

<Bedienungsanleitung\_Basic Gripping\_S7\_V1\_21(DE).docx>

Thema:

<FB Basic Gripping>

Version:

<1.21>

### Historie

Bearbeiter	Änderungsgrund/Änderungsumfang	Version	Freigabedatum
Armbruster	Erstellung	1.0	
Nock	Anzeige Parameteränderung mit Ausgangsbit Automatischer Reset der Richtungsmerker	1.21	18.03.2020

## Inhalt

1	Vorwort.....	4
2	Einbinden der Bibliothek .....	4
3	Datentypen deklarieren .....	4
4	Globale Strukturen im OB deklarieren .....	7
5	Prozessdaten mit Peripherie verknüpfen .....	7
6	Funktionsbaustein einfügen .....	8
7	Verwendung des Funktionsbausteins .....	10
8	Funktionen des Funktionsbausteins .....	11
8.1	Schrittfolge zurücksetzen „cmd_b_StepReset“ (BOOL).....	11
8.2	Datenübertragung mit Handshake „cmd_b_DataTransfer“ (BOOL) .....	11
8.3	Werkstückrezeptur speichern „cmd_b_WritePDU“ (BOOL).....	12
8.4	Richtungsmerker zurücksetzen „cmd_b_ResetDirectionFlag“ (BOOL).....	12
8.5	Fahren auf BasePosition „cmd_b_MoveToBase“ (BOOL) .....	12
8.6	Fahren auf WorkPosition „cmd_b_MoveToWork“ (BOOL) .....	12
8.7	Begrenzung der Bewegungszeit „t_MotionTimeout“ (TIME) und „b_MotionError“ (BOOL).....	12
8.8	Datenübertragung wird benötigt „b_DataTransferRequired“ (BOOL).....	12
8.9	Fehler in der Datenübertragung „b_DataTransferError“ (BOOL).....	12
8.10	Befehl wird ausgeführt „b_StepBusy“ (BOOL).....	13
8.11	Bereit für Befehle „b_StepDone“ (BOOL) .....	13
8.12	Bit 6 des StatusWord „b_GripperPLCActive“ (BOOL) .....	13
8.13	Bit 8 des StatusWord „b_BasePosition“ (BOOL) .....	13
8.14	Bit 9 des StatusWord „b_TeachPosition“ (BOOL) .....	13
8.15	Bit 10 des StatusWord „b_WorkPosition“ (BOOL) .....	13
8.16	Bit 11 des StatusWord „b_UndefinedPosition“ (BOOL).....	13
8.17	Bit 12 des StatusWord „b_DataTransferOK“ (BOOL) .....	13
8.18	Bit 13 des StatusWord „b_ControlWord_100“ (BOOL).....	13
8.19	Bit 14 des StatusWord „b_ControlWord_200“ (BOOL).....	13
8.20	Bit 15 des StatusWord „b_Error“ (BOOL) und „n_Diagnose“ (WORD).....	13
8.21	n_ActualPosition (WORD).....	13

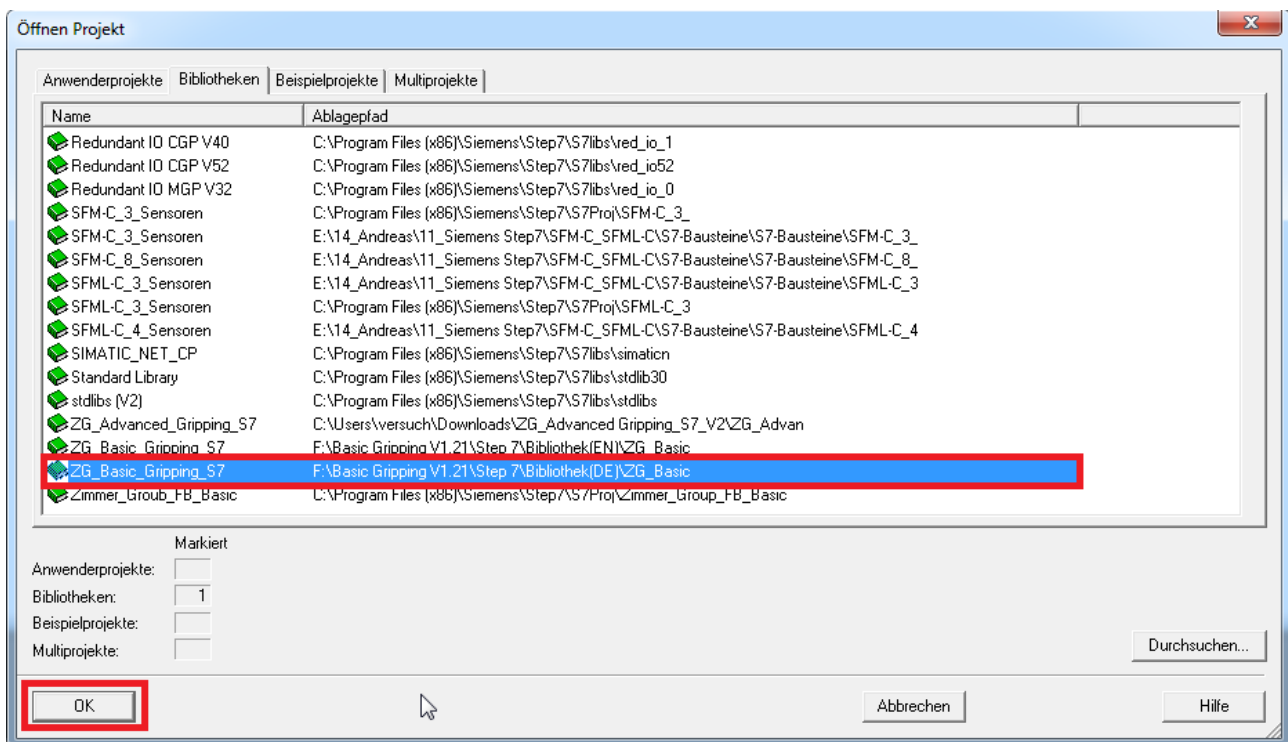
### 1 Vorwort

Für die Nutzung des Beispielprogramms muss zuerst eine korrekte Hardwarekonfiguration erstellt werden. In diesem Beispiel wird eine Siemens CPU314C-2 PN/DP mit einem Balluff IO Link Master verwendet. Nach den Hardwareeinstellungen kann der Funktionsbaustein implementiert werden. Dazu gehen Sie die nachfolgenden Schritte durch.

