INSEVIS

S7-Panel-SPS Handbuch

S7-Panel-SPS

S7-Kompakt-SPS

S7-Panels/HMI

Peripherie

Software

Energiemanagement

Inhaltsverzeichnis

Allgemeine Hinweise ... 8
Über INSEVIS ... 9
Produktfamilie S7-Panel-SPS ... 10
- Kommunikationsübersicht ... 10
- S7-CPUs in den Panel-SPS ... 11
- CPU-V und CPU-P ... 11
- CPU-T ... 12
- Bedieneinheiten in den Panel-SPS ... 13
- Generation I → ab Q3/2016 auch im Design der Generation II ... 13
- Generation II ... 13
- Peripherie für die Panel-SPS ... 14
- Kommunikation SPS zu Peripherie ... 14
- Zubehör S7-Panel-SPS ... 16
- Beschaltung der CPU-V/P ... 17
- Beschaltung der CPU-T ... 18
- S7-Panel-SPS Schaltschrankausschnitte ... 19

Allgemeines ... 22
- Speicherorganisation ... 22

Kommunikation ... 23
- Ethernet ... 23
  - IP-Adresse im SIMATIC-Manager einstellen ... 23
  - Anlegen der CPU und der Ethernet-Verbindung im SIMATIC-Manager ... 24
  - IP Konfiguration der SPS mit dem SFB 129 "IP_CFG" ... 25
  - Modbus TCP (Server) ... 25
  - S7 Aktive Verbindungen zu Partner-SPSen (RFC1006) ... 26
    - Daten lesen von einer Partner-SPS mit SFB 120 "GET" ... 26
    - Daten schreiben an eine Partner-SPS mit SFB 121 "PUT" ... 28
  - TCP Send/Receive-Verbindung (RFC 793) ... 30
    - Daten senden über TCP mit dem SFB 122 "TSEND" ... 30
    - Daten empfangen über TCP mit dem SFB 123 "TRECV" ... 32
    - Beenden der TCP-Verbindung mit dem SFB 124 "TDISCON" ... 34
    - Konfiguration der Partnerparameter bei TCP mit dem SFB 125 "TCONFIG" ... 35
    - Statusabfrage bei der TCP-Verbindung mit dem SFB 126 "TSTATUS" ... 36
  - UDP Send/Receive-Verbindung (RFC 768) ... 37
    - Daten senden über UDP mit dem SFB 127 "USEND" ... 38
    - Daten empfangen über UDP mit dem SFB 128 "URECV" ... 39
- Profinet IO Controller ... 42
  - Konfiguration ... 42
- UART ... 44
  - Senden von Daten mit SFB 60 "SEND_PTP", freies ASCII-Protokoll ... 44
  - Empfang von Daten mit SFB 61 "RCV_PTP", freies ASCII-Protokoll ... 45
  - Senden von Daten mit SFB 60 "SEND_PTP", Modbus RTU (Client) ... 46
  - Empfang von Daten mit SFB 61 "RCV_PTP", Modbus RTU (Client) ... 47
  - Modbus RTU (Server) ... 48
  - Empfang über RS232 mit Endeerkennung ... 49
  - Bearbeiten der UART-Konfiguration mit SFB 130 "CFG_PTP" ... 49
- Profibus DP ... 50
  - Profibus DP-Master ... 50
  - Konfiguration des DP-Master ... 50
  - Profibus DP-Slave ... 51
    - Konfiguration des DP-Slaves ... 51
    - DP Master Konfiguration für INSEVIS Profibus DP-Slave ... 54
    - Struktur der DP-Slave Diagnosetelegramme ... 55
    - DP-Slave Adressierung mit SFC 75 „SET_ADDR“ ... 58
- CAN ... 59
- CANopen®-Konfiguration mit der ConfigStage ... 59
- Freie CAN-Programmierung (Expertenlevel) ... 60
  - Senden von CAN-Telegrammen SFB105 ... 60
  - Empfang von CAN-Telegrammen SFB106 ... 60
  - Empfangsereignis OB106 ... 60
- CANopen - Hilfsfunktinen (utilities) ... 61
  - Lesen und Schreiben von CANopen-Variablen (Objekten) per SDO-Transfer SFB107 ... 61
    - FB107 SDO upload ... 61
    - FB108 SDO download ... 61
    - FB109 SDO upload ... 61

FB110 SDO download 62
FB111 SDO upload 62
FB112 SDO download 62
FB113 SDO Transfer String 62
Alternative Funktion des SFB107 62
CAN Statusabfrage SFB114 63
Fehlerereignis OB186 63
Automatische Wiederkehr 63
Konfiguration mit der Software „ConfigStage“ 65
Grundlegende Einstellungen 65
Änderung der Geräte-IP-Adresse 66
Wechsel des Zielgerätes 66
Adressierung der Onboard-Peripherie 66
Standardadressierung in den INSEVIS- SPSen 66
Abweichende Adressierungen in SPSen und Adressierungen von dezentraler Peripherie 66
CPU-Einstellungen 67
Kommunikationseinstellungen 69
RS232 und RS485 69
Ethernet 69
Profinet 71
CAN-Einstellungen 71
Dezentrale INSEVIS Peripherie 71
Dezentrale Fremdperipherie manuell konfigurieren 72
Bibliothekselemente aus eigenen CAN-Slaves erstellen 73
Dezentrale Fremdperipherie per EDS konfigurieren 74
Dezentrale Fremdperipherie - vordefiniert 75
Allgemeine Beispiele für CANopen Fremdgeräte 76
Systemfunktionen 78
Organisationsbausteine (OB) 78
Organisationsbausteine (OB) - kompatibel zu STEP®7 von Siemens 78
Organisationsbausteine (OB) - in Erweiterung zu STEP®7 von Siemens 78
Systemfunktionsbausteine (SFB) 78
Systemfunktionsbausteine (SFB) - kompatibel zu STEP®7 von Siemens 78
Systemfunktionsbausteine (SFB) - 78
die bei der INSEVIS-CPU noch NICHT vorhanden und nur auf Anfrage erhältlich sind 78
Systemfunktionsbausteine (SFB) - in Erweiterung zu STEP®7 von Siemens 79
Systemfunktionen (SFC) 79
Systemfunktionen (SFC) - kompatibel zu STEP®7 von Siemens 79
Systemfunktionen (SFC) - 80
die bei der INSEVIS-CPU noch NICHT vorhanden und nur auf Anfrage erhältlich sind 80
Systemfunktionen (SFC) - in Erweiterung zu STEP®7 von Siemens 80
Import der INSEVIS-Zusatzbausteine in das S7-Projekt 81
Installation der INSEVIS-S7-Bibliothek im Simatic®-Manager 81
Installation der INSEVIS-S7-Bibliothek im TIA-Portal® 82
Übersicht über SFCs zur Datensicherung und -wiederherstellung 83
Datenbausteine nach ROM/ ins Flash sichern 83
Life-Backup für Geräteaustausch 83
Manuelles Programm-Backup für Gerätehersteller 83
S7-Programm-Restore für beide Backup-Methoden 83
Daten archivieren /dearchivieren (zurücklesen) 84
Daten zurücklesen aus am PC editierter CSV-Archivdatei 84
Archivieren von Daten mit dem SFC 207 "ARCHIVE“ 84
Backup mit dem SFC 208 "BACKUP“ 88
Achivdaten lesen von Micro-SD-Karte mit SFC 209 "READFILE" 90
DB kopieren vom Arbeits- in den Ladespeicher mit SFC 210 "COPY_DBL" 91
DB kopieren vom Lade- in den Arbeitsspeicher mit SFC 211 "COPY_DBW" 93
Daten lesen aus csv-formatierter Archivdatei (SD) mit SFC 213 "READ_CSV" 95
LCD-Hinterleuchtungseinstellung mit dem SFC 214 "LCD_BRIG" 97
Von RAM nach ROM kopieren mit dem SFC 254 "RAM2ROM" 99
Betriebssystem in der SPS updaten 100
Hardwareseitiges Urlöschen 101
Visualisierung mit der Software „VisuStage“ 102
Download und Lizenzierung 102
Kommunikation zwischen SPS und abgesetztem Panel 102
Generelle Einstellungen der Programmoberfläche 103
Generelle Funktionsweise 104
Empfohlene Vorgehensweise 104
Ressourcen schonen 104
Übersicht behalten, Projekt bereinigen 104
Standardseiten als Projektvorlagen erstellen 104

Projekteinstellungen ... 105
Angabe der Ziel-IP-Adresse ... 105
Dateiformate ... 105
Datenschutz bei Upload ... 105
VNC-Server (nur CPU-T-Geräte) ... 105
Einrichten eines VNC-Viewers/ Clients ... 106
Bildschirmauflösung oder -drehung ändern ... 108
Anlage und Verwaltung von Ressourcen ... 108
Ressource Sprachen ... 108
Ressource Variablen ... 108
Multiplexvariablen ... 113
Ressourcen Texte und Textlisten ... 114
Ressourcen Bilder / Bildlisten ... 116
Ressourcen Meldungen (Alarme / Ereignisse) ... 118
Ressource Benutzerverwaltung ... 119
Ressource Partner-SPS (nur für HMI) ... 120
Partner-Gerät synchronisieren (nur für HMI) ... 121
Ressource Dialoge ... 121
Ressource Menuevorlagen ... 121
Ressource Trends ... 122
Starten der Trenderfassung mit dem SFC 202 "TRENDSRT" ... 123
Stoppen der Trenderfassung mit dem SFC 203 "TRENDSTP" ... 125
Fortsetzen der Trenderfassung mit dem SFC 204 "TRENDCNT" ... 126
Trendstatusabfrage mit SFC 205 "TRENDSTA" ... 127
Ressource Rezepturverwaltung ... 129
Rezepturhandling mit SFC 206 "RECIPE" ... 131
Allgemeine Funktionen ... 133
Abfrage der aktuellen Bildschirm-ID-Nummer mit dem SFC 200 "SCR_GET" ... 135
Bildschirmwechsel mit dem SFC 201 "SCR_SET" ... 136
Statische Objekte ... 137
Dynamische Objekte ... 138
diverse Tastenarten ... 138
Touchflächen ... 140
Zustandsflächen ... 141
Auswahl- und Markierungsfelder ... 142
E/A-Felder ... 143
Datumsfelder ... 144
Fortschrittsbalken ... 145
Betrachter ... 146
Bild- und Textlisten ... 146
Rezepturanzeige ... 147
Meldungsanzeige und -archivierung ... 151
Trendanzeige und Trendarchivanzeige ... 153
Funktionskurve ... 154
Systemdaten (IP-Adresse, Sprachen etc.) ... 156
Passwortfelder ... 157
Arbeiten mit Bibliothekselementen ... 158
Erstellen von Bibliothekselementen ... 158
Verwenden von Bibliothekselementen ... 158
Simulation und Download ... 159
Fehlerbereinigung ... 159
Simulation und Download ... 160
Remotezugang mit der Software „RemoteStage“ ... 161
Allgemeines ... 161
Auswahl der Remote-Steuerung ... 162
Benutzen und Einstellen des Remote-Bildschirmes ... 162
Abspeichern der remote (auf dem PC) archivierten Daten ... 162
Einlesen, Konvertieren und Abspeichern der Archivdaten ... 163
Ereignis- oder Alarmarchivdarstellung und -export ... 163
Trenddarstellung und -export ... 164
Rezepturhandling ... 165
Datenarchivhandling ... 166
Kommandozeilenbefehle für RemoteStage - Visualisierungsmodus ... 167
Automatisches Verbinden zum Remote-Panel nach Start ... 167
Einstellen der TCP-Portnummer für die S7-Kommunikation ... 167
Vollbildschirm nach Start ... 167
Kommandozeilenbefehle für RemoteStage - Archivierungsmodus ... 168
Lesen (Upload) der Meldungs- (Alarme und Ereignisse) und Trendarchive ... 168
Lesen (Upload) der Rezepturarchive vom Remote-Gerät ... 169
Zurückschreiben (Download) der Rezepturarchive in das Remote-Gerät ... 170

Lesen (Upload) der Daten (DB-) Archive vom Remote-Gerät....171
Zurückschreiben (Download) der Daten (DB-) Archive in das Remote-Gerät....171
Rückgabewerte....172
Arbeiten mit der Software „ServiceStage“....173
Funktionsübersicht....173
Allgemeine Informationen....173
Betriebszustand....173
Datum und Uhrzeit....173
Speicher....174
Diagnosepuffer....174
Know-how-Schutz....175
Block Update....175
Online Backup....176

Änderungen zu vorhergehenden Versionen des Handbuches

Rev. 2 / 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. 3 / 2012:
neu: Kapitel Rezepturverwaltung und Rezepturanzeige
geändert: SFB60/61 (UART) Parameterbezeichnung

Rev. 4 / 2012:
neu: Kapitel Remotezugang mit Software „RemoteStage“

Rev. 5 / 2012:
neu: Kapitel Menues (Bildschirmschoner, Putzbildschirm, etc) bei VisuStage

Rev. 6 / 2012:
geändert: Kapitel Alarmhandling, Beschreibung SFBs 122/127 korrigiert, ConfigStage und RemoteStage überarbeitet, VisuStage komplett überarbeitet, jetzt mit Korrespondenz zu Demovisu_2012_04

Rev. 7 / 2012:
geändert: VisuStage-Funktionen neu: Tasten mit Text/Farbumschlag, Einfärben und Begrenzen von E/A-Feldern, etc (ab VisuStage 2.0.1.4, mit Korrespondenz zu Demovisu_2012_04_1, ab BS 2.0.31 (Panel-SPS)

Rev. 8 / 2012:
geändert: Beschreibung der Profibus-Signale an Siemens-Beschreibung angepasst, Zykluszeitenbeschreibung erweitert (Kapitel „ConfigStage“), RemoteStage mit Archiveinlesen über Ethernet, Verbesserungen in der Rezeptur-verwaltung und der Objektanordnung
(ab VisuStage 2.0.1.5, mit Korrespondenz zu Demovisu_2012_04_2, ab BS 2.0.33 (Panel-SPS)

Rev. 1 / 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. 2 / 2013:
neu: ServiceStage eingefügt

Rev. 3 / 2013:
neu: Speicherorganisation beschrieben, CAN überarbeitet, neue Beschreibung für VisuStage 2.0.2.0 (BS2.0.37) mit Buzzer, mehr Rezepten, Alarmen, etc, jetzt mit Korrespondenz zu Demovisu_2013_01

Rev. 4 / 2013:
neu: Neue Beschreibung für VisuStage 2.0.2.2 / RemoteStage 1.0.3.3 (BS 2.0.38) mit Texttasten und Watchdog, etc, jetzt mit Korrespondenz zu Demovisu_2013_02

Rev. 5 / 2013:
neu: Neue Beschreibung für VisuStage 2.0.2.3 / RemoteStage 1.0.3.4 (SPS-BS 2.0.39, HMI-BS 1.0.7) mit Farbumschlag bei Texten, Bestätigungsabfragen bei Rezepturen, Sortierungen bei Meldearchiven, einstellbare Fontgrößen und Teilungen in Fortschrittsbalken und Trendarchiven, neuer Downloadmechanismus, E-/A-Felder mit Integer-Eingabe mit Komma, jetzt mit Korrespondenz zur Demovisu_2013_04

Rev. 6 / 2013:
geändert: Beschreibung von ConfigStage: PDO-/ SDO-Mapping, Adressübersichtsliste (ab ConfigStage 1.0.14.7)
Beschreibung VisuStage: Benutzung der SFC201-5 von SPS aus (bei abgesetztem Panel)
(ab SPS-BS 2.0.40 und HMI-BS 1.0.9)

Rev. 7 / 2013:
neu: Neue Remote-Funktionen; Rezepturdatensätze auslesen/ speichern (SFC206), (DB-) Archivierungen auslesen/ speichern (SFC207), Einlesen der Visualisierung-Binärdaten via Ethernet direkt von Panel, ab RemoteStage V1.0.3.7, ab SPS-BS 2.0.41

Rev. 8 / 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
geändert: Beschreibung CAN SFB114 ergänzt

Rev. 9 / 2013:
neu: VisuStage 2.0.3.1: Bibliotheks- und Gruppierfunktionen, RemoteStage 1.0.4.0: PIN-Code bei Binary-Upload ab SPS-Firmware 2.1.1
geändert: Beschreibung S7-Lib in Projekt einbetten in Systemfunktionen verschoben

Rev. 10 / 2013:
neu: VisuStage 2.0.3.4 / RemoteStage 1.0.4.2: Drehung der Bildschirminhalte möglich, Filtern von Variablen, Zusatzinfos über Trigger-und Bestätigungsvariablen bei Meldungen, ab Panel-SPS Betriebssystem 2.1.3

Änderungen zu vorhergehenden Versionen des Handbuches

Rev. 11 / 2013:
neu: VisuStage 2.0.3.5: Querverweis-Funktion und Anzeige der Listenressourcen bei Bild-/Textliste.
ConfigStage 1.0.14.10: Wechsel des Zielgerätes eingebracht
geändert: Beschreibung PIN-Änderung und UART SFB60/61 verbessert

Rev. 01 / 2014:
neu: VisuStage 2.0.3.6 und RemoteStage 1.0.4.3 : Raster/Fang, Menüvorlage/Masterfolie, Funktion „Menuewechsel mit PIN-Eingabe“, E/A-Felder: Eingabeformate wie bei Taschenrechner (mit „Komma“-Taste) möglich
ab SPS-Firmware 2.1.4, HMI-Firmware 1.1.4

Rev. 02 / 2014:
neu: VisuStage 2.0.3.9 und RemoteStage 1.0.4.5 : Funktionskurve neu, virtuelle Tastaturen erweitert, Demo 2014_02
SPS-Firmware 2.1.5: SFC209 READFILE, SFC2010 LOAD_DBL, SFC2011 LOAD_DBW neu
geändert: Struktur im Kapitel VisuStage verbessert, Backup&Restore- Zusammenfassung bei Systemfunktionen erstellt

Rev. 03 / 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. 04 / 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. 05 / 2014:
geändert: VisuStage: neue Such-, Übersichts- und Aufräumfunktionen ab 2.0.4.0

Rev. 06 / 2014:
neu: VisuStage 2.0.4.2: neue Funktion: Benutzerlevel-Auswahl vor PIN-Eingabe (mit Firmware 2.1.9 und Remote Stage 1.0.4.7)
geändert: verbesserte Beschreibung Verbindungskonfiguration Panel ↔ SPS

Rev. 01 / 2015:
neu: VisuStage 2.0.4.4: Lean-Version nach 30Tagen, neue Funktion: TSAP-Konfiguration, IP-Feld farblich ändern, hexadezimale Werte eingeben in E/A-Feld, ConfigStage 1.0.14.15 neue Funktion: TSAP-Konfiguration (mit Firmware 2.2.1)

Rev. 02 / 2015:
neu: VisuStage 2.0.4.5: Einfärben (Gestaltung) der Meldungstexte / Hintergründe in den Meldungsbetrachtern
(ab Panel-SPS 2.2.3, Panel.HMI 1.1.9) remote anzeigbar mit RemoteStage 1.0.4.10

Rev. 03 / 2015:
geändert: Systemfunktionen: Erweiterung der Übersicht OBs, SFBs, SFCs, Erläuterung zur Erstellung WLD-Datei ausführlicher
UART: Empfang RS232 mit Ende-Erkennung beschrieben

Rev. 04 / 2015:
neu: CPU-T-Geräte mit aufgenommen, Anpassungen (Erweiterungen) für CPU-T-Funktionen vorgenommen,
geändert: CAN: SFB107 + automatische Wiederkehr beschrieben, neue VisuStage-Funktionen bei Bild, Rechteck, Text, Bargraph, neue ServiceStage-Funktion Firmwareupdate mit ETH, + WLD-Erstellen beschrieben
gilt ab Firmware 2.3.0 (CPU-V/P/T), ConfigStage 1.0.14.21, VisuStage 2.0.4.7, ServiceStage 1.0.1.5, RemoteStage 1.0.4.13

Rev. 01 / 2016:
neu: Profinet-Handling (bei CPU-T-Geräten) eingefügt, VisuStage 2.0.5.0: Variablenimport vom TIA-Portal
geändert: Systemfunktionen: SFC208-Beschreibung verbessert, Bild eingefügt in manuelles Backup, allgemeine Fehlerbereinigung
ab Firmware 2.3.5 (CPU-V/-P) und 2.3.7 (CPU-T)

Rev. 02 / 2016:
neu: SFC214 „LCD_BRIG“ für Regelung der Helligkeit durch S7-Programm
geändert: Systemfunktionen: manuelles Backup Beschreibung geändert, VisuStage E/A-Felder Info zu Skalieren eingefügt

Rev. 03 / 2016:
neu: neue Up- und Downloads im Kommandozeilenmodul der RemoteStage, Online-Backup bei ServiceStage
ab SPS-Firmware 2.3.9 (alle CPUs)

Rev. 04 / 2016:
neu: VNC-Server in CPU-T-Panels (und Panel-SPSen), Umschaltung Sprache/Bildschirm per Variable
Zeitsynchronisation mit Partner-Gerät
ab VisuStage 2.0.5.2 und SPS-Firmware 2.4.0 (CPU-T) oder HMI-Firmware 1.2.9 CPU-T
geändert: RemoteStage: Beschreibung bei Download Rezepturen und Datenarchive in Remote-Gerät verbessert,
ServiceStage: Beschreibung Online Backup erweitert

Rev. 05 / 2016:
neu: VNC-Viewer in VisuStage beschrieben, ab VS 2.0.5.2, Firmware 2.4.0 (nur CPU-T)
geändert: Systemfunktionen: Beschreibung Hardware-Reset erweitert, ServiceStage: Beschreibung Online Backup nochmal erweitert

Rev. 06 / 2016:
neu: Multiplexvariablen in VisuStage ab VS 2.0.5.3, Firmware 2.4.3 (nur CPU-T),
IP-Rückfrage bei Download in ServiceStage ab V 1.0.2.0, 2.4.3 (nur CPU-T) / Firmware 2.4.0 (CPU-V/P)
geändert: Beschreibung E/A-Felder und Datumsfelder getrennt und verbessert

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 zentralem 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 weißt 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

Für das Verständnis dieser Dokumentation sind grundlegende Kenntnisse der Automatisierungstechnik allgemein sowie der Programmierung unter STEP®7 nötig.

Geltungsbereich des Handbuches

Diese Dokumentation gilt für alle S7-Panel-SPSen der INSEVIS Serie „PCxxxx“ mit den CPU-V/P ab Hardware-AS 3.0 und Firmware 2.0.0. und mit den 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-CPUs -V und -P bilden die Basis für die erfolgreichen INSEVIS Produktfamilien mit Profibus DP Master/Slave.
Mit der neuen S7-CPU-T stehen erstmals Panel-SPS und Panel-HMI der *Gereration II* mit Profinet IO Controller zur Verfügung.

Step®7-Programmierbarkeit

Die INSEVIS-S7-CPUs 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.

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.

Certificato Certyfikat Certificate Certificado Certificat

Zertifikat

TAW Cert

Die Zertifizierungsstelle TAW Cert
Zertifizierungsgesellschaft mbH für QM-Systeme und Personal

bescheinigt hiermit, dass das Unternehmen

INSEVIS
Gesellschaft für industrielle Systemelektronik und Visualisierung mbH
Am Weichselgarten 7
91058 Erlangen

für den Geltungsbereich
Entwicklung, Service und Vertrieb von Systemelektronik und Software für industrielle Automatisierungslösungen

ein Qualitätsmanagementsystem eingeführt hat und anwendet.

Durch ein Audit am 31.07.2014 wurde der Nachweis erbracht, dass die Anforderungen der
DIN EN ISO 9001
(Ausgabe 2008)
erfüllt sind.

Erstzertifizierung: 14.09.2011

Dieses Zertifikat ist gültig bis 13.09.2017

Zertifikat-Registrier-Nr.: 19010511

DAkkS

Altdorf, den 14.09.2014

Dr.-Ing. Dietmar Winner, Geschäftsführer TAW Cert GmbH

TAW Cert GmbH • Fritz-Bauer-Str. 13 • 90518 Altdorf

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

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.

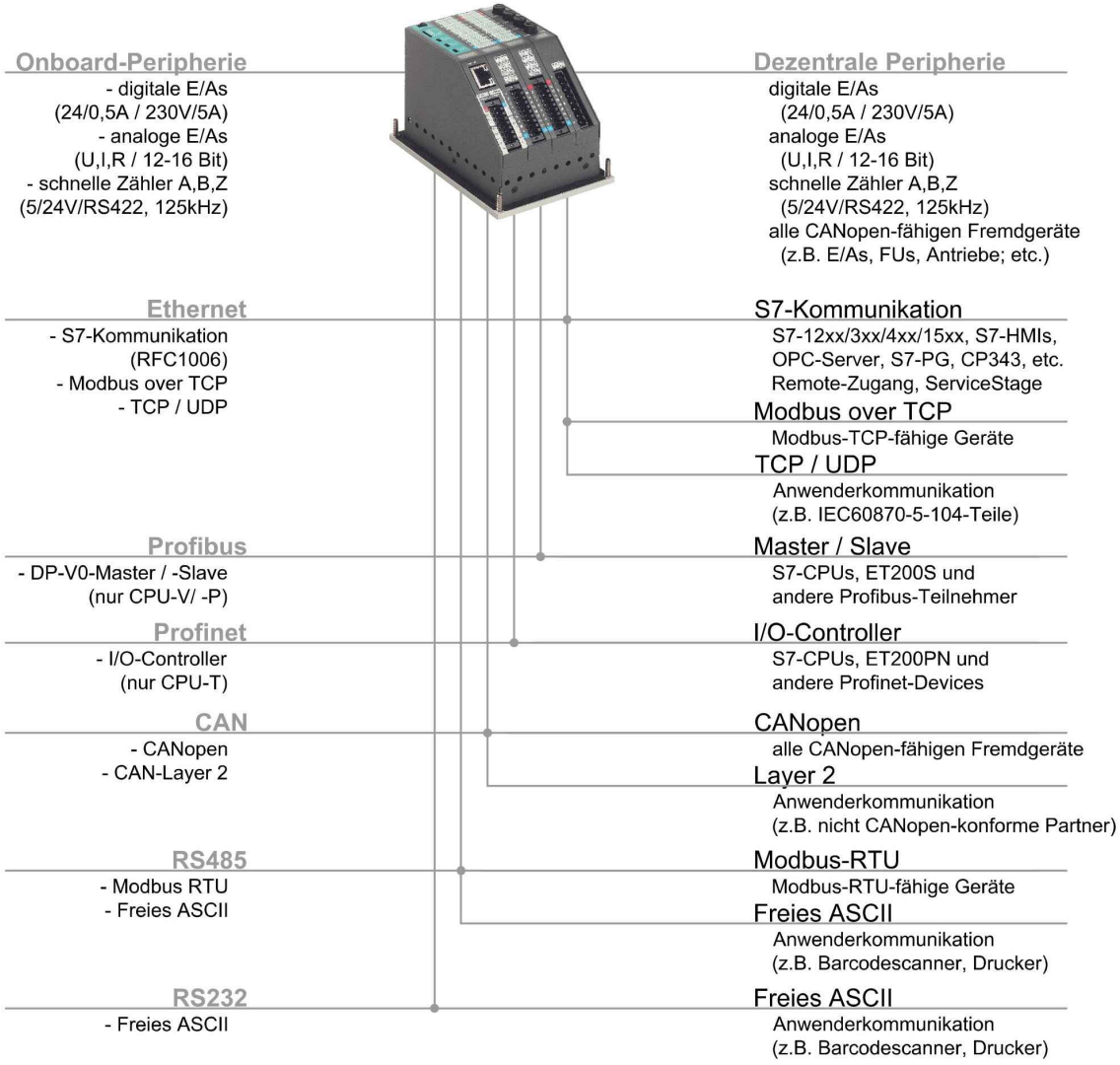
Sämtliche Produkte wurden umfangreichen Tests unterzogen, bevor sie in ausgewählten und zertifizierten Fertigungen in Deutschland hergestellt werden.

INSEVIS - Made in Germany

Produktfamilie S7-Panel-SPS

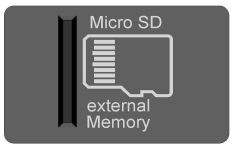
Kommunikationsübersicht

INSEVIS-S7-Panel-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 Panel-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 Störmeldungen, von Daten aus DBs, von Trends sowie zur Rezepturverwaltung bzw. zum Restore nötig und kann auch zum S7-Programm- und Firmwareupdate genutzt werden.



Wichtigste Eigenschaften in Stichpunkten

S7-Programmierung

Bestehende Siemens-S7-Programmierumgebung 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.

Backup & Restore

Einfach alle Daten sichern; Anwenderprogramm, Prozessdaten, Visualisierung und Archive - passwortgeschützt als eine Binärdatei zum Aufspielen auf ein baugleiches Gerät. Das arbeitet mit diesen Daten genau dort weiter, wo das alte Gerät aufgehört hat.

Datenarchiverung

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.

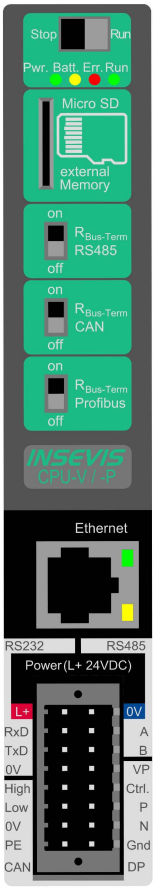
Gateway-Funktion

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-CPUs in den Panel-SPS

CPU-V und CPU-P

Geräte mit **CPU-V** und **CPU-P** sind ideal geeignet für kleine Anwendungen im LowCost-Bereich von Panel-SPSen mit hochwertiger Visualisierung (Typ V für 3,5 bis 5,7“ und Typ P bevorzugt für 7 bis 10,2“) und 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) 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	CPU-P
Arbeitsspeicher (davon akkugepuffert)	512kB 256 kByte remanent	640kB 384 kByte remanent
Ladespeicher	2MB Flash (nicht flüchtig)	2MB Flash (nicht flüchtig)
Visualisierungsspeicher	4MB Flash (nicht flüchtig)	24MB Flash (nicht flüchtig)
externer Speicher	Micro SD, bis max. 8 GB	Micro SD, bis max. 8 GB

Wichtigste Eigenschaften in Stichpunkten

S7-Programmierung

Bestehende Siemens-S7-Programmierumgebung 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.

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.

Individualisierung

Eigenes Logo als 3D-Doming auf die Front, als unlöschbares Bild fest in die Firmware oder auf die Rückseite samt eigener Artikelnummer? Die Einbauweise um 90°, 180° oder 270° drehen. Kein Problem für INSEVIS- Geräte.

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-CPUs in den Panel-SPS

CPU-T

Geräte mit **CPU-T** haben noch mehr Speicher, eine höhere Abarbeitungsgeschwindigkeit, bedienen größere, hochauflösendere Visualisierungen und sind ideal für mittlere Lösungen. Zwei getrennte Ethernet-Ports für verschiedene Netze sind bereits onboard. Profinet steht optional zur Verfügung. Panel-SPS und Panel-HMIs der *Generation II* sind ausschließlich mit dieser CPU ausgerüstet. (Ab Q3/2015 verfügbar.)

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) 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) 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)
Visualisierungsspeicher	48MB Flash (nicht flüchtig)
externer Speicher	Micro SD, bis max. 8 GB

Wichtigste Eigenschaften in Stichpunkten

S7-Programmierung

Bestehende Siemens-S7-Programmierumgebung 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.

2 Ethernet-Ports

Exclusiv bei der CPU-T:
Mit den beiden Ethernetschnittstellen als getrennte Ports mit eigenen IP-Adresskreisen die Steuerung als Verbindung zwischen Büronetzwerk und Maschinennetzwerk verwenden.

Know-how-Schutz

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

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.

Bedieneinheiten in den Panel-SPS

Allgemein

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 resisitive 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.

Generation I → *ab Q3/2016 auch im Design der Generation II*

Die S7-Panel-SPSen der Generation I verfügt über Frontfolien mit Prägung für den Einsatz kundenspezifischer Logos und über

folgende Displaydiagonalen:

- 3,5“ mit QVGA- Auflösung (320x240Pixel)
  → im 4:3-Format für den Fronttafeleinbau
  → in der Höhe der Normeinbaugehäuse 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

Generation II

Die S7-Panel-SPSen der Generation II verfügt über ungeprägte Frontfolien und über folgende Displaydiagonalen:

- 4,3“ mit Auflösung 480x272Pixel
  → im 16:9-Format für den Fronttafeleinbau
  → in der Höhe der Normeinbaugehäuse von 96mm

- 7“ mit WVGA- Auflösung (800x480Pixel)
  → im 16:9-Format für den Fronttafeleinbau und
  → einbaukompatibel zu 7“ der Generation I

- 10,1“ mit Auflösung 1024x600 Pixel
  → im 16:9-Format für den Fronttafeleinbau und
  → einbaukompatibel zu 10,1“ der Generation I

- 15,6“ mit Auflösung 1366x768 Pixel
  → im 16:9-Format für den Fronttafeleinbau

Peripherie für die Panel-SPS

Kommunikation SPS zu Peripherie

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

Bei den INSEVIS-S7-CPUs 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.

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 Relaisausgänge230V / 3A)
- **MIO84 /** (8 Digital- und 4 Analogein-/ ausgänge)
- **AI4O4 /** (4 Analogein- und 4 Analogausgänge)
- **AI8O2 /** (8 Analogein- und 2 Analogausgänge)
- **RTD8O2 /** (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
Betriebstemperatur-bereich Lagertemperaturbereich	-20°C ... +60°C (ohne Betauung) -30°C ... +80°C
Abmessungen B x H x T Gewicht	20 x 108 x 70 mm ca. 150 g
Leitungslänge - ungeschirmt (max.) - geschirmt (max.)	 30 m 100 m
Anschlusstechnik	Steckverbinder mit Zug-federkontakt für Quer-schnitte 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-AI4O4 kann mit ein wenig mehr Integrationszeit bis zu 16Bit auflösen. Natürlich ohne Aufpreis - So fair wie die gesamte INSEVIS- Philisophie

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 Panel-SPS

Konfiguration

Mit dem kostenlosen Konfigurationsstool „ConfigStage“ werden die Zusatzfunktionen der INSEVIS-CPUs parametriert und in die SPS geladen. Die onboard- bzw. dezentrale INSEVIS-Peripherie wird per Drag´n Drop auf die Steckplätze gezogen, parametriert 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.

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

- RS232 mit freiem ASCII
- RS485 mit freiem ASCII und ModbusRTU
- Ethernet-Verbindung (aktive S7-Verbindung-RFC1006, TCP, UDP, Modbus-TCP, INSEVIS-Panel-HMI)
- CAN (CANopen® über voreingestellte Parametersätze oder Import von EDS-Dateien)

Diverse S7-Bausteine für Motion-Control-Funktionen intelligenter CAN-Antriebe und Beispielkonfigurationen für Fremdperipherien stehen dokumentiert im Internet zum freien Download zur Verfügung.

Visualisierung

Mit dem firmenlizenzbasierten PC-Visualisierungstool „VisuStage“ können moderne multilinguale Visualisierungen auch von Anfängern einfach erstellt und simuliert werden. Bestehende S7-Variablen werden samt Symbolik aus den DBs des S7-Programms übernommen. Export- und Importfunktionen für Texte erleichtern die Übersetzung in Fremdsprachen enorm. Die Projektierung ist äußerst übersichtlich und effektiv, einfach zu verstehen und zu bedienen. Sie verfügt u.a. über folgende Funktionen:

- Programmsprache wahlweise deutsch oder englisch,
- unbeschränkte Anzahl verwendeter Sprachen (Full Version) bzw. 1 Sprache (kostenlose Lean Version),
- Stör-/Ereignismeldesystem und -archiv mit bis zu 1024 Alarmen,
- Trenddatenerfassung und -archiv für 4 Trends mit je 16 Kanälen,
- Rezepturverwaltung mit bis zu 64 Rezepturen mit 256 Elementen in 256 Datensätzen,
- Benutzerverwaltung mit 9 Ebenen Passwortschutz,
- Bildschirmschoner, Hinterleuchtungsabschaltung und Buzzer
- Bibliotheken mit 2D- und 3D-Symbolen, integriertes Simulationstool.

Trotz des großen Leistungsumfangs ist diese Software in weniger als 3 Minuten installiert und gestartet und benötigt nur minimale Hardwareanforderungen.

Remotezugang

Mit der kostenlosen 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 Portable-Version zur Verfügung, muss also nicht installiert werden. Die Software kommuniziert via TCP/IP mit der Panel-SPS und kann auch die Visualisierung von dort laden. Es wird lediglich die Binärdatei verwendet, die nicht rücklesbar ist und Ihre Quelldaten schützt.

Ebenfalls 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. Es können mehrere Instanzen dieses Programms nebeneinander arbeiten, so dass mehrere Remotegeräte gleichzeitig visualisiert werden können (Leitwartenfunktion).

Servicetool

Wer über eine Serviceabteilung zur Wartung der Steuerungen verfügt, kennt die Probleme zu einem mit den Lizenzkosten und zum andern, dass die immer komplexer werdende Software bei nicht täglicher Benutzung durch die Mitarbeiter nur schwer zu beherrschen ist.
Nicht so die kostenlose „ServiceStage“ von INSEVIS: In einer Minute installiert, ist sie intuitiv bedienbar und stellt den Servicemitarbeiten diese wichtigen Funktionen zur Verfügung:

- Auslesen gerätespezifische Daten (SNr., Firmwareversion, IP-Adresse (änderbar), MAC-Adresse, etc.),
- 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

Zubehör für die Panel-SPS

Zubehör S7-Panel-SPS

Für den Anschluss von INSEVIS-Geräten stehen pinmarkierte Stecker mit seitlichen Lösehebeln bzw. 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.

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

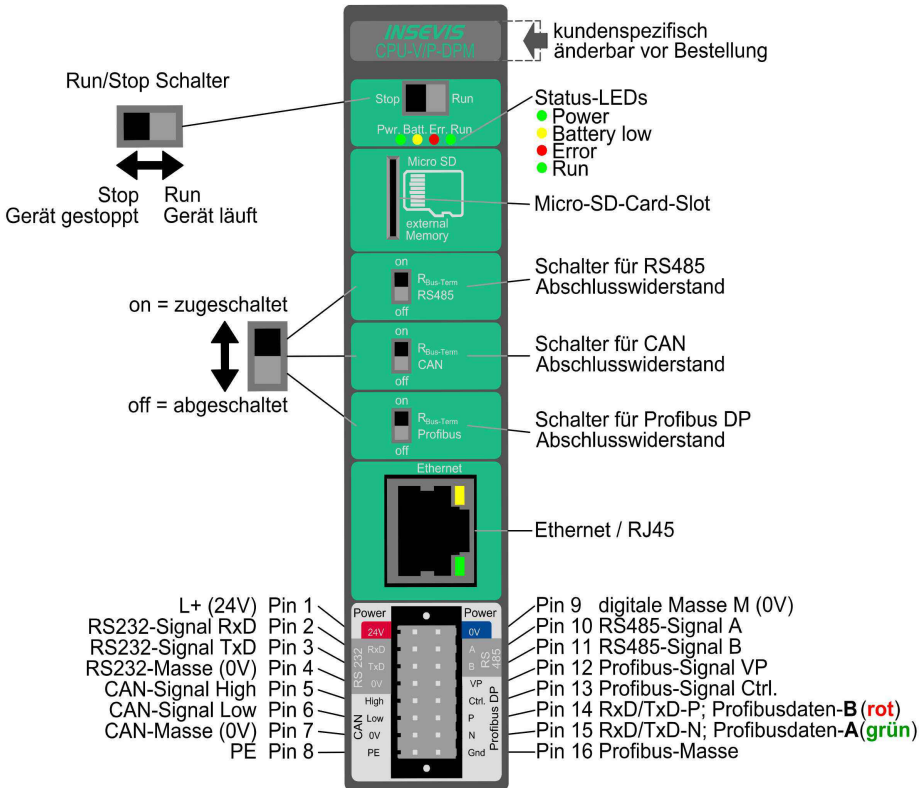
Abbildung Zubehör	Zubehör
Stecker E-CON16 E-CONS16 (pinmarkierte Stecker für max. 1,5mm² Anschluss)	**Für CPU V:** Steckverbinder 2x8polig, Lösehebel, Steckverbinder 2x8polig, Schraubflansch Profibus-Adapter für 12MBaud-Netze
 E-CON10 E-CONS10 (pinmarkierte Stecker für max. 1,5mm² Anschluss)	**Für CPU T:** Steckverbinder 2x5polig, Lösehebel, Steckverbinder 2x5polig, Schraubflansch
 (pinmarkierter Stecker für max. 1,5mm² Anschluss)	**Nur für PC351V/P (Peripherie):** Steckverbinder 2x4polig
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	3D-Doming Label mit Kundenlogo frontseitig OEM-Firmware mit eingebautem Logo Einschubstreifen H mit Kundenlogo rückseitig Einschubstreifen V mit Kundenlogo rückseitig
Software **Hinweis:** [3] Firmenlizenz, keine Stückkosten Software im Internet downloadbar	VisuStage Lean-Version ConfigStage RemoteStage ServiceStage VisuStage Voll-Version Firmenlizenz VisuStage Voll-Version Wartungslizenz
Ersatzteile **Hinweis:** Ein Befestigungssatz ist in jedem Lieferumfang enthalten. [4] Verpackungseinheit von 10 Stück	Zusätzlicher Befestigungssatz mit Erdungskralle für 3,5“ und 4,3“-Geräte Zusätzlicher Befestigungssatz mit Erdungskralle für 5,7“ und 7“-Geräte Zusätzlicher Befestigungssatz mit Erdungskralle für 10,1“ und 10,2“-Geräte Zusätzlicher Befestigungssatz mit Erdungskralle für 15,6“-Geräte

Anbauteile und Sonderkonstruktionen auf Anfrage.

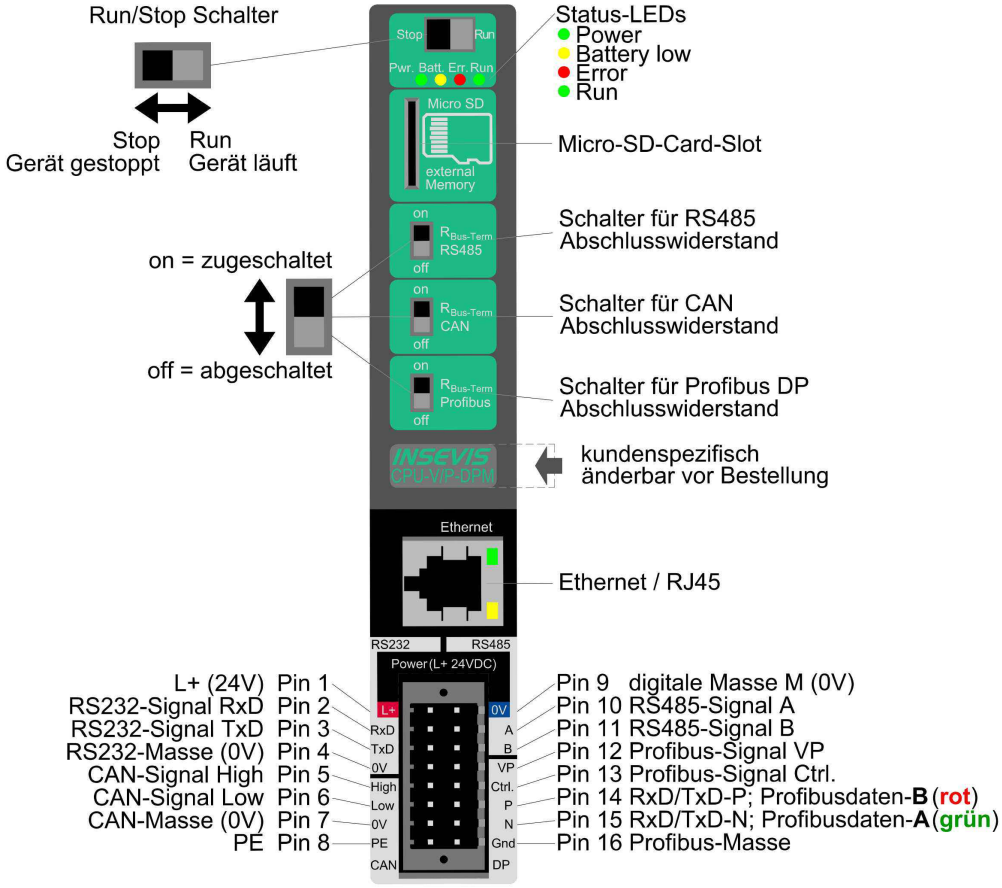
Beschaltung der CPU-V/P

Beschaltung der CPU-V/P in Geräten mit und ohne Peripherieslots

Beschaltung bei Grundgeräten (ohne Peripherieslots)



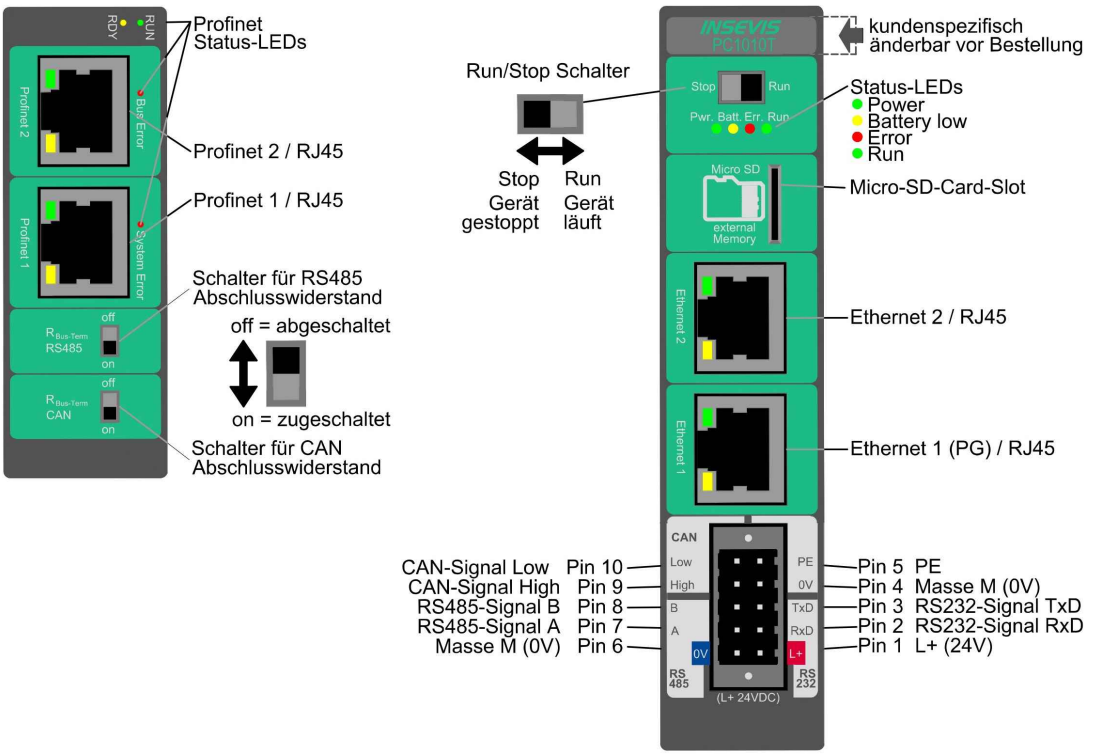
Beschaltung bei Geräten mit Peripherieslots



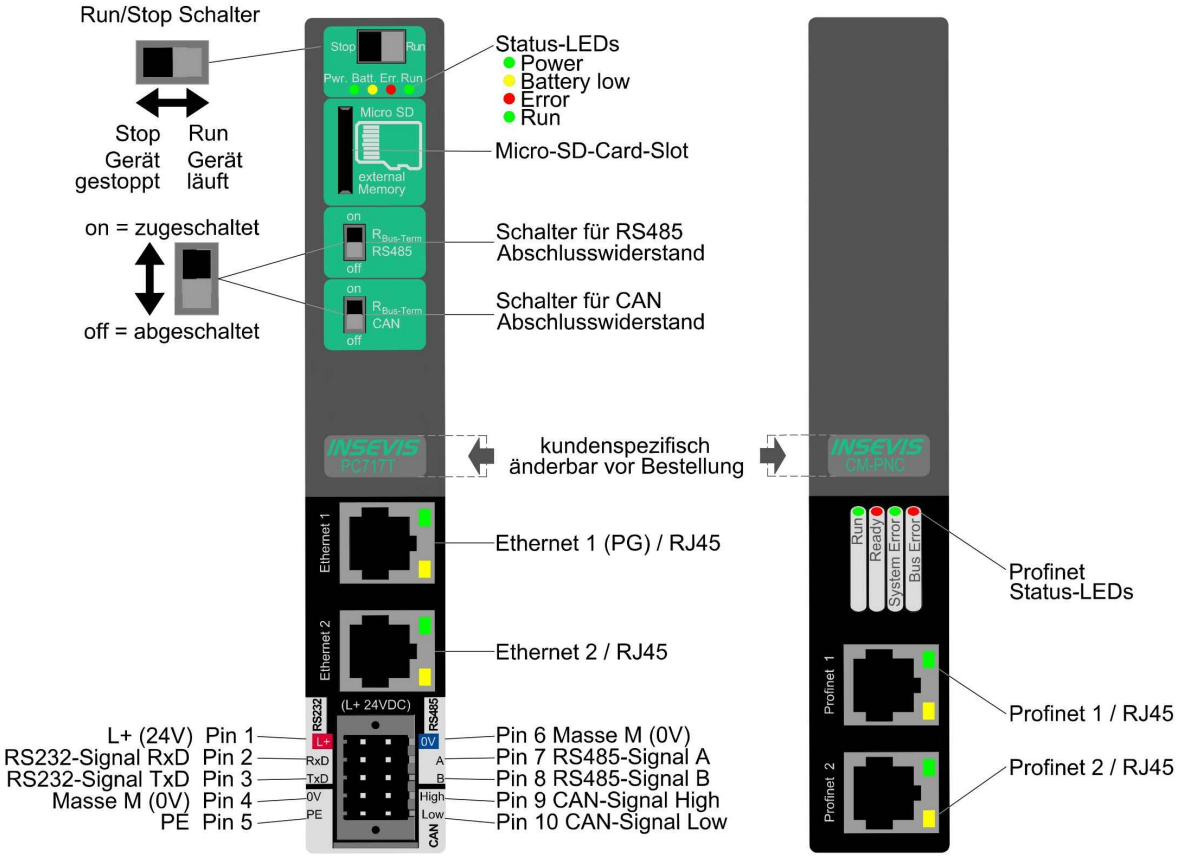
Beschaltung der CPU-T

Beschaltung der CPU-T in Geräten mit und ohne Peripherieslots

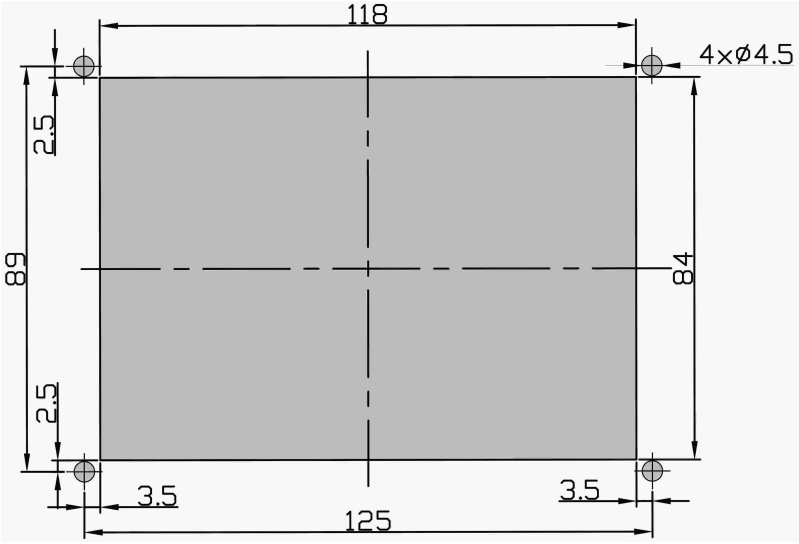
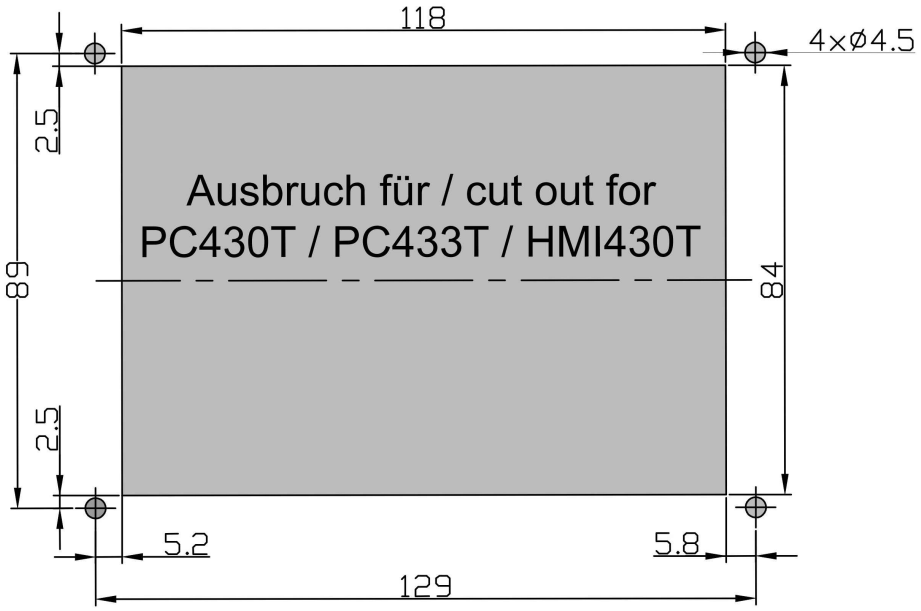
Beschaltung bei Grundgeräten (ohne Peripherieslots)



