

Feldbusschnittstelle WSG

Softwareanleitung - Firmware Version 4.0.x



Impressum

Urheberrecht:

Diese Anleitung bleibt urheberrechtlich Eigentum der SCHUNK GmbH & Co. KG. Sie wird nur unseren Kunden und den Betreibern unserer Produkte mitgeliefert und ist Bestandteil des Produktes. Ohne unsere ausdrückliche Genehmigung dürfen diese Unterlagen weder vervielfältigt noch dritten Personen, insbesondere Wettbewerbsfirmen, zugänglich gemacht werden.

Technische Änderungen:

Änderungen im Sinne technischer Verbesserungen sind uns vorbehalten.

Dokumentenummer: 1011119

Auflage: 01.00 | 17.08.2017 | de

© SCHUNK GmbH & Co. KG

Alle Rechte vorbehalten

Sehr geehrter Kunde,

wir gratulieren zu Ihrer Entscheidung für SCHUNK. Damit haben Sie sich für höchste Präzision, hervorragende Qualität und besten Service entschieden.

Sie erhöhen die Prozesssicherheit in Ihrer Fertigung und erzielen beste Bearbeitungsergebnisse – für die Zufriedenheit Ihrer Kunden.

SCHUNK-Produkte werden Sie begeistern.

Unsere ausführlichen Montage- und Betriebshinweise unterstützen Sie dabei.

Sie haben Fragen? Wir sind auch nach Ihrem Kauf jederzeit für Sie da.

Mit freundlichen Grüßen

Ihre SCHUNK GmbH & Co. KG

Spann- und Greiftechnik

Bahnhofstr. 106 – 134

D-74348 Lauffen/Neckar

Tel. +49-7133-103-0

Fax +49-7133-103-2399

info@de.schunk.com

www.schunk.com



Reg. No. 003496 QM08



Reg. No. 003496 QM08

Inhaltsverzeichnis

1	Einleitung	4
2	Schnittstellen	5
2.1	PROFIBUS.....	5
2.1.1	Installation der GSD-Datei.....	5
2.1.2	Konfiguration.....	6
2.2	PROFINET	6
2.2.1	Installation der GSDML-Datei	6
2.2.2	Konfiguration.....	7
2.3	Modbus/TCP	7
2.3.1	Konfiguration.....	8
3	Schnittstellenbeschreibung	9
3.1	Ausgaberegister (SPS zu WSG)	9
3.2	Eingaberegister (WSG zu SPS)	12
3.3	Diagnosemitteilungen PROFIBUS	16
4	Befehle.....	17
4.1	MOVE – Bewegen der Greiferfinger im Positionierungsmodus	17
4.2	GRIP – Greifen eines Teils.....	18
4.3	RELEASE – Loslassen eines Teils	20
4.4	HOMING – Referenzieren der Greiferfinger	21
4.5	STOP/ACK – Stoppen der Bewegung oder Quittieren eines FASTSTOP.....	22
4.6	FASTSTOP – Auslösen eines Fast Stop.....	23
4.7	JOG+ und JOG- – Jog Modus.....	24
5	Feldbusmonitor	26
5.1	Anhang A: Statuscodes.....	27
5.2	Anhang B: Systemstatus-Flags.....	29
5.3	Anhang C: Greifzustände	32
5.4	Anhang D: Demoprogramm	34

1 Einleitung

Die Greifer der WSG-Serie besitzen Schnittstellen für PROFIBUS DP V0 und/oder PROFINET, abhängig vom Gerätetyp.

PROFIBUS ist ein weit verbreitetes Feldbus-Protokoll in der industriellen Automation. Es unterstützt Einzel- und Multimastermodi.

PROFINET ist eine neue Generation von Feldbus-Schnittstelle, speziell entwickelt für Echtzeitkommunikation über Standard-Ethernet-Schnittstellen.

Jedes Gerät wird durch einen E/A-Registersatz repräsentiert, welcher periodisch mit dem PROFIBUS-Master oder einem PROFINET-Controller (z.B. SPS) synchronisiert wird.

Diese Anleitung setzt Kenntnisse über PROFIBUS- und/oder PROFINET-Technologien und der Siemens SIMATIC-Software voraus.

HINWEIS

PROFINET und Modbus/TCP sind optionale Features. Lizenzschlüssel können separat über SCHUNK bezogen werden.

HINWEIS

Auf der WSG-Weboberfläche befinden sich im Downloadbereich einfache Demoprogramme für Siemens SIMATIC S7-1200, ([☞ 5.4, Seite 34](#)).

2 Schnittstellen

2.1 PROFIBUS

Jeder PROFIBUS-Slave hat einen E/A-Registersatz, welcher periodisch ausgetauscht und vom PROFIBUS-Master gelesen bzw. geschrieben wird. Der E/A-Bereich des Greifers ist auf der Masterseite mit dem Geräteprofil (GSD-Datei) konfiguriert, wie es von der Produkt-CD oder unter der WSG-Weboberfläche heruntergeladen werden kann.

Weiterführende Informationen zu den einzelnen E/A-Register, ([☞ 3, Seite 9](#))

2.1.1 Installation der GSD-Datei

HINWEIS

Die GSD-Datei kann in Siemens STEP7 v11.0 (TIA) und neuer installiert werden.

Die GSD-Datei ist ein komprimiertes ZIP-Archiv, welches folgende Dateien enthält:

- WEIS5555.gsd (Gerätebeschreibungsdatei)
- WSG_D.bmp (Bilddatei)
- WSG_R.bmp (Bilddatei)
- WSG_S.bmp (Bilddatei)
- install.txt (Installationshinweise)

Die Installation der GSD-Datei in Siemens STEP7 11.0 folgendermaßen durchführen:

- 1 Zip-Archiv auf die Festplatte entpacken.
 - 2 In Siemens TIA die Projektansicht öffnen.
 - 3 Unter *Extras* -> *Gerätebeschreibungsdatei (GSD) installieren* wählen.
 - 4 Zu dem Verzeichnis, in welches die Dateien entpackt wurden wechseln und die GSD-Datei wählen.
- ⇒ Im Gerätecatalog unter *Weitere Feldgeräte*-> *PROFIBUS DP* -> *Drives* -> *Weiss Robotics GmbH & Co. KG* wird der Greifer angezeigt.

2.1.2 Konfiguration

Um die PROFIBUS-Schnittstelle benutzen zu können, muss diese zuerst über die Weboberfläche (*Settings -> Command Interface*) eingeschaltet werden. Die PROFIBUS-Teilnehmeradresse ist auf "7" voreingestellt, kann aber ebenfalls über die Weboberfläche geändert werden.

Weitere Informationen befinden sich in der Montage- und Betriebsanleitung des Greifers.

2.2 PROFINET

Die PROFINET-Schnittstelle verwendet dieselben E/A-Register wie die PROFIBUS-Schnittstelle. Wie bei PROFIBUS wird auch hierbei der E/A-Registersatz periodisch zwischen Greifmodul und PROFINET-Controller ausgetauscht. Der E/A-Registersatz verwendet ein auf dem Controller installiertes, vordefiniertes Geräteprofil (GSDML-Datei), welches von der Produkt-CD oder der Weboberfläche des Greifers heruntergeladen werden kann. Weiterführende Informationen zu den einzelnen E/A-Register, ([🔗 3, Seite 9](#)).

2.2.1 Installation der GSDML-Datei

HINWEIS

Die GSDML-Datei kann in Siemens STEP7 v11.0 (TIA) und neuer installiert werden.

Die GSD-Datei ist ein komprimiertes ZIP-Archiv, welches folgende Dateien enthält:

- GSDML-V2.31-Weiss Robotics-WSG-20140401.gsdml (Gerätebeschreibungdatei)
- GSDML-02A2-0001-WSG.bmp (Bilddatei)

Die Installation der GSDML Datei in Siemens STEP7 11.0 folgendermaßen durchführen:

- 1 Zip-Archiv auf die Festplatte entpacken.
- 2 In Siemens TIA die Projektansicht öffnen.
- 3 Unter *Extras -> Gerätebeschreibungdatei (GSD) installieren* wählen.
- 4 Zu dem Verzeichnis, in welches die Dateien entpackt wurden, wechseln und die GSDML-Datei wählen.

⇒ Im Gerätekatalog unter *Weitere Feldgeräte* -> *PROFINET IO* -> *I/O* -> *Weiss Robotics GmbH & Co. KG* wird der Greifer angezeigt.

2.2.2 Konfiguration

Um die Schnittstelle PROFINET verwenden zu können, muss die Schnittstelle zuerst über die Weboberfläche ausgewählt werden. Weitere Einstellungen, wie das Ändern der IP-Adresse oder der PROFINET-Gerätebezeichnung, können direkt über die Weboberfläche, über die verschiedenen Konfigurationsmöglichkeiten von PROFINET oder mit Entwicklungstools wie z. B. Siemens STEP7, vorgenommen werden.

HINWEIS

Beim Ändern der IP-Adresse mit Hilfe eines Entwicklungstools kann es passieren, dass die Webschnittstelle nicht mehr ansprechbar ist, falls die PROFINET-Verbindung verloren geht.