### 2 Einbinden der Bibliothek

Der Funktionsbaustein wird als Bibliothek zur Verfügung gestellt. Zum Öffnen wählen Sie bitte das Menü „Datei“ -> „Öffnen...“ aus. In dem nun sichtbaren Fenster können Sie die gewünschte Bibliothek auswählen und öffnen.

Bitte wählen Sie die globale Bibliothek „**ZG\_Basic\_Gripping\_S7**“ aus.



Nun ist die Zimmer Bibliothek für die Verwendung bereit und kann in Ihrem Projekt verwendet werden.

### 3 Datentypen deklarieren

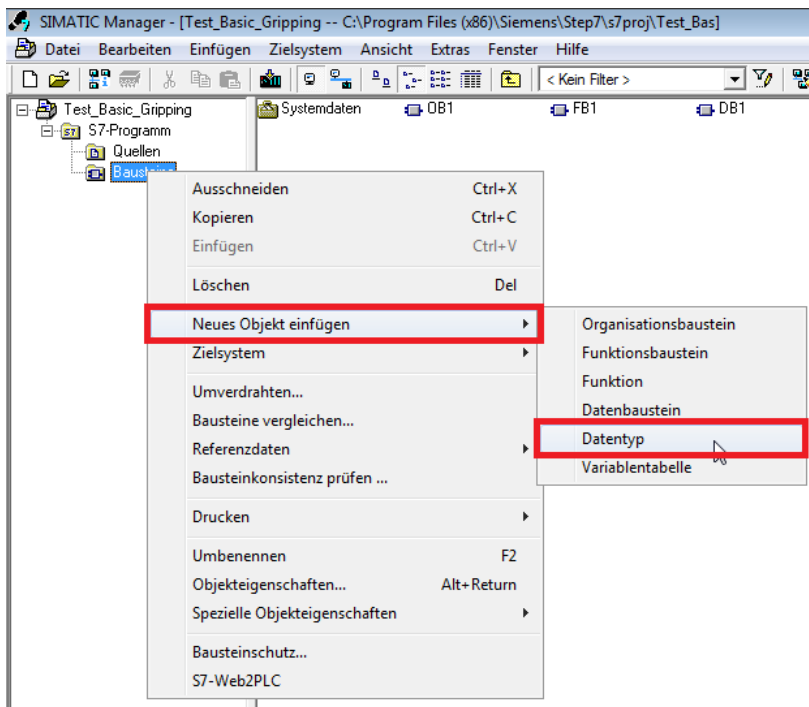
Die Ein- und Ausgangsdaten des Greifers - oder auch „Prozessdaten“ genannt – werden über spezifische Datentypen (Strukturen oder UDT) an den Baustein übergeben. Zur Kommunikation zwischen Funktionsbaustein und Greifer müssen folgende Datentypen angelegt werden:

Name	Symbolischer Name	Umfasst
UDT1	UDT_Basic_IN	6 Bytes Eingangsdaten
UDT2	UDT_Basic_OUT	8 Bytes Ausgangsdaten

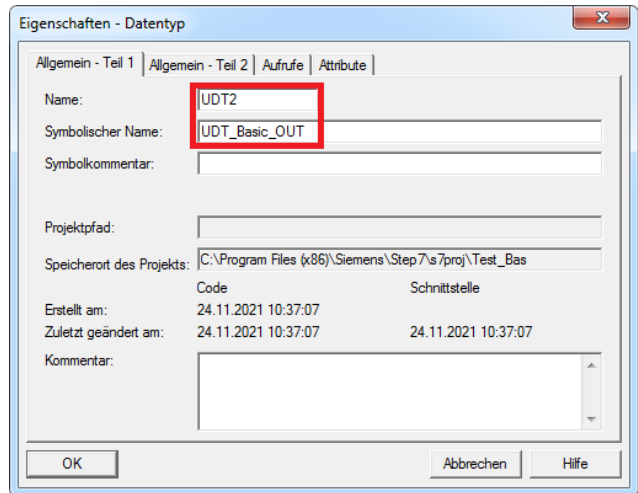
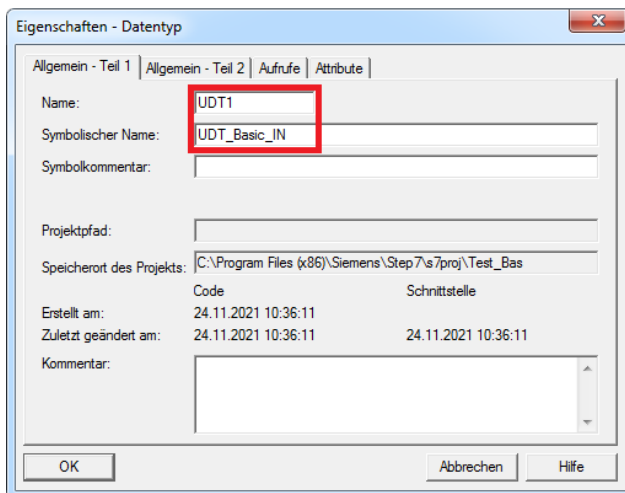
Ein neuer Datentyp kann mit Rechtsklick auf „Bausteine“ -> „Neues Objekt einfügen“ -> „Datentyp“ in das Projekt eingefügt werden.