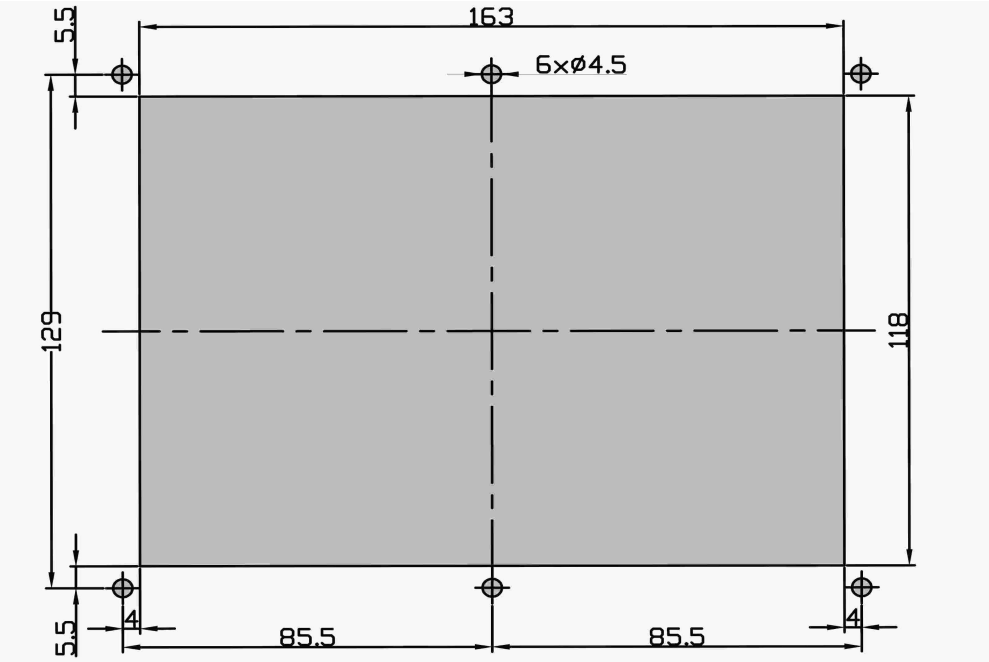
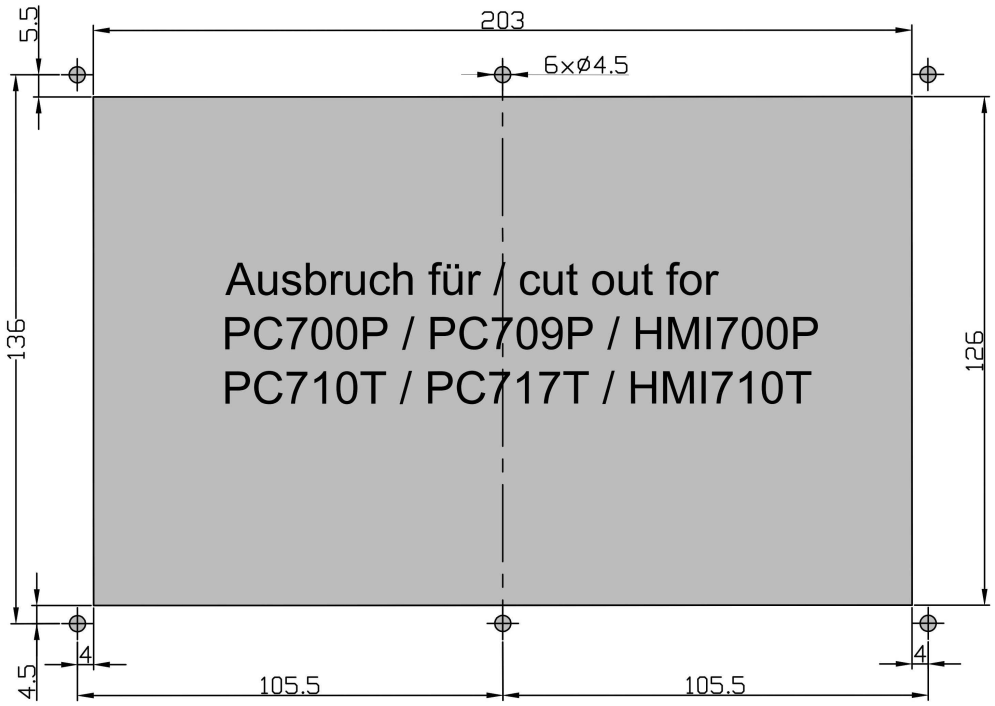
Beschaltung bei Geräten mit Peripherieslots



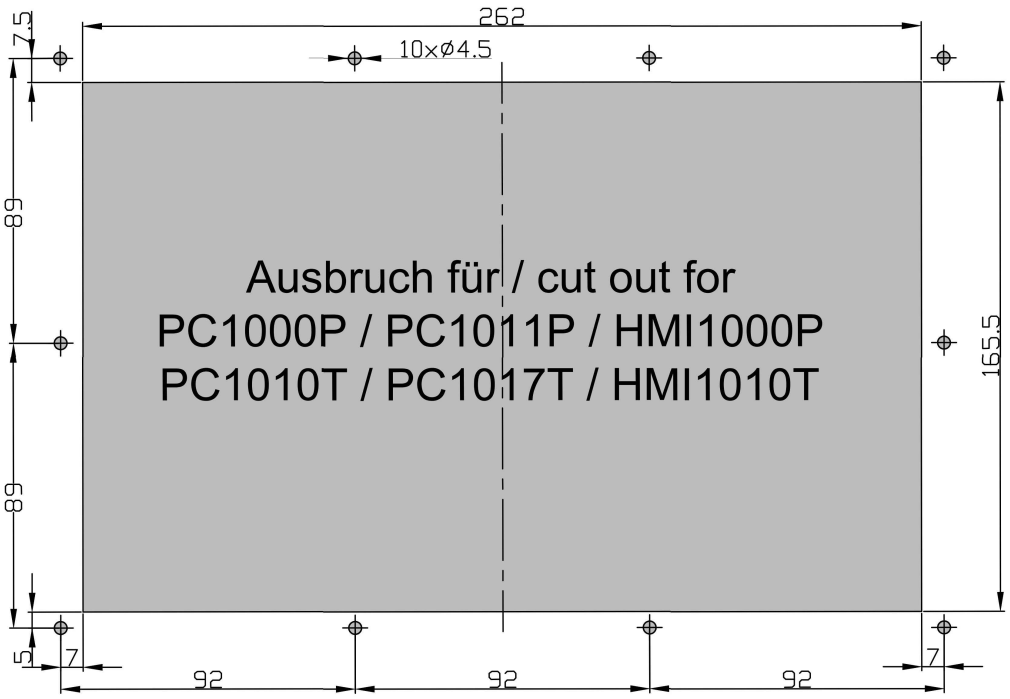
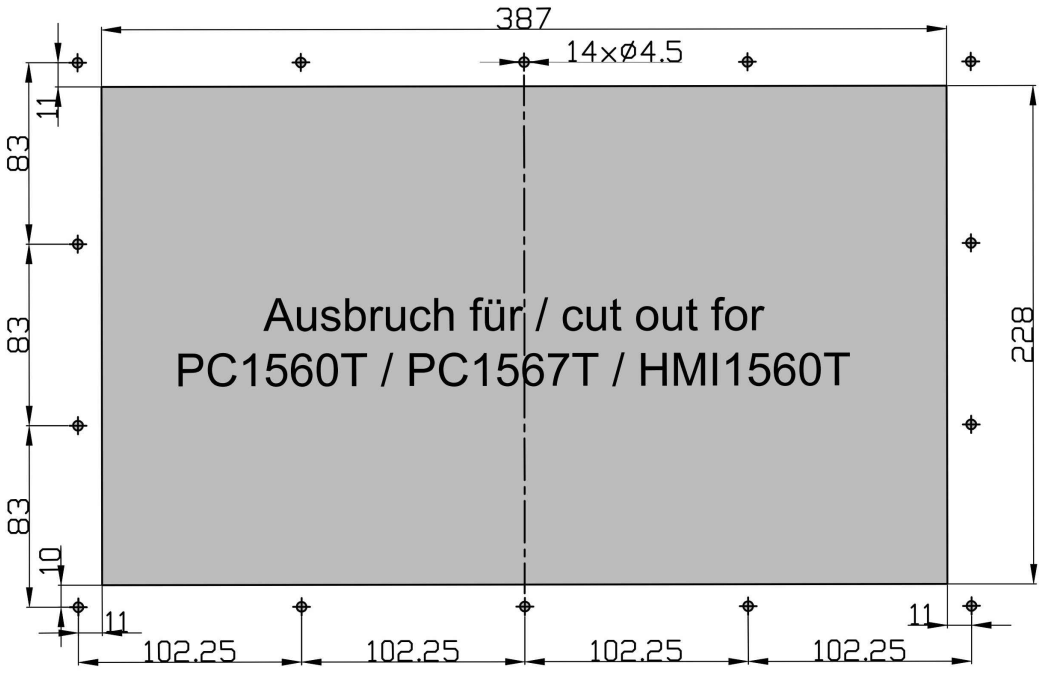
S7-Panel-SPS Schaltschrankausschnitte

Gerät	Schaltschrankausschnitt
Abmessungen Ausbruch B x H x T (mm) 118 x 84 4 Löcher mit D 4,5mm **Kabelabgang** (Sicht von hinten, bei horizontalem Einbau) **für PC350V/P, HMI350V/P** - RJ45 nach oben - Stecker 2x8 nach rechts Einbautiefe ca. 49mm max. **für PC353V/P** - alles 45° nach rechts Einbautiefe ca. 89mm max.	 Abb.: Ausbruch für alle 3,5”-Geräte
Abmessungen Ausbruch B x H x T (mm) 118 x 84 4 Löcher mit D 4,5mm **Kabelabgang** (Sicht von hinten, bei horizontalem Einbau) **PC430T, HMI430T** Kabelabgang - 2xRJ 45, und Stecker 2x5 nach rechts - 2xRJ45 (PN) nach oben Einbautiefe ca. 44mm max. **PC433T** Kabelabgang - 45° nach links Einbautiefe ca. 89mm max.	 Abb.: Ausbruch für alle 4,3”-Geräte

S7-Panel-SPS Schaltschrankausschnitte

Gerät	Schaltschrankausschnitt
Abmessungen Ausbruch B x H x T (mm) 163 x 118 6 Löcher mit D 4,5mm **Kabelabgang** (Sicht von hinten, bei horizontalem Einbau) **für PC570V/P, HMI570V/P** - RJ45 nach rechts Stecker 2x8 nach unten Einbautiefe ca. 49mm max. **für PC577V/P** - 45° nach unten Einbautiefe ca. 95mm max.	 Abb.: Ausbruch für alle 5,7”-Geräte
Abmessungen Ausbruch B x H x T (mm) 203 x 126 6 Löcher mit D 4,5mm **Kabelabgang** (Sicht von hinten, bei horizontalem Einbau) **PC700P, HMI700P,** Kabelabgang - RJ 45 nach rechts - Stecker 2x8 nach unten Einbautiefe ca. 49mm max. **PC710T, HMI710T** Kabelabgang - 2xRJ 45, und Stecker 2x5 nach rechts - 2xRJ45 (PN) nach oben Einbautiefe ca. 45mm max. **PC709P / PC717T:** Kabelabgang - 45° nach unten Einbautiefe ca. 95mm max.	 Abb.: Ausbruch für alle 7”-Geräte

S7-Panel-SPS Schaltschrankausschnitte

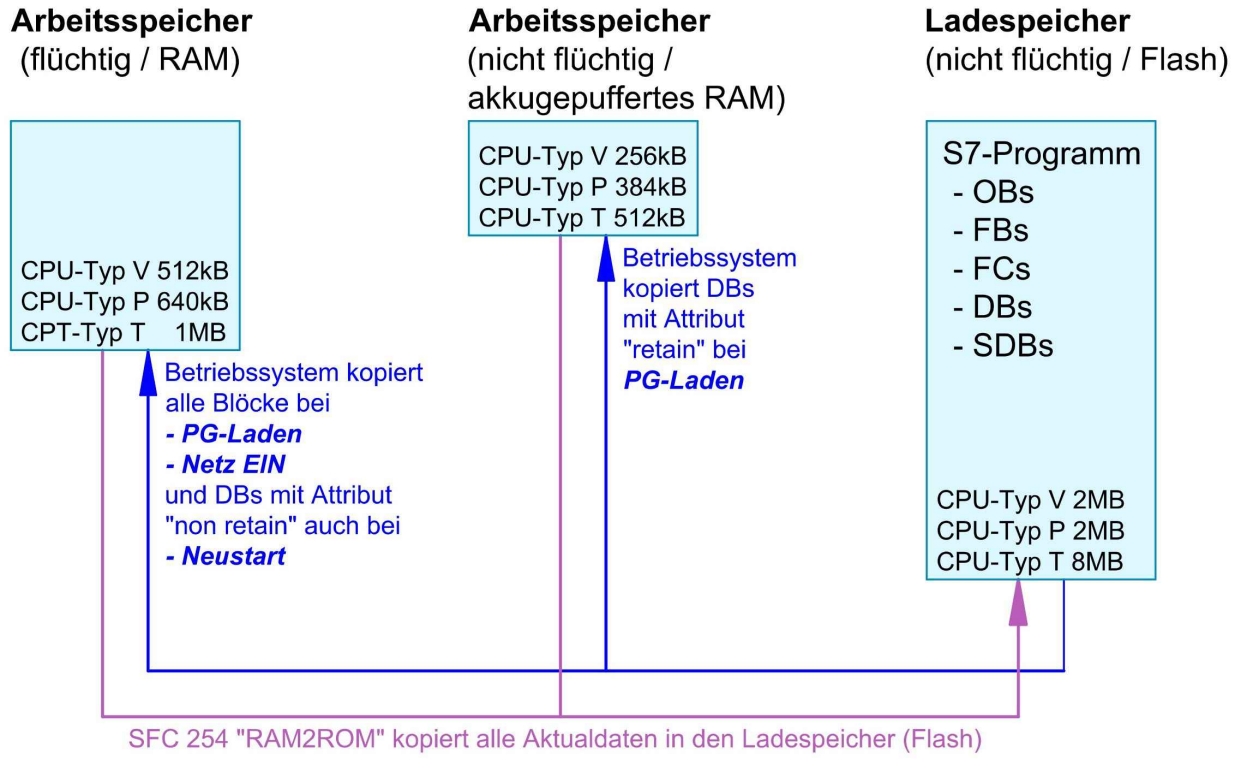
Gerät	Schaltschrankausschnitt
Abmessungen Ausbruch B x H x T (mm) 262 x 165,5 x 50 10 Löcher mit D 4,5mm **Kabelabgang** (Sicht von hinten, bei horizontalem Einbau) **für PC1000P, HMI1000P:** - RJ45 nach rechts - Stecker 2x8 nach unten Einbautiefe ca. 49mm max. **PC1010T, HMI1010T** Kabelabgang - 2xRJ 45, und Stecker 2x5 nach rechts - 2xRJ45 (PN) nach oben Einbautiefe ca. 47mm max. **für PC1011P / PC1017T:** Kabelabgang - 45° nach unten Einbautiefe PC1011P ca. 95mm max. Einbautiefe PC1017T ca. 98mm max.	 Abb.: Ausbruch für alle 10,1” und 10,2”-Geräte
Abmessungen Ausbruch B x H x T (mm) 203 x 126 6 Löcher mit D 4,5mm **Kabelabgang** (Sicht von hinten, bei horizontalem Einbau) **PC1560T, HMI1560T** Kabelabgang - 2xRJ 45, und Stecker 2x5 nach rechts - 2xRJ45 (PN) nach oben Einbautiefe ca. 55mm max. **PC1567T** Kabelabgang - 45° nach unten Einbautiefe ca. 104mm max.	 Abb.: Ausbruch für alle 15,6”-Geräte

Speicherorganisation

Die Speicheraufteilung entspricht im Wesentlichen denen der Siemens S7-CPUs. 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 wechselt in den Run-Modus.



Ethernet

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

Die INSEVIS- Panel-SPSen werden mit **IP-Adresse 192.168.80.50** ausgeliefert.

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

IP-Adresse im SIMATIC-Manager einstellen

Benötigte Ausrüstung

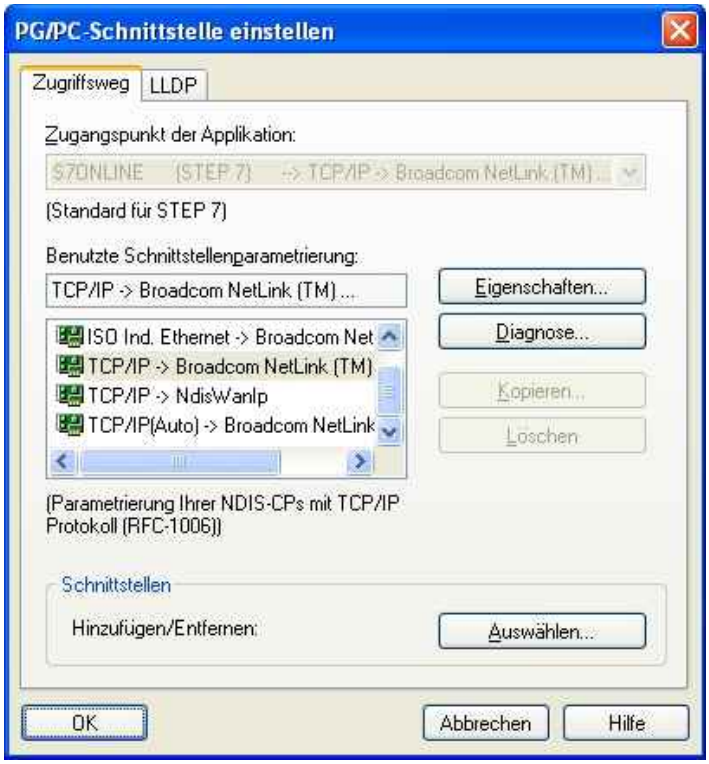
PC mit SIMATIC-Manager ab Version 5.4 SP5 (PG)
INSEVIS-SPS (AG)
Netzteil 24V
Patchkabel oder Crossover-patchkabel

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

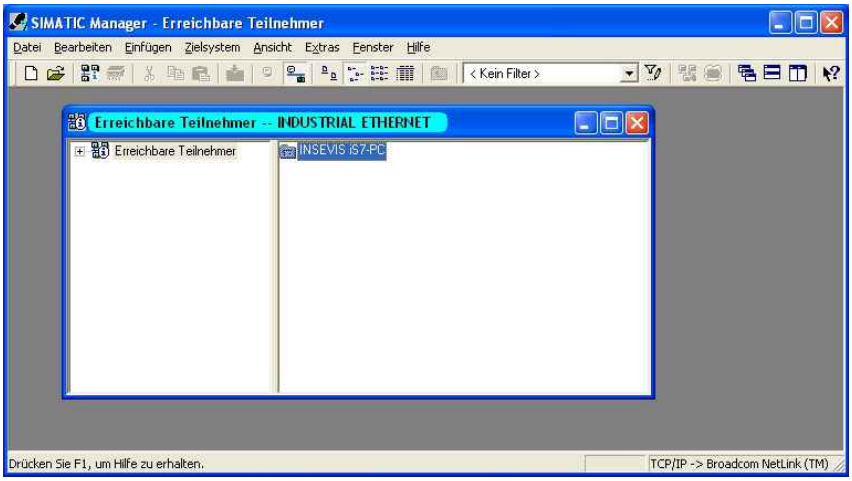


Steuerung anwählen

→ Zielsystem
→ Erreichbare Teilnehmer anzeigen
(Meldung als INSEVIS S7-PLC)
→ **INSEVIS S7-PLC** auswählen / markieren
→ Zielsystem
→ Ethernet Teilnehmer bearbeiten

Hinweis: → VIDEO verfügbar

Für dieses Beispiel ist auf den INSEVIS-Download-Seiten ein Video verfügbar



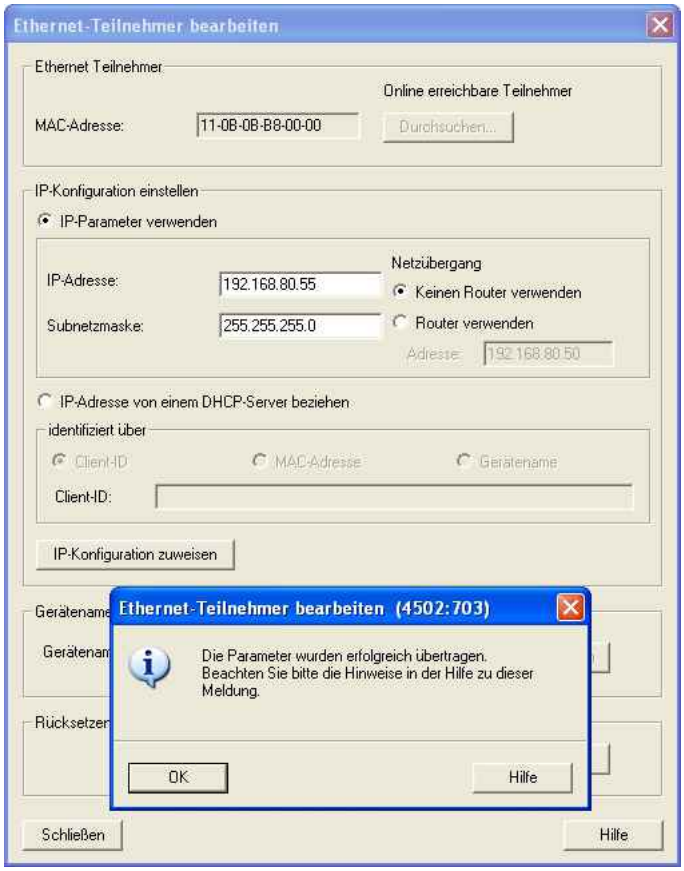
Kommunikation - Ethernet

IP-Adresse ändern
→ weiter in Box
„IP-Konfiguration einstellen“
„IP-Parameter anwählen“
→ IP-Adresse eintragen
→ Subnetzmaske 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!



Hinweis: → **VIDEO verfügbar:** Für dieses Beispiel ist auf den INSEVIS-Service-Seiten ein Video verfügbar

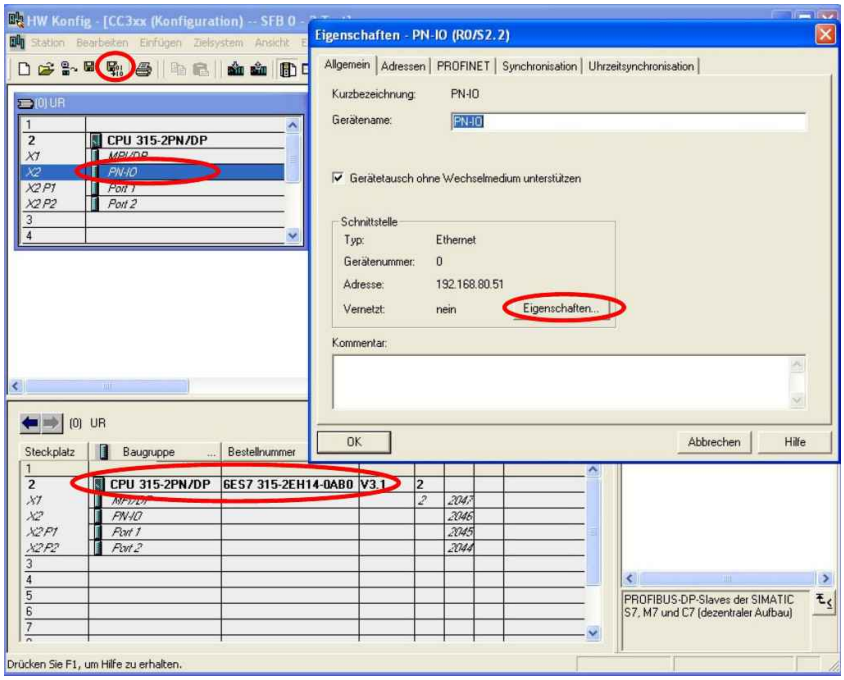
Anlegen der CPU und der Ethernet-Verbindung im SIMATIC-Manager

Anlegen der INSEVIS-CPU im SIMATIC- Manager von Siemens:

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 nötig.

Auswahl der Ethernet-Kommunikation im SIMATIC- Manager von Siemens:
- Doppelklick auf *PN-IO*
- Taste „Eigenschaften“ drücken
- IP-Adresse einstellen
- mit „OK“ bestätigen
  – Kompillieren & Speichern



Mögliche Probleme: Ihre Firewall verhindert die Verbindungsaufnahme → Bitte Firewall(s) rekonfigurieren
IP-Adresse des PC und der SPS müssen in einem gemeinsamen Netz ein (im Beispiel 192.168.**80**.xxx) → Mit dem Kommandozeilenbefehl "ipconfig" können Sie die aktuelle IP-Konfiguration Ihres PC's abfragen.

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

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

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.

S7 Aktive Verbindungen zu Partner-SPSen (RFC1006)

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

Maximal 16 Vebindungen 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.

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 (Verschachtelungsaufruf).
0	1	8001	Parameter-ID ist nicht korrekt.
0	1	8002	Verbindung mit ID ist nicht konfiguriert oder inkorrekter 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.
```

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 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 Put und Get in einer Verbindung

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

Wenn man z.B. hintereinander bei einer Verbindung den SFB 121 „Put“ und SFB 120 „Get“ aufruft, wird SFB 120 „Get“ 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 = Autrag 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.

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. Autrag 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 (Verschachtelungsaufruf).
0	1	8001	Parameter-ID ist nicht korrekt.
0	1	8002	Verbindung mit ID ist nicht konfiguriert oder inkorrekter 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.
```

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 "TRECV" 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 akzepitert 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)

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 inkorrekter 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.
```

Daten empfangen über TCP mit dem SFB 123 "TRECV"

Mit dem SFB 123 “TRECV" 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 soviele 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 inkorrekter 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).

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.
```

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 inkorrekter 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 inkorrekter 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 zugewiesenen wurde
      O     M    600.3                 //  oder Fehler aufgetreten sind,
      R     M    600.0                 //  setze Anforderungssignal zurück.
```

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 inkorrekter 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.
```

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 inkorrekter 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_IDDR ist nicht korrekt
0	0	1	8007	Auftrag fehlgeschlagen wegen Verbindungsproblem (Kabel abgezogen)

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.

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 soviele 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 inkorrekter 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

S7-Programmbeispiel für die Verwendung des SFB128

Ziel: Empfangen von Daten über UDP von irgendeinen 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

Hinweis: 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.

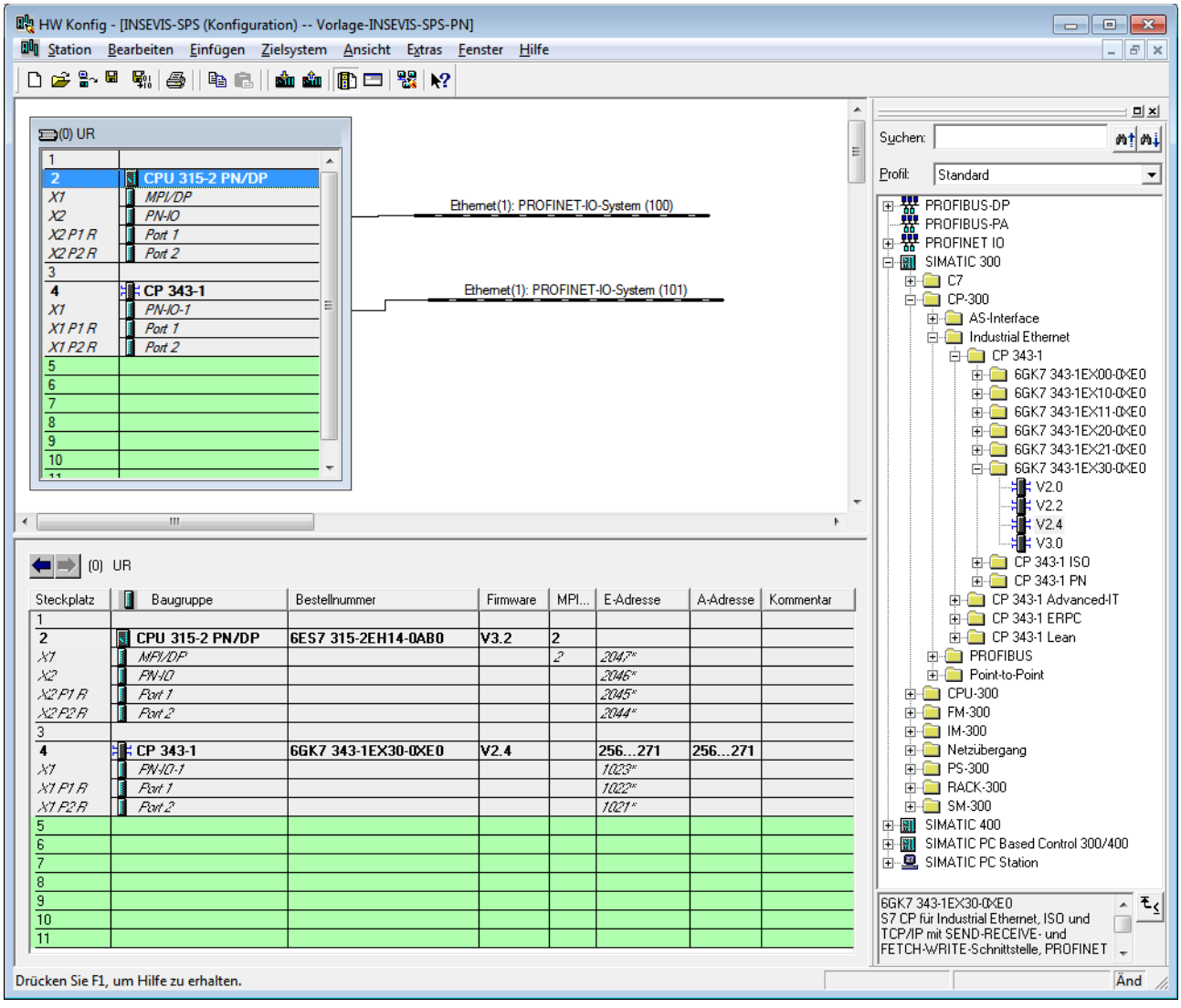
Beschaltung

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

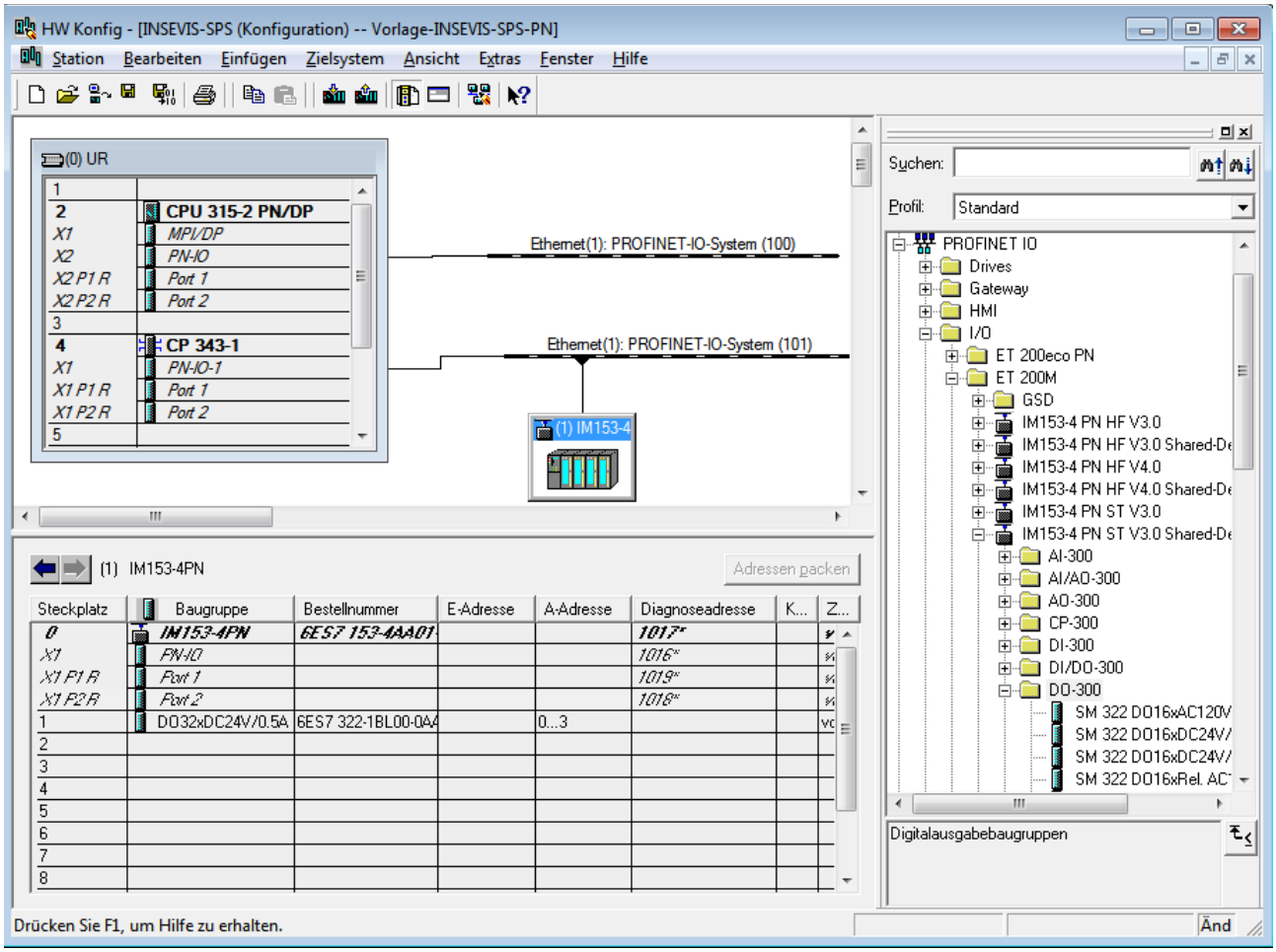
Profinet ist bei INSEVIS als Option zusätzlich zur CPU verfügbar. Um diese Option auch in der Konfiguration im SimaticManager / TIA-Portal darzustellen, ist es nötig, auch hier eine zusätzliche Baugruppe dafür zu verwenden. Diese Baugruppe ist der CP 343-1 (Siemens-Bestellnummer 6GK7 343-1EX30-0XE0 mit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



Kommunikation - Profinet IO Controller

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



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 muss keine Funktion zum zu Prozessaustausch aufgerufen werden.
- Die Alarm-OBs werden von der CPU selbstständig aufgerufen

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.

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 Erstaufruf 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

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

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 Längeninformationen 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 duch 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.

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.

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-Identifier UID ungültig
BAUD	INOUT	DINT	8	Baudrate in bit/s, z.B. "9600" Wert muss > 0 sein
DATA_FMRT	INOUT	BYTE	12	0 = 8 Datenbit, keine Parität 1 = 8 Datenbit, gerade Parität 2 = 8 Datenbit, ungerade Parität
UID	INOUT	BYTE	13	0 = Konfiguration ASCII oder Modbus-RTU Client 1..247 = Modbus-RTU Server Unit-Identifier (nur RS485)

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

Kommunikation - Profibus DP-Master

Profibus DP

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

Hinweis: Die Konfiguration im 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 Siemens-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.

Hinweis: → VIDEO verfügbar
Für dieses Beispiel ist auf den INSEVIS-Service-Seiten ein Video verfügbar

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

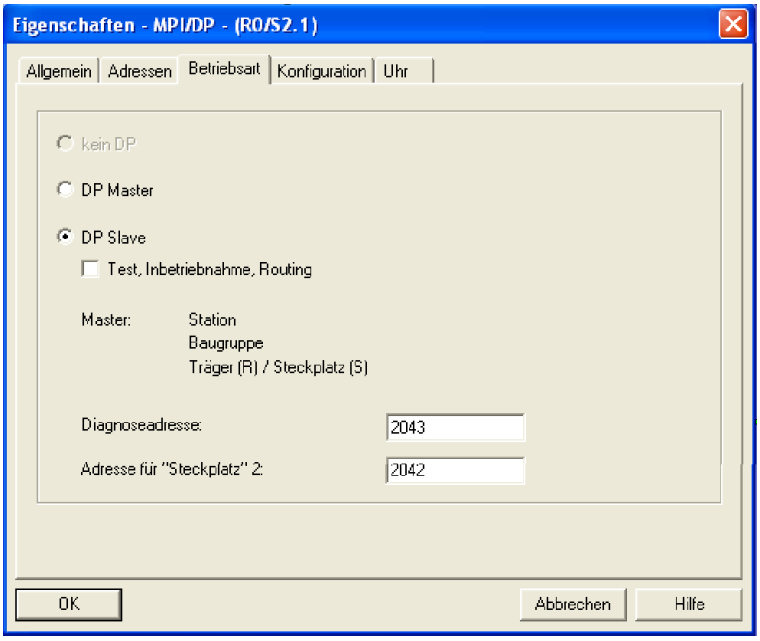
Hinweis: → VIDEO verfügbar
Für dieses Beispiel ist auf den INSEVIS-Service-Seiten ein Video verfügbar

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 <u>nicht</u> “Test, Inbetriebnahme, Routing” an!**
- **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 bestück 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)

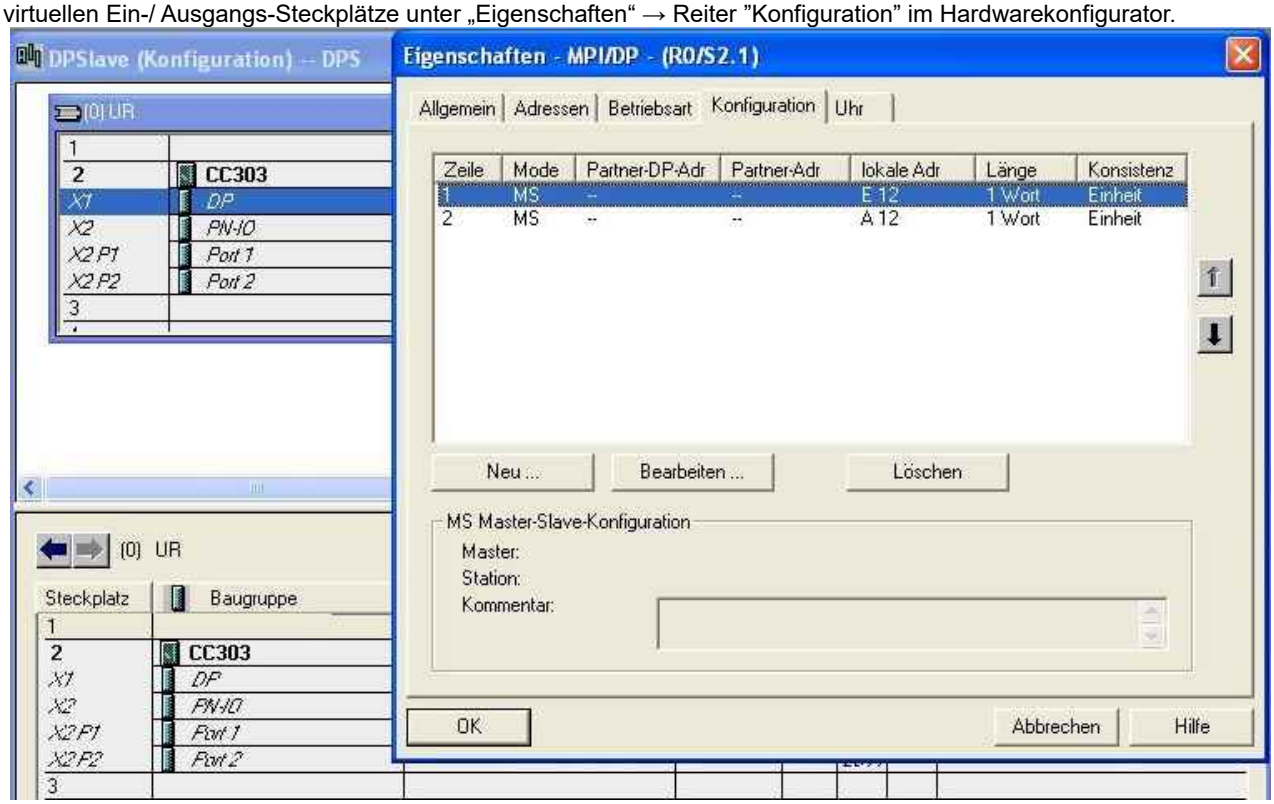
Kommunikation - Profibus DP-Slave

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.



Hinweis:

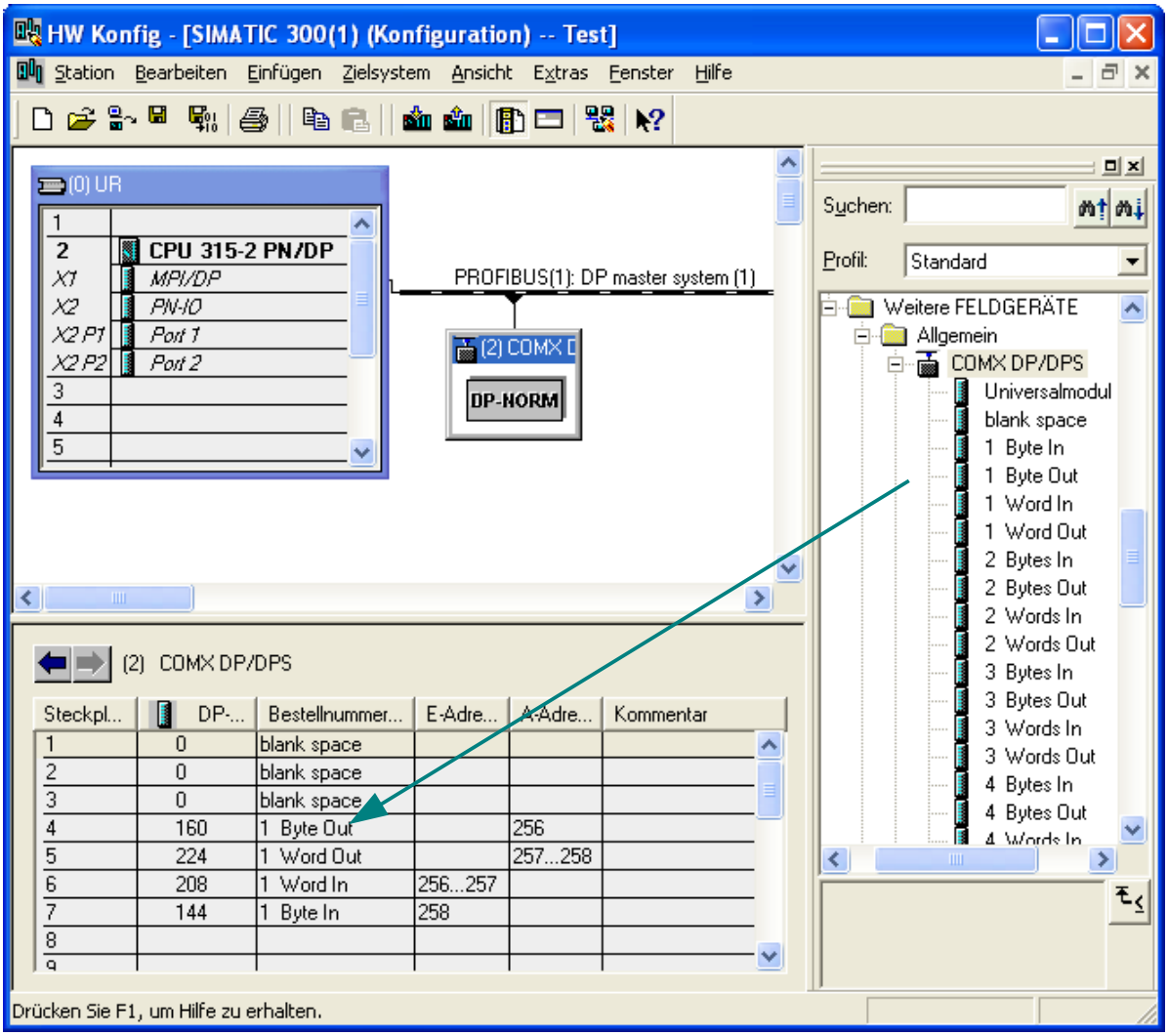
- Maximal 21 virtuelle Steckplätze (Reihen) können konfiguriert werden.
- Der Modus “Direkter Datenaustausch” (DX) wird nicht unterstützt, nur Master- Slave- Konfiguration (MS).
- Bei Ausfall des DP-Masters (oder Abziehen des Kabels) und direktem Peripheriezugriff (z.B. L PEW 2) auf die konfigurierten 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

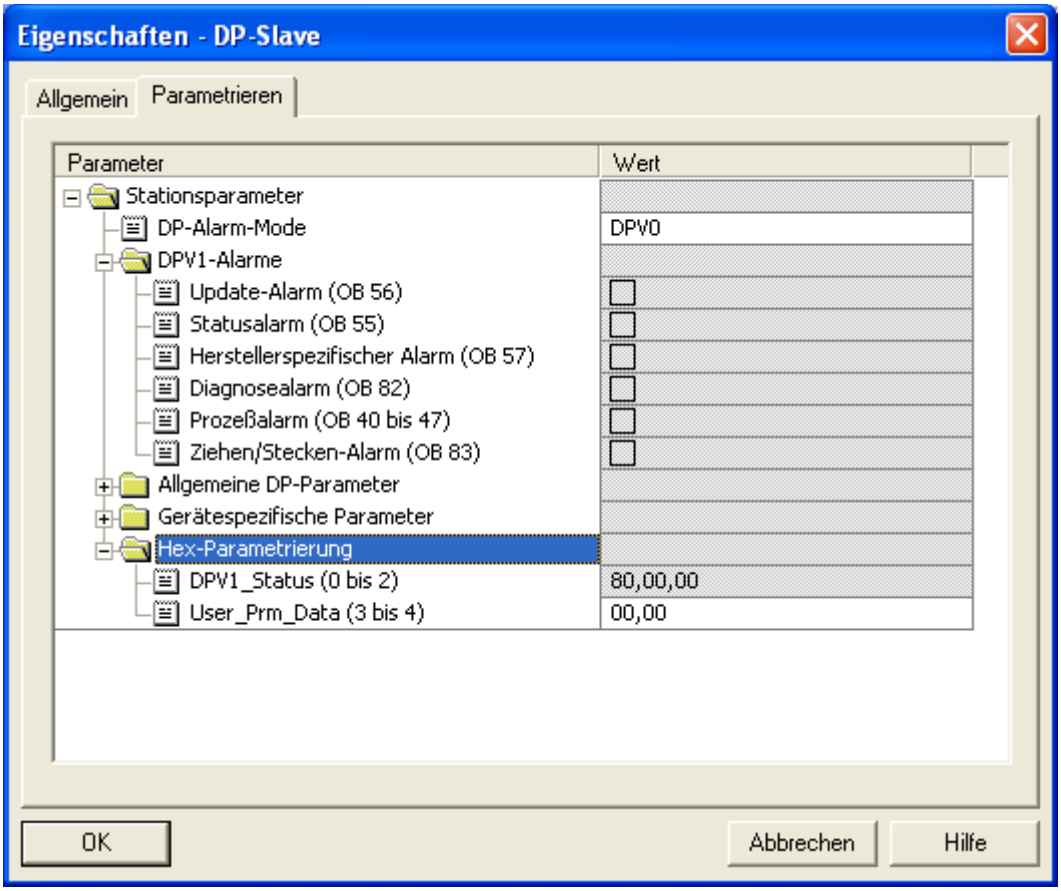
Bitte beachten Sie, dass bei der Konfiguration des DP- Masters die GSD-(Gerätestammdaten)-Datei “HIL_0C0F.GSD” verwendet wird, um die INSEVIS-SPS als DP-Slave zu konfigurieren.
Benutzen Sie nicht die “CONFIGURED STATIONS → CPU 31X” im Hardware Konfigurator.

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



- **Hinweise:**
- Die ersten 3 Steckplätze müssen leer bleiben (Platzhaltermodule)
- Ab dem 4. Steckplatz können die virtuellen Ein-/ Ausgänge konfiguriert werden **(Drag ´n Drop)**
  (Alternativ kann das Universalmodul auf einen freien Slot gezogen und manuell komplett parametriert 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 ... Byte x - 1	Moduldiagnose (identifikationsbezogene Diagnose) (Länge hängt ab von Anzahl konfigurierter Steckplätze)
Byte x … Byte y	Modulstatus (gerätebezogene Diagnose) (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
$0C_H$	$0F_H$

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

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 als DP-Slave konfiguriert 80C3H SFC aufgerufen in niedrigerer Prioritätsebene
BUSY	OUTPUT	BOOL	TRUE = Auftrag in Abarbeitung

CAN

Alle INSEVIS-Steuerungen 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**

CANopen®-Konfiguration mit der ConfigStage

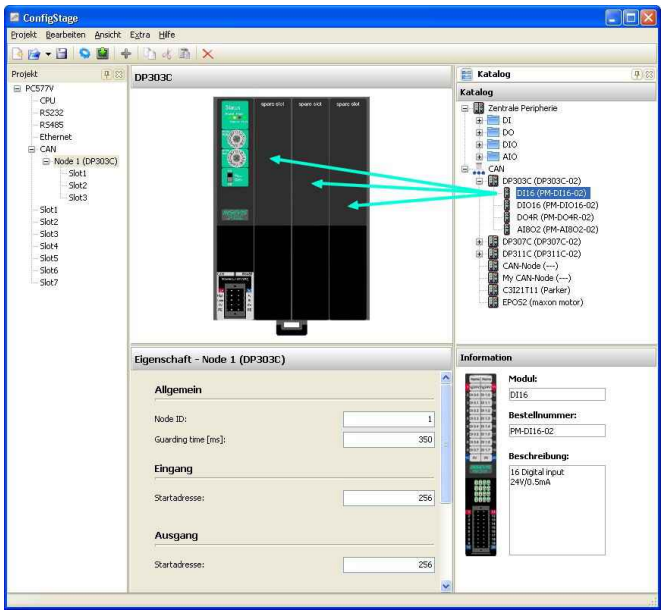
Die Einstellungen in der Konfigurationssoftware ConfigStage sind einfach durchzuführen und
→ im Kapitel „ConfigStage“ unter dem Punkt „CAN-Einstellungen“ detailliert beschrieben.

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.)

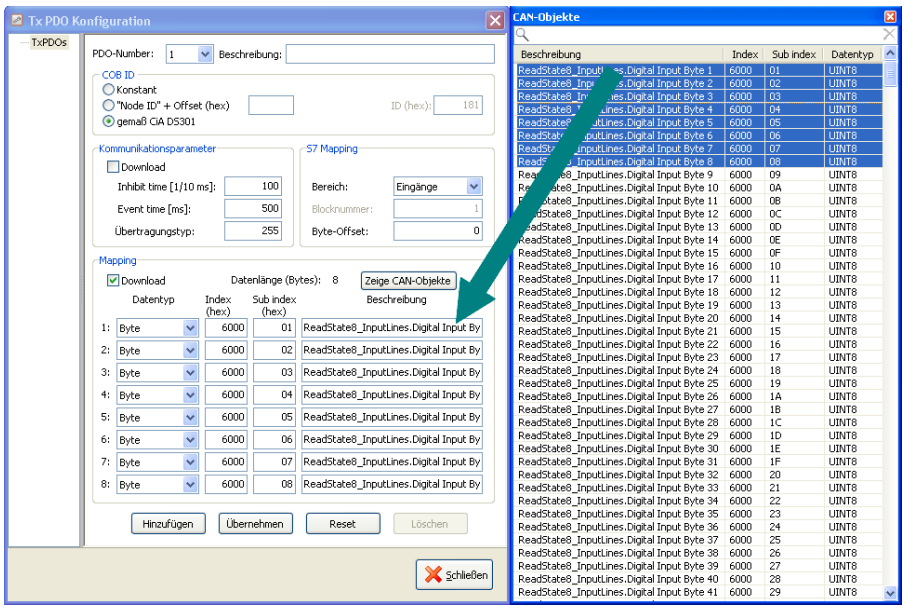
- Die Startadresse der dezentralen E/A-Adressierung kann nicht einzeln pro Peripheriemodul, sondern nur einmalig pro Kopfstation eingestellt werden. Danach erfolgt der Zugriff unter S7 genau so, wie auf die INSEVIS-onboard-Peripherie.



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

- **Einlesen der slavebezogenen EDS-Datei** und

- Konfiguration der CAN-Objekte mit der ConfigStage an.



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

FB110 SDO download Variable Wort	FB111 SDO upload Variable Doppelwort	FB112 SDO download Variable Doppelwort
Request Node:Knotennummer Object Subindex DatenWort Rückgabewerte: Done, Error, ErrorCode	Request Node:Knotennummer Object Subindex DatenDoppelwort Rückgabewerte: Done, Error, ErrorCode	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 einzeln 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 fortgestzt
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

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.

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.

