschaltern der CAN-Kopfstation eingestellt wurde)
- die gewünschte Überwachungszeit (Guarding Time) wird festgelegt. (Innerhalb dieser Zeit fragt die SPS den Zustand des CAN-Knotens ab und gibt Fehlermeldungen durch den SFB114 (CAN-Statusabfrage) in dem OB186 aus.)
- Die Startadresse der dezentralen E/A-Adressierung kann nicht einzeln pro Peripheriemodul, sondern nur einmalig pro Kopfstation eingestellt werden. Danach erfolgt der Zugriff unter S7 genau so, wie auf die INSEVIS-onboard-Peripherie.



Kommunikation - CAN

b) Zur **Anbindung externer CANopen®-konformer Slaves** bietet sich die einfache Konfiguration durch

Tx PDO Konfiguration

PDO-Nummer: 1 Beschreibung:

Setting

☐ PDO Disable-Flag ☐ RTR-not-allowed-flag ☐ 29-bit-enable-flag

COB ID

☐ Konstant ☐ "Node ID" + Offset (hex) ID (hex): 182 ☒ gemäß CIA DS301

Kommunikationsparameter

☒ Download

Inhibit time [1/10 ms]: 0

Event time [ms]: 0

Übertragungstyp: 255

S7 Mapping

Bereich: Eingänge

Blocknummer: 0

Byte-Offset: 0

Mapping

☒ Download Datenlänge (Bytes): 3

☒ Zeige CAN-Objekte

Datentyp	Index (hex)	Sub index (hex)	Beschreibung
Word	6041	00	Statusword
Byte	6061	00	Modes of Operation Display

Hinzufügen Übernehmen Reset Löschen

Schließen

CAN-Objekte

Suche CAN-Objekt

☐ Zeige Werte im hexadezimalen Format

Beschreibung	Index	Sub index	Datentyp	Unteres Limit	Vorgegebener Wert	Oberes Limit
Controlword	6040	00	UINT16			65535
Statusword	6041	00	UINT16			65535
VI Target Velocity	6042	00	INT16	-32768		32767
VI Velocity Demand	6043	00	INT16	-32768		32767
VI Control Effort	6044	00	INT16	-32768		32767
VI Nominal Percentage	6052	00	INT16	-32768		32767
VI Percentage Demand	6053	00	INT16	-32768		32767
VI Actual Percentage	6054	00	INT16	-32768		32767
Modes of Operation	6060	00	INT8	-128	2	127
Modes of Operation Display	6061	00	INT8	-128	2	127
Target Torque	6071	00	INT16	-32768		32767
Torque Demand Value	6074	00	INT16	-32768		32767
Torque Actual Value	6077	00	INT16	-32768		32767
Free Objects to Drive.P2050_2	2802	03	INT16	-32768		32767
Free Objects to Drive.P2050_3	2802	04	INT16	-32768		32767
Free Objects to Drive.P2050_5	2802	06	INT16	-32768		32767
Free Objects From Drive.R2051_2	2803	03	INT16	-32768		32767
Free Objects From Drive.R2051_3	2803	04	INT16	-32768		32767

- Einlesen der slavebezogenen EDS-Datei und
- Konfiguration der CAN-Objekte mit der ConfigStage an.

Kommunikation - CAN

Freie CAN-Programmierung (Expertenlevel)

Die freie Programmierung der CAN-Kommunikation unter STEP7 ist **nur empfohlen für Experten mit CANopen Erfahrung oder zur Anbindung von CAN-Slaves, die nicht CANopen-konform sind** und wird in diesem Kapitel nachfolgend beschrieben.

Senden von CAN-Telegrammen SFB105

Mit SFB 105 können beliebige CAN-Telegramme versendet werden. Dabei werden die Telegramme unmittelbar in einen Sendepuffer eingetragen und der weitere Ablauf muss nicht berücksichtigt werden. Der Rückgabewert signalisiert Pufferüberlauf, welcher i.d.R. nur bei physikalischen Busfehlern oder groben Programmfehlern auftritt.

Parameter	Deklaration	Datentyp	Offset	Beschreibung
COB-ID	INPUT	DWORD	0	CAN-Telegramm Identifier (11 oder 29 Bit, rechtsbündig)
RTR	INPUT	BOOL	4.0	„Remote Transmit Request“ fordert das Senden des CAN-Telegrammes des jeweiligen Identifiers an. Nur sinnvoll ohne Daten, d.h. DLC = 0.
IDE	INPUT	BOOL	4.1	Identifier-Format 0: 11 Bit 1: 29 Bit
DLC	INPUT	BYTE	5	„Data Length Code“ Anzahl zu sendender Bytes im Datenfeld (0...8)
DATA	INPUT	ARRAY[1..8] of BYTE	6	Nutzdaten
Status	OUTPUT	WORD	14	Rückgabewert mit Statusbits: Bit 0: Pufferüberlauf z. B. - durch „Fluten“ des Sendepuffers im Programmablauf - aufgrund eines Verdrahtungsfehlers können keine Telegramme versendet werden Bit 1: Parameterfehler COB-ID Bit 2: Parameterfehler DLC

Empfang von CAN-Telegrammen SFB106

Alle empfangenen CAN-Telegramme, die keinem Dienst explizit zugeordnet wurden, werden in einem Anwender-Empfangspuffer zwischengespeichert und können über SFB106 ausgelesen werden. Der Rückgabewert signalisiert Pufferinhalt und Pufferüberlauf.

Parameter	Deklaration	Datentyp	Offset	Beschreibung
COB-ID	OUTPUT	DWORD	0	CAN-Telegramm Identifier (11 oder 29 Bit, rechtsbündig)
RTR	OUTPUT	BOOL	4.0	„Remote Transmit Request“ fordert das Senden des CAN-Telegrammes des jeweiligen Identifiers an. Nur sinnvoll ohne Daten, d.h. DLC = 0.
IDE	OUTPUT	BOOL	4.1	Identifier-Format 0: 11 Bit 1: 29 Bit
DLC	OUTPUT	BYTE	5	„Data Length Code“ Anzahl empfangener Bytes im Datenfeld (0...8)
DATA	OUTPUT	ARRAY[1..8] of BYTE	6	Nutzdaten
Status	OUTPUT	WORD	14	Rückgabewert mit Statusbits: Bit 0 Buffer leer: keine Daten empfangen Bit 1 Bufferüberlauf: Zu viele Telegramme seit dem letzten Aufruf empfangen

Empfangsereignis OB106

Jeder Eintrag in den Anwender-Empfangspuffer wird durch einen OB106-Aufruf signalisiert.

Damit kann eine ereignisgesteuerte Kommunikation realisiert werden bzw. Pufferüberläufe bei langen Zykluszeiten und großem Datenvolumen verhindert werden.

CANopen - Hilfsfunktionen (utilities)

Lesen und Schreiben von CANopen-Variablen (Objekten) per SDO-Transfer SFB107

Auf Variablen CANopen-kompatibler Knoten kann mit SFC107 komfortabel per SDO-Kommunikation zugegriffen werden. Für vereinfachte Parameterdarstellung können als Interface FB107-FB113 aus dem Beispiel benutzt werden.

Zu Beginn eines SDO-Transfers ist das Requestbit zu setzen, bei laufender Kommunikation wird es ignoriert.

Da SDO-Kommunikation immer quittiert wird, ist die Funktion mehrfach aufzurufen, bis die Rückgabebits Done oder Error gesetzt sind. Bei segmentiertem Transfer werden bei jedem Aufruf bis zu 7 Bytes übertragen.

Parameter	Deklaration	Datentyp	Offset	Beschreibung
REQ	INPUT	BOOL	0.0	Request
ULD	INPUT	BOOL	0.1	Upload 0: SDO download, CANopen-Variable schreiben 1: SDO upload, CANopen-Variable lesen
SEG	INPUT	BOOL	0.2	Segmentierter SDO Transfer für Objekte > 4 Byte 0: expedited SDO (1..4 Byte) 1: segmentierter SDO Transfer
NODE	INPUT	BYTE	1	Knotennummer
OBJ	INPUT	WORD	2	CANopen-Objekt-Index
SUB	INPUT	BYTE	4	CANopen-Objekt-Subindex
CNT	INPUT	BYTE	5	Anzahl Bytes der Variable [1,2,4]
TIMEOUT	INPUT	WORD	6	Timeout in ms [1..32767], 0 = unendlich
DONE	OUTPUT	BOOL	8.0	fehlerfrei beendet
ERROR	OUTPUT	BOOL	8.1	mit Fehler beendet
STATE	OUTPUT	BYTE	9	Rückgabewert: 0: OK 1: Parameterfehler 2: kein SDOkanal verfügbar 3: Sendepufferüberlauf 4: SDO-Error Response erhalten 5: Fehler Objektindex/Subindex 6: unerwartete Antwort 7: fehlerhafte Länge bei seg. SDO 8: fehlerhafte Länge bei exp. SDO 9: Timeout
DATA	INOUT	ARRAY[1..6] of BYTE Bereichs- zeiger	10	Expedited SDO: Nutzdaten 1..4 Byte Segmentierte SDO: Bereichszeiger auf Nutzdaten
LEN	INOUT	WORD	16	Anzahl übertragener Bytes (nur für Segmented SDO relevant)
INTERN	STATIC	ARRAY[1..7] of BYTE	18..24	interne Variablen

Nachstehend Beispiele von Anwender-FBs

FB107 SDO upload Variable Byte Request Node:Knotennummer Object Subindex DatenByte Rückgabewerte: Done, Error, ErrorCode	FB108 SDO download Variable Byte Request Node:Knotennummer Object Subindex DatenByte Rückgabewerte: Done, Error, ErrorCode	FB109 SDO upload Variable Wort Request Node:Knotennummer Object Subindex DatenWort Rückgabewerte: Done, Error, ErrorCode
--	--	--

Kommunikation - CAN

FB110 SDO download Variable Wort Request Node:Knotennummer Object Subindex DatenWort Rückgabewerte: Done, Error, ErrorCode	FB111 SDO upload Variable Doppelwort Request Node:Knotennummer Object Subindex DatenDoppelwort Rückgabewerte: Done, Error, ErrorCode	FB112 SDO download Variable Doppelwort Request Node:Knotennummer Object Subindex DatenDoppelwort Rückgabewerte: Done, Error, ErrorCode
FB113 SDO Transfer String Request Upload-ControlFlag (0=download, 1=upload) Node:Knotennummer Object Subindex S7-Bereichszeiger in Daten Länge des Datenfeldes / der zu sendenden Daten / der empfangenen Daten in Byte Rückgabewerte: Done, Error, ErrorCode		

Alternative Funktion des SFB107

Die Konfigurationssoftware „ConfigStage“ hinterlegt für jeden konfigurierten CANopen-Knoten eine Liste an SDO-Downloads, die im Hochlauf der SPS abgearbeitet wird, um die Knoten zu konfigurieren. SDO-Objektindex 0 ist nach CANopen nicht als Variable erlaubt. Wird SFB107 mit SDO-Objektindex 0 aufgerufen, wird die komplette interne Konfigurationsliste des jeweiligen Knotens geschrieben.

Parameter	Deklaration	Datentyp	Offset	Beschreibung
REQ	INPUT	BOOL	0.0	Request
ULD	INPUT	BOOL	0.1	Upload 0: SDO download, CANopen-Variable schreiben
SEG	INPUT	BOOL	0.2	Segmentierter SDO Transfer für Objekte > 4 Byte 0: expedited SDO (1..4 Byte)
NODE	INPUT	BYTE	1	Knotennummer
OBJ	INPUT	WORD	2	CANopen-Objekt-Index (muss hier 0 sein)
SUB	INPUT	BYTE	4	CANopen-Objekt-Subindex (muss hier 0 sein)
CNT	INPUT	BYTE	5	Nicht benutzt
TIMEOUT	INPUT	WORD	6	Timeout in ms [1..32767], 0 = unendlich Timeout gilt für jeden einzelnen Transfer
DONE	OUTPUT	BOOL	8.0	Komplette Liste abgearbeitet
ERROR	OUTPUT	BOOL	8.1	Einzel-Transfer mit Fehler beendet; Wird der SFB107 weiterhin aufgerufen, wird die Abarbeitung der Liste fortgesetzt
STATE	OUTPUT	BYTE	9	Rückgabewert: 0: OK 1: Parameterfehler 2: kein SDO-Kanal verfügbar 3: Sendepufferüberlauf 4: SDO-Error Response erhalten 5: Fehler Objektindex/Subindex 6: unerwartete Antwort 9: Timeout
DATA	INOUT	ARRAY[1..6] of BYTE	10	Intern benutzt, Beobachten der aktuellen Nutzdaten möglich, Rückgabe des SDO-FehlerCode bei Abort
LEN	INOUT	WORD	16	Index des aktuell bearbeiteten Listeneintrags (über alle Knoten) Done: Gesamtanzahl bearbeiteter Einträge (aller Knoten)
INTERN	STATIC	ARRAY[1..7] of BYTE	18..24	Interne Variablen

Kommunikation - CAN

CAN Statusabfrage SFB114

SFB114 liefert in dem Instanzdatenbaustein ein Statuswort sowie eine Liste aktueller Statusinformation von CANopen-kompatiblen Knoten.

Parameter	Deklaration	Datentyp	Offset	Beschreibung
TXOFL	OUTPUT	BOOL	0.0	CAN-Transmitbuffer overflow
RXOFL	OUTPUT	BOOL	0.1	CAN-Receivebuffer overflow
L2BUFOFL	OUTPUT	BOOL	0.2	CAN-Anwender-Empfangspuffer overflow
NMT_ERROR	OUTPUT	BOOL	0.3	CANopen NMT-Fehler (Node-Guarding, Heartbeat)
OP_ERROR	OUTPUT	BOOL	0.4	CANopen Knoten nicht OPERATIONAL
HW_ERROR	OUTPUT	BOOL	0.5	CAN-Hardware Fehler
SDO_ERROR	OUTPUT	BOOL	0.6	CANopen Knoteninitialisierung über SDO fehlerhaft
NODESTATE	OUTPUT	ARRAY [1..128] of BYTE	2	Statuswerte der CANopen-Knotenliste: 00 hex BOOTUP = zuletzt Bootup-Telegramm vom Knoten erhalten 01 hex UNKNOWN = konfiguriert aber keine Rückmeldung vom Knoten 02 hex MISSED = keine Antwort mehr vom Knoten 03 hex UNAVAILABLE = kein Guarding auf Grund von Kommunikationsfehlern 04 hex STOPPED 05 hex OPERATIONAL 7F hex PREOPERATIONAL 80 hex Knoten nicht konfiguriert = unbenutzt

Fehlerereignis OB186

OB186 signalisiert einen Baugruppenausfall konfigurierter CANopen-kompatibler Geräte.

Parameter	Deklaration	Datentyp	Offset	Beschreibung
EVENT	OUTPUT	BYTE	0	39 hex Fehler kommt 38 hex Fehler geht = Knoten antwortet wieder
	OUTPUT	BYTE	1	C4 hex
	OUTPUT	BYTE	2	
	OUTPUT	BYTE	3	
	OUTPUT	DWORD	4	
	OUTPUT	BYTE	8	Knotennummer

Automatische Wiederkehr

Unter CANopen werden die Teilnehmer (Knoten) in einer Vorbereitungsphase (PREOPERATIONAL) konfiguriert und übertragen zur Laufzeit (OPERATIONAL) selbständig die Nutzdaten. Dabei erfolgt eine Überwachung, ob alle Knoten noch kommunizieren (NODE GUARDING oder HEARTBEAT)

Fällt die Kommunikation zu einem CAN-Knoten während des Betriebes zeitweilig aus, befindet sich der Knoten bei Wiederkehr i.A. im Zustand PREOPERATIONAL. Das Fehlerhandling und eine erneute Integration des Knotens liegt in der Verantwortung der Applikation (= des S7-Programmierers).

Ausgefallene Knoten werden über SFB114 erkannt.

Die Knoten-Zustandswerte sind die Rückgabewerte des NODE GUARDING oder HEARTBEAT.

Kommunikation - CAN

Befindet sich ein Knoten im PREOPERATIONAL ist er bereit wieder teilzunehmen. Es ist aber damit zu rechnen, dass dieser Knoten (z.B. durch Spannungsverlust) seine bisherige Konfiguration verloren hat.

Zur Neukonfiguration kann SFB107 (mit OBJ = 0) benutzt werden, das Kommando „goto OPEATIONAL“ wird mit SFB105 gesendet. Danach muss gewartet werden, bis ein neuer Knoten-Zustandswert vorliegt.

```
// call SFB114 (via local instance)
CALL #CAN_State
TXOFL :=
RXOFL :=
L2BUFOFL :=
NMT_ERROR:=
OP_ERROR :=
HW_ERROR :=
SDO_ERROR:=
NODESTATE:=

...
// test all critical nodes for state == PREOPERATIONAL or BOOTUP
U(
  L #CAN_State.NODESTATE[2] // Node 2
  L B#16#7F // 7F = PREOPERATIONAL
<>I
)
U(
  L #CAN_State.NODESTATE[2] // Node 2
  L B#16#00 // 00 = BOOTUP
<>I
)
SPB end // exit or test next node

// call SFB107 with OBJ = 0
CALL #SFB107i
REQ :=TRUE
ULD :=FALSE
SEG :=FALSE
Node :=B#16#2 // current Node
OBJ :=W#16#0
SUB :=B#16#0
CNT :=
TIMEOUT:=W#16#64 // 100ms
DONE :=
ERROR :=
STATE :=
DATA :=
LEN :=
...
U #SFB107i.ERROR // Error handling
...
U #SFB107i.DONE // finished: goto „send NMT command“
...
SPA end

// send NMT command „goto OPERATIONAL“
L B#16#1 // CMD goto operational
T #Data[1]
L B#16#2 // Node 2
T #Data[2]

CALL #CAN_Send
COBID :=DW#16#0
RTR :=FALSE
IDE :=FALSE
DLC :=B#16#2 // 2 Bytes
DATA :=#Data
STATUS:=

// wait > Guarding/Heartbeat-Time before use return value of SFB114 again
```


Konfiguration mit der Software „ConfigStage“

Das Konfigurationstool „ConfigStage“ läuft unter Windows 10 und dient zur Konfiguration bzw. Parametrierung von INSEVIS-spezifischen Eigenschaften wie Kommunikations- und Peripherie-Einstellungen und zum Download in den Hardwarebereich der INSEVIS-SPS. Die „ConfigStage“ Software ist auf den aktuell gängigen Windows Versionen lauffähig.

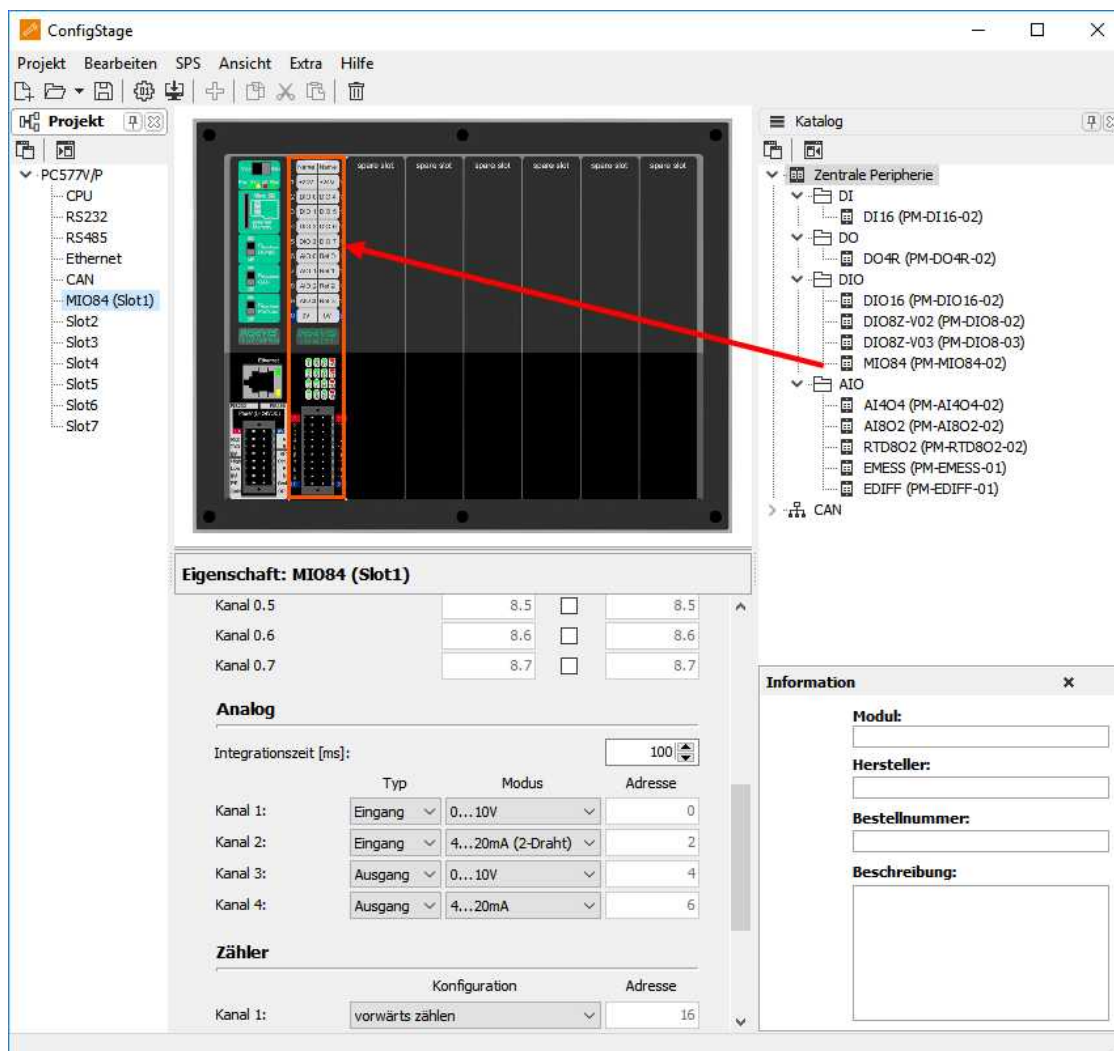


VIDEO-Tutorials verfügbar Sie finden detailliertere Erläuterungen in der ConfigStage-Playlist auf unserem YouTube®-Kanal „INSEVIS DE“. Dokumentierte Beispielprogramme und -einstellungen stehen im Downloadbereich der INSEVIS-Webseiten zur Verfügung.

Grundlegende Einstellungen

Zuerst suchen Sie Ihr Gerät aus der angebotenen Auswahlliste aus. Nach der Geräteauswahl öffnet sich ein Fenster mit mehreren Bereichen.

- Mitte ausgewählte Steuerung noch ohne Peripheriemodule
- Links selbstaufbauender Projektbaum mit den konfigurierbaren Funktionen der CPU, den freien Slots und der konfigurierten Peripherie
- Rechts Katalogbereich mit allen zur Auswahl stehenden Peripheriemodulen und dezentrale CAN-Peripherie
 - Die Peripheriemodule werden per **Drag'n Drop** an den gewünschten Slot gezogen und dort abgesetzt.
 - Die CAN- Kopfstationen und Fremdmodule werden per **Drag'n Drop** auf den grünen Bereich der CPU- bzw. auf die CAN- Pins des CPU-Steckers gezogen und dort abgesetzt.
- Unten Konfigurations- bzw. Parametrierbereich, in dem die gewünschten Eigenschaften festgelegt werden.



Rechts unten Informationsbereich des jeweils markierten Moduls



ACHTUNG Die Konfiguration des Profibus (und der CPU ohne die INSEVIS- spezifischen Einstellungen für Ethernet, seriell und CAN) wird im Simatic®-Manager vorgenommen und in die SPS geladen.

→ **Beim Download überschreibt der Simatic®- Manager alle anderen SDBs mit.**

Bitte **ZUERST** Profibus und evtl. CPU im Simatic®- Manager konfigurieren und in die SPS downloaden, dann die CPU mit Ethernet, RS232/485, Modbus, CAN und (de-)zentrale Peripherie mit der ConfigStage konfigurieren und **DANACH** downloaden. Dieser Download überschreibt nur die betreffenden SDBs und erhält die Profibus- und CPU- Konfiguration des Simatic®- Managers.

Konfiguration mit der Software „ConfigStage“

Änderung der Geräte-IP-Adresse

Entweder

- Klick auf „Ethernet“ im Projektbaum
- Eingeben der IP-Adresse, Netzmaske und Routeradresse (wenn aktiviert)

Oder

- in der Download-Dialogbox (F12)
(Hier wird die Funktion „Erreichbare Teilnehmer“ benutzt, die auch in anderen INSEVIS-Softwaretools zur Verfügung steht.)

Im Beispiel wurde in der Konfiguration dem Gerät in der ConfigStage eine IP-Adresse zugeteilt (192.168.80.50, siehe oben)

Beim Download muss jedoch die Änderung der IP-Adresse noch an die bisherige IP-Adresse 192.168.80.67 geschickt werden, die

- manuell einzugeben ist oder
- aus einer Liste der erreichbaren Geräte („Aktualisieren“ drücken) auszuwählen ist.

Zum Download wird die SPS in gestoppt und läuft dann wieder neu an.

IP-Protokoll

IP-Adresse: 192.168.80.50

Netzmaske: 255.255.255.0

☒ Routeradresse: 192.168.80.1

ConfigStage

IP-Adresse des Zielgerätes
Geben Sie die IP-Adresse des Zielgerätes ein oder wählen Sie das Gerät aus der Liste der erreichbaren Geräte aus.

IP-Adresse des Zielgerätes: 192.168.80.50

Erreichbare Geräte
Schnittstelle: Intel(R) Ethernet Connection I217-LM

Name	MAC-Adresse	IP-Adresse	Netzmaske	Routeradresse
INSEVIS PC57xV	00-50-C2-DF-3E-3D	192.168.80.67	255.255.255.0	192.168.80.67

Aktualisieren

Ok

Abbrechen

Wechsel des Zielgerätes

Soll eine bestehende Konfiguration auf eine andere INSEVIS-S7-SPS übertragen werden, kann unter dem Menüpunkt „Projekte“ dieses ausgewählt werden. Das erspart eine Neuanlage der Konfiguration.

Adressierung der Onboard-Peripherie

Standardadressierung in den INSEVIS- SPSen

Ohne Änderungen durch das Software-Tool „ConfigStage“ wird standardmäßig folgender Adressraum belegt:

Adressbereiche: Peripheriemodule

Digitalmodul: 4 Byte Eingänge, 4 Byte Ausgänge
Analogmodul: 16 Byte Eingänge, 16 Byte Ausgänge

Startadresse \ Slot	Slot 1	Slot 2	Slot 3	Slot n
digitale Eingänge	Byte 0	Byte 4	Byte 8	Byte (n-1)x4
digitale Ausgänge	Byte 0	Byte 4	Byte 8	Byte (n-1)x4
analoge Eingänge	Byte 128	Byte 144	Byte 160	Byte (n-1)x16 +128
analoge Ausgänge	Byte 128	Byte 144	Byte 160	Byte (n-1)x16 +128



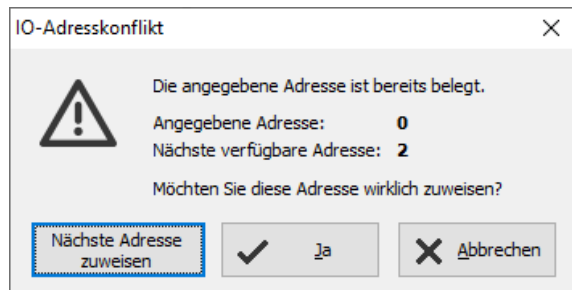
ACHTUNG:

Wird ein FM DIO8-Z eingebaut, so verschieben sich die folgenden digitalen Adressbereiche jeweils um 8 Byte.

Konfiguration mit der Software „ConfigStage“

Abweichende Adressierungen in SPSen und Adressierungen von dezentraler Peripherie

Wird **dezentrale Peripherie** eingesetzt **und/oder** sollen in der **CPU andere Adressbereiche** verwendet werden, so werden diese mit der dem Software-Tool „ConfigStage“ konfiguriert und in Systemdatenbausteinen (SDBs) abgespeichert.



Die verwendeten Adressbereiche können unter Ansicht/ Adressübersicht angezeigt werden. Bei Adressvergabe wird bereits ein Plausibilitätstest durchgeführt und bei Überschneidungen eine Warnung angezeigt. Wird diese Warnung ignoriert, dann werden die Adresskonflikte in dieser Übersicht mit einem roten Ausrufezeichen angezeigt. (Bild rechts)

Typ	Adresse	Modul	Node	Slot
Eingang	0 .. 1	DI16		1
Eingang	0 .. 1	DI16		2
Eingang	8 .. 9	DIO16		3
Ausgang	8 .. 9	DIO16		3
Eingang	12 .. 23	DIO8Z-V03		4
Ausgang	12 .. 23	DIO8Z-V03		4
Ausgang	24 .. 24	DO4R		7
Eingang	24 .. 25	DP303C.DIO16	1	1
Ausgang	25 .. 26	DP303C.DIO16	1	1

CPU-Einstellungen

Die Einstellung der INSEVIS-CPU entspricht der Einstellung der Siemens-CPU S7-315-2PN/DP im Simatic®-Manager. Die CPU-Einstellungen können entweder im Simatic®- Manager oder in der ConfigStage vorgenommen werden.

