

SPEZIFIKATION

Programmier-Schnittstelle Parker Compax-M/S[®] für Siemens Simatic S7[®]

1	EINLEITUNG	1
2	PROGRAMMIERKONVENTIONEN	2
2.1	HOCHSPRACHENIMPLEMENTIERUNG MIT S7-SCL.....	2
2.2	BEZEICHNUNGEN	2
2.3	BEMERKUNGEN ZUR REALISIERUNG	2
3	GERÄTEKONFIGURATION	3
3.1	PROFIBUS-S7-KONFIGURATION MIT DEM STEP7-HARDWARE-KONFIGURATOR.....	3
3.2	PROFIBUS-COMPAX-KONFIGURATION MIT DEM SERVOMANAGER COMPAX-M/S	3
3.2.1	COMPAX-Profibus-Parameter P196: Telegrammformat	5
3.2.2	COMPAX-Profibus-Parameter P194: Stationsadresse	5
3.2.3	COMPAX- Profibus-Parameter P193: Spontanmeldungen.....	5
3.2.4	COMPAX- Profibus-Parameter P191: Verhalten bei Bus-Time-Out	5
3.2.5	COMPAX- Profibus-Parameter P135 .. P142: Belegung des Prozessdatenkanals PZD.....	5
3.2.6	COMPAX-Parameter P221 .. P224: Maskierung der digitalen Ein- und Ausgänge	6
3.2.7	COMPAX-Parameter P227 und P229: Konfiguration einer Stillstandsanzeige.....	6
4	FUNKTIONSBAUSTEIN FB CPXMANAGER	6
4.1	PROJEKTÜBERSICHT.....	6
4.1.1	Funktionen	6
4.1.2	Software-Einheiten.....	7
4.1.3	Technische Parameter fbCPXManager	7
4.2	FB-AUFRUF UND PARAMETRIERUNG FB CPXMANAGER	8
4.2.1	Anlaufinitialisierung und Aufruf des fbCPXManager	8
4.2.2	Compax-Parameterzugriff über fbCPXManager	9
4.2.3	Parametrierung	10
5	BEMERKUNGEN ZU SICHERHEIT UND SIGNALSPIEL	11
6	REFERENZEN	14

1 Einleitung

Spezifikation

Die folgende Spezifikation beschreibt den Simatic-S7-Funktionsblock *fbCPXManager* für Parker Compax-MS und dessen Einbindung in die Anwendung. Die Dokumentation des Herstellers (siehe Kapitel *Referenzen*) hat grundsätzlich Vorrang, wenn Aussagen in dieser Spezifikation dazu widersprüchlich sind. Diese Grundaussage gilt sinngemäß auch für die implementierte Software.

Haftungsausschluss

Aus aktueller Sicht gibt es generell keine Software, die unter allen Umständen und Anforderungen fehlerfrei arbeitet. Inmotec lehnt darum jedwede Haftung für direkte und indirekte Schäden aller Art durch Einsatz des Treibermoduls ab, auch wenn das Modul entsprechend den Beschreibungen der vorliegenden Spezifikation verwendet wird.

Produktbeobachtungspflicht

Inmotec versucht im Rahmen der Produktbeobachtungspflicht, Gefahren, die aus dem Einsatz unserer Softwaremodule entstehen, zu erkennen und zu beschreiben. Durch die Komplexität und unsere beschränkte Einsicht in die Anlagen der Endkunden, die auch durch den Einsatz von Produkten Dritter geprägt sind, ist es uns jedoch nicht möglich, alle Gefahren zu erkennen.

2 Programmierkonventionen

2.1 Hochsprachenimplementierung mit S7-SCL

Der Treiberbaustein für CPX-M/S wurde mit dem Engineering-Tool S7-SCL (Structured Control Language, Strukturierter Text nach IEC61131-3) von Siemens implementiert, da sowohl kompakter als auch und vor allem verständlicher Code geschrieben werden kann.



Es gibt keine Einschränkungen bei der Nutzung von SCL-implementierten Bausteinen im AWL-, FUP-, KOP- oder Graph-Kontext. Das SCL-Paket muss nicht installiert sein.

2.2 Bezeichnungen

Alle Typen und Instanzen tragen englische Bezeichnungen wie auch die Funktionen und Schnittstellen aller bekannten Siemens-S7-Standardbausteine. Es wird angestrebt, einheitlich typabhängige Präfixe/Postfixe für Typen und deren Instanzen zu verwenden. Analog zu anderen IEC61131-3-fähigen SPS'n wird daher die Verwendung folgender Prä- und Postfixe vorgeschlagen, was für den AWL-Programmierer zumindest die Konsequenz hat, dass er die Namensgebung der Schnittstellen-Variablen zu übernehmen hat.

Präfix	Bedeutung	Postfix	Bedeutung
pb	POU Instanz Programmblock	RE	Steigende Flanke (Speicher)
fb	POU Instanz Funktionsblock	FE	Fallende Flanke (Speicher)
fc	POU Funktion	Q	Impuls
ds	Datenstruktur	Cnt	Zählervariable
st	Typ Datenstruktur (UDT)	Pin	Eingangsparameter
bo	Boolsche Variable (BOOL)	Pout	Ausgangsparameter
in	Integer-Variable (INT)	Const	Konstante als Makro-Vereinbarung
do	Double integer-Variable (DINT)		
re	Real-Variable (REAL)		
wo	Word-Variable (WORD)		
dw	Double word-Variable (DWORD)		
tn	Timer TON		
tf	Timer TOF		
tp	Timer TP		
tv	Timer-Werte als T#		

2.3 Bemerkungen zur Realisierung

Obwohl SCL mit symbolischen Bausteinennamen arbeitet, erzeugt der Compiler von SCL über die Symboltabelle zur Compile-Zeit absolute FB- und FC-Nummern, soweit sie nicht schon in der SCL-Quelle direkt vergeben wurden.