# Bedienungsanleitung Funktionsbaustein (Step 7)

## Basic Gripping



In dem sich öffnenden Fenster wird der „Symbolische Name“ des UDT festgelegt. Bitte geben Sie den vorgegeben Namen ein.



Bitte öffnen Sie den entsprechenden Datentyp und definieren Sie ihn wie folgt. Die Reihenfolge sollte dabei eingehalten werden.

UDT1:

Name	Typ	Anfangswert
n_StatusWord	WORD	W#16#0
n_Diagnose	WORD	W#16#0
n_ActualPosition	WORD	W#16#0

## Bedienungsanleitung Funktionsbaustein (Step 7)

### Basic Gripping

UDT2:

Name	Typ	Anfangswert
n_ControlWord	WORD	W#16#0
n_DeviceMode	BYTE	B#16#0
n_WorkpieceNo	BYTE	B#16#0
n_TeachPosition	WORD	W#16#0
n_GripForce	BYTE	B#16#0
n_PositionTolerance	BYTE	B#16#0

KOP/AWL/FUP - [UDT1 -- "UDT\_Basic\_IN" -- Test\_Basic\_Gripping\S7-Programm\... UDT1]

Datei Bearbeiten Einfügen Zielsystem Test Ansicht Extras Fenster Hilfe

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	n_StatusWord	WORD	W#16#0	
+2.0	n_Diagnose	WORD	W#16#0	
+4.0	n_ActualPosition	WORD	W#16#0	
=6.0		END_STRUCT		

KOP/AWL/FUP - [UDT2 -- "UDT\_Basic\_OUT" -- Test\_Basic\_Gripping\S7-Programm\... UDT2]

Datei Bearbeiten Einfügen Zielsystem Test Ansicht Extras Fenster Hilfe

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	n_ControlWord	WORD	W#16#0	
+2.0	n_DeviceMode	BYTE	B#16#0	
+3.0	n_WorkpieceNo	BYTE	B#16#0	
+4.0	n_TeachPosition	WORD	W#16#0	
+6.0	n_GripForce	BYTE	B#16#0	
+7.0	n_PositionTolerance	BYTE	B#16#0	
=8.0		END_STRUCT		

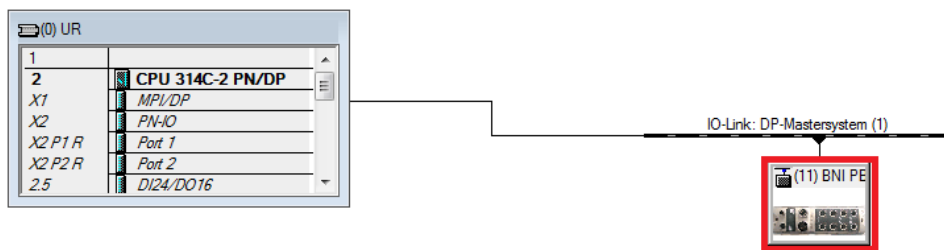
### 4 Globale Strukturen im OB deklarieren

Im verwendeten Operationsbaustein müssen von den angelegten Strukturen globale Variablen angelegt werden, um sie mit dem Baustein verknüpfen zu können. Legen Sie bitte im OB in der Deklarationstabelle die folgenden Variablen an:

Name	Datentyp	Adresse	Kommentar
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date And Time	12.0	Date and time OB1 started
g_UDT_Gripper1_IN	UDT_Basic_IN	20.0	
g_UDT_Gripper1_OUT	UDT_Basic_OUT	26.0	

### 5 Prozessdaten mit Peripherie verknüpfen

In der Hardwarekonfiguration wurden dem IO-Link-Port, an dem der Greifer angeschlossen ist, Adressbereiche für die Ein- und Ausgangsdaten zugewiesen. In diesem Beispiel startet der Bereich bei Peripherieadresse 256.



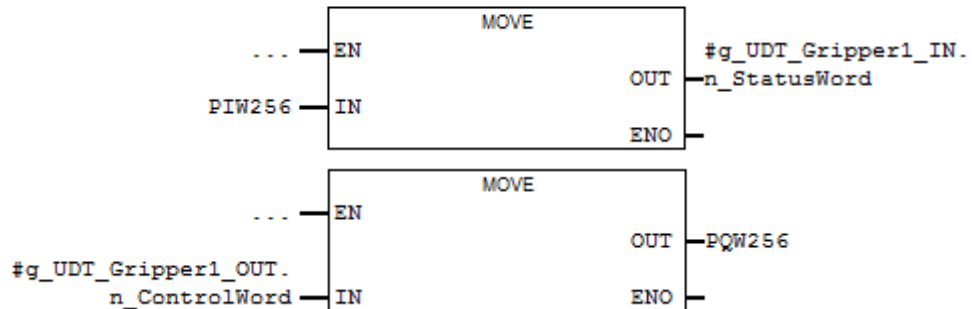
Steckplatz	DP-Kennung	Bestellnummer / Bezeichnung	E-Adresse	A-Adresse	Kommentar
1	0	BNI PBS-502-101-Z001			
2	192	IO_L I/O 8/ 8 byte	256..263	256..263	
3	0	Standard I/O			
4	0	Standard I/O			
5	0	Standard I/O			

Im Operationsbaustein müssen die Peripherieadressen mit den angelegten globalen Strukturen verknüpft werden. In unserem Beispiel wurden für die Zuweisungen „MOVE“-Bausteine verwendet. Die Eingangsdaten sollten im OB vor dem FB und die Ausgangsdaten nach dem FB verknüpft werden. Beim Zuweisen der Adressen muss die Reihenfolge der globalen Struktur eingehalten werden.