Allgemein

Angabe von Anlagen- und Ortskennzeichen

Anlauf

Haken gesetzt: CPU läuft auch an, wenn der tatsächliche Ausbau anders als im Projekt eingetragen ist.
Zeitüberlauf: Wert multiplizieren mit 100 Millisekunden, Maximalwert auf 1000 Sekunden beschränkt

Diagnose

Haken gesetzt: Meldet die Ursache für den STOP-Zustand an die angemeldeten Anzeigeräte (PG, Leitsystem,...). Die Meldung wird immer auch in den Diagnosepuffer eingetragen.

Know-how-Schutzmöglichkeiten

Hier wird der Passwortschutz vom Simatic®-Manager (de-) aktiviert und das Passwort eingetragen.

Kommunikation

INSEVIS-CPU's verfügen genau wie die 315-2PN/DP über systeminterne **16 passive S7-Verbindungen**. Zusätzlich können **bis zu 16 aktive S7-Verbindungen** in der ConfigStage parametrisiert und mit einer ID-Nummer versehen werden.
 (Mehr dazu bei **Information zu TSAPs**)

Konfiguration mit der Software „ConfigStage“

Zyklus

Zyklusüberwachungszeit

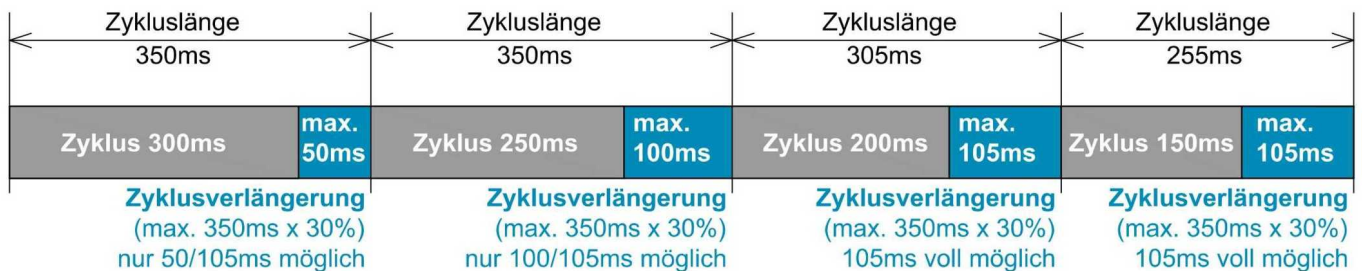
(max. 6000ms = 6 Sekunden):

Zyklusbelastung durch Kommunikation

(Zykluszeitverlängerung) (min. 10% , max. 50%, für Kommunikation (Ethernet, Feldbus, PG), Visualisierung, Filesystem der Micro-SD®-Karte)

Die Angabe der Zykluszeit im PG erfolgt von Kontrollpunkt zu Kontrollpunkt, d.h. einschließlich Kommunikation und Visualisierung. Unabhängig vom S7-Programm kann dadurch eine schwankende Zykluszeit entstehen.

In diesem Feld wird ein Grenzwert im Verhältnis zur Zyklusüberwachungszeit eingestellt, mit dem maximal „die freie Zeit“ bis zur nächsten Zyklusüberwachung für Kommunikations- (und Visualisierungs-, etc.) aufgaben aufgefüllt wird.



Im Beispiel kann die Zykluszeit um maximal 30% von 350ms (=105ms) verlängert werden. Wenn diese Verarbeitungszeit für die Kommunikation (und Visualisierung, etc.) nicht benötigt wird, steht sie der übrigen Verarbeitung zur Verfügung.

Taktmerker

Anwahl Merkerbyte und Angabe von dessen Nummer ab Wert 0

Remanenter Speicher

Merkerbytes

insgesamt 2.048 (davon 0..15 voreingestellt)

Timer und Zähler:

insgesamt 256 (keine voreingestellt)

Uhrzeitalarme

Haken gesetzt: Uhrzeitalarm aktiviert

Eintragen der Intervalle und von Startdatum und -zeit

Weckalarmer

Die **CPUs -V/-P** unterstützen nur den OB35

Angabe in Millisekunden (ms)

Maximalwert 1 Minute (60.000ms)

Die **CPUs-T** unterstützen zusätzlich

zum OB35 auch die OB 32 bis 34

Angabe in Millisekunden (ms)

Maximalwert 1 Minute (60.000ms)

Zyklus

Zyklusüberwachungszeit [ms]:

320

Zyklusbelastung durch Kommunikation [%]:

20

Größe des Prozeßabbilds der Eingänge:

128

Größe des Prozeßabbilds der Ausgänge:

128

Taktmerker

☒ Merkerbyte

64

Remanenter Speicher

Anzahl der Merkerbytes ab MB0:

16

Anzahl der S7-Timer ab T0:

24

Anzahl der S7-Zähler ab C0:

16

Uhrzeitalarme

☒ OB 10 (Priorität 2)

Ausführung:

Minütlich

Startdatum:

01.01.2019

Startzeit:

12:00:00

Weckalarmer

OB35 (Priorität 12)

Ausführung [ms]:

100

OB34 (Priorität 11)

Ausführung [ms]:

200

OB33 (Priorität 10)

Ausführung [ms]:

500

OB32 (Priorität 9)

Ausführung [ms]:

1000

Konfiguration mit der Software „ConfigStage“

WebServer

Die SPSen mit der CPU-T beinhalten einen statischen Webserver, der

- allgemeine Angaben (Snr, etc),
- aktuelle Diagnosepuffereinträge,
- Adressierung von I/Os ausgibt und
- ein Eingabefeld zur Abfrage des aktuellen Wertes von vorhandenen Variablen

zur Verfügung stellt.

Die Aktivierung erfolgt ganz unten bei den CPU-Einstellungen.



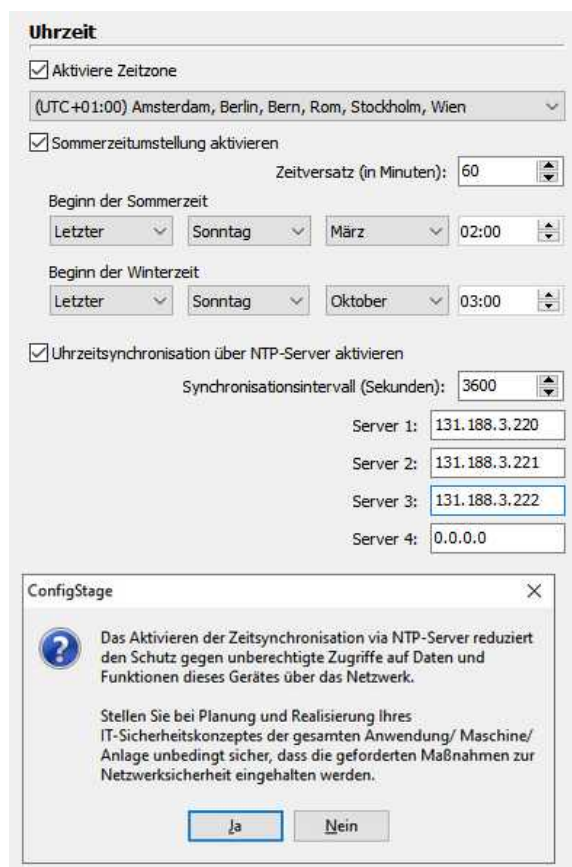
ACHTUNG:

Mit der Verbindung zum Internet ist es möglich, dass sich Unberechtigte den Zugang zu der Steuerung verschaffen und diese manipulieren, dass ein Schaden auftreten kann. Der Anwender und Betreiber sind zuständig für die Planung, Umsetzung und permanente Überwachung der Einhaltung geeigneter Schutzmaßnahmen. Achtung: INSEVIS übernimmt keinerlei Haftungen für die Folgen unberechtigten Zugriffs über diese Schnittstelle.



Uhrzeiteinstellungen (Zeitzone, Sommerzeit, NTP-Server)

Bei Steuerungen mit der CPU-T können ab ConfigStage 1.0.14.41 folgende Einstellungen hinsichtlich der Systemzeit vorgenommen werden:



Auswahl der Zeitzone

(Wichtig für die NTP-Nutzung)

Aktivierung der Umschaltung Sommerzeit/ Winterzeit

Hier werden die jeweiligen Umstellungsdaten eingetragen. Das System stellt sich dann automatisch um. Wenn das Gerät während der Umstellung ausgeschaltet war und der Akku genug Erhaltungsspannung für die Echtzeituhr hatte, passt das Gerät ab der Firmware 2.7.2 die Zeit während des Bootvorgangs selbst an die aktuelle Zeit an. Wenn das Gerät länger ausgeschaltet war, dass die Echtzeituhr spannungsfrei wurde, muss die Zeit manuell eingestellt werden.

Nutzung von NTP-Servern

NTP-Server senden die Koordinierte Weltzeit (UTC), die die Grundlage zur Berechnung der eigenen Zeit darstellt. In der Abbildung links wurden 3 TimeServer der FAU-Universität Erlangen-Nürnberg adressiert (131.188.3.220/1/ 2 entsprechen den ntp0/1/2 der fau.de).

ACHTUNG: Konfigurieren Sie in der ConfigStage unter „Ethernet“ den Router entsprechend den gewählten NTP-Servern (in diesem Fall 192.168.80.1)

ACHTUNG: Mit der Verbindung zum Internet ist es möglich, dass sich Unberechtigte den Zugang zu der Steuerung verschaffen und diese manipulieren, dass ein Schaden auftreten kann. Der Anwender und Betreiber sind zuständig für die Planung, Umsetzung und permanente Überwachung der Einhaltung geeigneter Schutzmaßnahmen. INSEVIS übernimmt keinerlei Haftungen für die Folgen unberechtigten Zugriffs über diese Schnittstelle.

Konfiguration mit der Software „ConfigStage“

Kommunikationseinstellungen

RS232 und RS485

Die Einstellungen für RS232 und RS485 sind selbsterklärend. Wenn bei RS485 das Protokoll Modbus-RTU ausgewählt und "Modbus Server" aktiviert wurde, erfolgt hier die Festlegung Knotennummer sowie das Zuweisen der S7-Operandenbereiche für Eingangsbits und -wörter sowie Ausgangsbits und -wörter. Wenn "Modbus Server" deaktiviert ist, werden über SFB 60/61 Modbus-RTU-Telegramme versendet/ empfangen.

Ethernet

Soll die Ethernet-Schnittstelle benutzt werden, sind bis zu 16 Verbindungen anzulegen, das Protokoll auszuwählen und zu parametrieren. Jede Ethernet-Verbindung erhält eine Verbindungs-ID zur Zuordnung im S7-Programm.

Eigenschaft: Ethernet

IP-Protokoll

IP-Adresse:

Netzmaske:

☐ Routeradresse:

Verbindungen

ID	Typ	Aktiv	Lokal	Partner	Partner IP
1	S7-Verbindung	Ja	10.02	10.02	192.168.80.10

CPU-V/-P:

Es steht EINE Ethernet-Schnittstelle (RJ45) zur Verfügung

CPU-T:

Es stehen ZWEI Ethernet-Schnittstellen (RJ45) zur Verfügung, die entweder
- als Switch mit gemeinsamer IP-Adresse (links) oder

Eigenschaft: Ethernet

Allgemein

Betriebsart
☒ 2-Port-Ethernet-Switch
☐ 2 separate Ethernet Ports

Ethernet 1
☒ Diesen Port zur Nutzung aktivieren
 Übertragung medium / duplex:
☐ Deaktivieren der Auto-Negotiation

Ethernet 2
☒ Diesen Port zur Nutzung aktivieren
 Übertragung medium / duplex:
☐ Deaktivieren der Auto-Negotiation

IP-Protokoll

IP-Adresse:

Netzmaske:

Router:

Schutz

Zugriff über S7-Kommunikation durch entfernten Partner (PG, PLC, HMI, OPC, ...) erlauben
☒ Ethernet 1
☒ Ethernet 2

Verbindungen

ID	Typ	Aktiv	Lokal	Partner
1	S7-Verbindung	Ja	10.02	10.02

Eigenschaft: Ethernet

Allgemein

Betriebsart
☐ 2-Port-Ethernet-Switch
☒ 2 separate Ethernet Ports

Ethernet 1
☒ Diesen Port zur Nutzung aktivieren
 Übertragung medium / duplex:
☐ Deaktivieren der Auto-Negotiation

Ethernet 2
☒ Diesen Port zur Nutzung aktivieren
 Übertragung medium / duplex:
☐ Deaktivieren der Auto-Negotiation

IP-Protokoll

Ethernet 1 Ethernet 2

IP-Adresse:

Netzmaske:

Router:

Schutz

Zugriff über S7-Kommunikation durch entfernten Partner (PG, PLC, HMI, OPC, ...) erlauben
☒ Ethernet 1
☒ Ethernet 2

Verbindungen

ID	Typ	Aktiv	Lokal	Partner
1	S7-Verbindung	Ja	10.02	10.02

- als getrennte Ports mit unterschiedlichen IP-Adressen und Netzmasken (rechts) konfiguriert werden
Es können beide Ports verwendet werden

Bei Port 2 wird die S7-Kommunikation unterbunden
(Schutz gegen unerlaubte Datenübertragung)

Konfiguration mit der Software „ConfigStage“



ACHTUNG:

Um unerlaubte Datenübertragung über die klassische S7-Kommunikation von vornherein zu unterbinden, erlaubt die ConfigStage das "Abschalten" dieser Kommunikation bei einem Port. Dies verhindert unerwünschte Manipulation der SPS-Daten.

Je nach Verbindungstyp wird die ausgewählte Verbindung in einer weiteren Konfigurationsbox parametriert.

Einstellungen bei S7-Verbindung (aktiv)

- Lokaler TSAP,
- Partner-TSAP,
- Partner-IP-Adresse

Einstellungen bei INSEVIS-Panel-HMI

- Lokaler TSAP
(nur nötig, wenn die SPS auf dem HMI Sprache und Seiten umschalten soll)

Einstellungen bei TCP Send/Receive-Verbindung

- Lokaler Port (fest oder variabel),
- Partner-Port,
- Partner-IP-Adresse

Einstellungen bei UDP Send/Receive-Verbindung

- Lokaler Port

Einstellungen bei Modbus (Server) siehe unten

Zuweisung der S7-Operandenbereiche

für bis 16 Modbus-Bereiche

- Eingangsbits
- Eingangswörter
- Ausgangsbits
- Ausgangswörter

Wenn "Modbus Server" deaktiviert ist, werden über SFB 60/61 Modbus-RTU-Telegramme versendet/ empfangen.

Eigenschaft: Ethernet

Allgemein

Betriebsart

☐ 2-Port-Ethernet-Switch

☒ 2 separate Ethernet Ports

Ethernet 1

☒ Diesen Port zur Nutzung aktivieren

Übertragung medium / duplex: Automatik

☐ Deaktivieren der Auto-Negotiation

Ethernet 2

☒ Diesen Port zur Nutzung aktivieren

Übertragung medium / duplex: Automatik

☐ Deaktivieren der Auto-Negotiation

IP-Protokoll

	Ethernet 1	Ethernet 2
IP-Adresse:	192.168.80.50	
Netzmaske:	255.255.255.0	
Router:		

Schutz

Zugriff über S7-Kommunikation durch entfernten Partner (PG, PLC, HMI, OPC, ...) erlauben

☒ Ethernet 1

☒ Ethernet 2

Verbindungen

ID	Typ	Aktiv	Lokal	Partner	Partner IP
1	S7-Verbindung	Ja	10.02	02.02	192.168.80.60
2	TCP Send/R...	Nein	2000	0	0.0.0.0
3	UDP Send/R...		2000		
4	ModBus TCP ...		502		

Eigenschaften - ModBus TCP Server Verbindung

Schnittstelle: Ethernet 1

Abzustimmender Datenbereich:

	ModBus Bereich	Startadr...	Anzahl	S7 Bereich	Blo...	Byte-Offset	Länge in Bytes	Byte-Swap
1	Coils (Ausgangsbits)	0	1	Datenbaustein	1	0	1	
2	Inputregister	200	1	Datenbaustein	1	0	2	Nein
3	Diskrete Eingänge (Bits)	400	1	Datenbaustein	1	0	1	
4	Halte- (Ausgangs) Register	700	1	Datenbaustein	1	0	2	Nein

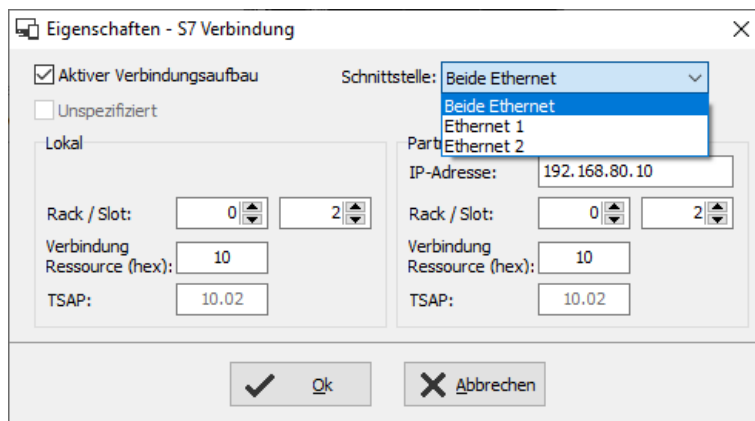
Konfiguration mit der Software „ConfigStage“

Kommunikation / Information zu TSAPs

INSEVIS-CPU's verfügen genau wie die 315-2PN/DP über systeminterne **16 passive S7-Verbindungen**. Deren lokale TSAPs sind nach Siemens-Definition xx.yy wie folgt festgelegt:

- für PG= 01.02,
- für OP= 02.02,
- für WinCC o.ä.= 03.01
- xx=01 (für PG),
- xx=02 (für OP) und
- xx=03 (für WINCC o.ä.)
- yy für alle 02 (= Slot2 für CPU)

CPU-T: Zuweisung der Eigenschaften an beide oder einzelne Ethernet-Schnittstellen



Zusätzlich können bis zu 16 aktive oder passive S7-Verbindungen parametrierbar und mit einer ID-Nummer versehen werden. Hier **gilt** die o.g. Siemens-Definition **nicht** und man sollte nur darauf achten,

1. die TSAPs nicht mehrfach zu vergeben und
2. nicht gerade die TSAPs der passiven Verbindungen zu benutzen.

Profinet

Profinet-Einstellungen werden wie auch die Profibus-Einstellungen im Simatic®-Manager oder im TIA®-Portal vorgenommen.

CAN-Einstellungen

Dezentrale INSEVIS Peripherie

Für die Anbindung von dezentraler INSEVIS- Peripherie an die INSEVIS-S7-CPU's ist **kein CANopen®-Wissen** nötig.

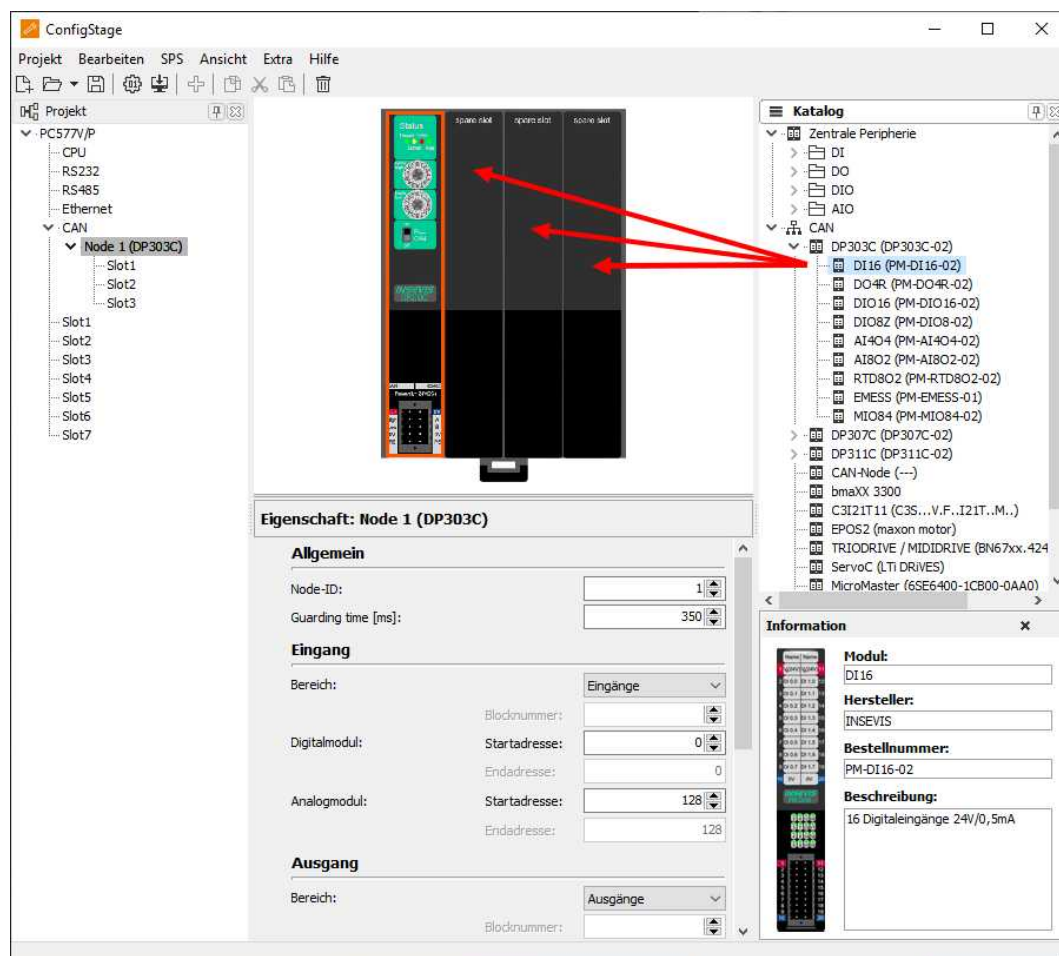
Nachdem Sie in der Startmaske eine CAN-Kopfstation von INSEVIS auf die CPU gezogen haben, erscheint diese im Projektbaum und in der Anzeige.

Jetzt können Sie für die Kopfstation Startadressen für alle Ein-/Ausgänge vergeben (bei den einzelnen Modulen dann nicht mehr).

Die Peripheriemodule werden aus einem extra Bereich des Katalogbaumes (CAN) per Drag'n Drop an den gewünschten Slot gezogen und dort abgesetzt.

Die direkt an der INSEVIS- Kopfstation an den hexadezimalen Drehschaltern eingestellte Knotennummer wird hier eingetragen und eine Überwachungszeit eingestellt.

Danach werden die E/As unter Step®7 angesprochen, wie Onboard-E/As bei einer INSEVIS-SPS.



Konfiguration mit der Software „ConfigStage“

Dezentrale Fremdperipherie manuell konfigurieren

Statt einer INSEVIS- Kopfstation ziehen Sie einen neutralen CAN- Knoten auf die CPU. Dann werden die allgemeinen Einstellungen zu **Knotennummer**, **Überwachungsmechanismen** und Netzwerkmanagement (**NMT**) vorgenommen.

Ist das Auswahlfeld **"NMT Control"** aktiviert, wird der Knoten gemeinsam mit der SPS gestartet und gestoppt. Dazu werden die NMT-Nachrichten "goto OPERATIONAL" beim Übergang in RUN (nach OB100, vor erstem OB1) und „goto PREOPERATIONAL" beim Übergang zu STOP an den Knoten gesendet. Ist der Knoten zum Zeitpunkt des SPS- Starts noch nicht bereit, wird die "goto OPERATIONAL"-Nachricht ignoriert. Dies muss im Anwenderprogramm z.B. mit einer Statusabfrage berücksichtigt werden. Ggf. müssen die NMT-Kommandos in der S7-Anwendung programmiert werden.

Mit dem Auswahlfeld **„NMT-Download“** wird festgelegt, ob die Kommunikationsparameter „Guarding-Time“ / „Lifetime“ bzw. „Heartbeat-Time“ im Hochlauf an den Knoten heruntergeladen werden sollen. Dies ist sinnvoll sofern diese im Knoten nicht fest implementiert sind oder über andere Werkzeuge fest konfiguriert wurden.

Für die Prozessdaten stehen je **32 Tx- und Rx-PDOs** zur Verfügung. Werden mehr als **4 PDOs** davon benutzt, müssen PDO-Identifizier anderer Knotennummern verwendet werden. Diese Knotennummern sollten dann nicht mehr vergeben werden. (Es sind nur CiA-konforme TxPDO-Identifizier zulässig)

Empfangene Tx-PDOs werden zwischen-gespeichert und kontroll-punktsynchron in das Prozessabbild übertragen,

RxPDOs werden immer ereignis- und zeitgesteuert im Kontrollpunkt gesendet.

Jeder PDO ist ein 8-Byte-Datenfeld an einem wählbaren S7-Operanden zuzuweisen.

Die Tx-PDO-Kommunikationsparameter beeinflussen das Sendeverhalten des Knotens, die Rx-PDO-Kommunikationsparameter das Sendeverhalten des Masters:

Eigenschaft: Node 2 (CAN-Node)

Allgemein

Node-ID: 2

Device monitoring:

☐ Aus

☒ Heartbeat

☐ Nodeguard

Consumer time [ms]: 350

Producer time [ms]: 350

Guarding time [ms]:

Lifetime factor:

NMT control: ☒

NMT download: ☒

TxPDOs **RxPDOs** **SDOs**

Tx PDO Konfiguration

PDO-Nummer: Beschreibung:

Setting

☐ PDO Disable-Flag ☐ RTR-not-allowed-flag ☐ 29-bit-enable-flag

COB ID

☐ Konstant

☐ "Node ID" + Offset (hex) ID (hex):

☐ gemäß CiA DS301

Kommunikationsparameter

☐ Download

Inhibit time [1/10 ms]:

Event time [ms]:

Übertragungstyp:

S7 Mapping

Bereich:

Blocknummer:

Byte-Offset:

Mapping

☐ Download

Datenlänge (Bytes):

Datenart Index (hex) Sub index (hex) Beschreibung

1: 2: 3: 4: 5: 6: 7: 8:

Hinzufügen Übernehmen Reset Löschen

SDO-Konfiguration

Zeige CAN-Objekte

Beschreibung:

Index (hex):

Sub index (hex):

Datenart:

Daten:

Hinzufügen Übernehmen Reset Löschen

Konfiguration mit der Software „ConfigStage“

Inhibit-Zeit (Sperrzeit) verhindert ein zu häufiges Senden wenn sich Daten ständig ändern; die Event-Zeit (Sendeereignis) erzwingt das Senden auch wenn keine Datenänderung stattfindet.

Der Eintrag der einzelnen CAN-Objekte (Index, Subindex) im Mapping-Feld ist optional, solange der Download des Mappings nicht aktiviert ist. Der Datentyp ist aber immer anzugeben um die Telegrammlänge zu definieren und richtiges Byteswapping zu ermöglichen.

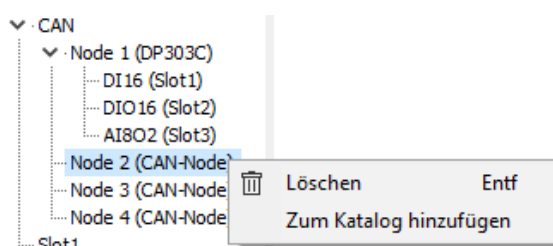
Bei aktiviertem Download der vollständigen Mapping-Parameter konfiguriert die SPS im Hochlauf den Inhalt der PDO.

Um weitere Konfigurationen im Anlauf vorzunehmen, können **SDOs** definiert werden. Damit werden z.B. Betriebsmodi oder Messbereiche eingestellt.

Bibliothekselemente aus eigenen CAN-Slaves erstellen

Optional: Wenn diese Einstellungen fertig gestellt wurden, kann die Konfiguration lokal in der ConfigStage gespeichert werden, um diesen Slave ab sofort als vordefiniertes Bibliothekselement einsetzen zu können.

Dazu im Projektbaum mit rechter Maustaste auf den eben konfigurierten CAN-Knoten gehen.



Danach öffnet sich eine Maske, in der die genaue Beschreibung (mit Bild) vorgenommen wird. Nach Bestätigung erscheint der neue CAN-Knoten mit den vorgelegtem Mapping in Ihrem Katalogverzeichnis auf der rechten Seite.

ConfigStage

Neues CAN-Knoten erzeugen
Festlegen der Beschreibung des CAN-Knotens

Bibliotheksdateiname:

Geräteinformation

Name:

Hersteller:

Bestellnr.:

Beschreibung:

Modulbild:

Das Bild wird im Modulfenster gezeigt, wenn das Modul ausgewählt ist.

Katalogbild:

Das Bild wird im Katalogfenster gezeigt, wenn das Modul ausgewählt ist.

Ok

Abbrechen

ACHTUNG:

Die Bezeichnungen TX (Transmit) und RX (Receive) sind bei CANopen®-PDOs immer aus Sicht des CAN-Knotens (Slave). D.h. mit einer TX-PDO sendet ein Knoten seine Eingangsdaten zur SPS und mit einer RX-PDO empfängt er seine Ausgangsdaten.

