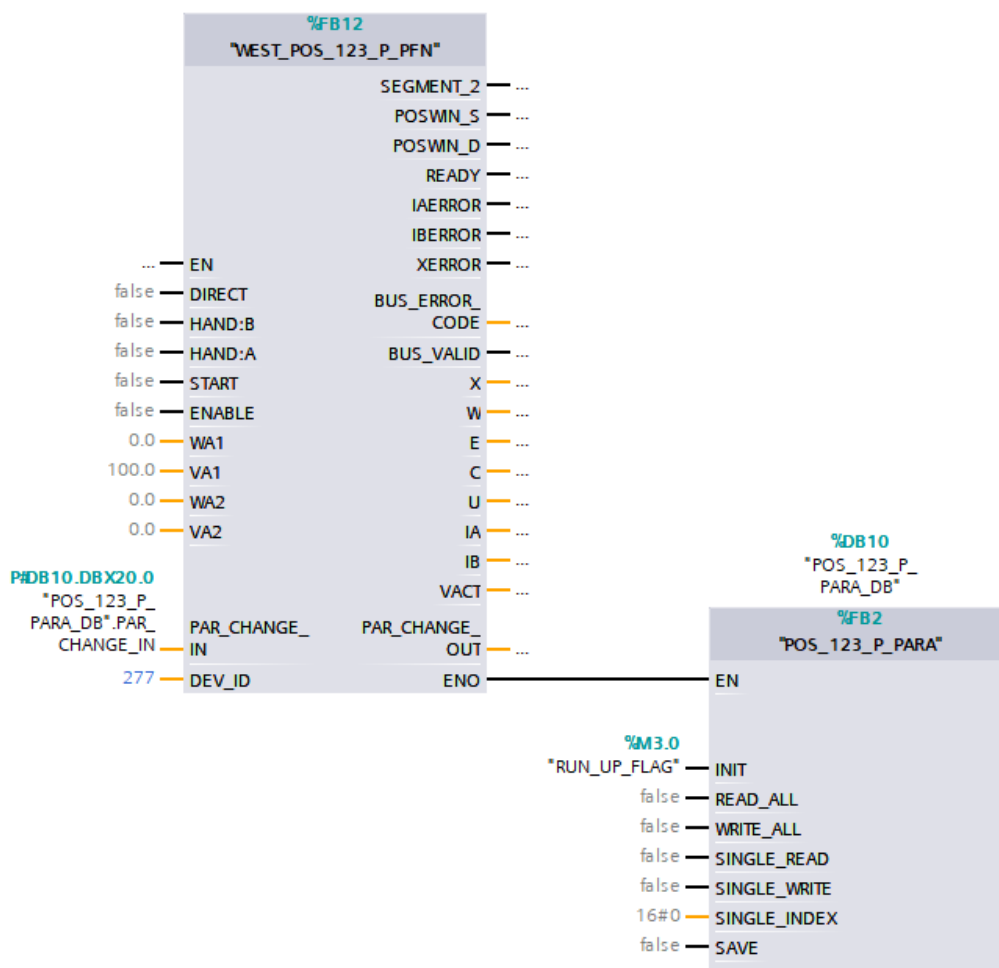


## Dokumentation

### POS\_123\_P\_PARA

Parametrierbaustein für den Positionsregler POS-123-P-PFN  
 Zur Verwendung mit Siemens-SPSen (S7-1200 / 1500 / 300 / 400)



*Electronics  
 Hydraulics meets  
 meets Hydraulics  
 Electronics*

## Inhaltsverzeichnis

1	Allgemeine Informationen.....	3
1.1	Zugehörendes Modul.....	3
1.2	Benötigte Software und Hardware.....	3
1.3	Impressum.....	3
2	Eigenschaften.....	4
3	Einbau in das Automationsprojekt.....	4
4	Funktionsweise.....	6
5	Praktische Handhabung.....	8
5.1	Variante 1: Alleinige Parametrierung über die SPS.....	8
5.2	Variante 2: Sicherung eines am Regler eingestellten Parametersatzes in der SPS.....	9
5.3	Integrität der Parametrierung.....	9

## 1 Allgemeine Informationen

### 1.1 Zugehörendes Modul

**POS-123-P-PFN** - Positioniersteuerung mit integrierter Leistungsendstufe, analoger Sensorschnittstelle und Profinet Schnittstelle

### 1.2 Benötigte Software und Hardware

Der hier beschriebene Programmbaustein ist zum Einsatz in der speicherprogrammierbaren Steuerung der Firma Siemens vorgesehen. Zur Projektierung wird die Programmierumgebung „TIA-Portal“, mindestens Version 13 benötigt.