In diesem Beispiel sehen die Zuweisungen – beginnend mit Adresse 256 – aus wie folgt:

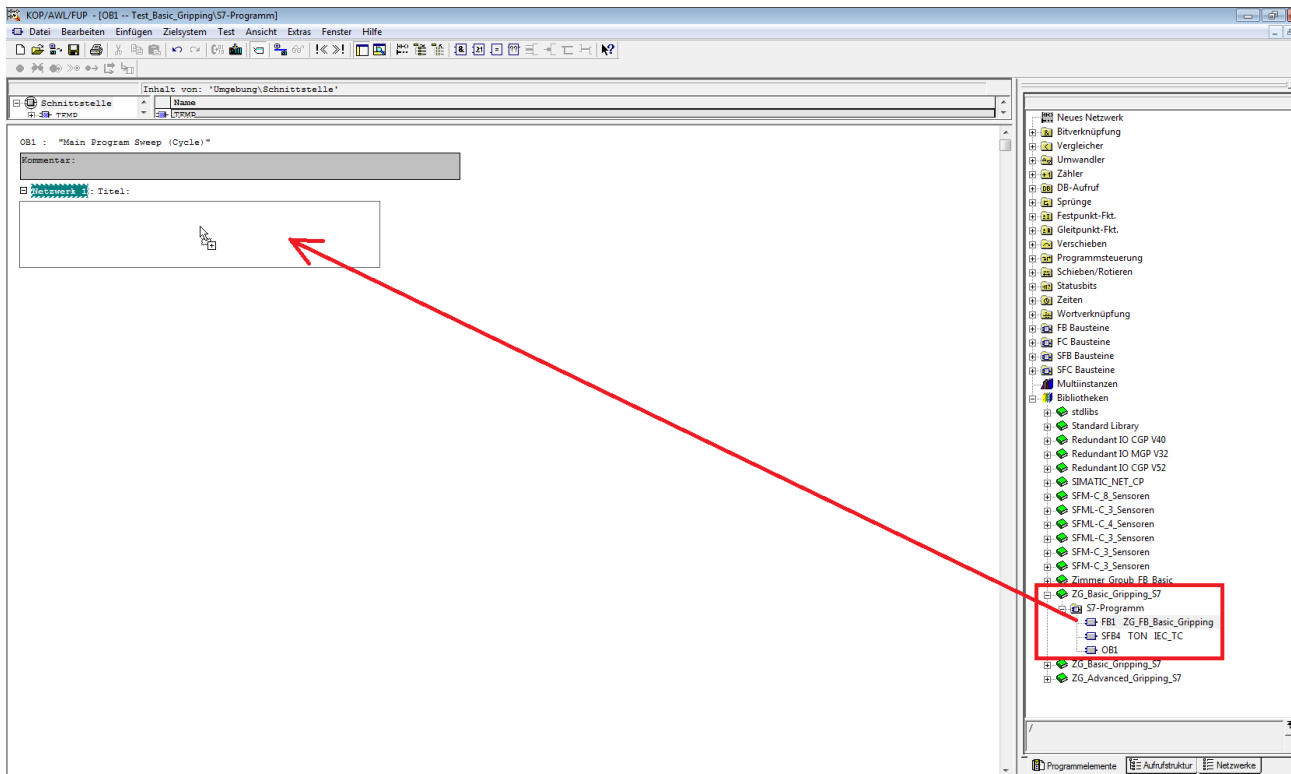
Zuweisung von ...	Zuweisung zu ...
PIW256	g_UDT_Gripper1_IN.n_StatusWord
PIW258	g_UDT_Gripper1_IN.n_Diagnose
PIW260	g_UDT_Gripper1_IN.n_ActualPosition
g_UDT_Gripper1_OUT.n_ControlWord	PQW256
g_UDT_Gripper_1OUT.n_DeviceMode	PQB258
g_UDT_Gripper_1OUT.n_WorkpieceNo	PQB259
g_UDT_Gripper_1OUT.n_TeachPosition	PQW260
g_UDT_Gripper_1OUT.n_GripForce	PQB262
g_UDT_Gripper_1OUT.n_PositionTolerance	PQB263

