<Operationmanual_Advanced Gripping_S7_V2(EN).docx>
topic: <FB Advanced Gripping>

version: <2.0>

# *History*

| Author | Reason for change/changes made | Release | Date |
|---|---|---|---|
| Armbruster | Creation | 1.0 | |
| Nock | Selection of Homing-DeviceMode<br>Compatibility with GEP2000IL-03-B<br>Optimization of drive command routines<br>Optimization of the MotorON routine<br>Bugfix: "DataTransferError" | 2.0 | 01.08.2021 |

# Content

1 Foreword ..... 5
2 Integrating the library ..... 5
3 Declare data types ..... 5
4 Declare global structures in the OB ..... 8
5 Linking process data with peripherals ..... 8
6 Insert function block ..... 9
7 Using the function block ..... 11
8 Functions of the function block ..... 13
8.1 Resetting the step sequence „Inp_cmd_b_StepReset“ (BOOL) ..... 13
8.2 Switching on motor „Inp_cmd_b_MotorON“ (BOOL) ..... 13
8.3 Switching off motor „Inp_cmd_b_MotorOFF“ (BOOL) ..... 13
8.4 Referencing the gripper „Inp_cmd_b_StartHoming“ (BOOL) ..... 13
8.5 Transfering data with handshake „Inp_cmd_b_DataTransfer“ (BOOL) ..... 13
8.6 Saving workpiece recipes „Inp_cmd_b_WritePDU“ (BOOL) ..... 13
8.7 Resetting the direction flags „Inp_cmd_b_RDF“ (BOOL) ..... 14
8.8 Drive to BasePosition „Inp_cmd_b_MoveToBase“ (BOOL) ..... 14
8.9 Drive to WorkPosition „Inp_cmd_b_MoveToWork“ (BOOL) ..... 14
8.10 Jog in direction WorkPosition „Inp_cmd_b_JogToWork“ (BOOL) ..... 14
8.11 Jog in direction BasePosition „Inp_cmd_b_JogToBase“ (BOOL) ..... 14
8.12 Limitting of the motion time „Inp_t_MotionTimeout“ (TIME) and „Out_b_MotionError“ (BOOL) 14
8.13 Switchover to use of a GEP2000IL-03-B "Inp_b_GEP2000_03" (BOOL) ..... 14
8.14 Setting homing mode "Inp_i8_DeviceModeHoming" (BYTE) ..... 14
8.15 Data transfer is required „Out_b_DataTransferReq“ (BOOL) ..... 14
8.16 Error in the DataTransfer „Out_b_DataTransferError“ (BOOL) ..... 15
8.17 Function block is busy „Out_b_StepBusy“ (BOOL) ..... 15
8.18 Ready for commands „Out_b_StepDone“ (BOOL) ..... 15
8.19 Bit 0 of the StatusWord „Out_b_HomingPositionOK“ (BOOL) ..... 15
8.20 Bit 1 of the StatusWord „Out_b_MotorON“ (BOOL) ..... 15
8.21 Bit 2 of the StatusWord „Out_b_InMotion“ (BOOL) ..... 15
8.22 Bit 3 of the StatusWord „Out_b_MovementComplete“ (BOOL) ..... 15
8.23 Bit 4 of the StatusWord „Out_b_JogBaseActive“ (BOOL) ..... 15
8.24 Bit 5 of the StatusWord „Out_b_JogWorkActive“ (BOOL) ..... 15
8.25 Bit 6 of the StatusWord „Out_b_GripperPLCActive“ (BOOL) ..... 15
8.26 Bit 7 of the StatusWord „Out_b_ControllerError“ (BOOL) ..... 15
8.27 Bit 8 of the StatusWord „Out_b_BasePosition“ (BOOL) ..... 15
8.28 Bit 9 of the StatusWord „Out_b_TeachPosition“ (BOOL) ..... 16
8.29 Bit 10 of the StatusWord „Out_b_WorkPosition“ (BOOL) ..... 16

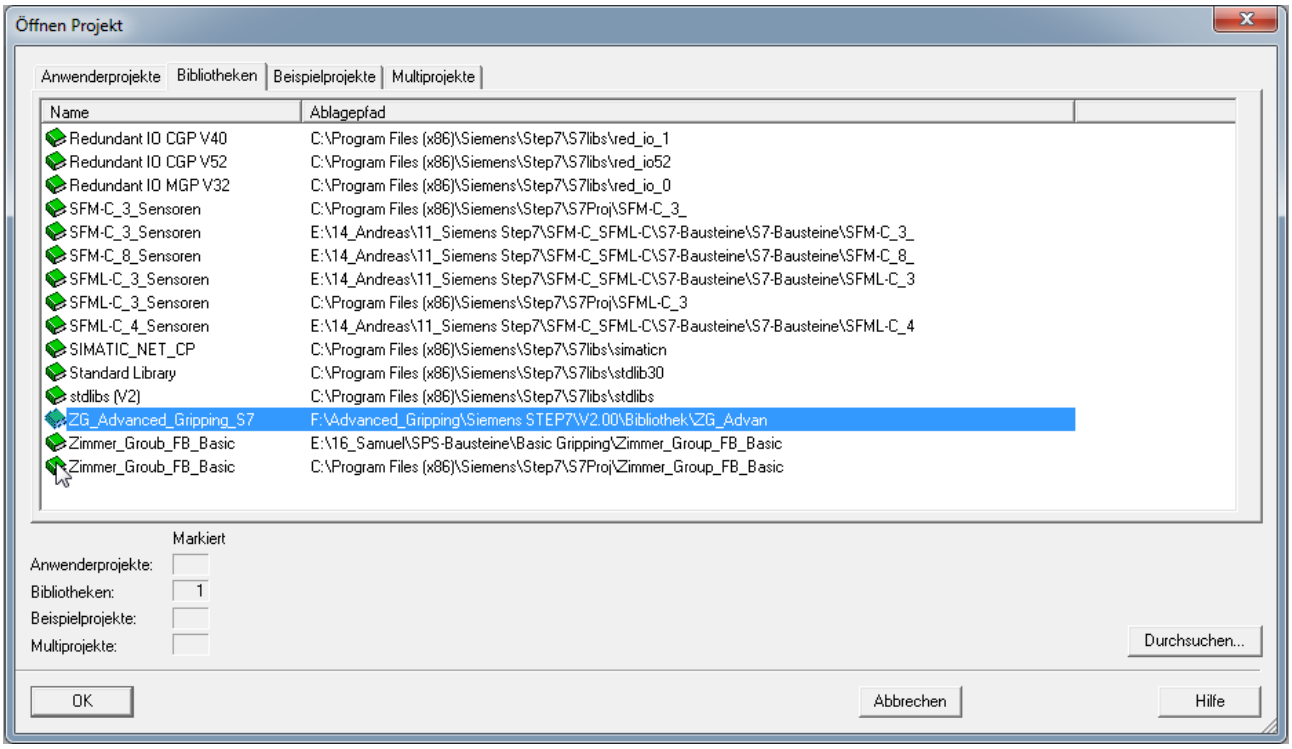
8.30 Bit 11 of the StatusWord „Out_b_UndefinedPosition“ (BOOL) ..... 16
8.31 Bit 12 of the StatusWord „Out_b_DataTransferOK“ (BOOL) ..... 16
8.32 Bit 13 of the StatusWord „Out_b_ControlWord_100“ (BOOL) ..... 16
8.33 Bit 14 of the StatusWord „Out_b_ControlWord_200“ (BOOL) ..... 16
8.34 Bit 15 of the StatusWord „Out_b_Error“ (BOOL) and „Out_i16_Diagnose“ (WORD) ..... 16
8.35 Actual position "Out_i16_ActualPosition" (WORD) ..... 16