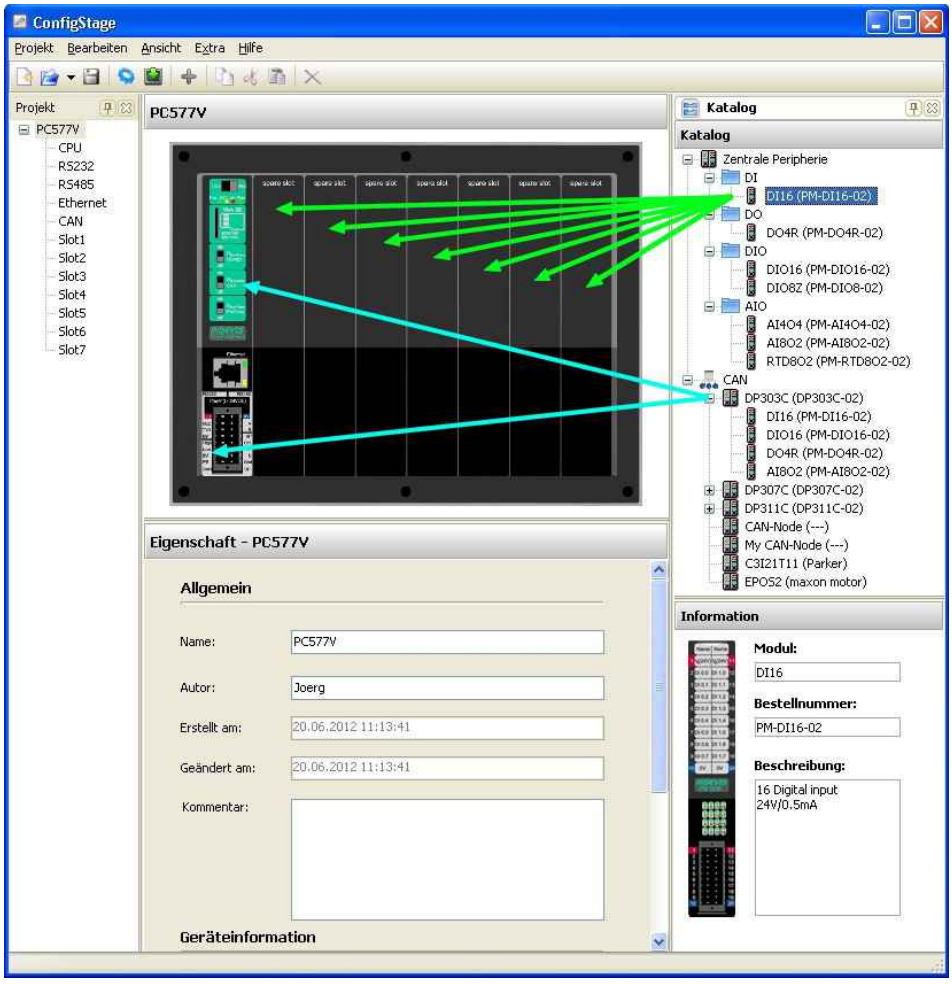
```
// call SFB114 (via local instance)
      CALL  #CAN_State
       TXOFL    :=
       RXOFL    :=
       L2BUFOFL :=
       NMT_ERROR:=
       OP_ERROR :=
       HW_ERROR :=
       SDO_ERROR:=
       NODESTATE:=
...
// test all critical nodes for bootup, state  == PREOPERATIONAL
      U(
      L     #CAN_State.NODESTATE[2]     // Node 2
      L     B#16#7F                     // 7F = PREOPERATIONAL
      ==I
      )
// 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

// 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“ dient zur Konfiguration bzw. Parametrierung von INSEVIS-spezifischen Eigenschaften wie Kommunikations- und Peripherie-Einstellungen und zum Download in den Hardwarebereich der INSEVIS-SPS. Zuerst suchen Sie Ihr Gerät aus der angebotenen Auswahlliste aus. Nach der Geräteauswahl öffnet sich ein Fenster mit mehreren Bereichen.

Grundlegende Einstellungen

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

Hinweis: 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.

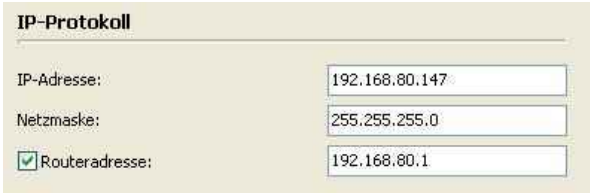
Konfiguration mit der Software „ConfigStage“

Änderung der Geräte-IP-Adresse

Ab ConfigStage 1.0.14.5 kann mit dieser Software auch die IP-Adresse des Ziel-Gerätes **GEÄNDERT** werden.

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

(Dazu ist das Betriebssystem 2.0.35 oder höher in den INSEVIS-SPSen nötig)



Die IP-Adresse des Zielgerätes wird bei in der Download-Dialogbox (F12) angegeben (rechts).

Im Beispiel wurde in der Konfiguration dem Gerät eine neue IP-Adresse zugeteilt (192.168.80.147, siehe oben)

Beim Download muss jedoch die Änderung der IP-Adresse noch an die bisherige IP-Adresse 192.168.80.148 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.
Nach dem Download hat das Gerät die neue Konfiguration erhalten und hat die IP-Adresse 192.168.80.147 übernommen.

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

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

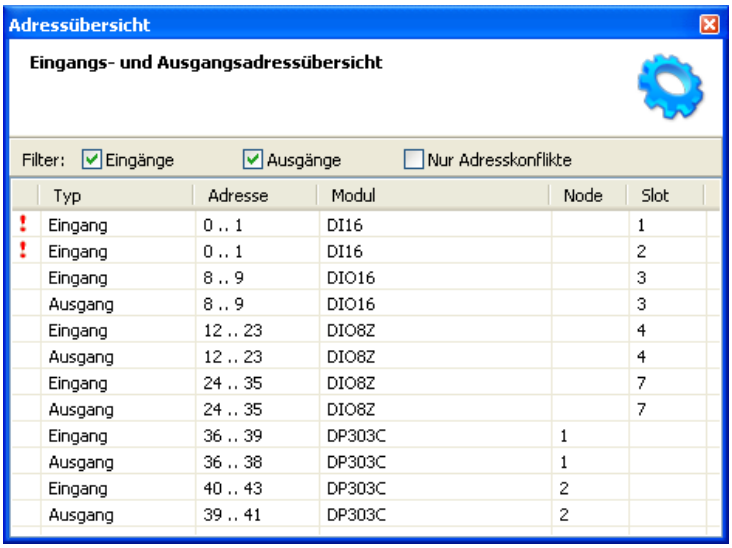
Abweichende Adressierungen in SPSen und Adressierungen von dezentraler Peripherie

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.

Konfiguration mit der Software „ConfigStage“

Ab der Version 1.0.14.7 können die verwendeten Adressbereiche bequem angezeigt werden. Unter Ansicht / Adressübersicht erscheinen alle vergebenen Adressen.
Bei Adressvergabe wird bereits ein Plausibilitätstest durchgeführt und bei Überscheidungen eine Warnung angezeigt.

Wird diese Warnung ignoriert, dann werden die Adresskonflikte in dieser Übersicht mit einem roten Ausrufezeichen angezeigt.

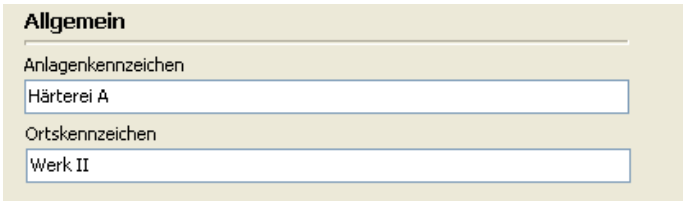


Typ	Adresse	Modul	Node	Slot
! Eingang	0 .. 1	DI16		1
! Eingang	0 .. 1	DI16		2
Eingang	8 .. 9	DIO16		3
Ausgang	8 .. 9	DIO16		3
Eingang	12 .. 23	DIO8Z		4
Ausgang	12 .. 23	DIO8Z		4
Eingang	24 .. 35	DIO8Z		7
Ausgang	24 .. 35	DIO8Z		7
Eingang	36 .. 39	DP303C	1	
Ausgang	36 .. 38	DP303C	1	
Eingang	40 .. 43	DP303C	2	
Ausgang	39 .. 41	DP303C	2	

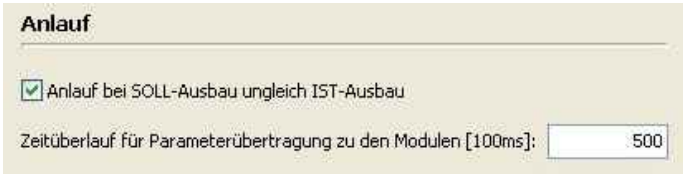
CPU-Einstellungen

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

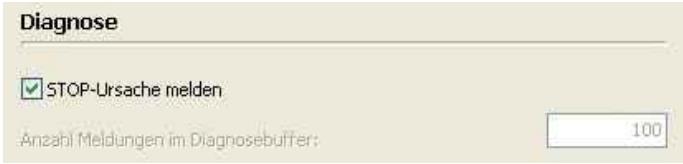
Allgemein
Angabe von Anlagen- und Ortskennzeichen



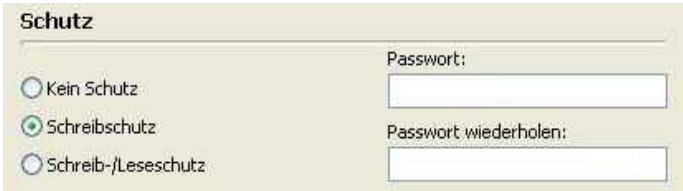
Anlauf
Haken gesetzt: CPU soll auch anlaufen, wenn der tatsächliche Ausbau anders als im Projekt eingetragen ist.
Zeitüberlauf: Wert multiplizieren mit 100 Millisekunden, Maximalwert auf 1000 Sekunden beschränkt



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



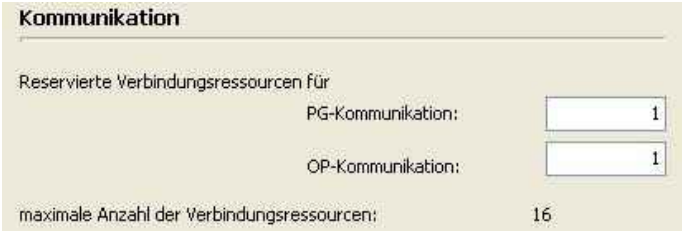
Know-how-Schutzmöglichkeiten
Hier wird der Passwortschutz vom Simatic®- Manager (de-)aktiviert und das Passwort eingetragen.



Kommunikation
INSEVIS-CPUs verfügen genau wie die 315-2PN/DP über systeminterne **16 passive S7-Verbindungen.**

Zusätzlich können **bis zu 16 aktive S7-Verbindungen** in der ConfigStage parametriert und mit einer ID-Nummer versehen werden.

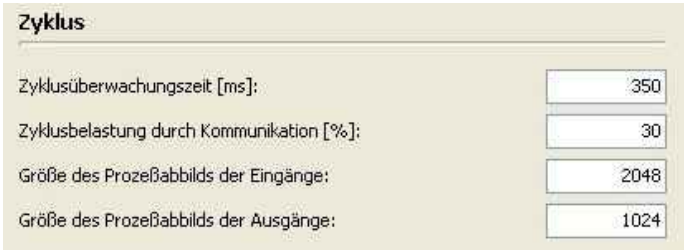
Mehr dazu bei **Information zu TSAPs**



Konfiguration mit der Software „ConfigStage“

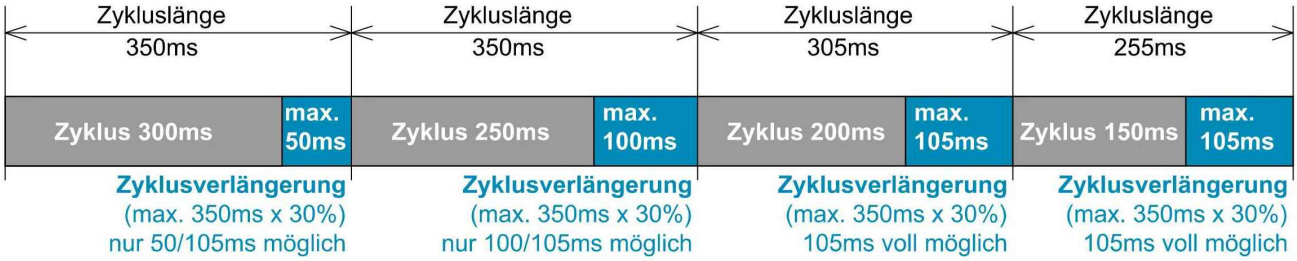
Zyklus
Zyklusüberwachungszeit
(max. 6000ms = 6 Sekunden):

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.

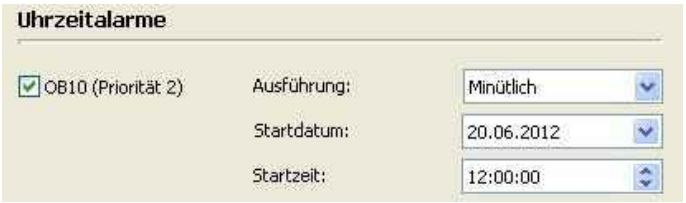


Im Beispiel kann die Zykluszeit um maximal 30% von 350ms (=105ms) verlängert werden. Wenn diese Verarbeitungszeit für die Kommunikation (und Visualisierung, etc.) nicht benötigt wird, steht sie der übrigen Verarbeitung zur Verfügung.

Taktmerker
Anwahl Merkerbyte und Angabe von
dessen Nummer ab Wert 0

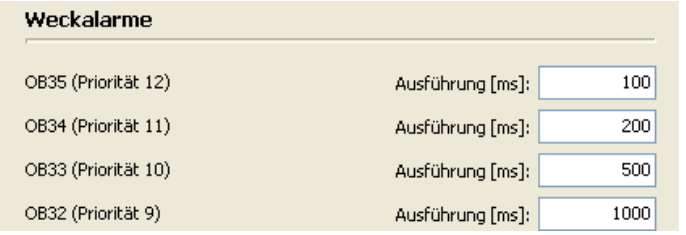
Remanenter Speicher
Merkerbytes
insgesamt 2.048 (davon 0..15 voreingestellt)
Timer und Zähler:
insgesamt 256 (keine voreingestellt)

Uhrzeitalarme
Haken gesetzt: Uhrzeitalarm aktiviert
Eintragen der Intervalle und von Startdatum und -zeit



Weckalarme
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)



Kommunikationseinstellungen

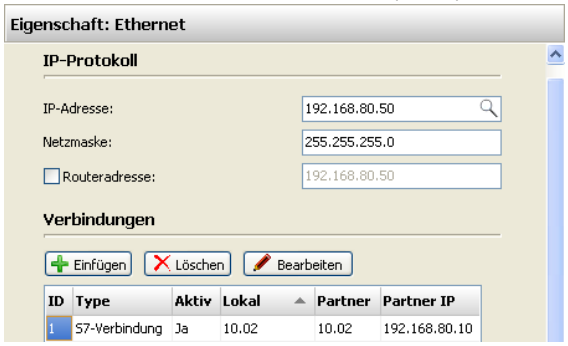
RS232 und RS485

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.

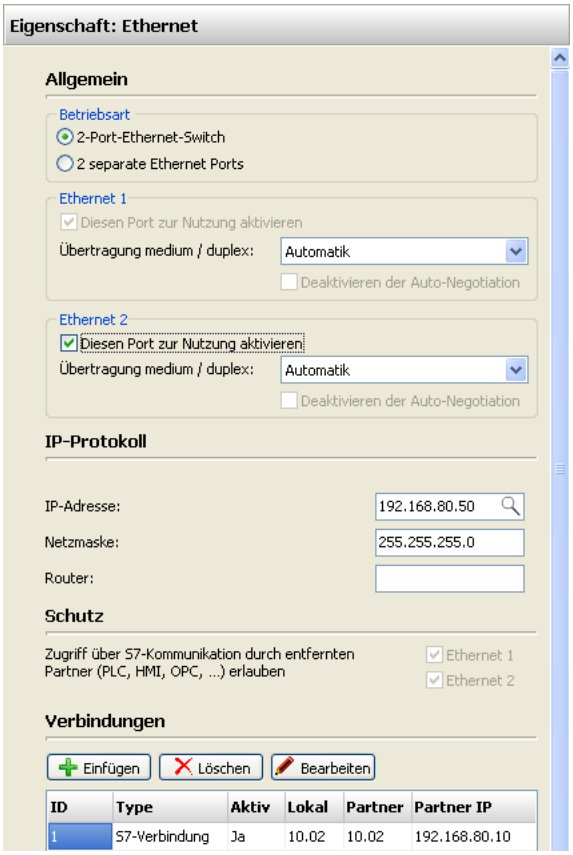
Ethernet

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

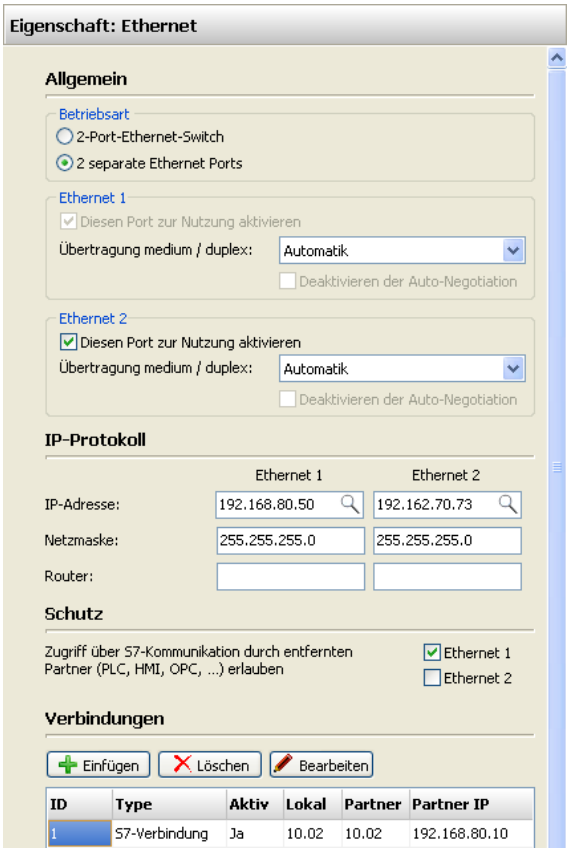
CPU-V/-P: Es steht EINE Ethernet-Schnittstelle (RJ45) zur Verfügung



CPU-T: Es stehen ZWEI Ethernet-Schnittstelle (RJ45) zur Verfügung, die entweder
- als Switch mit gemeinsamer IP-Adresse (links) oder
– 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“

Hinweis:

Um unerlaubte Datenübertragung über die klassische S7-Kommunikation von vornherein zu unterbinden, erlaubt die ConfigStage das “Abschalten” dieser Kommunikation bei einem Port. Das verhindert ungewünschte 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*

Einstellungen bei TCP Send/Receive-Verbindung
- Lokaler Port,
- Partner-Port,
- Partner-IP-Adresse

Einstellungen bei UDP Send/Receive-Verbindung
- Lokaler Port

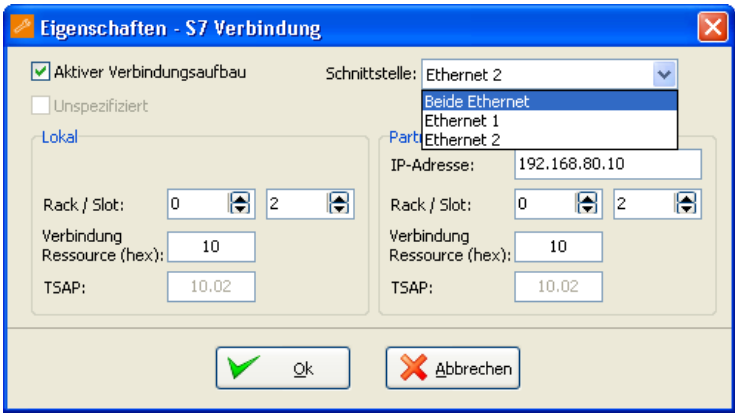
Einstellungen bei Modbus-TCP (Server)
Zuweisung der S7-Operandenbereiche für Modbus-
- Eingangsbits
- Eingangswörter
- Ausgangsbits
- Ausgangswörter

(Im Beispiel wieder das „Mapping“ beim Modbus-Server, diesmal unter Modbus TCP)

Konfiguration mit der Software „ConfigStage“

Kommunikation / Information zu TSAPs
INSEVIS-CPUs 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 parametriert 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 SimaticManager oder im TIA-Portal vorgenommen.

CAN-Einstellungen

Dezentrale INSEVIS Peripherie

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

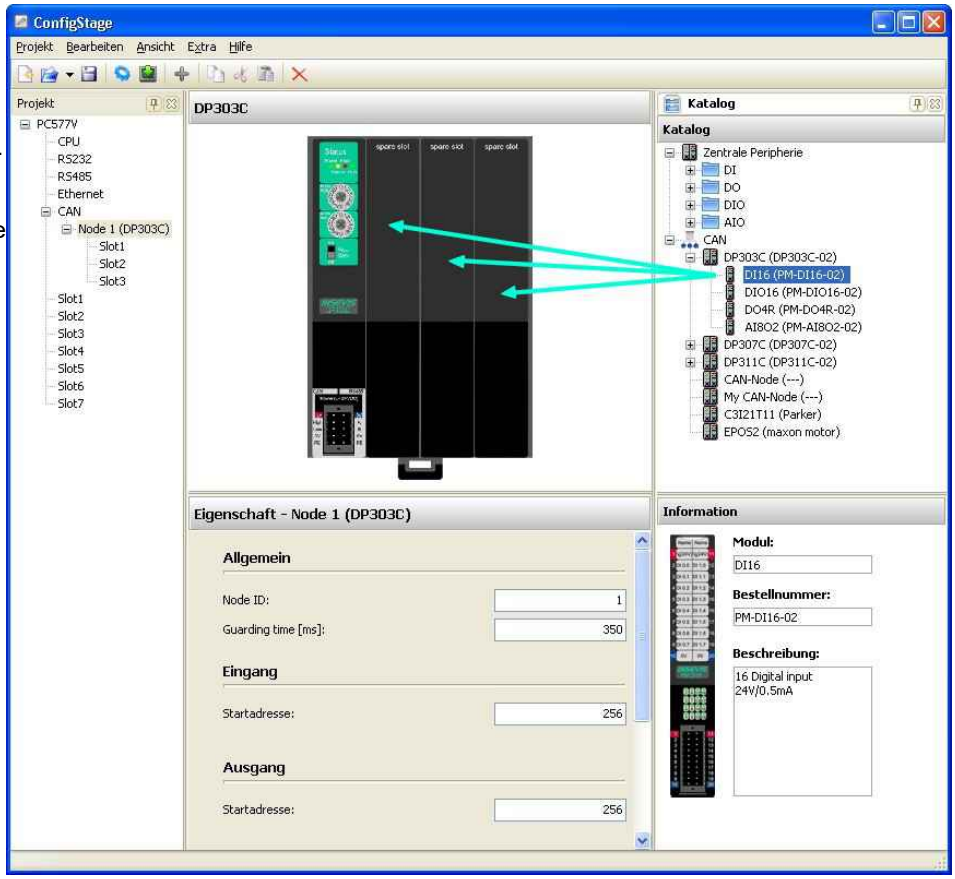
Nachdem Sie in der Startmaske eine CAN-Kopfstation von INSEVIS auf die CPU gezogen haben, erscheint diese im Projektbaum und in der Anzeige.

Jetzt können Sie für die Kopfstation Startadressen für die 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.



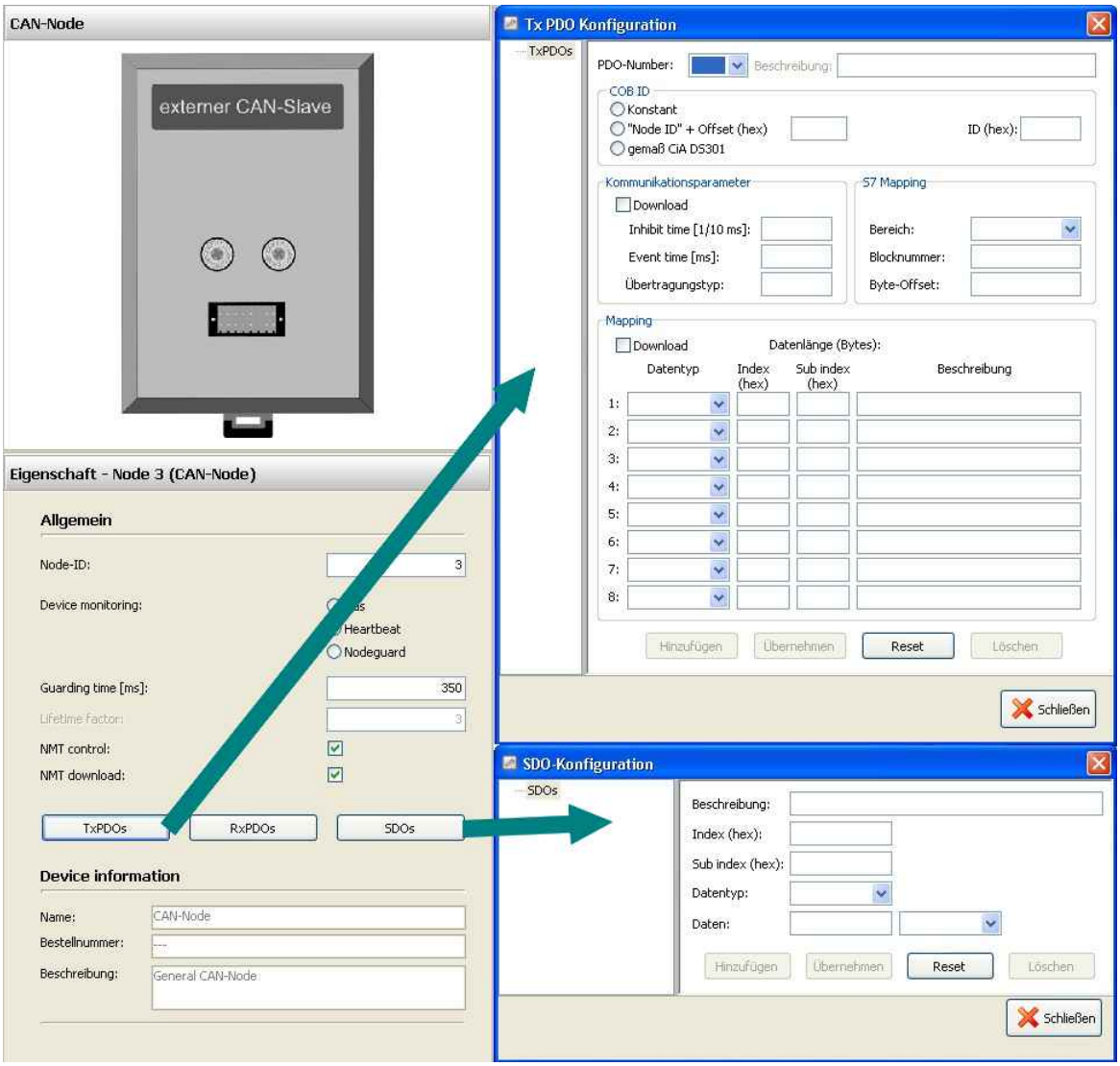
Konfiguration mit der Software „ConfigStage“

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-Identifier anderer Knotennummern verwendet werden. Diese Knotennummern sollten dann nicht mehr vergeben werden. (Es sind nur CiA-konforme TxPDO-Identifier zulässig)
Empfangene Tx-PDOs werden zwischengespeichert und kontrollpunktsynchron 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-Kommunikations-parameter das Sendeverhalten des Masters:

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, solang 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 **SDO**s definiert werden. Damit werden z.B. Betriebsmodi oder Messbereiche eingestellt.

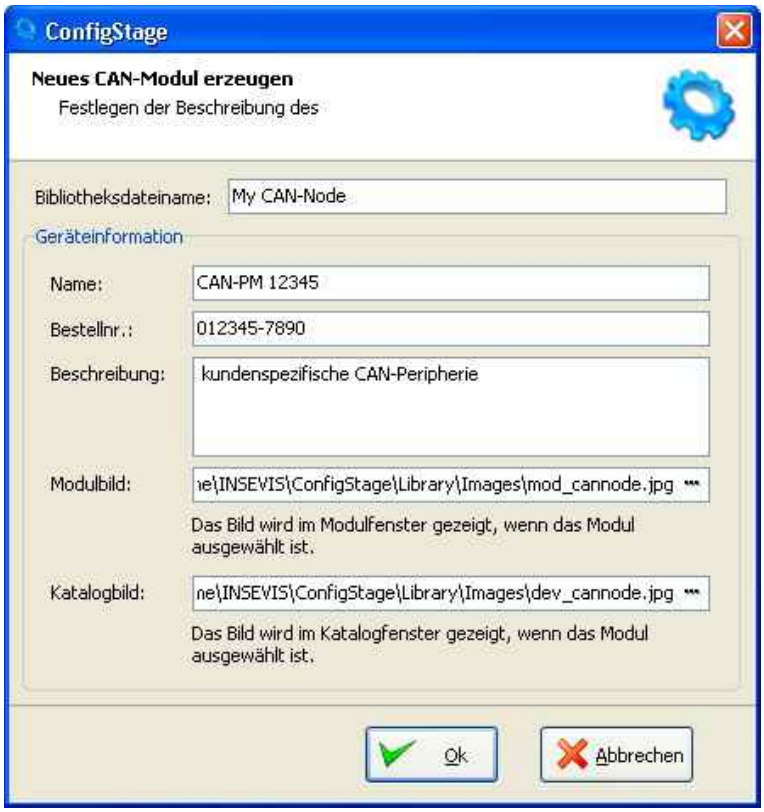
Bibliothekselemente aus eigenen CAN-Slaves erstellen

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 eint Maske, in der die genaue Beschreibung (mit Bild) vorgenommen wird.

Nach Bestätigung erscheint der neue CAN-Knoten mit den vorbelegtem Mapping in Ihrem Katalogverzeichnis auf der rechten Seite.



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.

Konfiguration mit der Software „ConfigStage“

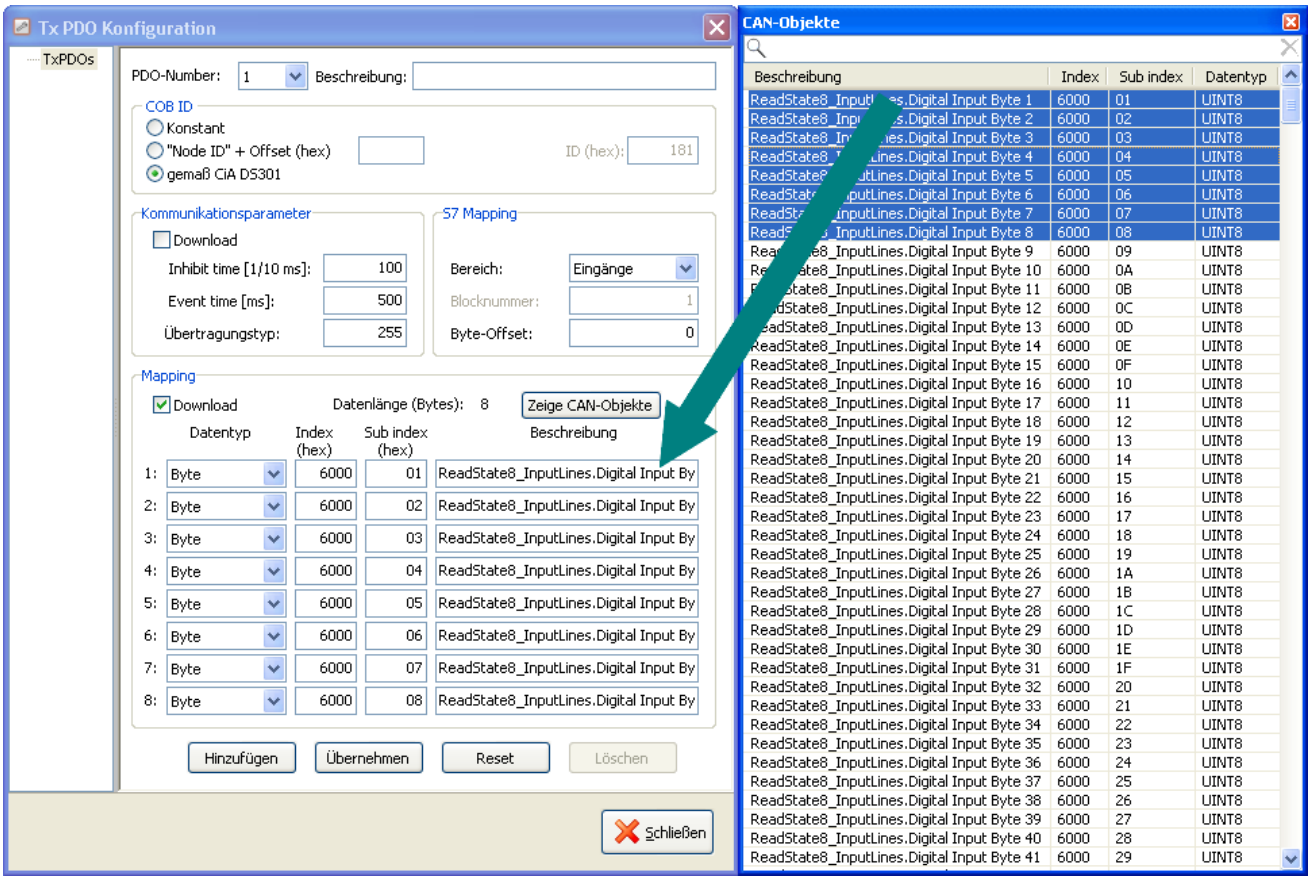
Dezentrale Fremdperipherie per EDS konfigurieren

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.

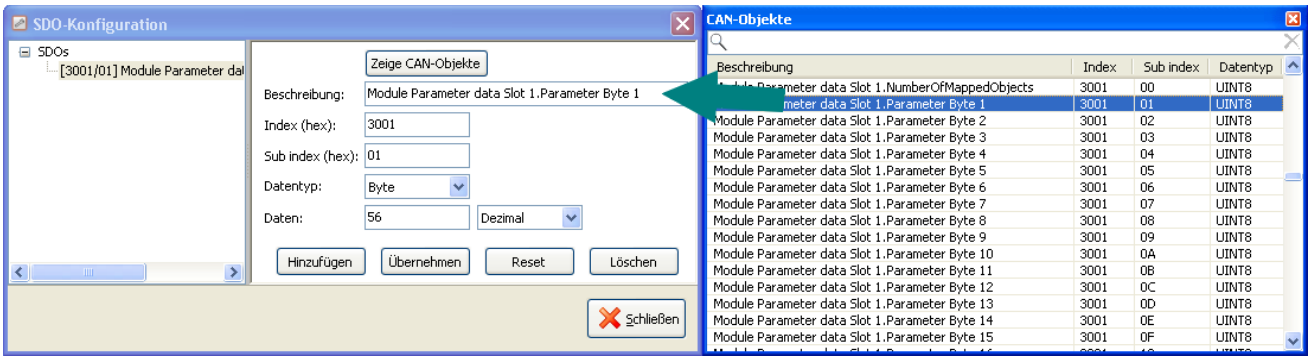
PDO-Konfiguration

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.



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 MotionController 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.

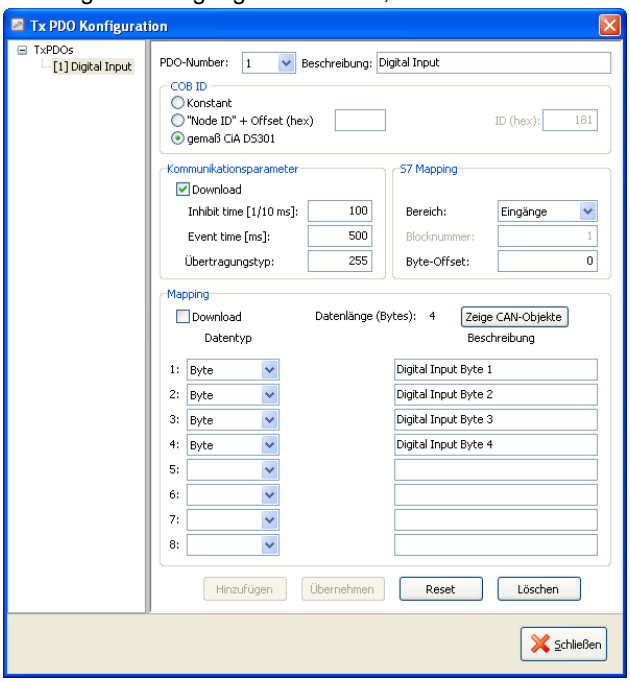
Konfiguration mit der Software „ConfigStage“

Allgemeine Beispiele für CANopen Fremdgeräte

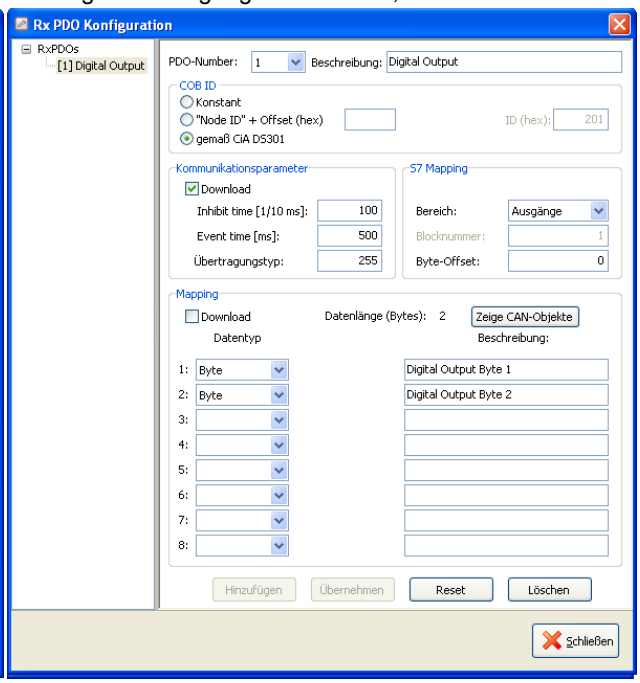
Beispielkonfiguration dezentrale CANopen-Peripherie nach DS401

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

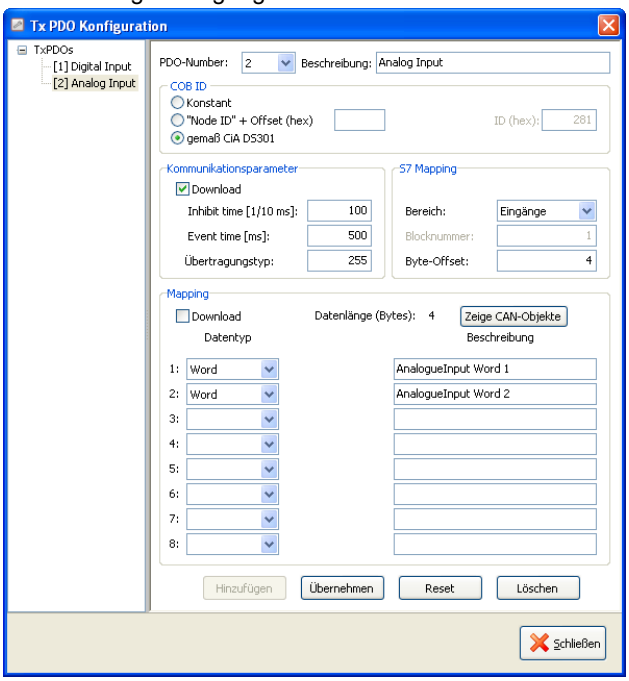
- die digitalen Eingänge in TxPDO1, z.B. 32 Bit



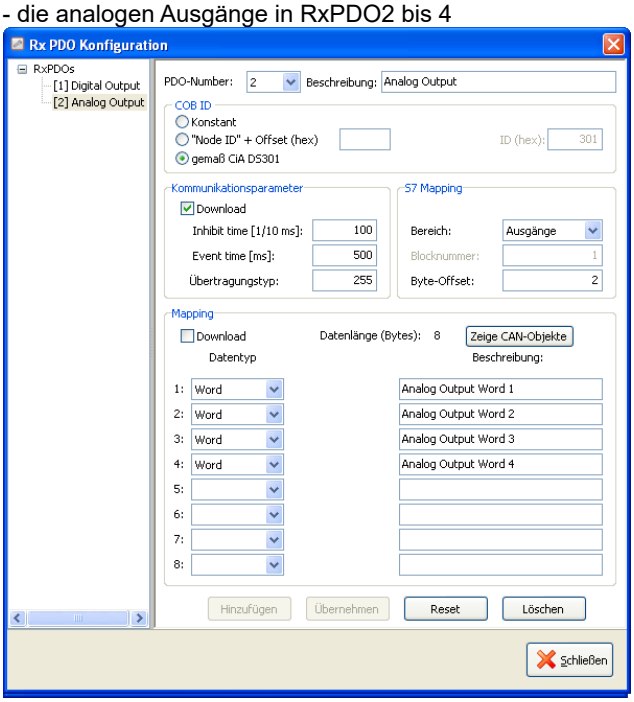
- die digitalen Ausgänge in RxPDO1, z.B. 16 Bit



- die analogen Eingänge in TxPDO2 bis 4



- die analogen Ausgänge in RxPDO2 bis 4



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

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

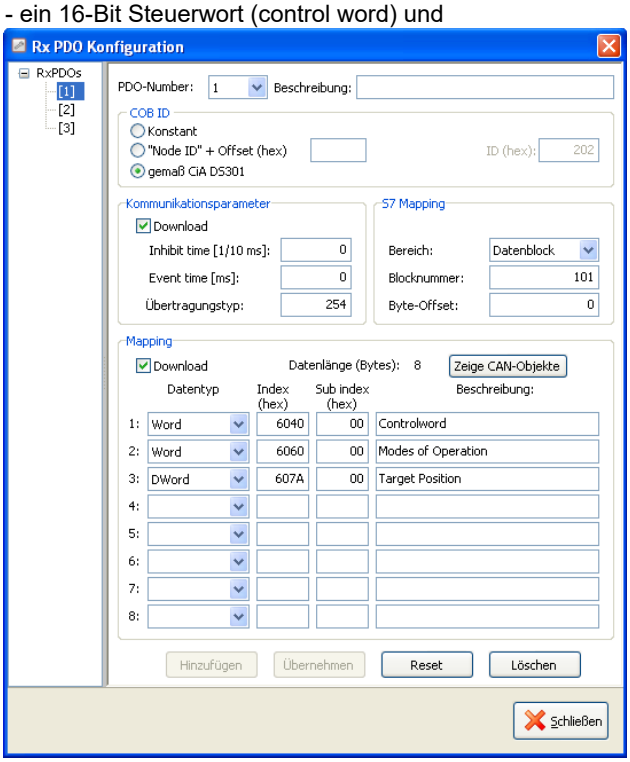
Konfiguration mit der Software „ConfigStage“

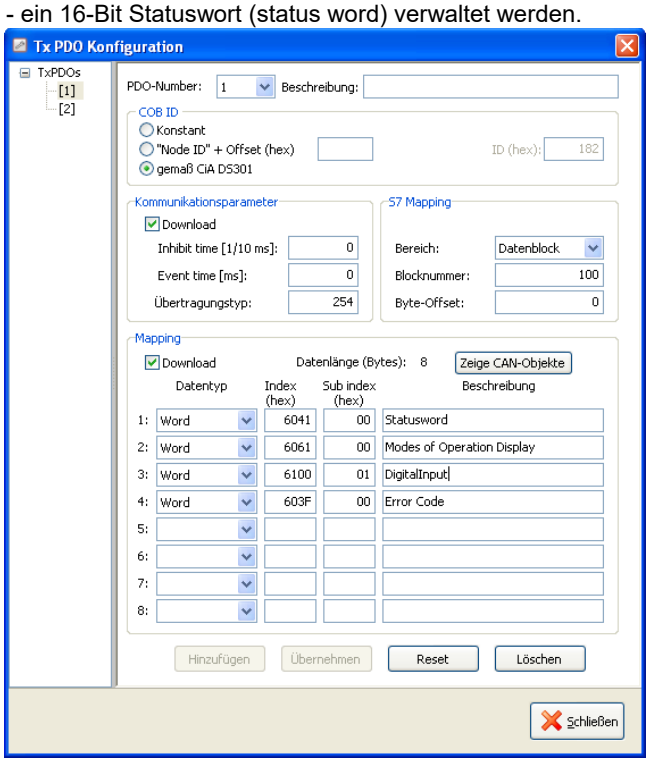
Um Parameter zu übertragen (z.B. Messbereiche für Analogmodule), müssen SDOs herstellerspezifisch konfiguriert werden.

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.

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 S7-315-2PNDP mit Firmware 3.2.7 (6ES7 315-2EH14-0AB0) stehen bei den INSEVIS-CPUs Typ V und Typ P 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) *Lokaldaten „OB86_MDL_ADDR“ (LW6) hat den festen Wert 2047 (W#16#7FF)*
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)

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	Auf Grund unterschiedlicher Hardware sind diese SFCs nicht möglich / nötig
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	IP-Adresse konfigurieren → SFB 129 IP_CFG verwenden

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 TRECV	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

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; *Parameter „LADDR“ muss festen Wert 2047 haben (W#16#7FF)*
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 / 66 / 67 / 68 / 69 / 72 / 73 / 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 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
SFC 214 LCD_BRIG	Hinterleuchtungshelligkeit einstellen
SFC 254 RAM2ROM	alle DBs von RAM nach ROM kopieren

Import der INSEVIS-Zusatzbausteine in das S7-Projekt

Achtung

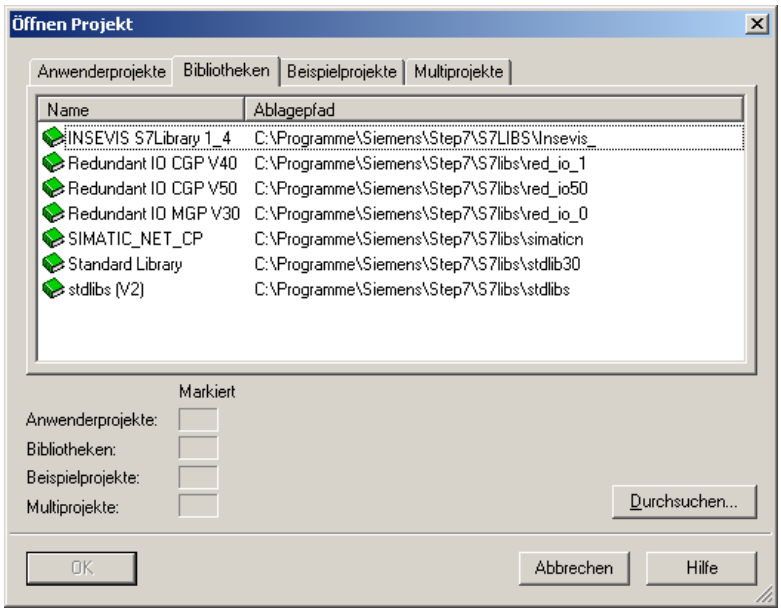
Die INSEVIS- Bausteine stehen als **S7-Bibliothek** im Internet unter www.insevis.de/de/service zum freien Download bereit. Es wird immer die aktuellste mit den meisten Funktionen bereitgehalten, die auch das aktuellste Betriebssystem der INSEVIS- Steuerung erfordert. (z.B. Insevis_S7-library_from_2_1_5.zip - funktioniert ab Betriebssystem 2.1.5 und höher).

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

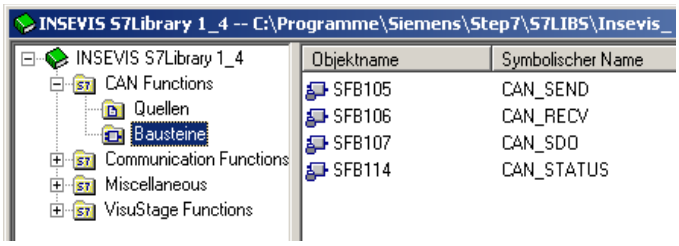
1) Datei → dearchivieren → Auswahl von INSEVIS S7-Bibliotheks.zip

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



Die Hinweise für das Einbinden der INSEVIS-SFCs in Win-S7 stehen in den Dokumenten „First Steps“, die im Servicebereich der INSEVIS-Webseiten www.insevis.de/de/service zu finden sind.

Hinweis: → VIDEO verfügbar
Für dieses Beispiel ist auf den INSEVIS-Service-Seiten ein Video verfügbar

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

Achtung

Die INSEVIS- Bausteine stehen als **S7-Bibliothek INSEVIS Library (TIA V12).zip** im Internet unter www.insevis.de/de/service zum freien Download bereit. Es wird immer die aktuellste mit den meisten Funktionen bereitgehalten, die ab INSEVIS- Betriebssystem 2.0.39 zur Verfügung und auf das jeweilige benötigte Betriebssystem referenziert sind. (z.B. Insevis_TIA-library_from_2_1_5.zip - funktioniert ab Betriebssystem 2.1.5 und höher).

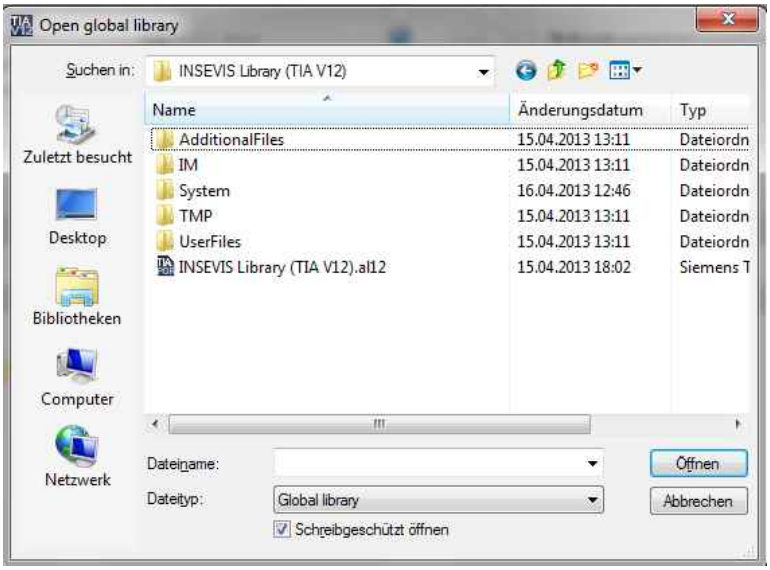


Bild links:

Aktuellste Version der INSEVIS Bibliothek für das TIA-Portal downloaden, speichern und in einen beliebigen Ordner extrahieren.

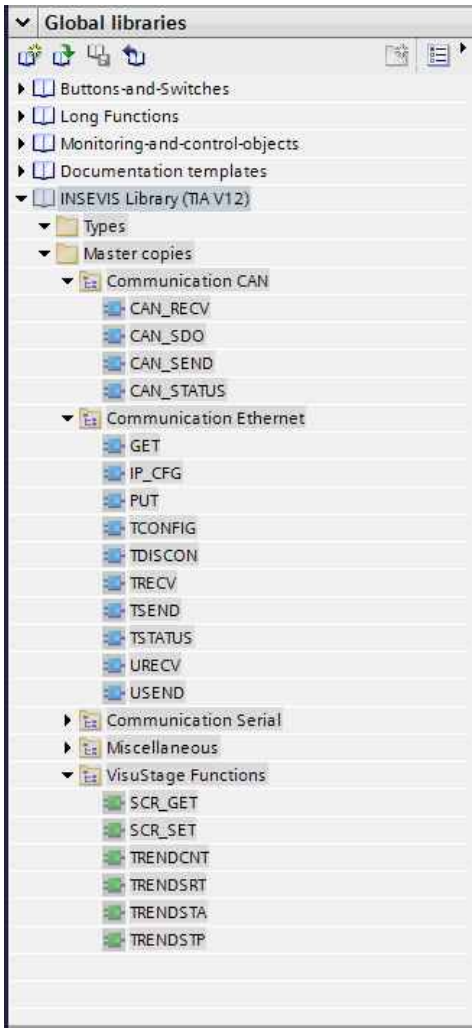
Bild rechts:

Bibliothek "***INSEVIS Library (TIA V2).al12***" aus jeweiligen Ordner im TIA-Portal öffnen als "Global library".

→ *Für Details zu Bibliotheksöffnen 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.*



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 batteriegepuffertem WArbeitsspeicher / 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 Geräteaustausch

Mit dem **SFC 208** BACKUP werden ALLE Daten (S7-Programm, Visualisierung, 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 Produktionzyklus´weiterarbeiten.

Manuelles Programm-Backup für Gerätehersteller

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

- Auf dem PC ein Verzeichnis „**backups**“ in Micro-SD-Karte erstellen,
- Erstellen eines S7-Memory-Card-Datei mit dem Namen „**s7programs.wld**“
  mit Menue: „Datei → Memory Card Datei → Neu“
- ALLE S7-Programme, Datenbausteine **und Systemdaten** (enthalten ConfigStage-Konfiguration) in S7-Memory-Card-Datei mit Drag´n Drop kopieren, Datei schließen und diese Datei in das SD-Verzeichnis „backups“ kopieren
- Visualisierungs-Binärdatei mit der VisuStage erstellen, umbenennen in „**visudata.vsbin**“ und auch diese Datei in das SD-Verzeichnis „backups“ kopieren
- Ab ServiceStage 1.0.1.7 kann man diese Schritte durch die Funktion „OnlineBackup“ automatisch ausführen lassen (empfohlen)

→ 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)

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 zurückgelesen werden. Ein Auslesen per Ethernet und Abspeichern in Netzlaufwerken ist auch 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 parametriert

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

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.

Bedeutung von CSV_STRUCT

Erstes Byte	Zweites Byte	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	Zweites Byte	
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

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
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.

```
      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
```

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 geschalten 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 speichen. (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 is 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) 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:

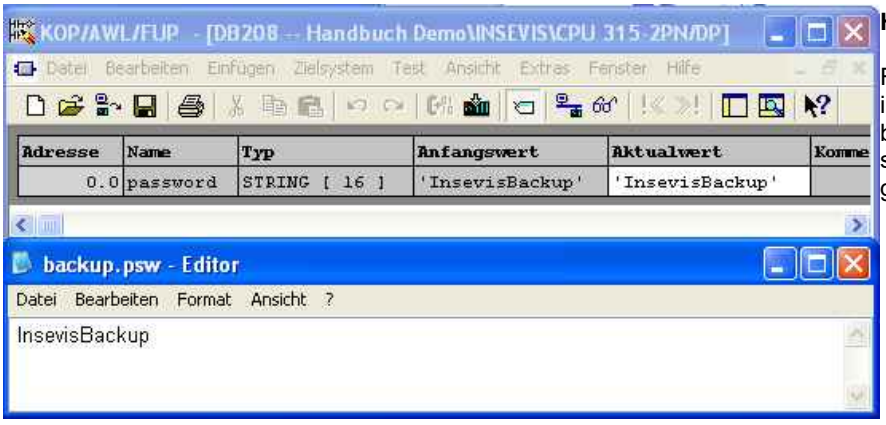
- Das STOP-Ereignis wird im Diagnosepuffereintrag mit
  EVENT ID W#16#497C "STOP triggered by integrated technology"
  gekennzeichnet. Der Backupvorgang dauert bis zu 5 Sekunden.
  .
  Das Backupergebnis wird im Diagnosepuffereintrag wie folgt gekennzeichnet
  EVENT ID 16#BF00
  OB 16#00
  PK 16#FF
  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
```



Hinweis:

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.

Achivdaten lesen von Micro-SD-Karte mit SFC 209 "READFILE"

Diese Funktion wird genutzt, um auf de 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

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)
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 compatible)

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 Dateinname
 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
```

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

Rückgabewert (RETVAL)	BUSY	Description
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 compatible)

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
```

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

Rückgabewert (RETVAL)	BUSY	Description
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-kompatible)

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
```

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

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 (z.B. SFC207 “ARCHIVE” …)
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)

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 sind, als in CSV_STRUCT „field count“ festgelegt, werden die überzähligen Datenfelder ignoriert.
- Wenn die Prozessdatengröße für den Zielbereich (DATA) kleiner sind 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
```

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

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)

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 connection ID number.
                                     In Panel-PLC (e.g PC700P) must be W#16#0.
 BRIGHTNESS      :=MW2024        // LCD Brightness value to change
                                     (in present, 0..100)
 RET_VAL         :=MW1000        // Result / return code
 DONE            :=M1002.0       // TRUE = Successful created
 BUSY            :=M1002.1       // TRUE = Request is in process
 ERROR           :=M1002.2       // TRUE = Error occurred
 BACKLIGHT       :=M2026.0       // LCD Back light status
 BRIGHTNESS_CUR :=MW2028         // Current LCD brightness value in present

O     M   1002.0                  // successful done
O     M   1002.2                  // or failed
R     M   1502.3                  // reset request signal

UN    M   1002.2                  // if not failed, then
BEB                               // end the function

L     MW  1000                    // error occurred
// error evaluation
// ...
BEA                               // end of function
```

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

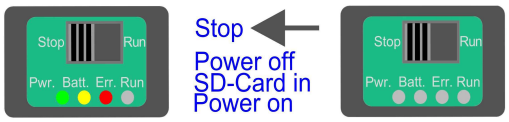
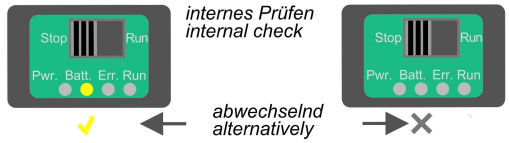
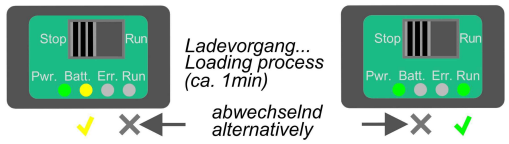
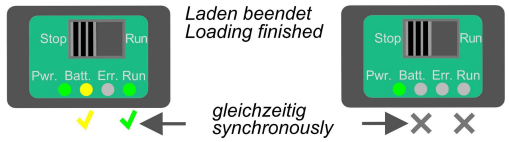
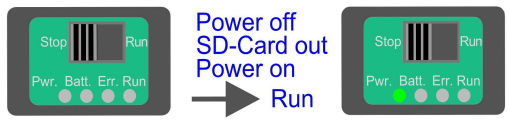
Ziel: Kopieren aller DBs in den Ladespeicher (Aktuelle Werte im Ladespeicher werden dadurch überschrieben).

```
      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, das Betriebssystem zu aktualisieren, **ohne dass das Anwenderprogramm beeinflusst wird**.