Netzwerk 1: Eingang: StatusWord



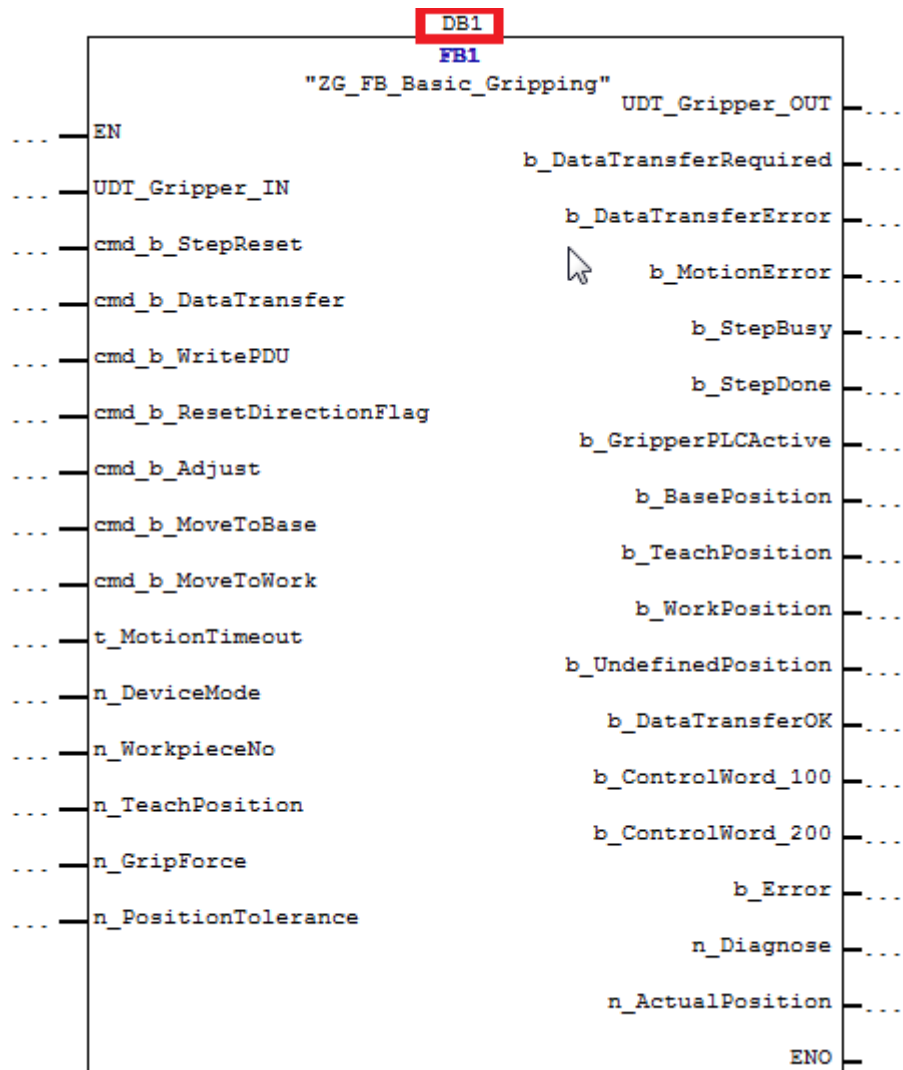
## 6 Funktionsbaustein einfügen

Um den „Basic-Gripping“-Baustein verwenden zu können, muss er im gewünschten Operationsbaustein aufgerufen werden. Der Baustein „ZG\_FB\_Basic\_Gripping“ kann aus der bereits eingebundenen Bibliothek per Drag&Drop in ein freies Netzwerk kopiert werden. Der Standardbaustein „SFB4 (TON)“ ist auch in dieser Bibliothek vorhanden, da er im „Basic-Gripping“-Baustein benötigt wird.





Da es sich um einen Funktionsbaustein handelt, wird ein Instanz-Datenbaustein benötigt. In diesem Beispiel wurde ein neuer Datenbaustein DB1 generiert.



Variablen, welche mit „b\_“ deklariert sind, stellen binäre Signale.

Variablen, welche mit „cmd\_“ deklariert sind, stellen Befehlseingänge dar. Diese können z.B. mit einem Taster gesteuert werden.

Variablen, welche mit „n\_“ deklariert sind, sind Eingangs- bzw. Ausgangsbytes bzw. -wörter. Diese werden für die Übertragung der einzelnen Positionen und Funktionen benötigt.

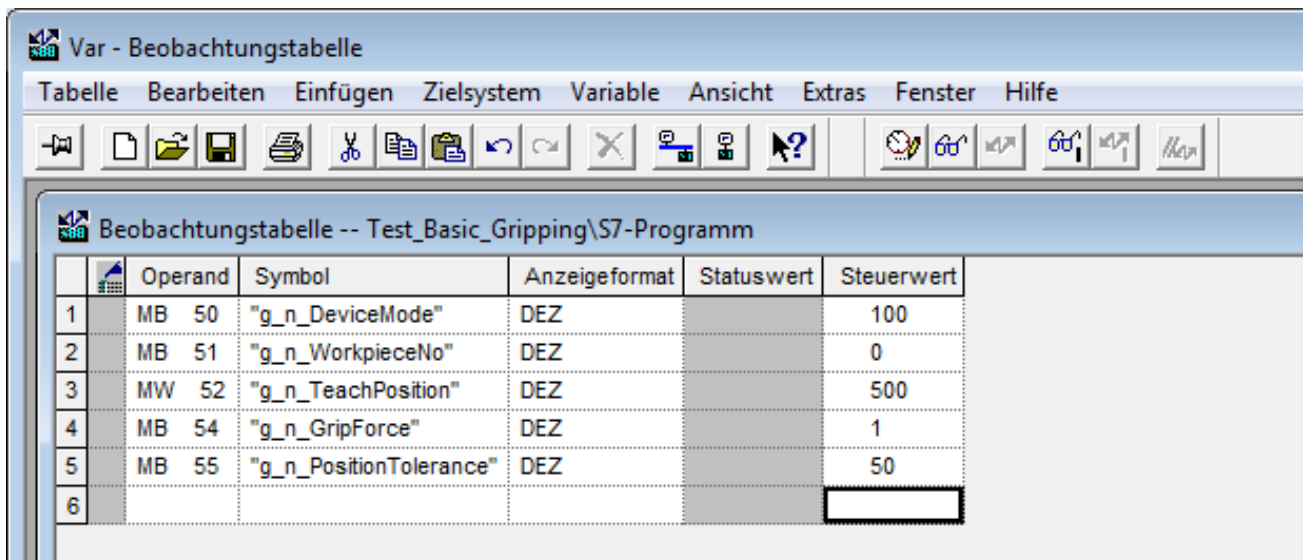
### 7 Verwendung des Funktionsbausteins

An dem unter Punkt 6 eingefügten Funktionsbaustein sind nun mehrere Ein – bzw. Ausgänge zu beschalten. Der am Baustein befindliche Eingang „UDT\_Gripper\_IN“ muss nun mit der zugehörigen Variable aus Punkt 4 verbunden werden. Das gleiche muss auch mit der Ausgangsvariable „UDT\_Gripper\_OUT“ durchgeführt werden. Nun kann der Baustein die einzelnen Zustände und Positionen des Greifer auslesen und diese im Baustein verarbeiten. Des Weiteren kann durch die Ausgangsbeschaltung der Greifer parametrisiert werden.