# 1 Foreword

To use the example program, a correct hardware configuration must first be created. In this example a Siemens CPU314C-2 PN/DP with a Balluff IO Link master is used. After the hardware settings the function block can be implemented. To do this, go through the following steps.

# 2 Integrating the library

The function block is provided as a library. To open it, please select the "File" -> "Open..." menu. In the now visible window you can select and open the desired library.
Please select the global library " **ZG_Advanced_Gripping_S7**".



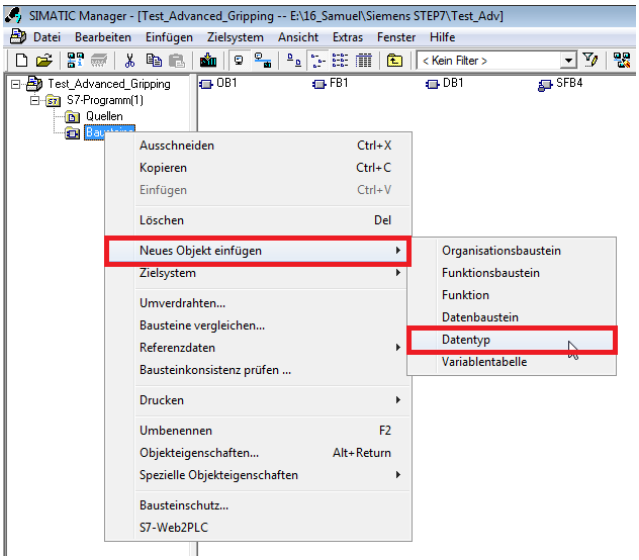

Now the Zimmer library is ready for use and can be used in your project.

# 3 Declare data types

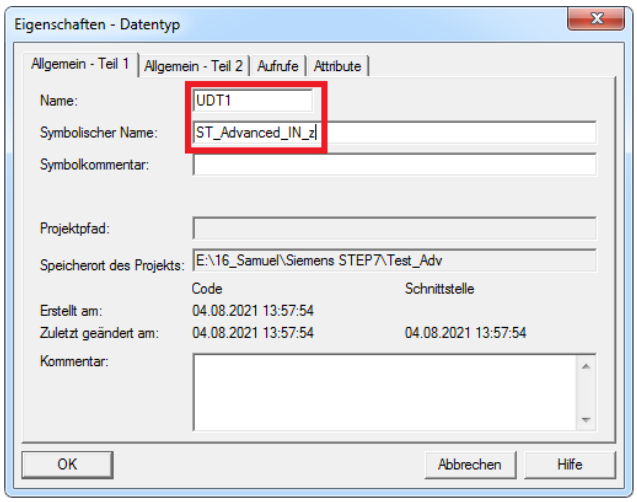
The input and output data of the gripper - or also called "process data" - are transferred to the function block via specific data types (structures or UDT). The following data types must be created for communication between the function block and the gripper:

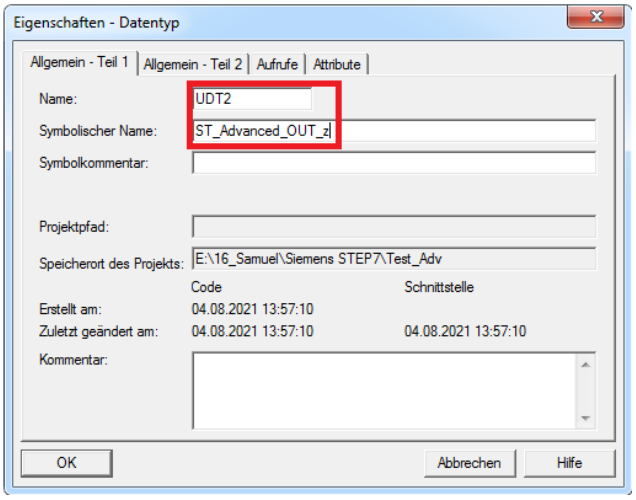
| Name | Symbolic name | Includes |
|---|---|---|
| UDT1 | ST_Advanced_IN_z | 6 bytes input data |
| UDT2 | ST_Advanced_OUT_z | 16 bytes output data |

A new data type can be inserted into the project by right-clicking on "Blocks" -> "Insert new object" -> "Data type".




The "Symbolic name" of the UDT is defined in the window that opens. Please enter the given name.