- Die vordefinierten Step7-Variablen OK und ENO werden unterstützt.
- Im CPX-M/S-Baustein werden ausschließlich systemweite SFC-Aufrufe und SFB-Instanzen verwendet und keine weiteren FB/FC-Instanzen aufgerufen, so dass vorgegebene Bausteinnummern des Treiber-Projektes grundsätzlich verschoben werden dürfen.
- Der CPX-M/S-Baustein verwendet keine globalen Ressourcen wie Merker, T-Zeiten und Z-Zähler, so dass diese dem Anwenderprogramm zur vollen Verfügung stehen, auch wenn der CPX-M/S-Baustein mehrfach instanziiert wird.
- Es wird empfohlen, die CPX-M/S-FB's im statischen Deklarationsteil eines Anwender-FB's zu instanziiieren und damit den Instanz-DB des aufrufenden FB's zu nutzen.
- Um die Kommunikation mit den CPX-M/S-Stationen zu realisieren, sind die Bausteine grundsätzlich mit der Startadresse der E/A-Bereiche der CPX-M/S-Station (vergeben im Hardware-Konfigurator der Step7-Software) zu versorgen. Durch die Vorgabe aller vier Slotadressen pro DP-Slave *Compax-MS* wird größtmögliche Flexibilität bei der Konfiguration erreicht (speziell für 400er CPU's sinnvoll).

- Auch wenn der Datenaustausch über einen externen Master erfolgen soll, müssen die Systembausteine *SFC14 (DPRD_DAT)* und *SFC15 (DPWR_DAT)* auf der CPU vorhanden sein, das ist bei CPU's ohne „-2DP“ nicht der Fall. Sollte anwenderseitig die Notwendigkeit bestehen, auf CPU's ohne internen Profibus-Master zurückzugreifen (z.B. mit CP342-5 und Nutzung von FC2/FC1), kann der realisierte Treiber auf CP-Kommunikation umgestellt werden.

3 Gerätekonfiguration

3.1 Profibus-S7-Konfiguration mit dem Step7-Hardware-Konfigurator

Die Konfiguration (Einbindung in S7-Profibus-Konfiguration) erfolgt mit dem S7-Hardware-Konfigurator. Parker Automation stellt dafür die Gerätestammdatei *CPX0EE95.GSD* für Compax-M/S zur Verfügung (Tool-CD). Nach der Installation des Servomanagers befindet sich diese Datei im Installationsverzeichnis *..SRVBOX\DATA*. Das Gerät wird nach GSD-Installation im Typenbaum unter *PROFIBUS-DP\Weitere Feldgeräte\Allgemein* als *COMPAX Option F3* angezeigt. Der DP-Slave ist zwingend als Typ *BPO1 BKD/PZD* oder *BPO1 BKD/PZD E/A getrennt* anzulegen.

Der BPO-Typ1 (E/Azusammenhängend oder E/A getrennt) (neue Bezeichnung PPO) stellt sowohl BKD- (neue Bezeichnung PKW) als auch eine erweiterte PZD-Kommunikation zur Verfügung. Der Treiberbaustein *fbCPXManager* benötigt zwingend diese Kommunikationstypen.

3.2 Profibus-Compax-Konfiguration mit dem Servomanager Compax-M/S

Der BPO-Typ 1 BKD/PZD legt fest, dass über einen 10-Byte-BKD-Bereich und einen 8 Bytes großen PZD-Bereich (PED = Prozesseingangsdaten für die Simatic S7, PAD = Prozessausgangsdaten für die Simatic S7) kommuniziert wird.

Der S7-Treiber legt für alle 4 Bereiche Mailboxen an, die wie folgt definiert sind.