Es wird empfohlen das Ändern der IP-Adresse ausschließlich über die Weboberfläche durchzuführen.

Weitere Informationen befinden sich in der Montage- und Betriebsanleitung des Greifers.

2.3 Modbus/TCP

Auch wenn sich das Modbus/TCP Protokoll von PROFIBUS und PROFINET unterscheidet, benutzt es die gleichen E/A-Register. Der Hauptunterschied ist, dass bei Modbus/TCP ein Register aus zwei Byte besteht, wobei das niedrigwertige Byte vorne steht (Little Endian). Dies bedeutet, dass die Register wie folgt auf die Modbus Register abgebildet werden:

Modbus Register	0															
Byte Nummer	0								1							
Bit Nummer	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8

Dieses Beispiel startet den MOVE Befehl und setzt das Benutzer-Flag IF6 auf "1":

Modbus Register 0	0000 0001 0010 0000 ₂ = 288 ₁₀															
Register Name	CMDFLAGS								IF							
Byte	0000 0001 ₂ = 1 ₁₀								0010 0000 ₂ = 32 ₁₀							
Bits	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1

Alle binären Ausgabeflags (Befehlsflags und Benutzerflags) können auch als Modbus "Coils" angesprochen werden. Ebenso können alle binären Eingabeflags (Greifzustand, Benutzerflag und Statuscode) als Modbus "Discrete Inputs" adressiert werden. Die Reihenfolge folgt dabei der natürlichen Bit-Reihenfolge (beginnend bei Byte 0 und Bit 0).

2.3.1 Konfiguration

Um die Modbus/TCP-Schnittstelle verwenden zu können, muss diese zuerst über die Weboberfläche (*Settings -> Command Interface*) eingeschaltet werden. Die IP-Adresse entspricht der des Geräts. Der Port ist fest auf den Standard Port "502" für Modbus/TCP eingestellt und kann nicht geändert werden. Weitere Informationen befinden sich in der Montage- und Betriebsanleitung des Greifers.

3 Schnittstellenbeschreibung

Die Feldbus-Schnittstelle des Greifers ist als 8-Byte Ausgabe- und 12-Byte Eingabe-Registersätze implementiert.

3.1 Ausgaberegister (SPS zu WSG)

Die Ausgaberegister werden vom PROFIBUS-Master, PROFINET-Controller (z. B. SPS) oder Modbus/TCP-Master zum Greifer übertragen. Die Ausgaberegister bestehen aus Befehlsflags, Benutzerflags und drei Parametern und werden zur Steuerung des Greifers verwendet. Aufgrund der registersatzorientierten Natur von PROFIBUS, PROFINET und Modbus steht über diesen Schnittstellen nur ein Teil des Funktionsspektrums des Greifers zur Verfügung. Die Anordnung der Register siehe folgende Tabelle.

Für die Modbus/TCP-Schnittstelle sind die Befehlsflags und die Benutzerflags auch als "Coils" verfügbar. Eine Beschreibung der Adressierung der Modbus-Register und "Coils" befindet sich unter [\(☞ 2.3, Seite 7\)](#).

Anordnung Ausgaberegister

Byte Nummer	Modbus Holding Register	Register Name	Beschreibung																											
0	0	CMDFLAGS	<p>Befehlsflags</p> <p>Ein Befehl wird erteilt, indem das entsprechende Bit von "0" auf "1" gesetzt wird (steigende Flanke). Weiterführende Informationen zu den Befehlen, (☞ 4, Seite 17).</p> <table border="1"> <thead> <tr> <th>Bit Index</th> <th>Name</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>Bit 0:</td> <td>MOVE</td> <td>Initiiert eine Vorpositionierung</td> </tr> <tr> <td>Bit 1:</td> <td>GRIP</td> <td>Greift ein Werkstück</td> </tr> <tr> <td>Bit 2:</td> <td>RELEASE</td> <td>Lässt ein Werkstück los</td> </tr> <tr> <td>Bit 3:</td> <td>HOMING</td> <td>Referenzieren der Greiferfinger</td> </tr> <tr> <td>Bit 4:</td> <td>STOP/ACK</td> <td>Stopp ohne Abschalten des Motors, Quittieren eines FASTSTOP</td> </tr> <tr> <td>Bit 5:</td> <td>FASTSTOP</td> <td>Stopp mit Abschaltung des Motors</td> </tr> <tr> <td>Bit 6:</td> <td>JOG+</td> <td>Jog-Modus in positiver Richtung</td> </tr> <tr> <td>Bit 7:</td> <td>JOG-</td> <td>Jog-Modus in negativer Richtung</td> </tr> </tbody> </table>	Bit Index	Name	Beschreibung	Bit 0:	MOVE	Initiiert eine Vorpositionierung	Bit 1:	GRIP	Greift ein Werkstück	Bit 2:	RELEASE	Lässt ein Werkstück los	Bit 3:	HOMING	Referenzieren der Greiferfinger	Bit 4:	STOP/ACK	Stopp ohne Abschalten des Motors, Quittieren eines FASTSTOP	Bit 5:	FASTSTOP	Stopp mit Abschaltung des Motors	Bit 6:	JOG+	Jog-Modus in positiver Richtung	Bit 7:	JOG-	Jog-Modus in negativer Richtung
Bit Index	Name	Beschreibung																												
Bit 0:	MOVE	Initiiert eine Vorpositionierung																												
Bit 1:	GRIP	Greift ein Werkstück																												
Bit 2:	RELEASE	Lässt ein Werkstück los																												
Bit 3:	HOMING	Referenzieren der Greiferfinger																												
Bit 4:	STOP/ACK	Stopp ohne Abschalten des Motors, Quittieren eines FASTSTOP																												
Bit 5:	FASTSTOP	Stopp mit Abschaltung des Motors																												
Bit 6:	JOG+	Jog-Modus in positiver Richtung																												
Bit 7:	JOG-	Jog-Modus in negativer Richtung																												

Byte Nummer	Modbus Holding Register	Register Name	Beschreibung																											
			ACHTUNG! Wenn das FASTSTOP oder das STOP/ACK Bit auf "1" gesetzt ist, werden alle Bewegungsbefehle ignoriert.																											
1		IF	Benutzer-Flags (Eingabe) Frei verfügbare Flags, die in Verbindung mit dem Skript-Interpreter verwendet werden können.																											
			<table border="1"> <thead> <tr> <th>Bit Index</th> <th>Name</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>Bit 0:</td> <td>IF1</td> <td>Eingabe Benutzer-Flag 1</td> </tr> <tr> <td>Bit 1:</td> <td>IF2</td> <td>Eingabe Benutzer-Flag 2</td> </tr> <tr> <td>Bit 2:</td> <td>IF3</td> <td>Eingabe Benutzer-Flag 3</td> </tr> <tr> <td>Bit 3:</td> <td>IF4</td> <td>Eingabe Benutzer-Flag 4</td> </tr> <tr> <td>Bit 4:</td> <td>IF5</td> <td>Eingabe Benutzer-Flag 5</td> </tr> <tr> <td>Bit 5:</td> <td>IF6</td> <td>Eingabe Benutzer-Flag 6</td> </tr> <tr> <td>Bit 6:</td> <td>IF7</td> <td>Eingabe Benutzer-Flag 7</td> </tr> <tr> <td>Bit 7:</td> <td>IF8</td> <td>Eingabe Benutzer-Flag 8</td> </tr> </tbody> </table>	Bit Index	Name	Beschreibung	Bit 0:	IF1	Eingabe Benutzer-Flag 1	Bit 1:	IF2	Eingabe Benutzer-Flag 2	Bit 2:	IF3	Eingabe Benutzer-Flag 3	Bit 3:	IF4	Eingabe Benutzer-Flag 4	Bit 4:	IF5	Eingabe Benutzer-Flag 5	Bit 5:	IF6	Eingabe Benutzer-Flag 6	Bit 6:	IF7	Eingabe Benutzer-Flag 7	Bit 7:	IF8	Eingabe Benutzer-Flag 8
Bit Index	Name	Beschreibung																												
Bit 0:	IF1	Eingabe Benutzer-Flag 1																												
Bit 1:	IF2	Eingabe Benutzer-Flag 2																												
Bit 2:	IF3	Eingabe Benutzer-Flag 3																												
Bit 3:	IF4	Eingabe Benutzer-Flag 4																												
Bit 4:	IF5	Eingabe Benutzer-Flag 5																												
Bit 5:	IF6	Eingabe Benutzer-Flag 6																												
Bit 6:	IF7	Eingabe Benutzer-Flag 7																												
Bit 7:	IF8	Eingabe Benutzer-Flag 8																												
2.3	1	WIDTH	Befehlsparameter WIDTH (Öffnungsweite) Anzufahrende Greiferfinger-Öffnungsweite in 1/100 Millimetern, d. h. ein Wert von "1220" bedeutet 12,20 mm. Kodiert als INT (signed).																											
4..5	2	SPEED	Befehlsparameter SPEED (Geschwindigkeit) Zu verwendende Relativgeschwindigkeit beider Greiferfinger in 1/100 Millimeter pro Sekunde (d. h. ein Wert von "3005" bedeutet 30.05 mm/s). Kodiert als WORD (unsigned). ACHTUNG! Das Setzen dieses Parameters auf einen Wert außerhalb der Grenzwerte des Systems und das Auslösen einer Bewegungsfunktion führt zu einem FASTSTOP.																											