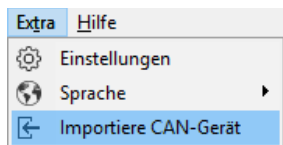
INSEVIS Vertriebs GmbH • Am Weichselgarten 7 • D-91058 Erlangen

Handbuch Kompakt-SPS, Rev. 01 / 2026 **91**

Konfiguration mit der Software „ConfigStage“

Dezentrale Fremdperipherie per EDS konfigurieren

Voraussetzung: Existenz eines zum verwendeten CAN-Slave passenden EDS-Files (*.eds). Diese Datei wird importiert.



Danach öffnet sich das gleiche Fenster wie bei der Anlage von CAN-Bibliothekselementen wie oben gezeigt. Nach Zuweisen von Text und Bildfeldern steht ein „leeres“ Bibliothekselement zur Verfügung, welches konfiguriert wird, wie nachfolgend beschrieben.

PDO-Konfiguration

Durch „Zeige CAN-Objekte“ öffnet sich ein Objekt-Browser mit den zur Verfügung stehenden CAN-Objekten. Diese wurden automatisch aus der EDS-Datei herausgefiltert. Durch Drag'n Drop werden einzelne oder mehrere CAN-Objekte in die Konfiguration übernommen.

TxPDOs

PDO-Nummer: 1 Beschreibung:

Setting

☐ PDO Disable-Flag ☐ RTR-not-allowed-flag ☐ 29-bit-enable-flag

COB ID

☐ Konstant ☐ "Node ID" + Offset (hex) ☒ gemäß CIA DS301

ID (hex): 18A

Kommunikationsparameter

☐ Download

Inhibit time [1/10 ms]: 100

Event time [ms]: 500

Übertragungstyp: 255

S7 Mapping

Bereich: Eingänge

Blocknummer:

Byte-Offset: 0

Mapping

☒ Download Datenlänge (Bytes): 8 **Zeige CAN-Objekte**

Datentyp	Index (hex)	Sub index (hex)	Beschreibung
Byte	6000	01	8 bit digital input block area.1. digital in
Byte	6000	02	8 bit digital input block area.2. digital in
Byte	6000	03	8 bit digital input block area.3. digital in
Byte	6000	04	8 bit digital input block area.4. digital in
Byte	6000	05	8 bit digital input block area.5. digital in
Byte	6000	06	8 bit digital input block area.6. digital in
Byte	6000	07	8 bit digital input block area.7. digital in
Byte	6000	08	8 bit digital input block area.8. digital in

Hinzufügen Übernehmen Reset Löschen

CAN-Objekte

Suche CAN-Objekt ☐ Zeige Werte im hexadezimalen Format

Beschreibung	Index	Sub index	Datentyp
8 bit digital input block area.1. digital input block	6000	01	UINT8
8 bit digital input block area.2. digital input block	6000	02	UINT8
8 bit digital input block area.3. digital input block	6000	03	UINT8
8 bit digital input block area.4. digital input block	6000	04	UINT8
8 bit digital input block area.5. digital input block	6000	05	UINT8
8 bit digital input block area.6. digital input block	6000	06	UINT8
8 bit digital input block area.7. digital input block	6000	07	UINT8
8 bit digital input block area.8. digital input block	6000	08	UINT8
8 bit digital input block area.9. digital input block	6000	09	UINT8
8 bit digital input block area.10. digital input block	6000	0A	UINT8
8 bit digital input block area.11. digital input block	6000	0B	UINT8
8 bit digital input block area.12. digital input block	6000	0C	UINT8
8 bit digital input block area.13. digital input block	6000	0D	UINT8
8 bit digital input block area.14. digital input block	6000	0E	UINT8
8 bit digital input block area.15. digital input block	6000	0F	UINT8
8 bit digital input block area.16. digital input block	6000	10	UINT8
8 bit digital input block area.17. digital input block	6000	11	UINT8
8 bit digital input block area.18. digital input block	6000	12	UINT8
8 bit digital input block area.19. digital input block	6000	13	UINT8
8 bit digital input block area.20. digital input block	6000	14	UINT8
8 bit digital input block area.21. digital input block	6000	15	UINT8
8 bit digital input block area.22. digital input block	6000	16	UINT8
8 bit digital input block area.23. digital input block	6000	17	UINT8
8 bit digital input block area.24. digital input block	6000	18	UINT8
8 bit digital input block area.25. digital input block	6000	19	UINT8
8 bit digital input block area.26. digital input block	6000	1A	UINT8
8 bit digital input block area.27. digital input block	6000	1B	UINT8
8 bit digital input block area.28. digital input block	6000	1C	UINT8
8 bit digital input block area.29. digital input block	6000	1D	UINT8
8 bit digital input block area.30. digital input block	6000	1E	UINT8
8 bit digital input block area.31. digital input block	6000	1F	UINT8
8 bit digital input block area.32. digital input block	6000	20	UINT8
8 bit digital input block area.33. digital input block	6000	21	UINT8
8 bit digital input block area.34. digital input block	6000	22	UINT8
8 bit digital input block area.35. digital input block	6000	23	UINT8
8 bit digital input block area.36. digital input block	6000	24	UINT8
8 bit digital input block area.37. digital input block	6000	25	UINT8

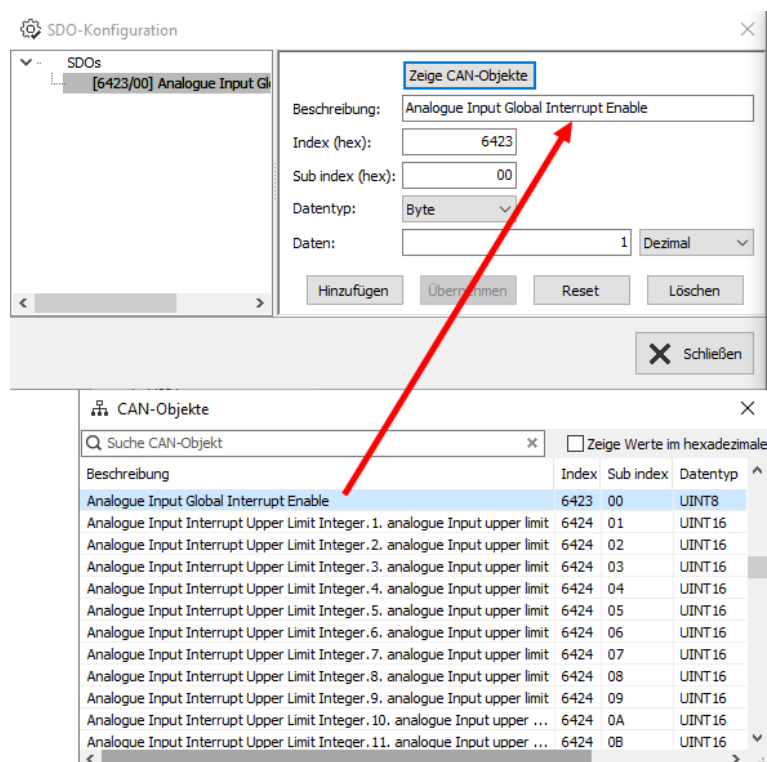
Schließen

Konfiguration mit der Software „ConfigStage“

SDO-Konfiguration

Durch „Zeige CAN-Objekte“ öffnet sich ein Objekt-Browser mit den zur Verfügung stehenden CAN-Objekten. Diese wurden automatisch aus der EDS-Datei herausgefiltert.

Durch Drag’n Drop werden einzelne oder mehrere CAN-Objekte in die Konfiguration übernommen. Im Datenfeld ist der Parameterwert einzutragen.



Dezentrale Fremdperipherie - vordefiniert

Vordefinierte CAN-Geräte, insbesondere Antriebsregler, ermöglichen dem Anwender über S7 sofort auf diese Geräte zuzugreifen. Da die Konfigurationsdaten offen liegen, können diese jedoch auch anwendungsspezifisch angepasst oder erweitert werden.

INSEVIS stellt auf seiner Webseite kostenlos ein umfangreiches S7-Beispielprogramm mit Funktionsbausteinen für Motion Controller zur Verfügung, mit denen hochkomplexe intelligente Antriebe (z.B. Parker C3, Maxon EPOS2, etc.) quasi ohne Aufwand in ein S7-Programm implementiert werden können. Diese S7-Programme setzen auf die vordefinierten CAN-Geräte der ConfigStage auf.

Aufgrund der Struktur mit vielen kleinen Funktionsbausteinen und einer gemeinsamen Schnittstelle zur Hardware (PDO-Mapping der ConfigStage) ist dieses Beispiel leicht auf andere Antriebe portierbar.

Die komplette Beschreibung mit Demoprogramm und Bausteinen steht im Internet unter <http://www.insevis.de/de/service> zur Verfügung.

Konfiguration mit der Software „ConfigStage“

Allgemeine Beispiele für CANopen Fremdgeräte

Beispielkonfiguration dezentrale CANopen-Peripherie nach DS401

Wird ein Gerät mit digitalen und analogen Ein- und Ausgängen nach CANopen-Profil DS401 eingesetzt, befinden sich

- die digitalen Eingänge in TxPDO1, z.B. 32 Bit

The screenshot shows the 'Tx PDO Configuration' window. The 'PDO-Nummer' is set to 1 and the 'Beschreibung' is 'Digital Inputs'. Under 'Setting', 'PDO Disable-Flag', 'RTR-not-allowed-flag', and '29-bit-enable-flag' are unchecked. 'COB ID' is set to 'gemäß CIA DS301'. Under 'Kommunikationsparameter', 'Download' is checked, 'Inhibit time [1/10 ms]' is 100, 'Event time [ms]' is 500, and 'Übertragungstyp' is 255. Under 'S7 Mapping', 'Bereich' is 'Eingänge', 'Blocknummer' is 1, and 'Byte-Offset' is 0. The 'Mapping' section shows a table with 8 rows, each with a 'Datentyp' (Byte) and a 'Beschreibung' (Digital Input Byte 0 to 7). Buttons at the bottom include 'Hinzufügen', 'Übernehmen', 'Reset', 'Löschen', and 'Schließen'.

- die digitalen Ausgänge in RxPDO1, z.B. 16 Bit

The screenshot shows the 'Rx PDO Configuration' window. The 'PDO-Nummer' is set to 1 and the 'Beschreibung' is 'Digital Outputs'. Under 'Setting', 'PDO Disable-Flag', 'RTR-not-allowed-flag', and '29-bit-enable-flag' are unchecked. 'COB ID' is set to 'gemäß CIA DS301'. Under 'Kommunikationsparameter', 'Download' is checked, 'Inhibit time [1/10 ms]' is 100, 'Event time [ms]' is 500, and 'Übertragungstyp' is 255. Under 'S7 Mapping', 'Bereich' is 'Ausgänge', 'Blocknummer' is 1, and 'Byte-Offset' is 0. The 'Mapping' section shows a table with 8 rows, each with a 'Datentyp' (Byte) and a 'Beschreibung' (Digital Output Byte 0 to 7). Buttons at the bottom include 'Hinzufügen', 'Übernehmen', 'Reset', 'Löschen', and 'Schließen'.

- die analogen Eingänge in TxPDO2 bis 4

The screenshot shows the 'Tx PDO Configuration' window. The 'PDO-Nummer' is set to 2 and the 'Beschreibung' is 'Analog Inputs'. Under 'Setting', 'PDO Disable-Flag', 'RTR-not-allowed-flag', and '29-bit-enable-flag' are unchecked. 'COB ID' is set to 'gemäß CIA DS301'. Under 'Kommunikationsparameter', 'Download' is checked, 'Inhibit time [1/10 ms]' is 100, 'Event time [ms]' is 500, and 'Übertragungstyp' is 255. Under 'S7 Mapping', 'Bereich' is 'Eingänge', 'Blocknummer' is 1, and 'Byte-Offset' is 4. The 'Mapping' section shows a table with 8 rows, each with a 'Datentyp' (Word) and a 'Beschreibung' (Analog Input Word 0 to 7). Buttons at the bottom include 'Hinzufügen', 'Übernehmen', 'Reset', 'Löschen', and 'Schließen'.

- die analogen Ausgänge in RxPDO2 bis 4

The screenshot shows the 'Rx PDO Configuration' window. The 'PDO-Nummer' is set to 2 and the 'Beschreibung' is 'Analog Outputs'. Under 'Setting', 'PDO Disable-Flag', 'RTR-not-allowed-flag', and '29-bit-enable-flag' are unchecked. 'COB ID' is set to 'gemäß CIA DS301'. Under 'Kommunikationsparameter', 'Download' is checked, 'Inhibit time [1/10 ms]' is 100, 'Event time [ms]' is 500, and 'Übertragungstyp' is 255. Under 'S7 Mapping', 'Bereich' is 'Ausgänge', 'Blocknummer' is 1, and 'Byte-Offset' is 2. The 'Mapping' section shows a table with 8 rows, each with a 'Datentyp' (Word) and a 'Beschreibung' (Analog Output Word 0 to 7). Buttons at the bottom include 'Hinzufügen', 'Übernehmen', 'Reset', 'Löschen', and 'Schließen'.

Die Ein- und Ausgänge können damit an der konfigurierten S7-Adresse (im Bsp. ab EB0, AB0) wie gewohnt angesprochen werden.



ACHTUNG:

Die PDO-Nummer 1 ist ausschließlich für digitale E/As reserviert, die analogen E/As können erst ab PDO-Nummer 2 zugewiesen werden!

Konfiguration mit der Software „ConfigStage“

Um Parameter zu übertragen (z.B. Messbereiche für Analogmodule), müssen SDOs herstellerspezifisch konfiguriert werden.

Beispielkonfiguration CANopen-Antriebsregler nach DS402

Wird ein CANopen-Antriebsregler nach Profil DS402 eingesetzt, muss als Minimum die reglerinterne Zustandsmaschine über

- ein 16-Bit Steuerwort (control word) und

- ein 16-Bit Statuswort (status word) verwaltet werden.

Diese beiden Wörter sind über das PDO-Mapping S7-Operanden zuzuordnen und dann kann/muss der Regler aus dem S7-Programm gesteuert werden.

Konfiguration mit der Software „ConfigStage“

Allgemeine Beispiele für CANopen Fremdgeräte

Beispielkonfiguration dezentrale CANopen-Peripherie nach DS401

Wird ein Gerät mit digitalen und analogen Ein- und Ausgängen nach CANopen-Profil DS401 eingesetzt, befinden sich

- die digitalen Eingänge in TxPDO1, z.B. 32 Bit

Tx PDO Konfiguration

PDO-Nummer: 1 Beschreibung: Digital Inputs

Setting

☐ PDO Disable-Flag ☐ RTR-not-allowed-flag ☐ 29-bit-enable-flag

COB ID

☐ Konstant ☐ "Node ID" + Offset (hex) ☐ gemäß CIA DS301

Kommunikationsparameter

☒ Download

Inhibit time [1/10 ms]: 100

Event time [ms]: 500

Übertragungstyp: 255

S7 Mapping

Bereich: Eingänge

Blocknummer: 1

Byte-Offset: 0

Mapping

☐ Download

Datenlänge (Bytes): 4

Zeige CAN-Objekte

Datentyp	Beschreibung
1: Byte	Digital Input Byte 0
2: Byte	Digital Input Byte 1
3: Byte	Digital Input Byte 2
4: Byte	Digital Input Byte 3
5:	
6:	
7:	
8:	

Hinzufügen Übernehmen Reset Löschen

Schließen

- die digitalen Ausgänge in RxPDO1, z.B. 16 Bit

Rx PDO Konfiguration

PDO-Nummer: 1 Beschreibung: Digital Outputs

Setting

☐ PDO Disable-Flag ☐ RTR-not-allowed-flag ☐ 29-bit-enable-flag

COB ID

☐ Konstant ☐ "Node ID" + Offset (hex) ☐ gemäß CIA DS301

Kommunikationsparameter

☒ Download

Inhibit time [1/10 ms]: 100

Event time [ms]: 500

Übertragungstyp: 255

S7 Mapping

Bereich: Ausgänge

Blocknummer: 1

Byte-Offset: 0

Mapping

☐ Download

Datenlänge (Bytes): 2

Zeige CAN-Objekte

Datentyp	Beschreibung
1: Byte	Digital Output Byte 0
2: Byte	Digital Output Byte 1
3:	
4:	
5:	
6:	
7:	
8:	

Hinzufügen Übernehmen Reset Löschen

Schließen

- die analogen Eingänge in TxPDO2 bis 4

Tx PDO Konfiguration

PDO-Nummer: 2 Beschreibung: Analog Inputs

Setting

☐ PDO Disable-Flag ☐ RTR-not-allowed-flag ☐ 29-bit-enable-flag

COB ID

☐ Konstant ☐ "Node ID" + Offset (hex) ☐ gemäß CIA DS301

Kommunikationsparameter

☐ Download

Inhibit time [1/10 ms]: 100

Event time [ms]: 500

Übertragungstyp: 255

S7 Mapping

Bereich: Eingänge

Blocknummer: 1

Byte-Offset: 4

Mapping

☐ Download

Datenlänge (Bytes): 4

Zeige CAN-Objekte

Datentyp	Beschreibung
1: Word	AnalogeInput Word 0
2: Word	AnalogeInput Word 1
3:	
4:	
5:	
6:	
7:	
8:	

Hinzufügen Übernehmen Reset Löschen

Schließen

- die analogen Ausgänge in RxPDO2 bis 4

Rx PDO Konfiguration

PDO-Nummer: 2 Beschreibung: Analog Outputs

Setting

☐ PDO Disable-Flag ☐ RTR-not-allowed-flag ☐ 29-bit-enable-flag

COB ID

☐ Konstant ☐ "Node ID" + Offset (hex) ☐ gemäß CIA DS301

Kommunikationsparameter

☒ Download

Inhibit time [1/10 ms]: 100

Event time [ms]: 500

Übertragungstyp: 255

S7 Mapping

Bereich: Ausgänge

Blocknummer: 1

Byte-Offset: 2

Mapping

☐ Download

Datenlänge (Bytes): 8

Zeige CAN-Objekte

Datentyp	Beschreibung
1: Word	Analog Output Word 0
2: Word	Analog Output Word 1
3: Word	Analog Output Word 2
4: Word	Analog Output Word 3
5:	
6:	
7:	
8:	

Hinzufügen Übernehmen Reset Löschen

Schließen

Die Ein- und Ausgänge können damit an der konfigurierten S7-Adresse (im Bsp. ab EB0, AB0) wie gewohnt angesprochen werden.



ACHTUNG:

Die PDO-Nummer 1 ist ausschließlich für digitale E/As reserviert, die analogen E/As können erst ab PDO-Nummer 2 zugewiesen werden!

Konfiguration mit der Software „ConfigStage“

Um Parameter zu übertragen (z.B. Messbereiche für Analogmodule), müssen SDOs herstellerspezifisch konfiguriert werden.

Beispielkonfiguration CANopen-Antriebsregler nach DS402

Wird ein CANopen-Antriebsregler nach Profil DS402 eingesetzt, muss als Minimum die reglerinterne Zustandsmaschine über

- ein 16-Bit Steuerwort (control word) und

Datenlänge (Bytes): 6	Datentyp	Index (hex)	Sub index (hex)	Beschreibung
1	Word	6040	00	Controlword
2	Word	6060	00	Modes of Operation
3	Word	607A	00	Target Position
4				
5				
6				
7				
8				

- ein 16-Bit Statuswort (status word) verwaltet werden.

Datenlänge (Bytes): 8	Datentyp	Index (hex)	Sub index (hex)	Beschreibung
1	Word	6041	00	Statusword
2	Word	6061	00	Modes of Operation Display
3	Word	6100	01	DigitalInput
4	Word	603F	00	Error code
5				
6				
7				
8				

Diese beiden Wörter sind über das PDO-Mapping S7-Operanden zuzuordnen und dann kann/muss der Regler aus dem S7-Programm gesteuert werden.

Systemfunktionen

INSEVIS bietet OBs, SFBs und SFCs an, die kompatibel zu STEP®7 von Siemens sind. Darüber hinaus stellt INSEVIS in einer kostenlosen S7-Bibliothek weitere OBs, SFBs und SFCs für die Sonderfunktionen im Internet zur Verfügung.

Im Vergleich zur Siemens-CPU S7-315-2PN DP mit Firmware 3.2.7 (6ES7 315-2EH14-0AB0) stehen bei den INSEVIS-CPUs -V / -P und -T folgende OB, SFB und SFC zur Verfügung

Organisationsbausteine (OB)

Organisationsbausteine (OB) - kompatibel zu STEP®7 von Siemens

OB	Funktion
OB 1	Zyklus-OB
OB 10	Uhrzeitalarm
OB 32,33,34	Weckalarme (nur CPU-T)
OB 35	Weckalarm
OB 80	Zeitfehler
OB 82	Diagnosealarm
OB 85	Programmablauffehler
OB 86	Baugruppenträgerausfall (bei INSEVIS Ausfall der externen Peripherie) <i>Lokaldaten „OB86_MDL_ADDR“ (LW6) hat den festen Wert 2047 (W#16#7FF)</i>
OB 100	Neustart
OB 121	Programmierfehler
OB 122	Peripheriezugriffsfehler

Organisationsbausteine (OB) - in Erweiterung zu STEP®7 von Siemens

OB	Funktion (mehr Informationen in einzelnen OB-Kapiteln)
OB 106	CAN- Empfangsereignis (siehe Kommunikation / CAN)
OB 186	CAN- Fehlerereignis

Systemfunktionsbausteine (SFB)

Systemfunktionsbausteine (SFB) - kompatibel zu STEP®7 von Siemens

SFB	Funktion
SFB 0 CTU	Vorwärtszähler
SFB 1 CTD	Rückwärtszähler
SFB 2 CTUD	Vorwärts- und Rückwärtszähler
SFB 3 TP	Impulsbildung
SFB 4 TON	Einschaltverzögerung
SFB 5 TOF	Ausschaltverzögerung
SFB 60	Daten senden mit ASCII-Treiber (aus CPU S7-314, B-Nr, 6ES7314-1AG14-0AB0)
SFB 61	Daten empfangen mit ASCII-Treiber (aus CPU S7-314, B-Nr, 6ES7314-1AG14-0AB0)
SFB 52 / 53 / 54	Asynchronen Datenrecord lesen / schreiben / Alarm lesen (Nur in der CPU-T (für Profinet) verfügbar)
SFB 81	Dynamische Parameter lesen (Nur in der CPU-T (für Profinet) verfügbar)

Systemfunktionsbausteine (SFB) - die bei der INSEVIS-CPU noch NICHT vorhanden und nur auf Anfrage erhältlich sind

SFB	Funktion
SFB 32 DRUM	Realisieren eines Schrittschaltwerks
SFB 52 / 53 / 54 / 81	Diese sind auf Grund unterschiedlicher CPU-Hardware nicht verfügbar
SFB 73 / 74	Profibus DP → Siemensinterner Baustein, für INSEVIS unnötig
SFB 75 SALRM	Diese Funktionen liegen außerhalb der angebotenen Profibus DP-V0-Funktion der INSEVIS-Profibus-Baugruppen
SFB 104 IP_CONF	Konfiguration von IP-Adressen → verwenden Sie SFB 129 IP_CFG dafür

Systemfunktionen

Systemfunktionsbausteine (SFB) - in Erweiterung zu STEP®7 von Siemens

SFB	Funktion (mehr Informationen in einzelnen SFB-Kapiteln)
SFB 105	CAN-Senden
SFB 106	CAN-Empfangen
SFB 107	CAN-SDO-Transfer
SFB 114	CAN-Statusabfrage
SFB 120 GET	Daten lesen von einer Partner SPS
SFB 121 PUT	Daten schreiben an eine Partner SPS
SFB 122 TSEND	Daten senden über TCP
SFB 123 TRECVC	Daten empfangen über TCP
SFB 124 TDISCON	Beenden der TCP-Verbindung
SFB 125 TCONFIG	Konfiguration der Partnerparameter bei TCP
SFB 126 TSTATUS	Statusabfrage bei TCP
SFB 127 USEND	Daten senden über UDP
SFB 128 URECV	Daten empfangen über UDP
SFB 129 IP_CFG	IP-Konfiguration der SPS
SFB 130 CFG_PTP	UART-Konfiguration ändern
SFB 131 IP_GET	Lesen der IP-Konfiguration (nur CPU-T)
SFB 132 IP_SET	Schreiben der IP-Konfiguration (nur CPU-T)

Systemfunktionen (SFC)

Systemfunktionen (SFC) - kompatibel zu STEP®7 von Siemens

SFC	Funktion
SFC 0 SET_CLK	Setzen der Uhrzeit
SFC 1 READ_CLK	Lesen der Uhrzeit
SFC 6 RD_SINFO	Startinformation des aktuellen OBs auslesen
SFC 11 DPSYC_FR	Gruppen von DP-Slaves synchronisieren; <i>Parameter „LADDR“ muss festen Wert 2047 haben (W#16#7FF)</i>
SFC 12 D_ACT_DP	Deaktivieren und Aktivieren von DP - Slaves
SFC 13 DPNRM_DG	Lesen der Diagnosedaten (Slave-Diagnose) eines DP-Slaves
SFC 14 DPRD_DAT	Konsistente Daten eines DP-Normslaves lesen
SFC 15 DPWR_DAT	Daten konsistent auf einen DP-Normslave schreiben
SFC 20 BLKMOV	Variable kopieren
SFC 21 FILL	Feld vorbesetzen
SFC 24 TEST_DB	Testen eines Datenbausteines
SFC 28 SET_TINT	Uhrzeitalarm stellen
SFC 29 CAN_TINT	Uhrzeitalarm stornieren
SFC 30 ACT_TINT	Uhrzeitalarm aktivieren
SFC 31 QRY_TINT	Uhrzeitalarm abfragen
SFC 43 RE_TRIGR	Zykluszeitüberwachung nachtriggern
SFC 46 STP	CPU in Betriebszustand STOP überführen
SFC 51 RDSYSST	Systemstatusliste lesen (derzeit nur unterstützt: SSL_ID W#16#0092, W#16#0292, W#16#0692 für ProfiBus DP Master System)
SFC 64 TIME_TCK	Systemzeit lesen
SFC 101 RTM	Hantieren mit Betriebsstundenzähler

Systemfunktionen

Systemfunktionen (SFC) -

die bei der INSEVIS-CPU noch NICHT vorhanden und nur auf Anfrage erhältlich sind

SFC	Funktion
SFC 2 / 3 / 4	Bei INSEVIS steht mit SFC101 ein 32bit Betriebsstundenzähler zur Verfügung
SFC 5 / 50 / 70 / 71	Diese Funktionen sind auf Grund unterschiedlicher Hardware nicht möglich / nötig
SFC 7 / 55 / 56 / 57 / 102	Diese Funktionen liegen außerhalb der angebotenen Profibus DP-V0-Funktion der INSEVIS-Profibus-Baugruppen
SFC 17 / 18 / 19 / 107 / 108	INSEVIS verfügt über eine eigene Alarmverarbeitung in VisuStage
SFC 22 CREAT_DB	Einen Datenbaustein erzeugen
SFC 23 DEL_DB	Einen Datenbaustein löschen
SFC 32 / 33 / 34 / 47	INSEVIS unterstützt keine Verzögerungsalarme
SFC 36	Synchronfehlerereignisse maskieren
SFC 37	Synchronfehlerereignisse demaskieren
SFC 38	Ereignisstatusregister lesen
SFC 39	nur bei CPU-T / nicht bei CPU-V/P Bearbeitung neuer Alarm- und Asynchronfehlerereignisse sperren
SFC 40	nur bei CPU-T / nicht bei CPU-V/P Bearbeitung neuer Alarm- und Asynchronfehlerereignisse freigeben
SFC 41	nur bei CPU-T / nicht bei CPU-V/P Bearbeitung von höherprioren Alarm- und Asynchronfehlerereignissen verzögern
SFC 42	nur bei CPU-T / nicht bei CPU-V/P Bearbeitung von höherprioren Alarm- und Asynchronfehlerereignissen freigeben
SFC 44	Ersatzwert in AKKU 1 übertragen
SFC 58	Datensatz schreiben
SFC 59	Datensatz lesen
SFC 65-69 / 72-74	INSEVIS unterstützt keine MPI- Kommunikation
SFC 81 UBLKMOV	Variable ununterbrechbar kopieren
SFC 82 CREA_DBL	Datenbaustein im Ladespeicher erzeugen, für INSEVIS unnötig
SFC 83 READ_DBL	Lesen aus einem Datenbaustein im Ladespeicher, für INSEVIS unnötig
SFC 84 WRIT_DBL	Schreiben in einen Datenbaustein im Ladespeicher, für INSEVIS unnötig
SFC 99 WWW	Siemens-interner Baustein, für INSEVIS unnötig

Systemfunktionen (SFC) - in Erweiterung zu STEP®7 von Siemens

SFC	Funktion (mehr Informationen in einzelnen SFC-Kapiteln)
SFC 75 SET_ADDR	Setzen der Profibus DP Slave Adresse
SFC 76 SET_DPS	Setzen der Profibus DP Slave Adresse und der Baudrate
SFC 200 SCR_GET	Abfrage der aktuellen Bildschirm-ID-Nummer
SFC 201 SCR_SET	Bildschirmwechsel
SFC 202 TRENDSRT	Starten der Trenderfassung
SFC 203 TRENDSTP	Stoppen der Trenderfassung
SFC 204 TRENDCNT	Fortsetzen der Trenderfassung
SFC 205 TRENDSTA	Trendstatusabfrage
SFC 206 RECIPE	Rezepturhandling
SFC 207 ARCHIVE	Daten archivieren auf Micro-SD-Karte
SFC 208 BACKUP	Backup
SFC 209 READFILE	Daten lesen aus Archiv-Datei
SFC 210 COPY_DBL	einige DBs von RAM nach ROM kopieren
SFC 211 COPY_DBW	einige DBs von ROM nach RAM kopieren
SFC 213 READ_CSV	Daten lesen aus csv-formatierter Archivdatei von Micro-SD-Karte

Systemfunktionen

Fortsetzung: Systemfunktionen (SFC) - in Erweiterung zu STEP®7 von Siemens

SFC	Funktion (mehr Informationen in einzelnen SFC-Kapiteln)
SFC 214 LCD_BRIG	Einstellen der Hinterleuchtungshelligkeit
SFC 215 LOGIN	Ändern des Benutzerlevels zur Laufzeit
SFC 220 RENAME	Umbenennen von Dateien / Ordnern auf der Micro-SD-Karte
SFC 221 DELETE	Löschen von Dateien / Ordnern auf der Micro-SD-Karte
SFC 222 COPY	Kopieren von Dateien auf der Micro-SD-Karte
SFC 223 CREATE	Erstellen von Ordnern auf der Micro-SD-Karte
SFC 224 INFO	Informationen zu Dateien / Ordnern auf der Micro-SD-Karte
SFC 225 FILEINFO	Informationen zu Dateinamen und -größen auf der Micro-SD-Karte
SFC 254 RAM2ROM	alle DBs von RAM nach ROM kopieren

Import der INSEVIS-Zusatzbausteine in das S7-Projekt

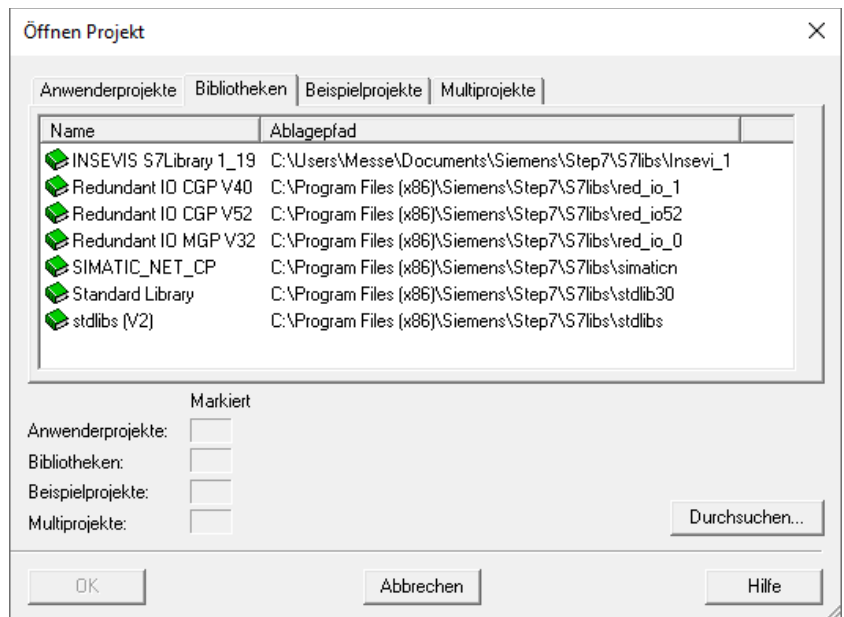


ACHTUNG:

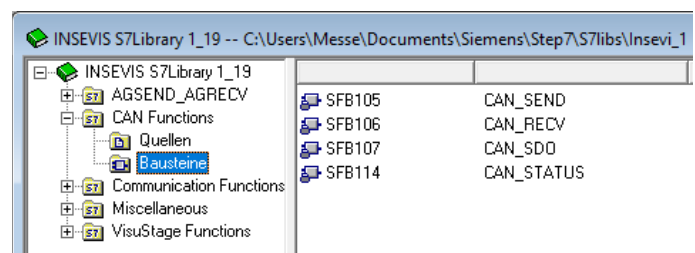
Die INSEVIS- Bausteine stehen als **S7-Bibliothek** auf den INSEVIS Webseiten im Downloadbereich unter „Erste Schritte / S7-Bibliotheken“ zum freien Download bereit. Es wird immer die aktuellste mit den meisten Funktionen bereitgehalten, die auch das aktuellste Betriebssystem der INSEVIS- Steuerung erfordert. (z.B. Insevis_S7-library_from_2_1_5.zip - funktioniert ab INSEVIS-Betriebssystem 2.1.5 und höher).