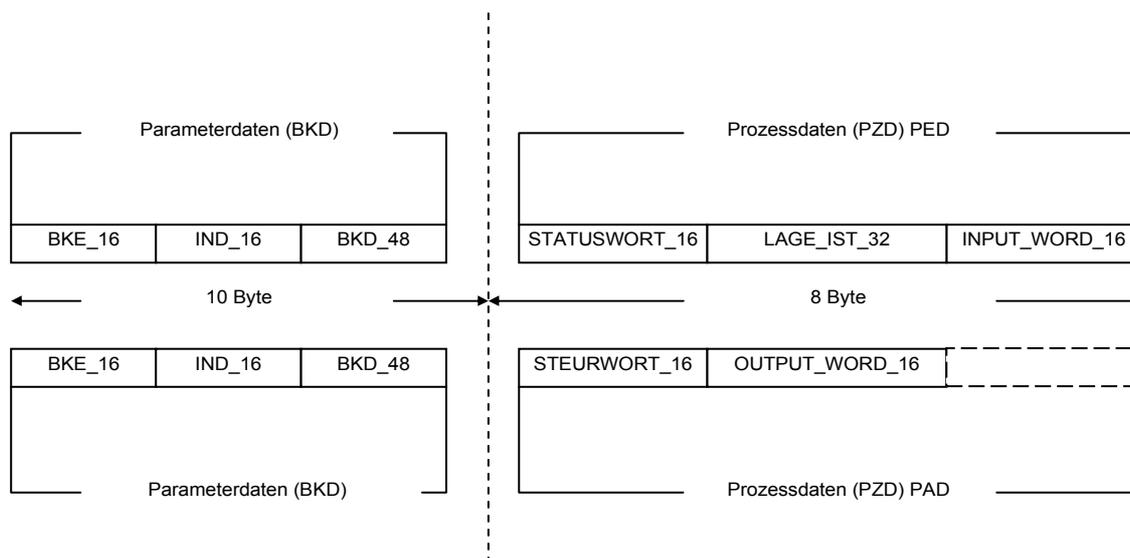


Abbildung 1 Aufbau des Compax-Telegramms

```
// BKD-Kommunikation In/Out
dsMailBoxBKDDSPIn, dsMailBoxBKDDSPOut:
  STRUCT
    woBKE: WORD := 16#0000; // Auftrag/Antwort (BKE = AK + BNU)
    woIND: WORD := 16#0000; // Index (IND)
    arBKD: ARRAY [0..5] OF
      BYTE := 16#0; // DSP Bytes des BDA-Feldes
  END_STRUCT;

// PZD-Kommunikation, PED
dsMailBoxPZDPED:
  STRUCT
```

```

        woStatusWord:   WORD := 16#0000;           // STATUSWORT, binär
        doActPos:       DINT := 0;                // Aktuelle Position, C4_3
        woInputWord:    WORD := 16#0000;         // Digitale Eingänge, binär
    END_STRUCT;

// PZD-Kommunikation, PAD
dsMailBoxPZDPAD:
    STRUCT
        woControlWord:  WORD := 16#0000;         // STEURWORT, binär
        woOutputWord:   WORD := 16#0000;         // Digitale Ausgänge, binär
        dwPlaceholder1: DWORD :=16#00000000;     // Platzhalter
    END_STRUCT;

// UDT „stControlType“ (UDT10) für das STEUERWORT
STRUCT
    // High byte
    boJogP:              BOOL;                    // Tippen+
    boJogN:              BOOL;                    // Tippen-
    boLeadByAG:          BOOL;                    // NB*: Compax wird geführt durch SPS
    boTeachRealZero:    BOOL;                    // NB: Setze Realnull
    boModeRelAbs:        BOOL;                    // NB: Modus =1 POSR und =0 POSA
    boNewSetPoint:       BOOL;                    // NB: Neuer Sollwert
    boPlaceholder1:      BOOL;                    // NB: Platzhalter
    boPlaceholder2:      BOOL;                    // NB: Platzhalter
    // Low byte
    boDisableBrakeClose: BOOL;                    // NB: OFF und Bremse schließen (haltend)
    boDisableBrakeOpen:  BOOL;                    // NB: OFF und Bremse öffnen (n. haltend)
    boStart:              BOOL;                    // NB: Start Satzprogramm oder Position.
    boSearchRealZero:     BOOL;                    // NB: Realnullfahrt
    boSerachMachineZero:  BOOL;                    // Maschinennullfahrt
    boBreak:              BOOL;                    // Break (Verwerfen des Verfahrssatzes)
    boStop:               BOOL;                    // NB: Stop
    boQuit:               BOOL;                    // Reset Compax-Fehler
END_STRUCT;
END_TYPE;

// UDT „stStatusType“ (UDT11) für das STATUSWORT
STRUCT
    // High byte
    boSetPointReached:   BOOL;                    // Compax in Position
    boLeadedByAG:        BOOL;                    // NB: Compax is geführt durch SPS
    boCtrlOpRunning:     BOOL;                    // Steuerbits im STEURWORT gesetzt
    boPlaceholder4:       BOOL;                    // NB: Platzhalter
    boNewSetPointNotValid: BOOL;                    // NB: Neuer Sollwert ist nicht gültig
    boNewSetPointAccepted: BOOL;                    // NB: Neuer Sollwert akzeptiert
    boTrackingError:     BOOL;                    // NB: Compax mit Schleppfehler
    boMotorBlocked:      BOOL;                    // NB: Motor is blockiert
    // Low byte
    boPlaceholder1:      BOOL;                    // NB: Platzhalter
    boPlaceholder2:      BOOL;                    // NB: Platzhalter
    boReadyForStart:     BOOL;                    // NB: Ready for Start
    boError:              BOOL;                    // Compax-Fehler
    boMachineZeroValid:  BOOL;                    // Maschinennullbezug gültig
    boPlaceholder3:      BOOL;                    // NB: Platzhalter
    boStandstillAfterStop: BOOL;                    // NB: Stillstand nach Stop/Break
    boNoStandstillSetPoint: BOOL;                    // Kein Stillstand
END_STRUCT;
END_TYPE;

```

*) bedeutet, dass der Treiberbaustein *fbCPXManager* (FB10) diese Bits nicht benutzt.

Das Festlegen einer Baudrate ist nicht erforderlich, da sich die Anschaltung des COMPAX automatisch auf die Master-Baudrate einstellt. Es werden alle genormten Baudraten unterstützt, maximal jedoch 1.5 Mbaud. Es muss beachtet werden, dass einige Profibus-Parameter des COMPAX erst nach erneutem Zuschalten der 24V-Versorgung (Logikspannung) am Compax-M/S aktiv werden.

3.2.1 COMPAX-Profibus-Parameter P196: Telegrammformat

<i>Funktion</i>	<i>Einstellung</i>	<i>Wertigkeit</i>
BPO-Typ	BPO-Typ 1	0
	BPO-Typ 2	1
	BPO-Typ 3	2
	BPO-Typ 4	3
Ein-/Ausgabe-Konfiguration	E/A zusammen	0
	E/A getrennt	4
DP-Konfiguration (Konsistenzbereiche)	3 Teilung (BKD + IND/BDA/PZD)	0
	2 Teilung (BKD/PZD)	8
Profibus-Protokoll	nur FMS-Betrieb	0
	nur FMS-Betrieb	32
	Mischbetrieb (DP- und FMS-Betrieb)	64
	nur DP-Betrieb	96

Für den P196 muss $0 + 0 + 8 + 96 = 104$ oder $0 + 4 + 8 + 96 = 108$ gewählt werden. Mit dieser Einstellung kann der PZD-Kanal vollständig genutzt werden.

3.2.2 COMPAX-Profibus-Parameter P194: Stationsadresse

Jeder DP-Slave meldet sich am Bus mit seiner Stationsnummer an. COMPAX-Stationsnummern werden exklusiv im Bereich von 1 bis 126 vergeben. Die S7-CPU und das Programmiergerät am Profibus als DP-Master Klasse 1 bzw. 2 belegen vordefinierte Stationsnummern, die dann für einen COMPAX-Slave nicht mehr vergeben werden darf.

Einstellung im Bereich P194 = 4 .. 126.

3.2.3 COMPAX- Profibus-Parameter P193: Spontanmeldungen

Bei Einsatz des Treiberbausteins dürfen prinzipiell keine Spontanmeldungen eingesetzt werden

Einstellung P193 = 0.

3.2.4 COMPAX- Profibus-Parameter P191: Verhalten bei Bus-Time-Out

Mit diesem Parameter wird das Verhalten der COMPAX-Positionierregler bei Bus-Time-Out festgelegt. Wenn P191 = 0 ist, sind alle Sollwerte bis zum Eintreffen eines neuen Befehls gültig. Bei P191 = 1 (oder 15) führt dieser Fehler zum Achsenstop mit anschließendem Stromlosschalten der Achse bei eingefallener Stillstandsbremse, so vorhanden.

Dieser Parameter hat keinen Einfluss auf das Verhalten des Treiberbausteins. Es wird jedoch aus Sicherheitserwägungen empfohlen, den P191 = 1 zu setzen.