Byte Nummer	Modbus Holding Register	Register Name	Beschreibung
6..7	3	FORCELIMIT	<p>Befehlsparameter FORCELIMIT (Kraftbegrenzung)</p> <p>Einzustellender Grenzwert für die Greifkraft in 1/100 Newton, d. h. ein Wert von "1050" bedeutet 10.50 N. Ausnahme: Beim WSG 70 ist die Einheit aufgrund der höheren Greifkraft 1/10 Newton, d. h. ein Wert von "3050" bedeutet 305.0 N.</p> <p>Die Greifkraft ist das Doppelte der Nennkraft, die auf das zu greifende Teil ausgeübt wird.</p> <p>Kodiert als INT (signed), nur positive Werte erlaubt.</p> <p>ACHTUNG! Das Setzen dieses Parameters auf einen Wert außerhalb der Grenzwerte des Systems und das Auslösen einer Bewegungsfunktion führt zu einem FASTSTOP.</p>

Um einen Befehl auszuführen, müssen die Befehlsparameter gesetzt und das entsprechende Befehlsflag von "0" auf "1" (steigende Flanke) gesetzt werden. Jog Modus-Flags werden ihrem Pegel entsprechend ausgewertet.

Eine detaillierte Beschreibung der einzelnen Befehle befindet sich unter ([👉 3.2, Seite 12](#)).

HINWEIS

Wenn mehr als ein Befehlsflag gleichzeitig geändert wurde, wird nur der Befehl mit der niedrigsten Bit-Nummer ausgeführt, d. h. das Setzen der Flags MOVE und GRIP von "0" auf "1" löst nur den Befehl MOVE aus.



ACHTUNG

Das Ändern von Parametern während der Greiferfingerbewegung löst einen FASTSTOP aus, z. B. das MOVING System-Flag ist "1".

3.2 Eingaberegister (WSG zu SPS)

Der Eingaberegistersatz (siehe folgende Tabelle) wird bei jedem Zyklus vom Greifer zum PROFIBUS-Master, PROFINET-Controller oder Modbus/TCP-Master übertragen. Der Eingaberegistersatz enthält die aktuellen Parameter des Greifers, den Betriebs- und Greifstatus, die Benutzer-Flags, sowie den Statuscode als Rückgabewert des zuletzt ausgeführten Befehls. Die Anordnung der Register siehe folgende Tabelle.

Für die Modbus/TCP-Schnittstelle sind der Greifzustand, die Benutzer-Flags und der Statuscode auch als "Discrete Inputs" verfügbar. Eine Beschreibung der Adressierung der Modbus-Register und "Discrete Inputs" befindet sich unter [\(☞ 2.3, Seite 7\)](#).

Anordnung Eingaberegister

Byte Nummer	Modbus Input Register	Register Name	Beschreibung																											
0	0	GSTATE	<p>Greifstatus</p> <p>Diese Flags zeigen den aktuellen Status des Greifvorganges und dienen der Steuerung und Überwachung des Greifvorganges:</p> <table border="1"> <thead> <tr> <th>Bit Index</th> <th>Name</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>Bit 0:</td> <td>IDLE</td> <td>Warte auf neuen Befehl</td> </tr> <tr> <td>Bit 1:</td> <td>GRIPPING</td> <td>Greiferfinger bewegen sich auf Teil zu</td> </tr> <tr> <td>Bit 2:</td> <td>NO_PART</td> <td>Kein Teil gefunden</td> </tr> <tr> <td>Bit 3:</td> <td>PART_LOST</td> <td>Teil wurde gegriffen, ging aber wieder verloren</td> </tr> <tr> <td>Bit 4:</td> <td>HOLDING</td> <td>Teil wird gehalten</td> </tr> <tr> <td>Bit 5:</td> <td>RELEASING</td> <td>Greiferfinger entfernen sich vom Teil</td> </tr> <tr> <td>Bit 6:</td> <td>POSITIONING</td> <td>Greiferfinger bewegen sich aufgrund des Vorpositionierungsbefehls MOVE</td> </tr> <tr> <td>Bit 7:</td> <td>ERROR</td> <td>Ein Fehler ist aufgetreten</td> </tr> </tbody> </table>	Bit Index	Name	Beschreibung	Bit 0:	IDLE	Warte auf neuen Befehl	Bit 1:	GRIPPING	Greiferfinger bewegen sich auf Teil zu	Bit 2:	NO_PART	Kein Teil gefunden	Bit 3:	PART_LOST	Teil wurde gegriffen, ging aber wieder verloren	Bit 4:	HOLDING	Teil wird gehalten	Bit 5:	RELEASING	Greiferfinger entfernen sich vom Teil	Bit 6:	POSITIONING	Greiferfinger bewegen sich aufgrund des Vorpositionierungsbefehls MOVE	Bit 7:	ERROR	Ein Fehler ist aufgetreten
Bit Index	Name	Beschreibung																												
Bit 0:	IDLE	Warte auf neuen Befehl																												
Bit 1:	GRIPPING	Greiferfinger bewegen sich auf Teil zu																												
Bit 2:	NO_PART	Kein Teil gefunden																												
Bit 3:	PART_LOST	Teil wurde gegriffen, ging aber wieder verloren																												
Bit 4:	HOLDING	Teil wird gehalten																												
Bit 5:	RELEASING	Greiferfinger entfernen sich vom Teil																												
Bit 6:	POSITIONING	Greiferfinger bewegen sich aufgrund des Vorpositionierungsbefehls MOVE																												
Bit 7:	ERROR	Ein Fehler ist aufgetreten																												

Byte Nummer	Modbus Input Register	Register Name	Beschreibung		
1	0	OF	Benutzerflags (Ausgabe) Frei verwendbare Flags zur Kommunikation zwischen SPS-Code und einem laufenden WSG-Skript		
			Bit Index	Name	Beschreibung
			Bit 0:	OF1	Ausgabe Benutzer Flag 1
			Bit 1:	OF2	Ausgabe Benutzer Flag 2
			Bit 2:	OF3	Ausgabe Benutzer Flag 3
			Bit 3:	OF4	Ausgabe Benutzer Flag 4
			Bit 4:	OF5	Ausgabe Benutzer Flag 5
			Bit 5:	OF6	Ausgabe Benutzer Flag 6
			Bit 6:	OF7	Ausgabe Benutzer Flag 7
			Bit 7:	OF8	Ausgabe Benutzer Flag 8
2..5	1..2	SYSSTATE	System Status Aktueller Systemzustand des Greifers als Bitvektor kodiert, (👉 5.2, Seite 29). Dieses Register wird bei jedem Buszyklus, ungeachtet laufender Befehle, aktualisiert. ACHTUNG! Die Systemzustandsflags nicht zur Steuerung des Greifvorgangs verwenden, stattdessen die Greifstatus-Flags verwenden.		
			Bit Index	Name	
			Bit 0	REFERENCED	
			Bit 1	MOVING	
			Bit 2	BLOCKED_MINUS	
			Bit 3	BLOCKED_PLUS	
			Bit 4	SOFT_LIMIT_MINUS	
			Bit 5	SOFT_LIMIT_PLUS	
			Bit 6	AXIS_STOPPED	
			Bit 7	TARGET_POS_REACHED	
			Bit 8	OVERDRIVE_MODE ¹	
			Bit 9	FORCECNTL_MODE	
			Bit 10	reserviert	
			Bit 11	reserviert	

Byte Nummer	Modbus Input Register	Register Name	Beschreibung	
			Bit 12	FAST_STOP
			Bit 13	TEMP_WARNING
			Bit 14	TEMP_FAULT
			Bit 15	POWER_FAULT
			Bit 16	CURR_FAULT
			Bit 17	FINGER_FAULT
			Bit 18	CMD_FAILURE
			Bit 19	SCRIPT_RUNNING
			Bit 20	SCRIPT_FAILURE
			Bit 21	reserviert
			Bit 22	reserviert
			Bit 23	reserviert
			Bit 24	reserviert
			Bit 25	reserviert
			Bit 26	reserviert
			Bit 27	reserviert
			Bit 28	reserviert
			Bit 29	reserviert
			Bit 30	reserviert
			Bit 31	reserviert
6..7	3	WIDTH	<p>Aktuelle Öffnungsweite</p> <p>Aktuelle Öffnungsweite der Greiferfinger in 1/100 Millimetern, d. h. ein Wert von "1220" bedeutet 12.2 mm.</p> <p>Kodiert als INT.</p> <p>Ungeachtet laufender Befehle wird dieses Register bei jedem Buszyklus aktualisiert.</p>	