Zusätzlich benötigt wird der generelle S7-Treiberbaustein (WEST\_POS\_123\_P\_PFN), Einbindung siehe Anleitung des Moduls, sowie eine Profinet-fähige SPS.

### 1.3 Impressum

**W.E.St.** Elektronik GmbH

Gewerbering 31  
41372 Niederkrüchten

Tel.: +49 (0)2163 577355-0  
Fax.: +49 (0)2163 577355 -11

Home page: [www.w-e-st.de](http://www.w-e-st.de)  
EMAIL: [contact@w-e-st.de](mailto:contact@w-e-st.de)

Datum: 27.01.2025

Die hier beschriebenen Daten und Eigenschaften dienen nur der Produktbeschreibung. Der Anwender ist angehalten, diese Daten zu beurteilen und auf die Eignung für den Einsatzfall zu prüfen. Eine allgemeine Eignung kann aus diesem Dokument nicht abgeleitet werden. Technische Änderungen durch Weiterentwicklung des in dieser Anleitung beschriebenen Produktes behalten wir uns vor. Die technischen Angaben und Abmessungen sind unverbindlich. Es können daraus keinerlei Ansprüche abgeleitet werden.

Dieses Dokument ist urheberrechtlich geschützt.

## 2 Eigenschaften

Ergänzender TIA-Portal Baustein für den Treiber des Positionsreglers

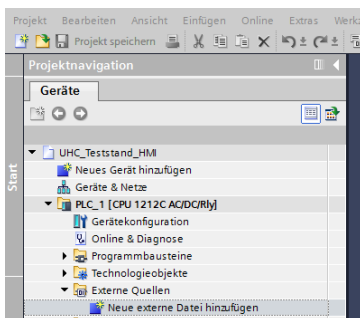
1. Funktion zum Lesen und Übertragen ganzer Datensätze (alle les- und schreibbaren Parameter)
2. Funktion zum Verändern einzelner Parameter während des Betriebs
3. Funktion zum Lesen einzelner Parameter

Diese Funktionen sind in einem Baustein zusammengefasst.

## 3 Einbau in das Automationsprojekt

Der Einbau des Grundbausteins „WEST\_POS\_123\_P\_PFN“ ist in der Dokumentation des Moduls beschrieben.

- 1.) Auch der Parametrierbaustein wird als SCL-Quelle zur Verfügung gestellt. Zum Einbau in das Projekt muss diese Datei im TIA-Portal als „neue externe Datei“ hinzugefügt werden:

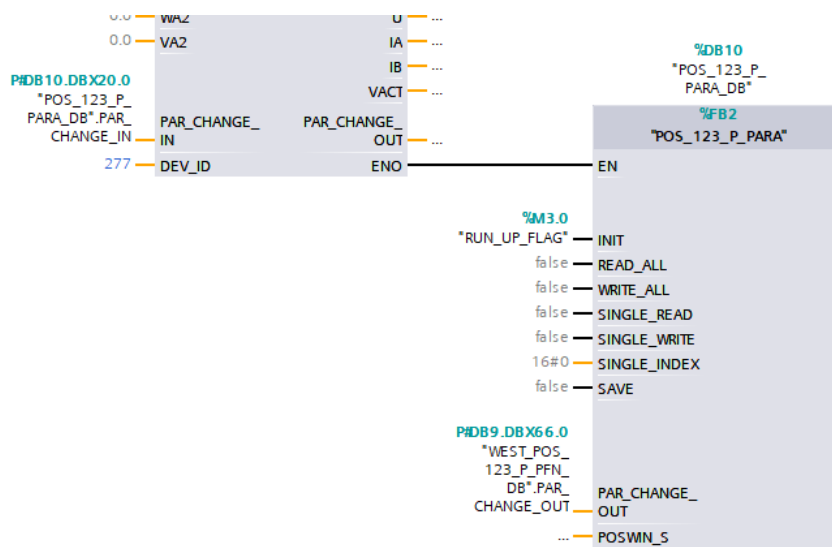


- 2.) Anschließend auf die importierte Datei mit der rechten Maustaste klicken und die Option „Bausteine aus Quelle generieren“ wählen. Nach der Übersetzung steht der Treiberbaustein im Bausteinordner zur Verfügung.