Installation der INSEVIS-S7-Bibliothek im Simatic®-Manager

- 1) Datei → dearchivieren → Auswahl von INSEVIS S7-Bibliotheks.zip
- 2) Speichern → (im Ordner „S7-Libs“) → mit „OK“ bestätigen
- 3) Bibliotheks-Projekt öffnen
→ Reiter „Bibliotheken“ wählen
→ „INSEVIS S7-Library“ auswählen
→ mit „OK“ bestätigen



- 4) Kopieren der ausgewählten SFB's durch Drag'n Drop in das S7-Projekt



VIDEO-Tutorial verfügbar

Für diesen Vorgang ist auf den INSEVIS-Download-Seiten ein Link auf ein entsprechendes Video im YouTube®-Kanal von INSEVIS verfügbar (für **Simatic®-Manager**).

Systemfunktionen

Installation der INSEVIS-S7-Bibliothek im TIA-Portal®

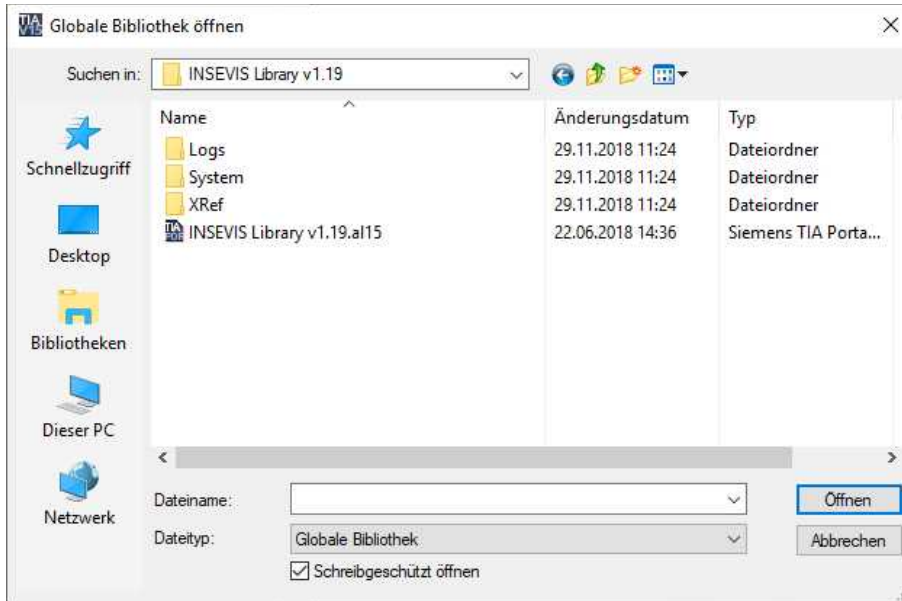


Bild links:

Aktuellste Version der INSEVIS Bibliothek für das TIA-Portal:

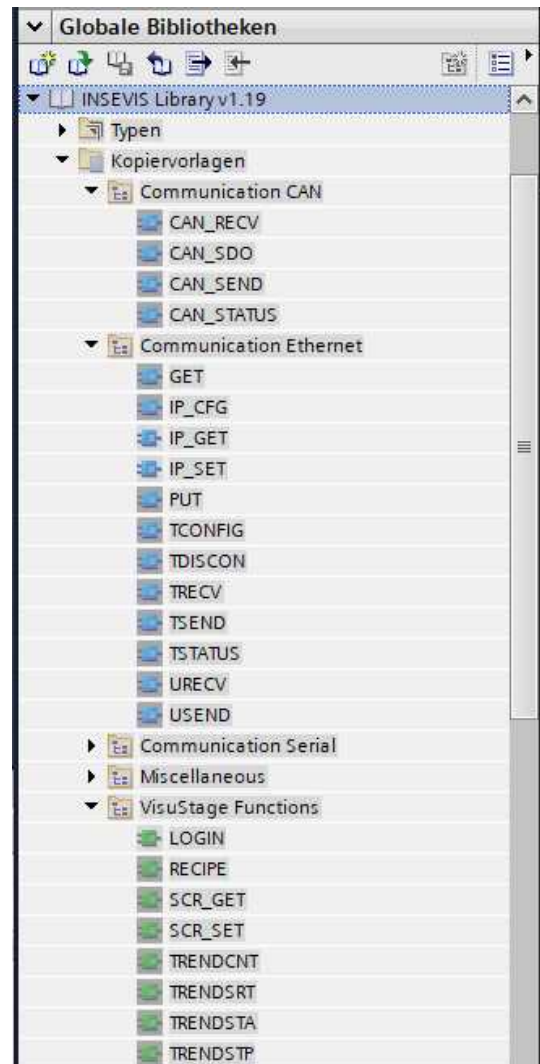
- downloaden,
- speichern,
- in einen beliebigen Ordner extrahieren.

Bild rechts:

Bibliothek **"INSEVIS Library v1.19 al15"** aus jeweiligen Ordner im TIA-Portal öffnen als "Global library".
→ Für Details zur Benutzung von Bibliothekselementen bitte die TIA-Portal Onlinehilfe benutzen.

Ziehen der gewünschten Bibliothekselemente
• per Drag'n Drop in die "Programmblöcke" Ihres Projektes

→ Für Details zur Benutzung von Bibliothekselementen bitte die TIA-Portal Onlinehilfe benutzen.



VIDEO-Tutorial verfügbar

Für diesen Vorgang ist auf den INSEVIS-Download-Seiten

ein Link auf ein entsprechendes Video im YouTube®-Kanal von INSEVIS verfügbar (für TIA-Portal).

Systemfunktionen

Übersicht über SFCs zur Datensicherung und -wiederherstellung

Nachfolgend eine Auswahl verschiedener Möglichkeiten zur Datensicherung und -wiederherstellung manuell oder mit System-SFCs.

Zur Beachtung: Das S7-Programm selbst ist bei INSEVIS immer im Ladespeicher (Flash) und muss nicht extra gesichert werden.

Datenbausteine nach ROM/ ins Flash sichern

Mit dem **SFC 254** RAM2ROM werden alle DBs vom batteriegepufferten Arbeitsspeicher (RAM) in den Ladespeicher (ROM) kopiert, so dass sie permanent behalten werden.

Der **SFC 210** COPY_DBL kopiert nur ausgewählte DBs vom batteriegepufferten Arbeitsspeicher (RAM) in den Ladespeicher (ROM).

Der **SFC 211** COPY_DBW arbeitet entgegengesetzt und lädt ausgewählte DBs vom Ladespeicher (ROM) in den batteriegepufferten Arbeitsspeicher (RAM).

→ Anwendungsbeispiel 1: Inbetriebnahme von Seriengeräten

- Nach dem Programmaufspielen (siehe unten) werden die anlagenspezifischen Prozessdaten manuell erstellt (befinden sich im batteriegepufferten Arbeitsspeicher / RAM)
- Vor dem Versand werden diese Daten ins Flash (Ladespeicher) kopiert und sind so vor Batteriespannungsausfall geschützt

→ Anwendungsbeispiel 2: Prozessdaten sichern bei Betrieb durch Endanwender

- Werden über die Toucheingabe Werte optimiert, kann der Endkunde diese selbst z.B. über zugangsbeschränkte Taste oder Schlüsselschalter mit der Funktion RAM2ROM sichern

Life-Backup für Geräte austausch → vom Gerät auf Micro-SD-Karte

Mit dem **SFC 208** BACKUP werden ALLE Daten (S7-Programm, Visualisierung -incl. der von Bedienern geänderten PINs), Prozessdaten) auf die Micro-SD-Karte in eine Binärdatei (auf Wunsch verschlüsselt) geschrieben. Dazu wird die SPS so lange in STOP gesetzt, bis das Backup erfolgt ist. Damit kann selbst ein Endanwender ganz ohne PC-Programme bei baugleichen Geräten eine Kopie des Ursprungsgerätes erstellt werden, die auch bereits erfasste Prozessdaten beinhaltet. Natürlich nur mit Genehmigung des Maschinenbauers und seiner Passwortdatei.

→ Anwendungsbeispiel: Redundante Ersatzmaschine bereitstellen

- Ein Kunde will eine Alternativ- oder Ersatzmaschine haben, die bei Wartungsarbeiten an der Urmaschine immer mit denselben aktuellen Parametern der Urmaschine weiterarbeiten kann
- Dazu wird z.B. am Ende eines jeden Produktionszyklus mit dem SFC208 ein Backup erstellt
- Wird die Urmaschine abgeschaltet / gewartet, kann die Alternativ-/Ersatzmaschine durch ein Restore mit den Prozessdaten des letzten Produktionszyklus weiterarbeiten.

Manuelles Programm-Backup für Gerätehersteller → von ServiceStage auf PC

Manuell und ganz ohne SFC für Programmierer mit Zugang zu den Quelldaten:

- Mit der ServiceStage die Funktion „OnlineBackup“ ausführen
- Auf dem PC wird durch die ServiceStage ein Verzeichnis „Backups“ erstellt und dahin wird die Datei „S7-programs.wld“ kopiert, die ALLE S7-Programmbausteine, Datenbausteine und **Hardwarekonfigurationsdaten** (also die ConfigStage-Konfiguration) enthält
- In das gleiche Verzeichnis wird die Binärdatei der Visualisierung mit dem Namen „visudata.vsbin“ kopiert, wenn dieser Zugriff in dem dem VisuStage-Projekt explizit freigegeben wurde. Geänderte PINs werden NICHT mit abgelegt.

Manuelles Programm-Restore auf einer SD-Karte erstellen

Manuelle Erstellung einer Micro-SD Karte mit den gewünschten Quelldaten:

Auf der SD-Karte den Ordner „Backups“ erstellen

Die gewünschte „S7programs.wld“ Systemdaten und allen OB, DB, FB, SFB und SFC

und

die gewünschte Visualisierungs-Binärdatei eine „Visudata.vsbin in dem Ordner speichern

→ Anwendungsbeispiel: Standardprogramm einspielen bei Seriengeräten

- Ein Programmierer übergibt seinen Kunden so eine Micro-SD-Karte mit den Programmkomponenten an
- Nach Hardware-Urlöschen lädt jede SPS seine Programmdaten von dieser Micro-SD-Karte ganz ohne PC

S7-Programm-Restore für beide Backup-Methoden

Egal wie ein Backup erzeugt wurde, es wird von der Micro-SD-Karte immer gleich in die SPS geladen:

- Power OFF, Micro-SD-Karte mit dem Backup in SPS stecken
- Power ON, Hardware-Reset ausführen, abwarten bis RUN- und BATT-LEDs nicht mehr gleichzeitig blinken (bei Backup über **SFC 208** muss ggf. die Passwort-Datei „*backup.psw*“ mit auf Micro-SD-Karte sein)

Systemfunktionen

Daten archivieren /dearchivieren (zurücklesen)

Frei definierbare beliebige Daten können mit dem **SFC 207 ARCHIVE** auf die Micro-SD-Karte (**Binär oder im csv-Format**) archiviert werden. Mit dem **SFC 209 READFILE** können **Binärdateien** wieder von der Micro-SD-Karte in die SPS zurückgelesen werden.

→ Dabei stehen die Dateien auf der Micro-SD-Karte hierarchisch unterhalb des Ordners UserData\.

Ein Auslesen in den PC per Ethernet und Abspeichern in Netzlaufwerken ist per RemoteStage möglich.

→ Anwendungsbeispiel: Prozessdaten sichern bei S7-Update

- Ein Kunde hat mit seinem S7-Programm über lange Zeit seine Prozessdaten erarbeitet.
- Jetzt soll er ein S7-Programmupdate mit neuen Funktionen erhalten.
- Wird das neue S7-Programm überspielt, gehen damit die kundenspezifischen Daten verloren.
- Mit den **SFC 207 ARCHIVE** und **SFC 209 READFILE** können diese Daten während des S7-Programm-Updates sicher ausgelagert werden und bleiben erhalten.
- Der Kunde kann mit dem neuen S7-Programm und seinen gesammelten Prozessdaten weiterarbeiten.

Daten zurücklesen aus am PC editierter CSV-Archivdatei

Während mit dem **SFC 209** nur Binärdaten - ohne Änderungsmöglichkeit- zurückgelesen werden können, kann mit dem **SFC 213** ein -eventuell vorab am PC geändertes- csv-Archiv- wieder in die SPS zurückgelesen werden.

→ Anwendungsbeispiel: Kundenspezifische Konfiguration von Seriengeräten

- Ihr S7-Programm ist immer gleich, aber die Voreinstellungen für Ihre Kunden sind unterschiedlich.
- Ein z.B. als Konfigurationsdatei mit dem **SFC 207** auf eine Micro-SD-Karte gespeichertes Datenarchiv kann vom Inbetriebnehmer an seinem PC editiert werden.
- Mit dem **SFC 213 READ_CSV** lesen Sie die csv-Datei von der Micro-SD-Karte wieder zurück in die SPS haben damit ganz einfach Ihre Serienprodukte bequem kundenspezifisch parametrisiert

Archivieren von Daten mit dem SFC 207 "ARCHIVE"

Mit dem SFC207 „ARCHIVE“ können S7-(Prozess- oder Datenbaustein-) daten im Binär- oder Text (csv)-Format auf die Micro-SD-Karte gespeichert werden.

Parameter	Bezeichnung	Datentyp	Beschreibung
REQ	INPUT	BOOL	TRUE startet den Job
FILENAME	INPUT	STRING	Dateiname (Muss dem 8.3-Format entsprechen)
APPEND	INPUT	BOOL	TRUE An bestehende Datei anhängen FALSE Bestehende Datei überschreiben
CSV	INPUT	BOOL	TRUE Konvertiere Daten zu Text und speichern als Textdatensatz im CSV-Format. FALSE Speichere Daten im Binärformat
CSV_STRUCT	INPUT	ANY	CSV Datensatzstruktur (Dieser Parameter wird bei CSV = FALSE nicht berücksichtigt)
DATA	INPUT	ANY	Zeiger auf S7-Prozessdaten
RET_VAL	OUTPUT	WORD	Rückgabewert des Jobs
BUSY	OUTPUT	BOOL	TRUE Job wird bearbeitet FALSE Job ist abgeschlossen
DONE	OUTPUT	BOOL	TRUE Job ist erfolgreich abgeschlossen
ERROR	OUTPUT	BOOL	TRUE Job ist fehlgeschlagen

Hinweise zu CSV-formatierter Textdatei (CSV=TRUE):

- Feldbegrenzer: “;” Semikoleon (ASCII-Code 59)
- Dezimalpunkt: “.” Punkt (ASCII-Code 46)
- Datensatz getrennt durch Zeilenumbruch (CRLF ASCII-Code 13, 10)
- Bitte beachten SIMATIC S7-(Prozess oder Datenblock) Datenversatz (Offset) in Datenbaustein.
(Daten im DB beginnen immer von einem geraden Offset außer BYTE, CHAR und BOOL)
- CSV_STRUCT zeigt auf die Datensatzstruktur der Archiv-Datei im txt (csv) Format.
Jedes Feld des Datensatzes in der csv-Datei belegt 2 Byte in CSV_STRUCT.
- das erste Byte bezeichnet den Datentyp
- das zweite Byte bezeichnet, wie die Daten zu Text zu konvertieren sind

Systemfunktionen



Die Struktur des Datenarchivs wird ausschließlich vom S7-Programm festgelegt.
 Die RemoteStage kann das Archiv zwar auslesen und abspeichern, kennt aber die Struktur nicht und kann deswegen das Archiv nicht ändern.
 Das FAT16/32 Filesystem auf der Micro-SD-Karte erlaubt einige Zeichen nicht in Datei- bzw.
 Verzeichnisnamen: \ / : * ? " < > | Solche Zeichen werden ignoriert.

Bedeutung von CSV_STRUCT

Erstes Byte (Datentyp)	Zweites Byte Anzeigeformat	Format
1	Daten sind BOOL	
	0	Als Integer 0 oder 1
	1	Als Text: FALSE oder TRUE
2	Daten sind BYTE	
	0	Als dezimaler Integer 0 .. 255
	1	Als hexadezimal B#16#0 .. B#16#FF
	2	Als hexadezimal 0x00 .. 0xFF
3	Daten sind CHAR	
	0	Als Zeichen, wenn Wert zwischen B#16#20 .. B#16#80. z.B. A
	1	Als hexadezimal B#16#0 .. B#16#FF
	2	Als hexadezimal 0x00 .. 0xFF
	3	Als dezimaler Integer -128 ... 127
4	Daten sind WORD	
	0	Als dezimaler Integer 0 ... 65535
	1	Als hexadezimal W#16#0 .. W#16#FFFF
	2	Als hexadezimal 0x0000 .. 0xFFFF
5	Daten sind INT	
	0	Als dezimaler Integer -32768 ... 32767
	1	Als hexadezimal W#16#0 .. W#16#FFFF
	2	Als hexadezimal 0x0000 .. 0xFFFF
6	Daten sind DWORD	
	0	Als dezimaler Integer 0 ... 4294967295
	1	Als hexadezimal DW#16#00000000 .. DW#16#FFFFFFFF
	2	Als hexadezimal 0x00000000 .. 0xFFFFFFFF
7	Daten sind DINT	
	0	Als dezimaler Integer -2147483648 ... 2147483647
	1	Als hexadezimal DW#16#00000000 .. DW#16#FFFFFFFF
	2	Als hexadezimal 0x00000000 .. 0xFFFFFFFF
8	Daten sind REAL	
	0..4	Anzahl der Nachkommastellen nach Dezimalpunkt (z.B. 12.456)
	Otherwise	2 Nachkommastellen nach Dezimalpunkt
		Achtung: Ausgabe NaN (Not A Number) wenn Daten nicht im REAL-Format sind

Systemfunktionen

Erstes Byte (Datentyp)	Zweites Byte Anzeigeformat	Format
9	Daten sind DATE	
	0	D#YYYY-MM-DD (z.B. D#2013-07-24)
	1	Als dezimaler Integer, 0..65535 = Anzahl Tage seit 01.01.1990
	2	Als hexadezimal W#16#0 .. W#16#FFFF
	3	Als hexadezimal 0x0000 .. 0xFFFF
10	Daten sind TIME_OF_DAY (TOD)	
	0	TOD#HH:MM:SD.MS (z.B. TOD#12:45:23.102) Wenn Stundenzahlwert > 23, dann als dezimaler Integer 0 ... 4294967295
	1	Als dezimaler Integer 0 ... 4294967295
	2	Als hexadezimal DW#16#00000000 .. DW#16#FFFFFFFF
	3	Als hexadezimal 0x00000000 .. 0xFFFFFFFF
11	Daten sind TIME	
	0	T#dhmsMS (T#24d20h31m23s647ms) d=Tag h=Stunde m=Minute s=Sekunde MS=Millisekunde Wenn eine Zeiteinheit = 0, dann wird sie nicht angezeigt z.B. T#20d2h10s130ms
	1	Als dezimaler Integer -2147483648 ... 2147483647
	2	Als hexadezimal DW#16#00000000 .. DW#16#FFFFFFFF
	3	Als hexadezimal 0x00000000 .. 0xFFFFFFFF
12	Daten sind S5TIME	
	0	S5T#hmsMS (S5T#2h46m30s) h=Stunde m=Minute s=Sekunde MS=Millisekunde
	1	Als dezimaler Integer 0 ... 65535
	2	Als hexadezimal W#16#0 .. W#16#FFFF
	3	Als hexadezimal 0x0000 .. 0xFFFF
13	Daten sind STRING	
	<ignored>	Nur die tatsächliche STRING-Länge wird kopiert
14	Daten sind DT	
	0	DT#YYYY-MM-DD-HH:MI:SS:MS YYYY – Jahr 1990 – 2089 MM – Monat DD – Tag HH – Stunde MI – Minute SS – Sekunde MS – Millisekunde z.B. DT#2013-07-24-13:05:12.340
	1	als 8 Byte-Stream z.B. 1307241305123404

Systemfunktionen

Bedeutung der Rückgabewerte

Rückgabewert (RETVAL)	BUSY	DONE	ERROR	Beschreibung
W#16#0000	0	1	0	Job ist erfolgreich abgeschlossen
W#16#7000	1	0	0	Job ist in Abarbeitung
W#16#7001	0	0	0	Verbindungsressource ist temporär gesperrt
W#16#7002	0	0	0	REQ Parameter ist FALSE
W#16#7003	0	0	1	Job nicht akzeptiert, weil vorheriger Job noch in Arbeit ist (Immer nur einen der SFC 207-209, 213 und 220-225 benutzen)
W#16#8000	0	0	1	Dateisystem ist nicht bereit (z.B. SD-Karte nicht gesteckt)
W#16#8001	0	0	1	Konnte nicht als Datei schreiben
W#16#8002	0	0	1	Ungültige Datentyp-ID-Nummer in CSV_STRUCT
W#16#8003	0	0	1	CSV_STRUCT – Größe muss eine gerade Zahl sein
W#16#822A	0	0	1	Ungültiges Format in FILENAME- Parameter
W#16#8xyy	0	0	1	Allgemeiner Fehlercode (S7-kompatibel)

S7-Programmbeispiel für die Verwendung des SFC207

Ziel: Schreiben der Daten aus dem DB207 in eine Archivdatei auf der Micro-SD-Karte, der Dateiname befindet sich auch in DB207.

Im Beispiel ist eine Konfiguration der CSV-Archivdatei mit folgender Struktur zu sehen:

1. Date&Time (1.Byte =14 / 2. Byte=0 / aufgeklappt)
2. String (1.Byte =13 / 2.Byte=0 / aufgeklappt)
3. Integer (1.Byte =5 / 2.Byte=0 / nicht aufgeklappt)
4. Integer (1.Byte =5 / 2.Byte=0 / nicht aufgeklappt)
5. Bool (1.Byte =1 / 2.Byte=0 / nicht aufgeklappt)

Name	Datentyp	Offset	Startwert	Remanenz	Sichtbar i...
Static					
ARCHIVE_NAME	String	0.0	'datenarchiv.csv'	☒	☒
ARCHIVE_CSV_FORMAT	Array[1..5] of Struct	256.0		☒	☒
ARCHIVE_CSV_FORMAT[1]	Struct	256.0		☒	☒
CSV_FELD_DATENTYP	Byte	256.0	14	☒	☒
CSV_FELD_ANZEIGE_FORMAT	Byte	257.0	16#0	☒	☒
ARCHIVE_CSV_FORMAT[2]	Struct	258.0		☒	☒
CSV_FELD_DATENTYP	Byte	258.0	13	☒	☒
CSV_FELD_ANZEIGE_FORMAT	Byte	259.0	16#0	☒	☒
ARCHIVE_CSV_FORMAT[3]	Struct	260.0		☒	☒
ARCHIVE_CSV_FORMAT[4]	Struct	262.0		☒	☒
ARCHIVE_CSV_FORMAT[5]	Struct	264.0		☒	☒
ARCHIVE_DATA	Array[1..100] of Struct	266.0		☒	☒
ARCHIVE_DATA[1]	Struct	266.0		☒	☒
Zeitstempel	Date_And_Time	266.0	DT# 1990-01-01-00:00...	☒	☒
Data 1	String[10]	274.0	''	☒	☒
Data 2	Int	286.0	0	☒	☒
Data 3	Int	288.0	0	☒	☒
Data 4	Bool	290.0	false	☒	☒
ARCHIVE_DATA[2]	Struct	292.0		☒	☒
ARCHIVE_DATA[3]	Struct	318.0		☒	☒
ARCHIVE_DATA[4]	Struct	344.0		☒	☒
ARCHIVE_DATA[5]	Struct	370.0		☒	☒
ARCHIVE_DATA[6]	Struct	396.0		☒	☒
ARCHIVE_DATA[7]	Struct	422.0		☒	☒

Systemfunktionen

```

CALL   SFC   207
REQ      :=M1501.3           // TRUE = Anforderung zur Datenarchivierung
FILENAME :=DB207.ARCHIVE_NAME // Archivdateiname
APPEND   :=TRUE              // Anfügen an das Dateiende
CSV      :=FALSE             // als Binärdatei
CSV_STRUCT:=DB207.ARCHIVE_CSV_FORMAT // CSV Feldformat
DATA     :=DB207.ARCHIVE_DATA // Daten zum Archivieren
RET_VAL  :=MW1000            // Rückgabe- / Ergebniswert
BUSY     :=M1701.3          // TRUE = Anfrage in Abarbeitung
DONE     :=M1002.0          // TRUE = Erfolgreich archiviert
ERROR    :=M1002.1          // TRUE = Fehler ist aufgetreten

U       M    1701.3          // Datenarchivieren ist in Bearbeitung
BEB
R       M    1501.3          // Setze Anforderungssignal zurück

O       M    1002.0          // Wenn erfolgreich erledigt,
ON      M    1002.1          // oder kein Fehler auftrat,
BEB      // dann Funktion beenden

L       MW   1000            // Fehler aufgetreten
// Fehlersuche
// ...

BEA      // Ende der Funktion

```


Systemfunktionen

Backup mit dem SFC 208 "BACKUP"

Mit dem SFC208 „BACKUP“ können alle Daten (S7-Anwenderprogramm, Prozessdaten, Visualisierung) auf die Micro-SD-Karte gesichert werden. In einer SPS mit gleicher Hardwarekonfiguration können diese Daten von der Micro-SD-Karte wieder zurückgelesen werden.