Byte Nummer	Modbus Input Register	Register Name	Beschreibung
8..9	4	GRIPPING FORCE	<p>Aktuelle Greifkraft</p> <p>Aktuelle Greifkraft in 1/100 Newton, d. h. ein Wert von "4050" bedeutet eine Greifkraft von 40.5 N.</p> <p>Ausnahme: Beim WSG 70 ist die Einheit aufgrund der höheren Greifkraft 1/10 Newton, d. h. ein Wert von "3050" bedeutet 305.0 N).</p> <p>Die Greifkraft ist das Doppelte der Nennkraft, die auf das zu greifende Werkstück ausgeübt wird.</p> <p>Kodiert als INT.</p> <p>Ungeachtet laufender Befehle wird dieses Register bei jedem Buszyklus aktualisiert. ACHTUNG! Wenn auf dem Greifer kein Messfinger installiert ist, wird diese Größe anhand des Motorstromes angenähert.</p>
10..11	5	STATUS CODE	<p>Ergebnis des letzten Befehls</p> <p>Dieses Feld behält seinen Zustand, bis ein neuer Befehl ausgeführt wird, (☞ 5.1, Seite 27).</p>
<p>¹⁾ Der Overdrive Modus wird nicht von allen Greifern unterstützt. Weitere Informationen befinden sich in der Montage- und Betriebsanleitung des Greifers.</p>			

3.3 Diagnosemitteilungen PROFIBUS

Der Greifer sendet Diagnosemitteilungen, welche die aktuellen Systemzustandflags enthalten, als das erste Double Word zum PROFIBUS-Master (SPS), wenn mindestens eines der folgenden fehlerbezogenen Flags des Systemzustand-Registers von "0" auf "1" wechselt:

- SF_SOFT_LIMIT_MINUS
- SF_SOFT_LIMIT_PLUS
- SF_FAST_STOP
- SF_TEMP_FAULT
- SF_POWER_FAULT
- SF_CURR_FAULT
- SF_FINGER_FAULT
- SF_CMD_FAILURE
- SF_SCRIPT_FAILURE

Eine detaillierte Beschreibung befindet sich unter ([☞ 5.2, Seite 29](#)).

Format der Diagnosemitteilungen

Byte Nummer	Beschreibung
0..3	Standard Diagnosedaten Diagnosedaten, wie sie in der Spezifikation des PROFIBUS definiert sind.
4..5	Slave Ident-Nr. Slave-Identifikationsnummer. Diese ist 0x5555 für den WSG.
6	Länge der Diagnosemitteilung Diagnosemitteilungen haben immer eine Länge 10 Bytes = 0x0A.
7..10	System Status Aktueller Systemzustand des Greifers als Bitvektor kodiert. Gleiche Kodierung wie bei Register SYSSTATE, (☞ 3.2, Seite 12).
10..15	reserviert Dieser Bereich ist für zukünftige Verwendungen reserviert.

4 Befehle

4.1 MOVE – Bewegen der Greiferfinger im Positionierungsmodus

Dieser Befehl kann verwendet werden, um die Greiferfinger vor dem Auslösen eines Greifvorganges auf eine bestimmte Öffnungsweite zu fahren. Wenn aufgrund von Prozessbedingungen die Greiferfinger vor dem Greifen größere Distanzen zurücklegen müssen, wird durch diesen Befehl das Greifen empfindlicher Werkstücke beschleunigt. MOVE kann nur ausgeführt werden, wenn der Greifer untätig ist, d.h. der Greifstatus ist IDLE.

Position des Befehlsflags

Bit 0

Verwendete Parameter

WIDTH, SPEED

Statuscode

Das Statuscode-Register wird zu Beginn der Bewegung auf E_CMD_PENDING und nach Ausführung des Befehls auf das entsprechende Ergebnis gesetzt.

Greifstatus

Der Greifstatus wechselt zu POSITIONING, wenn die Bewegung beginnt und wechselt zu IDLE, nachdem der Befehl erfolgreich abgearbeitet wurde. Im Fall eines Fehlers wird der Greifstatus auf ERROR gesetzt.

Systemstatus

Es treten verschiedene Übergänge auf. Sofern es keine sehr speziellen Anforderungen gibt, sollte der aktuelle Zustand des Greifvorganges, über das Greifstatus-Register ausgewertet werden.

4.2 GRIP – Greifen eines Teils

Greift ein Teil mit seiner Nennweite unter Berücksichtigung der vorgegebenen Geschwindigkeits- und Kraftbegrenzung. Wenn der Befehl ausgelöst wird, bewegt der Greifer die Greiferfinger zur Nennweite und versucht das erwartete Teil mit der vorher gesetzten Greifkraft einzuspannen. Wenn der Greifer die gewünschte Greifkraft innerhalb des festgelegten Klemmweges aufbauen kann, wurde das Teil gegriffen.

Wenn die Greiferfinger den Klemmweg durchfahren, ohne die Greifkraft aufzubauen, wurde kein Teil gefunden und der Greifstatus wird entsprechend aktualisiert. Der Klemmweg kann über die Weboberfläche des Greifers gesetzt werden. Der Greifstatus wird mit dem entsprechenden Ergebnis aktualisiert (entweder HOLDING oder NO_PART).

Wenn kein Teil gefunden wurde, gibt der Befehl den Statuscode E_CMD_FAILED als Ergebnis zurück. Nach erfolgreichem Greifen eines Teils ist die integrierte Greifteilüberwachung aktiv, welche die Greifkraft kontinuierlich überwacht. Wenn ein Teil aus dem Greifer entfernt wird, bevor der Release Befehl aufgerufen wurde, erkennt der Greifer dies und setzt den Greifstatus auf PART_LOST.

HINWEIS

Um den Aufprallimpuls von der Masse der Greiferfinger und der internen Mechanik zu begrenzen, kann die Greifgeschwindigkeit bei empfindlichen Werkstücken reduziert werden.

Der Greifstatus spiegelt den aktuellen Zustand des Greifvorganges wider. Dieser sollte nach jedem Befehl ausgewertet werden, um zu überprüfen, ob der Greifvorgang wie erwartet ausgeführt wurde.

Position des Befehlsflags

Bit 1

Verwendete Parameter

WIDTH, SPEED, FORCELIMIT

Statuscode

Das Statuscode-Register wird zum Beginn der Bewegung auf E_CMD_PENDING und nach Ausführung des Befehls auf das entsprechende Ergebnis gesetzt. Wenn kein Teil gefunden wurde, wird der Statuscode auf E_CMD_FAILED gesetzt.

Greifstatus

Während der Bewegung steht das Greifstatus-Register auf GRIPPING. Wenn ein Teil gefunden wurde, wechselt es zu HOLDING. Wenn kein Teil gefunden wurde, wird es auf NO_PART gesetzt. Wenn das Teil entfernt oder verloren wird, nachdem es erfolgreich gegriffen wurde, wechselt der Greifstatus zu PART_LOST. Im Fall eines Fehlers wird das Register auf ERROR gesetzt.

Systemstatus

Es treten verschiedene Übergänge auf. Sofern es keine sehr speziellen Anforderungen gibt, sollte der aktuelle Zustand des Greifvorganges über das Greifstatus-Register ausgewertet werden.

4.3 RELEASE – Loslassen eines Teils

Lässt ein Teil durch Öffnen der Greiferfinger mit einer vorgegebenen Geschwindigkeit und Öffnungsweite los. Der RELEASE-Befehl quetscht das Teil nicht ein. Dies ist gewährleistet durch sukzessives Erhöhen der internen Kraftgrenze beim Entfernen der Greiferfinger vom Teil. Die Greifteilüberwachung wird vor dem Loslassen abgeschaltet. Zum Loslassen wird die Nennkraft des Greifers verwendet.

Position des Befehlsflags

Bit 3

Verwendete Parameter

WIDTH, SPEED

Statuscode

Das Statuscode-Register wird zu Beginn der Bewegung auf E_CMD_PENDING und nach Ausführung des Befehls auf das entsprechende Ergebnis gesetzt.

Greifstatus

Während der Bewegung steht das Greifstatus-Register auf RELEASING. Bei Erreichen der Endposition wechselt das Register auf IDLE. Im Fehlerfall wird es auf ERROR gesetzt.

Systemstatus

Es treten verschiedene Übergänge auf. Sofern es keine speziellen Anforderungen gibt, sollte der aktuelle Zustand des Greifvorganges über das Greifstatus-Register ausgewertet werden.

4.4 HOMING – Referenzieren der Greiferfinger

Dieser Befehl referenziert den Greifer durch Ausführen einer Homing-Sequenz. Während des Homings fahren die Greiferfinger bis zum mechanischen Anschlag. Die Homing-Sequenz muss vorher auf der WSG- Weboberfläche unter *Settings -> Motion Configuration* konfiguriert werden. Die Richtung der Referenzierung (nach innen oder außen) sowie das automatische Homing beim Starten des Greifers ist einstellbar.

HINWEIS

Das Homing ist vor dem Ausführen aller Bewegungsbefehle erforderlich. Das beste Positionierergebnis wird erreicht, wenn das Homing in die Richtung erfolgt, in der die höhere Positioniergenauigkeit erforderlich ist.

HINWEIS

Während des Homings sind die Softlimits abgeschaltet. Kollidieren die Greiferfinger unterwegs mit einem Hindernis, kann dies einen falschen Referenzpunkt und somit eine falsche Positionierung zur Folge haben.

Position des Befehlsflags

Bit 3

Verwendete Parameter

keine

Statuscode

Das Statuscode-Register wird sofort auf E_CMD_PENDING und nach Abschluss des Befehls auf das entsprechende Ergebnis gesetzt.