3.2.5 COMPAX- Profibus-Parameter P135 .. P142: Belegung des Prozessdatenkanals PZD

<i>Parameter</i>	<i>Bedeutung</i>	<i>Festwert</i>
P135	PED-Objekt 1 LAGE_IST (Format C4_3, 1/1000 Auflösung)	1229056
P136	belegt durch PED-Objekt 1	0
P137	belegt durch PED-Objekt 1	0
P138	PED-Objekt 2 INPUT_WORD (binär)	1236224
P139	PAD-Objekt 1 OUTPUT_WORD (binär)	1236992
P140	belegt durch PAD-Objekt 1	0
P141	frei	0
P142	frei	0

3.2.6 COMPAX-Parameter P221 .. P224: Maskierung der digitalen Ein- und Ausgänge

Parameter	Bedeutung	Vorschlagswert
P221	Funktion der digitalen Eingänge E1 .. E8 (Steuereingänge E1 .. E6) wird aufgehoben, die Eingänge E1 .. E6 sind freie Eingänge. Im INPUT_WORD über Profibus steht das Eingangswort (Eingänge E1 .. E16)* zum Lesen bereit (durch PED-Belegung)	63
P223	Zuordnung der digitalen Ausgänge A1 .. A8 zur SPS, die Ausgänge können nicht durch das Betriebssystem oder ein Anwenderprogramm auf dem Compax verändert werden Bemerkung: Wird P223 = 192 gewählt, werden die Ausgänge bei P225 = 0 durch das Betriebssystem benutzt, um Handshake-Signale anzuzeigen	255
P224	Zuordnung der digitalen Ausgänge A9 .. A16** zur SPS, die Ausgänge können nicht durch das Betriebssystem oder ein Anwenderprogramm auf dem Compax verändert werden	255

* und **) Compax 1000SL hat jeweils nur 8 physische Ein- und Ausgänge, die bei der Konfiguration mit dem Servomanager den jeweiligen 16 virtuellen Ein- und Ausgängen zugeordnet werden. Bei allen anderen Geräten der Reihe Compax-M/S gibt es eine Eins-zu-Eins-Zuordnung zwischen virtuellen und physischen Ein- und Ausgängen.

3.2.7 COMPAX-Parameter P227 und P229: Konfiguration einer Stillstandsanzeige

Mit P227 = 2 kann der Ausgang A2 (und seine Umsetzung im STATUSWORT) als Stillstandsanzeige verwendet werden, die vom Sollwert (Sollgeschwindigkeit, nicht Istgeschwindigkeit) abgeleitet wird. Mit P229 wird dazu ein Stillstandsfenster definiert, wobei sich der Wert in P229 als Promille-Wert von der Nenndrehzahl P104 [U/min] darstellt.

4 Funktionsbaustein fbCPXManager

4.1 Projektübersicht

4.1.1 Funktionen

Mit dem Funktionsbaustein *fbCPXManager* erfolgt die komplette Steuerung für Geräte mit der Technologie-Variante x00 (Standard) für alle Geräte der Reihe Compax-M/S mit F3-Option (Profibus) wie CPX1000SL, CPX2500S, CPX4500/8500SL, CPX100/200/500/1500M und CPX3500M.

Funktionen

- Positionieren absolut und relativ, Vorgabe von Position/Distanz, Geschwindigkeit, Beschleunigung und Verzögerung, zeitoptimal durch inkrementelle Übertragung, da nur geänderte Profilwerte übertragen werden,
- Referenzfahren,
- Quittieren,
- Tippen +/-,
- Parameterübertragen über BKD-Kanal für beide Richtungen, maximal 5 Parameter in einem Block,
- Initialisierung Baustein
- Dauersetzen Compax-Ausgänge über zyklischen Kanal (PZD/PAD),
- Daueranzeige Compax-Eingänge über zyklischen Kanal (PZD/PED),
- Daueranzeige der aktuellen Compax-Position über zyklischen Kanal (PZD/PED),

- Automatische Compax-Fehleranzeige mit Fehlernummer.

Überwachungen und Statuswerte

- Parametrierbare Laufzeitüberwachung für Positionierungen und die Referenzfahrt,
- Zusätzliche In-Positions-Überwachung bei absoluten Positionierungen neben der eigentlich Inpositions-Meldung durch In-Fenster-Überwachung,
- Läuft-, Fertig- und Fehlermeldungen für Positionierungen und Referenzfahrten,
- Compax-Fehlermeldung mit aktuellem Fehler-Code,
- aktuelle Compax-Position, Zustandsanzeige der Compax-Eingänge,
- Kommunikationsfehleranzeige.

4.1.2 Software-Einheiten

<i>Einheit</i>	<i>Datei/Baustein</i>	<i>Bedeutung</i>
Projekt	OSRC2Vxy.zip	Projekt C2 Version xy = Versionsbezeichnung
Software-Einheiten	FB10 fbCPXManager	FB fbCPXManager
	OB1, FB100, DB100	Instanziierung und Aufruf von fbCPXMa- nager
	VAT_ErrorNo, VAT_GetSetParams, VAT_Homing, VAT_Positioning	Variablen tafeln zum Steuern von fbCPXManager
	UDT10 stControlType	Typ für STEURWORT
	UDT11 stStatusType	Type für STATUSWORT
erforderliche Standard- Funktionen- und Funkti- onsbausteine im AG	SFC14, SFC15, SFB3, SFB4	SFC14 „DPRD_DAT“ SFC15 „DPWR_DAT“ SFB3 „TP“ SFB4 „TON“

4.1.3 Technische Parameter fbCPXManager

S7-300/400

- MC7-Code: 7.076 kByte
- ab Step7-Version 5.2 + SP1
- GSD-Datei CPX0ee95.GSD für Compax-M/S-Servoregler, BPO-Typ 1 (BKD/BZD oder BKD/BZD E/A getrennt)

Compax-M/S

- Compax-Firmware ab V6.20, F03-Firmware ab V1.42
- Notwendige Parametereinstellungen im Compax-Parametereditor (Servomanager)
 - P191 = 1 Busfehlererkennung aktiviert
 - P193 = 0 Keine zusätzlichen
 - P194 = n Profibus Adresse
 - P196 = n 104/108 BPO1 (BKD/PZD)
 - P135 = 1229056 PED, Lageist
 - P136 = P137 = 0
 - P138 = 1236224 PED, Eingangswort
 - P139 = 1236992 PAD, Ausgangswort

P140 = P141 = P142 = 0	
P221 = 63	Abschalten der Funktion der digitalen Eingänge E1..E6
P223 = 192	A7/A8 über Ausgangswort für SPS zugreifbar
P224 = 255	A9..A16 über Ausgangswort für SPS zugreifbar
P225 = 0	A1..A6 Ansteuerung über Betriebssystem Compax
P227 = 2	A2 als Stillstandsanzeige
P229 = nn	Stillstandsfenster in Promille von Nenndrehzahl P104 [U/min]

4.2 FB-Aufruf und Parametrierung fbCPXManager

4.2.1 Anlaufinitialisierung und Aufruf des fbCPXManager



Die Zuordnung der Instanzen von fbCPXManager zu den DP-Slaves vom Typ Compax-MS erfolgt durch die Übergabe der Slot-E/A-Adressen aus dem S7 Hardware-Konfigurator. Diese und andere mögliche Initialisierungen werden im Bereich VAR (statisch) der fbCPXManager-Instanz abgelegt.