Während der Backups wird durch diesen SFC die SPS in den STOP-Zustand geschaltet und wechselt nach Neustart wieder in RUN, wenn Bit 3 des Parameters ATTRIBUTE gesetzt (=TRUE/1) ist.

Ein unberechtigtes Restore wird durch ein Passwort verhindert welches von einem PC auf die Micro-SD-Karte in das Backup-Verzeichnis geschrieben wird. Die gesetzten Schutzstufen werden bei dem Restore beibehalten.

Vorgehensweise

- Backupfunktion durch SFC208 in S7-Programm integrieren und auslösen (zugangsbeschränkte Taste mit Funktion oder Schlüsselschalter)
→ Die Backup-Datei *backup.bin* wird auf der Micro-SD-Karte im Verzeichnis „Backups“ erstellt.
- Micro-SD-Karte entnehmen und in neue SPS mit gleicher Hardwarekonfiguration stecken.
- **Optional:** Im S7-Programm definiertes Passwort als *backup.psw* auf Micro-SD-Karte im selben Verzeichnis wie Backupdatei speichern. (Siehe Bild nächste Seite)
- Während des Bootens der SPS hardwareseitiges Urlöschen ausführen, danach werden alle Backupdaten von der Micro-SD-Karte geladen und als Restore in die Steuerung geladen.

Parameter	Bezeichnung	Datentyp	Beschreibung
REQ	INPUT	BOOL	TRUE startet den Job
LADDR	INPUT	WORD	logische Adresse = 0 (für alle Panel-SPS) logische Adresse = 1...16 (Verbindungs-ID-Nummer zum Panel-HMI)
ATTRIBUTE	INPUT	BYTE	Attribute Bit0: 1= Backup MIT Aktualdaten der remanenten DBs 0= Backup OHNE Aktualdaten der remanenten DBs Bit1: 1= Backup MIT remanenten S7-Speicher 0= Backup OHNE remanenten S7-Speicher (remanente Merker, S7-Timer,Counter, Betriebsstunden) Bit2: 1= Backup MIT VisuStage Laufzeitdaten 0= Backup OHNE VisuStage-Laufzeitdaten (Meldungen, Trends, Anwender-PINs) Bit3: 1= NEUSTART der SPS nach Abschluss des Backups 0= SPS bleibt in STOP, kein Neustart Bit7: 1= Codieren der Backupdatei MIT Passwort 0= Codieren der Backupdatei OHNE Passwort
PASSWORD	INPUT	ANY	Zeiger zur Passwort-Decodierung. Datentyp muss BYTE oder CHAR sein. Passwort muss min. 5 max 16 Zeichen lang sein.
RETVAL	OUTPUT	WORD	Fehlercodes W#16#... 7000 = Backup ist in Abarbeitung 7001 = HMI-Verbindung ist in ungültigen Zustand (z.B. nicht verbunden) 7002 = Job wird nicht akzeptiert, solange vorheriger der vorherige Job noch in Abarbeitung ist 7003 = Verbindung belegt (kurzzeitig gesperrt) (Immer nur einen der SFC 207-209 und 221-225 benutzen) 7004 = REQ Parameter ist FALSE (Backup wird nicht akzeptiert) 8001 = Ungültige Verbindungs ID-Nummer 8003 = Micro-SD-Karte nicht gesteckt 842A = PASSWORD-Parameter ist nicht vom Typ BYTE oder CHAR 842B = PASSWORD-Parameterlänge ist ungültig 8xyy = Allgemeiner Fehlercode (S7-kompatibel)
ERROR	OUTPUT	BOOL	TRUE= Job fehlgeschlagen, Fehlercode in RETVAL



Das S7-Programm muss die Systemstabilität und Sicherheit gewährleisten, **bevor** der SFC208 aufgerufen wird.
Der Aufruf des SFC208 "BACKUP" stoppt die SPS und beginnt sofort mit der Sicherung der Daten auf die Micro-SD-Karte.
Während der Backups wird die SPS in den STOP-Zustand gehalten und wechselt nach Neustart wieder in RUN, wenn Bit 3 des Parameters ATTRIBUTE gesetzt (=TRUE/1) ist.

Hinweise:

- Das STOP-Ereignis wird im Diagnosepuffereintrag mit
EVENT ID W#16#497C "STOP triggered by integrated technology"
gekennzeichnet. Der Backupvorgang dauert bis zu 5 Sekunden.
- Das Backupergebnis wird im Diagnosepuffereintrag wie folgt gekennzeichnet
EVENT ID 16#BF00
OB 16#00
PK 16#FF
DatID1/2 Result
16#0002 = Successful
16#0001 = Backup failed
- Das Backupergebnis wird gekennzeichnet durch den OB100_STRT_INFO
Der Parameter des OB100 beim nächsten NEUSTART.
OB100_STRT_INFO (bit 16..23)
16#30 = Backup successful done
16#31 = Backup failed
- Wenn Bit 3 des ATTRIBUTE-Parameters auf TRUE gesetzt ist, dann wechselt die SPS wieder zu NEUSTART (RUN)
nachdem der Backupvorgang beendet ist, andererseits verbleibt die SPS im STOP-Zustand.

S7-Programmbeispiel für die Verwendung des SFC208

Ziel: Erstellen eines verschlüsselten Backups für S7-Programmes und für remanente Daten auf die Micro-SD-Karte.

```

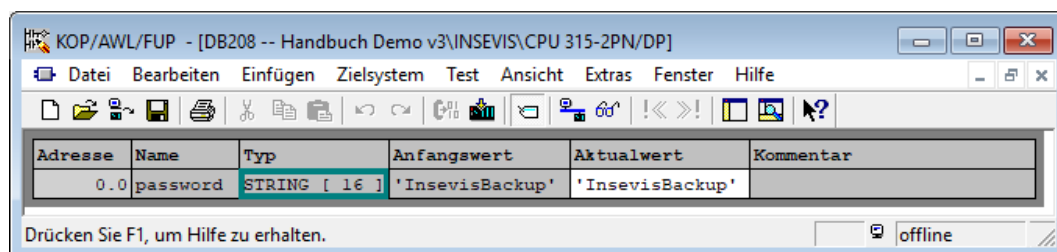
U    M    1501.5    // Backupanforderung wurde gesetzt
FP    M    1601.5    // Nur Start bei steigender Flanke von Backupanford.
SPBN    n000    //

CALL    SFC    208
REQ    :=TRUE    // TRUE = Backup starten
LADDR    :=MW2000    // VisuStage Verbindungs-ID-Nummer
// (Bei Panel-SPSen immer W#16#0 verwenden!)
ATTRIBUTE:=B#16#8F    // Entschlüsseln mit Passwort, Sichern aller Daten
PASSWORD :=P#DB208.DBX2.0 BYTE 13 // Zeiger auf Passwort
RET_VAL    :=MW1000    // Resultat / Rückgabewert
ERROR    :=M1002.1    // TRUE = Fehler ist aufgetreten

UN    M    1002.1    // Wenn kein Fehler auftrat,
BEB    // dann Funktion beenden
R    M    1501.5    // Rücksetzen der Backupanforderung

L    MW    1000    // Fehler aufgetreten
// Fehlersuche
// ...

n000: BEA    // Ende der Funktion
    
```



ACHTUNG:

Für ein erfolgreiches RESTORE muss das im S7-Programm definierte Passwort backup.psw auf Micro-SD-Karte auch im selben Verzeichnis wie die Backupdatei gespeichert werden.

Systemfunktionen

Rücklesen von Daten von Micro-SD-Karte mit SFC 209 "READFILE"

Diese Funktion wird genutzt, um auf der Micro-SD-Karte archivierte Daten in den S7-Datenbereich (z.B. in den DB) zurückzulesen.

Hinweise:

- Asynchrone Funktion (Ausführung zieht sich über mehrere SFC-Aufrufe hin)
- Diese Funktion darf nicht unterbrochen werden (z.B. durch Power OFF)

Parameter	Bezeichnung	Datentyp	Beschreibung
REQ	INPUT	BOOL	REQ = TRUE: Anforderung, aus der Datei zu lesen
FILENAME	INPUT	STRING	Dateiname, Länge muss kleiner als 60 Zeichen sein
OFFSET	INPUT	DWORD	Start Offset der Daten vom Dateianfang
LEN	INPUT	WORD	zu lesende Datengröße in Bytes, LEN = 0: Lesen der verfügbaren Daten vom Start offset bis Dateiende
RET_VAL	OUTPUT	WORD	Statuscode
BUSY	OUTPUT	BOOL	TRUE : Datei lesen in Bearbeitung
DONE	OUTPUT	BOOL	TRUE : Datei lesen erfolgreich beendet
ERROR	OUTPUT	BOOL	TRUE : Datei lesen mit Fehler beendet
DSTBLK	OUTPUT	ANY	Zielbaustein für das Schreiben der Daten
LEN_R	OUTPUT	WORD	Anzahl der gerade geschriebenen Daten nach DSTBLK in Bytes

Bedeutung der Rückgabewerte

Rückgabewert (RETVAL)	BUSY	DONE	ERROR	Beschreibung
W#16#0000	0	1	0	Erfolgreich gelesen
W#16#0080	0	1	0	Lesen erledigt, verfügbare Daten sind kleiner als angefordert.
W#16#0081	0	1	0	Zielbereich (DSTBLK) ist größer als angeforderte Daten (LEN), nur angeforderte Daten wurden kopiert
W#16#8081	0	1	0	Angeforderte Daten (LEN) sind größer als Zielbereich (DSTBLK), nur Daten kopiert, welche in den Zielbereich passen.
W#16#7000	1	0	0	Lesen ist in Bearbeitung
W#16#7001	0	0	0	Interne Ressource temporär besetzt
W#16#7002	0	0	0	REQ = FALSE, Kein Lesen gestartet
W#16#7003	0	0	1	Interne Ressource temporär besetzt durch andere Anforderung(en) (Immer nur einen der SFC 207-209, 213 und 220-225 benutzen)
W#16#8000	0	0	1	Micro-SD -Karte nicht gesteckt
W#16#8001	0	0	1	Datei nicht gefunden, oder konnte Datei nicht lesen
W#16#8002	0	0	1	Datei ist leer
W#16#8003	0	0	1	OFFSET Parameter ist größer als Dateigröße
W#16#822A	0	0	1	FILENAME Parameter ist ungültig.
W#16#8xyz	0	0	1	Allgemeiner Fehlercode (S7 kompatibel)

Systemfunktionen

S7-Programmbeispiel für die Verwendung des SFC209

Ziel: Einlesen der mit SFC207 archivierten Daten aus einem DB von der Mikro-SD-Karte zurück in den S7-Datenbereich

```
CALL SFC 209
REQ      :=M1501.6           // TRUE = Anforderung zum Datei-Einlesen
FILENAME:=DB209.ARCHIVE_NAME // Zeiger auf Dateiname
OFFSET   :=DW#16#0           // Offset vom Dateianfang
LEN       :=W#16#0            // Lesen aller Daten, die zu DST_BLK passen
RET_VAL  :=MW1000             // Resultat / Rückgabewert
BUSY      :=M1701.6           // TRUE = Anforderung in Bearbeitung
DONE      :=M1002.0           // TRUE = Erfolgreich gelesen
ERROR     :=M1002.1           // TRUE = Fehler aufgetreten
DSTBLK    :=DB209.ARCHIVE_DATA // Zieldatenbereich, wohin die Daten geschrieben werden sollen
LEN_R     :=MW1004            // Anzahl von zu schreibenden Bytes nach DSTBLK

U      M      1701.6           // Archivierung der Daten in Bearbeitung
BEB                               // dann Beenden der Funktion
R      M      1501.6           // Rücksetzen des Anforderungssignals

O      M      1002.0           // Wenn erfolgreich erledigt,
ON     M      1002.1           // oder kein Fehler auftrat,
BEB                               // dann Funktion beenden

L      MW     1000             // Fehler aufgetreten
                                   // Fehlersuche
                                   // ...

BEA                               // Ende der Funktion
```



Das FAT16/32 Filesystem auf der Micro-SD-Karte **erlaubt** einige Zeichen **nicht** in Datei- bzw. Verzeichnisnamen: Diese sind \ / : * ? " < > |

DB kopieren vom Arbeits- in den Ladespeicher mit SFC 210 "COPY_DBL"

Diese Funktion wird genutzt, um Datenbausteine (DBs) vom batteriegepufferten Arbeitsspeicher (RAM) in den Ladespeicher (ROM) zu kopieren.

Hinweise:

- Asynchrone Funktion (Ausführung zieht sich über mehrere SFC- Aufrufe hin)
- Diese Funktion darf nicht unterbrochen werden (z.B. durch Power OFF, oder PG speicherrelevante Funktionen wie „Clear all“, „Block delete“, etc.).
- Datenbausteine, die mit „Non-Retain“ bezeichnet sind, werden ebenfalls in den Ladespeicher kopiert (Aktualdaten)
- Wenn Parameter LOW_NR = HIGH_NR, dann wird nur ein Datenbaustein mit LOW_NR kopiert.

Parameter	Bezeichnung	Datentyp	Beschreibung
REQ	INPUT	BOOL	REQ = TRUE : Anforderung, die Datenbausteine vom Arbeits- in den Ladespeicher zu kopieren
LOW_NR	INPUT	WORD	Niedrigste Datenblocknummer. Muss > 0 sein.
HIGH_NR	INPUT	WORD	Höchste Datenblocknummer. Muss > 0 sein.
RET_VAL	OUTPUT	WORD	Statuscode
BUSY	OUTPUT	BOOL	TRUE : Kopieren in Bearbeitung

Systemfunktionen

Bedeutung der Rückgabewerte

Rückgabewert (RETVAL)	BUSY	Beschreibung
W#16#0000	0	Erfolgreich kopiert
W#16#7000	0	Erster Aufruf mit REQ=0, Kopieren wurde nicht gestartet
W#16#7001	1	Erster Aufruf mit REQ=0, Kopieren wurde gestartet
W#16#7002	1	Zwischenaufruf (REQ irrelevant): Kopieren in Bearbeitung
W#16#8001	0	Andere speicherverändernde Funktion ist aktiv (z.B. PG Block download, löschen, o.ä.) Kopieren wurde nicht gestartet.
W#16#8002	0	Ungültige Datenblocknummer (HIGH_NR ist kleiner als LOW_NR)
W#16#822A	0	LOW_NR ist ungültig.
W#16#832A	0	HIGH_NR ist ungültig.
W#16#8xyz	0	Allgemeiner Fehlercode (S7 kompatibel)

S7-Programmbeispiel für die Verwendung des SFC210

Ziel: Kopieren des DB209 in den Ladespeicher (Aktuelle Werte im Ladespeicher werden dadurch überschrieben).

UN	M	1502.0	// Wenn keine Anforderung gesetzt,
BEB			// dann Funktion beenden
CALL	SFC	210	
REQ		:=M1502.0	// TRUE = Anforderung zum Kopieren der DB(s) in den
			// Ladespeicher
LOW_NR		:=W#16#D1	// Nummer des Datenbaustein 209 (dez)
HIGH_NR		:=W#16#D1	// Nur ein DB
RET_VAL		:=MW1000	// Resultat / Rückgabewert
BUSY		:=M1702.0	// TRUE = Anforderung in Bearbeitung
U	M	1702.0	// Kopieranforderung in Bearbeitung
BEB			// dann Funktion beenden
R	M	1502.0	// Rücksetzen des Anforderungssignals
L	MW	1000	// Resultat / Rückgabewert
L	0		// Wenn kein Fehler,
==I			
BEB			// dann Funktion beenden
L	MW	1000	// Fehler aufgetreten
			// Fehlersuche
			// ...
BEA			

Systemfunktionen

DB kopieren vom Lade- in den Arbeitsspeicher mit SFC 211 "COPY_DBW"

Diese Funktion wird genutzt, um Datenbausteine (DBs) vom Ladespeicher (ROM) in den batteriegepufferten Arbeitsspeicher (RAM) zu kopieren.

- Hinweise:**
- Asynchrone Funktion (Ausführung zieht sich über mehrere SFC- Aufrufe hin)
 - Diese Funktion darf nicht unterbrochen werden (z.B. durch Power OFF, oder PG speicherrelevante Funktionen wie „Clear all“, „Block delete“, etc.).
 - Wenn Parameter LOW_NR = HIGH_NR, dann wird nur ein Datenbaustein mit LOW_NR kopiert.

Parameter	Bezeichnung	Datentyp	Beschreibung
REQ	INPUT	BOOL	REQ = TRUE: Anforderung, die Datenbausteine vom Lade- in den Arbeitsspeicher zu kopieren
LOW_NR	INPUT	WORD	Niedrigste Datenblocknummer. Muss > 0 sein.
HIGH_NR	INPUT	WORD	Höchste Datenblocknummer. Muss > 0 sein.
RET_VAL	OUTPUT	WORD	Statuscode
BUSY	OUTPUT	BOOL	TRUE : Kopieren in Bearbeitung

Bedeutung der Rückgabewerte

Rückgabewert (RETVAL)	BUSY	Beschreibung
W#16#0000	0	Erfolgreich kopiert
W#16#7000	0	Erster Aufruf mit REQ=0, Kopieren wurde nicht gestartet
W#16#7001	1	Erster Aufruf mit REQ=0, Kopieren wurde gestartet
W#16#7002	1	Zwischenaufruf (REQ irrelevant): Kopieren in Bearbeitung
W#16#8001	0	Andere speicherverändernde Funktion ist aktiv (z.B. PG Block download, löschen, o.ä.) Kopieren wurde nicht gestartet.
W#16#8002	0	Ungültige Datenblocknummer (HIGH_NR ist kleiner als LOW_NR)
W#16#822A	0	LOW_NR ist ungültig.
W#16#832A	0	HIGH_NR ist ungültig.
W#16#8xyz	0	Allgemeiner Fehlercode (S7-kompatibel)

Systemfunktionen

Programmierbeispiel für die Verwendung des SFC211

Ziel: Kopieren des DB209 in den Arbeitsspeicher (Aktuelle Werte im Arbeitsspeicher, die zuletzt mit SFB210 oder SFC25 gesichert oder vom PG übertragen wurden, werden dadurch überschrieben).

UN	M	1502.1	// Wenn keine Anforderung gesetzt,
BEB			// dann Funktion beenden
CALL	SFC	211	
REQ		:=M1502.1	// TRUE = Anforderung zum Kopieren der DB(s) in den
			// Arbeitsspeicher
LOW_NR		:=W#16#D1	// Nummer des Datenbaustein 209 (dez)
HIGH_NR		:=W#16#D1	// Nur ein DB
RET_VAL		:=MW1000	// Resultat / Rückgabewert
BUSY		:=M1702.1	// TRUE = Anforderung in Bearbeitung
U	M	1702.1	// Kopieranforderung in Bearbeitung
BEB			// dann Funktion beenden
R	M	1502.1	// Rücksetzen des Anforderungssignals
L	MW	1000	// Resultat / Rückgabewert
L		0	// Wenn kein Fehler,
==I			
BEB			// dann Funktion beenden
L	MW	1000	// Fehler aufgetreten
			// Fehlersuche
			// ...
BEA			

Systemfunktionen

Daten lesen aus csv-formatierter Archivdatei (SD) mit SFC 213 "READ_CSV"

Diese Funktion kopiert die Daten aus der csv-formatierten Archivdatei von der Micro-SD-Karte in den S7-Bereich (z.B. Datenbaustein). Dieser SFC ist asynchron, d.h., seine Ausführung erstreckt sich über mehrere SFC-Aufrufe hinweg. Nur ein Auftrag kann zur Abarbeitung anstehen.

Parameter	Bezeichnung	Datentyp	Beschreibung
REQ	INPUT	BOOL	REQ = TRUE: Anfrage zum Lesen aus der Datei
FILENAME	INPUT	STRING	Dateiname, Länge muss kleiner als 60 Zeichen sein
CSV_STRUCT	INPUT	ANY	CSV-Datensatzstruktur. → siehe SFC 207 "ARCHIVE" -Doku
RET_VAL	OUTPUT	WORD	Statuscode
BUSY	OUTPUT	BOOL	TRUE : Datei Lesen ist in Bearbeitung
DONE	OUTPUT	BOOL	TRUE : Datei Lesen erfolgreich beendet
ERROR	OUTPUT	BOOL	TRUE : Datei Lesen mit Fehler beendet
LINE	OUTPUT	INT	Wenn ERROR=TRUE: Anzeigen der Zeilennummer mit Konvertierungsfehler
FIELD	OUTPUT	INT	Wenn ERROR=TRUE: Anzeige der Feldnummer mit Konvertierungsfehler
DATA	INPUT	ANY	Zeiger auf S7-Prozessbereich

Bedeutung der Rückgabewerte

Rückgabewert (RETVAL)	BUSY	DONE	ERROR	Beschreibung
W#16#0000	0	1	0	Erfolgreich gelesen
W#16#7000	1	0	0	Lesen in Bearbeitung
W#16#7001	0	0	0	Interne Ressource temporär benutzt
W#16#7002	0	0	0	REQ = FALSE, Kein Lesen gestartet
W#16#7003	0	0	1	Interne Ressource temporär benutzt andere Anfrage (Immer nur einen der SFC 207-209, 213 und 220-225 benutzen)
W#16#8000	0	0	1	Micro-SD-Karte ist nicht gesteckt
W#16#8001	0	0	1	Datei existiert nicht in der Micro-SD-Karte
W#16#8002	0	0	1	Ungültiger Datentyp in CSV_STRUCT -Parameter
W#16#8003	0	0	1	Ungültige Länge in CSV_STRUCT -Parameter
W#16#8004	0	0	1	S7-Prozessdatenbereich muss von geraden Offset starten (DATA -Param.)
W#16#8005	0	0	1	Datei ist leer
W#16#8006	0	0	1	Fehler beim Dateilesen aufgetreten
W#16#8007	0	0	1	Ungültige Zeile / Datensatztrenner. Es muss CRLF sein
W#16#8008	0	0	1	Ungültiges Zeichen. Es dürfen ausschließlich ASCII-Zeichen sein.
W#16#8009	0	0	1	Konnte die Felddaten nicht konvertieren (z.B. Syntaxfehler ...)
W#16#800A	0	0	1	Zu große Datei. Nicht genug Systemressourcen, um Datei einzulesen.
W#16#822A	0	0	1	FILENAME -Parameter ist ungültig.
W#16#8xyz	0	0	1	Allgemeiner Fehlercode (S7-kompatibel)



Das FAT16/32 Filesystem auf der Micro-SD-Karte erlaubt einige Zeichen nicht in Datei- bzw.

Verzeichnisnamen: Diese sind \ / : * ? " < > |

Solche Zeichen werden ignoriert.

Systemfunktionen

Hinweise:

- Das Feldbegrenzerzeichen MUSS „;“ (Semikolon, ASCII-Zeichen 59) sein.
- Der Dezimalpunkt MUSS „.“ (Punkt, point, ASCII-Zeichen 46) sein.
- Datensatz MUSS durch „Carriage Return Line Feed -Zeichen“ (CRLF, ASCII-Zeichen 13, 10) getrennt werden.
- Der Prozessdaten (DATA -Parameter) -Offset MUSS von gerader Adresse starten.
- Die einzelne Datensatzgröße in Bytes MUSS gerade sein.
- Bei Leerfeldern (;), wird der entsprechende Prozessdatenbereich nicht beschrieben (keine Daten).
- Wenn ein Feld in einem Datensatz kleiner ist, als in CSV_STRUCT „field count“ festgelegt, wird das Konvertieren mit dem nächsten Datensatz weitergeführt (wie bei einem Zeilenumbruch) und die Daten in den Datenbereich des nächsten Datensatzes geschrieben.
- Wenn mehr Felder in einem Datensatz vorhanden sind, als in CSV_STRUCT „field count“ festgelegt, werden die überzähligen Datenfelder ignoriert.
- Wenn die Prozessdatengröße für den Zielbereich (DATA) kleiner ist als die konvertierte Archivdatei, werden die Datenfelder solange konvertiert, bis die Grenze erreicht ist und der Rest verworfen.
- Wenn die Prozessdatengröße für den Zielbereich (DATA) größer ist als die konvertierte Archivdatei, werden alle konvertierten Feld für Feld geschrieben und der Rest des Zielbereiches wird nicht überschrieben.

S7-Programmbeispiel für die Verwendung des SFC213

Ziel: Einlesen und Konvertieren Daten aus der csv-Datei in den DB213

```
CALL SFC 213
  REQ          :=M1501.7           // TRUE = Anfrage für CSV Datei lesen
  FILENAME     :=DB207.ARCHIVE_NAME_CSV // CSV Archivname
  CSV_STRUCT   :=DB207.ARCHIVE_CSV_FORMAT // CSV Feldformat
  RET_VAL      :=MW1000            // Rückgabewert
  BUSY         :=M1701.7           // TRUE = Anfrage in Bearbeitung
  DONE         :=M1002.0           // TRUE = Erfolgreich gelesen
  ERROR        :=M1002.1           // TRUE = Fehler aufgetreten
  LINE         :=MW1006            // Im Fehlerfall, die Zeilen-Nr. welche
                                   // den Konvertierfehler verursacht
  FIELD        :=MW1008            // Im Fehlerfall, die Feldnumr., welche
                                   // den Konvertierfehler verursacht
  DATA        :=DB213.ARCHIVE_DATA // Zieldatenbereich, welche Daten
                                   // werden geschrieben
U  M           1701.7              // Daten lesen in Bearbeitung
BEB
R  M           1501.7              // Setzt Anfragesignal zurück

O  M           1002.0              // Wenn erfolgreich erledigt,
ON M           1002.1              // oder kein Fehler, dann
BEB                                           // Funktionsende

L  MW          1000                // Fehler aufgetreten
                                   // Fehlersuche
                                   // ...

BEA
```

Systemfunktionen

Übersicht über SFCs für ein DateiManagementsystem

INSEVIS stellt SFCs bereit, mit denen sich Dateien und Ordner auf der Micro-SD-Karte erzeugen, kopieren, umbenennen und löschen lassen und womit man Informationen zu Änderungen an den Dateien und Ordnern erhält.



Dabei wird jeweils hierarchisch in Ebenen unterhalb des Ordners UserData\ operiert.

Erstellen von Ordnern und Dateien

Ordner: SFC 223 „CREATE“ erzeugt einen Ordner
Datei: SFC 207 „ARCHIVE“ erzeugt eine Datei

Auslesen von Ordnern und Dateien

Ordner: können nicht ausgelesen werden
Datei: SFC 209 „READFILE“ Dateiinhalte in SPS zurücklesen
 SFC 213 „READ_CSV“ Dateiinhalte in csv-Format in SPS zurücklesen

Kopieren von Ordnern und Dateien

Ordner: können (noch) nicht kopiert werden
Datei: SFC 222 „COPY“ kopiert eine Datei

Umbenennen von Ordnern und Dateien

Ordner: SFC 220 „RENAME“ benennt einen Ordner um
Datei: SFC 220 „RENAME“ benennt auch eine Datei um

Informationen zu Ordnern und Dateien

Ordner: SFC 224 „INFO“ zeigt die Einträge in einem Ordner an
Datei: SFC 224 „INFO“ zeigt die Dateigröße an

Informationen zu Dateien

Ordner: SFC 225 ist nicht relevant für Ordner
Datei: SFC 225 „FILEINFO“ zeigt die Namen und Größe der Dateien in einem Ordner an

Löschen von Ordnern und Dateien

Ordner: SFC 221 „DELETE“ löscht einen Ordner
Datei: SFC 221 „DELETE“ löscht auch eine Datei

Systemfunktionen

Erstellen von Ordnern auf Micro-SD mit SFC 223 "CREATE"

Der SFC222 "CREATE" wird zum Erstellen eines Ordners auf der Micro-SD-Karte genutzt (bezogen auf den Ordner UserData), d.h. alle (hierarchischen) Ebenen darunter). Dieser SFC ist asynchron, seine Ausführung zieht sich über mehrere SFC- Aufrufe hin.