Greifstatus

Während der Homing-Sequenz steht das Greifstatus-Register auf POSITIONING.

Systemstatus

Während der Bewegung ist das MOVING Flag auf "1" gesetzt. Wenn der Greifer referenziert wurde, wird das REFERENCED Flag auf "1" gesetzt.

4.5 STOP/ACK – Stoppen der Bewegung oder Quittieren eines FASTSTOP

Stoppt jede anstehende Bewegung ohne den Antrieb auszuschalten. Wenn gestoppt wird, während ein Werkstück gehalten wird, d. h. der Greifstatus steht auf HOLDING, wird die Greifteilüberwachung abgeschaltet und die eingestellte Greifkraft wird nicht länger aufgebracht.

Quittieren eines FASTSTOP Zustands:

Wenn der Greifer im FASTSTOP Modus ist, muss ein Übergang dieses Flags von "0" auf "1" erfolgen, um den Zustand zu quittieren und in den normalen Betriebsmodus zurückzukehren. Vor dem Quittieren muss das FASTSTOP-Flag zurückgesetzt werden.

Position des Befehlsflags

Bit 4

Verwendete Parameter

keine

Statuscode

Das Statuscode-Register wird auf E_SUCCESS gesetzt.

Greifstatus

Der Greifstatus wird auf IDLE gesetzt.

Systemstatus

Das AXIS_STOPPED Flag ist auf "1" gesetzt. Durch Quittieren eines FASTSTOP wird das FASTSTOP Flag gelöscht.

4.6 FASTSTOP – Auslösen eines Fast Stop

Diese Funktion ist mit einem “Notstopp“ vergleichbar. Sie stoppt unverzüglich jede Bewegung auf dem kürzesten Weg, schaltet den Antrieb ab und unterbindet die Ausführung weiterer Bewegungsbefehle. Der FASTSTOP Zustand muss quittiert werden, ([↩ 4.5, Seite 22](#)) Alle Bewegungsbefehle sind während eines FASTSTOP verboten und rufen einen E_ACCESS_DENIED Fehler hervor. Der FASTSTOP Zustand wird in den System-Flags angezeigt und in der System-Logdatei aufgezeichnet.

HINWEIS

Dieser Befehl sollte verwendet werden, um auf bestimmte Fehlerbedingungen zu reagieren. Um lediglich die aktuelle Bewegung zu stoppen, den STOP Befehl verwenden.

HINWEIS

Neben dem STOP/ACK Flag kann ein FASTSTOP auch interaktiv über die Weboberfläche zurückgesetzt werden. Dies reaktiviert den Antrieb, jedoch ist es erforderlich, das FASTSTOP Flag der PROFIBUS-Schnittstelle zurückzusetzen, um bewegungsbezogene Befehle wieder freizuschalten.

Position des Befehlsflags

Bit 5

Verwendete Parameter

keine

Statuscode

Das Statuscode-Register wird auf E_SUCCESS gesetzt.

Greifstatus

Der Greifstatus wird auf IDLE gesetzt.

Systemstatus

Das FASTSTOP Flag wird auf "1" gesetzt.

4.7 JOG+ und JOG- – Jog Modus

Um einen Prozess einzurichten, kann es erforderlich sein, die Greiferfinger manuell zu bewegen. Dies kann mit Hilfe der Jog Modus Flags realisiert werden. Die Flags werden ihrem Pegel entsprechend ausgewertet und ermöglichen den Antrieb der Greiferfinger mit konstanter Geschwindigkeit unter Verwendung zweier Taster an der SPS. Die Flags werden wie in der folgenden Tabelle interpretiert:

Interpretation der Flags

JOG+	JOG-	Movement direction
0	0	Jog Mode is disabled*
1	0	positive with SPEED
0	1	negative with SPEED
1	1	Stop
*Wenn beide Jog Flags auf "0" wechseln, wird der Jog Modus verlassen und der Antrieb gestoppt.		

Die Kraftgrenze (nur stromgesteuert) sowie die Geschwindigkeit können als Parameter übergeben werden. Es könnte ein Stellrad verwendet werden, um sie zu steuern. Es ist zu beachten, dass eine Bewegung mit hoher Geschwindigkeit durch das Einstellen einer zu niedrigen Kraftgrenze behindert werden kann.

HINWEIS

Im Gegensatz zu anderen Bewegungsbefehlen kann der SPEED Parameter auf "0" gesetzt werden, was dazu führt, dass der Wert intern auf die minimal zulässige Geschwindigkeit gesetzt wird.

HINWEIS

Der Jog Modus ist lediglich zum Einrichten gedacht und sollte nicht für den normalen Betrieb des Greifers benutzt werden!

Position des Befehlsflags

Bit 6 and 7

Verwendete Parameter

SPEED, FORCELIMIT

Statuscode

Das Statuscode-Register wird auf E_CMD_PENDING und nach Abschluss des Befehls auf das entsprechende Ergebnis gesetzt.

Greifstatus

Während der Bewegung der Greiferfinger wird das Register auf RELEASING gesetzt. Wenn die Endposition erreicht wurde oder im Fehlerfall, wird der Greifstatus auf IDLE gesetzt.

Systemstatus

Es treten verschiedene Übergänge auf. Sofern es keine sehr speziellen Anforderungen gibt, sollte der aktuelle Zustand des Greifvorganges über das Greifstatus-Register ausgewertet werden.

5 Feldbusmonitor

Der Greifer hat einen integrierten Feldbusmonitor. Über die Web-schnittstelle des Greifers kann unter *Diagnosis -> Fieldbus Monitor* auf den Feldbusmonitor zugegriffen werden.

Der Feldbusmonitor zeigt den aktuellen Inhalt der Eingabe- und Ausgaberegister und gibt einige Grundinformationen zum Zustand des Feldbus.

Während der Integration des Greifers in eine Anlage kann der Feldbusmonitor genutzt werden, um den Zustand der Feldbus-schnittstelle abzufragen.

WSG 50 Control Panel
Location: n/a, Contact: n/a

Settings Diagnostics Scripting Motion Help

Profibus

Bus State

Station Address	7
Bitrate	1.5 MB/s
Interface state	Online

I/O Register View

Output Register (Profibus Master to WSG)				Input Register (WSG to Profibus Master)			
Byte Index	Data	Description	Value	Byte Index	Data	Description	Value
0	20h	Command Flags	<input type="checkbox"/> MOVE <input type="checkbox"/> GRASP <input type="checkbox"/> RELEASE <input type="checkbox"/> HOMING <input type="checkbox"/> STORACK <input checked="" type="checkbox"/> FASTSTOP <input type="checkbox"/> JOG+ <input type="checkbox"/> JOG-	0	01h	Grasping State	<input checked="" type="checkbox"/> IDLE <input type="checkbox"/> GRASPING <input type="checkbox"/> NO_PART <input type="checkbox"/> PART_LOST <input type="checkbox"/> HOLDING <input type="checkbox"/> RELEASING <input type="checkbox"/> POSITIONING <input type="checkbox"/> --
1	C0h	User Flags	<input type="checkbox"/> F1 <input type="checkbox"/> F2 <input type="checkbox"/> F3 <input type="checkbox"/> F4 <input type="checkbox"/> F5 <input type="checkbox"/> F6 <input checked="" type="checkbox"/> F7 <input checked="" type="checkbox"/> F8	1	00h	User Flags	<input type="checkbox"/> OF1 <input type="checkbox"/> OF2 <input type="checkbox"/> OF3 <input type="checkbox"/> OF4 <input type="checkbox"/> OF5 <input type="checkbox"/> OF6 <input type="checkbox"/> OF7 <input type="checkbox"/> OF8
2	00h	Width	0.00 mm	2	00h	System State	00000000h
3	00h	Speed	0.00 mm/s	3	00h	System State	00000000h
4	00h	Speed	0.00 mm/s	4	00h	System State	00000000h
5	00h	Force Limit	0.00 N	5	00h	Current Opening Width	0.00 mm
6	00h	Force Limit	0.00 N	6	00h	Current Opening Width	0.00 mm
7	00h	Force Limit	0.00 N	7	00h	Grasping Force	0.00 N
				8	00h	Grasping Force	0.00 N
				9	00h	Error Code	E_SUCCESS
				10	00h	Error Code	E_SUCCESS
				11	00h	Error Code	E_SUCCESS

Grepper

State: idle
Position: [n/a]
Speed: 0.0 mm/s
Force: 0.0 N

Stop Ack

- Referenced
- Moving
- Blocked Minus
- Blocked Plus
- Soft Limit Minus
- Soft Limit Plus
- Axis stopped
- Target Pos reached
- Overtorque Mode
- Force Control Mode
- Fast Stop
- Temperature Warning
- Temperature Fault
- Power Fault
- Current Fault
- Finger Fault
- Command Failure
- Script is running
- Script Failure