Im folgenden Beispiel wurde ein Instanz *Axis1* von fbCPXManager benutzt.

```
// --- Anlaufbit aus Warm-/Kaltstart SPS
UN   M   100.0           // Anlaufmerker =1 für Initialisieren
SPB  NoA1               // bereits initialisiert
// --- Initialisierungen
// --- Profibus-Slots fbCPXManager
L    256                // Slot 1 BKD (lesend)
T    #Axis1.inAxisInMod1Addr
L    266                // Slot 2 PZD (lesend)
T    #Axis1.inAxisInMod2Addr
L    256                // Slot 1 BKD (schreibend)
T    #Axis1.inAxisOutMod1Addr
L    266                // Slot 2 PZD (schreibend)
T    #Axis1.inAxisOutMod2Addr
// --- Vorbereiten von 5 beliebigen Parameternummern für Compax-Parameter
L    1                  // Compax Parameter Nummer 1
T    #Axis1.arParamNo[0]
L    11                 // Compax Parameter Nummer 11
T    #Axis1.arParamNo[1]
L    12                 // Compax Parameter Nummer 12
T    #Axis1.arParamNo[2]
L    15                 // Compax Parameter Nummer 15
T    #Axis1.arParamNo[3]
L    16                 // Compax Parameter Nummer 16
T    #Axis1.arParamNo[4]
L    255                // Ende-Kennzeichen 255 !!!
T    #Axis1.arParamNo[5]
// --- Schutzzeiten
L    T#5S               // 5s Standard, Maximale Positionierzeit
                        // (auch dynamisch setzbar)
T    #Axis1.tvMaxPositioning
L    T#5S               // 5s Standard, Maximale Referenzierzeit
T    #Axis1.tvMaxHoming
// == nur bei Bedarf !!! ab hier
L    T#200MS
T    #Axis1.tvOverlapping // 200 ms Standard, Überlappungsschutz
                        // für Pos/Ref für Erreicht-Meldung
L    T#200MS
T    #Axis1.tvPulse      // 200 ms Standard, Reset oder Break-Puls
                        // (Stoppt Bewegung nach Timed-out)
// == nur bei Bedarf !!! bis hier
// --- Zusätzliche In-Positions-Überwachung bei absoluten Bewegungen
L    1.000000e+000      // 1.0 in Nutzereinheiten Standard,
                        // Positionsfenster für absolutes Positi-
```

```

// onieren
T      #Axis1.reInPosWindowSafety

NoA1: NOP  0

CALL #Axis1
boStartPositioningPin := // pos. Flanke Positionieren
boModeRelAbsPin      := // =0 abs. / =1 rel. Positionieren
boStartHomingPin     := // pos. Flanke Referenzieren
boJogPPin            := // Tippen+
boJogNPin            := // Tippen-
boBreakPin           := // Stop mit Fahrsatzverwerfen
boResetPin           := // Reset Störung Compax
boSetParamsPin       := // Schreibe Compax Parameter (max. 5)
boGetParamsPin       := // Lese Compax Parameter (max. 5)
boInitCPXManager     := M100.0 // =1 initialisieren
rePositionPin        := // Positionierwert Nutzereinheiten VK/NK
reSpeedPin           := // Drehzahl in %
doAccelPin           := // Beschleunigungsrampe in ms
doDecelPin           := // Abbremsrampe in ms
woOutputWordPin      := // Ausgangswort Compax

boPositioningRunsPout := // Flag „Positionierung läuft“
boHomingRunsPout      := // Flag „Referenzieren läuft“
boGetSetParamsDonePout := // Flag „Parameter gelesen/geschrieben“
boErrPositioningPout  := // Flag „Positionierfehler“
boErrHomingPout       := // Flag „Referenzierfehler“
boInPositionPout      := // Flag „In Position“
boStandStillPout      := // Flag „Stillstand“
boHomingOKPout        := // Flag „Referenz gültig“
boErrCompaxPout       := // Flag „Compax-Fehler“
boBKDErrorPout        := // Flag „BKD-Übertragungsfehler“
boErrCommPout         := // Flag „Kommunikationsfehler SFC14/15“
inErrCodeCompaxPout   := // Fehlercode Compax lt. Handbuch
reActPositionPout     := // Status „Aktuelle Position“ VK/NK
woActInputWordPout    := // Status „Eingangswort“

// --- Anlaufbit aus Warm-/Kaltstart SPS
CLR
=      M      100.0 // Anlaufmerker =0 für Initialisiert

```

4.2.2 Compax-Parameterzugriff über fbCPXManager

Mit dem fbCPXManager können bis zu 5 Compax-Parameter gelesen oder geschrieben werden. Die Übergabe an/zum Baustein erfolgt im Real-Format, die Übertragung intern vom/zum Compax im DSP-Format. Es können alle Parameter übertragen werden, die mit „VP“ gültig gesetzt werden können (oder sofort gültig sind). Beim Schreiben wird das Passwort „GOTO620“ zuerst übertragen, dann die Parameter und abschließend der VP-Befehl. Das Bit *boSetParamsPin* ist zu setzen, der Erfolg wird mit *boGetSetParamsDonePout* gemeldet. Gelesen werden können diese Parameter mit *boGetParamsPin*.

Beispiel für das Schreiben von 5 Parametern

```

L      1 // Compax Parameter Nummer 1
T      #Axis1.arParamNo[0]
L      0.0 // Wert für Parameter 1
T      #Axis1.arParamVal[0]
L      11 // Compax Parameter Nummer 11
T      #Axis1.arParamNo[1]
L      1000.0 // Wert für Parameter 11
T      #Axis1.arParamVal[1]
L      12 // Compax Parameter Nummer 12
T      #Axis1.arParamNo[2]
L      -1000.0 // Wert für Parameter 12
T      #Axis1.arParamVal[2]
L      15 // Compax Parameter Nummer 15

```

```

T      #Axis1.arParamNo[3]
L      100.0                // Wert für Parameter 15
T      #Axis1.arParamVal[3]
L      16                  // Compax Parameter Nummer 16
T      #Axis1.arParamNo[4]
L      200.0               // Wert für Parameter 16
T      #Axis1.arParamVal[4]
L      255                 // DYNAMISCHES Ende-Kennzeichen 255 !!!
T      #Axis1.arParamNo[5]
    
```

4.2.3 Parametrierung



Viele Funktionen werden über eine Flankenerkennung ausgeführt, trotzdem sind die Steuerbits nach Gebrauch wieder auf FALSE zu setzen, da z.B. bei aktiviertem Tippen etc. eine Referenzfahrt nicht ausgelöst werden kann.