Parameter	Bezeichnung	Datentyp	Beschreibung
REQ	INPUT	BOOL	TRUE Parameter zuweisen und Job starten
DIRNAME	INPUT	STRING	Name des zu erstellenden Ordners
RET_VAL	OUTPUT	WORD	Status / Fehlercode
BUSY	OUTPUT	BOOL	TRUE Job wird bearbeitet
DONE	OUTPUT	BOOL	TRUE Job wurde erfolgreich abgeschlossen
ERROR	OUTPUT	BOOL	TRUE Job ist fehlgeschlagen

Bedeutung der Rückgabewerte

RET_VAL	BUSY	DONE	ERROR	Beschreibung
W#16#0000	0	1	0	Job wurde erfolgreich abgeschlossen
W#16#7000	1	0	0	Job wird bearbeitet
W#16#7001	0	0	0	Job-Ressource ist temporär verriegelt

Bedeutung der Rückgabewerte

RET_VAL	BUSY	DONE	ERROR	Beschreibung
W#16#7002	0	0	0	REQ-Parameter ist FALSE
W#16#7003	0	0	1	Job wurde nicht akzeptiert, solange der vorhergehende Job noch in Abarbeitung ist. (Immer nur einen der SFC 207-209, 213 und 220-225 benutzen)
W#16#8000	0	0	1	Dateisystem ist noch nicht bereit (z.B, Micro-SD-Karte nicht gesteckt)
W#16#8001	0	0	1	Name ist ungültig (DIRNAME parameter)
W#16#8002	0	0	1	Name ist zu lang (DIRNAME parameter)
W#16#8003	0	0	1	Eintrag mit einem Namen existiert bereits (DIRNAME parameter)
W#16#8004	0	0	1	Konnte keinen Ordner erstellen
W#16#822A	0	0	1	Ungültiges Format des DIRNAME-Parameters
W#16#8xyy	0	0	1	Allgemeiner Fehlercode (S7-kompatibel)

S7-Anwenderprogramm zum Erstellen eines Ordners

```

CALL SFC 223
  REQ      :=M1800.0
  DIRNAME:=DB223.DIRNAME
  RET_VAL:=MW1000
  BUSY     :=M1800.1
  DONE     :=M1800.2
  ERROR    :=M1800.3

O      M      1800.2           // Job ausgeführt
O      M      1800.3           // oder fehlgeschlagen
R      M      1800.0           // Reset Request Signal

UN     M      1800.3
BEB                                         // Kein Fehler aufgetreten

L      MW 1000                       // Fehler aufgetreten
// error evaluation
// ...
BEA                                         // Ende der Funktion

```

Systemfunktionen

Kopieren von Dateien auf Micro-SD mit SFC 222 "COPY"

Der SFC222 "COPY" wird zum Kopieren einer Datei auf der Micro-SD-Karte genutzt (bezogen auf den Ordner UserData), d.h. alle (hierarchischen) Ebenen darunter). Dieser SFC ist asynchron, seine Ausführung zieht sich über mehrere SFC- Aufrufe hin.

Parameter	Bezeichnung	Datentyp	Beschreibung
REQ	INPUT	BOOL	TRUE Parameter zuweisen und Job starten
OVERWR	INPUT	BOOL	TRUE Kopiere Datei, auch wenn DSTNAME existiert
SRCNAME	INPUT	STRING	Name der zu kopierenden Datei
DSTNAME	INPUT	STRING	Zielname (-pfad)
RET_VAL	OUTPUT	WORD	Status / Fehlercode
BUSY	OUTPUT	BOOL	TRUE Job wird bearbeitet
DONE	OUTPUT	BOOL	TRUE Job wurde erfolgreich abgeschlossen
ERROR	OUTPUT	BOOL	TRUE Job ist fehlgeschlagen

Bedeutung der Rückgabewerte

RET_VAL	BUSY	DONE	ERROR	Beschreibung
W#16#0000	0	1	0	Job wurde erfolgreich abgeschlossen
W#16#7000	1	0	0	Job wird bearbeitet
W#16#7001	0	0	0	Job-Ressource ist temporär verriegelt
W#16#7002	0	0	0	REQ-Parameter ist FALSE
W#16#7003	0	0	1	Job wurde nicht akzeptiert, solange der vorheriger Job noch in Arbeit ist. (Immer nur einen der SFC 207-209, 213 und 220-225 benutzen)
W#16#8000	0	0	1	Dateisystem ist noch nicht bereit (z.B. Micro-SD-Karte nicht gesteckt)
W#16#8001	0	0	1	Datei mit Zielname existiert bereits (DSTNAME-Parameter)
W#16#8002	0	0	1	Datei mit Quellname nicht gefunden (SRCNAME-Parameter)
W#16#8003	0	0	1	Name ist zu lang (SRCNAME, DSTNAME-Parameter)
W#16#8004	0	0	1	Name ist ungültig (SRCNAME, DSTNAME-Parameter)
W#16#8005	0	0	1	Keine Datei (SRCNAME-Parameter)
W#16#8006	0	0	1	Konnte Datei nicht kopieren
W#16#832A	0	0	1	Ungültiges Format im SRCNAME-Parameter
W#16#842A	0	0	1	Ungültiges Format im DSTNAME-Parameter
W#16#8xyy	0	0	1	Allgemeiner Fehlercode (S7-kompatibel)

S7-Anwenderprogramm zum Kopieren einer Datei

```

CALL SFC 222
REQ      :=M1800.0
OVERWR   :=TRUE           // Überschreibe Datei, wenn eine existiert
SRCNAME:=DB222.SRCNAME
DSTNAME:=DB222.DSTNAME
RET_VAL:=MW1000
BUSY     :=M1800.1
DONE     :=M1800.2
ERROR    :=M1800.3

O      M    1800.2           // Job ausgeführt
O      M    1800.3           // oder fehlgeschlagen
R      M    1800.0           // Reset Request Signal

UN     M    1800.3
BEB                               // Kein Fehler aufgetreten

L      MW 1000                // Fehler aufgetreten
// error evaluation
// ...
BEA                               // Ende der Funktion

```

Systemfunktionen

Umbenennen von Ordnern/Dateien auf Micro-SD mit SFC 220 "RENAME"

Der SFC220 "RENAME" wird zum Umbenennen von Dateien / Ordner auf der Micro-SD-Karte genutzt (bezogen auf den Ordner UserData\, d.h. alle (hierarchischen) Ebenen darunter). Dieser SFC ist asynchron, seine Ausführung zieht sich über mehrere SFC- Aufrufe hin.

Parameter	Bezeichnung	Datentyp	Beschreibung
REQ	INPUT	BOOL	TRUE Parameter zuweisen und Job starten
OLDNAME	INPUT	STRING	Name der umzubenennenden Datei / des umzubenennenden Ordners
NEWNAME	INPUT	STRING	Ein neuer Name
RET_VAL	OUTPUT	WORD	Status / Fehlercode
BUSY	OUTPUT	BOOL	TRUE Job wird bearbeitet
DONE	OUTPUT	BOOL	TRUE Job wurde erfolgreich abgeschlossen
ERROR	OUTPUT	BOOL	TRUE Job ist fehlgeschlagen

Bedeutung der Rückgabewerte

RET_VAL	BUSY	DONE	ERROR	Beschreibung
W#16#0000	0	1	0	Job wurde erfolgreich abgeschlossen
W#16#7000	1	0	0	Job wird bearbeitet
W#16#7001	0	0	0	Job-Ressource ist temporär verriegelt
W#16#7002	0	0	0	REQ-Parameter ist FALSE
W#16#7003	0	0	1	Job wurde nicht akzeptiert, solange der vorhergehende Job noch in Abarbeitung ist. (Immer nur einen der SFC 207-209, 213 und 220-225 benutzen)
W#16#8000	0	0	1	Dateisystem ist noch nicht bereit (z.B. Micro-SD-Karte nicht gesteckt)
W#16#8001	0	0	1	Datei/Ordner mit neuem Namen nicht gefunden (NEWNAME-Parameter)
W#16#8002	0	0	1	Datei/Ordner mit altem Namen nicht gefunden (OLDNAME-Parameter)
W#16#8003	0	0	1	Name ist zu lang (OLDNAME, NEWNAME-Parameter)
W#16#8004	0	0	1	Name ist ungültig (OLDNAME, NEWNAME-Parameter)
W#16#8005	0	0	1	Konnte Datei / Ordner nicht umbenennen
W#16#822A	0	0	1	Ungültiges Format im OLDNAME-Parameter
W#16#832A	0	0	1	Ungültiges Format im NEWNAME-Parameter
W#16#8xyy	0	0	1	Allgemeiner Fehlercode (S7-kompatibel)

S7-Anwenderprogramm zum Umbenennen einer Datei

```

CALL SFC 220
REQ      :=M1800.0
OLDNAME:=DB220.OLDNAME
NEWNAME:=DB220.NEWNAME
RET_VAL:=MW1000           // Statuscode
BUSY     :=M1800.1
DONE     :=M1800.2
ERROR    :=M1800.3

O      M    1800.2           // Job ausgeführt
O      M    1800.3           // oder fehlgeschlagen
R      M    1800.0           // Reset Request Signal

UN     M    1800.3
BEB                                // Kein Fehler aufgetreten

L      MW 1000                // Fehler aufgetreten
// error evaluation
// ...
BEA                                // Ende der Funktion

```

Systemfunktionen

Informationen zu Ordnern/ Dateien auf Micro-SD mit SFC 224 "INFO"

Der SFC224 "INFO" wird zur Abfrage der Dateigröße oder Anzahl der Einträge in einen Ordner auf der MicroSD-Karte genutzt (bezogen auf den Ordner UserData), d.h. alle (hierarchischen) Ebenen darunter). Dieser SFC ist asynchron, seine Ausführung zieht sich über mehrere SFC- Aufrufe hin.

Parameter	Bezeichnung	Datentyp	Beschreibung
REQ	INPUT	BOOL	TRUE Parameter zuweisen und Job starten
DIRECTORY	INPUT	BOOL	TRUE Anzahl der Einträge im Ordner abfragen
ENTRYNAME	INPUT	STRING	Name der Datei / des Ordners
RET_VAL	OUTPUT	WORD	Status / Fehlercode
BUSY	OUTPUT	BOOL	TRUE Job wird bearbeitet
DONE	OUTPUT	BOOL	TRUE Job wurde erfolgreich abgeschlossen
ERROR	OUTPUT	BOOL	TRUE Job ist fehlgeschlagen
SIZE	OUTPUT	DWORD	Dateigröße / Anzahl von Einträgen in dem Ordner

Bedeutung der Rückgabewerte

Rückgabewert (RETVAL)	BUSY	DONE	ERROR	Beschreibung
W#16#0000	0	1	0	Job wurde erfolgreich abgeschlossen
W#16#7000	1	0	0	Job wird bearbeitet
W#16#7001	0	0	0	Job-Ressource ist temporär verriegelt
W#16#7002	0	0	0	REQ-Parameter ist FALSE
W#16#7003	0	0	1	Job wurde nicht akzeptiert, solange der vorhergehende Job noch in Abarbeitung ist. (Immer nur einen der SFC 207-209, 213 und 220-225 benutzen)
W#16#8000	0	0	1	Dateisystem ist noch nicht bereit (z.B, Micro-SD-Karte nicht gesteckt)
W#16#8001	0	0	1	Name ist ungültig (ENTRYNAME-Parameter)
W#16#8002	0	0	1	Name ist zu lang (ENTRYNAME-Parameter)
W#16#8003	0	0	1	Datei / Ordner nicht gefunden
W#16#8004	0	0	1	Konnte Informationen zum Eintrag nicht lesen (z.B. Datei- statt Ordnername erwartet, o.ä.)
W#16#832A	0	0	1	Ungültiges Format des ENTRYNAME-Parameters
W#16#8xyy	0	0	1	Allgemeiner Fehlercode (S7-kompatibel)

S7-Anwenderprogramm zur Abfrage der Dateigröße

```

CALL SFC 224
REQ      :=M1800.0
DIRECTORY:=FALSE           // Erhalte Dateigröße
ENTRYNAME:=DB224.FILENAME
RET_VAL  :=MW1000
BUSY     :=M1800.1
DONE     :=M1800.2
ERROR    :=M1800.3
SIZE     :=MD1004          // Dateigröße in Bytes

O   M   1800.2             // Job ausgeführt
O   M   1800.3             // oder fehlgeschlagen
R   M   1800.0             // reset request signal

UN   M   1800.3
BEB                                     // Kein Fehler aufgetreten

L   MW 1000                 // Fehler aufgetreten
// error evaluation
// ...
BEA                                     // Ende der Funktion

```

Systemfunktionen

Abfrage von Dateigröße und -name auf Micro-SD mit SFC 225 "FILEINFO"

Der SFC225 "FILEINFO" wird benutzt, um den Dateiname und die Dateigröße nullbasiert aus dem Ordner UserData\ der Micro-SD-Karte abzufragen. Dieser SFC ist asynchron, seine Ausführung zieht sich über mehrere SFC- Aufrufe hin.

Parameter	Bezeichnung	Datentyp	Beschreibung
REQ	INPUT	BOOL	TRUE Parameter zuweisen und Job starten
DIRECTORY	INPUT	STRING	(Unter-) Verzeichnisname für die zu suchende Datei
INDEX	INPUT	WORD	Nullbasierter Datei-Index im DIRECTORY
RET_VAL	OUTPUT	WORD	Status / Fehlercode
BUSY	OUTPUT	BOOL	TRUE Job wird bearbeitet
DONE	OUTPUT	BOOL	TRUE Job wurde erfolgreich abgeschlossen
ERROR	OUTPUT	BOOL	TRUE Job ist fehlgeschlagen
FILE_NAME	OUTPUT	STRING	Dateiname
FILE_SIZE	OUTPUT	DWORD	Dateigröße (in Byte)

Bedeutung der Rückgabewerte

RET_VAL	BUSY	DONE	ERROR	Beschreibung
W#16#0000	0	1	0	Job wurde erfolgreich abgeschlossen
W#16#7000	1	0	0	Job wird bearbeitet
W#16#7001	0	0	0	Job-Ressource ist temporär verriegelt
W#16#7002	0	0	0	REQ-Parameter ist FALSE
W#16#7003	0	0	1	Job wurde nicht akzeptiert, solange der vorherige Job noch in Arbeit ist. (Immer nur einen der SFC 207-209, 213 und 220-225 benutzen)
W#16#8000	0	0	1	Dateisystem ist noch nicht bereit (z.B, Micro-SD-Karte nicht gesteckt)
W#16#8001	0	0	1	Konnte das Verzeichnis nicht öffnen (evtl. existiert es nicht)
W#16#8002	0	0	1	INDEX-Parameter ist größer oder gleich wie die Anzahl verfügbarer Dateien
W#16#8003	0	0	1	Konnte Verzeichnis nicht lesen
W#16#812A	0	0	1	Ungültiges Format im DIRECTORY-Parameter
W#16#882A	0	0	1	Ungültiges Format im FILE_NAME-Parameter
W#16#8xyy	0	0	1	Allgemeiner Fehlercode (S7-kompatibel)

S7-Anwenderprogramm zum Erhalten einer Dateiinformation

```

UN    M    1900.0
BEB                                     // Job is not requested

CALL  SFC  225
REQ      :=M1900.0
DIRECTORY:=DB225.DIRECTORY           // Directory to be searched in
INDEX    :=MW1902                     // File index in directory (zero based)
RET_VAL  :=MW1000
BUSY     :=M1900.1
DONE     :=M1900.2
ERROR    :=M1900.3
FILE_NAME:=DB225.FILENAME            // file name
FILE_SIZE:=MD1904                    // file size in bytes

O      M    1900.2                     // Job done
O      M    1900.3                     // or failed
R      M    1900.0                     // reset request signal

UN    M    1900.3
BEB                                     // No error occurred

L      MW  1000                        // error occurred
// error evaluation
// ...
BEA                                     // end of function

```


Systemfunktionen

Löschen von Ordnern/Dateien auf Micro-SD mit SFC 221 "DELETE"

Der SFC221 "DELETE" wird zum Löschen von Dateien / Ordner auf der Micro-SD-Karte genutzt (bezogen auf den Ordner UserData\, d.h. alle (hierarchischen) Ebenen darunter). Dieser SFC ist asynchron, seine Ausführung zieht sich über mehrere SFC- Aufrufe hin.

Parameter	Bezeichnung	Datentyp	Beschreibung
REQ	INPUT	BOOL	TRUE Parameter zuweisen und Job starten
FILENAME	INPUT	STRING	Name der zu löschenden Datei / des zu löschenden Ordners
RET_VAL	OUTPUT	WORD	Status / Fehlercode
BUSY	OUTPUT	BOOL	TRUE Job wird bearbeitet
DONE	OUTPUT	BOOL	TRUE Job wurde erfolgreich abgeschlossen
ERROR	OUTPUT	BOOL	TRUE Job ist fehlgeschlagen

Bedeutung der Rückgabewerte

RET_VAL	BUSY	DONE	ERROR	Beschreibung
W#16#0000	0	1	0	Job wurde erfolgreich abgeschlossen
W#16#7000	1	0	0	Job wird bearbeitet
W#16#7001	0	0	0	Job-Ressource ist temporär verriegelt
W#16#7002	0	0	0	REQ-Parameter ist FALSE
W#16#7003	0	0	1	Job wurde nicht akzeptiert, solange der vorhergehende Job noch in Abarbeitung ist. (Immer nur einen der SFC 207-209, 213 und 220-225 benutzen)
W#16#8000	0	0	1	Dateisystem ist noch nicht bereit (z.B, Micro-SD-Karte nicht gesteckt)
W#16#8001	0	0	1	Datei / Order wird gerade benutzt
W#16#8002	0	0	1	Datei / Order wurden nicht gefunden (FILENAME-Parameter)
W#16#8003	0	0	1	Name ist zu lang (FILENAME-Parameter)
W#16#8004	0	0	1	Ordner ist nicht leer
W#16#8005	0	0	1	Konnte Datei / Ordner nicht löschen
W#16#822A	0	0	1	Ungültiges Format im FILENAME-Parameter
W#16#8xyy	0	0	1	Allgemeiner Fehlercode (S7-kompatibel)

S7-Anwenderprogramm zum Löschen einer Datei

```

CALL SFC 221
  REQ      :=M1800.0
  FILENAME:=DB221.FILENAME
  RET_VAL  :=MW1000
  BUSY     :=M1800.1
  DONE     :=M1800.2
  ERROR    :=M1800.3

O    M    1800.2           // Job ausgeführt
O    M    1800.3           // oder fehlgeschlagen
R    M    1800.0           // Reset Request Signal

UN   M    1800.3
BEB                                     // Kein Fehler aufgetreten

L    MW 1000                 // Fehler aufgetreten
// error evaluation
// ...

BEA                                     // Ende der Funktion

```

Systemfunktionen

LCD-Hinterleuchtungseinstellung mit dem SFC 214 "LCD_BRIG"

Diese Funktion wird für die Änderung der Hinterleuchtung (Helligkeit) einer Panel-SPS genutzt.

Dieser SFC ist asynchron, d.h., seine Ausführung erstreckt sich über mehrere SFC-Aufrufe hinweg.

Nur ein Auftrag kann zur Abarbeitung anstehen.

Parameter	Bezeichnung	Datentyp	Beschreibung
REQ	INPUT	BOOL	REQ = TRUE: Anfrage zum Einstellen/ Einlesen der LCD Helligkeit
SET	INPUT	BOOL	TRUE= Stelle Helligkeit, FALSE= Lese Helligkeit
LADDR	INPUT	WORD	logische Adresse = 0 (für alle Panel-SPS) logische Adresse = 1...16 (Verbindungs-ID-Nummer zum Panel-HMI)
BRIGHTNESS	INPUT	INT	LCD Helligkeitswert in Prozent 0..100%
RET_VAL	OUTPUT	WORD	Statuscode
DONE	OUTPUT	BOOL	TRUE= Auftrag ist erledigt
BUSY	OUTPUT	BOOL	TRUE= Auftrag ist in Bearbeitung
ERROR	OUTPUT	BOOL	TRUE=Auftrag fehlgeschlagen, Fehlercode in RETVAL
BACKLIGHT	OUTPUT	BOOL	TRUE= LCD Hinterleuchtung ist AN, FALSE= LCD Hinterleuchtung ist AUS (durch Tastendruck mit VisuStage-Systemfunktion)
BRIGHTNESS_CUR	OUTPUT	INT	Aktueller LCD-Helligkeitwert

Bedeutung der Rückgabewerte

Rückgabewert (RETVAL)	BUSY	DONE	ERROR	Beschreibung
W#16#0000	0	1	0	Erfolgreich gelesen
W#16#7000	1	0	0	Auftrag ist in Bearbeitung
W#16#7001	0	0	1	Verbindung in ungültigen Zustand (z.B. nicht verbunden)
W#16#7002	0	0	1	Auftrag nicht akzeptiert, solange vorhergehender Auftrag bearbeitet wird
W#16#7003	0	0	1	Verbindungsressource besetzt (temporär blockiert)
W#16#7004	0	0	0	REQ Parameter ist FALSE, Auftrag ist nicht gestartet
W#16#8001	0	0	1	Ungültige Verbindungs-ID-Nummer im LADDR-Parameter
W#16#8002	0	0	1	Visualisierung läuft nicht (auf Panel-HMI)
W#16#8004	0	0	1	Ungültiger LCD-Helligkeitswert im BRIGHTNESS-Parameter.
W#16#8xyz	0	0	1	Allgemeiner Fehlercode (S7-kompatibel)

Systemfunktionen

S7-Programmbeispiel für die Verwendung des SFC214

Ziel: Änderung des Hintergrundbeleuchtungswertes durch das MW2024

```
CALL SFC 214
  REQ      :=M1502.3
  SET      :=TRUE      // TRUE= Change the LCD brightness
  LADDR    :=MW2000     // VisuStage Verbindungs-ID-Nummer
                        // (Bei Panel-SPSen immer W#16#0 verwenden!)
  BRIGHTNESS :=MW2024   // LCD Helligkeitswert zum Ändern (aktuell 0..100)
  RET_VAL  :=MW1000     // Resultat / Rückgabewert
  DONE     :=M1002.0    // TRUE = erfolgreich erstellt
  BUSY     :=M1002.1    // TRUE = Anforderung in Abarbeitung
  ERROR    :=M1002.2    // TRUE = Fehler aufgetreten
  BACKLIGHT :=M2026.0   // LCD Hinterleuchtungsstatus
  BRIGHTNESS_CUR :=MW2028 // Aktueller LCD-Helligkeitswert

O      M 1002.0 // Erfolgreich ausgeführt
O      M 1002.2 // oder fehlgeschlagen
R      M 1502.3 // Rücksetzen des Anforderungssignals

UN     M 1002.2 // wenn nicht fehlgeschlagen, dann
  BEB                                     // dann Funktion beenden

      L   MW 1000 // Fehler aufgetreten
                        // Fehlersuche
                        // ...

      BEA // Ende der Funktion
```

Systemfunktionen

Von RAM nach ROM kopieren mit dem SFC 254 "RAM2ROM"

Die INSEVIS-SPSen verfügen über einen SFC 254 "RAM2ROM", um Datenbausteine vom (batteriegepufferten) Arbeits- in den Ladespeicher (Flash) zu kopieren.

- Hinweise:**
- Asynchrone Funktion (Ausführung zieht sich über mehrere SFC- Aufrufe hin)
 - Diese Funktion darf nicht unterbrochen werden (z.B. durch Power OFF, oder PG speicherrelevante Funktionen wie „Clear all“, „Block delete“, etc.).
 - Datenbausteine, die mit "Non-Retain" bezeichnet sind, werden ebenfalls in den Ladespeicher kopiert (Aktualdaten).

Beispiel: Einlagern der Steuerung mit aktuellen Prozessdaten. Vor Einlagern SFC254 ausführen, wenn Akku leer und RAM-Inhalt verloren, wird bei nächstem Power-On der Inhalt des Flashes wieder ins RAM geladen. Lediglich die Uhrzeit ist dabei neu einzustellen.

Parameter	Bezeichnung	Datentyp	Beschreibung
REQ	INPUT	BOOL	TRUE = Start des Prozesses
RET_VAL	OUTPUT	WORD	Statuscode W#16#0000 Kopieren ist in Abarbeitung, wenn BUSY=TRUE Kopieren ist abgeschlossen, wenn BUSY=FALSE W#16#7000 Kein Kopiervorgang läuft, wenn REQ = FALSE W#16#8001 Kopie ist nicht erlaubt, andere speicherveränderungen Operationen sind aktiv
BUSY	OUTPUT	BOOL	TRUE = Kopieren ist in Bearbeitung

Programmierbeispiel für die Verwendung des SFC254

Ziel: Kopieren aller DBs in den Ladespeicher (Aktuelle Werte im Ladespeicher werden dadurch überschrieben).

UN	M	1502.2	// Wenn keine Anforderung gesetzt,
BEB			// dann Funktion beenden
CALL	SFC	254	
REQ	:	=M1502.2	// TRUE = Anforderung zum Kopieren aller DBs in den
			// Ladespeicher
RET_VAL	:	=MW1000	// Resultat / Rückgabewert
BUSY	:	=M1702.2	// TRUE = Anforderung in Bearbeitung
U	M	1702.2	// Kopieranforderung in Bearbeitung
BEB			// dann Funktion beenden
R	M	1502.2	// Rücksetzen des Anforderungssignals
L	MW	1000	// Resultat / Rückgabewert
L	0		// Wenn kein Fehler,
==I			
BEB			// dann Funktion beenden
L	MW	1000	// Fehler aufgetreten
			// Fehlersuche
			// ...
BEA			// Ende der Funktion

Betriebssystem in der SPS updaten

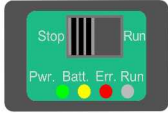
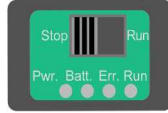
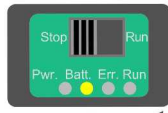
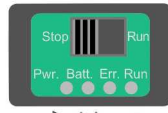
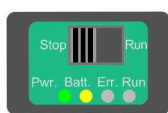
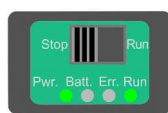
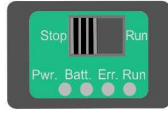
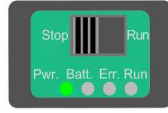
Es ist möglich auf verschiedenen Wegen möglich, das Betriebssystem zu aktualisieren, **ohne dass das Anwenderprogramm beeinflusst wird**. Die Firmware erhalten Sie kostenlos von INSEVIS auf Anfrage als ZIP-Datei, diese ist zu entpacken und mit der Binärdatei weiterzuarbeiten.

1.) Via Micro-SD-Karte - Bei allen Geräten möglich:

Dazu benötigt man lediglich eine Micro-SD-Karte **mit Betriebssystemdatei im Rootverzeichnis** (z.B. PC35Px.bin, PC57Vx.bin, etc), die INSEVIS-SPS (AG) und das 24V-Netzteil dafür.

Vorgehensweise

Anzeige im Display

Vorbereitung - Run / Stop-Schalter in STOP - Gerät Power off - Einstecken der Micro-SD-Karte - AG Power on	1)  Stop  Run Pwr. Batt. Err. Run Stop Power off SD-Card in Power on	
Anzeige „interner Check“ (hängt von bestehenden Softwarestand ab) LEDs Batt. (gelb) blinkt, solange CPU-Status geprüft	2)  Stop  Run Pwr. Batt. Err. Run internes Prüfen internal check abwechselnd alternatively	 Checking ...
Anzeige „Ladevorgang läuft“ (Achtung: keine Spannungsunterbrechung erlaubt!) Power-LED leuchtet permanent LEDs Batt. (gelb) und Run (grün) blinken abwechselnd	3)  Stop  Run Pwr. Batt. Err. Run Ladevorgang... Loading process (ca. 1min) abwechselnd alternatively	a)  Updating ... b)  Checking ...
Anzeige „Ladevorgang beendet“ Power-LED leuchtet permanent LEDs Batt. (gelb) und Run (grün) blinken gleichzeitig	4)  Stop  Run Pwr. Batt. Err. Run Laden beendet Loading finished gleichzeitig synchronously	 Update done
geladenes Betriebssystem initialisieren - Gerät Power off - SD-Karte entfernen - Gerät Power on	5)  Stop  Run Pwr. Batt. Err. Run Power off SD-Card out Power on	

2.) Via ServiceStage - Bei allen Geräten mit CPU-T möglich (selbsterklärend → siehe Kapitel ServiceStage)

Systemfunktionen

Hardwareseitiges Urlöschen

Prinzipiell gibt es 2 verschiedene Arten urlöschen:

- per Software: mit dem PG oder per ServiceStage (wird nicht weiter erläutert)
- per Hardware ohne PG/ServiceStage direkt an der SPS (nachfolgend erläutert)

Nach dem hardwareseitigen Urlöschen werden nur noch die IP-Adresse und die Uhrzeit im batteriegepufferten Speicher gehalten (wie bei Siemens auch)

Vorgehensweise hardwareseitiges Urlöschen:

*Run/Stop-Schalter nach Stop
Gerät spannungsfrei schalten
Gerät einschalten*

Während der ersten 3 Sekunden sind die LEDs „Power“, „Batt.“, „Error“ an,

1 Run/Stop-Schalter nach Run,
rote Error-LED geht aus
grüne Run-LED geht an
innerhalb der nächsten Sekunde

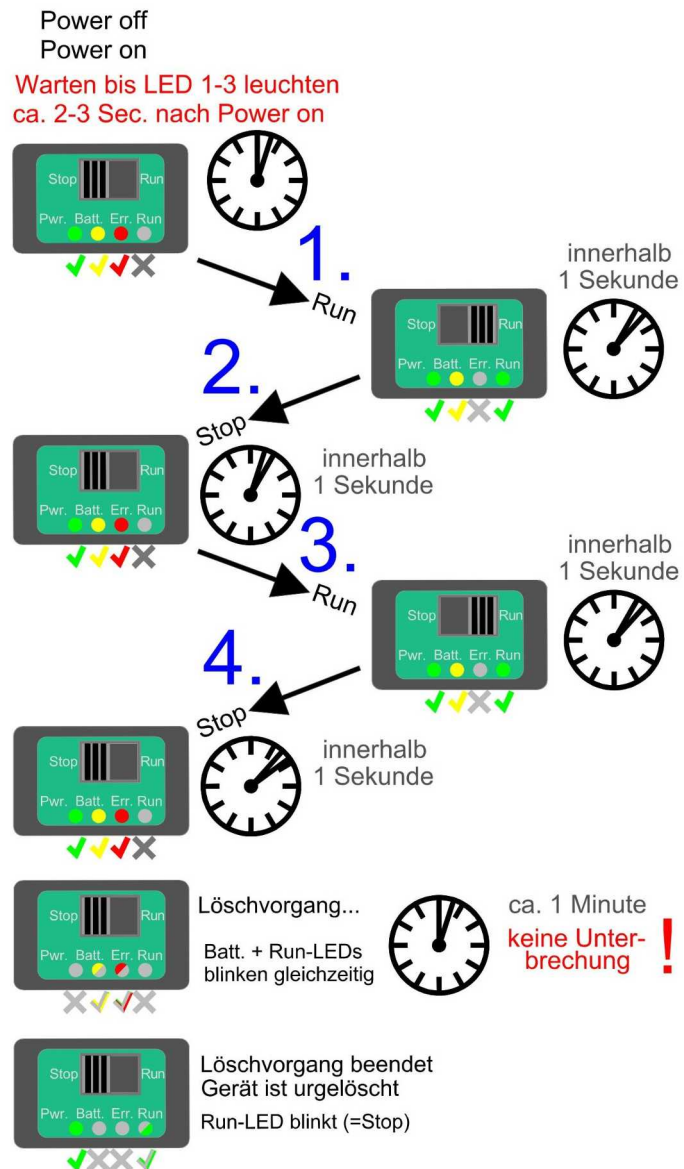
2 Run/Stop-Schalter nach Stop
rote Error-LED geht an
grüne Run-LED geht aus
innerhalb der nächsten Sekunde

3 Run/Stop-Schalter nach Run
rote Error-LED geht aus
grüne Run-LED geht an
innerhalb der nächsten Sekunde

4 Run/Stop-Schalter nach Stop
rote Error-LED geht an
grüne Run-LED geht aus

Löschvorgang für ca. 1-5 Minuten
Gerät nicht ausschalten !
LEDs Batt (gelb) und Err. (rot)
blinken gleichzeitig

Löschvorgang beendet
Run-LED blinkt (Gerät in Stop)



Hinweis zum RESTORE nach Hardware-Reset

Nach dem Hardware-Reset versucht die SPS, die Daten von der Micro-SD-Karte zu laden. Dabei blinken die BATT- und die RUN-LEDs gleichzeitig.
(Dazu muss auf der Micro-SD Karte ein Unterordner „Backups“ mit den Dateien „S7programs.wld“ und „Visudata.vsbin“ vorhanden sein.)