Feldbusmonitor, Beispiel

5.1 Anhang A: Statuscodes

Status Code	Symbol Name	Beschreibung
0	E_SUCCESS	Kein Fehler aufgetreten, Vorgang erfolgreich
1	E_NOT_AVAILABLE	Funktion oder Daten nicht verfügbar
2	E_NO_SENSOR	Kein Messumformer angeschlossen
3	E_NOT_INITIALIZED	Gerät nicht initialisiert
4	E_ALREADY_RUNNING	Datenerfassung wird bereits ausgeführt
5	E_FEATURE_NOT_SUPPORTED	Funktion ist momentan nicht verfügbar
6	E_INCONSISTENT_DATA	Einer oder mehrere Parameter sind inkonsistent
7	E_TIMEOUT	Zeitüberschreitungsfehler
8	E_READ_ERROR	Fehler beim Lesen von Daten
9	E_WRITE_ERROR	Fehler beim Schreiben von Daten
10	E_INSUFFICIENT_RESOURCES	Nicht genügend Speicher vorhanden
11	E_CHECKSUM_ERROR	Prüfsummenfehler
12	E_NO_PARAM_EXPECTED	Parameter übergeben, obwohl keiner erwartet
13	E_NOT_ENOUGH_PARAMS	Zu wenige Parameter für den Befehl übergeben
14	E_CMD_UNKNOWN	Unbekannter Befehl
15	E_CMD_FORMAT_ERROR	Befehlsformat Fehler
16	E_ACCESS_DENIED	Zugriff verweigert
17	E_ALREADY_OPEN	Schnittstelle ist bereits geöffnet
18	E_CMD_FAILED	Fehler während der Ausführung eines Befehls
19	E_CMD_ABORTED	Befehlsausführung vom Benutzer abgebrochen
20	E_INVALID_HANDLE	Ungültige Aktion
21	E_NOT_FOUND	Gerät oder Datei nicht gefunden
22	E_NOT_OPEN	Gerät oder Datei nicht geöffnet
23	E_IO_ERROR	Eingabe oder Ausgabe Fehler
24	E_INVALID_PARAMETER	Falscher Parameter
25	E_INDEX_OUT_OF_BOUNDS	Index außerhalb des zulässigen Bereichs
26	E_CMD_PENDING	Kein Fehler, aber der Befehl wurde noch nicht vollständig ausgeführt. Eine Rückmeldung mit Statuscode folgt nach Ausführung des Befehls.
27	E_OVERRUN	Datenüberlauf

Status Code	Symbol Name	Beschreibung
28	E_RANGE_ERROR	Bereichsfehler
29	E_AXIS_BLOCKED	Achse blockiert
30	E_FILE_EXISTS	Datei existiert bereits

5.2 Anhang B: Systemstatus-Flags

Die Systemstatus-Flags sind in einem 32-Bit breiten Integer angeordnet, der über das Eingaberegister des PROFIBUS bereitgestellt wird.

Die spezielle Bedeutung der einzelnen Bits ist in folgender Tabelle beschrieben:

Bit Nr.	Flag Name	Beschreibung
D31..21	reserviert	Diese Bits werden aktuell nicht verwendet.
D20	SF_SCRIPT_FAILURE	<p>Skript Fehler</p> <p>Das Flag wird gesetzt, wenn während der Ausführung eines Skripts ein Fehler aufgetreten ist und das Skript abgebrochen wurde.</p> <p>Das Flag wird zurückgesetzt, wenn ein Skript gestartet wurde.</p>
D19	SF_SCRIPT_RUNNING	<p>Skript wird ausgeführt</p> <p>Das Flag wird gesetzt, wenn ein Skript ausgeführt wird.</p> <p>Das Flag wird zurückgesetzt, wenn ein Skriptfehler auftrat, das Skript normal oder vom Benutzer manuell beendet wurde.</p>
D18	SF_CMD_FAILURE	<p>Befehl gescheitert</p> <p>Das Flag wird gesetzt, wenn der letzte Befehl einen Fehler zurückgegeben hat.</p>
D17	SF_FINGER_FAULT	<p>Finger Störung</p> <p>Das Flag wird gesetzt, wenn der Status mindestens eines Greiferfingers von OPERATING auf NOT CONNECTED gesetzt wurde.</p> <p>Für eine genauere Fehlerbeschreibung die Finger Flags prüfen.</p>
D16	SF_CURR_FAULT	<p>Zu hoher Motorstrom</p> <p>Das Flag wird gesetzt, wenn der Motor die maximal zulässige Wärmeverlustleistung erreicht hat. Gleichzeitig wird ein FAST STOP gesetzt.</p> <p>Das Flag wird automatisch zurückgesetzt, wenn sich der Motor abgekühlt hat. Der FAST STOP muss quittiert werden.</p>

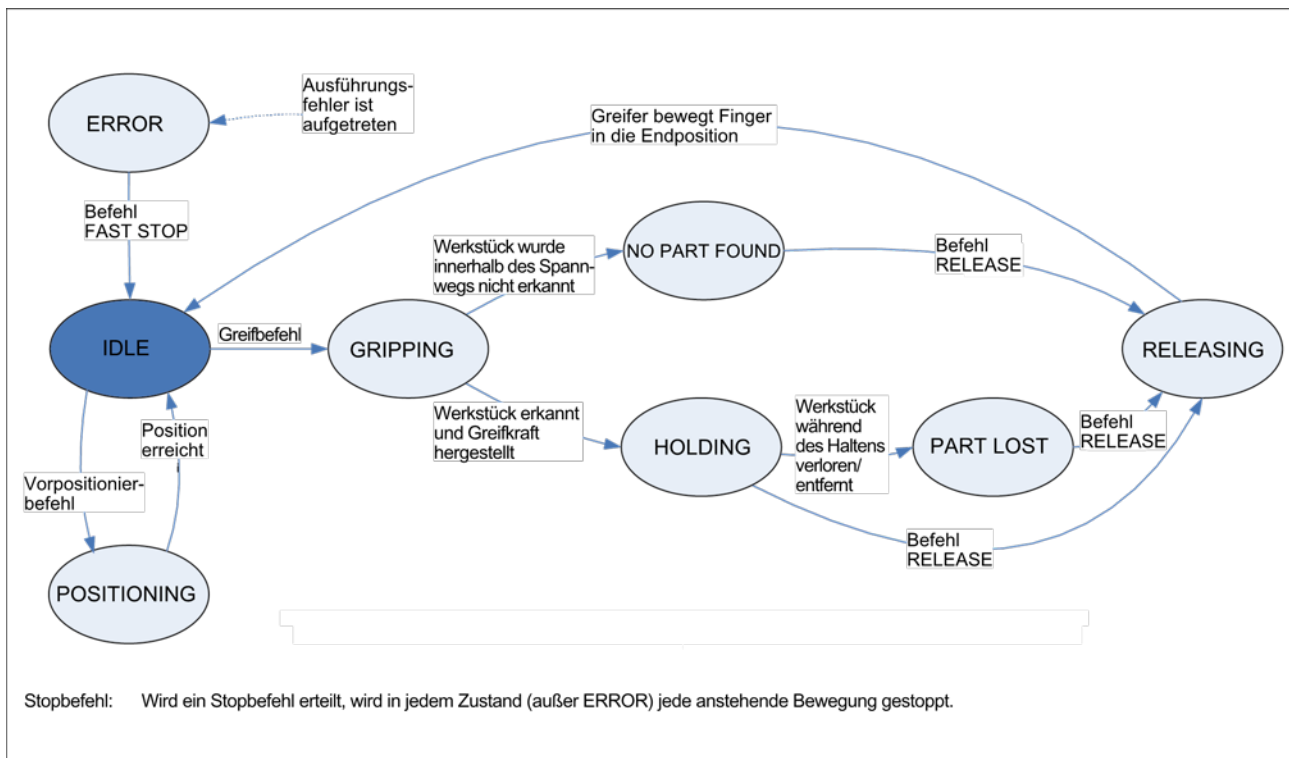
Bit Nr.	Flag Name	Beschreibung
D15	SF_POWER_FAULT	<p>Energieversorgung fehlerhaft</p> <p>Das Flag wird gesetzt, wenn die Versorgungsspannung außerhalb des zulässigen Bereichs liegt.</p> <p>Die Stromversorgung muss geprüft und ggf. angepasst werden.</p>
D14	SF_TEMP_FAULT	<p>Temperaturfehler</p> <p>Das Flag wird gesetzt, wenn der Greifer eine kritische Temperatur erreicht hat. Die Ausführung aller Bewegungsbefehle wird unterbunden.</p> <p>Das Flag wird zurückgesetzt, wenn die Temperatur wieder unter die kritische Grenze gefallen ist.</p>
D13	SF_TEMP_WARNING	<p>Temperaturwarnung</p> <p>Das Flag wird gesetzt, wenn der Greifer bald eine kritische Temperatur erreichen wird.</p>
D12	SF_FAST_STOP	<p>Fast Stop</p> <p>Das Flag wird gesetzt, wenn der Greifer aufgrund eines Fehlers gestoppt wurde. Der Antrieb wird abgeschaltet und die Ausführung aller Bewegungsbefehle unterbunden.</p> <p>Das Flag wird zurückgesetzt, wenn der Fehler quittiert wurde. Nach dem Quittieren werden die Bewegungsbefehle wieder aktiviert.</p>
D11..10	reserved	Diese Bits werden aktuell nicht verwendet.
D9	SF_FORCECNTL_MODE	<p>Kraft geregelter Modus</p> <p>Das Flag wird gesetzt, wenn die Kraftregelung unter Verwendung des installierten Kraftmessfingers (WSG-FMF) aktiv ist.</p> <p>Wenn das Flag nicht gesetzt ist, wird die Greifkraft auf Basis des Motorstroms näherungsweise geregelt.</p>