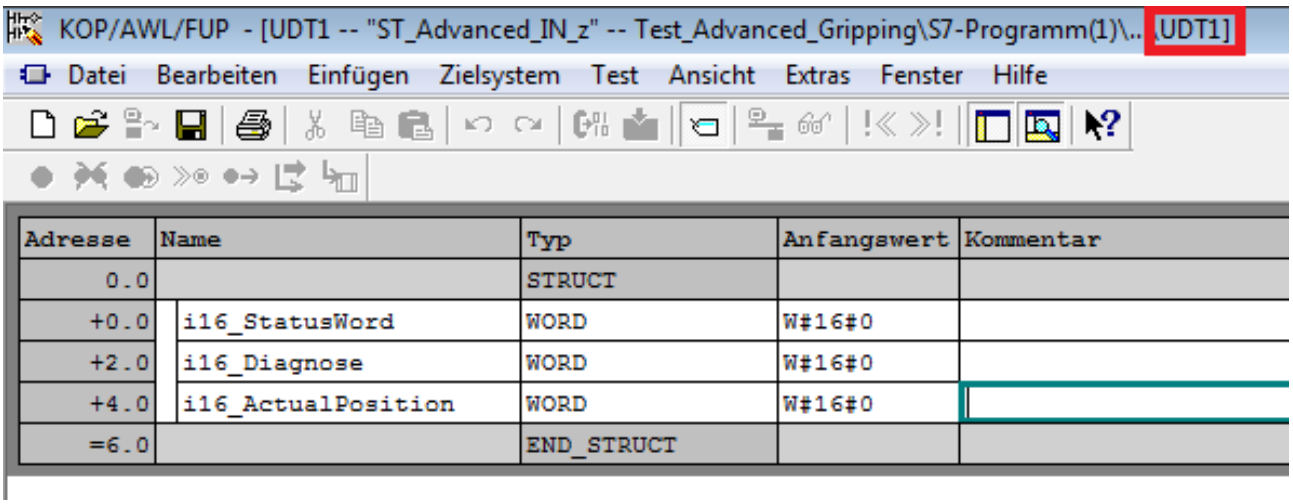

Please open the corresponding data type and define it as follows. The sequence should be followed.

UDT1:

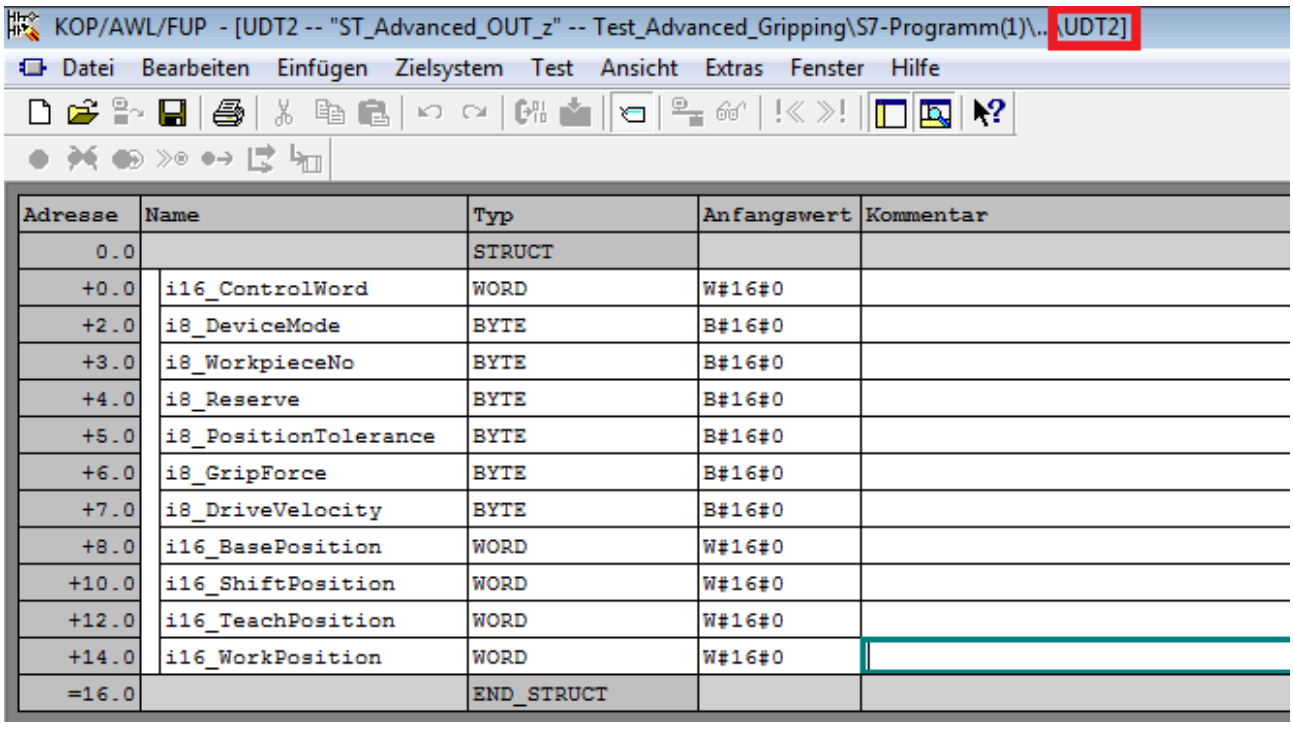
| **Name** | **Type** | **Initial value** |
|---|---|---|
| i16_StatusWord | WORD | W#16#0 |
| i16_Diagnose | WORD | W#16#0 |
| i16_ActualPosition | WORD | W#16#0 |

UDT2:

| Name | Type | Initial value |
|---|---|---|
| i16_ControlWord | WORD | W#16#0 |
| i8_DeviceMode | BYTE | B#16#0 |
| i8_WorkpieceNo | BYTE | B#16#0 |
| i8_Reserve | BYTE | B#16#0 |
| i8_PositionTolerance | BYTE | B#16#0 |
| i8_GripForce | BYTE | B#16#0 |
| i8_DriveVelocity | BYTE | B#16#0 |
| i16_BasePosition | WORD | W#16#0 |
| i16_ShiftPosition | WORD | W#16#0 |
| i16_TeachPosition | WORD | W#16#0 |
| i16_WorkPosition | WORD | W#16#0 |




| Adresse | Name | Typ | Anfangswert | Kommentar |
|---|---|---|---|---|
| 0.0 | | STRUCT | | |
| +0.0 | i16_StatusWord | WORD | W#16#0 | |
| +2.0 | i16_Diagnose | WORD | W#16#0 | |
| +4.0 | i16_ActualPosition | WORD | W#16#0 | |
| =6.0 | | END_STRUCT | | |




| Adresse | Name | Typ | Anfangswert | Kommentar |
|---|---|---|---|---|
| 0.0 | | STRUCT | | |
| +0.0 | i16_ControlWord | WORD | W#16#0 | |
| +2.0 | i8_DeviceMode | BYTE | B#16#0 | |
| +3.0 | i8_WorkpieceNo | BYTE | B#16#0 | |
| +4.0 | i8_Reserve | BYTE | B#16#0 | |
| +5.0 | i8_PositionTolerance | BYTE | B#16#0 | |
| +6.0 | i8_GripForce | BYTE | B#16#0 | |
| +7.0 | i8_DriveVelocity | BYTE | B#16#0 | |
| +8.0 | i16_BasePosition | WORD | W#16#0 | |
| +10.0 | i16_ShiftPosition | WORD | W#16#0 | |
| +12.0 | i16_TeachPosition | WORD | W#16#0 | |
| +14.0 | i16_WorkPosition | WORD | W#16#0 | |
| =16.0 | | END_STRUCT | | |