Dieser FB kann nun im Anwenderprogramm aufgerufen werden. Dies sollte in einem Weckalarm-OB mit einer Zykluszeit  $\geq 8$  ms geschehen. Empfohlen wird eine möglichst zügige Bearbeitung bis max. 50 ms, damit die in mehreren Zyklen (asynchron) ablaufende Übertragung nicht zu lange dauert.

Ein Einbau im freien Zyklus (OB1) ist nicht möglich!

Da der Parametrierbaustein die Zugriffsmechanismen des Treiberbausteins nutzt, sind die beiden Bausteine miteinander zu verschalten. Zudem ist der Aufruf im gleichen übergeordneten Baustein sinnvoll. Die folgende Darstellung zeigt beispielhaft eine Variante zur Verknüpfung unter FUP. Die notwendige Kopplung geschieht durch Anbinden der zugehörigen Strukturausgangsvariable aus dem Instanz-DB des jeweils anderen Bausteins an den Eingangsparameter. Es sind auf diese Weise die Anschlüsse „PAR\_CHANGE\_IN“ und „PAR\_CHANGE\_OUT“ zu verbinden. Diese Namen beziehen sich auf die Sicht des Treiberbausteins hinsichtlich Ein- und Ausgang. Somit heißt der Ausgang des Parametrierbausteins „PAR\_CHANGE\_IN“, da er mit diesem Eingang des Treibers verbunden wird.



Aufgrund der Handhabung des Instanz-DB des Parameterbausteins als Abbild der Einstellwerte ist es nicht zu empfehlen, diesen in Form einer Multiinstanz anzulegen.

An den Eingang „INIT“ ist eine Variable anzubinden, die beim Anlauf der Steuerung mindestens während eines Bausteinaufrufs „TRUE“ wird. Hierzu kann man z.B. einen Merker verwenden, den man im OB100 setzt und nach der Bearbeitung des Bausteins wieder zurücksetzt.

Die Parameter des Positionsreglers sind als EA-Parameter des Bausteins im Instanz-DB angebunden.

Es gibt für den Anwender zwei alternative Verwendungsmöglichkeiten:

- Einstellen und Beobachten der Werte in der Online-Ansicht des Instanz-DB im TIA-Portal
- Verknüpfen der Werte mit EA-Feldern einer WinCC, WinCC-flexible oder anderen HMI-Anwendungen.

## 4 Funktionsweise

Die folgenden Ein- und Ausgänge dienen der Steuerung der Funktion:

Steuerung POS_123_P_PARA		
Eingang / Ausgang	Name	Funktion
E	INIT	Baustein initialisieren: Die aktuellen Parameter und deren Grenzen werden aus dem Positionsregler ausgelesen und in einem internen Speicherbereich des Bausteins abgelegt.
E	READ_ALL	Eine steigende Flanke an diesem Eingang bewirkt, dass alle momentan eingestellten Parameter des Moduls ausgelesen werden und in die Parameter der Bausteinschnittstelle übertragen werden (EA-Bereich, siehe unten).
E	WRITE_ALL	Eine steigende Flanke an diesem Eingang bewirkt, dass die Parameter der Bausteinschnittstelle (EA-Bereich, siehe unten) in das Regelmodul übertragen werden.
E	SINGLE_READ	Die „SINGLE_..“ Eingänge sind für den Zugriff auf einzelne Werte vorgesehen. Eine steigende Flanke an diesem Eingang bewirkt, dass der Parameter, dessen Index gemäß der Parameterliste (Reglerdoku) am Eingang „SINGLE_INDEX“ angegeben ist, aus dem Regler ausgelesen wird. Der Wert wird unter „SINGLEACCESS“ (s.u.) ausgegeben.
E	SINGLE_WRITE	Eine steigende Flanke an diesem Eingang bewirkt, dass der Parameter, dessen Index gemäß der Parameterliste (Reglerdoku) am Eingang „SINGLE_INDEX“ angegeben ist, in den Regler geschrieben wird. Der Wert ist unter „SINGLEACCESS“ (s.u.) anzugeben.
E	SAVE	Eine steigende Flanke an diesem Eingang bewirkt, dass die Parameter im Regler dauerhaft gespeichert werden. Achtung: Nicht automatisch ansteuern, da sonst die Zahl der Schreibzyklen für das EEPROM zu groß werden könnte.
E	SINGLE_INDEX	Index für den Einzelzugriff
EA	„Parametername“	Abbild der Werte auf dem Regler bzw. zu schreibender Parametersatz. Die Kommentare zu den einzelnen Werten geben ausführlichere Information. Aufgrund der Sprachrestriktionen sind Sonderzeichen im Namen nicht möglich. Deshalb sind die Namen z.T. leicht modifiziert. Auswahlparameter werden als Zahlenwert angegeben. Die Auswahl eines ungültigen Wertes führt zu einer Fehlermeldung beim Schreiben.
EA	SINGLEACCESS	Parameter, der mittels der „SINGLE“-Kommandos geschrieben werden soll oder gelesen wurde.
A	STATUS	Der Zustand des Bausteins wird hier ausgegeben: 0 = inaktiv, 1 = bereit und initialisiert, 2 = beschäftigt, 3 = Fehler Während der Befehlsverarbeitung zeigt der Baustein über einige Zyklen „2“ = beschäftigt an. Danach wechselt der Status bei erfolgreicher Ausführung wieder auf „1“ = bereit. Sollte ein Fehler während der Bearbeitung auftreten, oder der Zugriff nicht erlaubt sein (z.B. ENABLE + WRITE_ALL) wechselt der Zustand auf „3“. Diese Meldung bleibt bis zur nächsten Befehlsausführung bestehen.