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
boStartPositioningPin	IN	BOOL	E,A,M,L,D	pos. Flanke startet Positionierbewegung
boModeRelAbsPin	IN	BOOL	E,A,M,L,D	=0 absolute, =1 relative Bewegung
boStartHomingPin	IN	BOOL	E,A,M,L,D	pos. Flanke startet Referenzierbewegung
boJogPPin	IN	BOOL	E,A,M,L,D	Tippen+
boJogNPin	IN	BOOL	E,A,M,L,D	Tippen-
boBreakPin	IN	BOOL	E,A,M,L,D	=1 Stopp (Verfahrssatz werfen), Tippen, Positionieren, Referenzieren werden gestoppt
boResetPin	IN	BOOL	E,A,M,L,D	pos. Flanke quittiert, Achse wird sofort nach Quittieren bei nicht mehr vorhandener Fehlerursache bestromt
boSetParamsPin	IN	BOOL	E,A,M,L,D	pos. Flanke, Compax-Parameter schreiben
boGetParamsPin				pos. Flanke, Compax-Parameter lesen
boInitCPXManager	IN	BOOL	E,A,M,L,D	=1 Initialisieren, kein Refresh von Statuswerten am FB-Ausgang alle anderen binären Eingänge = 0 machen! 1 Zyklus ist ausreichend
rePositionPin	IN	REAL	M,L,D	Position im REAL-Format -4000000 .. +4000000 mit Nachkommastellen (NK)
reSpeedPin	IN	REAL	M,L,D	Speed im REAL-Format in % von P104 (Nenndrehzahl) 0 .. 100 %
doAccelPin	IN	DINT	M,L,D	Beschleunigungsrampe im DINT-Format von 10 ms .. 65000 ms für Rampe von 0 auf P104
doDecelPin	IN	DINT	M,L,D	Abbremsrampe im DINT-Format von 10 ms .. 65.000 ms für Rampe von 104 auf 0
woOutputWordPin	IN	WORD	M,L,D	Ausgangswort für Compax-Ausgänge A1..A16
boPositioningRunsPout	OUT	BOOL	E,A,M,L,D	Flag „Positionierung läuft“
boHomingRunsPout	OUT	BOOL	E,A,M,L,D	Flag „Referenzieren läuft“

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
boGetSetParamsDonePout	OUT	BOOL	E,A,M,L,D	Flag „Parameter übertragen“
boErrPositioningPout	OUT	BOOL	E,A,M,L,D	Flag „Positionierung mit Fehler“
boErrHomingPout	OUT	BOOL	E,A,M,L,D	Flag „Referenzieren mit Fehler“
boInPositionPout	OUT	BOOL	E,A,M,L,D	Flag „In Position“
boStandStillPout	OUT	BOOL	E,A,M,L,D	Flag „Stillstand“ sollwertmäßig
boHomingOKPout	OUT	BOOL	E,A,M,L,D	Flag „Referenz gültig“
boErrCompaxPout	OUT	BOOL	E,A,M,L,D	Flag „Compax-Störung“, ab Version 1.1: der Fehlerausgang geht erst nach 800 ms weg, damit die Endstufe sicher bestromt ist, falls sich der Fehler quittieren ließ
boBKDErrorPout	OUT	BOOL	E,A,M,L,D	Flag „Übertragungsfehler BKD“, Accel, Decel, Speed, Parameter
boErrCommPout	OUT	BOOL	E,A,M,L,D	Flag „Kommunikationsfehler SFC14/15“
inErrCodeCompaxPout	OUT	INT	M,L,D	Fehlercode Compax entsprechend Compax-Handbuch
reActPositionPout	OUT	REAL	M,L,D	Status „Aktuelle Position“
woActInputWordPout	OUT	WORD	M,L,D	Status „Aktuelles Eingangswort Compax“
inAxisInMod1Addr	STATIC	INT	Def = 256	Slot Adresse Modul 1 lesend
inAxisInMod2Addr	STATIC	INT	Def = 266	Slot Adresse Modul 2 lesend
inAxisOutMod1Addr	STATIC	INT	Def = 256	Slot Adresse Modul 1 schr.
inAxisOutMod2Addr	STATIC	INT	Def = 266	Slot Adresse Modul 2 schr.
arParamNo	STATIC	ARRAY [0 .. 5] OF INT	Def = 1, 11, 12, 15, 16, 255	Parameternummern P1 Realnullversatz P11 pos. SW-Grenze P12 neg. SW-Grenze P15 max. Drehzahl von P104 P16 max. Moment
arParamVal	STATIC	ARRAY [0 .. 5] OF INT	Def = 0.0, 0.0, 0.0, 100.0, 200.0	dazu passende Werte, müssen überschrieben werden
tvMaxPositioning	STATIC	TIME	Def = T#5s	Maximalzeit für Positionierungen, kann dynamisch überschrieben werden
tvMaxHoming	STATIC	TIME	Def = T#5s	Maximalzeit für Referenzierungen, kann dynamisch überschrieben werden
reInPosWindowSafety	STATIC	REAL	Def = 1.0	zusätzliches In-Positions-Fenster symmetrisch um die Sollposition, nur für absolute Positionierungen

5 Bemerkungen zu Sicherheit und Signalspiel

- Auf dem PAD befinden sich keine Lagesollwerte, so dass es bei Zu- und Abschaltungen des SPS-Profibus-Masters nicht zu unerwünschten Bewegungen kommen kann. Alle Profilwerte und die Position / Distanz werden über den BKD-Kanal übertragen.
- Absolute Positionierungen werden zusätzlich über einen Soll-/Istwertvergleich abgesichert, daher muss der aktuelle Sollwert in *rePositionPin* stabil bis zur Positions-Erreicht-Meldung anstehen, wenn eine absolute Positionierung ausgeführt wird.