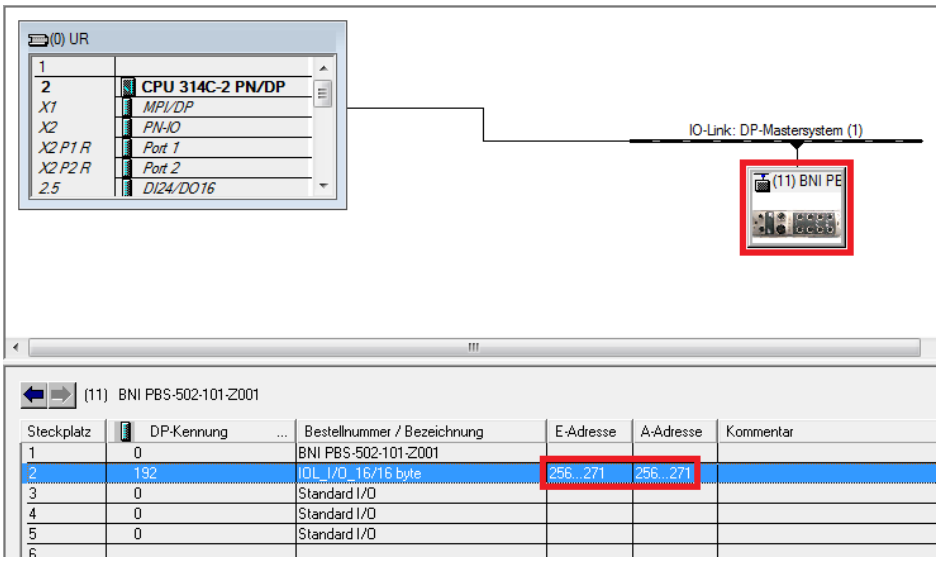
## 4 Declare global structures in the OB

In the used operation block global variables must be created from the created structures to link them to the block. Please create the following variables in the OB in the declaration table:

| | Name | Datentyp | Adresse | Kommentar |
|---|---|---|---|---|
| | 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_st_Gripper1_IN | ST_Advanced_IN_z | 20.0 | |
| | g_st_Gripper1_OUT | ST_Advanced_OUT_z | 26.0 | |

## 5 Linking process data with peripherals

In the hardware configuration, address ranges for the input and output data have been assigned to the IO-Link port to which the gripper is connected. In this example, the range starts at peripheral address 256.

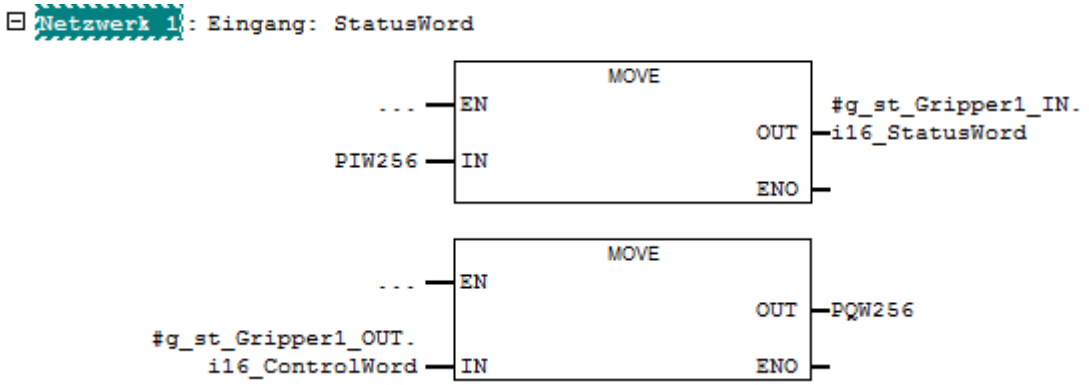



In the operation block, the peripheral addresses must be linked with the global structures created. In our example "MOVE" blocks were used for the assignments. The input data should be linked in the OB before the FB and the output data after the FB. When assigning the addresses, the sequence of the global structure must be observed.

In this example the assignments - starting with address 256 - look as follows:

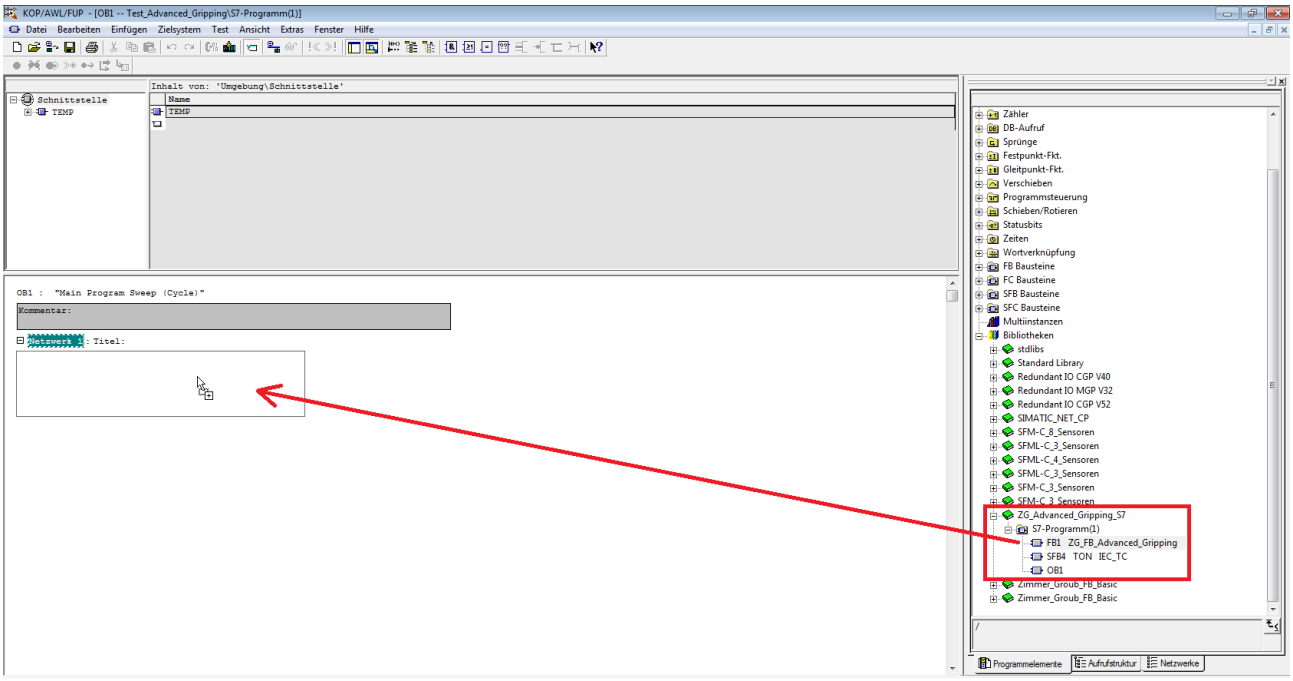
| **Assignment of …** | **Assignment to …** |
|---|---|
| PIW256 | g_st_Gripper1_IN.i16_StatusWord |
| PIW258 | g_st_Gripper1_IN.i16_Diagnose |
| PIW260 | g_st_Gripper1_IN.i16_ActualPosition |
| g_st_Gripper1_OUT.i16_ControlWord | PQW256 |
| g_st_Gripper_1OUT.i8_DeviceMode | PQB258 |
| g_st_Gripper_1OUT.i8_WorkpieceNo | PQB259 |
| g_st_Gripper_1OUT.i8_Reserve | PQB260 |
| g_st_Gripper_1OUT.i8_PositionTolerance | PQB261 |
| g_st_Gripper_1OUT.i8_GripForce | PQB262 |
| g_st_Gripper_1OUT.i8_DriveVelocity | PQB263 |
| g_st_Gripper_1OUT.i16_BasePosition | PQW264 |