Damit der Greifer gefahren werden kann, müssen diesem die verschiedenen Positionsdaten und Optionen (Fahrprofile) übertragen werden. Als Standardwerte können die in folgender Tabelle aufgeführten Werte verwendet werden. Diese sind beispielhaft und können projektspezifisch variieren. Sie können diese Parameter wie in diesem Beispiel als Konstanten am Baustein eintragen oder auch Variablen in entsprechender Länge verwenden, damit die Beschaltung flexibel ist. Bei Nichtbeschaltung sind die Variablen mit den Standardwerten vorinitialisiert.

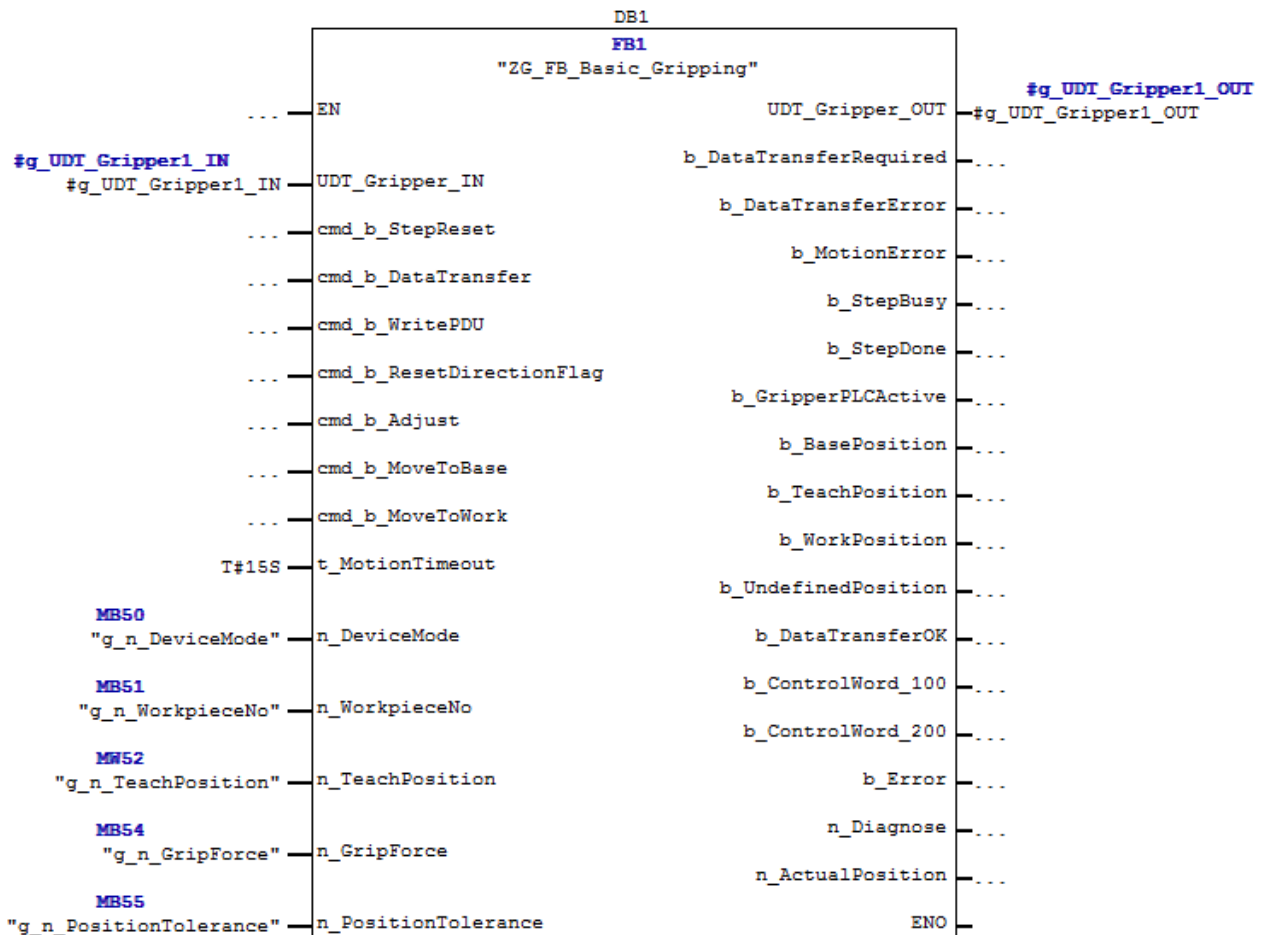
n_DeviceMode	100 (1 bei GEP/GED5000IL)
n_WorkpieceNo	0
n_TeachPosition	500
n_GripForce	1
n_PositionTolerance	50

Die Variable n\_DeviceMode entspricht dem Fahrprofil des Greifers. Diese Fahrprofile können der Montageanleitung des Greifers entnommen werden. In diesem Beispiel wurde der DeviceMode 100 (bei GEP2000IL oder GPP5000IL) bzw. 1 (bei GEP/GED5000IL) gewählt, welcher dem Fahrprofil „Universalbetrieb“ entspricht. Dieser kann als Standardwert verwendet werden.