Bit Nr.	Flag Name	Beschreibung
D8	SF_OVERDRIVE_MODE	<p>Overdrive Modus¹</p> <p>Das Flag wird gesetzt, wenn sich der Greifer im Overdrive Modus befindet. Die Greifkraft kann bis auf den Wert der Overdrive Kraftgrenze erhöht werden.</p> <p>Wenn dieses Flag nicht gesetzt ist, kann die Greifkraft nicht höher als die Nennkraft des Greifers gesetzt werden.</p> <p>¹) Der Overdrive Modus wird nicht von allen Greifern unterstützt. Weitere Informationen befinden sich in der Montage- und Betriebsanleitung des Greifers.</p>
D7	SF_TARGET_POS_REACHED	<p>Zielposition erreicht</p> <p>Das Flag wird gesetzt, wenn die Zielposition erreicht wurde.</p> <p>Das Flag ist nicht mit SF_MOVING synchronisiert, sodass eine Verzögerung zwischen dem Zurücksetzen von SF_MOVING und dem Setzen von SF_TARGET_POS_REACHED auftreten kann.</p>
D6	SF_AXIS_STOPPED	<p>Achse gestoppt</p> <p>Das Flag wird gesetzt, wenn ein Bewegungsbe- fehl durch den Befehl STOP/ACK abgebrochen wurde.</p> <p>Das Flag wird durch den nächsten Bewegungs- befehl wieder zurückgesetzt.</p>
D5	SF_SOFT_LIMIT_PLUS	<p>Soft Limit in positiver Richtung erreicht</p> <p>Das Flag wird gesetzt, wenn die Greiferfinger die definierten Soft Limits in positiver Bewegungsrichtung erreicht haben. Eine weitere Bewegung in diese Richtung ist nicht möglich.</p> <p>Das Flag wird zurückgesetzt, wenn sich die Greiferfinger wieder von der Position entfernt haben.</p>

Bit Nr.	Flag Name	Beschreibung
D4	SF_SOFT_LIMIT_MINUS	<p>Soft Limit in negativer Richtung erreicht</p> <p>Das Flag wird gesetzt, wenn die Greiferfinger die definierten Soft Limits in negativer Bewegungsrichtung erreicht haben. Eine weitere Bewegung in diese Richtung ist nicht möglich. Das Flag wird zurückgesetzt, wenn sich die Greiferfinger wieder von der Position entfernen haben.</p>
D3	SF_BLOCKED_PLUS	<p>Achse ist in positiver Bewegungsrichtung blockiert</p> <p>Das Flag wird gesetzt, wenn die Achse in positiver Bewegungsrichtung blockiert ist. Das Flag wird zurückgesetzt, wenn die Blockade gelöst oder ein Stopbefehl erteilt wurde.</p>
D2	SF_BLOCKED_MINUS	<p>Achse ist in negativer Bewegungsrichtung blockiert</p> <p>Das Flag wird gesetzt, wenn die Achse in negativer Bewegungsrichtung blockiert ist. Das Flag wird zurückgesetzt, sobald die Blockade gelöst oder ein Stopbefehl erteilt wurde.</p>
D1	SF_MOVING	<p>Greiferfinger sind in Bewegung</p> <p>Das Flag wird gesetzt, wenn eine Bewegung gestartet wurde, z. B. Befehl MOVE. Das Flag wird zurückgesetzt, wenn die Bewegung stoppt.</p>
D0	SF_REFERENCED	<p>Greiferfinger sind referenziert</p> <p>Ist dieses Flag gesetzt, dann ist der Greifer referenziert und akzeptiert Bewegungsbefehle.</p>

5.3 Anhang C: Greifzustände

Das folgende Diagramm illustriert die Greifzustände und Übergänge, die im normalen Betrieb vorgesehen sind.



5.4 Anhang D: Demoprogramm

Im Lieferumfang des Greifers befinden sich einfache Demoprojekte für Siemens-Steuerungen SIMATIC S7-1200 sowie STEP 7 TIA-Portal v12 (und höher). Die Demoprojekte können von der CD oder von der Weboberfläche heruntergeladen werden.

Die Demoprojekte wurden auf einer CPU vom Typ 1212C mit dem PROFIBUS Modul CM1243-5 unter Verwendung der Siemens-Steuerung STEP7 Basic v12.0 SP2 (TIA Portal) Projektumgebung implementiert und getestet. Die Demoprojekte vollführen eine Endlosschleife einfacher Greifzyklen, bestehend aus Vorpositionieren der Greiferbacken (MOVE), Greifen eines Teils (GRIP), Loslassen (RELEASE) und Zurückkehren zur Startposition (MOVE).

Wenn ein Teil erkannt wurde, wird es vom Greifer für einen kurzen Moment festhalten. Im Fall eines Fehlers führt das Greifmodul eine Homingsequenz aus und beginnt wieder von vorn. Der Greifer muss vor Ausführung des Programms referenziert sein.

Die SPS ist in dem Projekt auf die IP-Adresse 192.168.1.250 und die PROFIBUS-Adresse 2 konfiguriert. Der Greifer wird auf der PROFIBUS-Adresse 7 (Standard) erwartet.

Bei Verwendung des PROFINET-Demoprogramms wird erwartet, dass das Greifmodul auf die IP-Adresse 192.168.1.20 eingestellt ist (Auslieferungszustand).

HINWEIS

Um mehr Informationen zu möglichen Problemen zu erhalten, ist es hilfreich, den Feldbusmonitor über die Weboberfläche zu öffnen.

HINWEIS

Das Demoprogramm ist nur für Testzwecke gedacht und darf nicht in einer Produktionsumgebung eingesetzt werden.

SCL-Quelltext des Greifzyklus-Bausteins in den Demo-Projekten

```
1  //////////////////////////////////////
2  // Receive data type           //
3  //////////////////////////////////////
4
5  TYPE "WSG_RECEIVE"
6  VERSION : 0.1
7  STRUCT
8      STW1 : Struct
9          IDLE : Bool; 9 GRASPING : Bool;
10         GRASPING : Bool;
11         NO_PART : Bool;
12         PART_LOST : Bool;
13         HOLDING : Bool;
14         RELEASING : Bool;
15         POSITIONING : Bool;
16         ERROR : Bool;
17         OF1 : Bool;
18         OF2 : Bool;
19         OF3 : Bool;
20         OF4 : Bool;
21         OF5 : Bool;
22         OF6 : Bool;
23         OF7 : Bool;
24         OF8 : Bool;
25     END_STRUCT;
26     SYSSTATE : Struct
27         REFERENCED : Bool;
28         MOVING : Bool;
29         BLOCKED_MINUS : Bool;
30         BLOCKED_PLUS : Bool;
31         SOFT_LIMIT_MINUS : Bool;
32         SOFT_LIMIT_PLUS : Bool;
33         AXIS_STOPPED : Bool;
34         TARGET_POS_REACHED : Bool;
35         OVERDRIVE_MODE : Bool;
36         FORCECNTL_MODE : Bool;
37         RES10 : Bool;
38         RES11 : Bool;
39         FAST_STOP : Bool;
40         TEMP_WARNING : Bool;
41         TEMP_FAULT : Bool;
42         POWER_FAULT : Bool;
43         CURR_FAULT : Bool;
```

```

44         FINGER_FAULT : Bool;
45         CMD_FAILURE  : Bool;
46         SCRIPT_RUN   : Bool;
47         SCRIPT_FAILURE : Bool;
48         RES21        : Bool;
49         RES22        : Bool;
50         RES23        : Bool;
51         RES24        : Bool;
52         RES25        : Bool;
53         RES26        : Bool;
54         RES27        : Bool;
55         RES28        : Bool;
56         RES29        : Bool;
57         RES30        : Bool;
58         RES31        : Bool;
59     END_STRUCT;
60     WIDTH : Int;
61     FORCE  : UInt;
62     ERROR_CODE : UInt;
63 END_STRUCT;
64
65 END_TYPE
66
67
68 ////////////////////////////////////////////////////
69 // Send data type                                     //
70 ////////////////////////////////////////////////////
71
72 TYPE "WSG_SEND"
73 VERSION : 0.1
74     STRUCT
75         STW1 : Struct
76             MOVE : Bool;
77             GRASP : Bool;
78             RELEASE : Bool;
79             HOMING : Bool;
80             STOP_ACK : Bool;
81             FASTSTOP : Bool;
82             JOG_PLUS : Bool;
83             JOG_MINUS : Bool;
84             IF1 : Bool;
85             IF2 : Bool;
86             IF3 : Bool;
87             IF4 : Bool;