| | |
|---|---|
| g_st_Gripper_1OUT.i16_ShiftPosition | PQW266 |
| g_st_Gripper_1OUT.i16_TeachPosition | PQW268 |
| g_st_Gripper_1OUT.i16_WorkPosition | PQW270 |

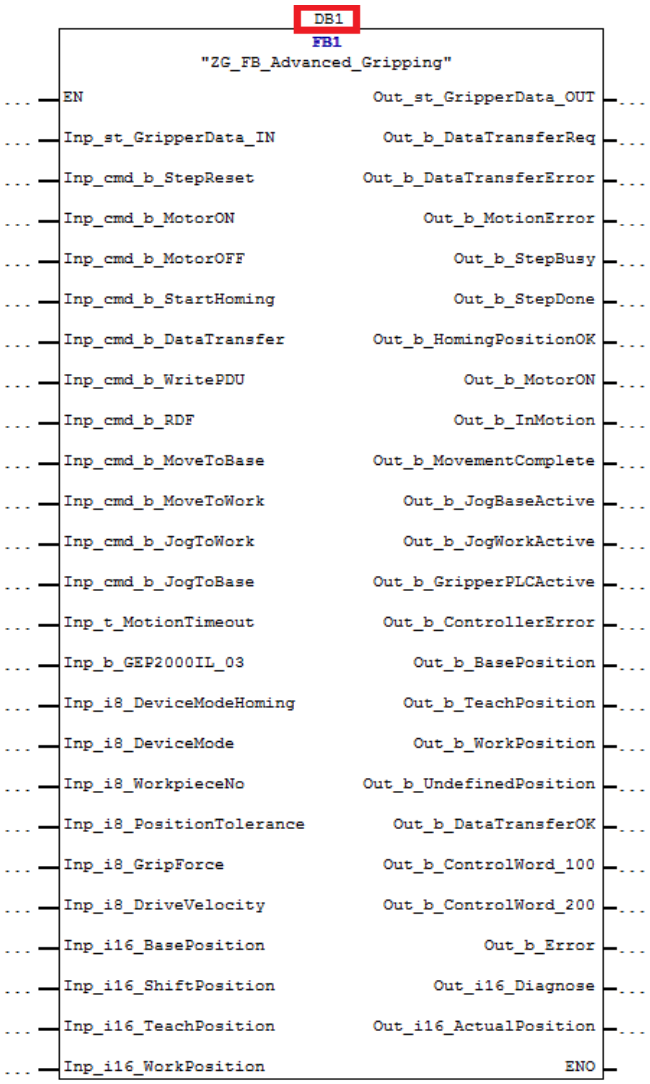



## 6 Insert function block

To be able to use the "Advanced-Gripping" block, it must be called in the desired operation block. The "ZG_FB_Advanced_Gripping" block can be copied from the already integrated library to a free network u-sing drag & drop. The standard block "SFB4 (TON)" is also present in this library, since it is required in the "Advanced-Gripping" block.



Since this is a function block, an instance data block is required. In this example, a new data block DB1 was generated.




Nomenclature:

| **Prefix** | **Meaning** |
|---|---|
| Inp | Input variable |
| Out | Output variable |
| cmd | Command input |
| b | Binary signal (BOOL) |
| i8 | Variable in byte size (BYTE) |
| i16 | Variable in word size (WORD) |
| st | Data structure (STRUCT) |
| t | Time (TIME) |
| g | Global variable |