- Um ein Durchlaufen von Anwenderschrittketten auch bei relativen Bewegungen (POSR, *boModeRelAbsPin = true*) und bei der Referenzierung zu vermeiden (das Positionerreicht-Signal von Compax geht bedingt durch z.B. eine große Positionierzone in P14 erst später weg oder geht auf Grund einer kurzen Positionierdistanz überhaupt nicht weg oder der S7-SPS-Zyklus ist sehr schnell) werden sogenannte Überlappungsschutzzeiten eingeführt, die die Zeit zwischen Start einer Positionierung / Referenzierung und dem Positionerreicht-Signal / Referenziert-Signal nur dann verlängern, wenn ein Positionierung / Referenzierung ausgeführt wird, deren Bewegung startet und schneller als die Überlappungsschutzzeit abgeschlossen ist (Standard 200 ms).
- Alle Positionierungen und Referenzierungen werden über Zeitglieder (IEC-Timer) überwacht. Will man das nicht, muss der Wert entsprechend groß gewählt werden.
- Nach Anstoßen einer Positionierung auf *boPositioningRunsPout* achten, erst wenn die Flag aktiv ist, wird die Positionierung tatsächlich ausgeführt (pos. Bestätigung des POSR/POSA-Befehls).
- Es ist möglich, Parameter an Compax zu übertragen oder von Compax zu lesen. Mit P1, P11, P12 kann über Lesen, Korrigieren und anschließendes Schreiben ein Zahnriemensprung einfach vom Operator über die SPS korrigiert werden. Alternativ kann man eine Referenzierung ohne Geber-Null-Erkennung benutzen.
- Alle Compax-Fehler werden über das Bit *boErrCompaxPout* gemeldet, dazu wird sofort der Wert *inErrCodeCompaxPout* aktualisiert, =0 kein Fehler.

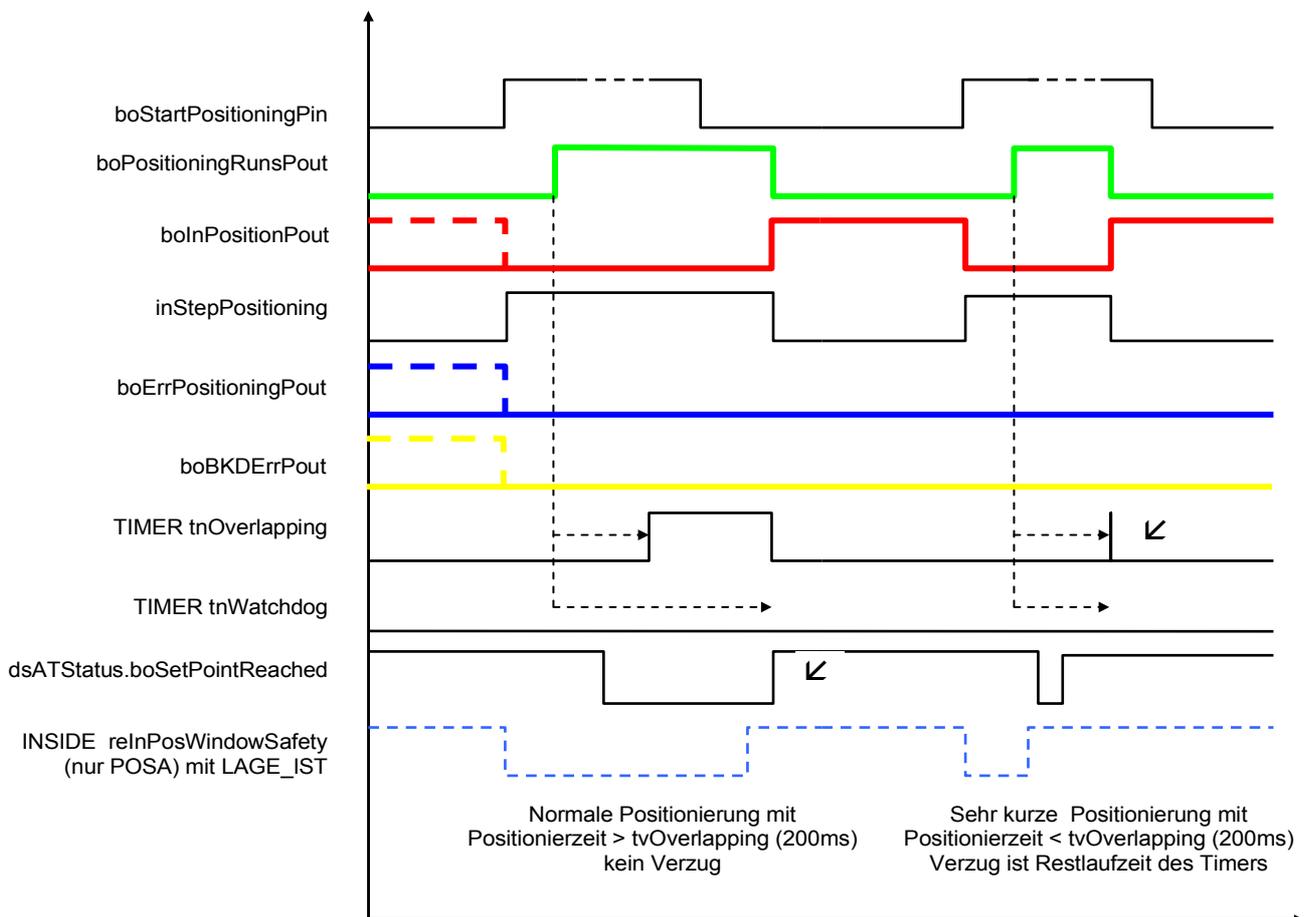


Abbildung 2 Signalspiel POSR/POSA ohne Fehler

Das Inpositions-Signal wird folgendermaßen abgeleitet:

```

dsATStatus.boSetPointReached // Statuswort – Position erreicht
UND
inStepPositioning = 0 // Schrittkette Positionieren nicht aktiv
UND
(rePositionPin – reInPosWindowSafety) <= reActpositionPout <= (rePositionPin + reInPosWindowSafety)
// Soll-/Istvergleich, nur bei absoluten Positionierungen
    
```