Dazu benötigt man lediglich eine Micro-SD-Card **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-Card - AG Power on	1) 	
Anzeige **„interner Check“** *(hängt von bestehenden Softwarestand ab)* LEDs Batt. (gelb) blinkt, solange CPU-Status geprüft	2) 	
Anzeige **„Ladevorgang läuft“** *(Achtung: keine Spannungs unterbrechung erlaubt!)* Power-LED leuchtet permanent LEDs Batt. (gelb) und Run (grün) blinken abwechselnd	3) 	a) b)
Anzeige **„Ladevorgang beendet“** Power-LED leuchtet permanent LEDs Batt. (gelb) und Run (grün) blinken gleichzeitig	4) 	
geladenes Betriebssystem initialisieren - Gerät Power off - SD-Karte entfernen - Gerät Power on	5) 	

Hardwareseitiges Urlöschen

Prinzipiell gibt es 2 verschiedene Arten urzulöschen:

- 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

Power off
Power on

Warten bis LED 1-3 leuchten
ca. 2-3 Sec. nach Power on

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

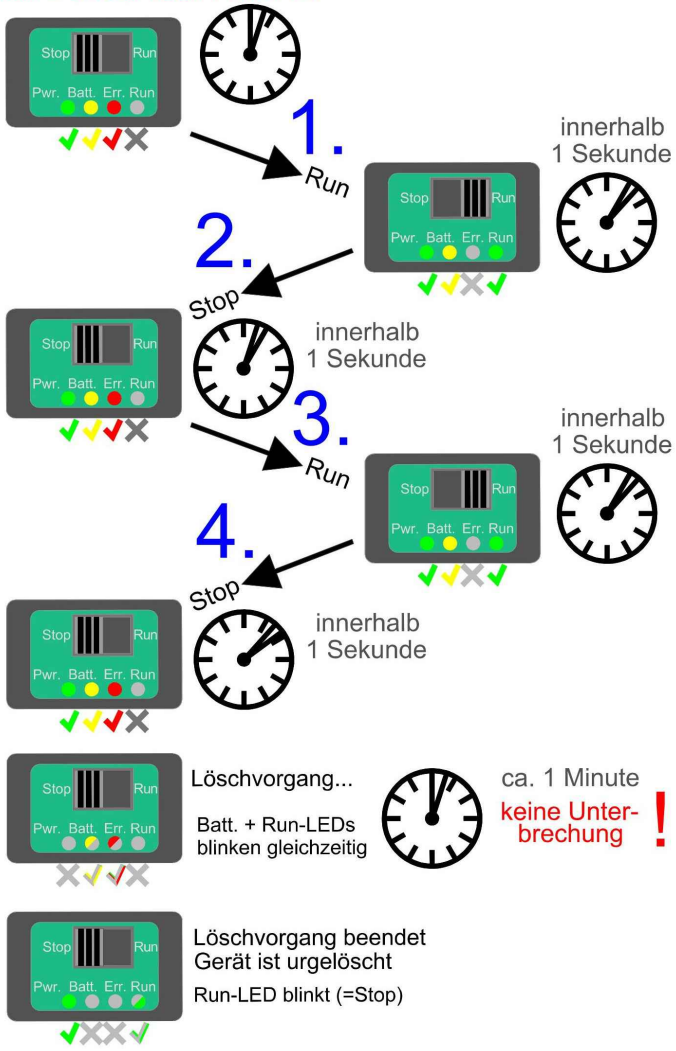
2 Run/Stop-Schalter nach Stop
rote Error-LED geht an
grüne Run-LED geht aus

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.

Ist der Ladevorgang beendet, oder steckt keine Micro-SD-Karte verbleibt die Steuerung im Stop-Zustand (RUN-LED blinkt).

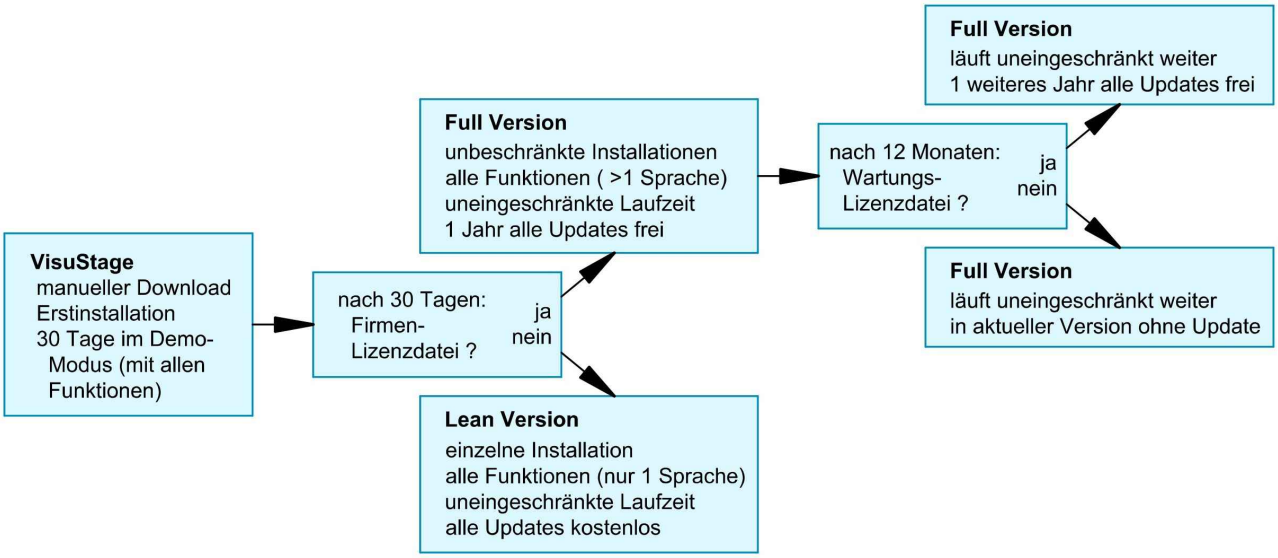
* Während der ersten 3 Sekunden ist ein Wert für die CPUs -V und-P. Die größere CPU-T benötigt länger für den Bootprozess (so etwa 6-8 Sekunden). Falls ein Display dabei ist, ist das der Zeitpunkt nach Ablauf des Fortschrittsbalkens.

Download und Lizenzierung

Die jeweils aktuellste Versionen der VisuStage liegt auf den INSEVIS-Webseiten zum kostenlosen Download bereit.

Bei Erstinstallation fragt die VisuStage nach der Lizenzdatei. Bei Nichteingabe läuft die Software 30 Tage im vollen Umfang als Demoversion und verlangt danach die Eingabe des Pfades zur Lizenzdatei. Über Den Reiter „Hilfe“ → „Über“ kann jederzeit der Pfad zu einer Lizenzdatei angegeben werden.

Die Lizenz kann innerhalb eines Unternehmens unbeschränkt oft für eine unbeschränkte Anzahl Installationen benutzt werden. (Bei Windows 7 oder Windows Vista die VisuStage einmalig als Administrator ausführen, damit nicht jedesmal die Lizenz abgefragt wird. (Rechte Maustaste auf visustage.exe im Explorer und „als Administrator ausführen“)



Systemfunktionen

Das S7-Betriebssystem bietet folgende Systemfunktionen (SFC) an, um die Visualisierung über S7-zu steuern

- **SFC 200** – Abfrage der aktuellen Bildschirm-ID- Nummer
- **SFC 201** – Bildschirmwechsel
- **SFC 202** – Trenderfassung starten
- **SFC 203** – Trenderfassung stoppen
- **SFC 204** – Trenderfassung fortsetzen
- **SFC 205** – Trendstatus abfragen
- **SFC 206** – Rezepturdatensätze lesen/schreiben

Um diese SFC´s zu nutzen, wird die **S7-Bibliothek** von INSEVIS benötigt, die im DownloadBereich der INSEVIS-Webseiten zur Verfügung stehen. Eine Beschreibung zu deren Installation erfolgt im Kapitel „Systemfunktionen“.

Kommunikation zwischen SPS und abgesetztem Panel

In einer Panel-SPS werden diese SFCs von einer gemeinsamen CPU für SPS- und Panelfunktion benutzt.
Bei SPSen mit abgesetzten Panel-HMIs ist folgendes zu beachten:

- Es können immer nur einzelne Abarbeitungsaufträge von SFCs bedient werden. Wenn der Parameter "BUSY" des jeweiligen SFCs den Wert FALSE zurück gibt, ist der Auftrag erledigt und ein neuer SFC kann abgearbeitet werden.

- Die Anlage der Verbindungsparameter von INSEVIS-Panel zu INSEVIS-SPS bzw. Fremd-SPS ist unter „Ressource Partner-SPS-Gerät“ in diesem Handbuch beschrieben.

Die Verbindungsnummer im S7-Programm werden an den Parameter LADDR übergeben
(z.B. LADDR := W#16#1 für die Panel-Verbindung mit ID-Nr. 1)

Hier z.B. kann in der abgesetzten SPS auf das Panel (z.B. mit der Verbindung mit ID-Nr. 1) zugegriffen werden, um einen Seitenwechsel von der SPS aus zu erzwingen.

```
CALL SFC201
   LADDR   :=  W#16#1
   SCREEN  :=  MW1006
   RET VAL :=  MW1008
   BUSY    :=  #busy
```

Generelle Einstellungen der Programmoberfläche

Oben: **Standard-Funktionsleiste** (selbsterklärend bzw. erklärt, wenn der Mauszeiger über dem Objekt steht)

Ausrichtungsfunktionen (2 und mehr Objekte)

ZUERST das Master-Element markieren,
DANN die anderen, die sich an dem Master ausrichten sollen
dann gewünschte Ausrichtung wählen (links, horizontal zentriert, rechts, oben, vertikal zentriert, unten)

Größenfunktionen (2 und mehr Objekte)

ZUERST das Master-Element markieren,
DANN die anderen, die sich an dem Master ausrichten sollen
dann gewünschte Größe wählen (gleiche Breite, gleiche Höhe, gleiche Höhe und Breite)
bzw. multiple Anpassung von Dimensionen verschiedener markierter Objekte (rechts)
mit Tastenkombination „STRNG+Pfeiltasten“ pixelweises Ändern der Größe von markierten Objekten

Ausrichtungsfunktionen (1, bzw. 3 und mehr Objekte)

ZUERST das Master-Element markieren,
DANN die anderen, die sich an dem Master ausrichten sollen
ZULETZT das Element, dass am weitesten vom Master entfernt bleiben soll
dann gewünschte Ausrichtung wählen (gleicher horizontaler Abstand, Einzelobjekt mittig in Bildschirm platzieren, gleicher vertikaler Abstand)

Raster-/ Fangfunktion (selbsterklärend)

Unter Extra / Einstellungen Rasterabstand parametrieren
Rasteranzeige/Fangmodus (de-) aktivieren durch Mausklick

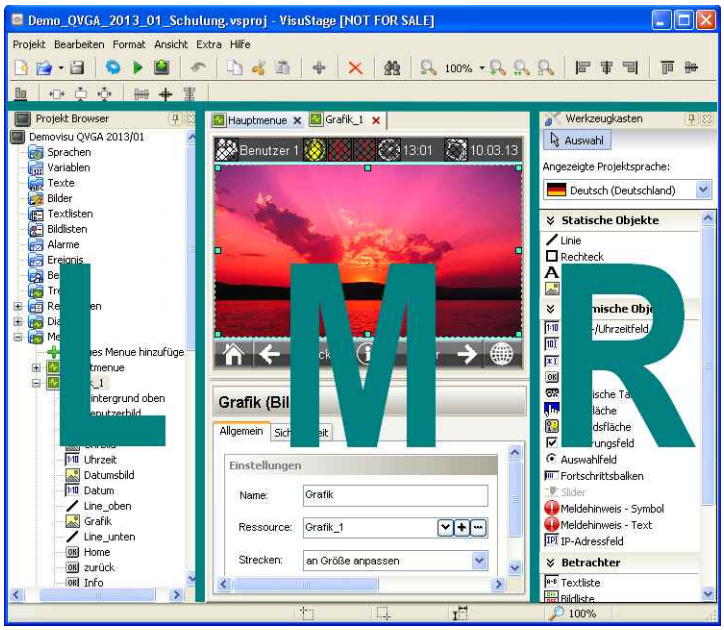
Allgemeine Einstellungen (ohne Abbildung, selbsterklärend)
Es empfiehlt sich, den Panelrand zu verbergen, damit die PC-Bildschirmfläche besser ausgenutzt werden kann. Raster- und Fangfunktionen erleichtern das Anordnen der Objekte enorm.

Oben: **Befehlsleiste** mit Reitern
Links: **Projekt-Browser** mit **Ressourceneinstellungen**,
hier werden die Ressourcen zugewiesen und parametriert
Bildschirme (Menues, selbstaufbauend, Anzeige des obersten Menues nach dem Bootvorgang)

Mitte: Darstellung der projektierten **Anzeige**
Anzeige der **Objektparameter** (darunter),
hier werden die Eigenschaften des markierten Objektes parametriert

Rechts: **Werkzeugkasten** mit
Projektsprache
(in der Sprache wird das Projekt angezeigt)
Funktionskatalog
mit den verfügbaren Objekten
Bibliothekskatalog
mit den angelegten Bibliothekselementen

Unten **Meldungsfenster**
zeigt Meldungen, Menuereferenzen, Querverweise und Suchergebnisse an (mit F6 ein/ausblendbar)



Beispielprojekt
Die nachstehende Beschreibung bezieht sich auf das Demoprojekt „**Demo_QVGA_2015_02**“, welches im Internet zum Download bereitsteht. Dieses Projekt und die im Handbuch beschriebenen Themen erfordern die VisuStage-Version 2.0.4.7 und die Betriebssysteme 2.3.1 (Panel-SPS) und 1.2.1 (Panel-HMI)

Details erschließen sich am besten durch gleichzeitiges Benutzen von dieser Dokumentation und dem Demoprojekt.

Visualisierung mit der Software „VisuStage“

Generelle Funktionsweise

Die VisuStage verwendet SPS-Variablen in verschiedensten Visualisierungsobjekten für verschiedene Bildschirme (Menues). Die Quelldaten, Bilder und Texte mit den installierten Fonts auf dem VisuStage-PC werden zu einer Binärdatei kompiliert, die per Ethernet in das PanelHMI/-SPS übertragen wird. Das dortige Betriebssystem interpretiert diese Daten und erzeugt Bilder und aktiviert Eingabefelder auf dem Touchdisplay des Gerätes. Es können keine Quelldaten mehr von dem Panel zurückgelesen werden. Eine Remote-Anzeige ist per RemoteStage möglich, die dafür die erzeugte Binärdatei (aus dem PC oder aus dem Panel-HMI/-SPS - wenn erlaubt) benötigt.

Empfohlene Vorgehensweise

Eine systematische Arbeitsweise reduziert den Projekterstellungsaufwand (nicht nur bei der VisuStage).

- Importieren Sie die S7-Variablen direkt aus den S7-Datenbausteinen (mit Symbolik) des SimaticManagers
- Legen Sie sich immer wiederkehrende Elemente in einer Menüvorlage fest, die Sie mehrfach verwenden können
- Haben Sie alle Texte z.B. in deutsch geschrieben, können Sie neue (leere) Sprachen anlegen und diese im CSV-Format exportieren, extern übersetzen lassen und dann im CSV-Format wieder importieren.
- Erstellen Sie die spezifischen Bilder (Images) vorab in beliebigen Bildverarbeitungsprogrammen (Als Dateiformat empfehlen wir "png", weil dieses mit verlustloser Komprimierung arbeitet und die Definition von Transparenz über einen Alpha-Kanal ermöglicht. Das sichert die Sichtbarkeit des Hintergrundes z.B. an abgerundeten Ecken)
- Verwenden Sie NAMEN für Ihre Ressourcen, das ermöglicht Ihnen, auch bei größeren Projekten die Übersicht zu behalten.

Ressourcen schonen

Für den Ablauf ist es wichtig, ressourcenschonend zu arbeiten, d.h., den Speicherbedarf und die nötige CPU-Last der Visualisierung gering zu halten. Folgende besondere Hinweise sind daher zu beachten:

- Benutzen Sie Buttons mehrfach (exakt identische Buttons (Auflösung, Image) werden nur 1x abgespeichert, enthaltene Symbole oder Texte sind davon nicht betroffen)
- Überlappen Sie keine dynamischen Objekte (Diese Überlappung ignoriert den Hintergrund bei bei transparenten Bereichen, welches optisch unschön wirkt)
- Besser weniger Screens mit mehr Objekten als viele Screens mit wenigen Objekten (Jeder Screen wird als bmp abgespeichert und benötigt Speicherressourcen.)
- Verwenden Sie nur die benötigten Sprachen, da viele Sprachen den Speicherbedarf vergrößern

Übersicht behalten, Projekt bereinigen

Benutzen Sie Funktionen zu besseren Orientierung (Wo wird welche Ressource verwendet?), zur Reduzierung angezeigter Ressourcen (Filter) und zum Löschen unbenutzter Ressourcen (Bereinigen)

- Behalten Sie die Übersicht über verwendete Ressourcen mit der Funktion „Querverweis“
- Löschen Sie nicht verwendete Ressourcen durch die Funktion „Bereinigen“ (im Pull-Down-Menü „Bearbeiten“)

Standardseiten als Projektvorlagen erstellen

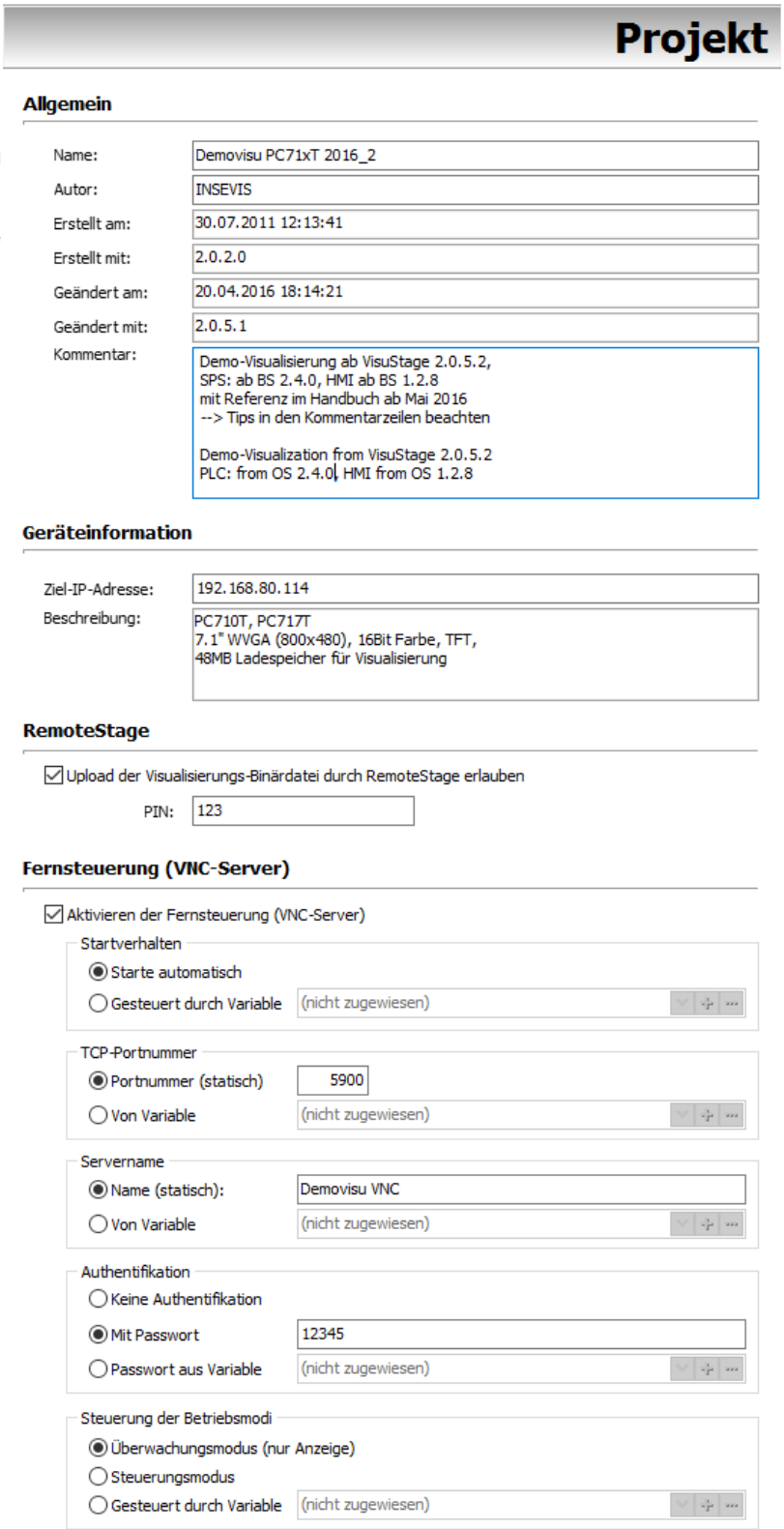
Einmal aufwändig erstellte Objektgruppen sollten für mehrere Projekte zur Verfügung stehen, das ergibt ein einheitliches Bedienphilosophie und spart Zeit.

- Benutzen Sie einmal erstellte Objektgruppen als Bibliothekselement weiter
- Benutzen Sie die Menuevorlage- Funktion innerhalb eines Projektes

Visualisierung mit der Software „VisuStage“

Projekteinstellungen

Unter dem Reiter „Projekt“ werden grundlegende Einstellungen vorgenommen.



Angabe der Ziel-IP-Adresse

- Die Verbindung zwischen der Steuerung und dem Visualisierungsrechner erfolgt über Ethernet-S7-Kommunikation. Daher muss im Rechner die IP-Adresse derjenigen Panel-SPS / des Panels angegeben werden, an welche(s) die Visualisierung geschickt werden soll. **(Hier wird NICHT die IP-Adresse der Steuerung geändert!)**
- Doppelklick auf „Projekt“
- Eingeben der Daten
- Bestätigung mit „Enter“

Dateiformate

Die VisuStage erzeugt 3 Dateiformate:

- ***.vsproj** enthält die Quelldaten der Visualisierung
- ***.res** enthält die Ressourcendaten der Visualisierung
  Hinweis: *Bei Weitergabe der Quelldaten diese beiden Dateien verschicken!*
- ***.bin** enthält die zu übertragenden Binärdaten
  Hinweis: *Enthält keine Quelldaten*

Datenschutz bei Upload

- Angabe eines PIN-Codes, um unberechtigten Upload der Visualisierungs- Binärdatei mit der RemoteStage zu verhindern
- Wird von der RemoteStage beim Upload abgefragt
- Muss bei „Online-Backup“ des ServiceStage eingeschaltet sein

VNC-Server (nur CPU-T-Geräte)

- Einstellung, ob VNC-Server immer oder variablenabhängig zur Verfügung steht
- statische / dynamische Portnummer (Verwendung Port 5900 empfohlen!)
- Passworteinstellungen (Wird im VNC-Client benötigt)
- Anzeige im VNC-Client
  - passiv (nur 1:1 Anzeige)
  - aktiv (übernimmt das HMI)
  - wechselnd durch Variable (z.B. Schlüsselschalter)

Mit Verwendung einer VNC-Client/-Viewer als App ist die Visualisierung auch in mobilen Endgeräten möglich.Je nach Anbieter sind die Bedienungs- und Sicherheitseinstellungen selbst vorzunehmen.
Für den Datenschutz ist der Anwender zuständig! Benutzen Sie den VNC-Server nicht ohne Authentifikation und erlauben Sie nur bekannten Geräten den Zugang.

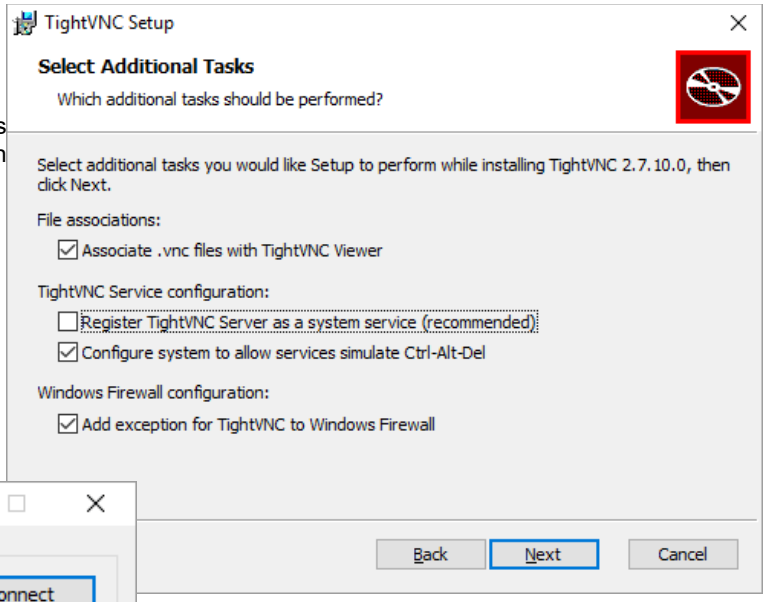
Einrichten eines VNC-Viewers/ Clients

Bei der Wahl des passenden VNC-Viewers/-Clients ist auf folgendes zu achten:

- Unterstützung RFB V 3.8 mit Verschlüsselung ZRLE, RAW-Protokoll
- Abwahl von „Clipboard transfer“ (Copy & Paste für Texte)

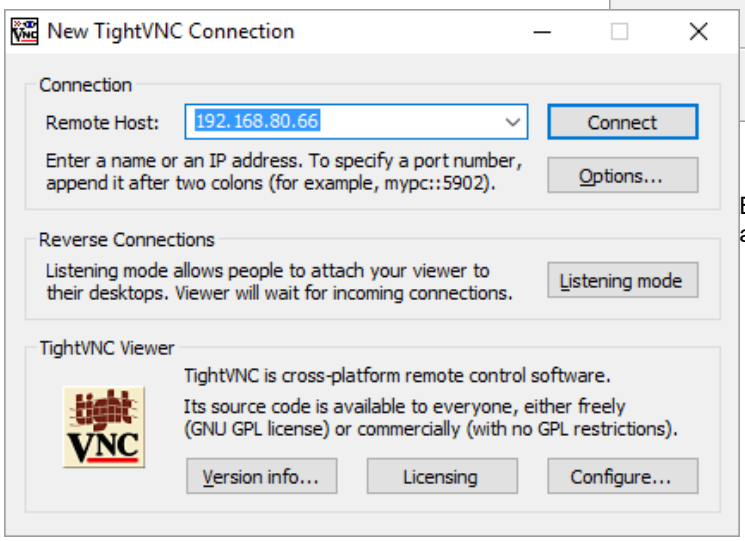
Am Beispiel des kostenlosen **Tight VNC-Viewers** hier die wichtigsten Schritte von der Installation zur Remote-Visualisierung:

Bei Installation des Programms Serverfunktionalität abwählen



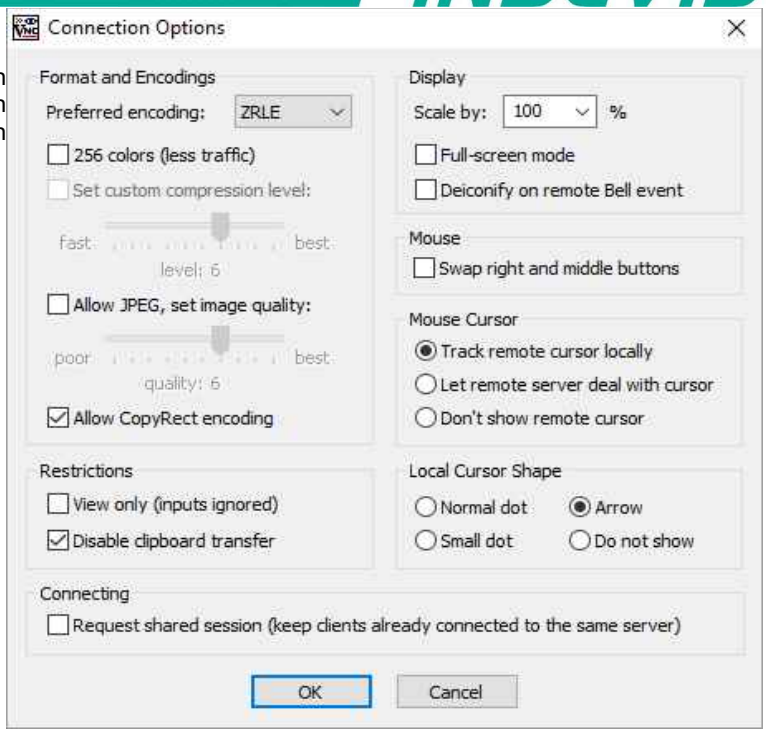
INFO

Bilder wurden bei der Installation des Tight VNC-Viewers erstellt. Dieser VNC-Viewer ist einfach zu parametrieren und damit empfehlenswert.

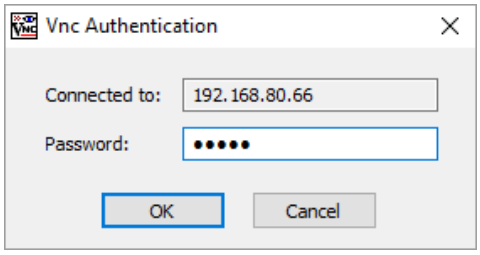


Bei der Konfiguration IP-Adresse des Remote-Gerätes angeben und auf „Options“ gehen

ZLRE-Encoding anwählen
Clipboard Transfer abwählen
Shared Session abwählen



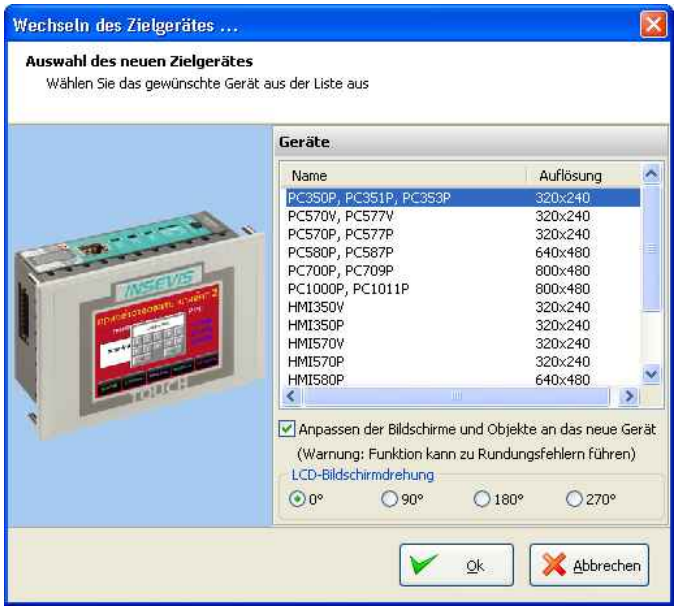
In VisuStage festgelegtes Passwort eingeben



Bildschirmauflösung oder -drehung ändern

Unter dem Menuepunkt „Projekte / Wechsel des Zielgerätes“ wird das Projekt auf ein anderes Zielsystem angepasst. Diese können sein:

- andere Panel-SPSen / Panel-HMIs von INSEVIS
- andere Ausrichtungen (90°-Drehungen) desselben Panels - **nur für CPU-P-Versionen**)



Dieser Vorgang ist erfolgt rechnerisch und bedingt manuelle Nacharbeit, um eine homogene Visualisierung zu erhalten (Schriftgrößen, Tastengrößen, etc. anpassen) und die Speichergröße des Binärfiles zu reduzieren.

Anlage und Verwaltung von Ressourcen

Ressource Sprachen

Es sind alle Sprachen verfügbar, die auf dem PC installiert sind. Diese werden jede einzelne als bmp abgespeichert.
Hinweis: Wenn die Visualisierung von einem anderen PC, wo eine projektierte Sprache fehlt, compiliert und downgeloadet wird, wird diese Sprache nicht richtig dargestellt.

- Hinzufügen/ Entfernen → durch rechte Maustaste
- Auswahl der Projektsprache → durch rechte Maustaste → rechts durch Pull down-Menü
  (z.B. um zu sehen, ob ein Text in allen Sprachen in ein Objekt passt)
- Auswahl der Referenzsprache (ist fett markiert) → durch rechte Maustaste → „als Standard festlegen“
  (auf diese Sprache beziehen sich Texte und Bilder)

Hinweis:
Text- und Bildressourcen sind mit der Sprache verknüpft, in der sie angelegt wurden. Löscht man diese Sprache, werden nach einem Abfragefenster auch diese Ressourcen mit gelöscht.

Ressource Variablen

Neben der manuellen Neuanlage gibt es mehere Möglichkeiten, Variablen aus dem S7-Programm zu importieren:

1.) Import der in den Systemdatenfile (sdf) abgelegten Variablen (SimaticManager)

- Erzeugen einer S7-Symbol-Tabelle im SimaticManager (Datei muss Endung *.sdf haben)
- Import dieser Datei (im sdf-Dateiformat) über diese Menübefehle in der VisuStage
  → Extra
  → Importieren
  → Variablen

2.) Import der Variablen direkt aus den DBs (SimaticManager)

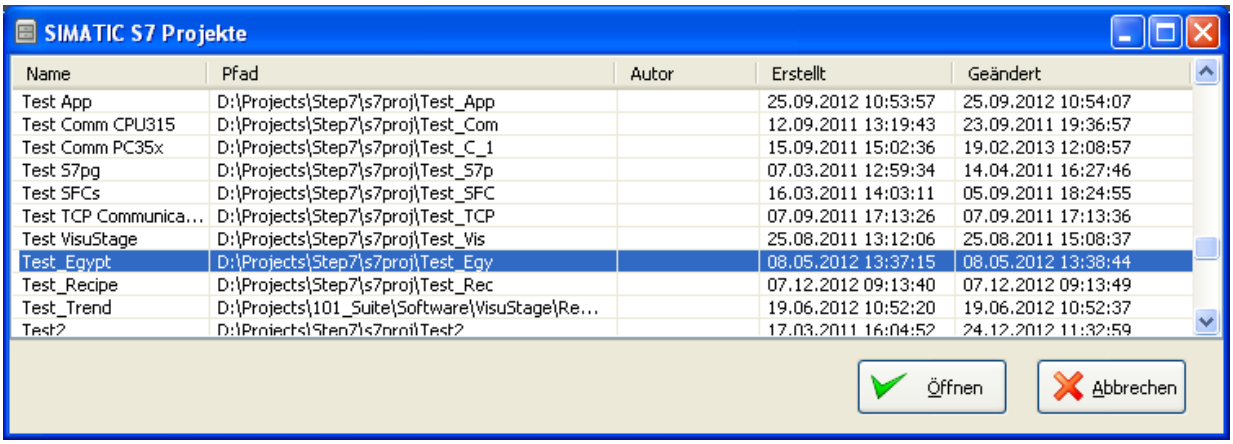
des korrespondierenden S7-Projektes (mit Symbolik)

(Der SimaticManager muss auf demselben PC installiert sein)

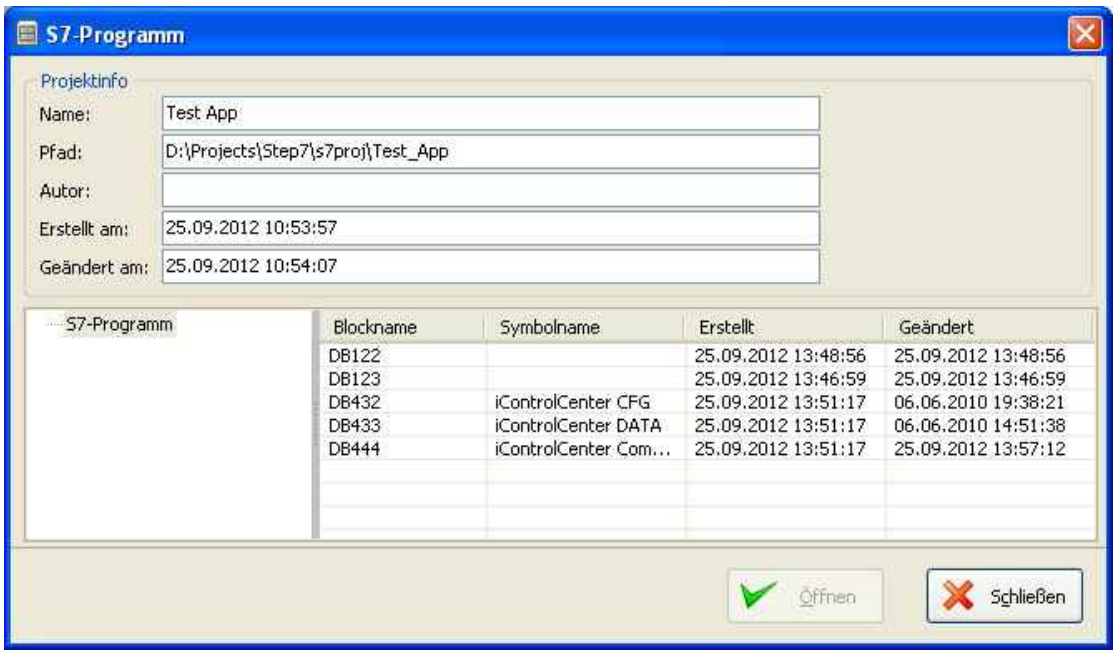
2.) Über diese Menübefehle in der VisuStage:

- über
  → Extra
  → Importieren
  → Variablen vom S7-Projekt

- Markieren des korrespondierenden Projektes
- Öffnen des des korrespondierenden Projektes



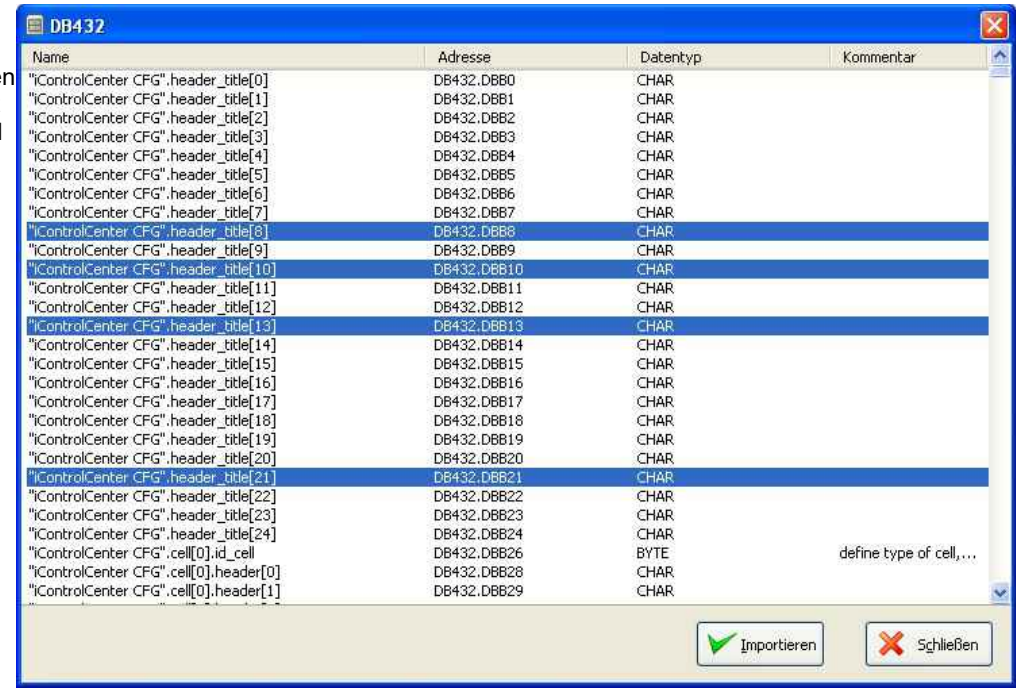
- Auswahl des gewünschten Datenbausteines aus der S7-Programmliste **(keine Instanzdatenbausteine)**



- Markieren der gewünschten Variablen (auch mehrfach mit Shift bzw. Strg) und
- Importieren

Hinweis:
→ VIDEO verfügbar

Für dieses Beispiel ist auf den INSEVIS-Service-Seiten ein Video verfügbar



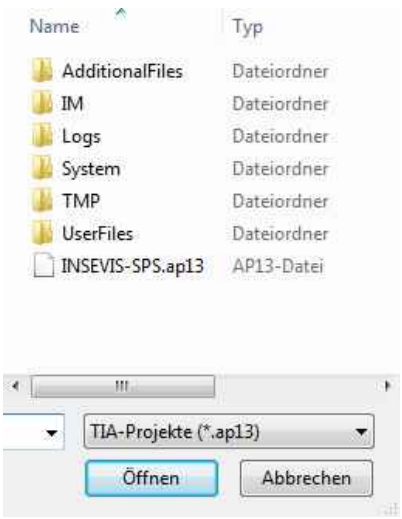
3.) Import der Variablen mit Symbolik (TIA-Portal)

Das TIA-Portal muss nicht installiert sein auf dem Rechner.

Das TIA-Projekt **MUSS** geschlossen sein bei dem Import.

Über diese Menübefehle in der VisuStage
→ Extra
→ Importieren
→ Variablen vom S7-Projekt

Erfolgt eine Auswahl des TIA-Projekts in einem neuen Fenster



AUSWAHL

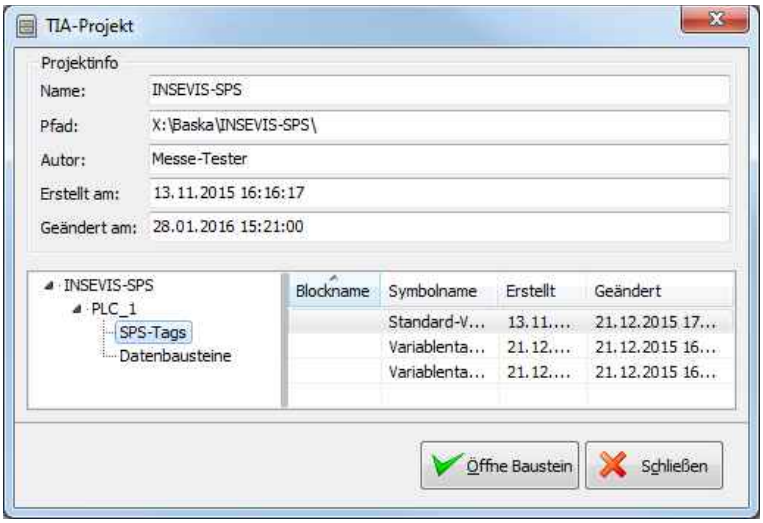
→ der **Station**
(hier INSEVIS SPS)

und dann

→ der **CPU**
(hier PLC_1)

und dann

ENTWEDER
der **Symboltabelle**
(hier SPS-Tags)



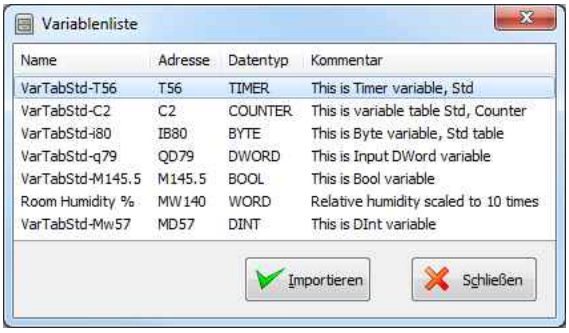
Visualisierung mit der Software „VisuStage“

(Im rechten Fenster werden
alle Variablentabellen angezeigt)

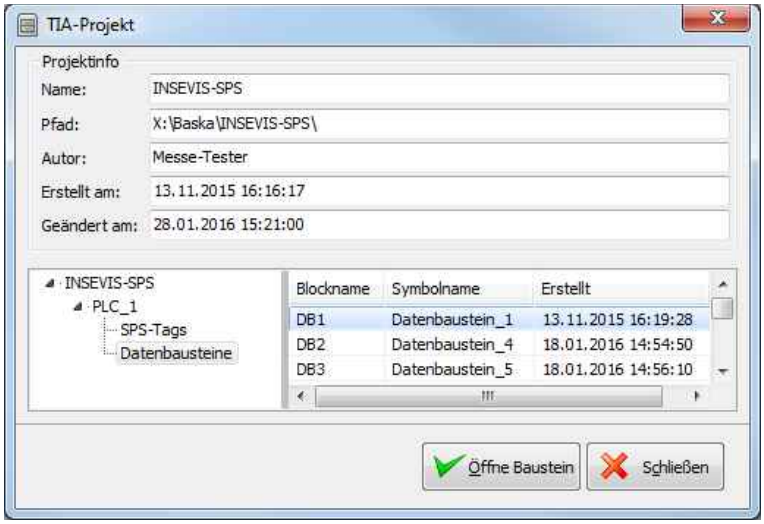
Das Öffnen einer Variablentabelle
zeigt alle vorhandenen Variablenlisten

Auswahl der Variablenlisten
durch Markieren (auch mehr-
fach mit Shift bzw. Strg)

Importieren der ausgewählten Variablenlisten



ODER
der **Datenbausteine**
(hier Data Blocks)



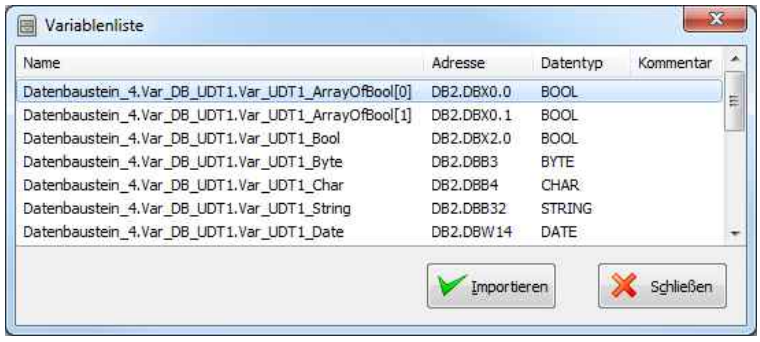
(Im rechten Fenster werden
alle Datenbausteine angezeigt)

Auswahl eines Datenbausteines
durch Markieren (nur einzeln möglich)

Öffnen des Datenbausteines

Auswahl der Variablen aus den Datenbausteinen
durch Markieren (auch mehrfach mit Shift bzw.
mit Strg)

Importieren der ausgewählten Variablen



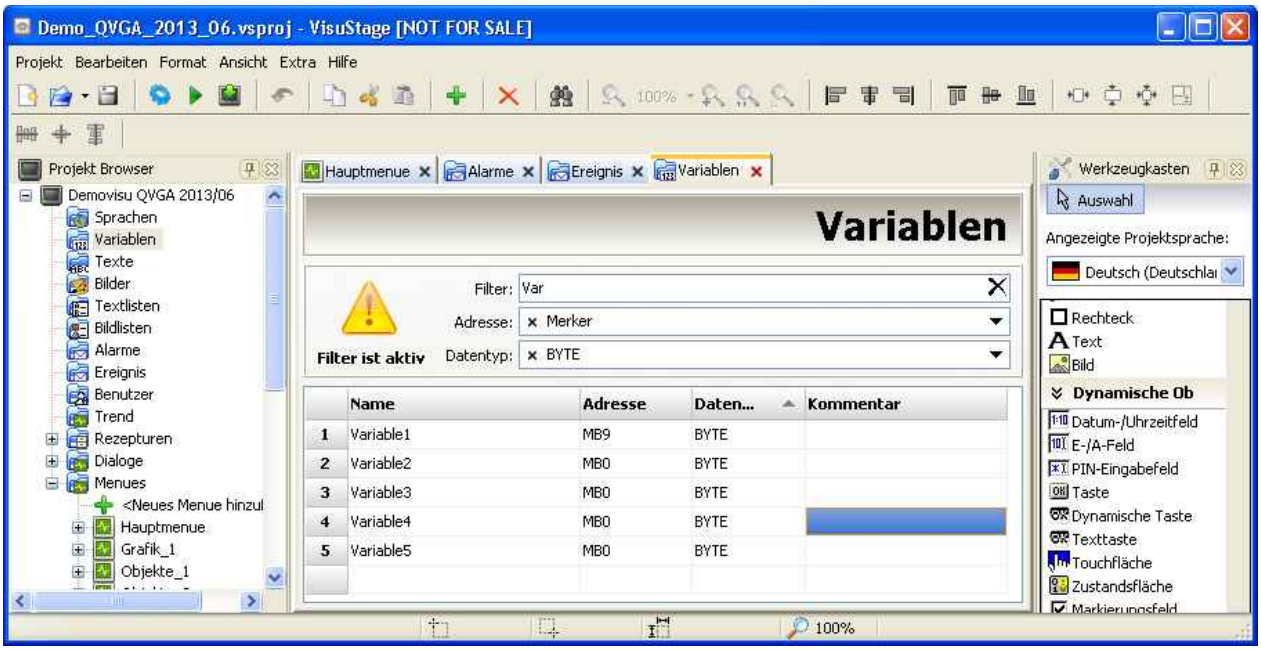
Visualisierung mit der Software „VisuStage“

Bei folgenden Fällen verlangt die VisuStage vor dem Variablenimport eine Bestätigung:
- Variable mit gleichen Namen existiert bereits, aber mit unterschiedlicher Adresse oder Datentyp
- Variable mit gleicher Adresse existiert bereits, aber mit unterschiedlichem Namen oder Datentyp
- Variable mit gleichen Namen und Adresse existiert bereits, aber mit unterschiedlichem Datentyp

Achtung:
Beim Überschreiben der Variablen durch die VisuStage werden die Verknüpfungen zu den Objekten erhalten, wenn man die Variablen manuell löscht, werden diese Verknüpfungen entfernt.

Der **Export** der Variablen erfolgt ebenfalls im
- **sdf-Dateiformat** (Systemdatenformat, nur Variablen),
  oder im
- **sym-Dateiformat** (Symboltabellenformat mit Variablen und Datenbausteinen)

Filter setzen / Variablen suchen



Variablen können nach
- Namen
- Adress- und
- Datentyp

gefiltert werden.

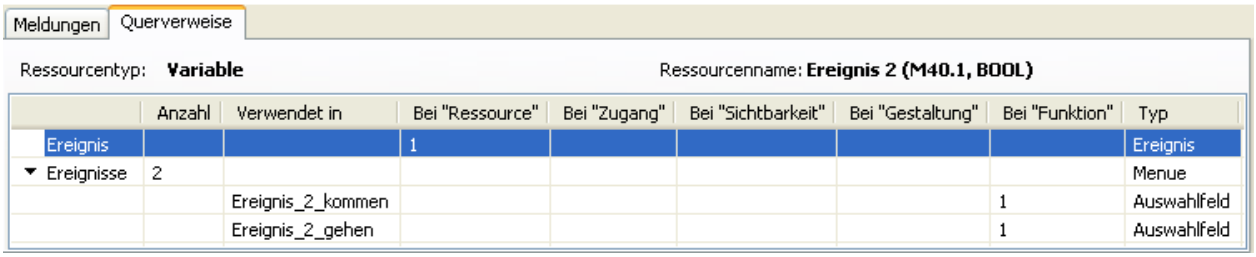
Im Beispiel werden alle Variablen angezeigt, die
- „Var“ enthalten,
- als Merker und
- Datentyp BYTE definiert sind.

Querverweise für Variablen

Um sich einen Überblick zu verschaffen, welche Variablen wo verwendet werden, können diese durch die Funktion „Querverweis“ (Variable markieren und dann rechte Maustaste) nach Verwendung in Ressourcen und in Menues aufgelistet werden.

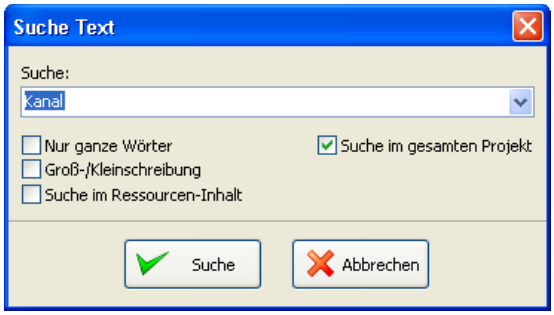
Durch Doppelklick auf eine Ressource gelangt man direkt in das entsprechende Ansichtsfenster, das Meldungsfenster selbst wird durch die Taste F6 aus- oder eingeblendet.

Im Beispiel wurde die die Verwendung der Variable M40.1 angezeigt.

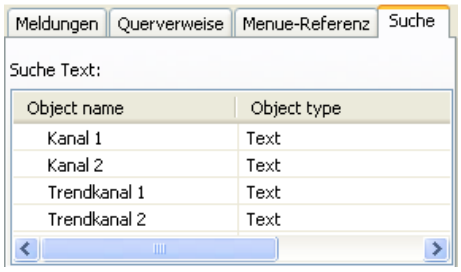


Variablen suchen nach Text

Ressourcen suchen mit der „Finden“-F <STRG+F> unter dem Reiter „Bearbeiten“, um den Überblick zu behalten.



Das Ergebnis wird in der Meldungsliste (Ein- / Ausblenden mit „F6“) angezeigt.



Bereinigen ungenutzter Variablen

Löschen ungenutzter Ressourcen mit der „Bereinigen“-Funktion unter dem Reiter „Bearbeiten“, um das Projekt „aufzuräumen“

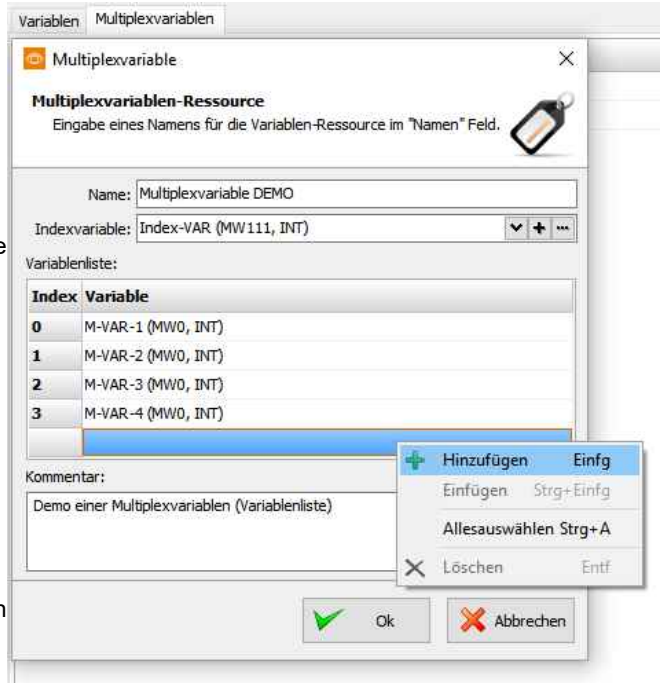
Multiplexvariablen

Ab der V 2.0.5.3 ist die Anlage von Multiplexvariablen möglich. Unter der Ressource Variablen, im Reiter Multiplexvariablen wird eine Variablenliste angelegt (siehe Beispiel rechts).

Als Indexvariable werden nur ganzzahlige Variablenformate akzeptiert, die Variablen in der Variablenliste sind frei (STRNG-Typen sind nicht mit anderen Variablentypen mischbar). Dabei können auch DBs, Merker, Eingänge und Ausgänge verwendet werden (siehe unten)

Index	Variable
0	Variable (DB1.DBW0, INT)
1	Variable1 (DB4.DBW10, INT)
2	Variable2 (DB10.DBW0, INT)

Mit Multiplexvariablen lassen sich E/A-Felder z.B. mehrfach nutzen. Je nach Wert der Indexvariable werden die jeweils zugewiesenen Multiplexvariablen zur Laufzeit angezeigt.