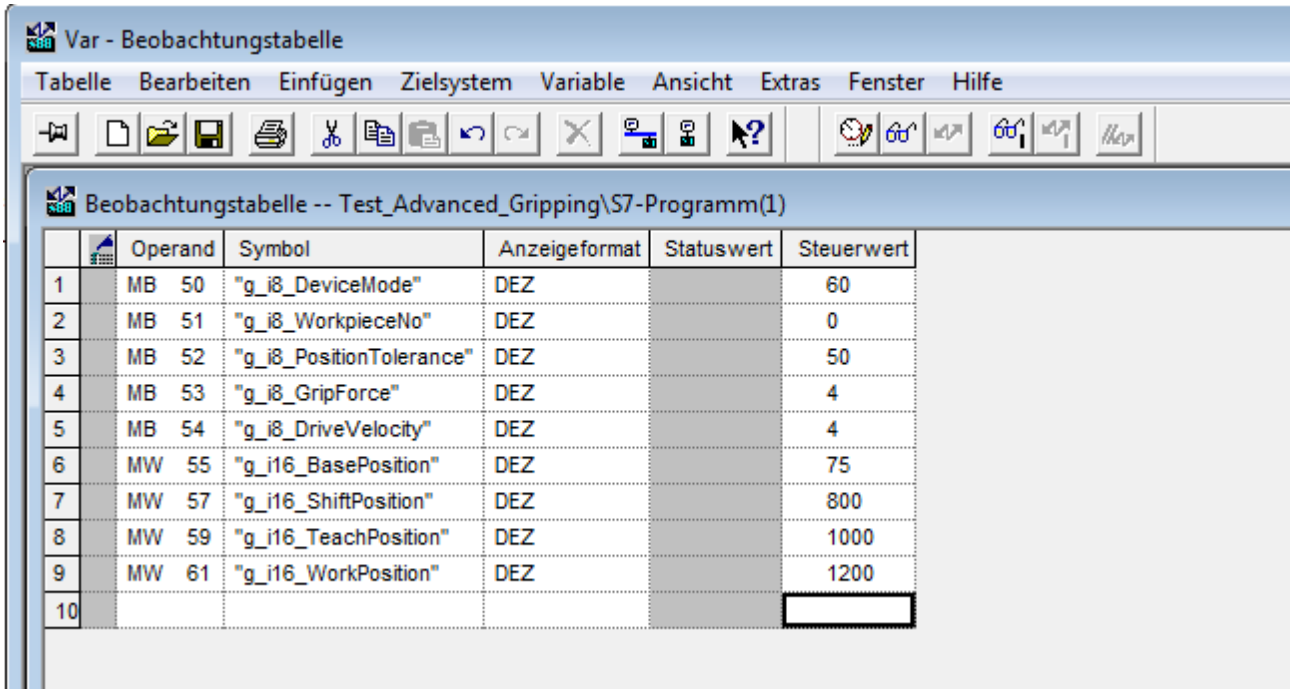
# 7 Using the function block

To enable the function block to access the address ranges of the IO-Link data, the variables "Inp_st_GripperData_IN" and "Out_st_GripperData_OUT" must be linked to the variables created in point 4. The gripper can be parameterized by writing to the input variables on the module.

To move the gripper, the position and travel data must be transferred. The values listed in the following table can be used as standard values. Other values may prove to be more suitable. Please refer to the installation and operating instructions. In this example, the inputs have been wired with flags and controlled with a variable table. When not connected, the variables are pre-initialized with the default values.

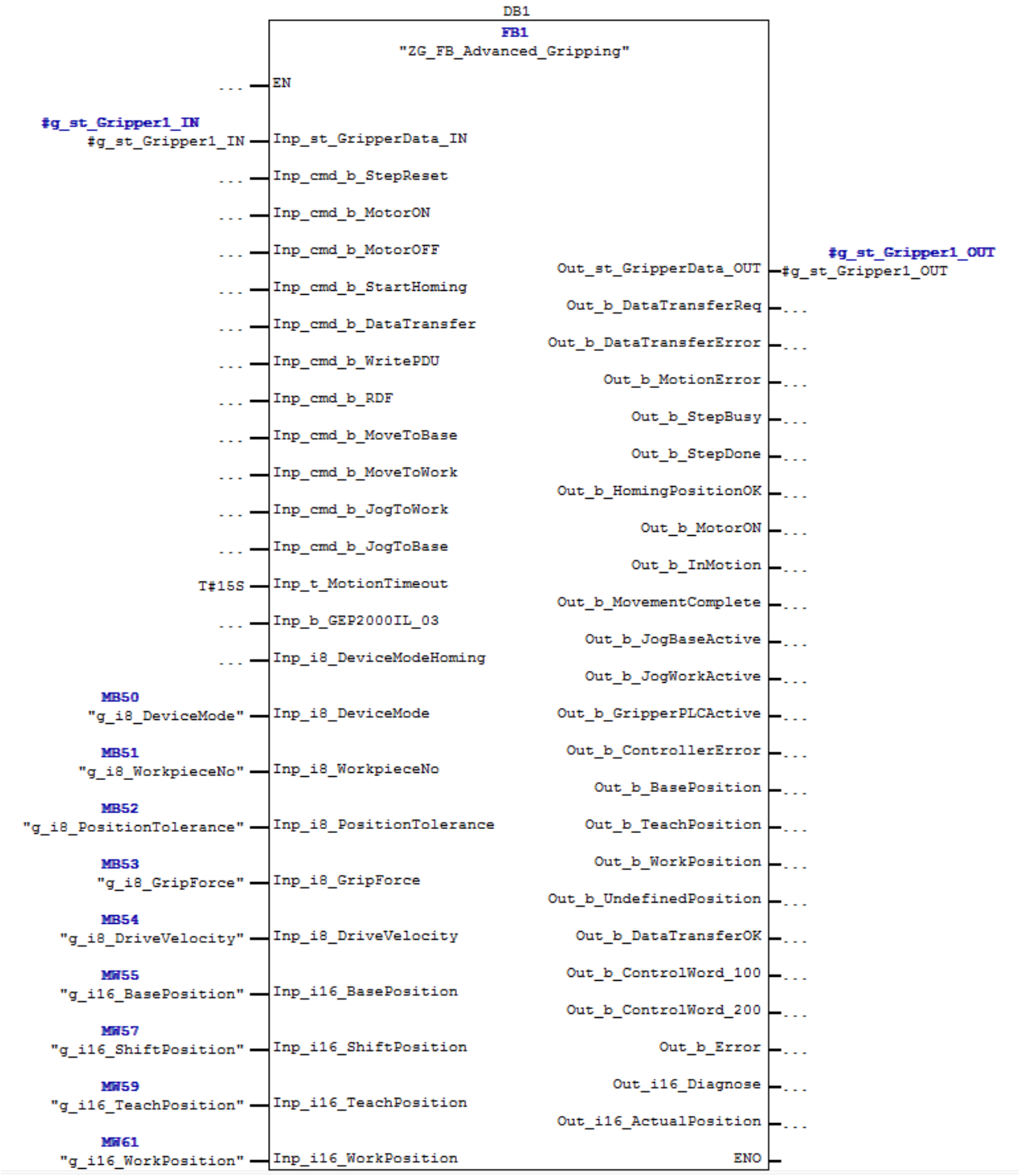
| **Variable** | **Value** |
|---|---|
| Inp_t_MotionTimeout | T#15s |
| Inp_i8_DeviceMode | 60 |
| Inp_i8_WorkpieceNo | 0 |
| Inp_i8_PositionTolerance | 50 |
| Inp_i8_GripForce | 4 |
| Inp_i8_DriveVelocity | 4 |
| Inp_i16_BasePosition | 75 |
| Inp_i16_ShiftPosition | 800 |
| Inp_i16_TeachPositon | 1000 |
| Inp_i16_WorkPosition | 1200 |

The "Inp_i8_DeviceMode" variable corresponds to the travel profile of the gripper. These travel profiles can be found in the installation and operating instructions for the gripper. In this example, DeviceMode 60 was selected, which corresponds to the "Force profile outside gripping" travel profile.