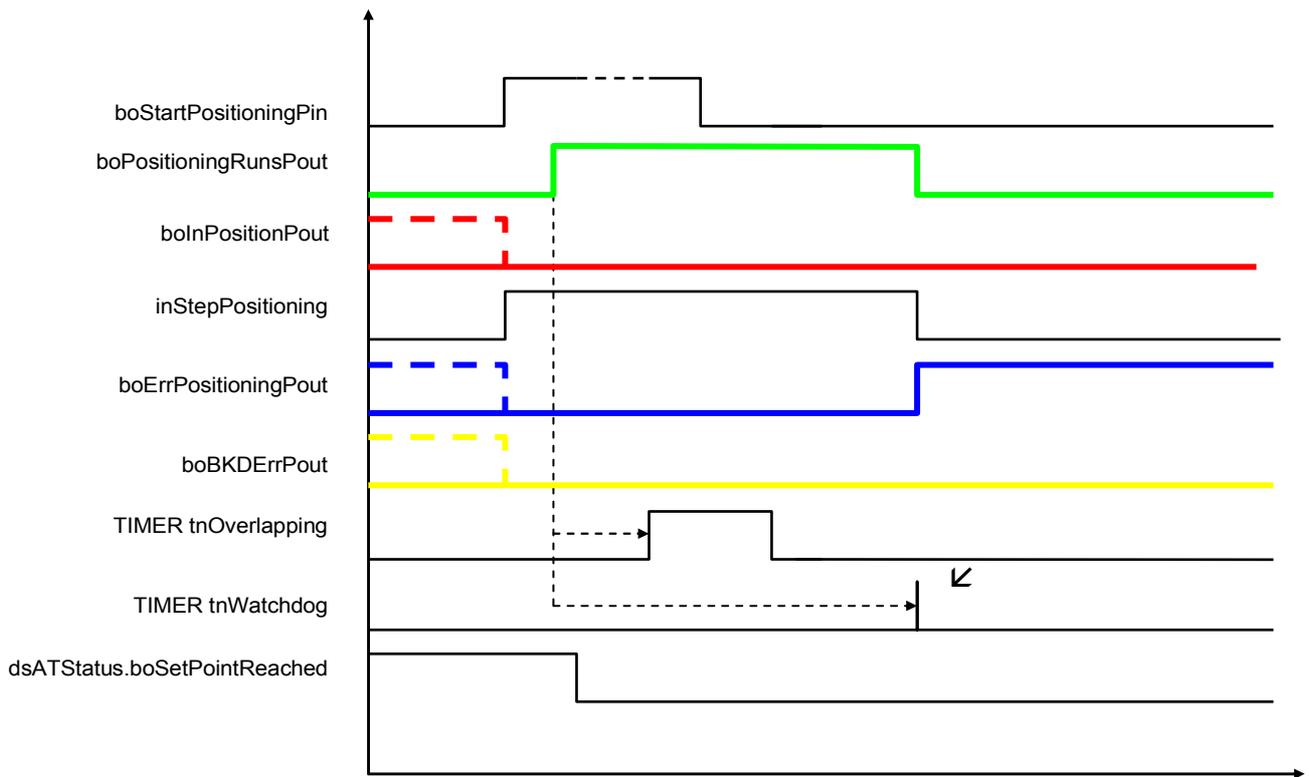


Abbildung 3 Signalspiel POSR mit Timed-Out-Fehler

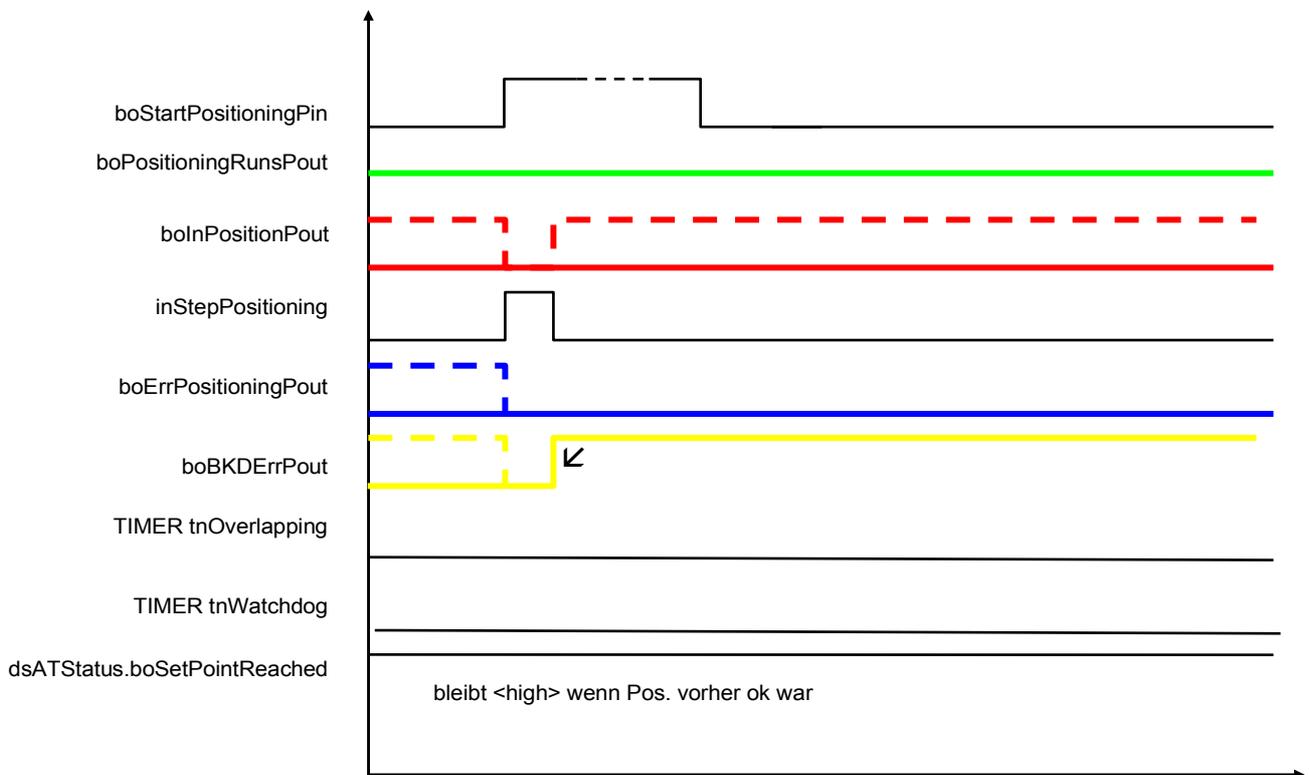


Abbildung 4 Signalspiel mit POSR mit BKD-Fehler (Übertragungsfehler)

6 Referenzen

[1] Produkthandbuch Compax komplett: Parker/Hauser-Art.-Nr.: 190-040053, Serie N02

[2] Profibus-Beschreibung / Option F3: Parker/Hauser-Art.-Nr.: 190-040036, Serie N07