Ressourcen Texte und Textlisten

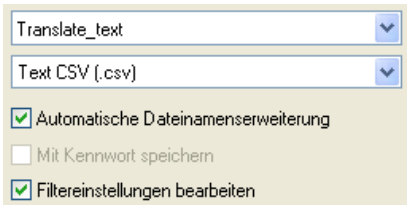
Texte

Texte werden in den vorab angelegten Projektsprachen angelegt und können über die Export-/ Import-Funktion als csv-Datei abgespeichert bzw. eingelesen werden. Das erleichtert die Übersetzung durch externe Dienstleister.

- Export über → Extra → Export → Texte
- Auswahl der zu exportierenden Sprachen
- csv-Format festlegen und abspeichern
- Öffnen/Import durch Tabellenkalkulationsprogramm (csv-Einstellungen müssen identisch sein !)
- Bearbeiten der csv-Datei (Struktur muss gleich bleiben, keine Zellen entfernen/einfügen)
- Einlesen über → Extra → Import → Texte

In den ersten Zeilen befinden sich Systemtexte. Ist der Text nur in der Referenzsprache angegeben und nicht in den andern Projektsprachen, wird er dort auch in der Referenzsprache dargestellt.

Hinweis:
Wenn der Import nicht erfolgt, hat Ihr Tabellenkalkulationsprogramm die csv-Formatierung geändert. Dieses kontrolliert man, wenn man z.B. bei OpenOffice die csv-Datei mit „Speichern unter“ abspeichert und dort die Funktion „Filtereinstellungen bearbeiten“ aktiviert. Danach werden diese Einstellung zur Änderung angeboten.

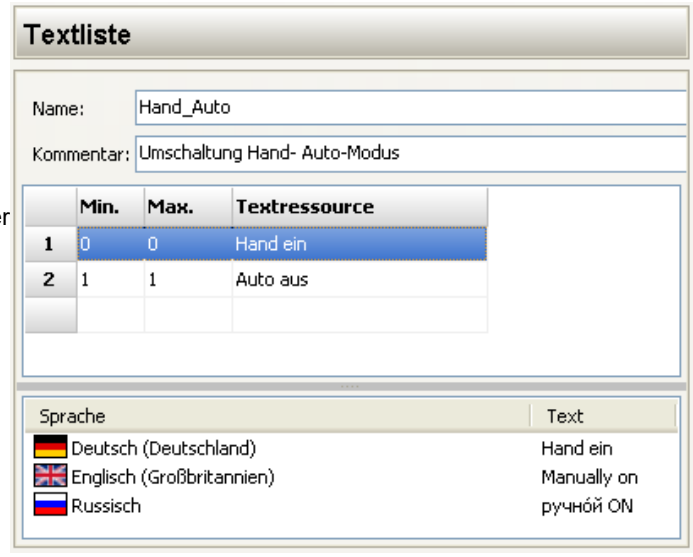


Hinweis: „Alle Textzellen zitieren“ bei manchen Windows-7 Installationen nötig.

Textlisten

Sind sinnvoll, wenn verschiedene Texte in Abhängigkeit einer Variable angezeigt werden sollen, z.B. bei der Benutzerverwaltung)

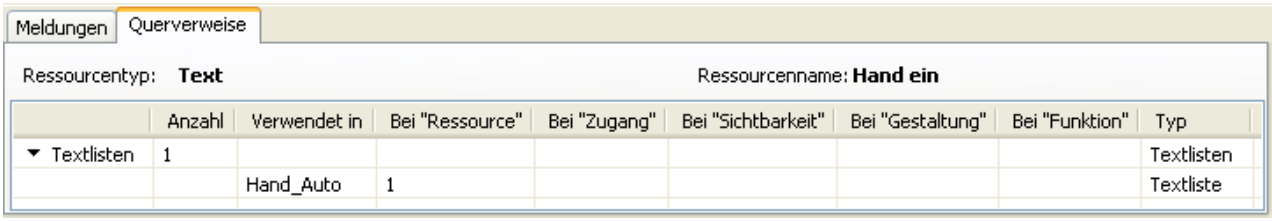
- Textlisten werden manuell angelegt:
  rechte Maustaste in Feld „Name“ und **neuen Namen** der Textliste, ggf. Kommentar eingeben.
- Bereichsgrenzen (**Min**. / **Max**.) eingeben und eine **Textressource** (die entweder vorher angelegt wurde, oder durch rechten Mausklick im Feld Textressource neu angelegt wird), in dem diese Textressource angezeigt werden soll.
- Ab V2.0.3.5 werden die in den jeweiligen Textlisten verwendeten Textressourcen zur besseren Orientierung unterhalb der Textliste angezeigt.



ACHTUNG:
Wenn ein anderer Wert als die in den Zeilen 1...n angegeben Werte anliegt, wird IMMER die Textressource angezeigt, die an ERSTER STELLE (Zeile 1) zugewiesen wurde.

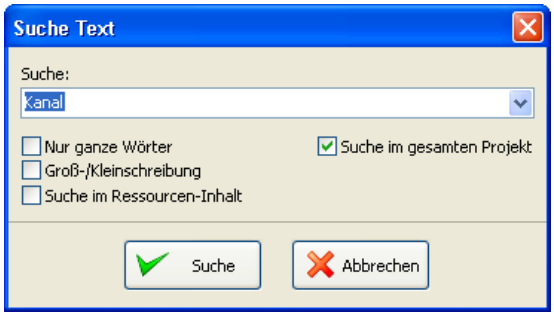
Querverweise für Texte und Textlisten

Um sich einen Überblick zu verschaffen, welche Texte/Textlisten wo verwendet werden, können diese durch die Funktion „Querverweis“ (Text/Textliste markieren und dann rechte Maustaste) nach Verwendung in Ressourcen und in Menues aufgelistet werden. Durch Doppelklick auf einen Text/Textliste gelangt man direkt in das entsprechende Ansichtsfenster, das Meldungsfenster selbst wird durch die Taste F6 aus- oder eingeblendet.



Texte/Textlisten suchen nach Text

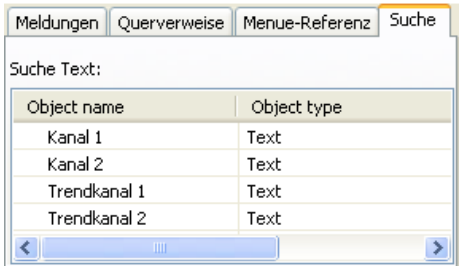
Ressourcen suchen mit der „Finden“-F <STRG+F> unter dem Reiter „Bearbeiten“, um den Überblick zu behalten.



Bereinigen ungenutzter Texte/Textlisten

Löschen ungenutzter Ressourcen mit der „Bereinigen“-Funktion unter dem Reiter „Bearbeiten“, um das Projekt „aufzuräumen“

Das Ergebnis wird in der Meldungsliste (Ein- / Aus-blenden mit „F6“) angezeigt.



Ressourcen Bilder / Bildlisten

Bilder

INSEVIS liefert einige Bilder, Symbole mit der VisuStage mit. Das Erstellen eigener Bilder ist sehr einfach mit fast jedem Bildbearbeitungsprogramm möglich. Wir haben das kostenlose Programm „PhotoFiltre“ verwendet.

- empfohlenes Bilddateiformat: png mit transparentem Alpha-Kanal. Bei den meisten Bildprogrammen kann eine Farbe durch Transparenz ersetzt werden. Das bearbeite Bild noch als PNG abspeichern (Bei „PhotoFiltre“ die Option „Interlaced / Automatic“ anwählen), fertig.

- in der Projektsprache angelegte Bilder werden in allen anderen Sprachen genutzt und dort → halbtransparent angezeigt.

- Es ist möglich, für dieses Bild in anderen Sprachen andere Bilder zu verwenden. Durch rechte Maustaste ein neues Bild zuweisen. Dieses wird dann wieder in vollen Farben angezeigt.

Bildlisten

Für variablenabhängige Bildwechsel oder Farbumschläge, ist die Bildliste das geeignete Werkzeug.

Bildlisten werden manuell angelegt:

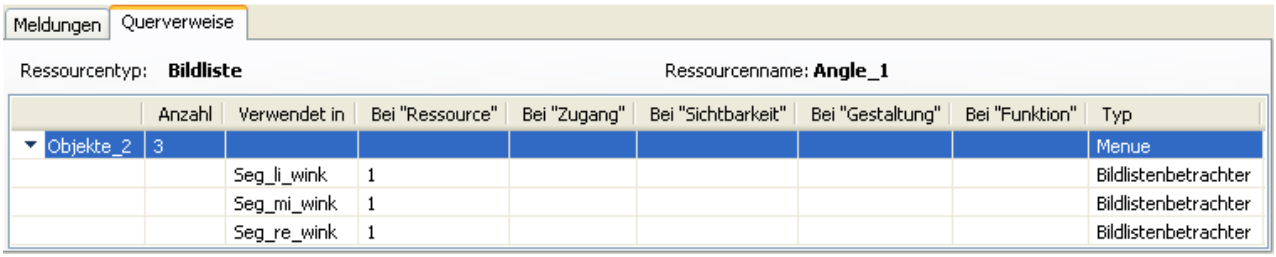
- mit rechter Maustaste in Feld „Name“ klicken und neuen **Namen** der Bildliste und ggf. Kommentar eingeben
- Bereichsgrenzen (**Min**. / **Max**.) eingeben und dafür
- eine **Bildressource** (das passende Bild) angeben
  Dieses muss entweder vorher angelegt werden, oder durch rechten Mausklick im Feld Bildressource neu angelegt werden),
- Ab V2.0.3.5 werden die in den jeweiligen Bildlisten verwendeten Bildressourcen zur besseren Orientierung unterhalb der Bildliste angezeigt.

ACHTUNG:
Wenn ein anderer Wert als die in den Zeilen 1...n angegeben Werte anliegt, wird IMMER die Bildressource angezeigt, die an ERSTER STELLE (Zeile 1) zugewiesen wurde.

Querverweise für Bilder und Bildlisten

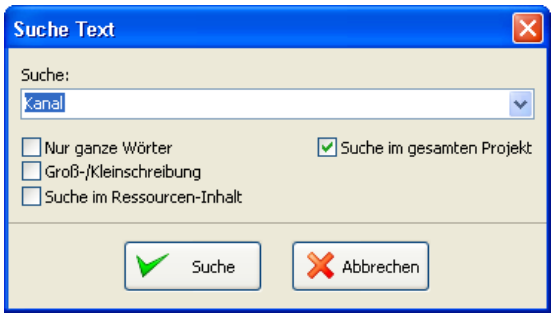
Um sich einen Überblick zu verschaffen, welche Bilder/Bildlisten wo verwendet werden, können diese durch die Funktion „Querverweis“ (Bild/Bildliste markieren und dann rechte Maustaste) nach Verwendung in Ressourcen und in Menues aufgelistet werden.

Durch Doppelklick auf ein Bild/Bildliste gelangt man direkt in das entsprechende Ansichtsfenster, das Meldungsfenster selbst wird durch die Taste F6 aus- oder eingeblendet.



Bilder/Bildlisten suchen nach Text

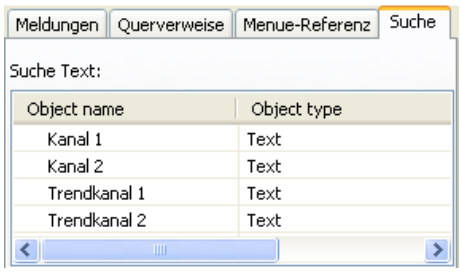
Ressourcen suchen mit der „Finden“-F <STRG+F> unter dem Reiter „Bearbeiten“, um den Überblick zu behalten.



Bereinigen ungenutzter Bilder/Bildlisten

Löschen ungenutzter Ressourcen mit der „Bereinigen“-Funktion unter dem Reiter „Bearbeiten“, um das Projekt „aufzuräumen“

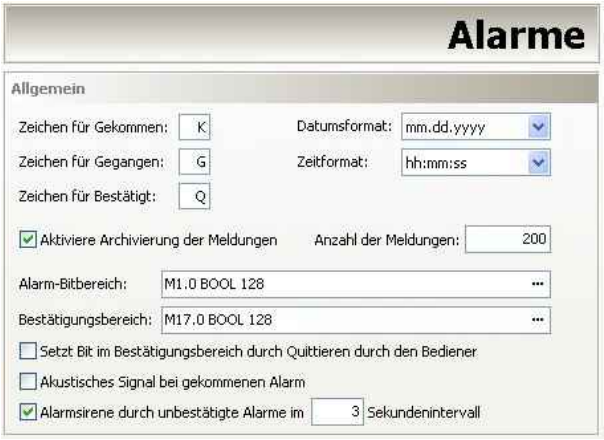
Das Ergebnis wird in der Meldungsliste (Ein- / Aus-blenden mit „F6“) angezeigt.



Ressourcen Meldungen (Alarme / Ereignisse)

Alarme/ Ereignisse

Es können bei der CPU Typ V 128 Alarme und bei CPU Typ P 1024 Alarme und immer 128 Ereignisse verarbeitet und bei beiden CPUs jeweils bis zu 500 Meldungen (Alarme/ Ereignisse) archiviert werden.



- Meldungen können über die Export-/ Import-Funktion als csv- Datei abgespeichert bzw. eingelesen werden.
- Ein manuelles Anlegen erfolgt in den vorab angelegten Projektsprachen.
- Die Archivierung mit Häkchen aktivieren und Micro SD-Card stecken (Archivierung erfolgt in binären Format, die Konvertierung in das csv-Format erfolgt mit der RemoteStage, auch in Batchfiles automatisierbar)
- Ein akustisches Signal bedeutet einen 1 Sekunde langen einmaligen Ton des internen Buzzers
- Eine Alarmsirene ist ein in einstellbaren Intervallen von 1...600Sekunden wiederkehrender 1 Sekunden langer Ton des internen Buzzers

- **Alarm- / Ereignis-Bitbereich** (Bereich der Triggerbitadressen, die den Alarm/das Ereignis auslösen) ist bei CPU V 128 Bit / bei CPU Typ P 1024 Bit groß und **muss zusammenhängend sein**.
  Jedes Bit gehört zu einer Meldung. (Bit 0 zu Alarm/Ereignis 1, Bit 1 zu Alarm/ Ereignis 2, etc...)
  Der Meldungsstatus kann im S7-Programm durch ein Bit aus diesem Bereich gesteuert werden.
  Mit Bit=1 erscheint diese Meldung als „gekommen“.
  Mit Bit=0 wird diese Meldung als „gegangen“ interpretiert.

- **Bestätigungsbereich (nur für Alarme)** (Bereich der Quittierungsbitadressen, die bei Quittierung des jeweiligen Alarmes gesetzt werden) ist auch CPU V 128 Bit / Typ P 1024 Bit groß und **muss zusammenhängend sein**.
  Jedes Bit gehört zu einem Alarm. (Bit 0 zu Alarm 1, Bit 1 zu Alarm 2, etc...)
  Der Alarmstatus kann im S7-Programm durch ein Bit aus diesem Bereich gesteuert werden
  Mit Bit=1 erscheint eine anstehende Meldung als „quittiert“.
  Mit Bit=0 wird diese Meldung als nicht quittiert interpretiert.

Beide Bereiche dürfen sich nicht überlappen (Fehlermeldung erfolgt beim Kompillieren).

Wichtig:

- Der Baustein für Trigger- und Quittierungsbitbereiche **MUSS** auch in der SPS **vorhanden** und mindestens **so groß** (128/1024Bit) **sein**, sonst erscheinen die Meldungen / Quittierungen nicht!
- Wird das Bestätigungsbit **aktiviert**, muss es in der SPS verarbeitet (nach gewisser Zeit **rückgesetzt**) werden, sonst erscheint der nächste gleiche Alarm bereits als quittiert.
- Wird es **nicht aktiviert** und man erzeugt und betreibt eine Visualisierung mit der RemoteStage wird bei dem Quittieren auf der RemoteStage-Visualisierung kein Bit in der SPS gesetzt und bei einem anderen Panel-HMI verbleibt diesen Alarm als **nicht quittiert**.

Anzeige der Alarm-/ Ereignis-Meldungen

Alarm-Meldungen

	Deutsch ...	Englisch ...	Russis...	Gruppe	Triggerbitadresse	Quittierungsbitadresse	Variable
1	Alarmtext von...	alert text of al...	тревога 1	1	M1.0	M17.0	
2	Alarmtext von...	alert text of a...	тревога 2	2	M1.1	M17.1	
3					M1.2	M17.2	

- Alarme gleicher **Gruppe** werden immer gemeinsam bestätigt.
- Alarme der **Gruppe 0** werden **automatisch** bestätigt.
- **Triggerbit- und Quittierungsbitadressen** werden nur zur Information angezeigt und sind nicht änderbar.
- Die Angabe einer **Variable** ermöglicht die Anzeige eines zusätzlichen dynamischen Fehlercodes (z.B. eines berechneten Wertes für die quantitative Anzeige eines Zustandes in der Alarmmeldung) als Dezimalzahl (optional).
  → Wenn diese Variable hier eingetragen wird, erscheint sie automatisch in dem Meldetext
  Das ist keine Triggervariable für die Meldung!
- In der Meldearchivanzeige können die Meldungen auf-/ absteigend sortiert werden

Visualisierung mit der Software „VisuStage“

Ressource Benutzerverwaltung

Es können bis zu 9 Benutzer mit Passwörtern (bis zu 8-stelligen PINs) verwaltet werden. Benutzer mit Level 9 haben alle Zugangsrechte, Benutzer mit Level 1 die wenigsten.

In der oberen Maske werden eingestellt:

- **Zielbildschirme** für den Login (und unten für Logout) oder
- gleich der **PIN-Eingabedialog** ausgewählt

- automatische Logoutzeiten (in Minuten) zugewiesen. (Bei Logout-Zeit 0 wird **nicht automatisch** ausgeloggt.) und
- eine Variable eingetragen, die der SPS mitteilt, welcher Benutzerlevel gerade aktiv ist
- Die Abfrage nach dem Benutzerlevel vor der PIN-Eingabe verlangt

In der unteren Maske werden für die einzelnen Levels (1...9)

- die Benutzernamen angelegt und
- die zugehörigen PIN-Codes zugeordnet

Hinweis:

- Der Zielbildschirm nach erfolgreichem Login wird bei dem dynamischen Objekt „PIN-Eingabefeld“ festgelegt.
- Mit dem **PIN-Änderungsdialog** kann eine PIN vom Bediener geändert werden (Sein Level bleibt gleich.), wird diese vergessen, kann der Urzustand durch Neuaufspielen der Visualisierung wiederhergestellt werden.
- Für **Benutzerlevelabhängige Bildschirmwechsel** steht ab Version 2.0.3.6 eine Funktion zur Verfügung, die z.B. einer Taste zugeordnet werden kann.

Visualisierung mit der Software „VisuStage“

Ressource Partner-SPS (nur für HMI)

Das Panel/HMI kommuniziert über Ethernet (S7-Kommunikation Put/Get) mit der Partner-SPS.

Die Verbindung wird

a) im BIOs des Panel-HMIs
direkt eingetragen

oder

b) in VisuStage und ConfigStage
wie folgt konfiguriert:

Für INSEVIS-Panels und INSEVIS-SPSen

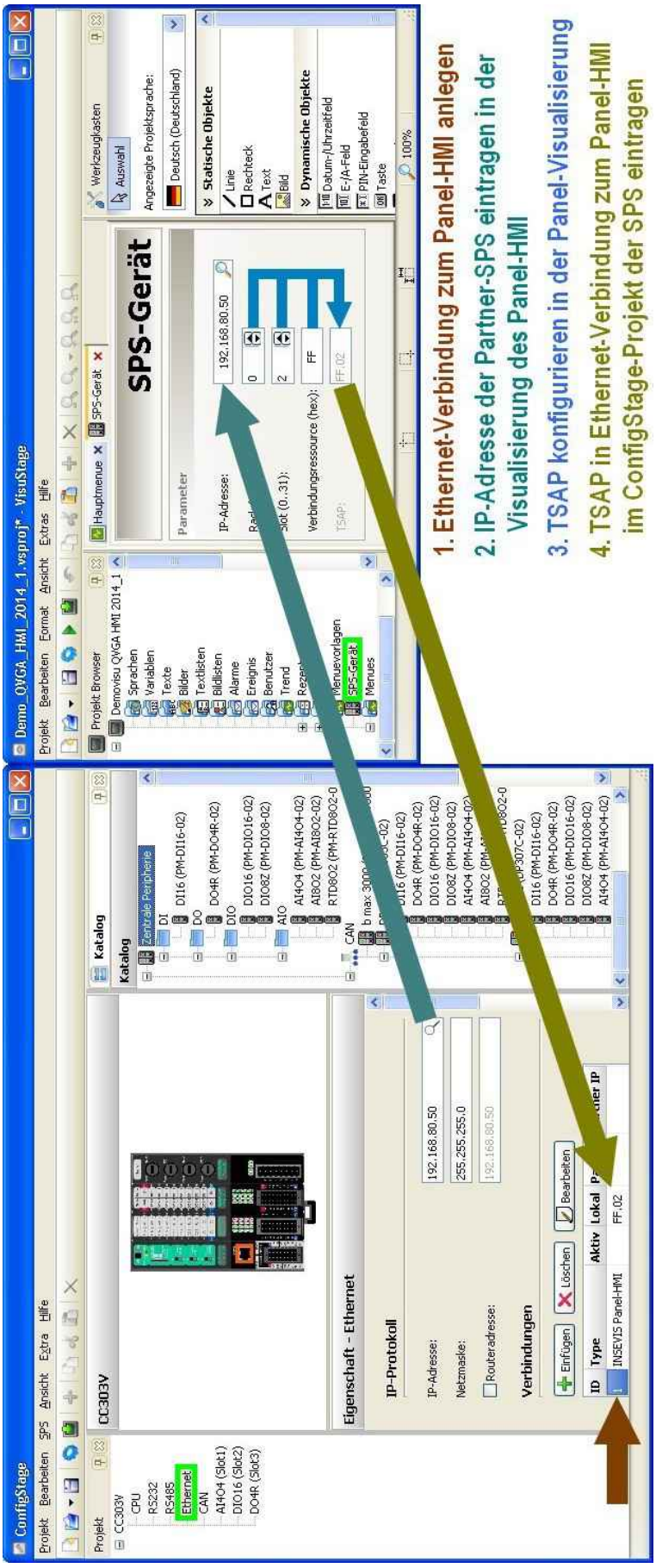
- Ethernet-Verbindung zu Panel-HMI in ConfigStage der SPS anlegen und deren ID-Nr. merken,
- IP-Adresse der Partner-SPS in VisuStage eintragen,
- TSAP in VisuStage konfigurieren (gebildet aus Rack, Slot, ID-Nr. der Verbindung), hier 0, 2, FF
- in ConfigStage der SPS den TSAP (hier FF.02) in der Ethernet-Verbindung zu Panel-HMI eintragen

Für INSEVIS-Panels und Siemens-CPUs

- IP-Adresse der Partner-SPS in VisuStage eintragen,
- TSAP in VisuStage konfigurieren (gebildet aus **Rack, Slot, ID-Nr**. der Verbindungsressource),

  für Siemens-CPUs 300/400
  ist das **0, 2, 02**,

  für Siemens-CPUs 1200
  ist das **0, 1, 02**



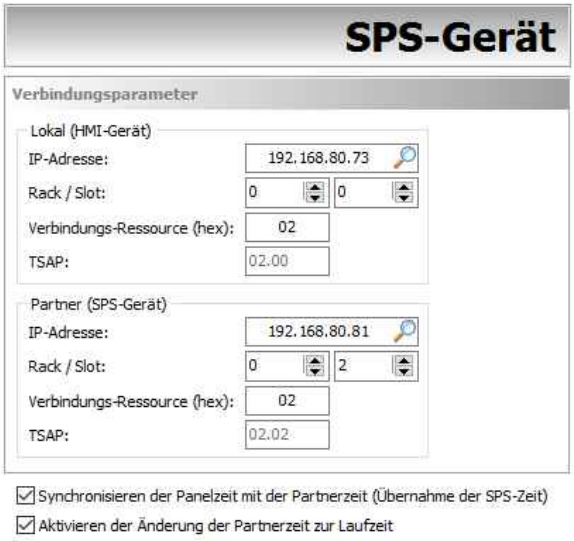
Partner-Gerät synchronisieren (nur für HMI)

Es wird empfohlen, mit der Zusweisung von Verbindungseinstellungen auch eine Zeit-Synchronisation zwischen Panel und SPS vorzunehmen

Erste Check-Box
Das HMI wird mit der SPS-Zeit synchronisiert

Zweite Check-Box
Die SPS-Zeit kann am Panel zur Laufzeit geändert werden

ACHTUNG:
Ohne Synchronisation können die Uhrzeiten beider Geräte auseinanderlaufen und so Abläufe zu unvorhergesehenen Zeiten starten.

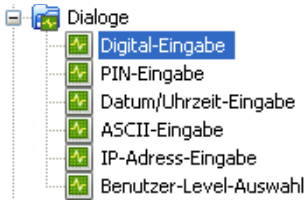


Ressource Dialoge

Die Größe und das Design der automatisch eingeblendeten virtuellen Tastatur bei

- E/A-Feldern,
- Loginfeldern,
- Datum-/Uhrzeitfeldern
- Stringeingabefeldern
- IP-Adresseingabefeldern
- Benutzer-Level-Auswahlfeldern

kann unter der Ressource „Dialoge“ angepasst werden.

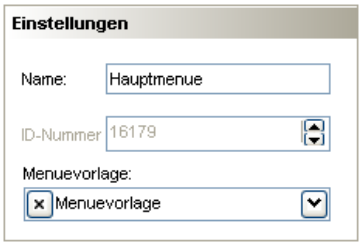


Die Größe des gesamten Objektes bestimmt am Ende die automatisch errechnete Tastengröße. (Die LCD-Auflösung nicht überschreiten!)

Die restlichen Einstellungen sind selbsterklärend.

Ressource Menuevorlagen

Menuevorlagen werden wie ein eigener Bildschirm angelegt und können einmal in jedem Menue als Vorlage verwendet werden.
Dazu wird dem jeweiligen Menue die Menuevorlage aktiviert und ausgewählt.
Jede in der Menuevorlage durchgeführte Änderung wird automatisch in alle Menues übertragen, bei denen die Menuevorlage aktiviert ist.



Ressource Trends

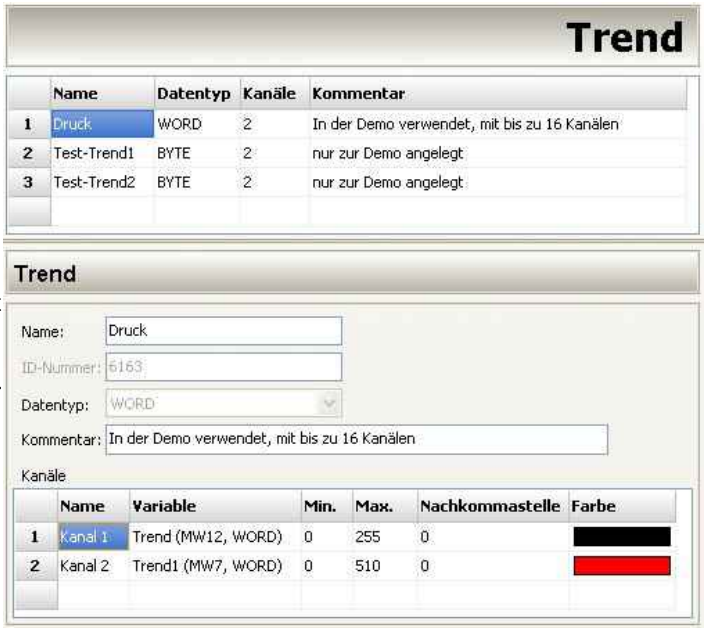
Trends stellen einen Wert in Abhängigkeit der Zeit dar. Es stehen Trends mit je 16 Kanälen zur Verfügung. In jedem Kanal ist die Zeitbasis gleich (kleinste Auflösung = 1 Sekunde)

Die Trends können zur besseren Identifizierung mit einem Namen versehen werden, der bei der Trendanzeige und Abspeicherung mit der RemoteStage als Dateiname verwendet wird.

Pro Trend wird immer NUR EINE Skala angezeigt, (die anderen Trends dieses Kanals werden überhöht dargestellt), daher ist die Zuweisung verschiedener Farben eine gute Möglichkeit, die Übersicht zu wahren. Die Größe der Skala wird aus den Min/Max-Werten jedes Kanals generiert.

Die Auswahl des Trends erfolgt durch selbst zu projektierende *Auswahl- und Markierungsfelder*.

Die Anzeige der richtigen Skale (Y-Achse) erfolgt durch Berühren des jeweiligen Trendnamens (über der Y-Achse) auf dem Bildschirm



Die Konfiguration des Trends erfolgt über die Funktion „Trend starten“, die einer Taste oder einer Touchfläche zugewiesen wird.

Wenn „Archivierung der Samples“ aktiviert wurde, kann man die Anzahl der zu aktivieren Messwerte angeben, bevor nach FIFO- Prinzip der Ringpuffer überschrieben wird.

Es wird ausschließlich auf der Micro-SD-Card archiviert. Jeder Trend wird als EINE Binärdatei abgespeichert.

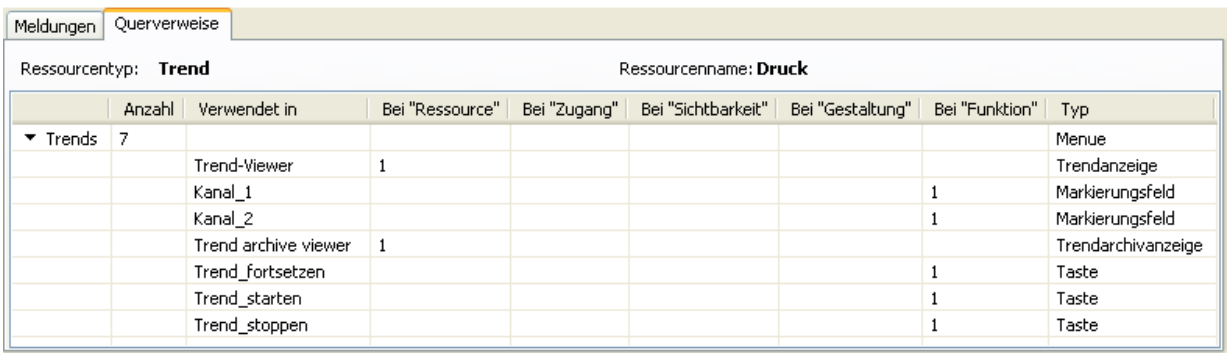
Diese Datei kann mit der RemoteStage in den PC eingelesen, angezeigt und in das csv-Format konvertiert.

Die Trenddaten können
- zyklisch (in Sekunden) oder
- variablengesteuert

(direkt aus dem S7-Programm) archiviert werden.

Querverweise für Trends

Um sich einen Überblick zu verschaffen, in welchen Menues Trends gestartet, gestoppt und fortgesetzt oder angezeigt werden, können diese durch die Funktion „Querverweis“ (Trend markieren und dann rechte Maustaste) nach Verwendung in Ressourcen und in Menues aufgelistet werden. Durch Doppelklick auf eine Verwendung gelangt man direkt in das entsprechende Ansichtsfenster, das Meldungsfenster selbst wird durch die Taste F6 aus- oder eingeblendet.



Starten der Trenderfassung mit dem SFC 202 "TRENDSRT"

Mit dem SFC202 "TRENDSRT" wird die in der VisuStage konfigurierte Trendaufzeichung gestartet. Das Intervall und die Anzahl der Aufzeichnungen werden dem Trend zugewiesen.

Nach dem Start erfasst das Betriebssystem die konfigurierten Operandenwerte in dem vorgegebenen Intervall und speichert diese auf der Micro-SD-Karte, wenn der „ARCHIVE“- Parameter auf den Wert „TRUE“ gesetzt wurde.
Werden die „CYCLIC“-Parameter auf den Wert „FALSE“ gesetzt, wird die Erfassung nur einmalig ausgeführt und die Trenderfassung wechselt in den „STOP“-Zustand.

Trenddaten werden in die Datei „**trend_xxx.bin**“ geschrieben (xxx ist die Trend-ID-Nummer). Dieses Archiv-File basiert auf einen **Ringpuffermechanismus** mit „COUNT“-Aufzeichnungen.

Die Daten können mit der RemoteStage von der SD-Karte entweder direkt oder über Ethernet eingelesen werden und als csv abgespeichert werden. Diese Funktion ist auch batchprozessfähig.

Parameter	Deklaration	Datentyp	Beschreibung
LADDR	INPUT	WORD	logische Adresse = 0 (für alle Panel-SPS) logische Adresse = 1...16 (Verbindungs-ID-Nummer zum Panel-HMI)
TREND	INPUT	INT	Trend ID-Nummer, (von VisuStage vergeben, bei Ressourcen)
INTERVAL	INPUT	WORD	Aufzeichnungsintervall in Sekunden (1..65535 Sekunden sind zulässig).
COUNT	INPUT	WORD	Anzahl von Aufzeichnungen zum Aufzeichnen/Archivieren (1..65535 sind zulässig).
CYCLIC	INPUT	BOOL	TRUE: Die COUNT-Aufzeichnungen werden zyklisch aufgezeichnet. FALSE: Die COUNT-Aufzeichnungen werden einmalig aufgezeichnet, dann wird die Aufzeichnug gestoppt.
ARCHIVE	INPUT	BOOL	TRUE: Die aufgezeichneten Werte werden in der Micro-SD-Karte gespeichert FALSE: Kein Archiv
RET_VAL	OUTPUT	WORD	Fehlercodes W#16#0000 – Kein Fehler W#16#8002 – VisuStage läuft nicht / nicht konfiguriert W#16#8003 – Ungültige „TREND“-ID-Nummer W#16#8004 – Ungültiger Zustand / bereits gestartet W#16#8005 – Ungültiger „INTERVAL“-Wert W#16#8006 – Ungültiger „COUNT“-Wert W#16#8xyy – Allgemeine Fehlercodes (kompatibel zu STEP®7) nur für Panel-HMI relevant: W#16#7000 – Job wird bearbeitet W#16#7001 – Ungültiger Verbindungszustand, z.B. nicht verunden W#16#7002 – Auftrag nicht akzepiert, weil anderer Auftrag in Arbeit ist W#16#7003 – Verbindung belegt (kurzzeitig gesperrt) W#16#8001 – Ungültige Verbindungs-ID-Nummer oder Verbindung nicht konfiguriert
BUSY	OUTPUT	BOOL	TRUE: Job wird bearbeitet

Wenn der Trend einmal gestartet wurde und die SPS geht in Power OFF, läuft der Trend nach Power ON automatisch weiter.

→ Hier geht es zur Projektierung mit den Objekten *Trendanzeige und Trendarchivanzeige*.

S7-Programmbeispiel für die Verwendung des SFC202

Ziel: Trend mit der ID-Nummer 6163 mit folgender Konfiguration starten:

- Aufzeichnungsintvervall 1 Sekunde,
- zyklische Aufzeichnungen,
- Archivierung auf Micro-SD-Karte,
- Ringpuffergröße 65.535 Aufzeichnungen

```
      UN    M   1500.2                 // Wenn keine Anforderung gesetzt,
      BEB                              // dann Funktion beenden

      CALL  SFC  202
       LADDR   :=W#16#0                // VisuStage Verbindungs-ID-Nummer
                                       // (Bei Panel-SPSen immer W#16#0 verwenden!)
       TREND   :=6163                  // ID-Nummer des in der VisuStage konfigurierten Trends
       INTERVAL:=W#16#1                // Aller 1s Werte aufzeichnen
       COUNT   :=W#16#FFFF             // Max. Anzahl von Aufzeichnungen 65535 (dez) => FFFF (hex)
       CYCLIC  :=TRUE                  // Zyklische Aufzeichnung als Ringpuffer.
                                       // （Puffergröße wird in COUNT-Parameter vorgegeben)
       ARCHIVE :=TRUE                  // Aktiviert das Archivieren auf Micro-SD-Karte
       RET_VAL :=MW1000                // Resultat / Rückgabewert
       BUSY    :=M1700.2               // TRUE = Anforderung in Abarbeitung

      U     M   1700.2                 // Trend starten ist in Abarbeitung,
      BEB                              // dann Funktion beenden
      R     M   1500.2                 // Setze Anforderungssignal zurück

      L     MW  1000                   // Resultat / Rückgabewert
      L     0                          // Kein Fehler
      ==I                              // Wenn erfolgreich erledigt, oder kein Fehler auftrat,
      BEB                              // dann Funktion beenden

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

      BEA                              // Ende der Funktion
```

Stoppen der Trenderfassung mit dem SFC 203 "TRENDSTP"

Die Trendaufzeichnung kann durch den Aufruf des SFC 204 “TRENDSTP“ gestoppt werden.

Parameter	Deklaration	Datentyp	Beschreibung
LADDR	INPUT	WORD	logische Adresse = 0 (für alle Panel-SPS) logische Adresse = 1...16 (Verbindungs-ID-Nummer zum Panel-HMI)
TREND	INPUT	INT	Trend ID-Nummer, (von VisuStage vergeben, bei Ressourcen)
RET_VAL	OUTPUT	WORD	Fehlercodes W#16#0000 – Kein Fehler W#16#8002 – VisuStage läuft nicht / nicht konfiguriert W#16#8003 – Ungültige „TREND“-ID-Nummer W#16#8xyy – Allgemeine Fehlercodes (kompatibel zu STEP®7) nur für Panel-HMI relevant: W#16#7000 – Job wird bearbeitet W#16#7001 – Ungültiger Verbindungszustand, z.B. nicht verunden W#16#7002 – Auftrag nicht akzepiert, weil anderer Auftrag in Arbeit ist W#16#7003 – Verbindung belegt (kurzzeitig gesperrt) W#16#8001 – Ungültige Verbindungs-ID-Nummer oder Verbindung nicht konfiguriert
BUSY	OUTPUT	BOOL	Busy flag „TRUE“: Job wird bearbeitet

S7-Programmbeispiel für die Verwendung des SFC203

Ziel: Trend mit der ID-Nummer 6163 stoppen

```
      UN    M   1500.3          //  Wenn keine Anforderung gesetzt,
      BEB                       //  dann Funktion beenden

      CALL  SFC  203
       LADDR  :=W#16#0          //  VisuStage Verbindungs-ID-Nummer
                                //  (Bei Panel-SPSen immer W#16#0 verwenden!)
       TREND  :=6163            //  ID-Nummer des in der VisuStage konfigurierten Trends
       RET_VAL:=MW1000          //  Resultat / Rückgabewert
       BUSY   :=M1700.3         //  TRUE = Anforderung in Abarbeitung

      U     M   1700.3          //  Trend stoppen ist in Abarbeitung,
      BEB                       //  dann Funktion beenden
      R     M   1500.3          //  Setze Anforderungssignal zurück

     L     MW  1000             //  Resultat / Rückgabewert
     L     0                    //  Kein Fehler
     ==I                        //  Wenn erfolgreich erledigt, oder kein Fehler auftrat,
     BEB                        //  dann Funktion beenden

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

     BEA                        //  Ende der Funktion
```

Fortsetzen der Trenderfassung mit dem SFC 204 "TRENDCNT"

Die vorher gestoppte Trendaufzeichnung kann durch den Aufruf des SFC 204 “TRENDCNT“ fortgesetzt werden. Das Aufzeichnungsintervall muss neu zugewiesen werden.

Die Anzahl von Aufzeichnungen und der Aufzeichnungsmodus (zyklisch überschreibend, archivierend) werden so beibehalten,wie zuvor zugewiesen.

Parameter	Deklaration	Datentyp	Beschreibung
LADDR	INPUT	WORD	logische Adresse = 0 (für alle Panel-SPS) logische Adresse = 1...16 (Verbindungs-ID-Nummer zum Panel-HMI)
TREND	INPUT	INT	Trend ID-Nummer (von VisuStage vergeben, bei Ressourcen)
INTERVAL	INPUT	WORD	Aufzeichnungsintervall in Sekunden (1..65535 Sekunden sind zulässig).
RET_VAL	OUTPUT	WORD	Fehlercodes W#16#0000 – Kein Fehler W#16#8002 – VisuStage läuft nicht / nicht konfiguriert W#16#8003 – Ungültige „TREND“-ID-Nummer W#16#8004 – Trend wurde nicht gestartet oder Aufzeichnungszähler erreicht „0“ W#16#8xyy – Allgemeine Fehlercodes (kompatibel zu STEP®7) nur für Panel-HMI relevant: W#16#7000 – Job wird bearbeitet W#16#7001 – Ungültiger Verbindungszustand, z.B. nicht verunden W#16#7002 – Auftrag nicht akzepiert, weil anderer Auftrag in Arbeit ist W#16#7003 – Verbindung belegt (kurzzeitig gesperrt) W#16#8001 – Ungültige Verbindungs-ID-Nummer oder Verbindung nicht konfiguriert
BUSY	OUTPUT	BOOL	Busy flag „TRUE“: Job wird bearbeitet

→ Hier geht es zur Projektierung mit den Objekten *Trendanzeige und Trendarchivanzeige*.

S7-Programmbeispiel für die Verwendung des SFC204

Ziel: gestoppten Trend mit der ID-Nummer 6163 fortsetzen

```
      UN    M   1500.4              // Wenn keine Anforderung gesetzt,
      BEB                           // dann Funktion beenden

      CALL  SFC  204
       LADDR   :=W#16#0             // VisuStage Verbindungs-ID-Nummer
                                    // (Bei Panel-SPSen immer W#16#0 verwenden!)
       TREND   :=6163               // ID-Nummer des in der VisuStage konfigurierten Trends
       INTERVAL:=W#16#1             // Aller 1s Werte aufzeichnen
       RET_VAL :=MW1000             // Resultat / Rückgabewert
       BUSY    :=M1700.4            // TRUE = Anforderung in Abarbeitung

      U     M   1700.4              // Trend fortsetzen ist in Abarbeitung,
      BEB                           // dann Funktion beenden
      R     M   1500.4              // Setze Anforderungssignal zurück

      L     MW  1000                // Resultat / Rückgabewert
      L     0                       // Kein Fehler
      ==I                           // Wenn erfolgreich erledigt, oder kein Fehler auftrat,
      BEB                           // dann Funktion beenden

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

      BEA                           // Ende der Funktion
```

Trendstatusabfrage mit SFC 205 "TRENDSTA"

Mit dem SFC205 “TRENDSTA” wird der Status des Trends abgefragt. Dabei wird der Status der Trenderfassung und/oder der Trendarchivierung zurückgegeben.

Parameter	Deklaration	Datentyp	Beschreibung
LADDR	INPUT	WORD	logische Adresse = 0 (für alle Panel-SPS) logische Adresse = 1...16 (Verbindungs-ID-Nummer zum Panel-HMI)
TREND	INPUT	INT	Trend ID-Nummer (von VisuStage vergeben, bei Ressourcen)
RET_VAL	OUTPUT	WORD	Fehlercodes W#16#0000 – Kein Fehler W#16#8002 – VisuStage läuft nicht / nicht konfiguriert W#16#8003 – Ungültige „TREND“-ID-Nummer W#16#8xyy – Allgemeine Fehlercodes (kompatibel zu STEP®7) nur für Panel-HMI relevant: W#16#7000 – Job wird bearbeitet W#16#7001 – Ungültiger Verbindungszustand, z.B. nicht verunden W#16#7002 – Auftrag nicht akzepiert, weil anderer Auftrag in Arbeit ist W#16#7003 – Verbindung belegt (kurzzeitig gesperrt) W#16#8001 – Ungültige Verbindungs-ID-Nummer oder Verbindung nicht konfiguriert
BUSY	OUTPUT	BOOL	Busy flag „TRUE“: Job wird bearbeitet
TREND_STATUS	OUTPUT	BYTE	Trendstatuscode B#16#00 – Trendaufzeichnung ist nicht initialisiert B#16#01 – Trendaufzeichnung ist initialisiert und läuft B#16#02 – Trendaufzeichnung ist beendet/ gestoppt
ARCHIVE_STATUS	OUTPUT	BYTE	Archivierungsstatuscode B#16#00 – Trendarchivierung ist nicht initialisiert B#16#01 – Trendarchivierung ist initialisiert und läuft B#16#02 – Trendarchivierung wurde erfolgreich beendet B#16#03 – SD-Card ist nicht gesteckt oder wurde nicht erkannt B#16#04 – Trendarchivierung gestoppt: Konnte Datei nicht erstellen B#16#05 – Trendarchivierung gestoppt: Konnte Datei nicht lesen B#16#06 – Trendarchivierung gestoppt: Konnte Dateigröße nicht einstellen B#16#07 – Trendarchivierung gestoppt: Konnte Header nicht in Datei schreiben B#16#08 – Trendarchivierung gestoppt: Fehler bei Dateisuche B#16#09 – Trendarchivierung gestoppt: Konnte Aufzeichnung nicht in Datei schreiben

→ Hier geht es zur Projektierung mit den Objekten *Trendanzeige und Trendarchivanzeige*.

S7-Programmbeispiel für die Verwendung des SFC205

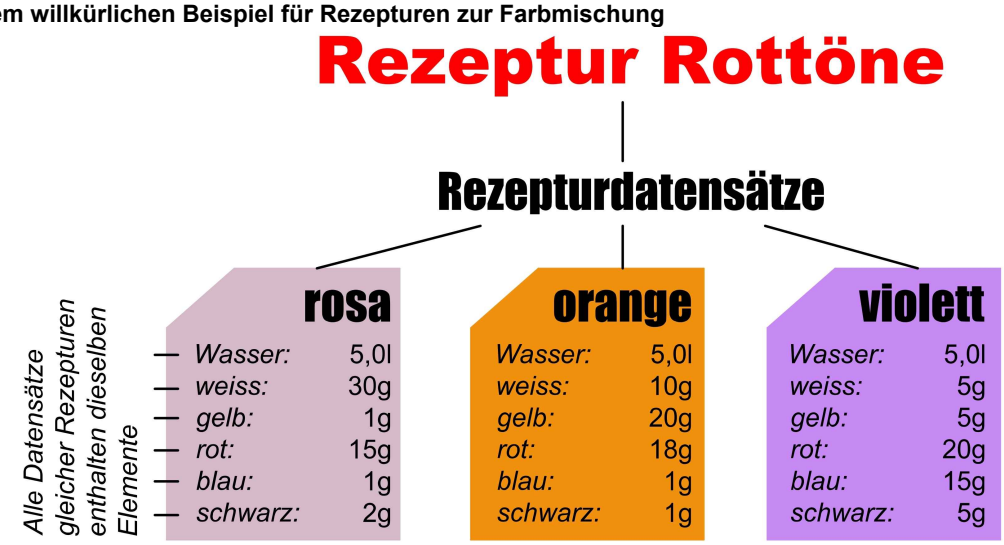
Ziel: Status des Trends mit der ID-Nummer 6163 prüfen

```
      UN    M   1500.5                  // Wenn keine Anforderung gesetzt,
      BEB                               // dann Funktion beenden

      CALL  SFC  205
       LADDR          :=W#16#0          // VisuStage Verbindungs-ID-Nummer
                                        // (Bei Panel-SPSen immer W#16#0 verwenden!)
       TREND          :=6163            // ID-Nummer des in der VisuStage konfigurierten Trends
       RET_VAL        :=MW1000          // Resultat / Rückgabewert
       BUSY           :=M1700.5         // TRUE = Anforderung in Abarbeitung
       TREND_STATUS   :=MB2008          // Statuscode des Trends
       ARCHIVE_STATUS:=MB2009           // Statuscode der Trendarchivierung

      U     M   1700.5                  // Trend starten ist in Abarbeitung,
      BEB                               // dann Funktion beenden
      R     M   1500.5                  // Setze Anforderungssignal zurück

      L     MW  1000                    // Resultat / Rückgabewert
      L     0                           // Kein Fehler
      ==I                               // Wenn erfolgreich erledigt, oder kein Fehler auftrat,
      BEB                               // dann Funktion beenden

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

      BEA                               // Ende der Funktion
```

Ressource Rezepturverwaltung

INSEVIS-Panels und Panel-SPSen können
- bei CPU Typ V bis zu 64 Rezepturen mit je bis zu 128 Datensätzen mit je bis zu 128 Elementen und bei
- bei CPU Typ P bis zu 64 Rezepturen mit je bis zu 256 Datensätzen mit je bis zu 256 Elementen verarbeiten.

Aufbau an einem willkürlichen Beispiel für Rezepturen zur Farbmischung

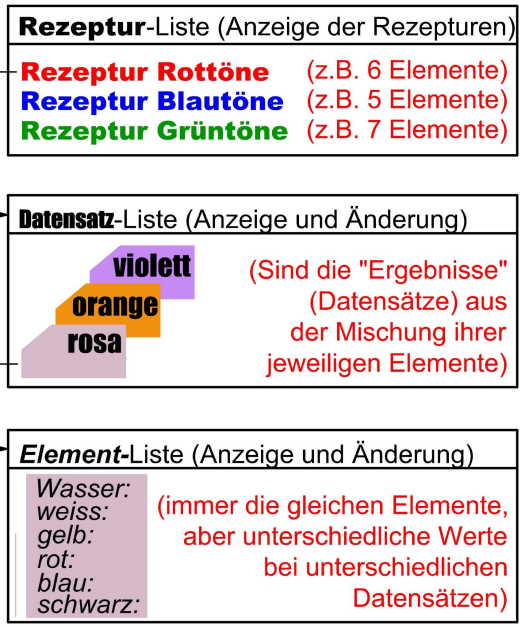


Die verschiedene Serien von Farbtönen (**Rottöne**, **Blautöne**, **Grüntöne**) stellen Rezepturen dar.
Jede Rezeptur besteht aus immer den gleichen Elementen (z.B. **Rottöne**: *Wasser, weiss, gelb, rot, blau, schwarz).*
Je nach Verhältnis der Elemente entstehen die **Datensätze** (Farben **rosa**, **orange**, **violett**) der Rezeptur **Rottöne**.

Der Bediener wählt zuerst eine Rezeptur aus der **Rezepturliste** aus. (im Beispiel die Rezeptur **Rottöne**)
Es können keine neuen Rezepturen zur Laufzeit angelegt werden, weil neue Elemente eingefügt werden müssen, die erst elektrisch in die Anwendung eingebunden werden müssen (Klappen, Waagen, etc.)

Die **Datensatzliste** zeigt die bestehenden Datensätze an. Hier kann ein bestehender Datensatz ausgewählt oder gelöscht, oder ein neuer hinzugefügt und gespeichert werden.
(im Beispiel die Datensätze (Farben) **rosa**, **orange**, **violett**)
Bei der Anlage des Rezeptes in der VisuStage wird dem Element bereits jeweils ein Wert zugewiesen. Damit entsteht quasi der erste Datensatz. Speichern unter anderem Namen dupliziert die Datensätze.

Nach Auswahl eines Datensatzes gelangt der Benutzer in die ***Elementliste***, wo der Wert der vorab in der Rezeptur festgelegten Elemente geändert werden kann.
(im Beispiel die Elemente *Wasser, weiss, gelb, rot, blau, schwarz)*



INSEVIS-spezifische Hinweise zur Rezepturverwaltung

1. INSEVIS- Steuerungen legen die Datensätze der Rezepturverwaltung in der Micro-SD-Card ab.
2. Es ist nur eine Rezepturanzeige in einem Screen möglich und auch sinnvoll.
3. Wenn die Visualisierung geändert und neu downgeloaded wird, wird auch die Rezepturstruktur mit sich selbst wieder überschrieben, aber die Daten auf der Micro-SD-Card bleiben erhalten. Damit bleiben alle gesicherten Datensätze der Rezepturen auch bei einem Update des Visualisierungsprojektes erhalten.
4. Wegen der Ablage der Daten auf einem externen Speicher ist bei der Rezepturverwaltung keine Simulation mit der SimuStage möglich.
5. Die Zugangsbeschränkung erfolgt über eine selbst zugangsbeschränkte Taste, die mit der Funktion Menüwechsel in das Menue „Rezepturverwaltung“ belegt ist.
6. Mit dem SFC206 kann ein automatisches Laden eines bestimmten Rezepturdatensatzes in die SPS durch ein Ereignis (z-B. Einlesen eines Barcodes o.ä.) vorgenommen werden. Die Zuweisungstabelle ist in S7 zu erstellen.
7. Der Export der Rezepturdatensätze ist mit RemoteStage ab Version 1.0.3.7. möglich

Projektierung an einem Beispiel für Rezepturen zur Farbmischung

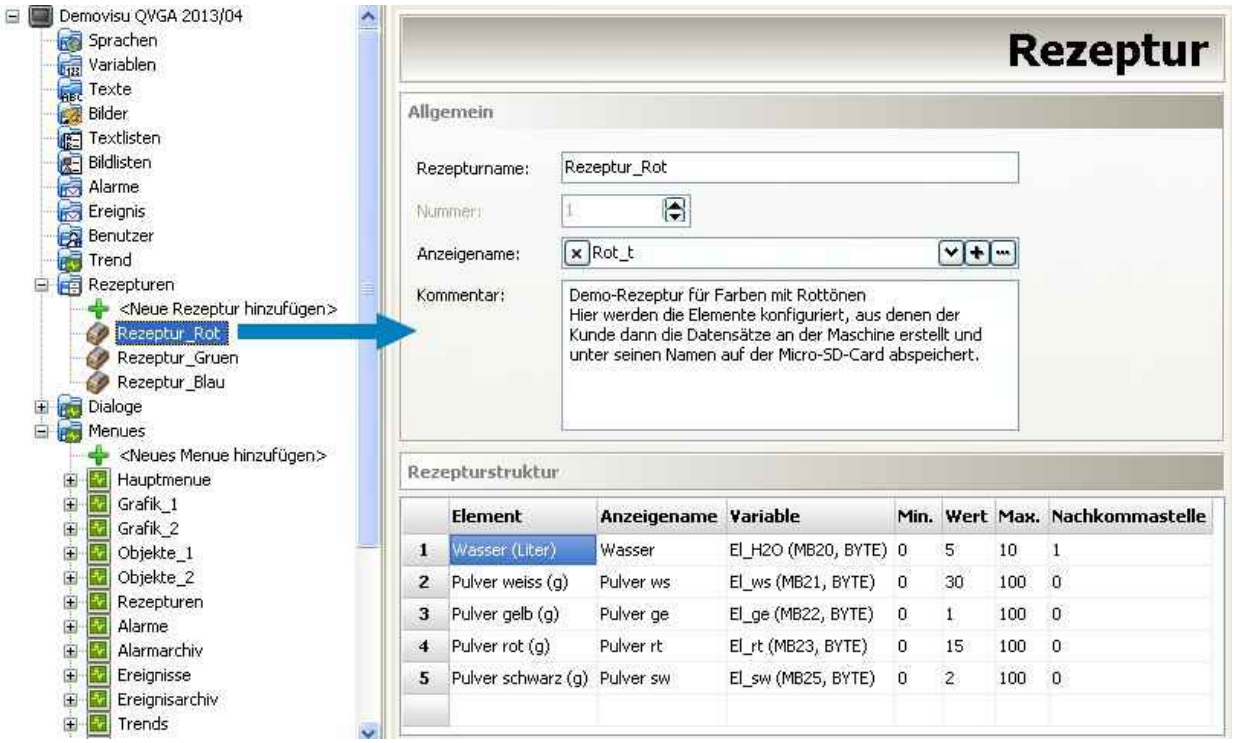
Neuanlage einer Rezeptur wählen, in neuem Fenster Rezeptur parametrieren

Allgemeine Angaben

Rezepturname:	interner (projektsprachenunabhängiger) Name der Rezeptur
Anzeigename:	sichtbarer (projektsprachenabhängiger) Name der Rezeptur
Kommentar:	Bemerkungen zu dieser Rezeptur eintragen
(Nummer)	automatisch hochgezählte interne Nummerierung der Rezeptur (nicht änderbar)

Rezepturstruktur

Element	interner (projektsprachenunabhängiger) Name des Elementes
Anzeigename:	sichtbarer (projektsprachenabhängiger) Name des Elementes (hier kann neben der Bezeichnung noch die physikalische Einheit (z.B. kg, sec., cm, etc.) mit angegeben werden). Basiert auf einer Textressource, die separat angelegt wird
Variable	Zuordnung einer Variable zu dem Element (optimal, wenn das S7-Programm die Variable so skaliert, dass die abgemessenen Werte gleich direkt eingegeben werden können)
Min.	Eingabe des Minimalwertes des Elements (Wenn Bediener einen kleineren Wert eingibt, wird automatisch dieser Minimalwert übernommen.)
Wert	Eingabe des Vorgabewertes des Elements (Damit werden die Werte des ersten Datensatzes quasi vorbelegt und im Speicher gehalten, bis der Datensatz erstmals gespeichert wird.)
Max.	Eingabe des Minimalwertes des Elements (Wenn Bediener einen größeren Wert eingibt, wird automatisch dieser Maximalwert übernommen.)
Nachkommastellen:	Eingabe der Anzahl von Nachkommastellen des Wertes



Rezepturhandling mit SFC 206 "RECIPE"

Mit dem SFC206 “RECIPE” können Rezepturdatensätze

- von der Micro-SD-Karte eingelesen und in die SPS geschrieben werden und
- von der SPS auf die Micro-SD-Karte geschrieben werden (Dateiname = Rezepturdatensatzname).