A	ERROR_LINE	Bei Parameterfehlern (Bereichsverletzung oder Timeout beim Schreiben) wird die Nummer des betroffenen Parameters hier ausgegeben.
A	DISC_LINE	Sollte es eine Abweichung zwischen den Offline-Parametern (an den EAs dieses Bausteins) und den Werten im Regler geben, wird hier die Nummer des ersten abweichenden Wertes angezeigt. Diese Funktion erlaubt geringfügige Abweichungen, die sich durch Rundungsfehler beim Speichern der Werte im Modul ergeben können.
A	DISC_COUNT	Anzahl der Abweichungen. Wird bei nicht erfolgreicher Überwachung (z.B. Bus inaktiv) auf „-1“ gesetzt.

## 5 Praktische Handhabung

### 5.1 Variante 1: Alleinige Parametrierung über die SPS

Nachdem das Modul angeschlossen wurde und die ProfiNet-Verbindung steht, liest der Baustein die Parameter des Moduls und deren Grenzen in einen internen Speicherbereich. Da noch keine sinnvollen Werte an den EA-Parametern stehen, werden zunächst alle Parameter als abweichend angesehen:

Output	Parameter	Datentyp	Min	Max	Startwert	Abweichend	...	...	...	...	...	...	...
11	STATUS	Int	12.0	0	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
12	ERROR_LINE	Int	14.0	0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		0 = no active, 1 = ready and initialize
13	DISC_LINE	Int	16.0	0	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		number of the faulty parameter (range)
14	DISC_COUNT	Int	18.0	0	32	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		line of 1st deviation / Zeile mit der er
15	PAR_CHANGE_IN	Struct	20.0			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		number of deriving parameters / Zahl
													structure input for parameter change

Es kann nun ein Auslesen der Default Parameter erfolgen (steigende Flanke „READ\_ALL“). Nach einigen Sekunden Wartezeit sind die Werte angepasst und können angezeigt werden:

Output	Parameter	Datentyp	Min	Max	Startwert	Abweichend	...	...	...	...	...	...	...
1	STATUS	Int	12.0	0	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
2	ERROR_LINE	Int	14.0	0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
3	DISC_LINE	Int	16.0	0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
4	DISC_COUNT	Int	18.0	0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
5	PAR_CHANGE_IN	Struct	20.0			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
6	PARVAL	DWord	20.0	16#0	16#0000_0000	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
7	PARAMODE	Bool	24.0	false	FALSE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
8	ADDRESS	Word	26.0	16#0	16#0000	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
9	PARAVALID	Bool	28.0	false	FALSE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
10	PARAREAD	Bool	28.1	false	FALSE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
11	READULIM	Bool	28.2	false	FALSE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
12	READLLIM	Bool	28.3	false	FALSE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
13	InOut					<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
14	POSWIN_S	Dint	30.0	0	200	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
15	POSWIN_D	Dint	34.0	0	200	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
16	OFFSET_X	Dint	38.0	0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
17	SENS	Int	42.0	0	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
18	EOUT	Int	44.0	0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
19	HAND_A	Int	46.0	0	3333	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
20	HAND_B	Int	48.0	0	-3333	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
21	SYS_RANGE	Int	50.0	0	100	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		

Nun erfolgt zunächst die Einstellung der Werte, die aufgrund der Auslegung oder aus Datenblättern bekannt sind, wie z.B. SYS\_RANGE, N\_RANGE:X, CURRENT...