Ist der Ladevorgang beendet, oder steckt keine Micro-SD-Karte verbleibt die Steuerung im Stop-Zustand (RUN-LED blinkt).

* Während der ersten 3 Sekunden ist ein Wert für die CPUs -V und -P.
Die größere CPU-T benötigt länger für den Bootprozess (etwa 6-8 Sekunden).
Falls ein Display dabei ist, ist das der Zeitpunkt nach Ablauf des Fortschrittsbalkens.

Remotezugang mit der Software „RemoteStage“

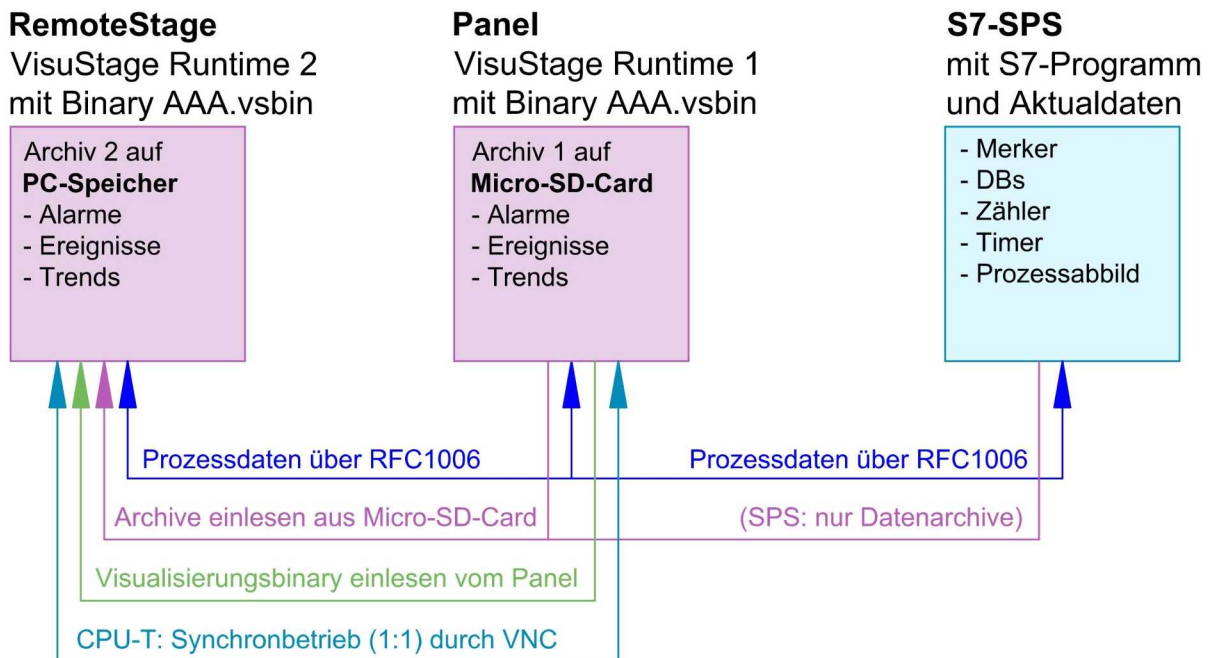
Allgemeines

Die RemoteStage ist eine multiinstanzfähige Kommandozeilen-Software, die nicht installiert wird, sondern als **remotestage.exe** zusammen mit der Datei für die deutsche Bedienung **remotestage.deu** in ein Verzeichnis auf dem Remote-PC oder auf einen mobilen Datenträger abgelegt wird. Als Kommandozeilentool ist es möglich, diverse Funktionen in einer Batchdatei zu einem bestimmten Zeitpunkt programmgesteuert aufzurufen. Die „RemoteStage“ Software ist auf den aktuell gängigen Windows Versionen lauffähig.

Die RemoteStage benutzt die Binärfiles (*.vsbin) von vorab mit der „VisuStage“ erstellten Visualisierungen. Die Binärdaten sind nicht rückkonvertierbar, welches effektiven Know-how-Schutz bietet und z.B. die Weitergabe an Endkunden erlaubt. Es ist möglich, die Visualisierungs-Binärdaten via Ethernet direkt von der Panel-SPS/ dem Panel-HMI herunterzuladen, welches den Verwaltungsaufwand reduziert. In der VisuStage kann dieser Upload erlaubt und PIN-Code-geschützt werden.

Die RemoteStage arbeitet **wie ein zusätzliches Panel-HMI** auf dem Remote-PC, welches die Aktual- (Prozess-) daten über S7-Put/Get von der verbundenen SPS bezieht. Systemdaten wie z.B. Uhrzeit und Buzzer beziehen sich dann auch nur auf diesen PC. Das ermöglicht eine Remote-Visualisierung und Zugriff auch auf Fremd-S7-CPU's über S7-Ethernet (Put/Get).

In einer zweiten Instanz (Archivmodus) liest die RemoteStage Archivdaten (direkt von der Speicherkarte oder über Ethernet) von der Micro-SD-Karte ein, zeigt sie an und konvertiert diese in ein csv-Format.



Wenn die RemoteStage für den Kunden als **Batchaufruf** bei dem Endkunden installiert werden soll, reicht folgende Eingabeaufforderung aus:

```
[RemoteStage-Pfad]\remotestage.exe /r=[IP-Adresse der Remote-SPS im selben Subnetz]
/v=[Visualisierungsbinary-Pfad]\vollständiger Dateiname]
/start
```

Auswahl der Remote-Steuerung

Beim Öffnen der Binärdatei der Visualisierung („visualisierung“.vsbin) wird einmalig die IP-Adresse der Remote-Steuerung angefordert (Abb.) Dabei wird mit der INSEVIS-Sytemfunktion im Netz nach erreichbaren Teilnehmern gesucht

Danach ist diese IP-Adresse bis zu einer manuellen Änderung mit dieser Datei verknüpft und muss nicht wieder eingegeben werden.

Das Dialogfeld 'Remote-Adresse' enthält folgende Elemente:

- Titel: Remote-Adresse
- Textfeld: Bitte Remote-SPS-Adresse eingeben
- Checkbox: Remote TCP-Port
- Textfeld: Verwenden Sie eine andere TCP-Portnummer als die voreingestellte.
- Textfeld: TCP-Portnummer: 102
- Text: Achtung: Vorgeschriebener TCP-Port ist 102.
- Buttons: OK, Abbrechen

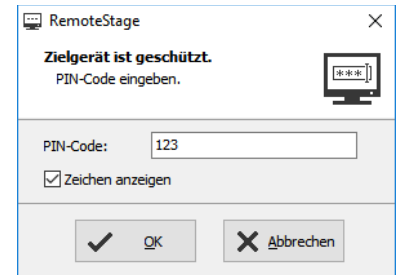
Remotezugang mit der Software „RemoteStage“

Eingabe des Passwortes

Wurde in der VisuStage der Upload des kompilierten VisuStage-Projektes (Binärdatei vsbin) erlaubt, kann er mit einem Passwort versehen werden. Dann erscheint ein entsprechendes PIN-Code-Eingabefenster (rechts).

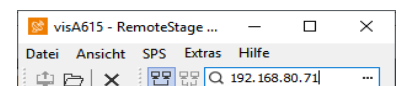
Dieses Passwort muss bei jedem erneuten Upload eingegeben werden (z.B. auch bei Backups über die ServiceStage). Bei INSEVIS-Demoprojekten ist dieses Passwort 123.

Nach erfolgreicher Upload des Binaries in den Remote-PC ist ein schwarzes Fenster zu sehen, welches nach Klick auf die Connect-Taste den in der VisuStage geplanten Startbildschirm anzeigt (nicht 1:1 -das wäre VNC-, sondern ggf. abweichende Seiten).



Benutzen und Einstellen des Remote-Bildschirmes

Die PopUp-Menüs sind selbsterklärend. Der Startbildschirm ist erst einmal schwarz, bis man die Taste „Verbinden“ drückt, danach erscheint die Visualisierung.



Hinweis: Bei Systemeinstellungen zum Buzzer, Erstellen von Archiven, Uhrzeit lesen werden immer die PC-Systemdaten angesprochen, nicht die des Remote-Systems.

Folgende Eigenschaften/Funktionen/Einstellungen werden von der RemoteStage **nicht unterstützt**:

- "Displayhinterleuchtung" und "Bildschirm- und Sprachsteuerung" - aus menü „Einstellung“
- "Synchronisieren der Panelzeit mit der Partnerzeit" - aus Menü „SPS-Gerät Einstellung“
- "Aktivieren der Änderung der Partnerzeit zur Laufzeit" - aus Menü „SPS-Gerät Einstellung“
- „System Uhrzeit ändern“ über Screen-Objekt "Datum-/Uhrzeitfeld"
- Tasten-Funktionen: „Helligkeit erhöhen/verringern“, „Hinterleuchtung aus“, „Aktiviere Putzbildschirm“
- VNC Server

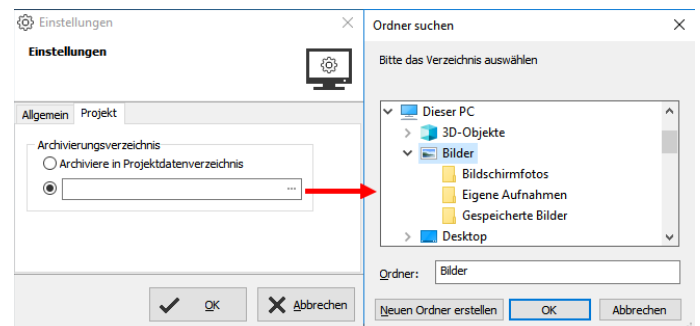
Abspeichern der remote (auf dem PC) archivierten Daten

(noch im Visualisierungsmodus festzulegen!)

Die während der Remote- Sitzung am Remote-PC angefallenen Archivdaten werden **nur** am Arbeitsplatz des PCs abgelegt (unabhängig vom Archiv in der Micro-SD-Karte bei de Panel-SPS/HMI, es gibt also jeweils 2 Archive). Ein Trend muss z.B. in der Remote-Visualisierung extra gestartet werden.

Unter dem Menüpunkt „Einstellungen“ wird ausgewählt, ob die Archivdaten

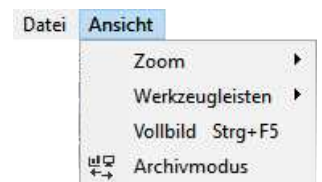
- unter dem Projektverzeichnis
- (wo auch die Visualisierungsbinärdatei steht)
- oder einem anderen Pfad abgespeichert werden.



Hinweis: Wenn Sie die Visualisierung von dem Remote-Gerät in Ihren Remote-PCs hoch laden (diese Funktion wird im VisuStage-Projekt aktiviert), wird diese im lokalen TEMP-Verzeichnis Ihres PCs zwischengespeichert. Dort werden während der Remote-Sitzung auch die temporären Archive im Binärformat gespeichert. Zum Einlesen, Speichern und Konvertieren in das CSV-Format bitte im Archivmodus bei „Datei Öffnen“ das lokale TEMP-Verzeichnis angeben.

Diese meisten der Archivdaten liegen auf der Speicherkarte als Binärdaten vor und können ausschließlich von der RemoteStage in csv-Dateien konvertiert werden.

Dazu öffnen Sie innerhalb der RemoteStage eine neue Programm-Instanz mit den Archivkonvertierungsmodus unter Ansicht (rechts)



ACHTUNG:

Verwenden Sie nur die Visualisierungs-Binärdatei der zu dem Archiv passenden Visualisierung!

Remotezugang mit der Software „RemoteStage“

Einlesen, Konvertieren und Abspeichern der Archivdaten

Folgende Daten können von folgenden Produkten auf der Micro-SD-Karte archiviert werden:

Meldungen (Alarme, Ereignisse)	von allen Panel-SPSen und Panel-HMIs → durch Setzen des Häkchens „archivieren“ im VisuStage-Projekt
Trends (mit allen Kanälen)	von allen Panel-SPSen und Panel-HMIs → durch Setzen des Häkchens „archivieren“ im VisuStage-Projekt
Rezepturen (mit allen Datensätzen / Elementen)	von allen Panel-SPSen und Panel-HMIs → werden immer archiviert, weil auf Micro-SD-Karte gespeichert
Archiv-Datenbausteine (als bin bzw. csv)	von allen Panel-SPSen und Kompakt-SPSen → werden durch SFC 207 im S7-Programm konfiguriert

Im neuen PC-Fenster erscheinen verschiedene Gruppen von Icons mit verschiedenen Funktionen:



Anzeige durch das Einstecken der Micro-SD-Karte in PC-Kartenslot
v.l.n.r.:
- Daten-(DB-) archiv*
- Alarmarchiv*
- Ereignisarchiv*
- Trendarchiv*

Einlesen des Archivs per Ethernet über IP-Adresse direkt in den PC
v.l.n.r.:
- Alle Archive zusammen
- Alarmarchiv
- Ereignisarchiv
- Rezepturarchiv
- Trendarchiv
- Daten-(DB-)archiv

Speichern der Archivdaten auf PC/ in Netzwerk
v.l.n.r.:
- Alarmarchiv
- Ereignisarchiv
- Rezepturarchiv
- Trendarchiv
- Daten-(DB-)archiv

Ändern und Downloaden geänderter Archive
v.l.n.r.:
- Archiv speichern
- Neues Archiv hinzufügen
- Archiv umbenennen
- Archiv kopieren*
- Rezepturen downloaden

* Um die Dateien zu öffnen ist es nötig, irgendeine halbwegs gültige IP-Adresse einzugeben und die Binärdatei der jeweiligen Visualisierung zu öffnen, um die Archivdateien in das csv-Format konvertieren zu können.

Ereignis- oder Alarmarchivdarstellung und -export

Die Archivdaten werden von der Micro-SD-Karte eingelesen, können angezeigt und z.B. manuell als csv-Datei abgelegt werden.

unten: Anzeige des Alarmarchivs

Datum	ID-Nr	Status	Text	Data
04.01.2019 09:20:57	1	K	Alarm 1 hat ausgelöst	
04.01.2019 09:20:58	2	K	Alarm 2 hat ausgelöst	
04.01.2019 09:20:59	1	G	Alarm 1 hat ausgelöst	
04.01.2019 09:20:59	2	G	Alarm 2 hat ausgelöst	
04.01.2019 09:21:00	1	K	Alarm 1 hat ausgelöst	
04.01.2019 09:21:00	2	K	Alarm 2 hat ausgelöst	
04.01.2019 09:21:19	2	G	Alarm 2 hat ausgelöst	
04.01.2019 09:21:20	1	G	Alarm 1 hat ausgelöst	
04.01.2019 09:21:33	2	Q	Alarm 2 hat ausgelöst	
04.01.2019 09:21:33	1	Q	Alarm 1 hat ausgelöst	

(Anzeige aller archivierten Alarmeinträge von der Panel-SPS mit der IP-Adresse 192.168.80.55)

unten: Export des Alarmarchivs als csv-Datei

Exportmeldungen
Auswahl der Sprachen
Auswahl der Feldbegrenzer und Texttrenner

Sprachen:
☒ Deutsch (Deutschland)
☐ Englisch (Großbritannien)
☐ Russisch (Russische Föderation)

☐ alle auswählen

Feldbegrenzer:
☒ Semikolon ☐ Tab
☐ Komma ☐ Leerzeichen

Texttrenner:
☒ " ☐ { ☐ {

Vorschau:

```
"Id";"Status";"Time stamp";"Value";"1031 Deutsch (Deutschland)"
"1";"K";"14.11.2018 11:13:57";";";"Alarm 1 hat ausgelöst"
"2";"K";"14.11.2018 11:13:59";";";"Alarm 2 hat ausgelöst"
"1";"K";"04.01.2019 09:20:57";";";"Alarm 1 hat ausgelöst"
```

OK Abbrechen



Hinweis:

Nicht verwechseln mit den während der Remote-Verbindung am Remote-PC archivierten Alarmeinträgen!
Auf die kundenspezifisch individuelle Formatierung des csv-Formates ist selbst zu achten.

Remotezugang mit der Software „RemoteStage“

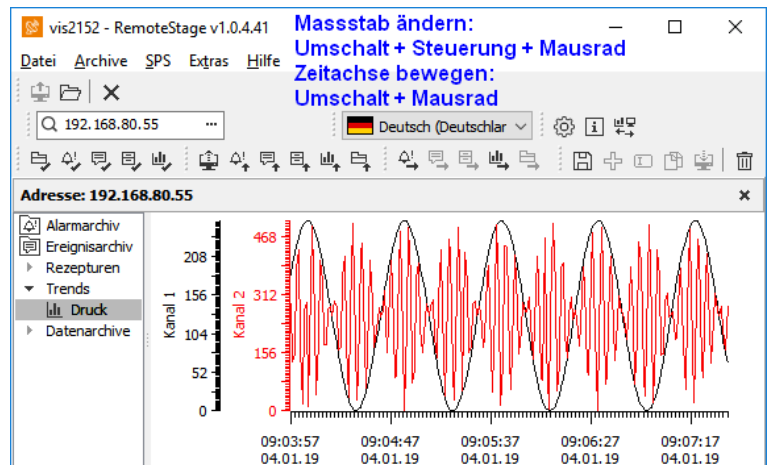
Trenddarstellung und -export

Die Archivdaten werden von der Micro-SD-Karte eingelesen, können angezeigt und als csv-Datei abgelegt werden.

rechts: Anzeige des Trendarchivs „Druck“ mit Hinweis zur Navigation in der Anzeige

(Anzeige aller archivierten Trendaufzeichnungen des Trends „Druck“ von der Panel-SPS mit der IP-Adresse 192.168.80.55).

Anzeige der Zeit mit dem am PC eingestellten Format (normal: DE = deutsches Zeitformat)



Hinweis: Nicht verwechseln mit den während der Remote-Verbindung am Remote-PC archivierten Trends! Der csv-Export verhält sich wie bei den Ereignis- oder Alarmarchiven. Auf die kundenspezifisch individuelle Formatierung des csv-Formates ist selbst zu achten.

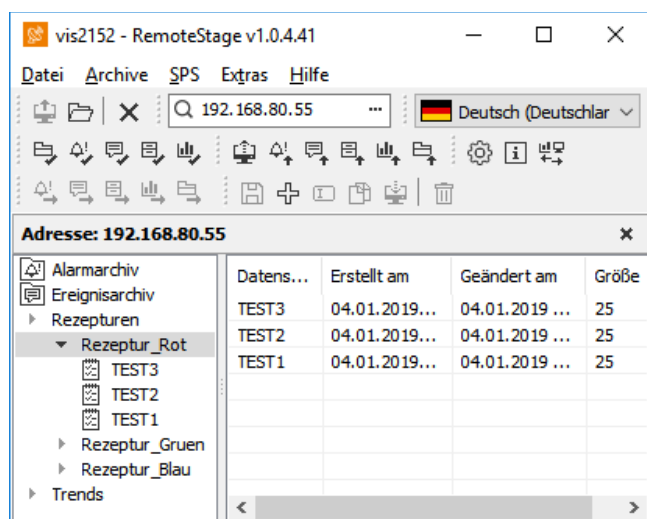
Remotezugang mit der Software „RemoteStage“

Rezepturhandling

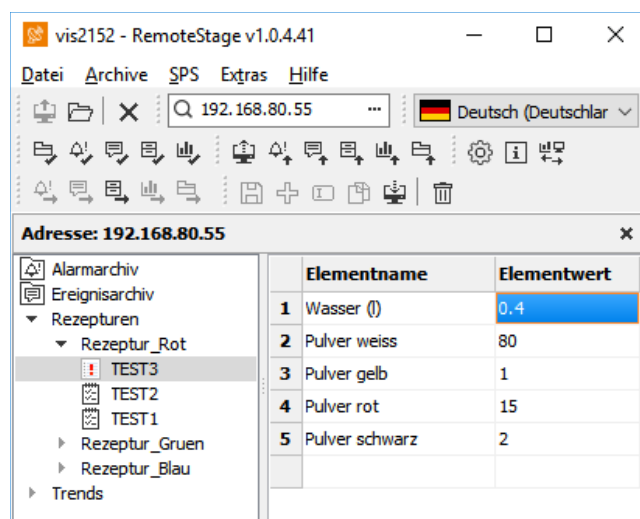
Die Rezepturdaten werden auf der Micro-SD-Karte abgelegt und können mit der RemoteStage von dort aus

- eingelesen werden (immer das ganze Root-Verzeichnis der Micro-SD-Karte auswählen!)
- angezeigt (und geändert) werden (selbsterklärend) und
- **zum BEARBEITEN auf dem PC im CSV-Format abgespeichert werden:**
- als csv-Datei im PC gespeichert, dort geändert und zurück in Remotegerät gespielt werden (selbsterklärend),
- **zum KOPIEREN in ein anderes Gerät in Binärformat auf dem PC zwischengespeichert werden**
- als Binärdatei gespeichert, neu geöffnet und zurück in ein anderes Remote-Gerät übertragen werden (Immer das ganze „recipe“-Verzeichnis mit seinen Rezeptur-Unterordnern auf dem PC auswählen!)

unten links: Liste der Rezepturen mit den Datensätzen



unten rechts: Liste der Datensätze mit den Elementen



Der Datensatz „TEST3“ wurde in der RemoteStage geändert und deswegen mit einem „!“ gekennzeichnet. Nach dem Übertragen (Download) in das Remote-Gerät entfällt diese Kennzeichnung wieder.

Das Speichern aller Rezepturen (als Binärdateien in einem eigenen Ordner im Netzlaufwerk zum Editieren mit der RemoteStage) erfolgt, wenn in der Baumübersicht (links) „Rezepturen“ markiert wurden.

Wenn Rezepturen auf einem PC-Laufwerk geöffnet werden, ist in der Adresszeile deren Pfad zu sehen, wenn Rezepturen von dem Remote-Gerät eingelesen werden, sieht man dessen IP-Adresse.

Das Neuanlegen eines Datensatzes erfolgt, wenn in der Baumübersicht (links) eine einzelne Rezeptur markiert wurde.



ACHTUNG:

Die Struktur der Rezepturen MUSS erhalten bleiben. Datensätze können innerhalb der jeweiligen Rezepturen hinzugefügt oder gelöscht werden.

Remotezugang mit der Software „RemoteStage“

Datenarchivhandling

Frei definierbare beliebige Daten können mit dem **SFC 207 ARCHIVE** auf die Micro-SD-Karte (Binär oder im csv-Format) archiviert und mit dem **SFC 209 READFILE** wieder von der Micro-SD-Karte (immer im Unterordner UserData) in die SPS zurückgelesen werden (siehe Systemfunktionen).

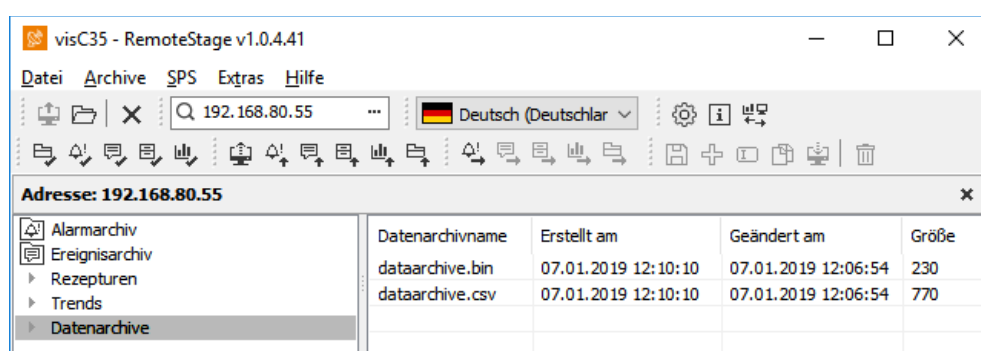
In der RemoteStage-Oberfläche können keine Datenarchive manuell zurück auf die Micro-SD-Karte der Steuerung geschrieben werden. Das kann per Kommandozeilenbefehlen erfolgen (Beispiele auf den folgenden Seiten).

Datenarchivdarstellung und -abspeicherung

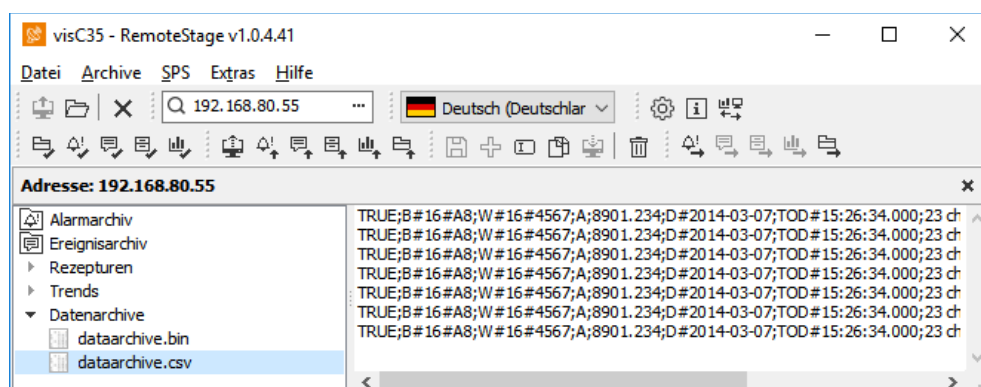
Die vom S7-Programm archivierten Daten werden per RemoteStage von der Micro-SD-Karte eingelesen, können in der RemoteStage angezeigt und (mit ihrem im S7-Programm im SFC207 festgelegten Format) im PC gespeichert werden.

Diese Datei ist mit der RemoteStage nicht änderbar. Das Ändern kann nur mit Kundentools erfolgen, weil der RemoteStage die Struktur unbekannt ist.

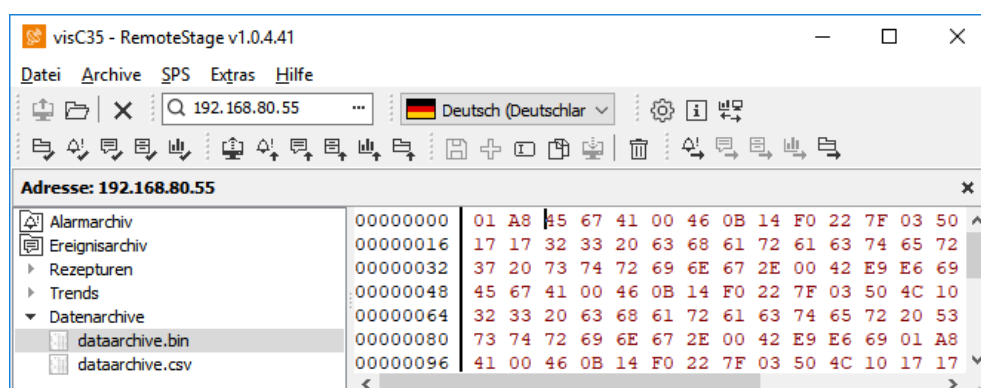
unten: Anzeige der verfügbaren Datenarchive auf der Micro-SD-Karte



unten: Anzeige des auf der Micro-SD-Karte im Textformat zur Verfügung stehenden Datenarchives



unten: Anzeige des auf der Micro-SD-Karte im Binärformat zur Verfügung stehenden Datenarchives



Remotezugang mit der Software „RemoteStage“

Kommandozeilenbefehle für RemoteStage - Visualisierungsmodus

Die RemoteStage kann auch über ein Batchfile in den Startvorgang des Remote-PCs (Windows®-Funktion „geplante Tasks“ bzw. „anstehende Aufgaben“) mit eingebunden werden. Verknüpfungen von mehreren Aufrufparametern sind möglich.