Mit dem SFC206 kann ein automatisches Laden eines bestimmten Rezepturdatensatzes in die SPS durch ein Ereignis (z.B. Einlesen eines Barcodes o.ä.) vorgenommen werden. Die dafür nötige Zuweisungstabelle (z.B. Welcher Barcode löst welchen Rezepturdatensatz aus) ist in S7 zu erstellen.

Parameter	Bezeichnung	Datentyp	Beschreibung
REQ	INPUT	BOOL	TRUE Zuweisen und Job starten
LADDR	INPUT	WORD	logische Adresse = 0 (für alle Panel-SPS) logische Adresse = 1...16 (Verbindungs-ID-Nummer zum Panel-HMI)
RECIPE	INPUT	INT	Rezeptur-Nummer (von VisuStage vergeben, bei Ressourcen)
REC_NAME	INPUT	STRING	Rezepturdatensatzname bis zu 60 ASCII-Zeichen, wenn mehr, dann Rückgabewert W#16#842A)
FUNCTION	INPUT	BYTE	Funktionsnummer 1 Lese Datensatz von SD-Karte und schreibe in SPS (SD → SPS) 2 Lese Datensatz von SPS und schreibe auf SD (SPS → SD) 3 Lese Datensatz von SPS und überschreibe SD (SPS → SD) 4 Lösche Datensatz von SD 5 Prüfe Vorhandensein von Datensätzen auf SD
RETVAL	OUTPUT	WORD	Rückgabewert des Jobs
BUSY	OUTPUT	BOOL	TRUE Job wird bearbeitet
DONE	OUTPUT	BOOL	TRUE Job ist erfolgreich abgeschlossen
ERROR	OUTPUT	BOOL	TRUE Job ist fehlgeschlagen, Fehlercode in RETVAL

Bedeutung der Rückgabewerte

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	1	ungültiger Verbindungszustand (z.B. nicht verbunden)
W#16#7002	0	0	1	Job nicht akzeptiert, weil vorheriger Job noch in Arbeit ist
W#16#7003	0	0	1	Verbindungsressource ist temporär blockiert
W#16#7004	0	0	0	REQ Parameter ist FALSE, kein Job in Bearbeitung und/oder Job nicht zugewiesen / gestartet (Zum Starten des Jobs muss REQ=TRUE gesetzt werden.)
W#16#8001	0	0	1	Ungültige Verbindungs-ID-Nummer
W#16#8002	0	0	1	Visualisierung läuft nicht (auf Panel-HMI) oder Rezeptur nicht konfiguriert
W#16#8003	0	0	1	Ungültige Rezeptur-ID-Nummer
W#16#8004	0	0	1	Dateisystem ist nicht bereit (z.B. SD-Karte nicht gesteckt)
W#16#8005	0	0	1	Konnte nicht auf Datensatz-Datei auf SD-Karte zugreifen
W#16#8006	0	0	1	Ungültiger Dateiinhalt
W#16#8007	0	0	1	Datensatzdatei existiert bereits auf SD-Karte
W#16#842A	0	0	1	Ungültiger Rezepturdatensatzname REC_NAME- Parameter (z.B. Leerstring oder mehr als 60 ASCII-Zeichen)
W#16#852B	0	0	1	Ungültige Funktions ID-Nummer in FUNCTION- Parameter
W#16#8xyy	0	0	1	Allgemeiner Fehlercode General (S7-kompatibel)

→ Hier geht es zur Projektierung mit dem Objekt *Rezepturanzeige*.

S7-Programmbeispiel für die Verwendung des SFC206

Ziel: Erstellen (bzw. Überschreiben eines bestehenden) Rezeptur-Datensatzes auf der Micro-SD-Karte

```
      CALL  SFC  206
       REQ      :=M1500.6              //  TRUE = Anforderung zur Erstellung eines neuen
                                       //  Rezeptur-Datensatzes
       LADDR    :=W#16#0               //  VisuStage Verbindungs-ID-Nummer
                                       //  (Bei Panel-SPSen immer W#16#0 verwenden!)
       RECIPE   :=1                    //  In der VisuStage erzeugte Rezepturnummer
       REC_NAME:=DB205.RECORD_NAME     //  Rezeptur-Datensatzname
       FUNCT    :=B#16#3               //  Datensatz auf Micro-SD-Karte schreiben
                                           (oder bestehenden Datensatz dort überschreiben)
       RET_VAL :=MW1000                //  Resultat / Rückgabewert
       BUSY     :=M1700.6              //  TRUE = Anfrage in Abarbeitung
       DONE     :=M1002.0              //  TRUE = Erfolgreich erstellt
       ERROR    :=M1002.1              //  TRUE = Fehler aufgetreten

      U     M   1700.6                //  Rezeptur-Datensatz ist in Abarbeitung
      BEB                             //  dann Funktion beenden
      R     M   1500.6                //  Setze Anforderungssignal zurück

      U     M   1002.0                //  Wenn erfolgreich erledigt,
      BEB                             //  dann Funktion beenden

      UN    M   1002.1                //  Wenn kein Fehler auftrat,
      BEB                             //  dann Funktion beenden

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

      BEA                             // Ende der Funktion
```

Allgemeine Funktionen

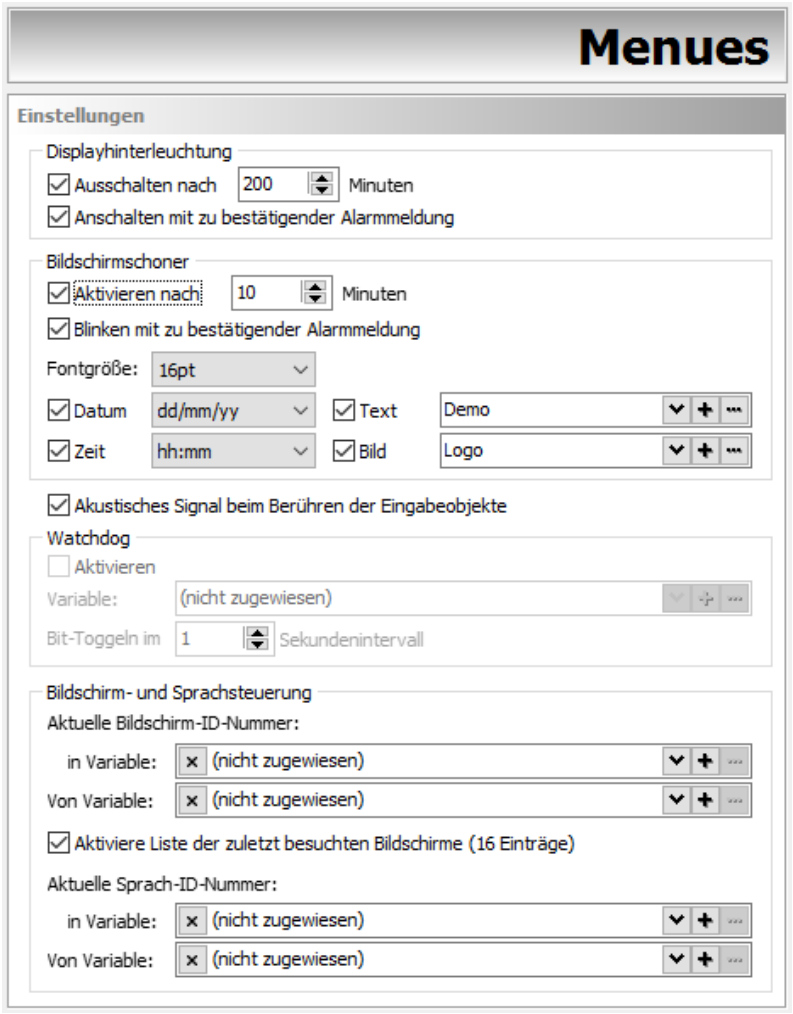
Kopieren/ Ausschneiden/ Einfügen, Löschen und Gruppieren/Auflösen

Es können mehrere Objekte mit der linken Maustaste markiert und mit der rechten Maustaste bearbeitet werden. Das Einfügen ist im selben Projekt in jedem Menue wieder mit dergleichen X-Y-Position möglich. Werden die Objekte im selben Menue eingefügt, werden sie mit einem X-Y Offset von je 5 Pixeln eingefügt, um die Übersicht zu bewahren.

Menues

Der Unterpunkt „Menues“ zeigt eine Übersicht über alle bestehenden Menues der Visualisierung.
Hier werden auch generelle Einstellungen getroffen bezüglich

- **Displayhinterleuchtung**
  - selbstständiges Ausschalten
  - Alarmmeldung nach Anschalten
- **Bildschirmschoner**
  - selbstständiges Einschalten
  - Blinken bei Alarmmeldung
  - Design einer Anzeige von Datum/Uhrzeit, Bild und/oder Text, welches während des Bildschirm-schoners angezeigt werden soll
- **Akustischen Signale** (durch den eingebauten Buzzer) Als Rückmeldung bei Tastendruck (Clickton)
- **Watchdog** - Überwachung der Verbindung zur SPS mit einem toggelndem (Live-) Bit ***(nur bei Panel-HMI)***
- **Steuerung** von
  - Bildschirmen (Menues) und
  - Sprachen durch Variablen (aktuelles Menue/Sprache melden der ID an Variable oder umschalten durch Variable)
- **Aktivierung** eine Historie der letzten 16 besuchten Seiten (für Tastenfunktion „Menue wechseln → vorheriges Menue“)



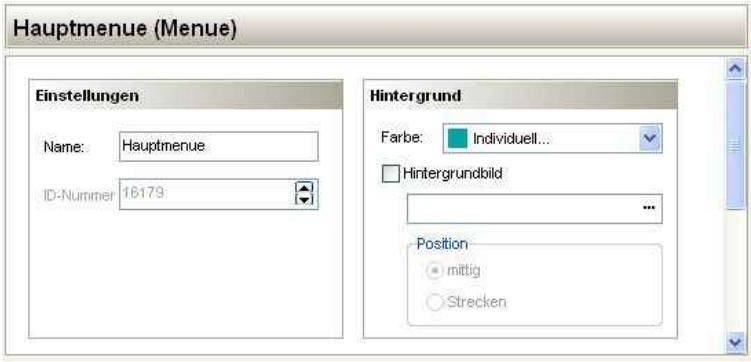
Hinweis: *Die Bildgröße (in Pixeln) des Bildes im Bildschirmschoner muss* ***DEUTLICH KLEINER*** *als die Auflösung des Displays sein, damit der Bildschirmschoner „wandern“ kann.*

Hintergrundbild

Es können Farben oder Bilder als Hintergrund verwendet werden. Bilder erfordern wesentlich mehr Speicher als Farben, sind daher nur bei Bedarf einzusetzen.

Die ID-Nummer des Menüs wird verwendet, wenn ein Bildschirmwechsel durch die SPS erzeugt werden soll und ist nicht manuell veränderbar.

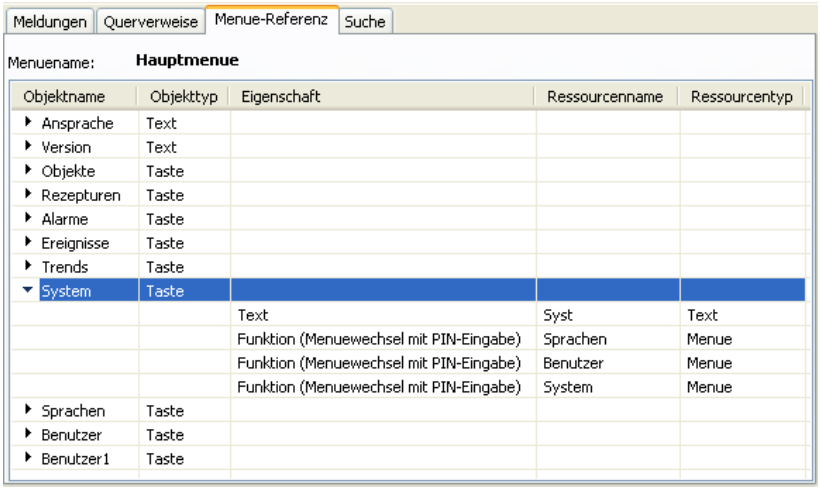
Erreichbar durch Mausklick direkt auf den Hintergrund (neben die Objekte)



Anzeige der verwendeten Ressourcen in einem Menue

Zur Übersicht über die in einem Menuefenster verwendeten Ressourcen gibt es die Funktion „Menue-Referenz“

- Menue im Menuebaum (links) auswählen und markieren.
- Rechte Maustaste öffnet Pull-down-Menue, dort „Menue-Referenz“ anwählen
- Im Meldungsfenster erscheinen alle in diesem Menue verwendeten Objekte (ein-/ ausklappbar)



Putzbildschirm

Der **Putzbildschirm** wird mittels einer Funktion (über den Reiter „Funktion“) ausgelöst und dabei parametriert.

Sichtbarkeit

Die Sichtbarkeit von Objekten kann variablenabhängig gesteuert werden und wird wie folgt eingestellt:

- - Auf Reiter „Sichtbarkeit“ gehen
- - mit „ + “ eine neue Variable hinzufügen
  - mit „ v “ eine Variable aus der Variablenliste auswählen
  - mit „ … “ bestehende Variable ändern

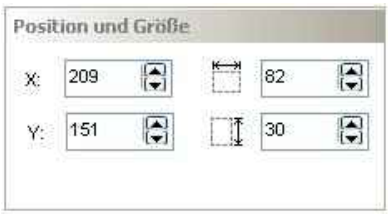
In diesem Beispiel wird das Objekt versteckt, wenn das 4. Bit (von rechts gezählt) den binären Wert 1 annimmt, sonst ist das Objekt sichtbar.

Im Beispiel oben wird das Objekt versteckt, wenn die Variable einen Wert zwischen 4 und 6 annimmt, sonst ist das Objekt sichtbar.

Objektplatzierung

Die gewünschten Objekte werden im rechten Katalogbaum angewählt und mit der linken Maustaste im Menü grob aufgezogen.
Die Feinplatzierung wird in Pixeln vorgenommen und erfolgt über die Box

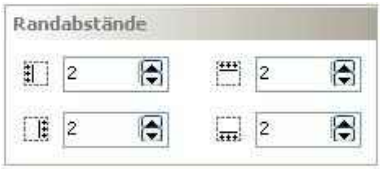
„Position und Größe“ oder über die Tastenkombination SHIFT + Pfeiltasten (jeweils 8 Pixel) Die Position von X und Y wird in Pixeln angegeben, wobei die Koordinaten X;Y / 0;0 im Display **oben links** sind.
Die Größe des Objektes wird in den rechten Feldern in Pixeln eingestellt.



Randabstände

Hier wird der Abstand von den auf einem Objekt befindlichen Texten und/oder Symbolen zum Rand eingestellt.

Diese Einstellungen werden in Pixeln vorgenommen.



Gruppieren

Um mehrere Objekte in ihrer Lage zueinander zu fixieren, werden diese markiert und (rechte Maustaste) gruppiert.
Das ist besonders bei Anlegen von Bibliothekselementen sinnvoll (Namen vergeben!).

Die Steuerung kann erfragen, welches Menue angezeigt wird, und kann dieses Menue auch in an anderes wechseln.

Abfrage der aktuellen Bildschirm-ID-Nummer mit dem SFC 200 "SCR_GET"

Die ID-Nummer des aktuell angezeigten Bildschirmes kann durch den SFC 200 “SCR_GET“ abgefragt werden.

Parameter	Deklaration	Datentyp	Beschreibung
LADDR	INPUT	WORD	logische Adresse = 0 (für alle Panel-SPS) logische Adresse = 1...16 (Verbindungs-ID-Nummer zum Panel-HMI)
RET_VAL	OUTPUT	WORD	Fehlercodes W#16#0000 – kein Fehler W#16#8002 – VisuStage läuft nicht / nicht konfiguriert W#16#8xyy – Allgemeine Fehlercodes (kompatibel zu STEP®7) nur für Panel-HMI relevant: W#16#7000 – Job wird bearbeitet W#16#7001 – Ungültiger Verbindungszustand, z.B. nicht verunden W#16#7002 – Auftrag nicht akzepiert, weil anderer Auftrag in Arbeit ist W#16#7003 – Verbindung belegt (kurzzeitig gesperrt) W#16#8001 – Ungültige Verbindungs-ID-Nummer oder Verbindung nicht konfiguriert
BUSY	OUTPUT	BOOL	Busy flag „TRUE“: Job wird bearbeitet
SCREEN	OUTPUT	INT	Aktuell angezeigte Bildschirm-ID-Nummer, (von VisuStage vergeben, bei Ressourcen)
LANG	OUTPUT	INT	Aktuell ausgewählte Sprach-ID-Nummer (entspricht der internationalen Norm - bei csv-Export der Texte sichtbar)

Programmierbeispiel für die Verwendung des SFC200

Ziel:

```
      UN    M   1500.0              // Wenn keine Anforderung gesetzt,
      BEB                           // dann Funktion beenden

      CALL  SFC  200
       LADDR  :=W#16#0              // VisuStage Verbindungs-ID-Nummer
                                    // (Bei Panel-SPSen immer W#16#0 verwenden!)
       RET_VAL:=MW1000              // Resultat / Rückgabewert
       BUSY   :=M1700.0             // TRUE = Lesen in Bearbeitung
       SCREEN :=MW2002              // Aktuelle Menue-ID-Nummer
       LANG   :=MW2004              // Aktuelle Sprach-ID-Nummer

      U     M   1700.0              // Lesen in Bearbeitung,
      BEB                           // dann Funktion beenden
      R     M   1500.0              // Rücksetzen des Anforderungssignals

      L     MW  1000                // Resultat / Rückgabewert
      L     0                       // Kein Fehler
      ==I                           // Wenn erfolgreich erledigt, oder kein Fehler auftrat,
      BEB                           // dann Funktion beenden

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

      BEA                           // Ende der Funktion
```

Bildschirmwechsel mit dem SFC 201 "SCR_SET"

Der Bildschirm kann mit dem SFC 201 “SCR_SET“ gewechselt werden.

Parameter	Deklaration	Datentyp	Beschreibung
LADDR	INPUT	WORD	logische Adresse = 0 (für alle Panel-SPS) logische Adresse = 1...16 (Verbindungs-ID-Nummer zum Panel-HMI)
SCREEN	INPUT	INT	Bildschirm-ID-Nummer, in die gewechselt werden soll (von VisuStage vergeben, bei Ressourcen)
RET_VAL	OUTPUT	WORD	Fehlercodes W#16#0000 – Kein Fehler W#16#8002 – VisuStage läuft nicht / nicht konfiguriert W#16#8003 – Unzulässige Bildschirm-ID-Nummer W#16#8xyy – Allgemeine Fehlercodes (kompatibel zu STEP®7) nur für Panel-HMI relevant: W#16#7000 – Job wird bearbeitet W#16#7001 – Ungültiger Verbindungszustand, z.B. nicht verunden W#16#7002 – Auftrag nicht akzepiert, weil anderer Auftrag in Arbeit ist W#16#7003 – Verbindung belegt (kurzzeitig gesperrt) W#16#8001 – Ungültige Verbindungs-ID-Nummer oder Verbindung nicht konfiguriert
BUSY	OUTPUT	BOOL	Busy flag „TRUE“: Job wird bearbeitet

Programmierbeispiel für die Verwendung des SFC201

Ziel: Wechsel in das Menue mit der ID-Nummer 16917

```
      UN    M   1500.1              // Wenn keine Anforderung gesetzt,
      BEB                           // dann Funktion beenden

      CALL  SFC  201
       LADDR  :=W#16#0              // VisuStage Verbindungs-ID-Nummer
                                    // (Bei Panel-SPSen immer W#16#0 verwenden!)
       SCREEN :=16917               // Menue-ID-Nummer, wohin gewechselt werden soll
       RET_VAL:=MW1000              // Resultat / Rückgabewert
       BUSY   :=M1700.1             // TRUE = Anforderung in Bearbeitung

      U     M   1700.1              // Anforderung in Bearbeitung,
      BEB                           // dann Funktion beenden
      R     M   1500.1              // Rücksetzen des Anforderungssignals

      L     MW  1000                // Resultat / Rückgabewert
      L     0                       // Kein Fehler
      ==I                           // Wenn erfolgreich erledigt, oder kein Fehler auftrat,
      BEB                           // dann Funktion beenden

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

       BEA
```

Statische Objekte

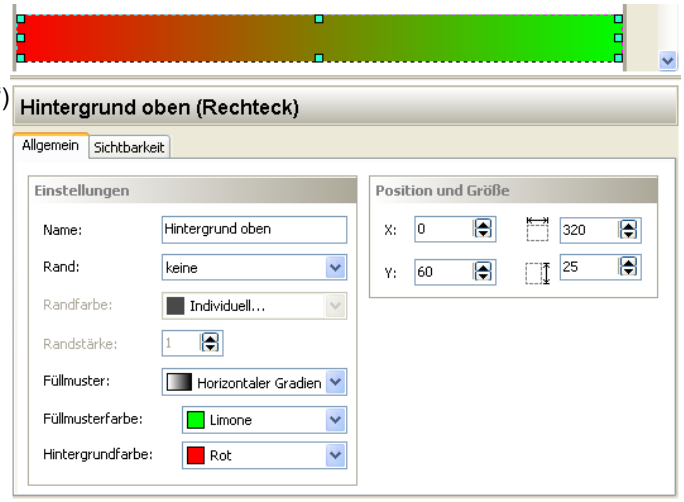
Linien, Rechtecke, Texte, Bilder

Bilder

Objekt platzieren: (siehe „Allgemeine Funktionen“)
Sichtbarkeit einstellen: (siehe „Allgemeine Funktionen“)
Objekt parametrieren (Reiter „Allgemein“):

- Bestehende **Ressource** auswählen oder neue direkt anlegen
- **Strecken: „an Größe anpassen“** erleichtert das Platzieren, weil das Bild das aufgezogene Objekt voll ausfüllt (Die Relation wird dabei von der Objektgröße bestimmt)
- **Ausrichtung** des Objektes (wenn bei Strecken „proportional“ oder „keine“ gewählt wurde
- **Drehen** und **Spiegeln** des Objektes

Rechtecke



Objekt platzieren: (siehe „Allgemeine Funktionen“)
Sichtbarkeit einstellen: (siehe „Allgemeine Funktionen“)
Objekt parametrieren (Reiter „Allgemein“):

- Im Beispiel wurde ein Rechteck mit der Größe 320x25px an Position x;y=0;60ohne Rand mit horizontalen Farbgradient von rot nach grün erstellt. Andere Füllmuster, Ränder und Farben sind selbsterklärend.

Linie (komplett selbsterklärend)

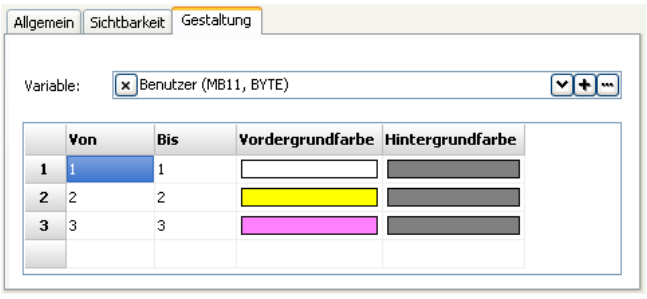
Texte

Objekt platzieren: (siehe „Allgemeine Funktionen“)
Sichtbarkeit einstellen: (siehe „Allgemeine Funktionen“)
Objekt parametrieren (Reiter „Allgemein“):

- Bestehende **Ressource** auswählen oder neue direkt anlegen
- Font, Fett, Kursiv, Größe, Farbe zuweisen
- Alignment horizontal/vertikal zuweisen,
- Ankreuzen, wenn der Text mehrzeilig sein soll

Objekt parametrieren (Reiter „Gestaltung“):

- variablenabhängige Farbumschläge konfigurieren



Allgemeine Tips

- statische Objekte **dürfen sich überlappen**.
- Die Linienstärke geht symmetrisch von der gezeigten Mittellinie aus.
- Für Bilder das Dateiformat PNG benutzen (transparenter α-Kanal lässt Hintergrund durch die transparenten Bereiche durchscheinen, z.B. an abgerundeten Ecken, etc.)

Dynamische Objekte

diverse Tastenarten

Die VisuStage bietet Standard- , dynamische und Text-Tasten.

- Die **Standard-Tasten** liegen als Template vor und verfügen über 65000 Farben. Sie verfügen jeweils über ein fest zugewiesenes ein Design, einen Text und ein Symbol.
- **Dynamische Tasten** werden in der VisuStage definiert, verfügen nur über 256 Farben, bieten aber zusätzlich einen Farbumschlag, multilinguale Textumschläge (Textlisten) und Symbolumschläge (Bildlisten).
- **Texttasten** benötigt man, wenn man Textumschläge mit sehr umfangreichen Textlisten benutzen möchte.

Alle Tastenarten verfügen über eine sogenannte „Totmann-Funktion“ (**Ausführen der Funktion auch bei seitlichem Verlassen der Taste**), die anwählbar ist.

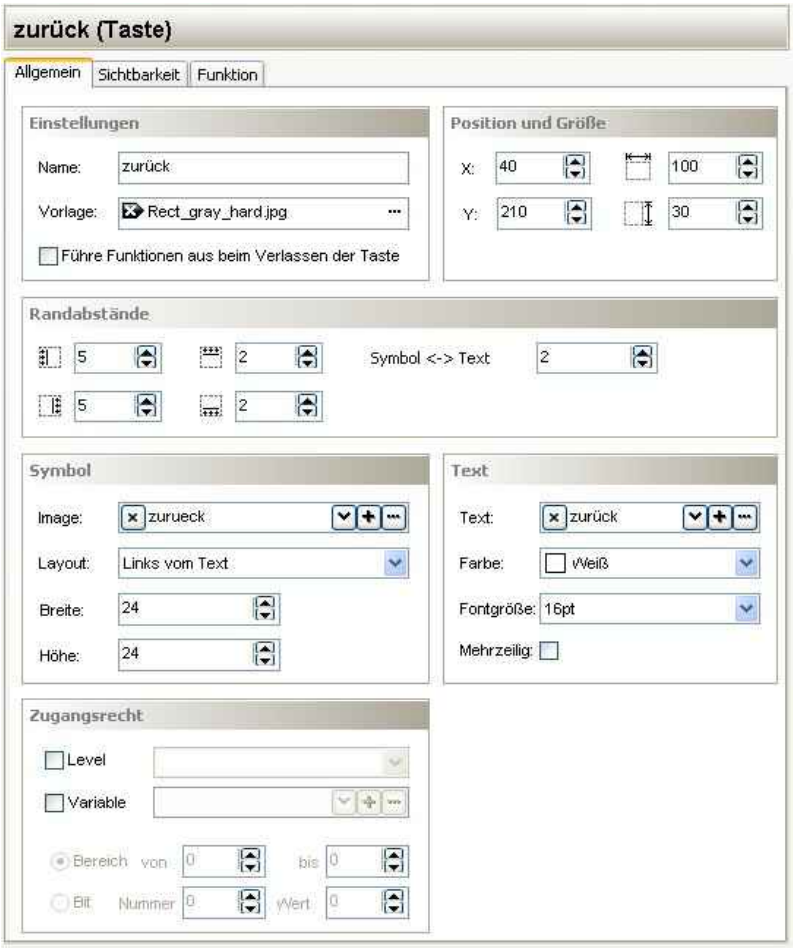
Normalerweise kann man bei Fehlbetätigung einer Taste die Ausführung verhindern, in dem man sich seitlich „ausschleicht“. Das ist voreingestellt, aber manchmal gefährlich, wie das Beispiel der Funktion „Bit invertieren“ zeigt:
Bit = 1 → Motor einschalten bei Taste drücken, Bit = 0 → Motor ausschalten bei Taste loslassen.
Ist der Haken nicht aktiviert, kann der Benutzer versehentlich seitlich aus dem Tastenbereich rausrutschen und der Motor fährt weiter bis zum Endkontakt.

Standard-Tasten

- **Objekt platzieren:** (siehe „Allgemeine Funktionen“)
- **Sichtbarkeit einstellen:** (siehe „Allgemeine Funktionen“)

Objekt parametrieren (Reiter „Allgemein“):

- **Ausrichten** des Textes/ des Symboles zum Objektrand über Box „Randabstände“ (in Pixeln)
- **Symbole** (Bilder) auswählen und konfigurieren, welche auf der Taste stehen sollen (oder neu erstellen)
- **Text** (Label) auswählen und konfigurieren, welcher auf der Taste stehen sollen (oder neu erstellen)
- **Zugangsrechte** festlegen (mit den unter „Benutzer“ zugewiesenen Levels oder Variablen)



Verwendung eigener Tasten
Tasten werden mit installiert. Es ist jedoch möglich, EIGENE Tasten zu verwenden:

- 3 **gleichnamige** Bilder der Standard-Tasten ablegen in je einem der 3 Verzeichnisse
  \inactiv, \pressed und \unpressed des Stammverzeichnisses
  C:\Programme\INSEVIS\
  VisuStage\Buttons.
  (In der Mitte der Taste möglichst gleichen Farbton lassen.)

Achtung: Wenn Projekt als *.vsproj und *.res an einem anderen Arbeitsplatz geöffnet wird, wo diese Tasten nicht installiert sind, erscheinden diese Tasten auch nicht in der Visualisierung!

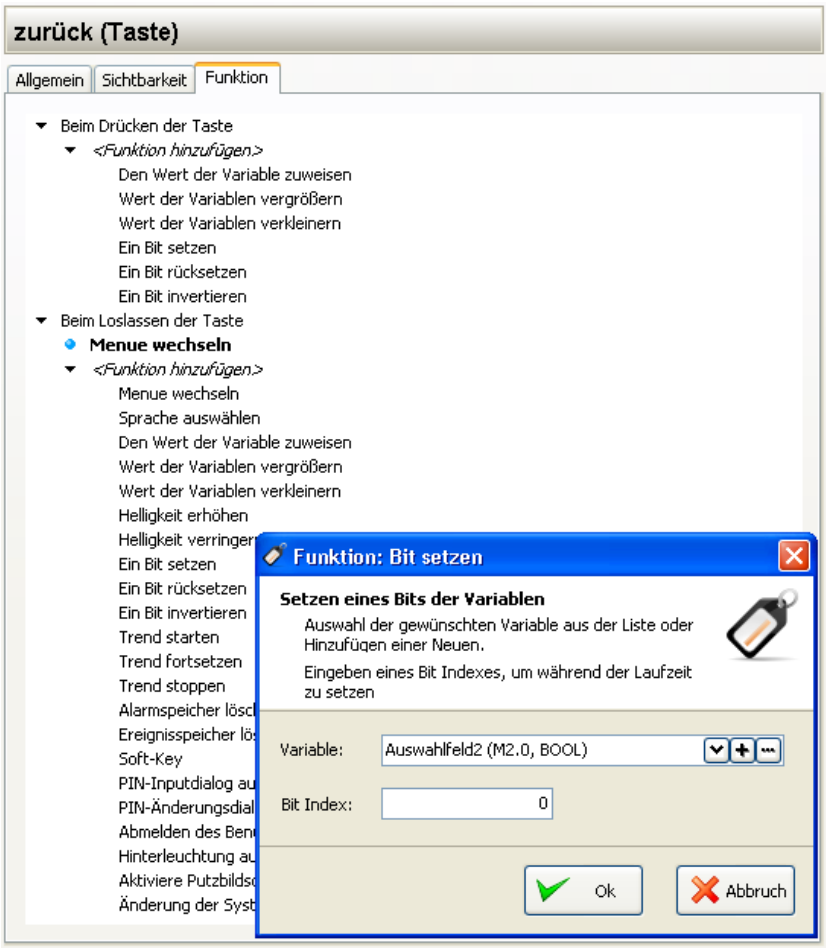
Visualisierung mit der Software „VisuStage“

Funktion zuweisen (Reiter „Funktion“

- Entscheid, WANN die Funktion ausgeführt werden soll:
  - beim DRÜCKEN oder
  - beim LOSLASSEN
  der Taste.
- Nach Auswahl (Doppelklick) einer Funktion öffnet sich ein Fenster, um die jeweils benötigten Daten dafür einzugeben.
- Einer Taste können mehrere Funktionen zugewiesen werden.
- Einer Taste können Funktionen beim Drücken und beim Loslassen zugewiesen werden.

Verwendung der Soft-Key-Funktion

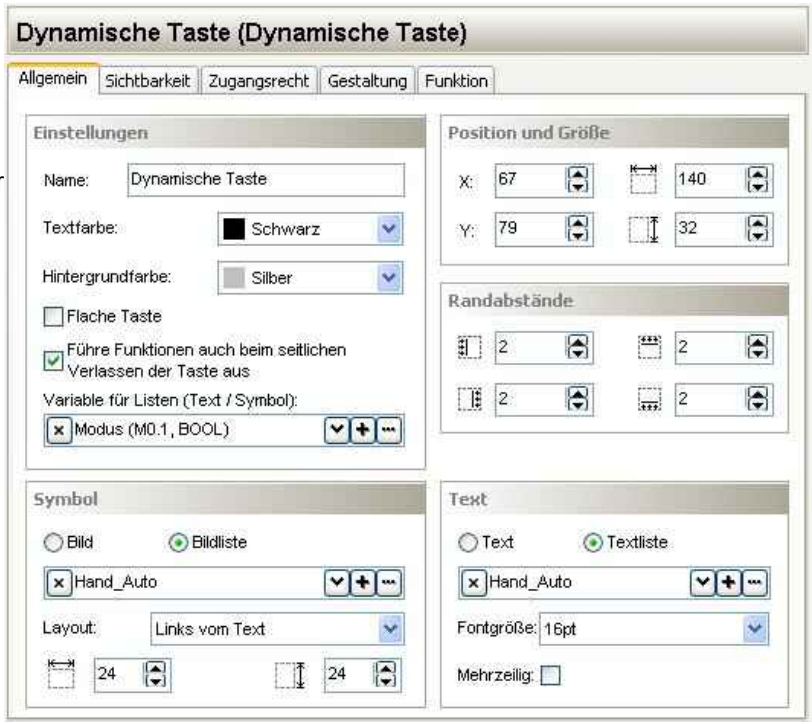
- Dient zur manuellen Anlage der Navigationsbuttons von Betrachtern (Viewern) und ist nur im selben Menue verwendbar.



Dynamische Tasten

Objekt platzieren und parametrieren wie bei den Standard-Tasten, aber ZUSÄTZLICH einstellbar:

- **Symbolumschläge:** (im Beispiel Hand/Auto-Zeichen)
  Bildlisten auswählen, welche auf der dynamischen Taste stehen sollen
- **Textumschläge:** (im Beispiel Tastenbeschreibung/ -zustand))
  Textlisten auswählen, welche auf der dynamischen Taste stehen sollen
- **Variablenangabe**, die Symbol- und Textumschlag steuert



Objekt parametrieren (Reiter „Gestaltung“):

Festlegen der variablenabhängigen Farbumschläge

- **Farbumschläge**
  für Vordergrund (Textfarbe) und Hintergrund (Farbe der Tastenfläche)
- **zusätzliche Variablenangabe**, die den Farbumschlag steuert

ACHTUNG:
Wenn ein anderer Wert als die in den Zeilen 1...n angegeben Werte anliegt, wird IMMER die Gestaltung angezeigt, die an ERSTER STELLE (Zeile 1) zugewiesen wurde.

Texttasten

gleichen in der Verwendung dynamischen Tasten ohne Symbolumschlag, nur dass Textumschläge mit größeren Textlisten möglich sind, ohne den Speicherbedarf zu sehr zu belasten.

Keine Verwendung eigene Tasten
Die dynamischen und Texttasten werden in der VisuStage selbst angelegt, eigene Sonderformen stehen nicht zur Verfügung.

Allgemeine Tips

- dynamische Objekte sollten sich **NICHT überlappen**. (Ausnahme: Touchflächen), da eine zustandsabhängige Änderung des Hintergrundes unter transparenten Flächen ignoriert wird. Es wird dafür bei dem Kompillieren eine Fehlermeldung ausgegeben.
- Das Verwenden von gleichen Objekten und wenig Sprachen spart Speicherplatz!
- Simulation mit „ ► “ erzeugt Fehlerbericht, der direkt zum fehlerhaften Objekt führt.

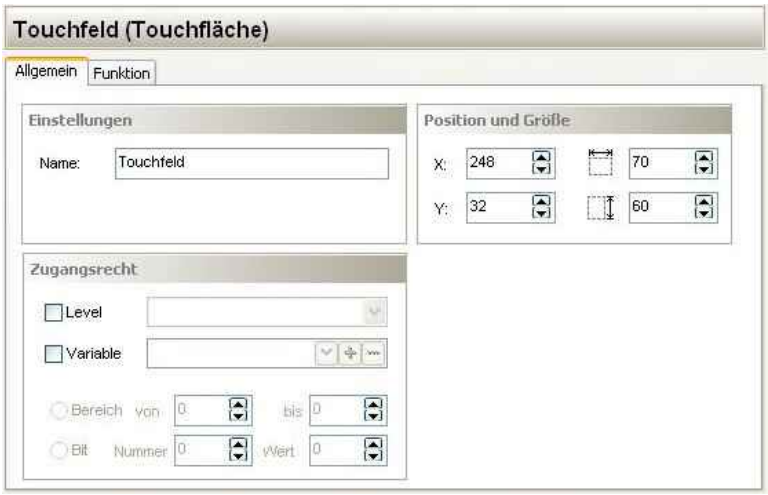
Touchflächen

Diese Objekte werden über ein Bild, Text oder andere statische Objekte gelegt, um eine Funktion auszulösen (wie eine unsichtbare Taste).

Objekt platzieren:
(siehe „Allgemeine Funktionen“)

Objekt parametrieren
(Reiter Allgemein):

- **Zugangsrechte** festlegen (mit den unter „Benutzer“ zugewiesenen Levels oder Variablen)
- **Funktion** zuweisen : siehe Tasten



Allgemeine Tips

- Touchflächen dürfen andere dynamische Objekte überlappen.
- Simulation mit „ ► “ erzeugt Fehlerbericht, der direkt zum fehlerhaften Objekt führt.

Visualisierung mit der Software „VisuStage“

Zustandsflächen

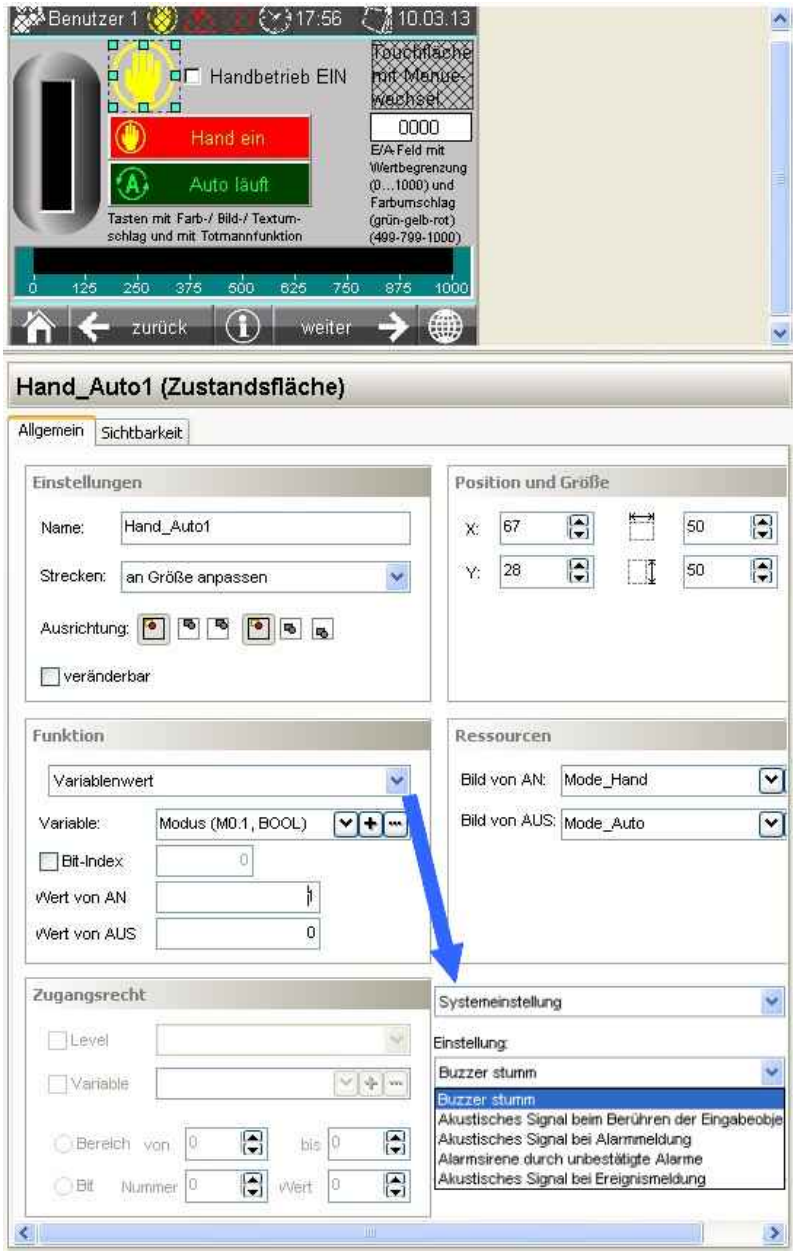
Diese Objekte dienen zur wechselnden Anzeige zweier Bilder, entweder über Variable oder per Hand durch Berühren (wenn „veränderbar“ gewählt wurde).

Objekt platzieren:
(siehe „Allgemeine Funktionen“)
Sichtbarkeit einstellen:
(siehe „Allgemeine Funktionen“)

Objekt parametrieren (Reiter Allgemein):

- **Strecken: „an Größe anpassen“** erleichtert das Platzieren, weil das Bild das aufgezogene Objekt voll ausfüllt
- **Ausrichtung** des Objektes (wenn bei Strecken „proportional“ oder „keine“ gewählt wurde
- **Variable** zuweisen oder neu erstellen und Wert für **AN** und **AUS** festlegen

  oder
- **Systemeinstellung** zuweisen: verschiedene Einstellungen der Akustischen Rückmeldung durch den integrierten Buzzer
- **Bilder** zuweisen oder neu erstellen (jeweils ein Bild, welches im **AN**- bzw. im **AUS** Zustand gezeigt werden soll)
- wenn **veränderbar** ausgewählt wurde, können die **Zugangsrechte** festgelegt werden (mit den unter „Benutzer“ zugewiesenen Levels oder Variablen)



Allgemeine Tips

- dynamische Objekte sollten sich **NICHT überlappen**. (Ausnahme: Touchflächen), da eine zustandsabhängige Änderung des Hintergrundes unter transparenten Flächen ignoriert wird. Es wird dafür bei dem Kompillieren eine Fehlermeldung ausgegeben.
- Für Bilder das Dateiformat PNG benutzen (transparenter α-Kanal lässt Hintergrund durchscheinen, z.B. an abgerundeten Ecken.)
- Das Verwenden von gleichen Objekten und wenig Sprachen spart Speicherplatz!
- Simulation mit „ ► “ erzeugt einen Fehlerbericht, der direkt zum fehlerhaften Objekt führt.

Auswahl- und Markierungsfelder

Mit **Auswahlfeldern** kann eine **Entweder-Oder-Wahl** (z.B. bei Alarm Kommen und Gehen) getroffen werden.

Bei **Markierungsfeldern** können auch **mehrere Felder einzeln an- oder abgewählt** werden (z.B. bei Filtern der Trendkanäle, Umschalten einer Zustandsfläche) und stehen für Buzzer-Systemfunktionen zur Verfügung

Objekt platzieren:
(siehe „Allgemeine Funktionen“)
Sichtbarkeit einstellen:
(siehe „Allgemeine Funktionen“)

Objekt parametrieren
(Reiter „Allgemein“)

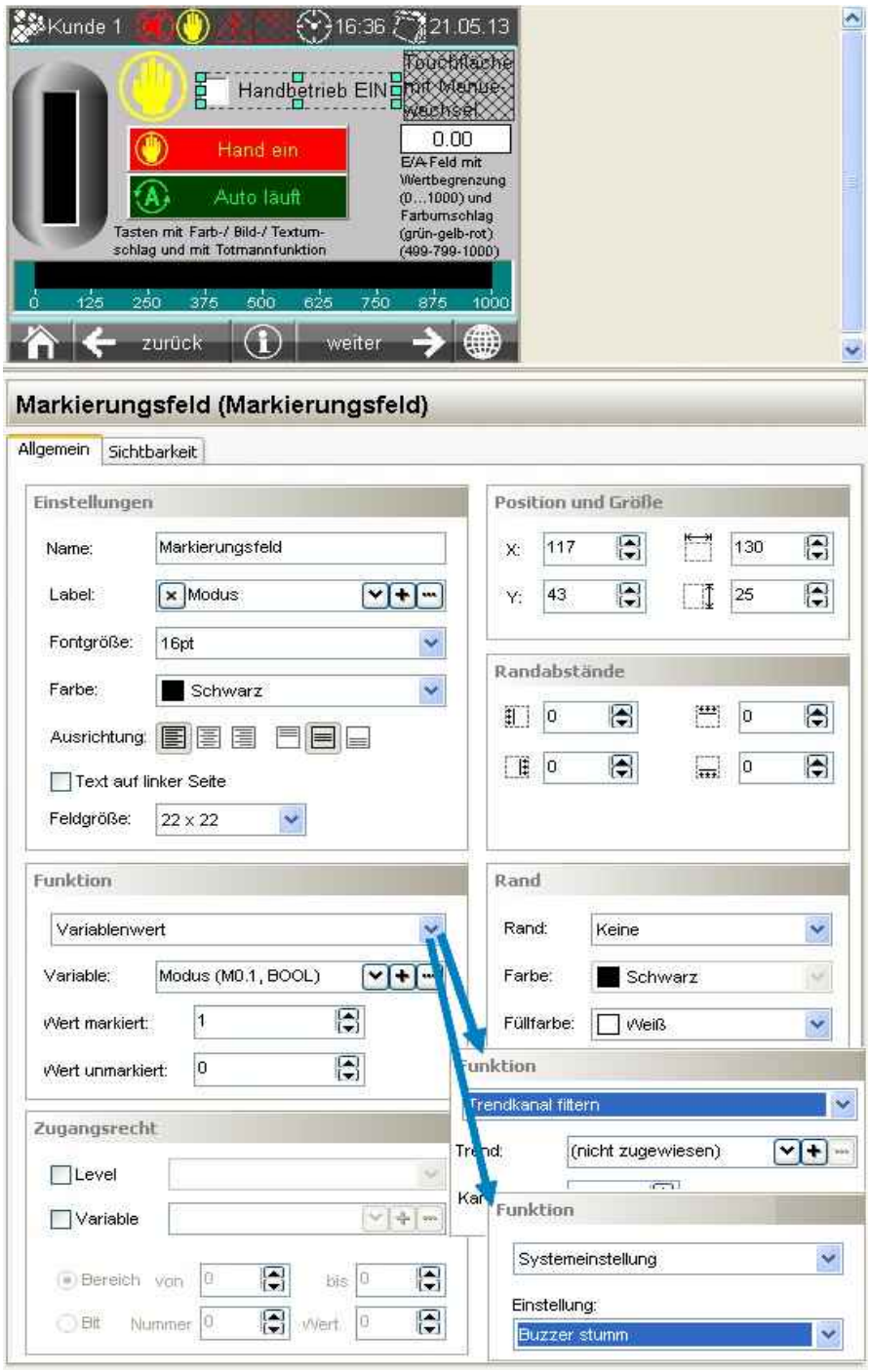
- Ausrichten der Anzeige zum Objektrand über Box **„Randabstände“**
- **Ausrichtung** des Textblocks im Feld (horizontal und vertikal)
- **Label** (Text) zuweisen oder neu erstellen für das Objekt
- **Rand** des Objektes einstellen (flach, versenkt, erhaben)
- **Feldgröße** der Auswahl- / Markierungsbox festlegen
- **Zugangsrechte** festlegen (mit den unter „Benutzer“ zugewiesenen Levels oder Variablen)
- **Sichtbarkeit** einstellen (wie bei „Allgemeine Funktionen“ beschrieben)

Funktion zuweisen

- **Variable** zuweisen oder neu erstellen
  (Wert für **MARKIERT** und für **UNMARKIERT** festlegen

  oder *(nur bei Markierungsfeld)*
- **Filter** festlegen für Trendkanal, der angezeigt werden soll

  oder *(nur bei Markierungsfeld)*
- **Systemeinstellung** für Buzzerfunktion (an/aus) vornehmen



Allgemeine Tips

- dynamische Objekte sollten sich **NICHT überlappen**. (Ausnahme: Touchflächen), da eine zustandsabhängige Änderung des Hintergrundes unter transparenten Flächen ignoriert wird. Es wird dafür bei dem Kompillieren eine Fehlermeldung ausgegeben.
- Simulation mit „ ► “ erzeugt einen Fehlerbericht, der direkt zum fehlerhaften Objekt führt.

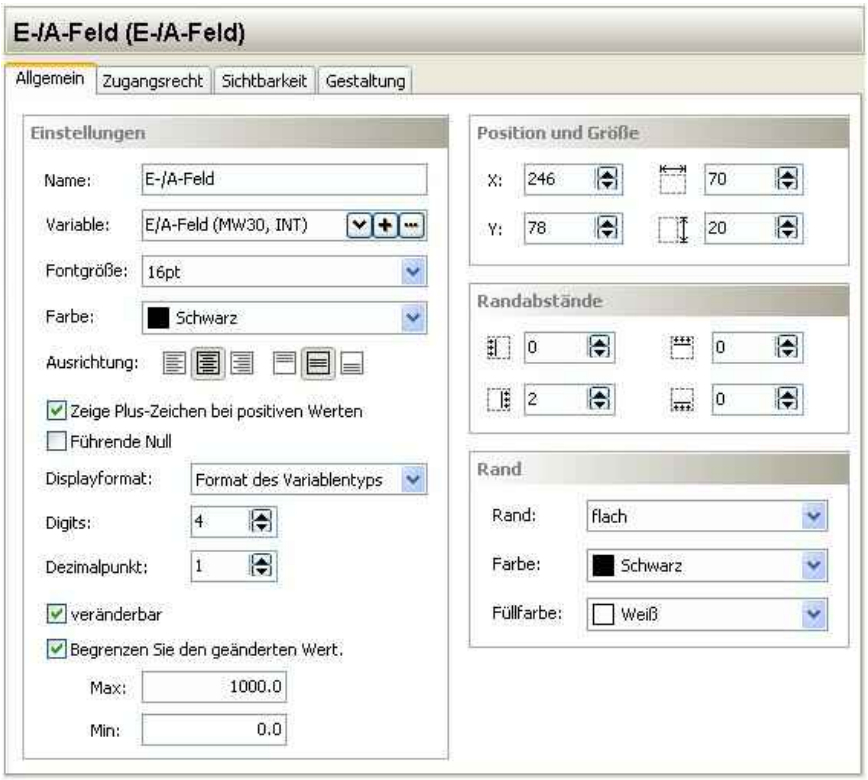
E/A-Felder

Diese Felder dienen als **Anzeige** und (wenn „veränderbar“ angeklickt ist) auch als **Eingabefeld** mit selbstständig eingeblendeter virtueller Systemtastatur (konfigurierbar unter Ressource „Dialoge“).

Objekt platzieren:
(siehe „Allgemeine Funktionen“)
Sichtbarkeit einstellen:
(siehe „Allgemeine Funktionen“)

Objekt parametrieren
(Reiter „Allgemein“)

- Ausrichten der Anzeige zum Objektrand über Box **„Randabstände“**
- **Ausrichtung** des Ziffernblocks im Feld (horizontal und vertikal)
- **Variable** zuweisen oder neu erstellen
- **Format** festlegen:
  - **E/A-Feld:** Displayformat: ob bei Eingabe z.b. die Kommataste benutzt werden soll oder ob einfach das Format des Variablentyps angezeigt werden soll
- **Begrenzen** des Eingabewertes (Minimum, Maximum)
  - im Wertebereich des jeweiligen Variablentyps



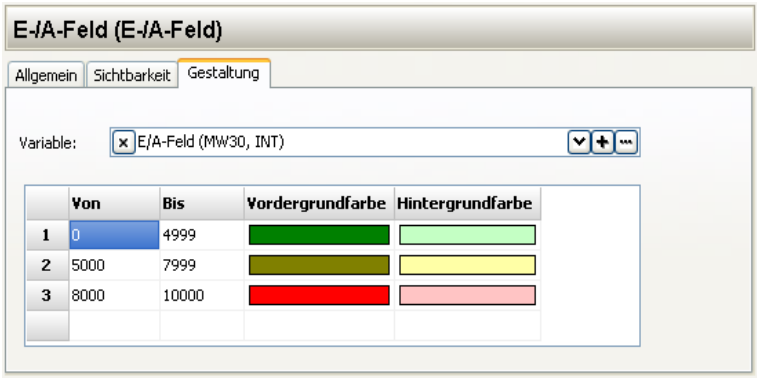
Achtung

- Bei **STRNG** die S7-Definitionen (Längenangabe in den ersten beiden Bytes) beachten! Bei INSEVIS ist die maximale STRNG-Länge 60 Byte ABER der DB dafür muss mindestens 60+2Byte groß sein!
- Bei **TOD** Anzeige der Zeit in hh:mm:ss, bei **TIME**, **S5TIME** und **TIMER** Anzeige der Zeit in ms (Millisekunden)
- Bei **DATE** Anzeige des Datums in Tagen (seit 01.01.1990).