Diese Werte werden anschließend mit dem Kommando „WRITE\_ALL“ auf das Modul übertragen.

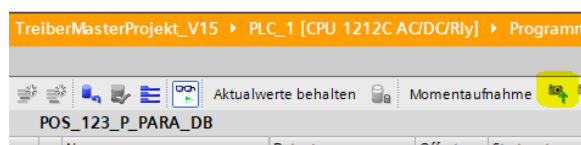
Andere Werte sind im Betrieb anzupassen, wie z.B. D:A / D:B, MIN:A / MIN:B, OFFSET.

Während der Optimierung wird hierzu zumeist auf einzelne Parameter zugegriffen, dies kann über SINGLE\_WRITE, über WPC oder alternativ auch durch die Übertragung des gesamten Parametersatzes über „WRITE\_ALL“ geschehen.

Es ist jedoch zu beachten, dass bei Einzelzugriff kein Abgleich des internen Online-Abbildes erfolgt.

Also muss man zu dessen Aktualisierung zum Schluss über den „READ\_ALL“-Befehl des kompletten Parametersatz auslesen.

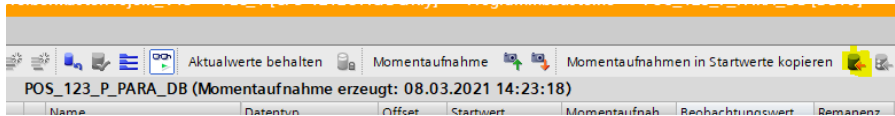
Sobald alle Parameter wunschgemäß eingestellt sind, sollte der Zustand im TIA / S7-Projekt offline gesichert werden. Hierzu in die Online-Ansicht des Bausteins gehen und eine Momentaufnahme der Beobachtungswerte machen:





Die ausgelesenen Werte werden nun in einer weiteren Spalte angezeigt.

Über die Schaltfläche „Momentanwerte in Startwerte kopieren“ werden die Parameter in die Spalte „Startwert“ übernommen:



Anschließend das TIA-Portalprojekt speichern und den geänderten Baustein aus dem Bausteinordner noch einmal auf die Steuerung laden.

Auf diese Weise sind die eingestellten Werte sowohl auf der Steuerung als auch im Offline-Dateiordner gesichert, jedoch nicht im EEPROM des Moduls. Dies sollte nach Abschluss der Einstellarbeiten erfolgen, damit der Positionsregler nach Spannungswiederkehr die aktuellen Parameter ohne nochmaliges Beschreiben aus der Steuerung verwendet.

Das Speichern im Regler wird durch eine steigende Flanke des Eingangssignals an „SAVE“ ausgelöst. Dies darf nicht zyklisch aus dem Programm geschehen, um nicht die maximalen Schreibzyklen des Gerätes zu überschreiten.

## 5.2 Variante 2: Sicherung eines am Regler eingestellten Parametersatzes in der SPS

Es ist auch möglich, die Einstellungen am Regler komplett mit dem Programm WPC durchzuführen und anschließend eine Sicherung des Parametersatzes auf der Steuerung und dem Offline-Programmordner zu erstellen.

Sollte der Regler getauscht werden müssen, ist es einfach möglich diesen Datensatz durch die Befehlssequenz „WRITE\_ALL“ gefolgt von „SAVE“ auf den neuen Regler zu übertragen.

Zum Anlegen der Sicherung:

- READ\_ALL durchführen
- Momentaufnahme der Beobachtungswerte (s.o.)
- Momentanwerte als Startwerte übernehmen
- TIA-Projekt speichern

## 5.3 Integrität der Parametrierung



Für die ordnungsgemäße Funktion des Positioniermoduls ist eine korrekte Parametrierung von fundamentaler Bedeutung. Eine falsche oder unvollständige Parametrierung kann zu gravierenden Fehlfunktionen führen.

Es ist daher wichtig, die Integrität der Parameter zu überwachen. So ist z.B. nach einem Schreibvorgang „WRITE\_ALL“ zu prüfen, ob dieser erfolgreich durchgeführt wurde (STATUS = 1) und es keine Abweichungen mehr gibt (DISC\_COUNT = 0). Falls man die Übertragung des gesamten Parametersatzes von der SPS aus automatisch durchführen möchte, ist eine entsprechende Kontrolle nach dem Schreiben im Anwenderprogramm der SPS durchzuführen, bevor die Regelfunktion freigegeben wird.