In einer Batchdatei sind folgende Befehle möglich:

Automatisches Verbinden zum Remote-Gerät nach Start

Hier wird eine vorab erstellte Binärdatei der Visualisierung aufgerufen und mit einer IP-Adresse des Remote-Panels verbunden (solange dieses erreichbar ist).

Kommandozeile

<Pfad>\remotestage.exe /start /r=<remote> /v=<project binary>

Hinweis:

Wenn der Name der Binärdatei Leerzeichen enthält, muss der komplette komplette Pfad UND der Dateiname in Hochkommas gesetzt werden. (z.B. „C:\Doku\Test 1 2 3.vsbin“)

Der Aufrufparameter „start“ ermöglicht ein automatisches Verbinden und Anzeigen bei der Panel-Ansicht (-Instanz) der RemoteStage.

Automatisches Wiederverbinden zum Remote-Gerät nach Verbindungsabbruch

Hier wird nach einer einstellbaren Wartezeit die Verbindung zum Partnergerät wieder aufgebaut.

Kommandozeile

<Pfad>\remotestage.exe /restart=[seconds] /r=<remote> /v=<project binary>

Hinweis:

Wenn der Name der Binärdatei Leerzeichen enthält, muss der komplette komplette Pfad UND der Dateiname in Hochkommas gesetzt werden. (z.B. „C:\Doku\Test 1 2 3.vsbin“)

Wenn der Aufrufparameter „restart=0 (default) ist, wird die Verbindung nur einmalig aufgebaut, sonst entspricht der Wert der Wartezeit in Sekunden vom Verbindungsabbruch bis zum geplanten Start des Verbindungsneuaufbaus.

Einstellen der TCP-Portnummer für die S7-Kommunikation

Wenn z.B. mehrere Steuerungen über einen internetfähigen Router/ Gateway mit Portforwarding-Service erreicht werden sollen. In diesem Fall beginnt die RemoteStage eine Kommunikation mit einem Remote-Gerät mit der IP-Adresse 192.168.80.50 über den TCP-Port mit der Nr. 4500.

Kommandozeile

<Pfad>\remotestage.exe /v=d:\visu\process.vsbin /r=192.168.80.50 /p=4500

Hinweis:

Die vorgeschriebene Portnummer für TCP ist Port 102 bei den Steuerungen.

Vollbildschirm nach Start

Hier wird eine vorab erstellte Binärdatei der Visualisierung mit einer IP-Adresse verbunden (solange diese erreichbar ist) und die RemoteStage geht in Vollbildformat. (Der Bildschirm bleibt dunkel, weil keine Verbindung eingetragen wurde...)

Kommandozeile

<Pfad>\remotestage.exe /start /full /r=<remote> /v=<project binary>

Hinweis:

Umschalten Voll-/Teilbildschirm bei der RemoteStage mit STRG+F5

Der Aufrufparameter „start“ ermöglicht ein automatisches Verbinden und Anzeigen bei der Panel-Ansicht (-Instanz) der RemoteStage.

Remotezugang mit der Software „RemoteStage“

Kommandozeilenbefehle für RemoteStage - Archivierungsmodus

Die RemoteStage kann auch in einem Batchprozess mit diesen Kommandozeilenparametern benutzt werden, um automatisch Archive irgendwo unter einem selbst zu definierenden Namen abzulegen.

Im PC-Betriebssystem Windows® kann man im Taskmanager **zeitgesteuert** ein Batchfile aufrufen:

Startmenü → Alle Programme → Zubehör → Systemprogramme → Geplante Tasks → Geplanten Task hinzufügen

Im Wizard - Dialog kann man Programme, Uhrzeit, etc. für den Job einstellen.

Lesen (Upload) der Meldungs- (Alarme und Ereignisse) und Trendarchive

Hier werden die Binärdaten der in der VisuStage vordefinierten und zur Laufzeit erzeugten Meldungen und Trends ausgelesen, konvertiert und in den gewünschten Zielpfad außerhalb der Panel-SPS/des Panel-HMIs gespeichert

Kommandozeile (ohne Zeilenumbruch bitte!)

```
<Pfad>\remotestage.exe /mode=<mode> /r=<remote> /o=<output file>
/v=<project binary> /t=<trend number>
```

Parameter

mode	Einlesen der Archivdatei alarm_archive = Einlesen des Alarmarchivs vom Remote-Gerät (alarm.bin) event_archive = Einlesen der Ereignisarchivs vom Remote-Gerät (event.bin) trend_archive = Einlesen des Trendarchivs vom Remote-Gerät (trend_<trend number>.bin) (wobei die Trend-ID-Nummer fest von der VisuStage vergeben wird – siehe Ressource Trend)
remote	Remote-Gerät IP-Adresse
output file	Dateiname des Archivs bei der Abspeicherung im PC Das ist ein optionaler Parameter. Wenn der nicht angegeben ist, wird der vorgegebene Archivname benutzt. (z.B. alarm.bin) und im selben Verzeichnis gespeichert, wo sich die remotestage.exe befindet. Wenn der Dateiname Leerzeichen enthält, sollten diese mit " (Anführungszeichen) begrenzt werden. Die Archive werden in das csv- Format konvertiert, wenn der Dateiname die Endung .csv oder .txt besitzt, anderenfalls wird im Binärformat abgespeichert.
project binary	Visualisierungsprojekt-Binärdateiname Hinweis: Wenn der Name der Binärdatei Leerzeichen enthält, muss der komplette Pfad UND der Dateiname in Hochkommas gesetzt werden. (z.B. „C:\Doku\Test 1 2 3.vsbin“)
trend number	Trend-ID, welche von der VisuStage vergeben wird (nur bei Trendarchiv-Modus)

Beispiel

Angenommen, die RemoteStage liegt unter C:\ Um beispielsweise ein Alarmarchiv von der INSEVIS Panel-SPS mit der IP-Adresse 192.168.80.50 einzulesen und als csv-Datei nach „D:\archive\alarm.csv“ zu schreiben, wird folgende Kommandozeile benötigt:

Kommandozeile(ohne Zeilenumbruch bitte!)

```
C:\remotestage.exe /mode=alarm_archive /r=192.168.80.50
/o=D:\archive\alarm.csv /v=D:\visu\myvisu.vsbin
```

Angenommen, die RemoteStage liegt unter C:\ Um beispielsweise das Trendarchiv mit der ID6163 (aus Demo-Visualisierung für ein PC1560T) von der INSEVIS Panel-SPS mit der IP-Adresse 192.168.80.50 einzulesen und als csv-Datei nach „D:\archive\trend6163.csv“ zu schreiben, wird folgende Kommandozeile benötigt:

Kommandozeile(ohne Zeilenumbruch bitte!)

```
C:\remotestage.exe /mode=trend_archive /r=192.168.80.50
/o=D:\archive\trend6163.csv /t=6163 /v=D:\PC156xT_2017_03.vsbin
```


Remotezugang mit der Software „RemoteStage“

Lesen (Upload) der Rezepturarchive vom Remote-Gerät

Hier werden die Binärdaten der in der VisuStage vordefinierten Rezepturen und die zur Laufzeit vom Operator abgespeicherten oder mit SFC206 vom S7-Programm erstellten Datensätze ausgelesen, konvertiert und in den gewünschten Zielpfad außerhalb der Panel-SPS/des Panel-HMIs gespeichert

Kommandozeile(ohne Zeilenumbruch bitte!)

```
<Pfad>\remotestage.exe /mode=recipe /r=<remote> /v=<project binary> /recipe=<recipe name>
/record=<record name> /o=<output file> /format=csv
```

Parameter

mode	recipe = Einlesen der Rezepturdaten vom Remote-Gerät
remote	Remote-Gerät IP-Adresse
project binary	Visualisierungsprojekt-Binärdateiname
recipe name	in Visualisierungsprojekt projektierter Rezepturname (NICHT: Anzeigename)
record name	Datensatzname zum Abspeichern BESTIMMTER (einzeln) Datensätze Wenn dieser Parameter nicht vergeben wird, liest die RemoteStage ALLE Datensätze ein, die zu der Rezeptur <recipe name> gehören und speichert diese in dem durch den Parameter "/o=" definierten Ordner. → Um dann das Format der Ausgabedateien als csv zu bestimmen, muss der Parameter "/format=csv" gesetzt werden, andernfalls werden alle Daten als Binärdaten abgespeichert. Bei Format csv sind als Feldbegrenzer ; (Semikoleon) und als Texttrenner " (Anführungszeichen) gesetzt. Wenn dieser Parameter vergeben ist, MUSS mit der Erweiterung des Datensatznamens das Format der Ausgabedatei vorgegeben werden; → Wenn der Dateiname die Endung .csv oder txt besitzt, dann wird das Format csv, anderenfalls das Binärformat abgespeichert
output file:	Pfad und Dateiname des Rezepturdatensatzes bei der Abspeicherung im PC (optionaler Parameter) Wenn unter Parameter "/o=" keine Angabe erfolgt, wird im Projektverzeichnis (wo die Visualisierungs-Binärdatei liegt) ein Ordner "Recipes\<recipe name>" automatisch erstellt und der Rezepturdatensatz als <record name>.rec gespeichert. Wenn der Dateiname Leerzeichen enthält, sollten diese mit " (Anführungszeichen) begrenzt werden.

Beispiele

Angenommen, die RemoteStage liegt unter C:\ Um beispielsweise einen einzigen Rezepturdatensatz „Orange“ aus der Rezeptur „Rezeptur_Rot“ von der INSEVIS Panel-SPS mit der IP-Adresse 192.168.80.50 einzulesen und als csv-Datei nach „D:\archive“ zu schreiben, wird folgende Kommandozeile benötigt:

Kommandozeile (ohne Zeilenumbruch bitte!)

```
C:\remotestage.exe /mode=recipe /r=192.168.80.50 /v=d:\visu\myvisu.vsbinary
/recipe=Rezeptur_Rot /record=Orange /o=D:\archive\Orange.csv
```

Angenommen, die RemoteStage liegt unter C:\ Um beispielsweise einen ALLE Rezepturdatensatz aus der Rezeptur „Rezeptur_Rot“ von der INSEVIS Panel-SPS mit der IP-Adresse 192.168.80.50 einzulesen und als csv-Dateien nach „D:\archive“ zu schreiben, wird folgende Kommandozeile benötigt:

Kommandozeile (ohne Zeilenumbruch bitte!)

```
C:\remotestage.exe /mode=recipe /r=192.168.80.50 /v=d:\visu\myvisu.vsbinary
/recipe=Rezeptur_Rot /o=D:\archive /format=csv
```

Remotezugang mit der Software „RemoteStage“

Zurückschreiben (Download) der Rezepturarchive in das Remote-Gerät

Hier werden die aus dem Remote-Gerät gelesenen (und danach am PC evtl. editierten) Rezepturen wieder in das Remote-Gerät (Panel-SPS oder Panel-HMI) zurückgeschrieben.

(Ab Rev. 1.0.4.17 und SPS-Firmware 2.3.9 (alle CPUs) oder HMI-Firmware 1.2.7 CPU-V/-P und 1.2.6 CPU-T)

Achtung: Die Struktur der Rezepturen im Remote-Gerät darf zwischen Lesen und Zurückschreiben nicht verändert werden. (Kein Element hinzufügen, löschen oder Reihenfolge ändern)

Kommandozeile (ohne Zeilenumbruch bitte!)

```
<Pfad>\remotestage.exe /mode=recipe_download /v=<project binary>
/recipe=<recipe name> /record=<record file(s)>
```

Parameter

project binary Binärdatei des Visualisierungsprojektes

recipe name In VisuStage projektierter Rezepturname, (ACHTUNG: nicht Anzeigename!)

record file(s) Rezepturdaten zum Zurückschreiben in das Remote-Gerät

Beispiele

Angenommen, die RemoteStage liegt unter C:\. Nachstehende Befehlszeile überträgt den EINEN Rezepturdatensatz „Rot_123.rec“ der Rezeptur „Recipe_Rot“ aus dem Ordner D:\recipes in das Remote-Gerät mit der IP-Adresse 192.168.80.50

Kommandozeile (ohne Zeilenumbruch bitte!)

```
C:\remotestage.exe /mode=recipe_download /v=d:\visu\process.vsbin
/recipe=Recipe_Rot /record=d:\recipes\Rot_123.rec /r=192.168.80.50
```

Angenommen, die RemoteStage liegt unter C:\. Nachstehende Befehlszeile überträgt ALLE Rezepturdatensätze der Rezeptur „Recipe_Rot“ aus dem Ordner D:\recipes in das Remote-Gerät mit der IP-Adresse 192.168.80.50

Kommandozeile (ohne Zeilenumbruch bitte!)

```
C:\remotestage.exe /mode=recipe_download /v=d:\visu\process.vsbin
/recipe=Recipe_Rot /record=d:\recipes\*.rec /r=192.168.80.50
```

Remotezugang mit der Software „RemoteStage“

Lesen (Upload) der Daten (DB-) Archive vom Remote-Gerät

Diese Funktion kopiert die im S7-Programm mit SFC207 erstellten Datenarchiv-Dateien in den gewünschten Zielpfad außerhalb der SPS. Es kann hier kein Einfluss auf die Dateistruktur genommen werden.

Kommandozeile

<Pfad>\remotestage.exe /mode=data_archive /r=<remote> /archive=<archive file> /o=<output file>

Parameter

mode	data_archive = Einlesen des Datenarchivs vom Remote-Gerät
remote	Remote-Gerät IP-Adresse
archive file:	Dateiname des Datenarchivs Wenn diesem Parameter kein Dateiarhiv zugewiesen wird, speichert die RemoteStage alle Datenarchive in dem durch den Parameter "/o=" definierten Ordner.
output file:	Pfad und Dateiname des Datenarchivs bei der Abspeicherung im PC (optionaler Parameter) Wenn unter Parameter "/o=" keine Angabe erfolgt, wird im Programmverzeichnis (wo die RemoteStage.exe liegt) ein Ordner "UserData" erzeugt. Wenn der Dateiname Leerzeichen enthält, sollten diese mit " (Anführungszeichen) begrenzt werden.

Beispiel

Angenommen, die RemoteStage liegt unter C:\ Um beispielsweise ein Datenarchiv „Test-DB“ von der INSEVIS SPS mit der IP-Adresse 192.168.80.50 einzulesen und als csv-Datei nach „D:\archive\Testdaten.csv“ zu schreiben, wird folgende Kommandozeile benötigt:

Kommandozeile

C:\remotestage.exe /mode=data_archive /r=192.168.80.50 /archive=Test-DB /o=D:\archive\Testdaten.csv

Zurückschreiben (Download) der Daten (DB-) Archive in das Remote-Gerät

Hier werden die aus dem Remote-Gerät gelesenen (und danach am PC evtl. editierten) Daten-(DB-) Archive wieder in das Remote-Gerät zurückschrieben (Ab Rev. 1.0.4.17 und SPS-Firmware 2.3.9 (alle CPUs)).

Achtung: Für die Struktur der Daten (DB-) Archive ist der S7-Programmierer verantwortlich. Wenn diese geändert wird, müssen alle damit verbundenen Auswertungen mit geändert werden!

Kommandozeile

<Pfad>\remotestage.exe /mode=data_download /archive:=<archive file(s)>

Parameter

archive file(s)	S7-Nutzerdaten / Daten (DB-) Archiv(e) zum Zurückschreiben in das Remote-Gerät
------------------------	--

Beispiele

Angenommen, die RemoteStage liegt unter C:\ Nachstehende Befehlszeile überträgt den EINE Archivdatei "default_cfg.csv" aus dem Ordner D:\configs in das Remote-Gerät mit der IP-Adresse 192.168.80.50

C:\remotestage.exe /mode=data_download /archive=d:\configs\default_cfg.csv /r=192.168.80.50

Angenommen, die RemoteStage liegt unter C:\ Nachstehende Befehlszeile überträgt den ALLE Archivdateien aus dem Ordner D:\configs in das Remote-Gerät mit der IP-Adresse 192.168.80.50

C:\remotestage.exe /mode=data_download /archive=d:\configs*. * /r=192.168.80.50

Remotezugang mit der Software „RemoteStage“

Rückgabewerte

Rückgabewerte der remotestage.exe zeigen das Ergebnis der Operation

Rückgabewert	Bedeutung
0	Erfolgreich ausgelesen und gespeichert
1	Ungültiger Parameter oder wichtige Parameter fehlen
2	Projekt-Binärdatei ist ungültig oder wurde nicht gefunden
3	Remote-IP-Adresse ist ungültig
4	Konnte keine Verbindung aufbauen zum Remote-Gerät
5	SD-Karte nicht gesteckt beim Remote-Gerät
6	Archivdatei nicht gefunden beim Remote-Gerät
7	Remote-Gerät unterstützt Archivdatei-Einleseeservice nicht
8	Ungültige Archivdaten
9	Nicht genügend Systemressourcen
11	Nicht genügend Systemressourcen im Remote-Gerät
14	Konnte nicht schreiben in vorgegebene <Ausgabedatei>
15	Verbindung wurde unterbrochen

Beispiel einer Nutzung der RemoteStage in einem Batchprozess

```
@remotestage /mode=alarm_archive /r=192.168.80.50 /o=d:\archive\alarm.csv /v=d:\visu\myvisu.vsbm
@echo.
@goto res%ERRORLEVEL%

:res4
@echo Could not connect to remote device
@goto :EOF

:res0
@echo Successful read out and saved
@goto :EOF
```

Arbeiten mit der Software „ServiceStage“

Mit der kostenlosen Software „ServiceStage“ ist es via Ethernet möglich, die INSEVIS-CPUs online zu erkennen, deren Zustand und oder Programmdateien zu ändern, ein Backup/Restore durchzuführen sowie Know-how-Schutzstufen zu setzen. Die „ServiceStage“ Software ist auf den aktuell gängigen Windows Versionen lauffähig.



VIDEO-Tutorials verfügbar Sie finden detailliertere Erläuterungen in der ServiceStage-Playlist auf unserem YouTube®-Kanal „INSEVIS DE“.

Durch die „ServiceStage“ müssen für Updates dem Service oder anderen Ausführenden keine Quelldaten mehr zur Verfügung gestellt werden. Damit kann ein Service-Mitarbeiter ohne teure Lizenzen die wesentlichen Funktionen für Diagnose und Update direkt von seinem PC/LapTop aus durchführen.

Dazu wählt man den Netzwerkadapter des PCs aus, über den auf das Netzwerk und damit auf die CPUs zugegriffen. Man wählt eine der im linken Fenster angezeigten CPUs aus und drückt „Aktualisierung“. Mit einem „LED Blinktest“ kann man die ausgewählte CPU identifizieren

In diese Maske kann die IP-Adresse des verbundenen Gerätes geändert werden (Zuweisen der IP-Adresse)

Funktionsübersicht

Allgemeine Informationen

Ansicht der nicht änderbaren Systemdaten wie Seriennummer, Firmwareversion, MAC-Adresse sowie der änderbaren Daten wie

- IP-Adresse,
- Netzmaste und
- Router-IP-Adresse

sowie eine Blinktest-Möglichkeit der ausgewählten CPU.

Nur bei Geräten mit CPU-T:

Firmwareupdate per Ethernet möglich (Firmware vorab bei INSEVIS anfordern)

Betriebszustand

Hier können die CPU-Zustände RUN/STOP geändert werden (nicht bei Panels-HMI). Diese Änderung verlangt eine extra Bestätigung.

Datum und Uhrzeit

Zur manuellen Einstellung von Systemdatum und Uhrzeit.

Durch Anhängen der Funktion „Übernehmen vom PC“ werden aktuelles Datum und Uhrzeit vom PC übernommen, wenn „Anwenden“ gedrückt wird.

Arbeiten mit der Software „ServiceStage“

Speicher

Anzeige der Auslastung von Lade- und Arbeitsspeicher (nicht bei Panels-HMI).

Die manuell ausführbare Funktion „**Speicher komprimieren**“ sollte bei oftmaligem Laden von S7-Programmen verwendet werden, um den damit belegten Speicher wieder freizugeben.

Die ebenfalls manuell ausführbare Funktion „**RAM nach ROM kopieren**“ kopiert den Ladespeicherinhalt (z.B. Prozessdaten) vom batteriegepufferten Speicher in das Onboard-Flash, um Datenverluste zu verhindern.

Die Funktion „**Urlöschen**“ löscht sämtliche Anwenderdaten im Lade- und Arbeitsspeicher. Das Betriebssystem bleibt erhalten.

Diagnosepuffer

Hier werden die jeweils letzten 100 Einträge in den Diagnosepuffer nach Datum und Uhrzeit angezeigt (nicht bei Panels-HMI).

Auf Wunsch erfolgt das durch Anhaken der Funktion „**Ereignisinformation hexadezimal ausgeben**“ auch in einem hexadezimalen System.

Durch „**Aktualisieren**“ werden inzwischen angefallene Ereignisse neu eingetragen, mit „**Speichern unter**“ werden die Ereignisse in eine Textdatei (*.txt) abgelegt.

Block Update

Diese Funktion ermöglicht es,

- das S7-Programm als MemoryCard-file

.wld (nicht bei Panel-HMI).

- die Visualisierung als Binärfile

*.vsbin und

- die Konfigurationsdaten als

*.csbin (nicht bei Panel-HMI).

über das Netzwerk einzeln in die CPU zu laden.

Falls in dem Hardwarekonfigurationsbaustein des Zielgerätes eine andere IP-Adresse eingetragen sein sollte, als in den Konfigurationsbausteinen der übertragenen Böcke, kann es dazu führen, dass nach dem Übertragen die Verbindung abbricht. Um das zu vermeiden, wird ab Rev. 1.0.1.9 gefragt, ob die „Vor Ort“-Adresse im Zielgerät überschrieben werden soll.



Hinweis: * Was ist eine WLD-Datei?

Die WLD-Datei ist eine Datei mit S7-Systemdaten (SDBs) und S7-Programmblocken (FCs, FB, OBs, DBs, ...). Nach dem Anlegen einer WLD-Datei im SimaticManager/TIA-Portal (unter dem Menue: „Datei → Memory Card Datei → Neu“) werden BELIEBIGE S7-Daten und Systemdaten in das Fenster der S7-Memory-Card-Datei mit Drag'n Drop gezogen (kopiert).

→ Man kann also mit der WLD-Datei GANZE S7-Projekte ODER EINZELNE DBs mit / ohne ihren Systemdaten übertragen.



ACHTUNG: Wenn man im SimaticManager/ TIA-Portal Systemdaten mit in die wld-Datei einfügt,

- sollten die aus einem Gerät mit bereits enthaltener ConfigStage-Konfiguration zurückgelesen worden sein (Menue: „Zielsystem → Station Laden in PG“), denn dann sind die ConfigStage-Konfiguration mit darin enthalten)
- die SDBs des SimaticManagers/ TIA-Portals überschreiben die ConfigStage-Systemkonfiguration. In diesem Fall muss die *.csbin-Datei des Projektes nach der wld-Datei erneut aufgespielt werden.

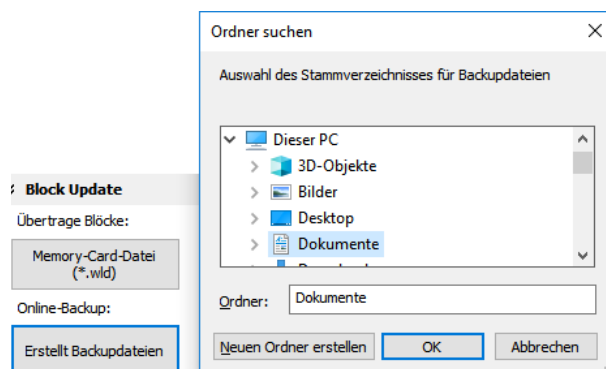
Arbeiten mit der Software „ServiceStage“

Backupdateien erstellen (Online Backup)

Diese Funktion erstellt auf dem ServiceStage-Rechner einen Ordner „Backups“ mit

- einer wld*-Datei „S7programs.wld“ Systemdaten und allen OB, DB, FB, SFB und SFC und
- einer Visualisierungs-Binärdatei eine „Visudata.vsbin“ (wenn das vorab in der VisuStage vorab der Upload erlaubt wurde, siehe unten)

D.h. mit diesen Daten kann einfach ein Backup des bestehenden Systems erstellt und mit dem Block Update wieder in das verbundene Gerät zurückgespielt werden (Online mit der ServiceStage oder per Micro-SD-Karte), mehr unter „Systemfunktionen“, Kapitel „Datensicherung und -wiederherstellung“



ACHTUNG

Bei Verwendung für Backup / Restore

- Ab der Firmware 2.3.6 kann ein Restore nur noch mit WLD- und VSBIN-Dateien erfolgen, die CSBIN-Datei wird beim Einlesen verworfen. D.h., die WLD-Datei muss zwingend auch die Systemdaten enthalten, die vorher mit der CSBIN separat nachgeladen werden konnten.
- Sind die Systemdaten nicht erhalten, bleibt das mit diesem restore geladene Programm wegen fehlender Daten stehen und die SPS geht in Stop.



ACHTUNG

- Um das Visualisierungs-Backup zu erstellen, muss in der VisuStage (unter Projekt) die Upload-Funktion erlaubt werden, ggf. wird eine PIN-Eingabe gefordert. Zusätzlich kann ggf. auch ein für die Kommunikation vergebenes Passwort (SimaticManager/ TIA/ ConfigStage) abgefragt werden.
- Wenn die Schutzstufe „Leseschutz“ gesetzt wurde, ist kein Online Backup möglich.

Know-how-Schutz

Hier können verschiedene Stufen zum Schutz des S7-Programmes gegen Auslesen oder Überschreiben gesetzt werden. Dazu wird die SPS in STOP geschaltet.

Know-how-Schutz		
Schutzstufe	Lesezugriff	Schreibzugriff
☐ kein Schutz	✓	✓
☒ Leseschutz	✗	✓
☐ Schreibschutz	✓	✗
☐ Schreib-/ Leseschutz	✗	✗
Schutzstufe setzen		

Kein Schutz: Die Programmbausteine können von jedem gelesen und geändert werden. Die Schutzstufe kann erhöht werden, indem in der „ServiceStage“ eine andere Stufe zugewiesen wird. Eine Authentifizierung per Passwort wird dann verlangt, wenn das im Simatic-Manager oder TIA-Portal so konfiguriert wurde. Damit kann nur berechtigtes Personal diesen Schutz aktivieren.

Leseschutz: Die S7-Programmteile können von jedem geändert aber nicht mehr gelesen werden. Die Panel-HMIs können die Prozessdaten (DBs) weiterhin auslesen. Diese Schutzstufe kann mit der PG-Funktion „Clear All“/ Utlöschen (auch mit ServiceStage) zurückgesetzt werden (danach ist die SPS gelöscht) und man spielt das ungeschützte Original-S7-Programm wieder in die SPS.

Schreibschutz: Die S7-Programmteile können von jedem gelesen aber nicht mehr geändert werden. Diese Schutzstufe kann nur noch mit Hardware-Utlöschen direkt am Gerät zurückgesetzt werden, danach ist das ungeschützte Original-S7-Programm neu in die SPS zu übertragen.

Schreib-/Leseschutz: Die S7-Programmteile können weder geändert noch gelesen werden. Die Panel-HMIs können die Prozessdaten weiterhin auslesen. Diese Schutzstufe kann nur noch mit Hardware-Utlöschen direkt am Gerät zurückgesetzt werden, danach ist das ungeschützte Original-S7-Programm neu in die SPS zu übertragen.

Arbeiten mit der Software „ServiceStage“

Beobachtungstabelle

Zum schnellen Beobachten von Variablen steht ab der ServiceStage V 1.0.3.3 die Funktion **Beobachtungstabelle** zur Verfügung.

Mit dieser können einzelne Werte beobachtet werden. Ebenso können diese Werte auch gesteuert werden. Es können auch Variablen (*.vartab Dateien) importiert werden, so können auch größere Mengen an Daten schnell beobachtet werden, oder für schnelle Funktionstests genutzt werden.

Alle Funktionen der Beobachtungstabelle werden bei Verweilen des Mauszeigers über der Schaltfläche mit einem Tooltip beschrieben.

Beobachtungstabelle				
	Adresse	Anzeigeformat	Beobachtungswert	Steuern deaktivieren Steuerwert
1	M128.0	BOOL	true	☐ false
2	M1.0	BOOL	true	☐ true
3	MW12	DEZ (+/-)	55	☐ 10
4	MW7	DEZ (+/-)	111	☐ 11



Hinweis: Die ServiceStage als schnelles Beobachtungstool

Parallel zu anderen Programmen kann man mit der ServiceStage ausgewählte Variablen beobachten. Dies funktioniert sehr performant und als kleines Fenster. Eine Vereinfachung, wenn man Daten aus mehreren Programmen gleichzeitig im Auge behalten muss.

INSEVIS Vertriebs GmbH

Am Weichselgarten 7
D - 91058 Erlangen

Fon: +49(0)9131-691-440
Fax: +49(0)9131-691-444
Web: www.insevis.de
E-Mail:

Zertifiziert nach DIN EN ISO 9001:2015