| | Operand | Symbol | Anzeigeformat | Statuswert | Steuerwert |
|---|---|---|---|---|---|
| 1 | MB 50 | "g_i8_DeviceMode" | DEZ | | 60 |
| 2 | MB 51 | "g_i8_WorkpieceNo" | DEZ | | 0 |
| 3 | MB 52 | "g_i8_PositionTolerance" | DEZ | | 50 |
| 4 | MB 53 | "g_i8_GripForce" | DEZ | | 4 |
| 5 | MB 54 | "g_i8_DriveVelocity" | DEZ | | 4 |
| 6 | MW 55 | "g_i16_BasePosition" | DEZ | | 75 |
| 7 | MW 57 | "g_i16_ShiftPosition" | DEZ | | 800 |
| 8 | MW 59 | "g_i16_TeachPosition" | DEZ | | 1000 |
| 9 | MW 61 | "g_i16_WorkPosition" | DEZ | | 1200 |
| 10 | | | | | |

The finished module should now correspond to the following figure:




Finally, you need to debug the settings and load them into the device.

# 8 Functions of the function block

Depending on the input wiring of the function block, the corresponding functions are executed. Further information can also be found in the comments in the block header.

## 8.1 Resetting the step sequence „Inp_cmd_b_StepReset“ (BOOL)

The input variable "Inp_cmd_b_StepReset" resets the step chain within this block. This happens regardless of which step the block is currently in. If the block issues the error "Out_b_DataTransferError" or "Out_b_MotionError", it can only be acknowledged by this input.

## 8.2 Switching on motor „Inp_cmd_b_MotorON“ (BOOL)

The gripper can only move when the motor is switched on. The gripper represents its states in the "Status-Word". If the gripper is switched off, the "Out_b_MotorON" signal is "FALSE". If the signal edge is positive, the motor is switched on and the "Out_b_MotorON" signal is set to "TRUE". The function block converts this signal into "DeviceMode" 3 and sends it to the gripper. In order for the gripper to use this "DeviceMode", a data transmission with handshake must be carried out. After successful transfer of the data, the "b_MotorON" bit changes from "FALSE" to "TRUE". The motor is now switched on. The function block does this automatically.

## 8.3 Switching off motor „Inp_cmd_b_MotorOFF“ (BOOL)

To switch off the gripper motor, a positive signal edge must be set at the "Inp_cmd_b_MotorOFF" input. The "DeviceMode" is automatically set to the value 5 and a data transfer is performed. The motor can be switched off at any time (except during a reference run) and is independent of which command the gripper has previously received.

## 8.4 Referencing the gripper „Inp_cmd_b_StartHoming“ (BOOL)

The gripper must be referenced so that it can always output the correct position. The "Out_b_HomingPositionOK" signal indicates the current status of the referencing. If this signal is set to "FALSE", the gripper does not know in which position the gripper jaws are and process-safe operation would therefore not be guaranteed. The gripper can be referenced again with the "Inp_cmd_b_StartHoming" signal. There are different referencing modes that can be set with the "Inp_i8_DeviceModeHoming" input. With a positive signal edge, the gripper starts a new homing run with the set referencing mode. A reference run must not take place in the gripped state. Make sure in advance that the gripper is free to move.

## 8.5 Transfering data with handshake „Inp_cmd_b_DataTransfer“ (BOOL)

After each change of a process parameter (except "ControlWord") or at a cold start of the gripper, the parameters must be taken over with a data transfer. If the "Out_b_DataTransferReq" output variable is "TRUE", the gripper is not yet operating with the currently set parameters. In this case, the process parameters must be transferred with a positive signal edge at the "Inp_cmd_b_DataTransfer" input. The variable "Out_b_DataTransferReq" then changes to "FALSE". Thereby the "ControlWord" is set to value 1 and waits for bit 12 of the "Status-Word". Bit 12 becomes "TRUE" as soon as the data transfer is completed. Then the "ControlWord" is set to 0 again and waited until bit 12 becomes "FALSE". This procedure is a handshake and should be used for error-free data transfer.

## 8.6 Saving workpiece recipes „Inp_cmd_b_WritePDU“ (BOOL)

With a positive signal edge, the currently set process parameters at the function block input are stored in the currently set "WorkpieceNo". The "ControlWord" is set to value 2 and bit 12 of the "StatusWord" is waited for. This procedure can take up to 30 seconds. The parameters are stored in the internal recipe locations and can be reloaded by specifying the "WorkpieceNo". Up to 32 recipes can be stored in the gripper.

### 8.7 Resetting the direction flags „Inp_cmd_b_RDF“ (BOOL)

If a gripper is moved in the direction of "WorkPosition", for example, bit 14 of the "Status word" is set in the gripper. This signal remains until a movement in the other direction or a cold start of the gripper. If a gripper is to be moved several times in succession in the same direction, e.g. by changing positions, then this bit must first be reset. This can be done by a positive signal edge at the input "Inp_cmd_b_RDF". Thereby the "ControlWord" is set to the value 4 and waits until bit 13 and bit 14 of the "StatusWord" change to "FALSE". After this, a new movement in the same direction can take place. From function block version V1.21 onwards, this procedure is carried out automatically before the gripper is moved, if necessary.

### 8.8 Drive to BasePosition „Inp_cmd_b_MoveToBase“ (BOOL)

With a positive signal edge, the gripper jaws move with the set travel profile in the direction of the set "BasePosition". The "ControlWord" is set to the value 256.

### 8.9 Drive to WorkPosition „Inp_cmd_b_MoveToWork“ (BOOL)

With a positive signal edge, the gripper jaws move with the set travel profile in the direction of the set "WorkPosition". The "ControlWord" is set to the value 512.

### 8.10 Jog in direction WorkPosition „Inp_cmd_b_JogToWork“ (BOOL)

The gripper moves in inching mode. No software limit switches are active in this mode. When this input is set to "TRUE", the "DeviceMode" is automatically set to the value 11, a handshake is performed and the corresponding bit of the "ControlWord" is set. The gripper jaws move at low speed in the direction of the "WorkPosition". When the input is set to "FALSE", the gripper stops again.

### 8.11 Jog in direction BasePosition „Inp_cmd_b_JogToBase“ (BOOL)

The gripper moves in inching mode. No software limit switches are active in this mode. When this input is set to "TRUE", the "DeviceMode" is automatically set to the value 11, a handshake is performed and the corresponding bit of the "ControlWord" is set. The gripper jaws move at low speed in the direction of the "BasePosition". When the input is set to "FALSE", the gripper stops again.