	Operand	Symbol	Anzeigeformat	Statuswert	Steuerwert
1	MB 50	"g_n_DeviceMode"	DEZ		100
2	MB 51	"g_n_WorkpieceNo"	DEZ		0
3	MW 52	"g_n_TeachPosition"	DEZ		500
4	MB 54	"g_n_GripForce"	DEZ		1
5	MB 55	"g_n_PositionTolerance"	DEZ		50
6					

Der fertige Baustein sollte nun folgender Abbildung entsprechen:



Abschließend müssen Sie die Einstellungen übersetzen und in das Gerät laden.

## 8 Funktionen des Funktionsbausteins

Abhängig von der Eingangsbeschaltung des Funktionsbausteins werden die entsprechenden Funktionen ausgeführt. Weitere Informationen können Sie auch im Bausteinkopf finden.

### 8.1 Schrittkette zurücksetzen „cmd\_b\_StepReset“ (BOOL)

Die Eingangsvariable „cmd\_b\_StepReset“ setzt die Schrittkette innerhalb dieses Bausteins zurück. Dies geschieht unabhängig davon, in welchem Schritt sich der Baustein gerade befindet. Wenn der Baustein den Fehler „b\_DataTransferError“ oder „b\_MotionError“ ausgibt, kann er nur durch diesen Eingang quittiert werden.

### 8.2 Datenübertragung mit Handshake „cmd\_b\_DataTransfer“ (BOOL)

Nach jeder Änderung eines Prozessparameters (ausgenommen „ControlWord“) oder bei einem Kaltstart des Greifers müssen die Parameter mit einer Datenübertragung übernommen werden. Wenn die Ausgangsvariable „b\_DataTransferRequired“ „TRUE“ ist, arbeitet der Greifer noch nicht mit den aktuell eingestellten Parametern. In diesem Fall muss der Eingang „cmd\_b\_DataTransfer“ getriggert werden, damit die Prozessparameter übertragen werden. Die Variable „b\_DataTransferRequired“ wechselt anschließend auf „FALSE“. Dabei wird das „ControlWord“ auf Wert 1 gesetzt und auf das Bit 12 des „StatusWord“ gewartet. Das Bit 12 wird „TRUE“, sobald die Datenübertragung abgeschlossen ist. Anschließend wird das „ControlWord“ wieder auf 0 gesetzt und gewartet, bis das Bit 12 „FALSE“ wird. Diese Prozedur ist ein Handshake und sollte für die fehlerfreie Datenübertragung angewandt werden.

#### 8.3 Werkstückrezeptur speichern „cmd\_b\_WritePDU“ (BOOL)

Wenn dieser Eingang auf „TRUE“ gesetzt wird, werden die aktuell gesetzten Prozessparameter am Bausteineingang in die aktuell eingestellte „WorkpieceNo“ gespeichert. Dabei wird das „ControlWord“ auf Wert 2 gesetzt und auf das Bit 12 des „StatusWord“ gewartet. Diese Prozedur kann bis zu 30 Sekunden dauern. Die Parameter werden stromausfallsicher im Greifer abgespeichert und können über die Angabe der „WorkpieceNo“ wieder angewählt werden. Es können bis zu 32 Rezepturen im Greifer abgespeichert werden.

#### 8.4 Richtungsmerker zurücksetzen „cmd\_b\_ResetDirectionFlag“ (BOOL)

Wenn ein Greifer z.B. in Richtung „WorkPosition“ gefahren wird, wird im Greifer das Bit 14 des „StatusWord“ gesetzt. Dieses Signal bleibt bis zu einer Bewegung in die andere Richtung oder einem Kaltstart des Greifers bestehen. Wenn ein Greifer mehrmals nacheinander in die gleiche Richtung gefahren werden soll, dann muss dieses Bit zuerst zurückgesetzt werden. Dies kann durch den Eingang „cmd\_b\_ResetDirectionFlag“ durchgeführt werden. Dabei wird das „ControlWord“ auf den Wert 4 gesetzt und gewartet bis Bit 13 und Bit 14 des „StatusWord“ auf „FALSE“ wechseln. Danach kann eine erneute Bewegung in die gleiche Richtung erfolgen. Ab der Bausteinversion V1.21 wird diese Prozedur falls notwendig vor der Bewegung des Greifers automatisch durchgeführt.

#### 8.5 Fahren auf BasePosition „cmd\_b\_MoveToBase“ (BOOL)

Beim Setzen dieses Eingangssignals auf „TRUE“ bewegen sich die Greiferbacken mit dem eingestellten Fahrprofil und mit der eingestellten Greifkraft auf die „BasePosition“. Dabei wird das „ControlWord“ auf den Wert 256 gesetzt.

#### 8.6 Fahren auf WorkPosition „cmd\_b\_MoveToWork“ (BOOL)

Beim Setzen dieses Eingangssignals auf „TRUE“ bewegen sich die Greiferbacken mit dem eingestellten Fahrprofil und mit der eingestellten Greifkraft auf die „WorkPosition“. Dabei wird das „ControlWord“ auf den Wert 512 gesetzt.

#### 8.7 Begrenzung der Bewegungszeit „t\_MotionTimeout“ (TIME) und „b\_MotionError“ (BOOL)

Falls der Greifer eine Bewegung nicht ausführen kann oder das gewünschte Ziel nicht erreicht, würde die Schrittkette stehenbleiben und der Baustein wäre für weitere Befehle blockiert. Damit dies nicht passiert, kann die Zeit „t\_MotionTimeout“ am Baustein definiert werden, welche der Greifer maximal für seine Bewegung bis zur Position benötigen darf. Dies ist abhängig von den Eingangsparametern des Greifers und muss projektspezifisch angepasst werden. Wenn der Greifer innerhalb der eingestellte Zeit sein Ziel nicht erreicht, springt die Schrittkette in einen Fehlerschritt. Der Ausgang „b\_MotionError“ wird auf „TRUE“ gesetzt und kann nur durch Setzen des Eingangs „cmd\_b\_StepReset“ wieder zurückgesetzt werden.