Objekt parametrieren (Reiter „Zugangsrecht“)
Durch Markieren von **„veränderbar“** lässt sich der Wert später auch über Touch verändern, dafür kann man hier Zugangsrechte festlegen (mit den unter „Benutzer“ zugewiesenen Levels oder Variablen).

Objekt parametrieren (Reiter „Gestaltung“)
Hier können variablenabhägige Farben für Text und Hintergrund eingestellt werden:

- **Farbumschläge** (In Reiter „Gestaltung“) für Vordergrund (Ziffernfarbe) und Hintergrund (Farbe der Feldfläche)
- **Variablenangabe**, die den Farbumschlag steuert



Allgemeine Tips:

- Wenn ein E/A-Feld zu schmal ist, werden die Zeichen, die nicht mehr vollständig angezeigt werden können gar nicht mehr angezeigt!
- dynamische Objekte sollten sich **NICHT überlappen**. (Ausnahme: Touchflächen), da eine zustandsabhängige Änderung des Hintergrundes unter transparenten Flächen ignoriert wird. Es wird dafür bei dem Kompillieren eine Fehlermeldung ausgegeben.
- Simulation mit „ ► “ erzeugt Fehlerbericht, der direkt zum fehlerhaften Objekt führt.

- **Skalieren:** im S7-Programm mit dem Siemens-FC105 SCALE
  (Ergebnis REAL-Wert, der bei der VisuStage nicht zum Gestalten/Einfärben der E/A-Felder zulässig ist),
  Wenn Gestaltung/Einfärben gewünscht, dann Verwenden der S7-Befehle Abrunden mit RND oder Abschneiden mit TRUNC (Ergebnis: Ganzzahl-Wert, mit dem auch ein Einfärben der E/A-Felder zulässig ist)

Datumsfelder

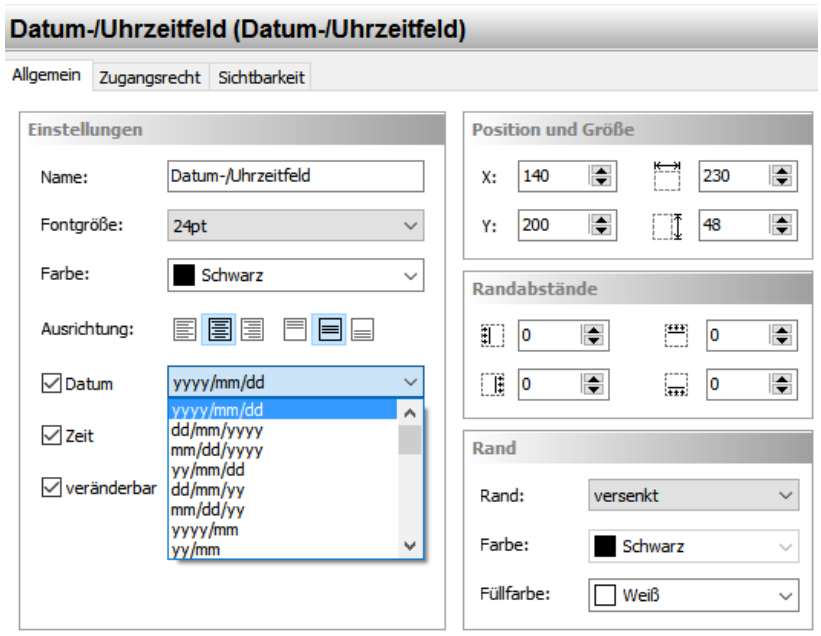
Diese Felder dienen als **Anzeige** und (wenn „veränderbar“ angeklickt ist) auch als **Eingabefeld** mit selbstständig eingeblendeter virtueller Systemtastatur (konfigurierbar unter Ressource „Dialoge“).

Objekt platzieren:
(siehe „Allgemeine Funktionen“)
Sichtbarkeit einstellen:
(siehe „Allgemeine Funktionen“)

Objekt parametrieren
(Reiter „Allgemein“)

- Ausrichten der Anzeige zum Objektrand über Box **„Randabstände“**
- **Ausrichtung** des Ziffernblocks im Feld (horizontal und vertikal)
- **Format** festlegen:
  - Zeitformat
  - Datumsformat
  - veränderbar

Reiter „Zugangsrecht“:
Durch Markieren von **„veränderbar“** lässt sich der Wert später auch über Touch verändern, dafür kann man hier Zugangsrechte festlegen (mit den unter „Benutzer“ zugewiesenen Levels oder Variablen).



Allgemeine Tips:

- Wenn ein Datumsfeld zu schmal ist, werden die Zeichen, die nicht mehr vollständig angezeigt werden können gar nicht mehr angezeigt!
- dynamische Objekte sollten sich **NICHT überlappen**. (Ausnahme: Touchflächen), da eine zustandsabhängige Änderung des Hintergrundes unter transparenten Flächen ignoriert wird. Es wird dafür bei dem Kompillieren eine Fehlermeldung ausgegeben.
- Simulation mit „ ► “ erzeugt Fehlerbericht, der direkt zum fehlerhaften Objekt führt.

Fortschrittsbalken

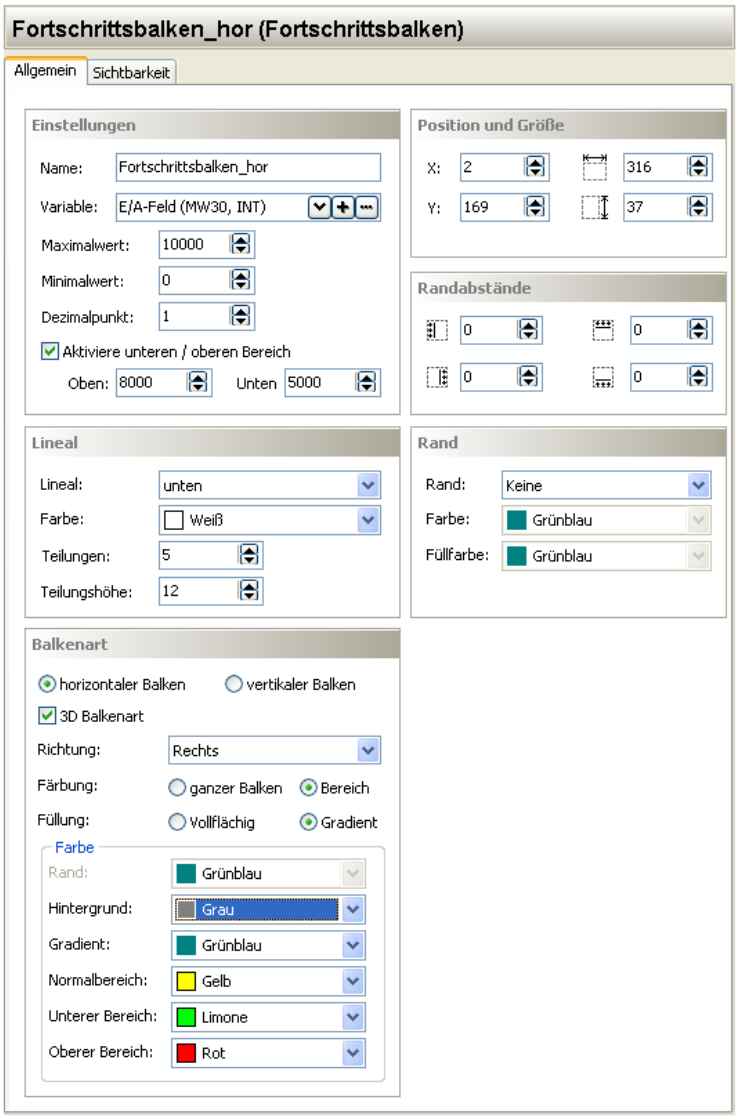
Ideal, um Füllstände oder andere Werte in analog einem Balken anzuzeigen, der verschiedene Farbbereiche beinhalten kann.

Dieser Balken wurde mit den folgenden Einstellungen parametriert:

Objekt platzieren:
(siehe „Allgemeine Funktionen“)
Sichtbarkeit einstellen:
(siehe „Allgemeine Funktionen“)

Objekt parametrieren (Reiter Allgemein)

- Ausrichten des Balkens zum Objektrand über Box **„Randabstände“**
- **Rand** einstellen (flach, versenkt, erhaben)
- **Variable** zuweisen oder neu erstellen, Begrenzungen vornehmen von
  - Minimalwert,
  - Maximalwert,
  - Dezimalpunkt
- Bei INTEGER-Werten ist der Wert ohne Dezimalpunkt einzugeben.
  Hier: 10000 = 1000.0
- **Bereichstrennung** unteren/oberen Bereich anwählen und Grenzwerte eingeben
- **Lineal:** Position, Farbe und Teilung der Skala, Höhe der Bezeichnung festlegen
  → Bei ungünstigen Teilungen sind Rundungsfehler möglich!
- **Balkenart** auswählen
  - horizontal, vertikal oder 3D
- Laufrichtung des Balkens festlegen
- Färbung „ganzer Balken“: Balken wird komplett in Bereichsfarbe dargestellt
- Färbung „Bereich“: Balken wird in 3 Bereichen farblich dargestellt:
  - unterer / normaler / oberer Bereich



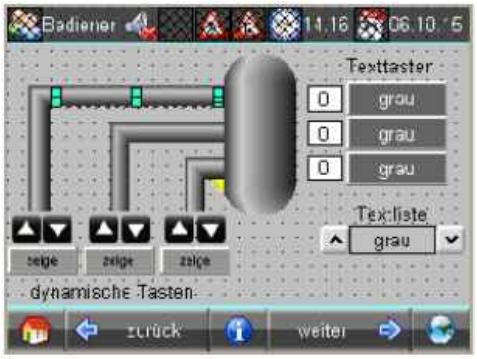
Allgemeine Tips

- dynamische Objekte sollten sich **NICHT überlappen**. (Ausnahme: Touchflächen), da eine zustandsabhängige Änderung des Hintergrundes unter transparenten Flächen ignoriert wird. Es wird dafür bei dem Kompillieren eine Fehlermeldung ausgegeben.
- Simulation mit „ ► “ erzeugt Fehlerbericht, der direkt zum fehlerhaften Objekt führt.

Betrachter

Bild- und Textlisten

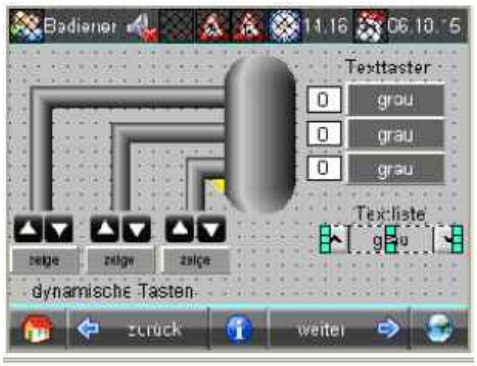
Bildlistenbeispiel: Der Farbumschlag bei Rohren. Die Rohre werden Segmente geteilt, die jeweils in allen Farben als png zur Verfügung stehen. In der Ressource „Bildliste“ wurden den Farben Werte zugewiesen. In dem Objekt Bildliste werden diese Werte durch eine Variable gesteuert (die selbst durch Pfeiltasten je um den Wert „1“ verändert wird und die in einem E/A-Feld ausgegeben wird).



Objekt platzieren: (siehe „Allgemeine Funktionen“)
Sichtbarkeit einstellen: (siehe „Allgemeine Funktionen“)

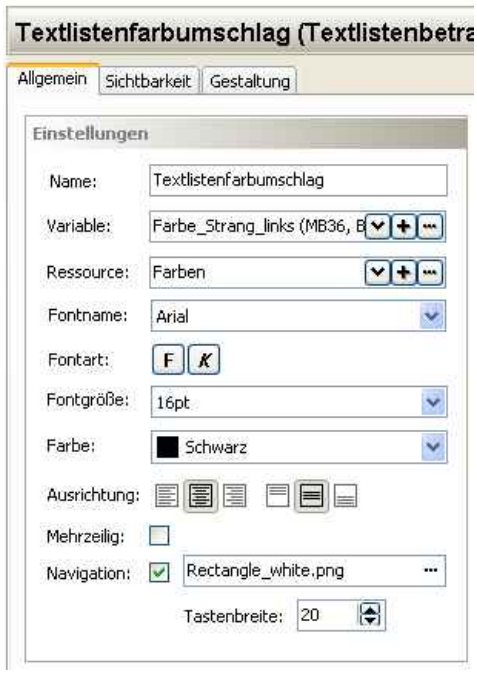
Objekt parametrieren: (Reiter „Allgemein“)
- **Strecken: „an Größe anpassen“** erleichtert das Platzieren, weil das Bild das aufgezogene Objekt voll ausfüllt
- **Ausrichtung** des Objektes (wenn bei Strecken „proportional“ oder „keine“ gewählt wurde
- **Spiegeln** und **Rotieren** wenn nötig
- **Ressource** Bildliste zuweisen (muss schon erstellt sein)
- **Variable** festlegen, die den Wert der Bildliste steuern soll

Textlistenbeispiel: Die Angabe des aktuell eingeloggten Benutzers. Hier wurden 3 Benutzer (Benutzer 1, .._2, .._3) festgelegt. Hier sind ab Version 2.0.2.3 variablenabhängige Farbumschläge möglich.



Objekt platzieren: (siehe „Allgemeine Funktionen“)
Sichtbarkeit einstellen: (siehe „Allgemeine Funktionen“)

Objekt parametrieren: (Reiter „Allgemein“)
- **Variable** festlegen, die den Wert der Bildliste steuern soll
- **Ressource** Bildliste zuweisen (muss schon erstellt sein)
- Text **formatieren** (Font, Größe, Fontart, Farbe)
- Bei Bedarf „**Mehrzeilig**“ aktivieren
- Art und Breite der **Scroll-Tasten** festlegen



Zusatzfunktionen bei Textlistenbetrachter

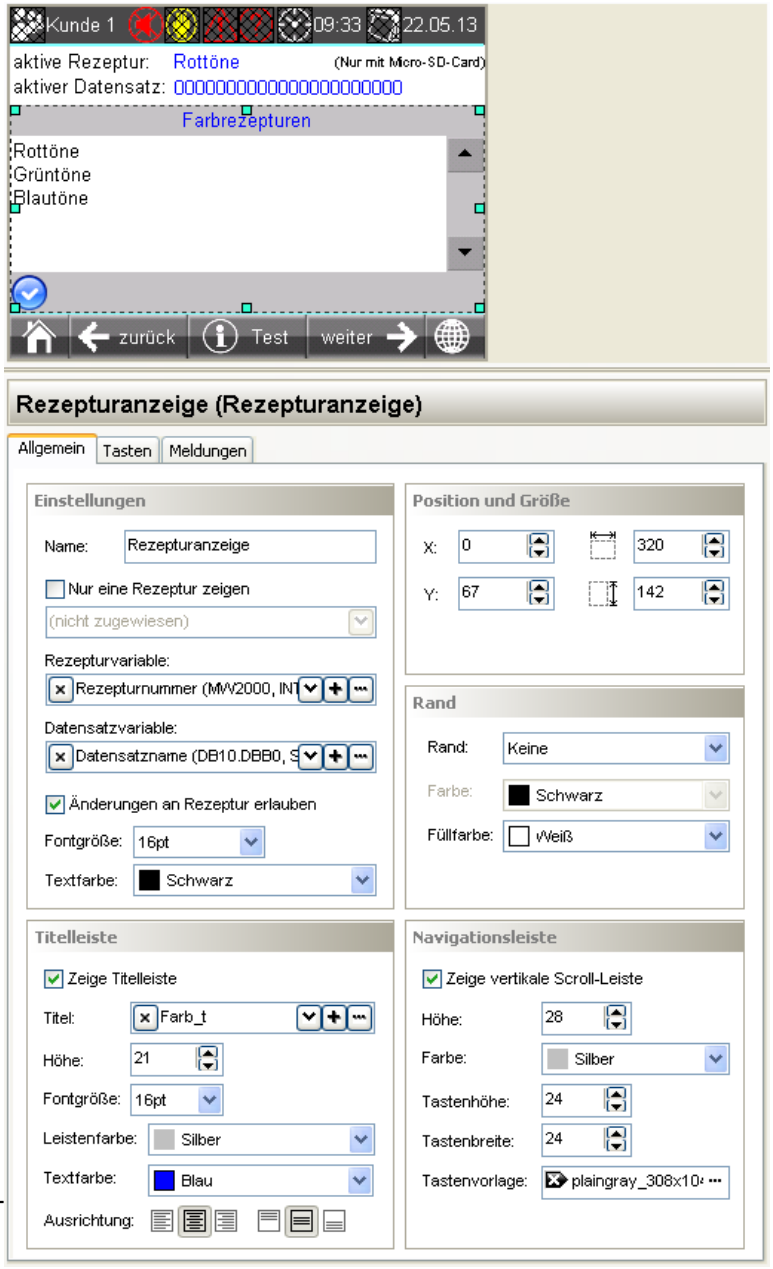
Wenn eine mehrzeilige Textliste verwendet wird, gibt es die Möglichkeit, eigene Navigations-Tasten zum Scrollen (auf/ab) festzulegen (links) und variablenabhängige Farbumschläge zu konfigurieren (unten)

Rezepturanzeige

Projektierung des Rezepturbildes:
- Auswahl,
- Positionierung und
- Randeinstellungen

wie bei allen INSEVIS-Objekten

Es wird zuerst die Rezepturliste im Reiter **„Allgemein“** angezeigt, über den Reiter „**Tasten**“ gelangt man zu den anderen Ansichten.



Allgemeine Einstellungen

Name: interner (projektsprachenunabhängiger) Name der Rezepturanzeige

Nur eine Rezeptur zeigen: beschränkt Ansicht auf eine Rezeptur (Wenn z.B. diese Maschine für nur eine Rezeptur ausgelegt wurde, aber das Projekt mehrere Rezepturen umfasst)

Änderungen an Rezeptur erlauben: Erlaubt dem Benutzer, die vorab festgelegten Werte zu ändern, neue Datensätze anzulegen, oder bestehende umzubenennen oder zu löschen.

Der aktuell benutzte Datensatz wird in der Textfarbe der Titelleiste angezeigt

Die **aktive Rezeptur** lässt sich über eine Textliste anzeigen, die die einzelnen Rezepturnamen beinhaltet und von der Rezepturvariable gesteuert wird.

Der **aktive Datensatz** lässt sich anzeigen, wenn die hier festgelegte Datensatzvariable als STRING angelegt und in einem separatem E/A-Feld angezeigt wird.

Hinweis zur Simulation
Bei Simulation des aktiven Datensatzes manuell in die Variable einmalig einen Wert manuell eintragen, damit die Simulation erfolgen kann, dann Taste drücken „in SPS schreiben“)

Parametrierung der Titelleiste (allgemein für Rezepturlisten, Datensatzlisten, Elementlisten)

Es empfiehlt sich, eine Titelleiste zu benutzen, und einen Titel für die Rezepturliste in der Visualisierung festzulegen.

Die folgenden Titel der darunter liegenden Listen werden automatisch von den angelegten Daten erzeugt
(Datensatzliste im Beispiel „Rottöne“, Elementliste im Beispiel Rosa“)

Die restlichen Einstellungen sind selbsterklärende Einstellungen zum Design der Titelleiste.

Parametrierung der Scroll-Leiste (allgemein für Rezepturlisten, Datensatzlisten, Elementlisten)

Die Aktivierung der Scroll-Leiste empfiehlt sich, wenn mehr Rezepturen oder Datensätze oder Elemente benutzt werden, als bei der gewählten Schriftgröße in das Anzeigefenster passen.

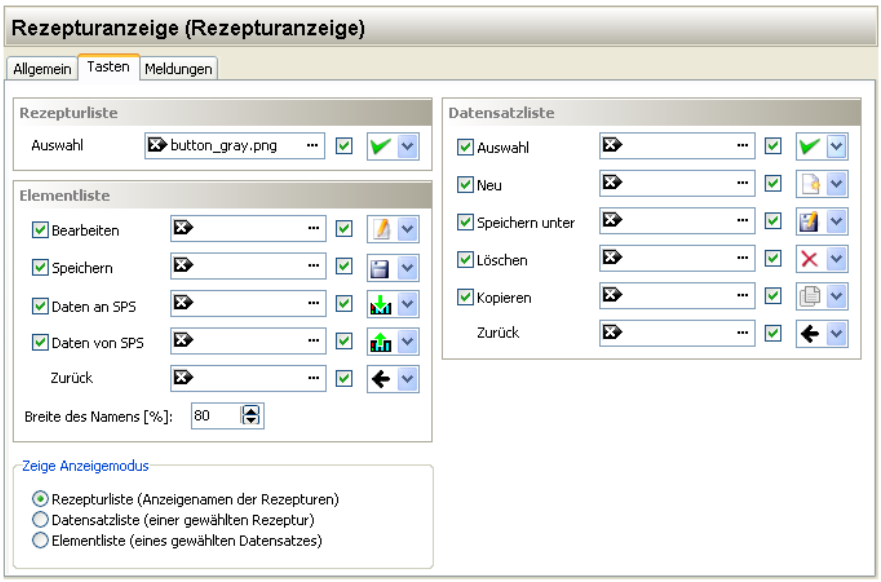
Die Einstellungen sind selbsterklärende Einstellungen zum Design der Scroll-Leiste.

Parametrierung der Bedienelemente (allgemein für Rezepturlisten, Datensatzlisten, Elementlisten)

Unter dem Reiter „Tasten“ werden die nötigen Bedienelemente ausgewählt und können kunden-spezifische Tasten zugewiesen bekommen.

Die Breite des Namens in der Anzeige ist zur besseren Bildschirmeinteilung frei skalierbar.

Im unteren Bereich wird ausgewählt, welche Liste in der aktuellen Ansicht in der VisuStage gezeigt wird.



Rezepturliste:

Auswahl:	aktiviert eine Rezeptur aus der Liste (auch durch Berühren direkt möglich)

Elementliste:

Bearbeiten:	öffnet eine virtuelle Tastatur, um den Wert des Elementes zu ändern
Speichern:	überschreibt den bestehenden Datensatz in der Micro-SD-Card mit den aktuellen Werten
Daten an SPS:	übertragt die gespeicherten Werte und Datensätze an die Steuerung
Daten von SPS:	liest die Variablenwerte die die Elemente aus der Steuerung
Zurück:	Kehr in übergeordnete Liste (Datensatzliste) zurück

Datensatzliste:

Auswahl:	aktiviert einen Datensatz aus der Liste (auch durch Berühren direkt möglich)
Neu:	Erzeugt einen neuen Datensatz mit bestehender Elementstruktur
Speichern unter:	speichert den bestehenden Datensatz in der Micro-SD-Card unter neuen Namen ab (ein Stringeingabefeld öffnet sich dabei automatisch)
Löschen:	Löscht einen bestehenden Datensatz
Kopieren:	Kopiert einen bestehenden Datensatz
Zurück:	Kehr in übergeordnete Liste (Rezepturliste) zurück

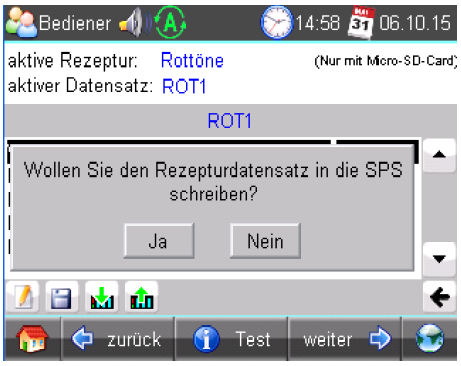
Auswahl von Bestätigungsmeldungen

Es ist in den verschiedenen Listenansichten (Rezeptur- / Datensatz- / Element-) sinnvoll, vor dem Überschreiben eine Abfrage auszuführen, die unter dem Reiter „Meldungen“ konfiguriert werden können. Beim Textexport stehen diese Texte als Systemtexte immer ganz oben in der csv-Datei.

Diese Meldungen werden VOR dem Ausführen

- von Speicher- oder Löschvorgängen sowie
- Datenübertragungen von/zur SPS

erstellt.



Damit soll vor unabsichtliche Betätigungen gewarnt werden.

Visualisierung mit der Software „VisuStage“

Ansichten von Rezepturlisten, Datensatzlisten, Elementlisten zur Laufzeit

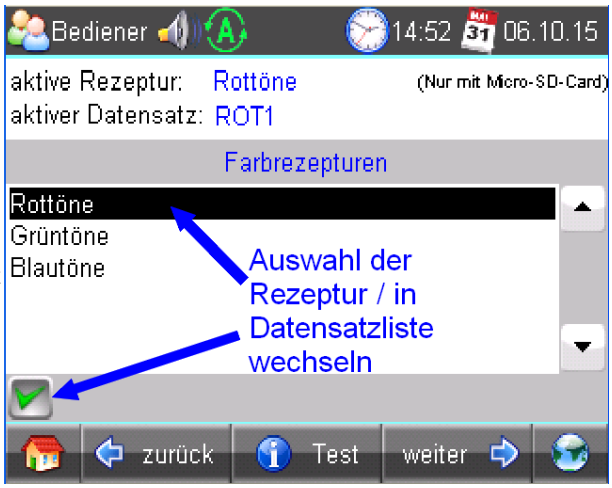
Rezepturliste

Die Rezepturliste beinhaltet die in der VisuStage angelegten Rezepte mit einer festgelegten Struktur von ebenfalls vorab festgelegter Elemente.

Der vorab zugewiesene Name für die ganze Rezepturliste wird in der Titelleiste angezeigt.

Das Aktivieren der Rezepturen erfolgt über die Touchbedienung und die Taste unten links, damit wird in das nächste Fenster mit den Datensätzen gewechselt.

Man verlässt die Rezepturverwaltung mit einem eigenen Button außerhalb der Rezepturliste mit einer selbst zu definierenden Taste mit der Funktion „Menue wechseln“.



Datensatzliste

Die Datensatzliste beinhaltet die vom Anwender angelegten Datensätze innerhalb einer Rezeptur (in diesem Fall: Rezeptur „Rottöne“).

Der betreffende Rezepturname wird oben in der Titelleiste angezeigt. Die Auswahl des zu ändernden Datensatzes erfolgt über die Touchbedienung und die Taste unten links, damit wird in das nächste Fenster mit der Elementtabelle gewechselt, um dort die Werte zu ändern und zu speichern. Dabei öffnet sich automatisch ein Stringeingabefeld.

Neu ab VS 2.0.1.5: Der aktuelle Datensatz wird in der Textfarbe der Titelleiste angezeigt.

Links unten sind die eingestellten Funktionstasten angeordnet:
Ändern | Neu erstellen | Speichern | Löschen
Rechts unten gelangt man mit „←“ wieder in die Rezepturliste.

Hinweis:
Wegen der Ablage der Datensätze auf der Micro-SD-Card der SPS/des Panels ist diese Anzeige nicht zu simulieren, weil sie am PC stattfindet und dieser keinen Micro-SD-Slot hat.

Elementliste

Die Elementliste beinhaltet die vom Anwender eingegebenen Werte für die Elemente eines bestimmten Datensatzes (in diesem Fall des Datensatzes „ROSA_1“).

Der betreffende Datensatzname wird oben in der Titelleiste angezeigt.

Bei Berühren des Wertes öffnet sich ein numerisches Eingabefeld zur Eingabe des neuen Wertes.

Links unten sind die eingestellten Funktionstasten angeordnet:
Neu erstellen | Speichern | von SPS laden | in SPS schreiben
Rechts unten gelangt man mit „←“ wieder in die Datensatzliste.

Hinweis:
Wegen der Ablage der Datensätze auf der Micro-SD-Card der SPS/ des Panels ist diese Anzeige nicht zu simulieren, weil sie am PC stattfindet und dieser keinen Micro-SD-Slot hat.

Meldungsanzeige und -archivierung

Es gibt verschiedene Objekte zum Anzeigen vorhandener Meldungen:
- blinkende Symbole (Meldehinweis Symbol)
- blinkende Textzeilen (Meldehinweis Text)
- älteste/jüngste Meldungszeile (Meldeanzeige Zeile)
- Anzeigebox (Viewer) für aktuelle Meldungen (Meldeanzeige Mehrzeilig)
- Anzeigebox für Meldungsarchive (Meldearchivanzeige)

Meldehinweis Symbol und Text

Diese dynamischen Objekte dienen dazu, durch **Anzeige** und ggf. **Blinken** auf anstehende Meldungen (Alarme/ Ereignisse) aufmerksam zu machen, mit einer **Textzeile** oder einem **Symbol** (dem Indikator).
- **Blinkt**, wenn , solange der Alarm oder das Ereignis kommt **(K),**
- **Steht still**, wenn der Alarm quittiert wurde **(Q),**
- **Erlischt**, wenn das Ereignis oder der Alarm (nach dem Quittieren) Gegangen ist **(G).**

Indikator bei Alarmen:

Blinken	wenn Alarm gekommen und nicht quittiert, ist
Anzeige ohne Blinken	wenn quittiert ist, aber noch anstehend
Erlischt	wenn quittiert und gegangen ist

Indikator bei Meldungen:

Blinken	wenn gekommen ist
Erlischt	wenn gegangen ist

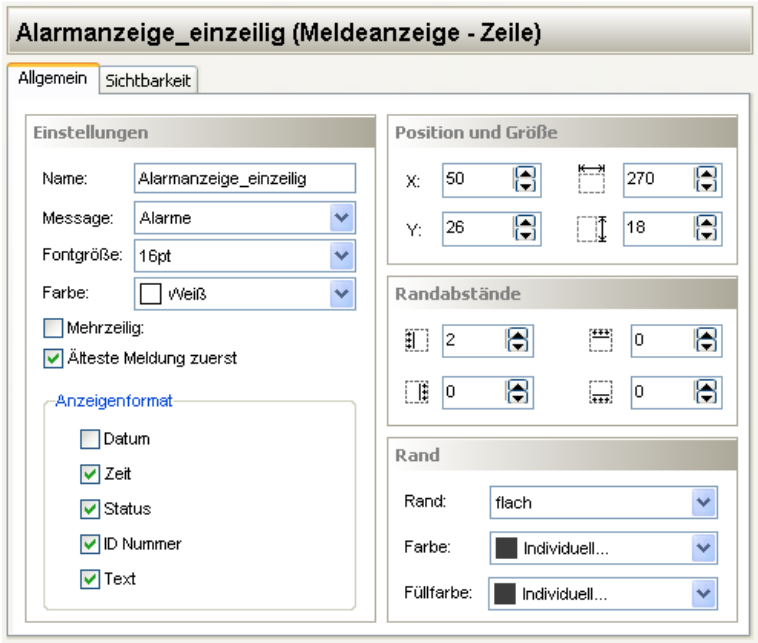
Die **Platzierung** wie bei allen anderen Objekten, die **Parametrierung** ist selbsterklärend.

Meldeanzeige - Zeile

Diese Meldeanzeige ist dafür gedacht, nur die aktuellste anstehende Meldung **anzuzeigen**, quasi als Status, ohne direkte Quittiermöglichkeit.

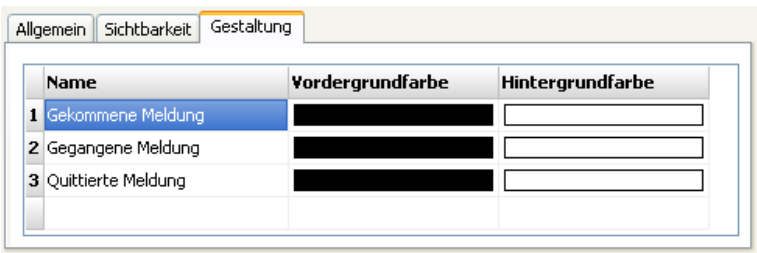
Objekt platzieren:
(siehe „Allgemeine Funktionen“)
Sichtbarkeit einstellen:
(siehe „Allgemeine Funktionen“)

Objekt parametrieren (Reiter „Allgemein“):
- **Meldungstyp:**- Alarm oder Ereignis
- **Zeilenformat:** - ein-/ mehrzeilig
- **Sortierung:** neueste/ älteste Meldung zuerst
- **Meldezeile** formatieren:
  - Anzeige vom Datum
  - Anzeige von Zeit
  - Anzeige vom Meldestatus
  - Anzeige von der ID-Nr. (Gruppe)
  - Anzeige von Meldetext

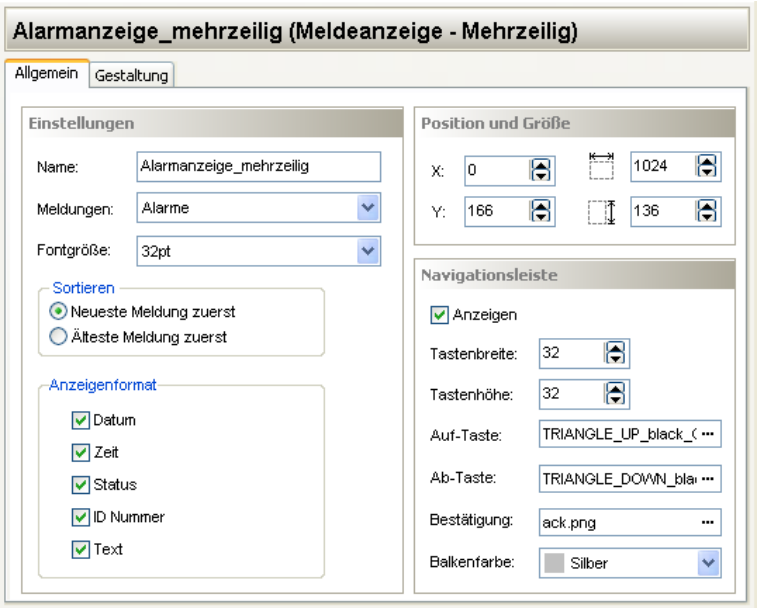


Objekt parametrieren (Reiter „Gestaltung“):

Ab VisuStage Version 2.2.5 und Firmware 1.1.9 (HMI) und 2.2.3 (Panel-SPS) können bei den Meldeanzeigen die einzelnen Meldungen nach Status (G,K,Q) farblich angepasst werden. Ab RemoteStage Version 1.0.4.10 auch remote. In diesem Fall alles schwarzer Text auf weißem Hintergrund.



Meldeanzeige - Mehrzeilig



Diese Meldeanzeige ist dafür gedacht, um **alle anstehenden** Meldungen **anzuzeigen** und besitzt Navigierfunktionen (zeilenweises **Scrollen**) und eine **Quittierfunktion**.

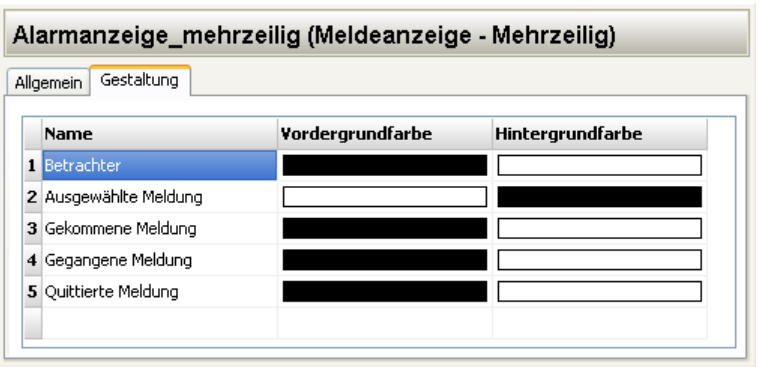
Platzierung wie bei allen anderen Objekten

Meldungstyp Alarm oder Ereignis

Sortierung neueste/älteste Meldung zuerst

Meldezeile formatieren:
- Anzeige vom Datum
- Anzeige von Zeit
- Anzeige vom Meldestatus
- Anzeige von der ID-Nr. (Gruppe)
- Anzeige von Meldetext

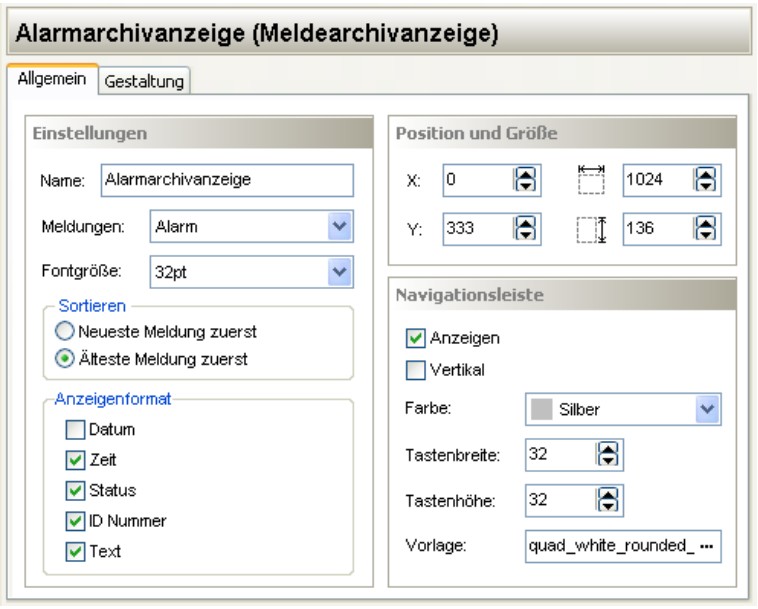
Navigationsleiste formatieren:
- An- und Abschalten der Leiste
- Verwendung eigener Tastendesigns
- Zuweisen einer Balkenfarbe



Gestaltung
Ab VisuStage Version 2.2.5 und Firmware 1.1.9 (HMI) und 2.2.3 (Panel-SPS) können bei den Meldeanzeigen die einzelnen Meldungen nach Status (G,K,Q) farblich angepasst werden. Ab RemoteStage Version 1.0.4.10 auch remote.

In diesem Fall alles schwarzer Text auf weißem Hintergrund.

Meldearchivanzeige



Diese Meldeanzeige ist dafür gedacht, um **alle archivierten** Meldungen **anzuzeigen** und besitzt Navigationsfunktionen (zeilen- und seitenweises **Scrollen)**.

Platzierung wie bei allen anderen Objekten

Meldungstyp: Alarm oder Ereignis

Sortierung: neueste/älteste Meldung zuerst

Meldezeile formatieren:
- Anzeige vom Datum
- Anzeige von Zeit
- Anzeige vom Meldestatus
- Anzeige von der ID-Nr. (Gruppe)
- Anzeige von Meldetext

Navigationsleiste formatieren:
- An- und Abschalten der Leiste
- Verwendung eigener Tastendesigns
- Zuweisen einer Balkenfarbe

Gestaltung
Wie bei Meldeanzeige - Mehrzeilig, nur ohne gestaltbare „ausgewählte Meldung“.

Allgemeine Tips

- dynamische Objekte sollten sich **NICHT überlappen**. (Ausnahme: Touchflächen), da eine zustandsabhängige Änderung des Hintergrundes unter transparenten Flächen ignoriert wird. Es wird dafür bei dem Kompilieren eine Fehlermeldung ausgegeben.
- Simulation mit „ ► “ erzeugt Fehlerbericht, der direkt zum fehlerhaften Objekt führt.

Visualisierung mit der Software „VisuStage“

Trendanzeige und Trendarchivanzeige

Zeigt den Verlauf von bis zu 16 Kanälen innerhalb eines Trends in Abhängigkeit von der Zeit grafisch an.

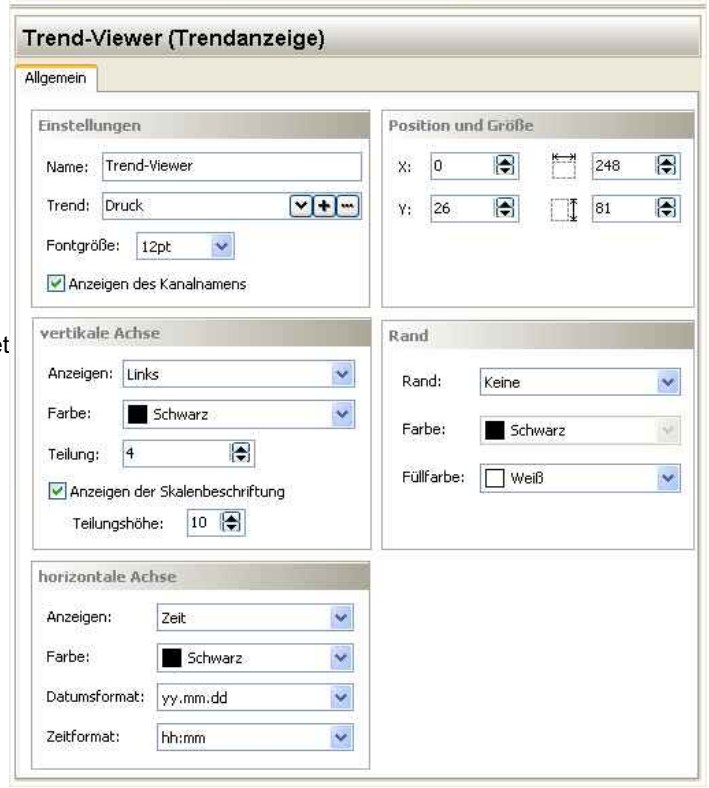
Objekt platzieren:
(siehe „Allgemeine Funktionen“)
Sichtbarkeit einstellen:
(siehe „Allgemeine Funktionen“)

Objekt parametrieren (Reiter „Allgemein“):

Trend-Ressource zuweisen
- Anzeige des **Kanalnamens** auswählen
- **Fontgröße** auswählen
- **Rand** des Objektes einstellen

- **vertikale Achse:** Lineal definieren
  - An/Aus und Anzahl / Größe der Teilungen

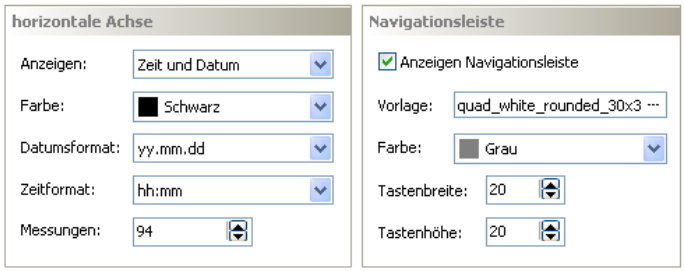
- **horizontale Achsen** Zeitformat definieren
  Nur Trendarchivanzeige:
  - **Anzahl** der anzuzeigenden Messungen angeben *
  - **Navigationsleiste** definieren (zum vorwärts/ rückwärts-scrollen)



Hinweis zur Zeitachse (X-Achse)
Weil die Zeiten auf der X-Achse zur Laufzeit errechnet und dargestellt werden, werden immer nur
- die Zeit der ersten Messung (ganz links) und die
- die Zeit der letzten Messung (ganz rechts)
dargestellt. (Und dazwischen nichts weil es sich überschreiben würde).

Nachdem die Messkurve den rechten Rand erreicht hat, wird die Kurve um ca. 10% nach links versetzt, bis sie wieder den rechten Rand erreicht hat.
(Und die unten angezeigten Zeiten ändern sich)

Unterschied bei Trendarchivanzeige:



* Beim Trendarchiv kann die Anzahl der Werte auf der horizontalen (Zeit-)-Achse vorgegeben werden. Damit wird die Auflösung (=Messwerte/Pixel) definiert.

Ebenso erscheint die Box zur Konfiguration der Navigationsleiste

Allgemeine Tips

- Trends können nur archiviert werden, wenn eine **Micro SD-Karte** steckt
- Es wird immer nur EINE vertikale Trendskala mit zugehörigem Trendnamen darüber angezeigt. Durch Antippen des Namens wird die Skala und der Name auf den nächsten Trend umgeschaltet, bis der erste Trend wieder erscheint, usw..
- dynamische Objekte sollten sich **NICHT überlappen**.
- Simulation mit „ ► “ erzeugt Fehlerbericht, der direkt zum fehlerhaften Objekt führt.

Funktionskurve

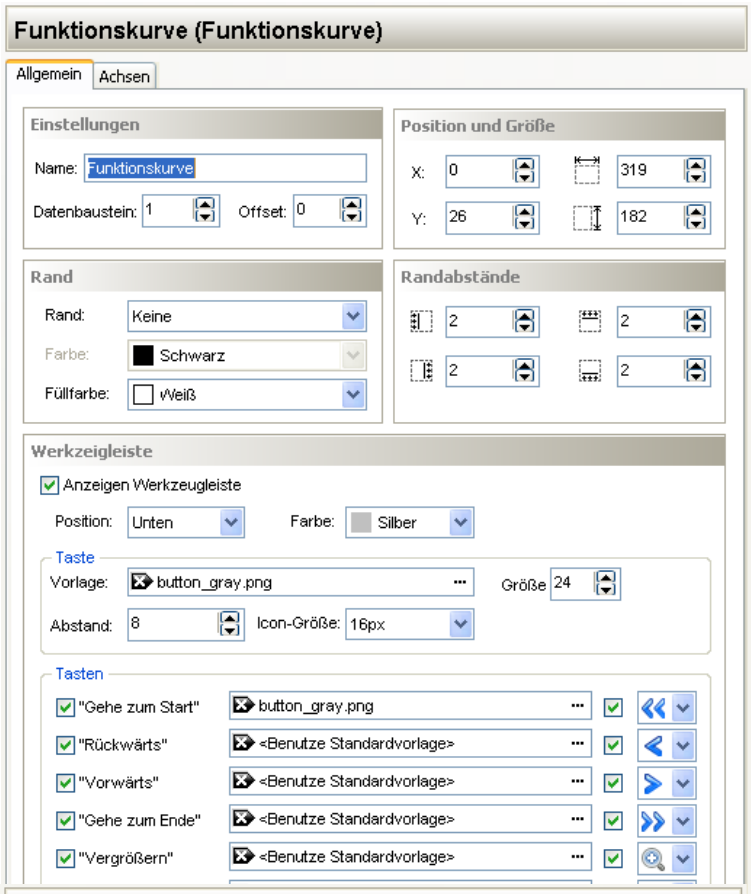
Diese Anzeige ist für die grafische Darstellung von zwei Werten aus einem Datenbaustein gedacht. Es können bis zu 2000 Wertepaare dargestellt werden, jedes wird als ein Pixel dargestellt.

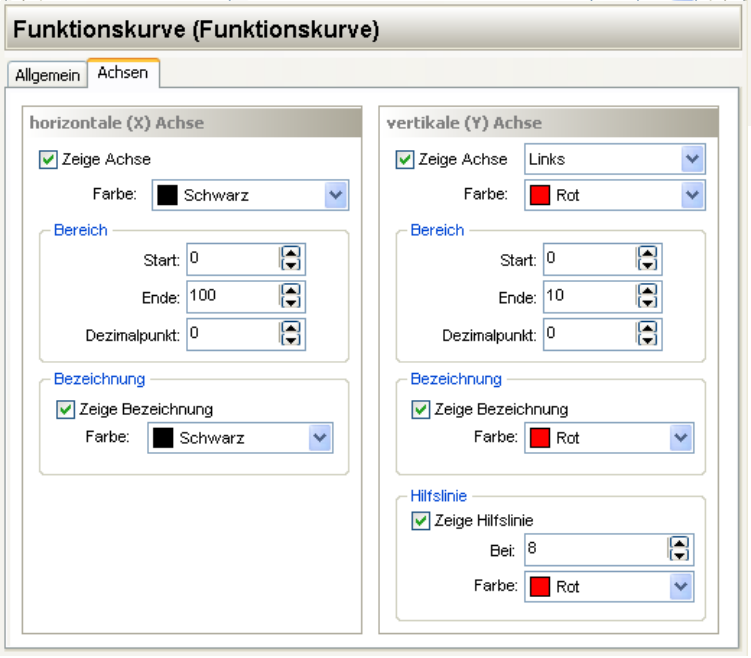
Objekt platzieren:
(siehe „Allgemeine Funktionen“)
Sichtbarkeit einstellen:
(siehe „Allgemeine Funktionen“)

Objekt parametrieren (Reiter „Allgemein“)

- **Datenbaustein und Offset zuweisen**
  Dieser Datenbaustein enthält die anzuzeigenden Werte
- **Rand** des Objektes einstellen (flach, versenkt, erhaben)
- **Randabstände** der Kurve zum Rand des Objektes einstellen
- **Werkzeugleiste** anpassen:
  - Position der Navigationsleiste
  - Abstand der Navigationstasten
  - Auswahl der Navigationstasten (aus allen verfügbaren Tasten, auch eigene können angelegt werden, siehe Kapitel *diverse Tastenarten)*
  - Auswahl der Tastensymbole (aus bestehenden Symbolen)

Objekt parametrieren (Reiter „Achsen“
- **Skalenwerte** zuweisen
  - Startwert
  - Endwert
  - Dezimalpunkt
- **Skalenfarbe** zuweisen
- **Hilfslinie** zuweisen
  (eine Hilfslinie für einen Y-Wert möglich)





Allgemeine Tips:

Achtung

- dynamische Objekte sollten sich **NICHT überlappen**.
- Simulation mit „ ► “ erzeugt Fehlerbericht, der direkt zum fehlerhaften Objekt führt.

Visualisierung mit der Software „VisuStage“

Der für die Funktionskurve benutzte Datenbaustein muss aus zwei Blöcken (Kopf/Header und Daten/Data) bestehen

- Header Information über die Koordinatenstruktur
- Data Array von [X,Y]-Paaren. Der Datentyp muss im Header definiert werden.

Kopf/Header-Struktur

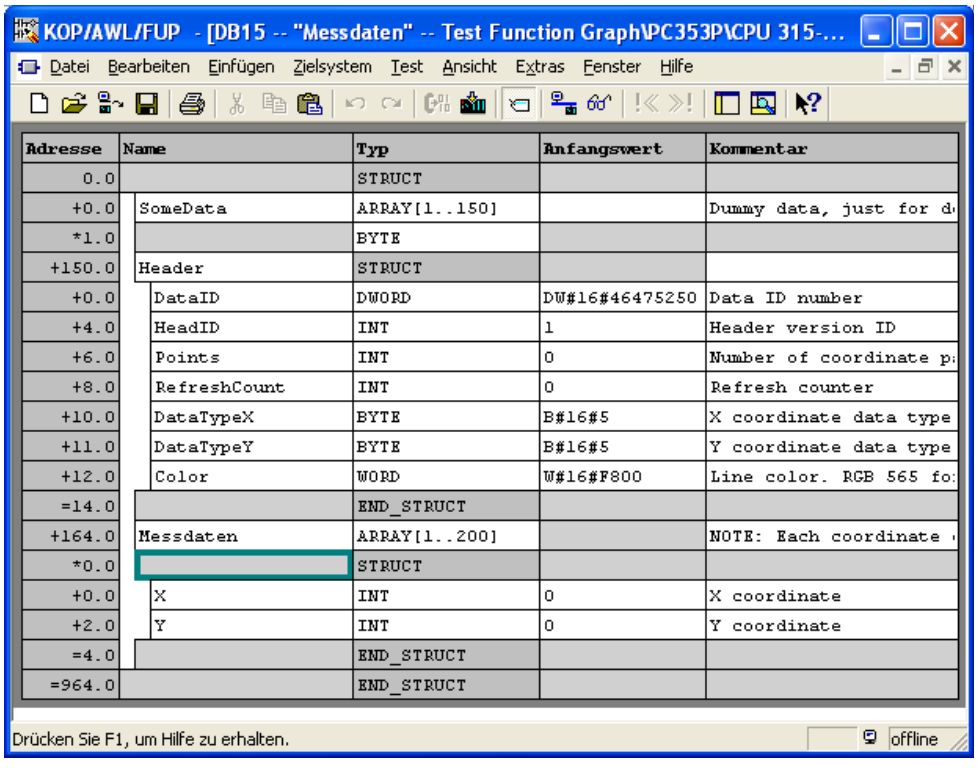
Offset	Name	Datentyp	Wert	Bemerkungen
0	DataID	DWORD	DW#16#46475250	Konstanter Wert
4	HeadID	INT	1	Konstanter Wert
6	Points	INT		Anzahl von [X,Y]-Paaren [0 .. 200]
8	RefreshCount	INT		Refresh request counter S7-Anwenderprogramm muss den Wert inkrementieren, um einen Refresh in der VisuStage zu erzwingen.
10	DataTypeX	BYTE		Datentyp des X-Koordinatenwertes 2 = BYTE 3 = CHAR 4 = WORD 5 = INT 6 = DWORD 7 = DINT 8 = REAL
11	DataTypeY	BYTE		Datentyp des Y-Koordinatenwertes 2 = BYTE 3 = CHAR 4 = WORD 5 = INT 6 = DWORD 7 = DINT 8 = REAL
12	Color	WORD		RGB Farbwert, Format 565 Bit0..Bit4 blaue Komponente Bit5..Bit10: grüne Komponente Bit11..Bit15: rote Komponente

DB Datenstruktur

Die Struktur des Array von [X,Y]-Paaren ist kundenspezifisch und durch den Anwender selbst komplett in der Headerstruktur zu definieren.

Folgender DB beinhaltet Aufzeichnungskoordinaten mit folgenden Eigenschaften:

- jeden Koordinatenwert (16 Bit Integer, Wertebereich von -32768 .. 32767)
- 200 Koordinatenpaare
- Linienfarbe ist blau



Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	SomeData	ARRAY[1..150]		Dummy data, just for d
*1.0		BYTE		
+150.0	Header	STRUCT		
+0.0	DataID	DWORD	DW#16#46475250	Data ID number
+4.0	HeadID	INT	1	Header version ID
+6.0	Points	INT	0	Number of coordinate p
+8.0	RefreshCount	INT	0	Refresh counter
+10.0	DataTypeX	BYTE	B#16#5	X coordinate data type
+11.0	DataTypeY	BYTE	B#16#5	Y coordinate data type
+12.0	Color	WORD	W#16#F800	Line color. RGB 565 fo
=14.0		END_STRUCT		
+164.0	Messdaten	ARRAY[1..200]		NOTE: Each coordinate
*0.0		STRUCT		
+0.0	X	INT	0	X coordinate
+2.0	Y	INT	0	Y coordinate
=4.0		END_STRUCT		
=964.0		END_STRUCT		

Systemdaten (IP-Adresse, Sprachen etc.)

Während Sprachen, Helligkeit, Putzbildschirm oder Buzzereinstellungen Funktionen sind, die Tasten oder Touchflächen zugewiesen werden können, ist die Änderung der IP-Adresse ein eigenes Objekt.