### 8.12 Limitting of the motion time „Inp_t_MotionTimeout“ (TIME) and „Out_b_MotionError“ (BOOL)

The "Inp_t_MotionTimeout" time can be used to define the maximum time the gripper may take to move until it reaches its target position. This depends on the parameterization of the gripper and must be adapted project-specifically. If the gripper does not reach its target position within the set time, the "MotionError" error is activated. The "Out_b_MotionError" output is set to "TRUE".

### 8.13 Switchover to use of a GEP2000IL-03-B "Inp_b_GEP2000_03" (BOOL)

In addition to the GEH6000IL, this function block is also compatible with the GEP2000IL-03 gripper series. Setting the input to "TRUE" signals to the function block that it is operated with a GEP2000IL-03-B. The functions that a GEP2000IL-03-B does not have compared to a GEH6000IL are thus deactivated.

### 8.14 Setting homing mode "Inp_i8_DeviceModeHoming" (BYTE)

The desired referencing mode (e.g. "14" for DeviceMode 14) can be set at this input. The modes can be taken from the installation and operating instructions. When not connected, DeviceMode 10 is set as default. It is not allowed to connect the input with the value "0". Homing is started with the "Inp_b_cmd_StartHoming" input (see 8.4).

### 8.15 Data transfer is required „Out_b_DataTransferReq“ (BOOL)

The variable "Out_b_DataTransferReq" is automatically activated if at least one process parameter was changed at the inputs. As long as this variable is active, the gripper has not yet transferred the changed values. For data transfer, a positive signal edge must be set at the "Inp_cmd_b_DataTransfer" input variable.

The "Out_b_DataTransferReq" variable then changes to "FALSE" and the gripper uses the currently set parameters.

### 8.16 Error in the DataTransfer „Out_b_DataTransferError“ (BOOL)

The "Out_b_DataTransferError" output is set to "TRUE" if the data transfer could not be carried out successfully and the feedback of the gripper was not sent within one second. This can occur, among other things, if the set process parameters are not plausible. The error code can be taken from the variable "Out_i16_Diagnose". The error codes are described in more detail in the installation and operating instructions. This error can be acknowledged by setting the "Inp_cmd_b_StepReset" input.

### 8.17 Function block is busy „Out_b_StepBusy“ (BOOL)

If the block is processing a command and is in a step, this output is active and signals that it is blocked for further commands.

### 8.18 Ready for commands „Out_b_StepDone“ (BOOL)

If the block is in the initial step and ready for commands, this output is "TRUE". Querying this bit before a command for programming step chains is recommended.

### 8.19 Bit 0 of the StatusWord „Out_b_HomingPositionOK“ (BOOL)

The gripper has an internal distance measuring system which does not have to be referenced under normal circumstances. As long as the gripper has a valid referencing, this signal is active. As soon as this signal is set to "FALSE", the gripper must be referenced again (see 8.4).

### 8.20 Bit 1 of the StatusWord „Out_b_MotorON“ (BOOL)

As long as this signal is "FALSE", the gripper cannot be moved. The motor of the gripper must first be switched on (s 8.2).

### 8.21 Bit 2 of the StatusWord „Out_b_InMotion“ (BOOL)

This signal is active as long as the gripper jaws are moving.

### 8.22 Bit 3 of the StatusWord „Out_b_MovementComplete“ (BOOL)

This signal indicates that a movement has been completed and the gripper is at a standstill.

### 8.23 Bit 4 of the StatusWord „Out_b_JogBaseActive“ (BOOL)

This signal is active as long as the gripper is controlled in jog mode and moved in the direction of the "Base position".

### 8.24 Bit 5 of the StatusWord „Out_b_JogWorkActive“ (BOOL)

This signal is active as long as the gripper is controlled in jog mode and moved in the direction of the "WorkPosition".

### 8.25 Bit 6 of the StatusWord „Out_b_GripperPLCActive“ (BOOL)

This signal indicates the operational readiness of the control in the gripper. In the event of a cold start or restart after a power failure, the gripper can only receive data again when this signal is "TRUE".

### 8.26 Bit 7 of the StatusWord „Out_b_ControllerError“ (BOOL)

Error in the internal controller.

### 8.27 Bit 8 of the StatusWord „Out_b_BasePosition“ (BOOL)

As soon as the gripper has reached its set "BasePosition" and is at standstill, this signal is activated. The size of the range is defined by the "PositionTolerance".

### 8.28 Bit 9 of the StatusWord „Out_b_TeachPosition“ (BOOL)

As soon as the gripper has reached its set "TeachPosition" and is at standstill, this signal is activated. The size of the range is defined by the "PositionTolerance".

### 8.29 Bit 10 of the StatusWord „Out_b_WorkPosition“ (BOOL)

As soon as the gripper has reached its set "WorkPosition" and is at standstill, this signal is activated. The size of the range is defined by the "PositionTolerance".

### 8.30 Bit 11 of the StatusWord „Out_b_UndefinedPosition“ (BOOL)

If the gripper is stationary and is neither at "BasePosition" nor at "TeachPosition" or "WorkPosition", this signal is "TRUE".

### 8.31 Bit 12 of the StatusWord „Out_b_DataTransferOK“ (BOOL)

With this bit the gripper gives the feedback that a data transmission has been successfully executed. Therefore it is used in a handshake procedure.

### 8.32 Bit 13 of the StatusWord „Out_b_ControlWord_100“ (BOOL)

This direction flag becomes active when the gripper has received a "MoveToBase" command. The gripper cannot execute another "MoveToBase" command in this state. The flag is set to "FALSE" again when the gripper receives a "MoveToWork" command or a reset is performed manually via "Inp_cmd_b_RDF" (see 8.7).

### 8.33 Bit 14 of the StatusWord „Out_b_ControlWord_200“ (BOOL)

This direction flag becomes active when the gripper has received a "MoveToWork" command. The gripper cannot execute another "MoveToWork" command in this state. The flag is set to "FALSE" again when the gripper receives a "MoveToBase" command or a reset is performed manually via "Inp_cmd_b_RDF" (see 8.7).

### 8.34 Bit 15 of the StatusWord „Out_b_Error“ (BOOL) and „Out_i16_Diagnose“ (WORD)

If the diagnostic value of the gripper is not 0, this bit is set. The error code is output in the data word "Out_i16_Diagnose". The descriptions of the error codes can be taken from the assembly and operating instructions.

### 8.35 Actual position "Out_i16_ActualPosition" (WORD)

In this data word the actual position of the gripper jaws is output in 0.01mm.