#### 8.8 Datenübertragung wird benötigt „b\_DataTransferRequired“ (BOOL)

Die Variable „b\_DataTransferRequired“ ist aktiv, wenn mindestens eine der Output-Variablen, die zum Greifer geschickt werden, geändert wurde. Solange diese Variable aktiv ist, hat der Greifer die geänderten Werte noch nicht übernommen. Zur Datenübernahme muss die Eingangsvariable „cmd\_b\_DataTransfer“ getriggert werden. Die Variable „b\_DataTransferRequired“ wechselt dann auf „FALSE“ und der Greifer verwendet die aktuell eingestellten Parameter.

#### 8.9 Fehler in der Datenübertragung „b\_DataTransferError“ (BOOL)

Der Ausgang „b\_DataTransferError“ wird auf „TRUE“ gesetzt, wenn die Datenübertragung („ControlWord“ = 1) nicht erfolgreich durchgeführt werden konnte und das Feedback des Greifers nicht innerhalb einer Sekunde gesendet wurde. Dies kann mehrere Gründe zur Ursache haben. Einen Fehlercode kann über den Ausgang „n\_Diagnose“ entnommen werden. Die einzelnen Fehlercodes sind in der Montageanleitung näher beschrieben. Durch Setzen des Eingangs „cmd\_b\_StepReset“ kann dieser Fehler quittiert werden.

#### 8.10 Befehl wird ausgeführt „b\_StepBusy“ (BOOL)

Wenn der Baustein einen Befehl bearbeitet und in einem Schritt steht, ist dieser Ausgang aktiv und signalisiert, dass er für weitere Befehle blockiert ist.

#### 8.11 Bereit für Befehle „b\_StepDone“ (BOOL)

Wenn der Baustein im Initialschritt steht und bereit für Befehle ist, ist dieser Ausgang auf „TRUE“.

#### 8.12 Bit 6 des StatusWord „b\_GripperPLCActive“ (BOOL)

Dieses Signal gibt die Betriebsbereitschaft der Steuerung im Greifer wieder. Bei einem Kaltstart oder Neustart nach Spannungsausfall, kann die Steuerung erst dann wieder Daten empfangen, wenn dieses Signal „TRUE“ ist.

#### 8.13 Bit 8 des StatusWord „b\_BasePosition“ (BOOL)

Sobald der Greifer seine eingestellte „BasePosition“ erreicht hat, wird dieses Signal aktiviert. Die Größe des Bereichs wird durch die „PositionTolerance“ definiert.

#### 8.14 Bit 9 des StatusWord „b\_TeachPosition“ (BOOL)

Sobald der Greifer seine eingestellte „TeachPosition“ erreicht hat, wird dieses Signal aktiviert. Die Größe des Bereichs wird durch die „PositionTolerance“ definiert.

#### 8.15 Bit 10 des StatusWord „b\_WorkPosition“ (BOOL)

Sobald der Greifer seine eingestellte „WorkPosition“ erreicht hat, wird dieses Signal aktiviert. Die Größe des Bereichs wird durch die „PositionTolerance“ definiert.

#### 8.16 Bit 11 des StatusWord „b\_UndefinedPosition“ (BOOL)

Wenn der Greifer stillsteht und weder auf „BasePosition“, „TeachPosition“ oder „WorkPosition“ steht, ist dieses Signal „TRUE“.

#### 8.17 Bit 12 des StatusWord „b\_DataTransferOK“ (BOOL)

Mit diesem Bit gibt der Greifer die Rückmeldung, dass eine Datenübertragung („ControlWord“ = 1) erfolgreich durchgeführt wurde. Daher wird es bei einer Handshakeprozedur verwendet.

#### 8.18 Bit 13 des StatusWord „b\_ControlWord\_100“ (BOOL)

Dieser Richtungsmerker wird „TRUE“, wenn der Greifer einen „MoveToBase“-Befehl erhalten hat. Der Greifer kann in diesem Zustand keinen weiteren „MoveToBase“-Befehl ausführen. Der Merker wird wieder auf „FALSE“ gesetzt, wenn der Greifer einen „MoveToWork“-Befehl erhält oder per „cmd\_b\_ResetDirectionFlag“ (s. 8.4) manuell ein Reset durchgeführt wird.

#### 8.19 Bit 14 des StatusWord „b\_ControlWord\_200“ (BOOL)

Dieser Richtungsmerker wird „TRUE“, wenn der Greifer einen „MoveToWork“-Befehl erhalten hat. Der Greifer kann in diesem Zustand keinen weiteren „MoveToWork“-Befehl ausführen. Der Merker wird wieder auf „FALSE“ gesetzt, wenn der Greifer einen „MoveToBase“-Befehl erhält oder per „cmd\_b\_ResetDirectionFlag“ (s. 8.4) manuell ein Reset durchgeführt wird.

#### 8.20 Bit 15 des StatusWord „b\_Error“ (BOOL) und „n\_Diagnose“ (WORD)

Wenn der Diagnosewert des Greifers nicht 0 ist, wird dieses Bit gesetzt. Der Fehlercode wird in dem Datenwort „n\_Diagnose“ ausgegeben. Die Beschreibungen zu den Fehlercodes können aus der Bedienungsanleitung entnommen werden.

#### 8.21 n\_ActualPosition (WORD)

In diesem Datenwort wird die aktuelle Position der Greiferbacken ausgegeben.