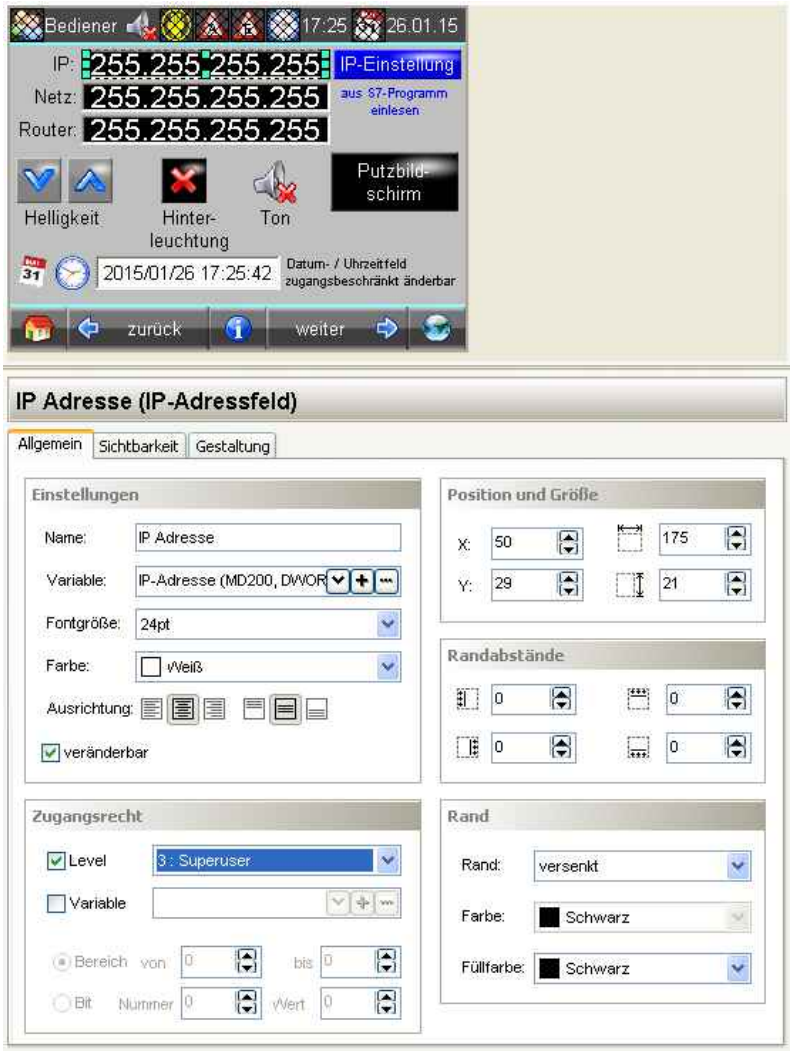
Objekt platzieren:
(siehe „Allgemeine Funktionen“)
Sichtbarkeit einstellen:
(siehe „Allgemeine Funktionen“)

Objekt parametrieren (Reiter „Allgemein“)

- Ausrichten der Anzeige zum Objektrand über Box **„Randabstände“**
- **Variable** zuweisen oder neu erstellen
- durch Markieren von **„veränderbar“** lässt sich der Wert später auch über Touch verändern, dafür kann man
- **Zugangsrechte** festlegen (mit den unter „Benutzer“ zugewiesenen Levels oder Variablen)
- **Rand** des Objektes einstellen (flach, versenkt, erhaben)

Objekt parametrieren (Reiter „Gestaltung“)

- Wenn ein variablenabhängiger Zugang gewährt wird, hier das Design wertabhängig ändern (z.B. „Disabled-Look“ und „Enabled-Look“)



Allgemeine Tips:

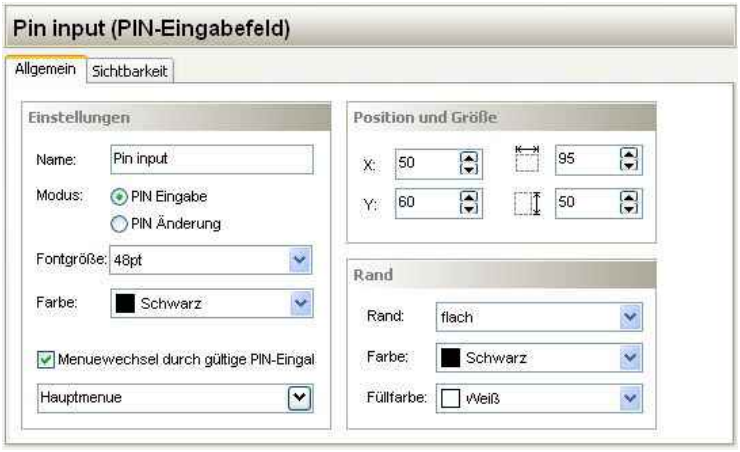
- dynamische Objekte sollten sich **NICHT überlappen**. (Ausnahme: Touchflächen), da eine zustandsabhängige Änderung des Hintergrundes unter transparenten Flächen ignoriert wird. Es wird dafür bei dem Kompilieren eine Fehlermeldung ausgegeben.
- Simulation mit „ ► “ erzeugt Fehlerbericht, der direkt zum fehlerhaften Objekt führt.

Passwortfelder

In der Benutzerverwaltung können 9 bis zu 8-stellige numerische PINs festgelegt / geändert werden, mit denen die Zugriffsrechte geregelt werden.

- **„PIN Eingabe“** ist nötig, wenn die Zugangsbeschränkung eine solche verlangt
- **„PIN Änderung“** ist nötig, wenn bestehende PINs geändert werden sollen, weil sie z.B. bekannt geworden sind.

Zum Einloggen wird automatisch eine Eingabetastatur aufgerufen. Es gibt drei Möglichkeiten, diesen Dialog aufzurufen:

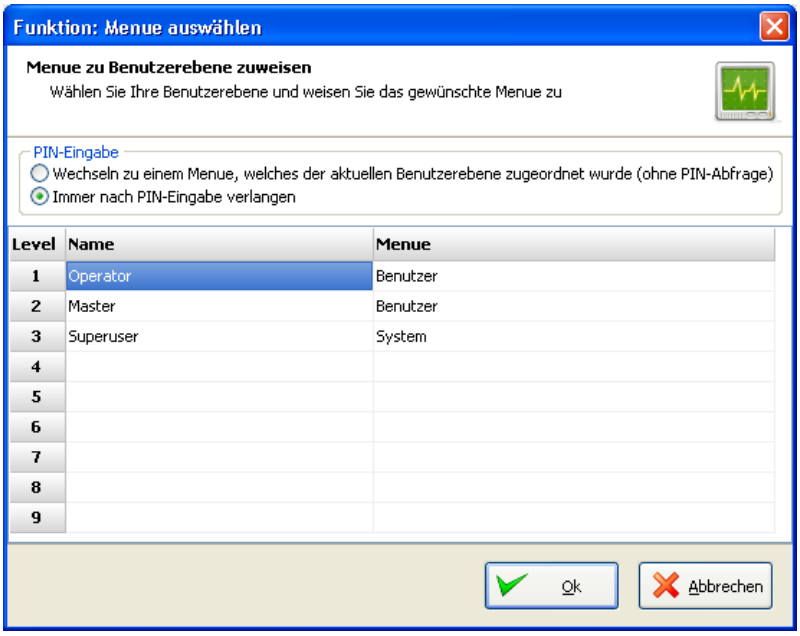


1) PIN-Eingabefeld (Objekt)
Der Vorteil dieser Variante liegt darin, dass unter der Ressource genau die Zielmenues festgelegt wurden, wohin nach erfolgreichem Login/ Logout gewechselt werden soll.

Hier kann speziell noch ein gesonderter Menuewechsel ermöglicht werden.

2) PIN-Dialoge Eingabe/Änderung (Funktion)
Der Vorteil dieser Variante liegt darin, dass jeder Taste oder Touchfeld diese Funktion zugewiesen werden kann. Der Nachteil ist die fehlende Zuordnung von Menuewechseln bei diesen Funkionen (rechts).

3) Menuewechsel mit PIN-Eingabe (Funktion)
Mit dieser Funktion werden verschiedenen Benutzerebenen verschiedene Zielmenues nach erfolgreichem Login zugewiesen.
Bei Anwahl dieser Funktion öffnet sich ein automatisch Konfigurationsfenster für die Zielmenues (unten).



Allgemeine Tips

- dynamische Objekte sollten sich **NICHT überlappen**. (Ausnahme: Touchflächen), da eine zustandsabhängige Änderung des Hintergrundes unter transparenten Flächen ignoriert wird. Es wird dafür bei dem Kompilieren eine Fehlermeldung ausgegeben.
- Das Verwenden von gleichen Objekten und wenig Sprachen spart Speicherplatz!
- Simulation mit „ ► “ erzeugt Fehlerbericht, der direkt zum fehlerhaften Objekt führt.

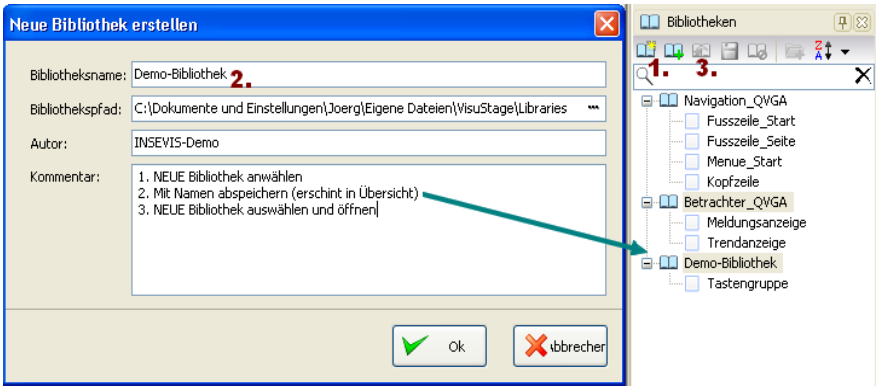
Arbeiten mit Bibliothekselementen

Bibliothekselemente werden benötigt, um in verschiedenen Projekten wiederkehrende Objekte und Objektgruppen vorkonfiguriert zur Verfügung zu haben. Zur Verwaltung werden diese Elemente in Bibliotheken abgelegt.

Das erleichtert die Erstellung mehrerer Visualisierungen und sichert eine sinnvolle Durchgängigkeit bei wiederkehrenden Funktionen (z.B. Statusanzeigen, Navigationsleisten, LogIn-Seiten)

Erstellen von Bibliothekselementen

Es können einzelne und mehrere Objekte (gruppiert oder ungruppiert) in die Bibliothek aufgenommen werden. Bei einem Bibliothekselement aus mehreren Objekten ist eine vorherige Gruppierung sinnvoll, da die Positionen der Objekte untereinander dann immer gleich bleiben.



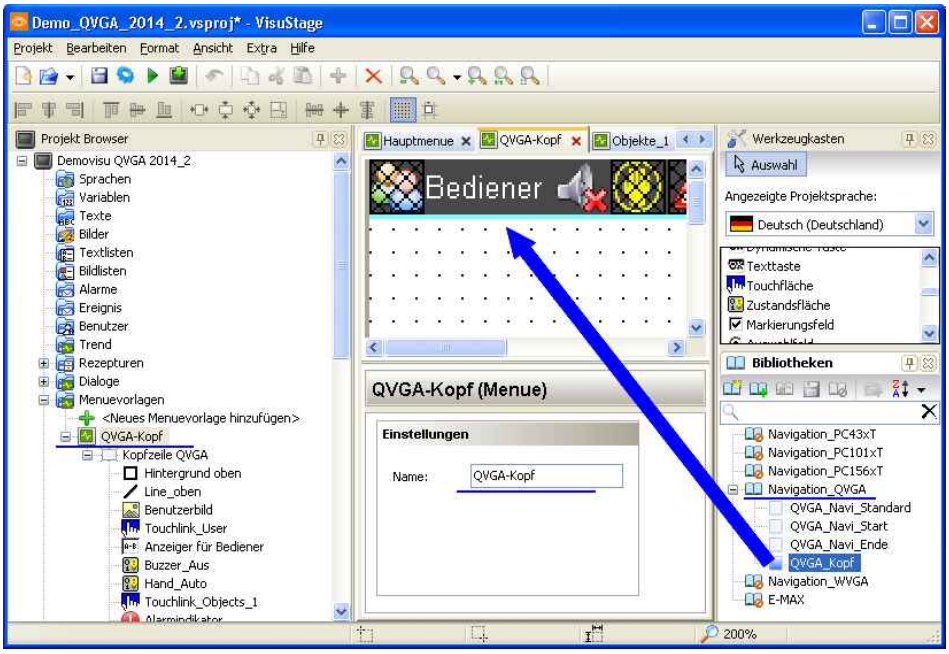
Neue Bibliothek erstellen:
- „Neue Bibliothek“ erstellen
- Daten eingeben und abspeichern
- Auswählen (Markieren)
- Öffnen der Bibliothek

Neues Bibliothekselement erstellen:
- Öffnen der Bibliothek
- Ziehen des gewünschten Elementes per Drag´n Drop in das Anzeigefenster

Verwenden von Bibliothekselementen

Um Bibliothekselemente verwenden zu können, muss die jeweilige Bibliothek geöffnet werden. (Doppelklick)

Danach werden die gewünschten Bibliotheks-elemente einzeln per Drag´n Drop in das jeweiligen Bild-schirmfenster gezogen und erscheinen in der Projektliste als Gruppe. Um die einzelnen Objekte der Gruppe zu konfigurieren, muss die Gruppe nicht aufgelöst werden.



Bei Einfügen der Bibliotheks-elemente in ein neues Projekt / Menue werden
- Variablen,
- Texte und
- Bilder

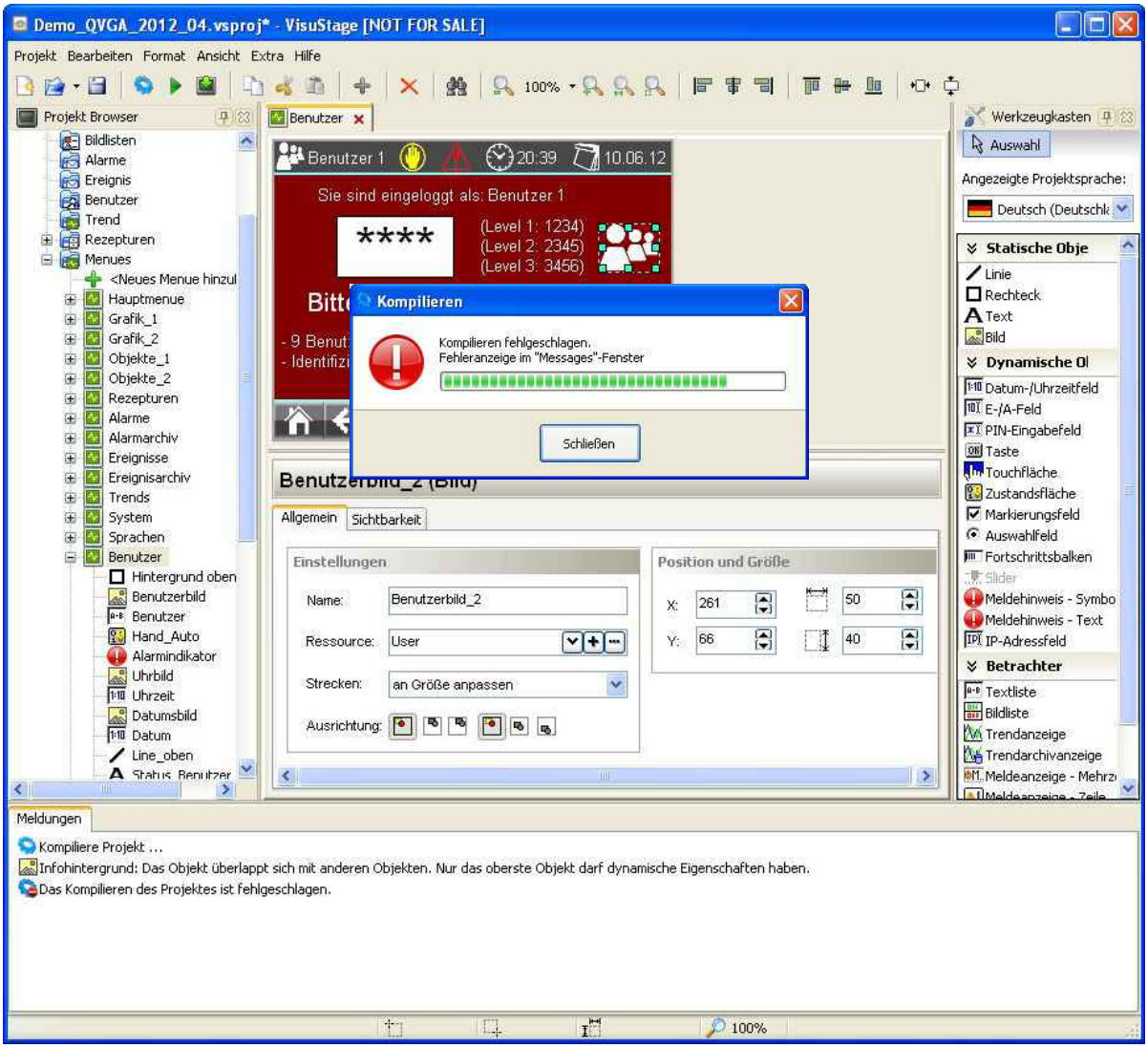
1:1 übernommen, alle anderen Ressourcen müssen neu angelegt / zugewiesen werden.

Bibliothekselemente werden in das Visualisierungsprojekt (*.vsproj) eingebettet. Um anderen seine Bibliotheken (*.vsl) zur Verfügung zu stellen, sollten immer ganze Bibliotheken (Ordner im VisuStage-PC) versendet werden.

Simulation und Download

Fehlerbereinigung

- Nach Fertigstellung Simulation starten durch F9 oder „▶“
- Wenn Fehler vorhanden, bleibt der Compiler mit einer Meldung auf Objekt und Art des Fehlers stehen.
- (Doppelklick auf Fehler führt direkt zu fehlerhaften Objekt)
- Fehler beheben und Simulation neu starten mit „▶“



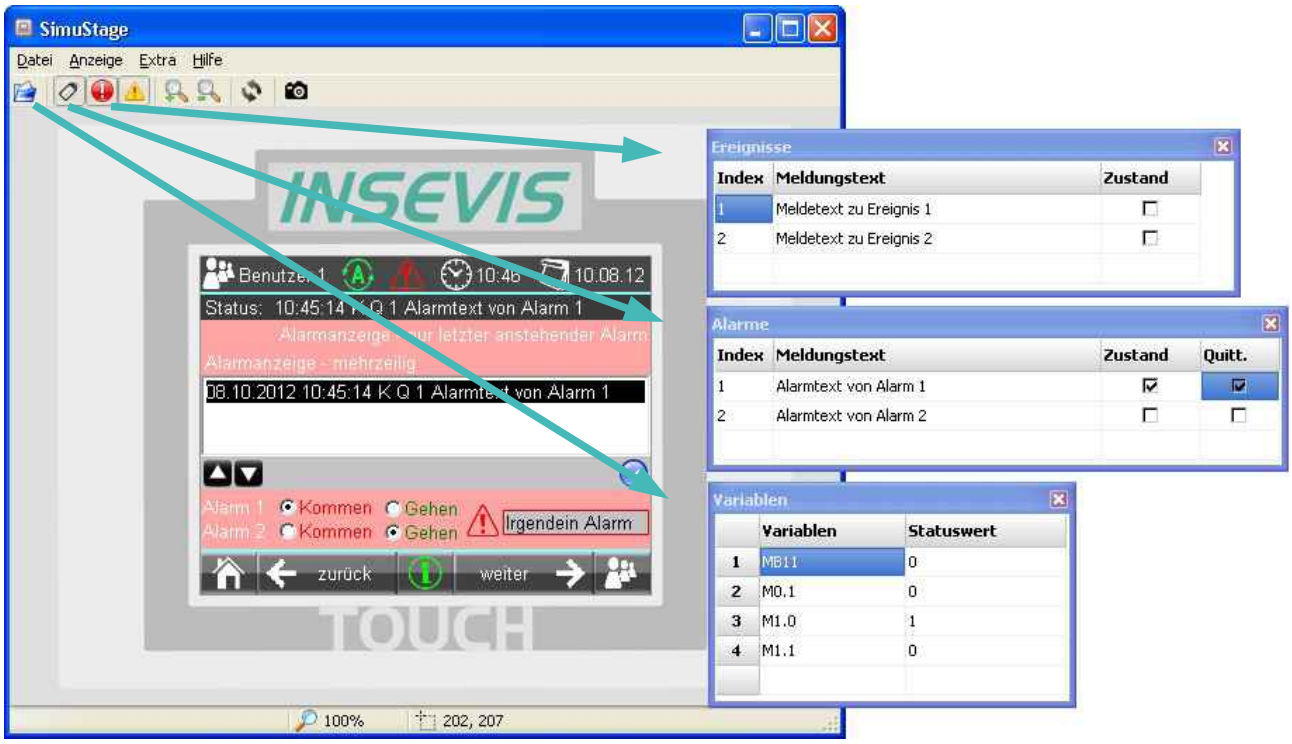
Wenn kein Fehler vorhanden, wird der Compiler das *.bin-File erzeugen und es erfolgt ein automatischer Start des Simulationsprogramms **SimuStage** und der Simulation der Visualisierung.

Allgemeiner Tip

- Das Fehlerfeld wird mit der Taste „F6“ wieder geschlossen.

Simulation und Download

Die VisuStage lädt das aktuelle Projekt und startet die Simulation (SimuStage) in neuem Fenster.
Die Bedienung erfolgt über Mauszeiger wie am Touchscreen.



Über die Variablentabelle können Variablen manipuliert werden, um das Ergebnis in der Visualisierung zu simulieren.
Ab Version 2.0.1.0 können die Ereignisse (Kommen-Gehen-Quittieren) simuliert werden, ohne dafür Auswahlboxen wie im Beispiel in der Visualisierung zu verwenden.
Ebenfalls kann zur Erstellung der Anlagendokumentation ein Screenshot der Anzeige als jpg abgespeichert werden.

Übertragen in das Gerät
(starten durch Taste F12 oder Icon)

- Übernahme der Eingabe der IP-Zieladresse aus den Projektdaten

  oder

- Auswahl der in der Netzwerkumgebung zur Verfügung stehenden Geräte

Genereller Hinweis:

- Wenn nacheinander **mehrere verschiedene Gerät mit der gleichen IP-Adresse** verwendet werden sollen, wird dieses vom Windows-Betriebssystem verhindert. Um diese Funktion auszuschalten, bitte unter **cmd** folgenden Befehl eingeben: **arp -d** (dieser löscht die in Ihrem PC abgelegte MAC-Adresse des vorhergehenden Gerätes).

Remotezugang mit der Software „RemoteStage“

Allgemeines

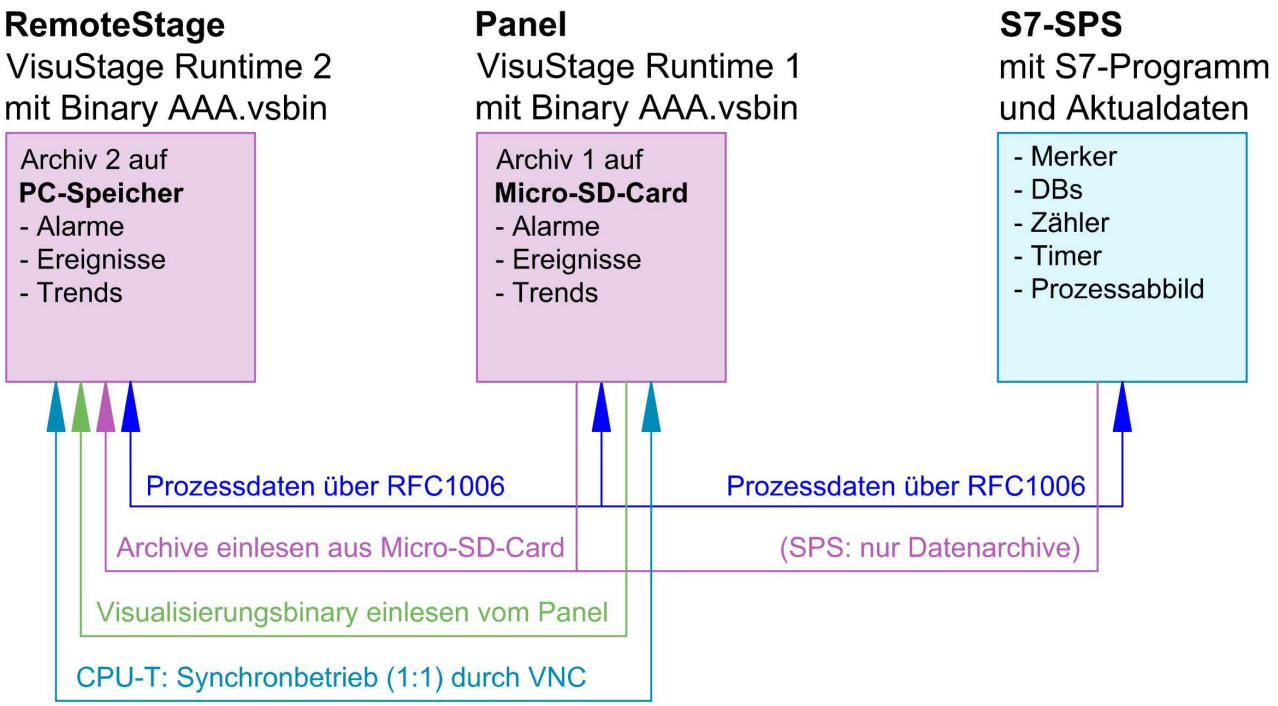
Die RemoteStage ist eine multiinstanzfähige Software, die ab Version 1.0.3.3 nicht mehr installiert werden muss, sondern als **remotestage.exe** zusammen mit der Datei für die deutsche Bedienung **remotestage.deu** in ein Verzeichnis auf dem Remote-PC oder auf einen mobilen Datenträger abgelegt wird.

Die RemoteStage benutzt die Binärfiles **(*.vsbin)** von vorab mit der „VisuStage“ erstellten Visualisierungen. Die Binärdaten sind nicht rückkonvertierbar, welches einen 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 helfen soll, den Verwaltungsaufwand zu reduzieren. In der VisuStage kann dieser Upload PIN-Code-geschützt werden.

Die RemoteStage arbeitet **wie ein zusätzliches Panel-HMI** auf dem Remote-PC, welches die Aktual-(Prozess-)daten über TCP/IP 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-CPUs über RFC1006 (und damit eine ähnliche Funktionalität wie WinCC Runtime auf einem PC)



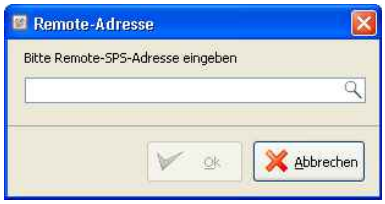
Wenn die RemoteStage für den Kunden als **Batchaufruf** bei dem Endkunden installiert werden soll, reicht folgende Eingabeaufforderung aus:

[RemoteStage-Pfad]\remotestage.exe ***/r=[IP-Adresse der Remote-SPS im selben Subnetz]***
/v=[Visualisierungsbinary-Pfad]\vollständiger Dateiname]
/start

Remotezugang mit der Software „RemoteStage“

Auswahl der Remote-Steuerung

Beim Öffnen der Binärdatei der Visualisierung („visualisierung“.vsbin)
wird einmalig die IP-Adresse der Remote-Steuerung angefordert (Abb.)
Dabei wird angeboten, im Netz nach erreichbaren Teilnehmern zu suchen (Lupe)



Danach ist diese IP-Adresse bis zu einer manuellen Änderung mit dieser Datei verknüpft und muss nicht wieder eingegeben werden.

Benutzen und Einstellen des Remote-Bildschirmes

Die PopUp-Menues sind absolut selbsterklärend.
Im Fuß der Anzeige wird der aktuelle Status angezeigt

- **Gelb:** Verbindungsaufbau
- **Grün:** Online und Remote-Steuerung in RUN
- **Rot:** Online und Remote-Steuerung in STOP

Die Umschaltung Online/Offline kann über den grünen Button neben der IP-Adressanzeige oder über den Menüpunkt „Zielsystem“ erfolgen.

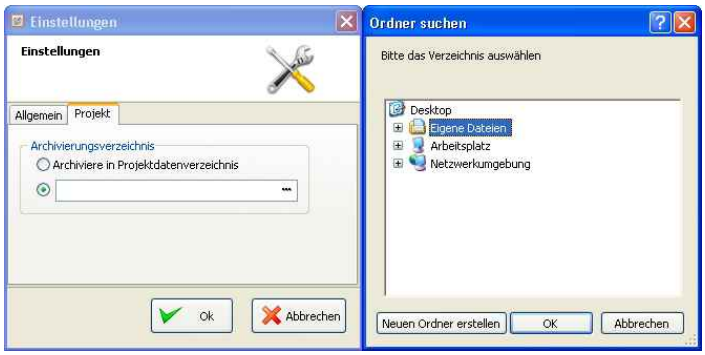
Hinweis: Bei Systemeinstellungen zum Buzzer, Erstellen von Archiven, Uhrzeit lesen werden immer die PC-Systemdaten angesprochen, nicht die des Remote-Systemes.

Abspeichern der remote (auf dem PC) archivierten Daten

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.



Remotezugang mit der Software „RemoteStage“

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 mit dem rechts gezeigtem Icon:

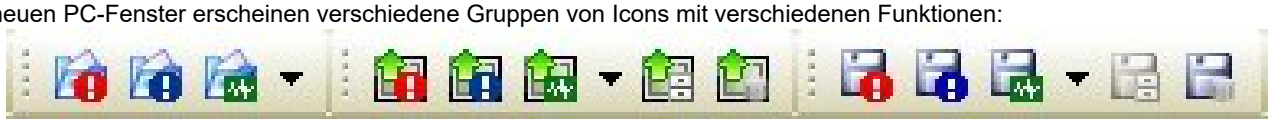
ACHTUNG:
Verwenden Sie nur die Binärdatei der zu dem Archiv passenden Visualisierung!

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	von allen Panel-SPSen und Panel-HMIs → werden immer archiviert, weil auf Micro-SD-Karte gespeichert
Archiv-Datenbausteine	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.:
- Alarmarchiv
- Ereignisarchiv
- Trendarchiv

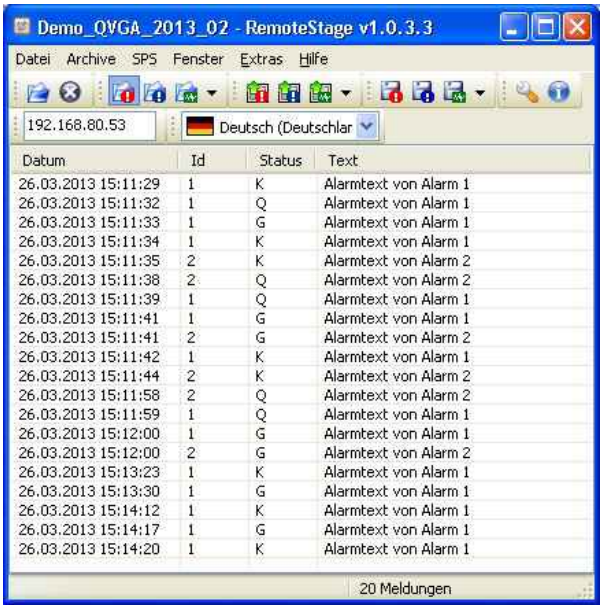
Einlesen des Archivs per Ethernet über IP-Adresse direkt in den PC
v.l.n.r.:
- Alarmarchiv
- Ereignisarchiv
- Trendarchiv
- Rezepturarchiv
- Daten (DB-)archiv

Export / Abspeichern der Archivdaten auf den PC / in das PC-Netzwerk
v.l.n.r.:
- Alarmarchiv
- Ereignisarchiv
- Trendarchiv
- Rezepturarchiv
- Daten (DB-)archiv

Ereignis- oder Alarmarchivdarstellung und -export

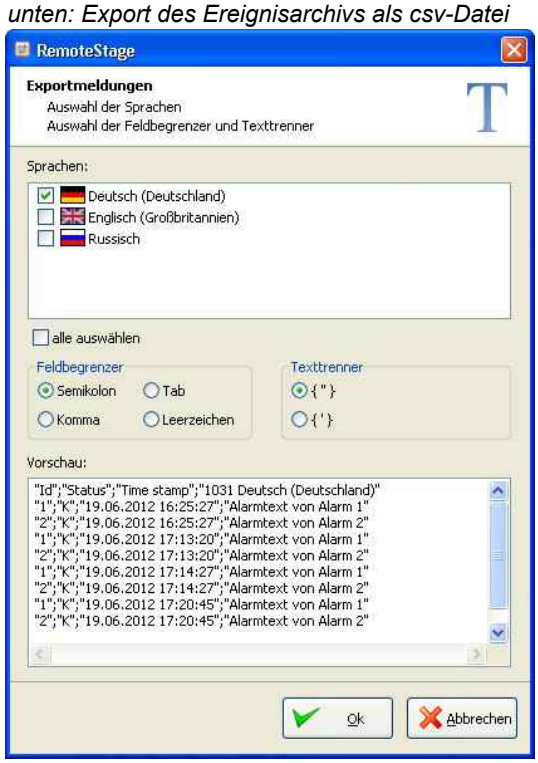
Die Archivdaten werden von der Micro-SD-Karte eingelesen, können angezeigt und als csv-Datei abgelegt werden.

unten: Anzeige des Ereignisarchivs



(Anzeige aller archivierten Alarmeinträge von der Panel-SPS / in dem Panel-HMI mit der IP-Adresse 192.168.80.50)

unten: Export des Ereignisarchivs als csv-Datei



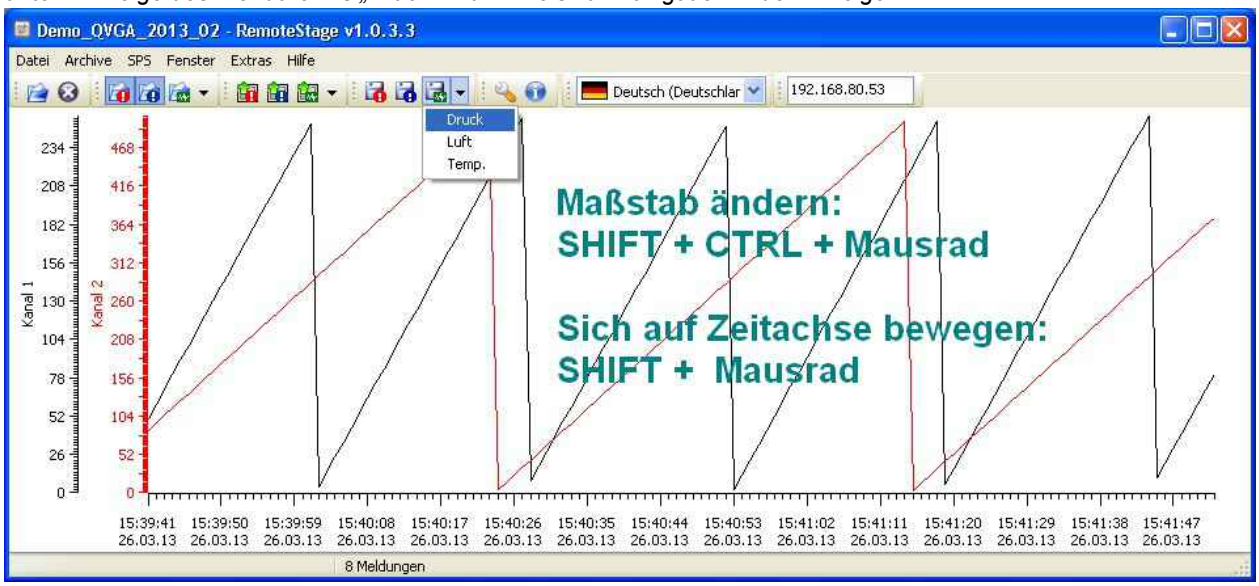
ACHTUNG:
Nicht verwechseln mit den während der Remote-Verbindung am Remote-PC archivierten Alarmeinträgen!
Auf die kundenspezifisch individuelle Formatierung des csv-Formates ist selbst zu achten.

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.

unten: Anzeige des Trendarchivs „Druck“ mit Hinweis zur Navigation in der Anzeige



(Anzeige aller archivierten Trendaufzeichnungen des Trends „Druck“ von der Panel-SPS / in dem Panel-HMI mit der IP-Adresse 192.168.80.53). Anzeige der Zeit mit dem am PC eingestellten Format (normal: DE = deutsches Zeitformat)

ACHTUNG: 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.

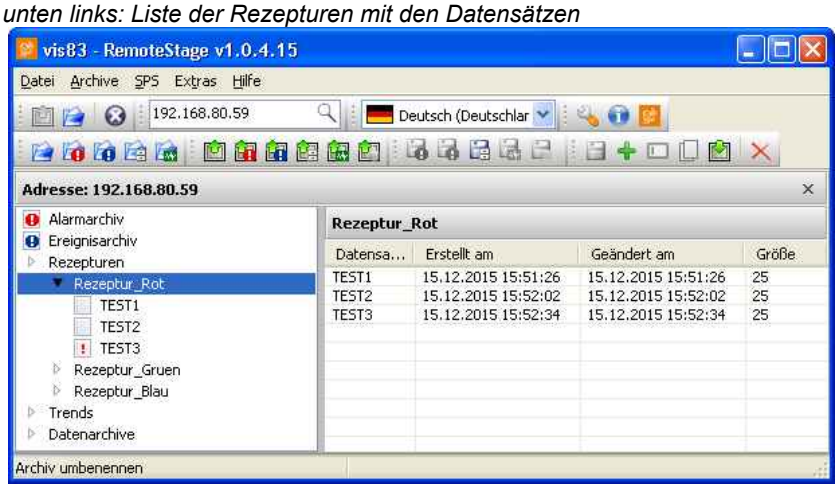
Trends können mit diesem Icon gelöscht werden

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 werden (selbsterklärend),
- als csv-Datei im PC gespeichert werden (selbsterklärend),
- als Binärdatei gespeichert, editiert und zurück in das Remote-Gerät übertragen werden

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 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.

Mit diesen aktiven Icons werden Datensätze umbenannt, kopiert, ins Remote-Gerät geladen oder gelöscht

ACHTUNG:
Die Struktur der Rezepturen MUSS erhalten bleiben. Datensätze können innerhalb der jeweiligen Rezepturen hinzugefügt oder gelöscht werden.

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 in die SPS zurückgelesen werden (siehe Systemfunktionen).

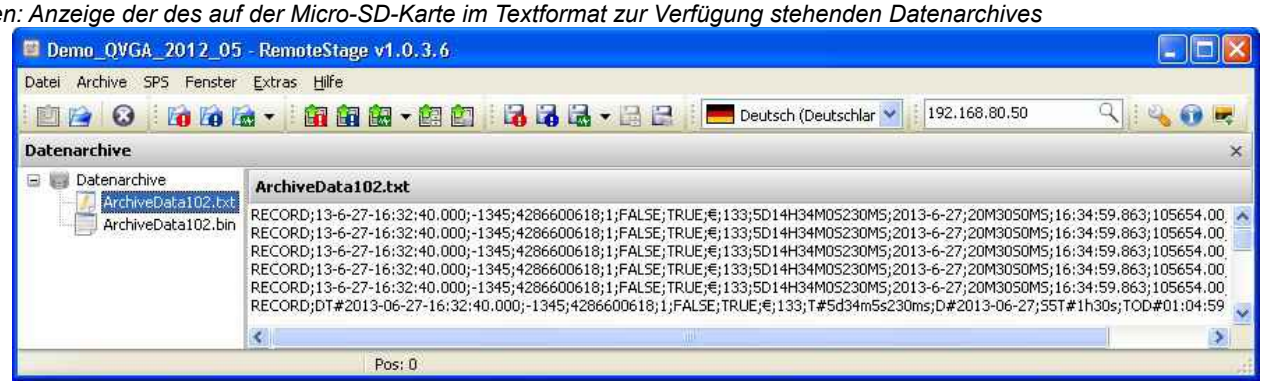
Datenarchivdarstellung und -abspeicherung

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.

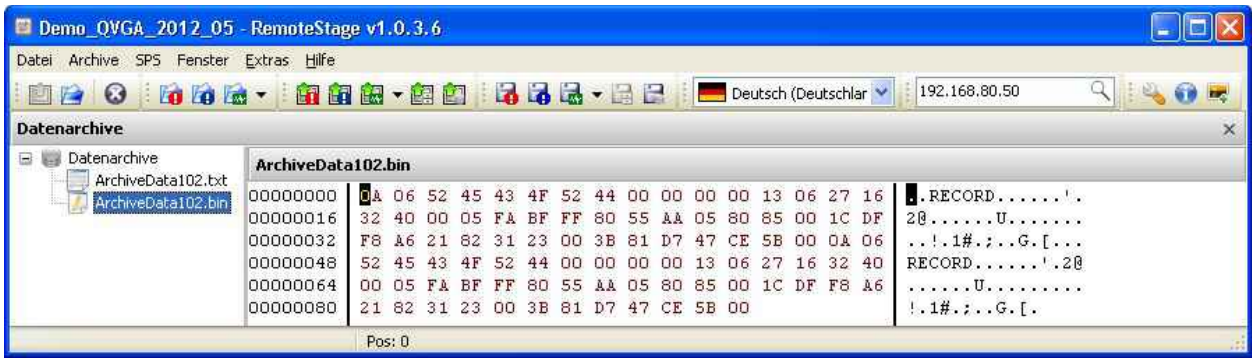
Diese Datei ist mit der RemoteStage nicht änderbar. Das Ändern kann nur mit Kundentools erfolgen, weil der RemoteStage die Struktur unbekannt ist.

unten: Anzeige der verfügbaren Datenarchive auf der Micro-SD-Karte

unten: Anzeige der des auf der Micro-SD-Karte im Textformat zur Verfügung stehenden Datenarchives



unten: Anzeige der des auf der Micro-SD-Karte im Binärformat zur Verfügung stehenden Datenarchives



Mit diesen aktiven Icons werden Datenarchive umbenannt, kopiert und ab RemoteStage V 1.0.4.18 ins Remote-Gerät geladen oder gelöscht.

Remotezugang mit der Software „RemoteStage“

Kommandozeilenbefehle für RemoteStage - Visualisierungsmodus

Die RemoteStage kann auch in den Startvorgang des Remote-PCs (geplante Tasks) mit eingebunden werden.
Hier sind folgende Befehle möglich:

Automatisches Verbinden zum Remote-Panel nach Start

Hier wird eine vorab erstellte Binärdatei der Visualisierung aufgerufen und mit einer IP-Adresse des Remote-Panels verbunden (solang dieses erreichbar ist).

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“)

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.

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 (solang diese erreichbar ist) und die RemoteStage geht in Vollbildformat. (Der Bildschirm bleibt dunkel, weil keine Verbindung eingetragen wurde...)

remotestage.exe /start /full /r=<*remote*> /v=<*project binary*>

Hinweis:
Umschalten Voll-/Teilbildschirm bei der RemoteStage mit STRG+F5

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

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)

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 Trendnummer, welche in der VisuStage konfiguriert wird (nur bei Trendarchiv-Modus)

Beispiel

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:

C:\>remotestage.exe /mode=alarm_archive /r=192.168.80.50
/o=D:\archive\alarm.csv /v=D:\visu\myvisu.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

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 (einzelner) 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

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:

C:\>remotestage.exe /mode=recipe /r=192.168.80.50 /v=d:\visu\myvisu.vsbin /recipe=Rezeptur_Rot /record=Orange /o=D:\archive\Orange.csv

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:

C:\>remotestage.exe /mode=recipe /r=192.168.80.50 /v=d:\visu\myvisu.vsbin /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

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

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

remotestage.exe /mode=recipe_download /v=d:\visu\process.vsbin
/recipe=Recipe_Rot /record=d:\recipes\Rot_123.rec /r=192.168.80.50

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

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

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 Dateiarchiv 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.

Beispiele

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:

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 zurückgeschrieben (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

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

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

remotestage.exe /mode=data_download /archive=d:\configs\default_cfg.csv /r=192.168.80.50

Nachstehende Befehlszeile überträgt den ALLE Archivdateien aus dem Ordner D:\configs in das Remote-Gerät mit der IP-Adresse 192.168.80.50

remotestage.exe /mode=data_download /archive=d:\configs*.* /r=192.168.80.50

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-Einleseservice 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

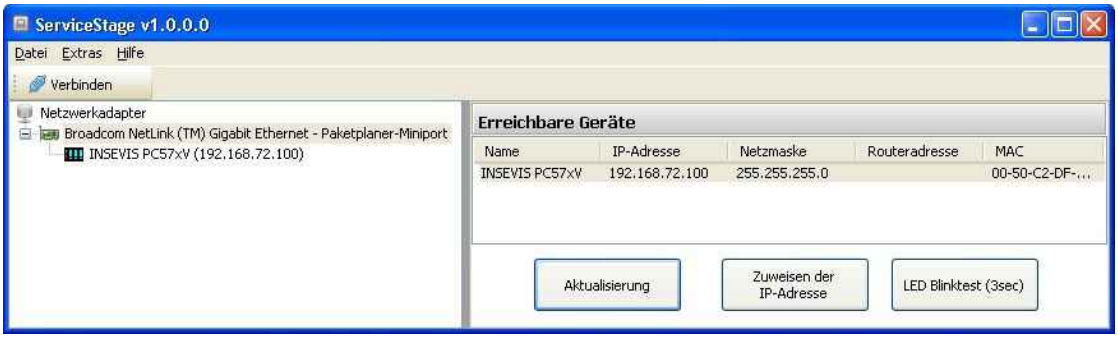
Beispiel einer Nutzung der RemoteStage in einem Batchprozess

```
@remotestage /mode=alarm_archive /r=192.168.80.50 /o=d:\archive\alarm.csv /v=d:\visu\myvisu.vsbin
@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 Programmdaten zu ändern, sowie Know-how-Schutzstufen zu setzen.
Durch die „ServiceStage“ müssen für Updates dem Service oder anderen Ausführenden keine Quelldaten mehr zur Verfügung gestellt werden. Damit kann ein Service-Mitarbeiter ganz 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 wird und drückt **„Aktualisieren“**.



Mit einem **„Blinktest“** kann man die ausgewählte CPU identifizieren (Die gelbe Batterie-LED blinkt für 3 Sekunden). Ebenso kann hier gleich die IP-Adresse, Netzwerkmaske und Router-IP-Adresse der CPU geändert werden.

Funktionsübersicht

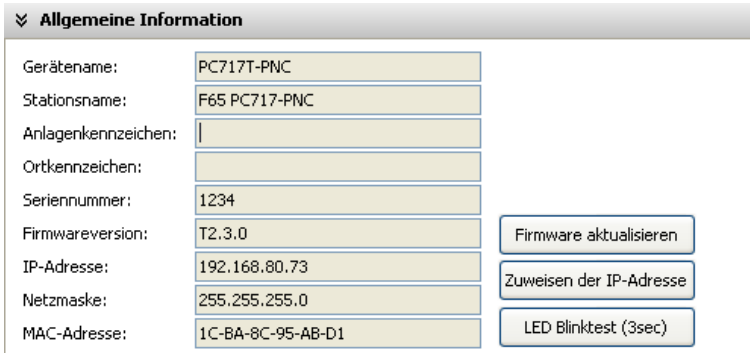
Allgemeine Informationen

Ansicht der nicht änderbaren Systemdaten wie Seriennummer, Firmwareversion, MAC-Adresse sowie der änderbaren Daten wie

- IP-Adresse,
- Netzmaske und
- Router-IP-Adresse

sowie eine Blinktest-Möglichkeit der ausgewählten CPU.

Nur bei CPU-T:
Firmwareupdate per Ethernet möglich



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 Anhaken der Funktion **„Übernehmen vom PC“** werden aktuelles Datum und Uhrzeit vom PC übernommen, wenn **„Anwenden“** gedrückt wird.

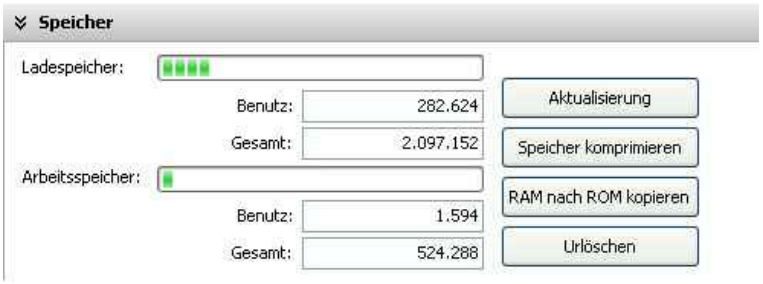
Speicher

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.

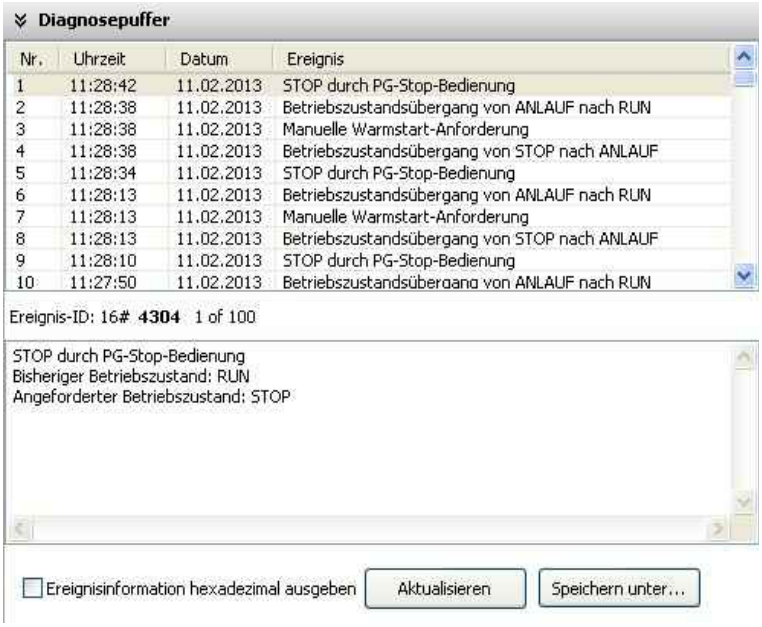


Diagnosepuffer

Hier werden die jeweils letzten 100 Einträge in den Diagnosepuffer nach Datum und Uhrzeit angezeigt (nicht bei Panels-HMI).

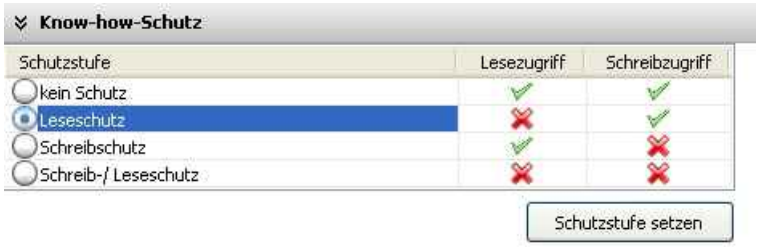
Auf Wunsch erfolgt das durch Anhaken der Funktion **„Ereignisinformation hexadezimal ausgeben“** auch in einem hexadezimalen System.

Durch **„Aktualisieren“** werden inzwischen angefallene Ereignisse neu eingetragen, mit **„Speichern unter“** werden die Ereignisse in eine Textdatei (*.txt) abgelegt.



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.



Kein Schutz: Die Programmbausteine können von jedem gelesen und geändert werden. Die Schutzstufe kann erhöht werden, in dem 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-Programmbausteine 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“/ Urlöschen (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-Programmbausteine können von jedem gelesen aber nicht mehr geändert werden. Diese Schutzstufe kann nur noch mit Hardware-Urlöschen direkt am Gerät zurückgesetzt werden, danach ist das ungeschützte Original-S7-Programm neu in die SPS zu übertragen.

Schreib-/Leseschutz: Die S7-Programmbausteine können weder geändert noch gelesen werden. Die Panel-HMIs können die Prozessdaten weiterhin auslesen. Diese Schutzstufe kann nur noch mit Hardware-Urlöschen direkt am Gerät zurückgesetzt werden, danach ist das ungeschützte Original-S7-Programm neu in die SPS zu übertragen.

Block Update

Diese Funktion ermöglicht es,
- das S7-Programm als MemoryCard-file ***.wld** * (nicht bei Panels-HMI).
- die Visualisierung als Binärfile ***.vsbin** und
- die Konfigurationsdaten als ***.csbin** (nicht bei Panels-HMI).

über das Netzwerk einzeln in die CPU zu laden.

* Die WLD-Datei ist eine Binärdatei mit S7-Systemdaten (SDBs) und S7-Programmblöcken (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).

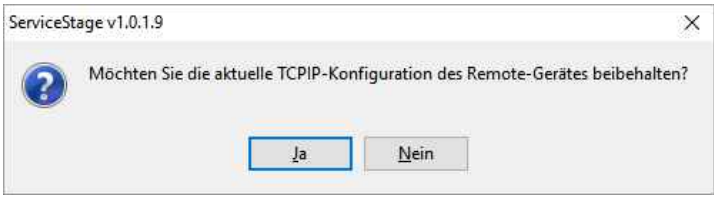
→ Man kann also mit der WLD-Datei GANZE S7-Projekte **ODER** EINZELNE DBs mit / ohne ihren Systemdaten übertragen.

ACHTUNG: Wenn man Systemdaten mit 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) oder
- die SDBs des SimaticManagers überschreiben die ConfigStage-Systemkonfiguration.
  In diesem Fall muss die *.csbin-Datei des Projektes nach der wld-Datei erneut aufgespielt werden.

Verwendung für Backups
- 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.

Arbeiten mit der Software „ServiceStage“

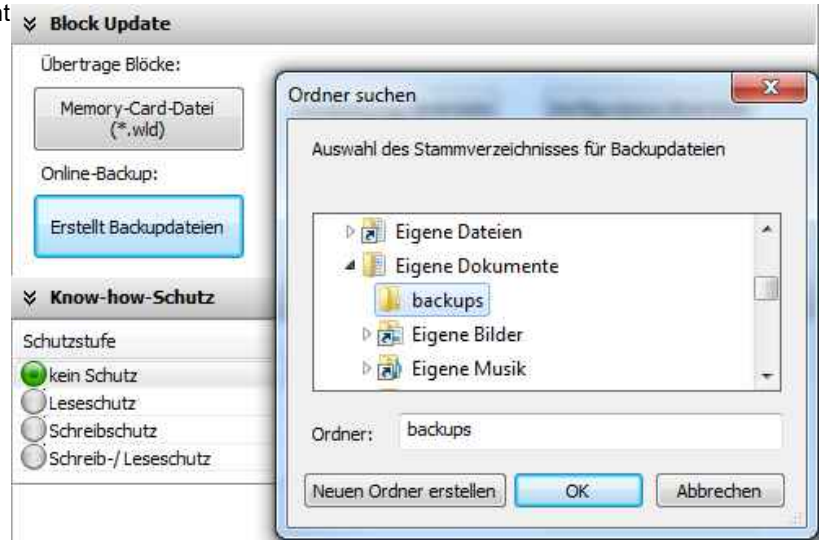
Falls in dem Hardwarekonfigurationsbaustein des Zielgerätes eine andere IP-Adresse eingetragen sein sollte, als in den Konfigurationsbausteinen der übertragenen Böcke, kann es dazu führen, dass nach dem Übertragen die Verbindung abbricht. Um das zu vermeiden, wird ab Rev. 1.0.1.9 gefragt, ob die „Vor Ort“-Adresse im Zielgerät überschieben werden soll.



Online Backup

Diese Funktion erstellt eine WLD-Datei samt Systemdaten und allen OB, DB, FB, SFB und SFC und kopiert diese zusammen mit der Binärdatei der Visualisierung in einen Ordner „backups“.

Mit diesen Daten lassen sich S7-Programme und Visualisierungen per Micro-SD-Karte ganz ohne PC durchführen (mehr unter „Systemfunktionen“, Kapitel „Datensicherung und -wiederherstellung“)



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.

INSEVIS

INSEVIS - Gesellschaft für industrielle
Systemelektronik und Visualisierung mbH

Am Weichselgarten 7
D - 91058 Erlangen

Fon: +49(0)9131-691-440
Fax: +49(0)9131-691-444
Web: www.insevis.de
E-Mail: info@insevis.de

Zertifiziert nach DIN EN ISO 9001:2008