```

```
88             IF5 : Bool;
89             IF6 : Bool;
90             IF7 : Bool;
91             IF8 : Bool;
92     END_STRUCT;
93     WIDTH : Int;
94     SPEED : UInt;
95     FORCELIMIT : UInt;
96 END_STRUCT;
97
98 END_TYPE
100
101 ////////////////////////////////////////////////////
102 // Gripping Cycle FB //
103 ////////////////////////////////////////////////////
104
105 FUNCTION_BLOCK "GrippingCycle"
106 { S7_Optimized_Access := 'TRUE' }
107 VERSION : 0.1
108 VAR DB_SPECIFIC
109     dp_data_in { S7_HMI_Visible := 'False' } : Array [1..12] of Byte;
110     receive { S7_HMI_Accessible := 'False'; S7_HMI_Visible := 'False' }
111 AT
112     dp_data_in : "WSG_RECEIVE";
113     dp_data_out { S7_HMI_Visible := 'False' } : Array [1..8] of Byte;
114     send { S7_HMI_Accessible := 'False'; S7_HMI_Visible := 'False' } AT
115 END_VAR
116 VAR
117     holding_active : Bool;
118     state : Int := -2;
119     timer_expired : Bool;
120     CycleConfig : Struct
121         PreposWidth : Int := 3000;
122         PreposSpeed : UInt := 40000;
123         PreposForce : UInt := 8000;
124         GraspWidth : Int := 2200;
125         GraspSpeed : UInt := 40000;
126         GraspForce : UInt := 5000; ;
127         HoldingTime : Time := T#1000ms
128         ReleaseWidth : Int := 3000;
129         ReleaseSpeed : UInt := 40000;
130         ReleaseForce : UInt := 8000;
131         StartWidth : Int := 6000;
```

```

132         StartSpeed : UInt := 40000;
133         StartForce : UInt := 8000;
134         CycleFinished : Bool := true;
135         ErrorCount : UInt := 0;
136         TimerExpired : Bool;
137     END_STRUCT;
138 END_VAR
139
140 VAR_TEMP
141     ret_val : Int;
142     do_next_step : Bool;
143 END_VAR
144
145
146 BEGIN
147 // Implementation of state machine
148
149 // Call receive function block
150 // Note: The address parameter comes from the Profibus module.
151 // Check default tag table, system constants tab, Profibus interface
and
152 // convert the decimal address listed there to hex and enter it here.
153 #ret_val := DPRD_DAT( LADDR := W#16#113, RECORD => #dp_data_in );
154
155 // Initial values
156 #do_next_step := false;
157
158 // State transitions
159 CASE #state OF
160
161     // NOTE: All states <= 0 belong to ERROR HANDLING!
162
163 // Step -1 (error state)
164     -1:
165     // Reset all control flags TO get a defined #state
166         #CycleConfig.ErrorCount := #CycleConfig.ErrorCount + 1;
167         #send.STW1.FASTSTOP := false;
168         #send.STW1.GRASP := false;
169         #send.STW1.HOMING := false;
170         #send.STW1.JOG_MINUS := false;
171         #send.STW1.JOG_PLUS := false;
172         #send.STW1.MOVE := false;
173         #send.STW1.RELEASE := false;

```

```
174
175 // Set STOP/ACK flag to true to resolve error condition
176     #send.STW1.STOP_ACK := true;
177
178     // Go to next step
179     #do_next_step := true;
180
181 // Step 0 (initial start): Execute homing sequence
182     0:
183     // Error handling. State *must* be IDLE at this point.
184     IF #receive.STW1.IDLE = false THEN
185         #state := -1;
186     END_IF;
187
188
189     // Reset STOP/ACK flag and set HOMING command flag
190     IF #receive.STW1.IDLE = true THEN
191         #send.STW1.STOP_ACK := false;
192         #send.STW1.HOMING := true;
193         #do_next_step := true;
194     END_IF;
195
196 // Step 1: Check if HOMING is running
197     1:
198     // Error handling
199     IF #receive.STW1.ERROR = true THEN
200         #state := -1;
201     END_IF;
202
203     // Check for gripper state set to POSITIONING
204     IF #receive.STW1.POSITIONING = true THEN
205         #send.STW1.HOMING := false;
206         #do_next_step := true;
207     END_IF;
208
209 // Step 2: Wait for gripper state to become IDLE
210 // and check if gripper is referenced
211     2:
212     IF #receive.STW1.ERROR = true THEN
213         #state := -1;
214     END_IF;
215
216     IF #receive.STW1.IDLE = true THEN
217         IF #receive.SYSSTATE.REFERENCED = false THEN
218             #state := -1;
```

```

219         ELSE
220             #do_next_step := true;
221             END_IF;
222         END_IF;
223
224 // Step 3: When idle, move to pre-position width
225     3:
226         // Error handling
227         IF #receive.STW1.ERROR = true THEN
228             #state := -1;
229         END_IF;
230
231         // Trigger move command to pre-position the gripper jaws
232         IF #receive.STW1.IDLE = true THEN
233             #send.WIDTH := #CycleConfig.PreposWidth;
234             #send.SPEED := #CycleConfig.PreposSpeed;
235             #send.FORCELIMIT := #CycleConfig.PreposForce;
236             #send.STW1.MOVE := true;
237             #do_next_step := true;
238         END_IF;
239
240 // Step 4: Check if gripper state is set to POSITIONING, i.e. gripper
241 is moving
242     4:
243         // Error handling
244         IF #receive.STW1.ERROR = true THEN
245             #state := -1;
246         END_IF;
247
248         // Reset move command flag
249         IF #receive.STW1.POSITIONING = true THEN
250             #send.STW1.MOVE := false;
251             #do_next_step := true;
252         END_IF;
253
254 // Step 5: When idle, start grasping
255     5:
256         IF #receive.STW1.ERROR = true THEN
257             #state := -1;
258         END_IF;
259
260         IF #receive.STW1.IDLE = true THEN
261             #send.WIDTH := #CycleConfig.GraspWidth;
262             #send.SPEED := #CycleConfig.GraspSpeed;

```



```
263         #send.FORCELIMIT := #CycleConfig.GraspForce;
264         #send.STW1.GRASP := true;
265         #do_next_step := true;
266         END_IF;
267
268 // Step 6: When grasping is active, go to next step
269     6:
270         IF #receive.STW1.ERROR = true THEN
271             #state := -1;
272             END_IF;
273
274         #IF receive.STW1.GRASPING = true THEN
275             #send.STW1.GRASP := false;
276             #do_next_step := true;
277         END_IF;
278
279 // Step 7: When holding, wait. If no part found/part lost, release -
280 immediately.
281     7:
282         IF #receive.STW1.ERROR = true THEN
283             #state := -1;
284             END_IF;
285
286         // Part found. Hold for some time, then go to next step.
287         IF #receive.STW1.HOLDING = true AND #holding_active = false THEN
288             #holding_active := true;
289             #ret_val := SRT_DINT( OB_NR := 20, DTIME := #CycleConfig. HoldingTime,
291 SIGN := 1 );
290             END_IF;
291
292
293         // No part found or part lost. Go to next step.
294         IF #timer_expired = true OR #receive.STW1.NO_PART = true
295 OR #receive.STW1.PART_LOST = true THEN
296             #holding_active := false;
297             #timer_expired := false;
298             #send.WIDTH := #CycleConfig.ReleaseWidth;
299             #send.SPEED := #CycleConfig.ReleaseSpeed;
300             #send.FORCELIMIT := #CycleConfig.ReleaseForce;
301             #send.STW1.RELEASE := true;
302             #do_next_step := true;
303         END_IF;
304
305 // Step 8: When releasing is active, go to next step
306     8:
307         IF #receive.STW1.ERROR = true THEN
```

```

308         #state := -1;
309     END_IF;
310
311     IF #receive.STW1.RELEASING = true THEN
312         #send.STW1.RELEASE := false;
313         #do_next_step := true;
314     END_IF;
315
316 // Step 9: When idle, move to start position
317     9:
318     IF #receive.STW1.ERROR = true THEN
319         #state := -1;
320     END_IF;
321
322     IF #receive.STW1.IDLE = true THEN
323         #send.WIDTH := #CycleConfig.StartWidth;
324         #send.SPEED := #CycleConfig.StartSpeed;
325         #send.FORCELIMIT := #CycleConfig.StartForce;
326         #send.STW1.MOVE := true;
327         #do_next_step := true;
328     END_IF;
329
330 // Step 10: When positioning is active, go to next step
331     10:
332     IF #receive.STW1.ERROR = true THEN
333         #state := -1;
334     END_IF;
335
336     IF #receive.STW1.POSITIONING = true THEN
337         #send.STW1.MOVE := false;
338         #do_next_step := true;
339     END_IF;
340
341     // Default (state is not -1..10)
342     ELSE
343
344     // Go to error state
345     IF #receive.STW1.ERROR = true THEN
346         #state := -1;
347     END_IF;
348
349     // Start cycle from the beginning (without homing)
350     IF #receive.STW1.IDLE = true THEN
351         #state := 3;

```

```
352             #CycleConfig.CycleFinished := true;
353         END_IF;
354
355     END_CASE;
356
357     // Increment state variable
358     IF #do_next_step = true THEN
359         #state := #state + 1;
360     END_IF;
361
362     // Call send function block
363     #ret_val := DPWR_DAT( LADDR := W#16#114, RECORD := #dp_data_out );
364
365 END_FUNCTION_BLOCK
```

