

Motion Terminal VTEM

Festo_VTEM_DevCon Versionen __.__.37

Bibliothek zur Steuerung (Device Control) des
Motion Terminals VTEM von Festo
ab Firmware-Version 4.22.x



Beschreibung

TIA Portal
ab V19

Titel..... Festo_VTEM_DevCon
Version des Dokuments1.07
Originalde
Autor Festo

Letztes Speicherdatum 2026/05/06

Versions-Info und Historie der Bibliothek

=====

Festo SE & Co. KG, Esslingen
Copyright © 2026, all rights reserved

Siemens TIA Portal function block library for VTEM
Version 3.7

Designed for use with
- Festo Motion Terminal VTEM
- Firmware version 4.22.x

Created
- with TIA V19
- 2026-02

=====

H I S T O R Y

=====

Version 3.7

object	version	change
eRespondingReadMalfunctionList	1.00	- no changes
eResponseTeachInRun	1.06	- no changes
eValveMode	1.02	- no changes
eValveState	1.00	- no changes
eResponseToValveModeSet	1.01	- no changes
eValveType_MA_01_02	1.00	- no changes
eMA100OperatingMedium	1.00	- no changes
eSavingResponse	1.01	- no changes
eTransferResponse	1.03	- no changes
stMalfunction	1.01	- no changes
stTimeStamp	1.00	- no changes
stTransferPackage	1.00	- no changes
FB_Read_Bits_From_Byte	1.01	- no changes
FB_Read_Bits_From_Word	1.02	- no changes
FB_Write_Value_To_Word	1.01	- no changes
FB_AppOptionGenerator_MA_02	1.01	- optimized valve access enabled
FB_TeachInRun	1.02	- no changes
FB_ValveControl	2.04	- no changes
FB_StateInterpreter_MA_01	1.02	- optimized valve access enabled
FB_StateInterpreter_MA_02	1.03	- optimized valve access enabled
FB_StateInterpreter_MA_03	1.02	- optimized valve access enabled
FB_StateInterpreter_MA_04	1.02	- optimized valve access enabled
FB_StateInterpreter_MA_05	1.03	- optimized valve access enabled
FB_StateInterpreter_MA_06	1.02	- optimized valve access enabled
FB_StateInterpreter_MA_07	1.02	- optimized valve access enabled
FB_StateInterpreter_MA_08	1.02	- optimized valve access enabled
FB_StateInterpreter_MA_10	1.01	- optimized valve access enabled
FB_StateInterpreter_MA_11	1.03	- optimized valve access enabled

FB_StateInterpreter_MA_12	1.02	- optimized valve access enabled
FB_StateInterpreter_MA_33	1.01	- optimized valve access enabled
FB_StateInterpreter_TeachInRun	1.02	- optimized valve access enabled
FB_Download_Multiple_Parameters	2.02	- no changes
FB_Download_Single_Parameter	2.02	- no changes
FB_ReadMalfunctionList	1.02	- optimized valve access enabled
FB_Save_Configuration_Persistent	2.03	- no changes
FB_Upload_Multiple_Parameters	2.02	- optimized valve access enabled
FB_Upload_Single_Parameter	2.02	- optimized valve access enabled
FB_Read_config_AI_Module	2.01	- optimized valve access enabled
FB_Read_config_DI_Module	2.01	- optimized valve access enabled
FB_Read_config_MA_01	2.01	- optimized valve access enabled
FB_Read_config_MA_02	2.01	- optimized valve access enabled
FB_Read_config_MA_03	2.01	- optimized valve access enabled
FB_Read_config_MA_04	2.01	- optimized valve access enabled
FB_Read_config_MA_05	2.01	- optimized valve access enabled
FB_Read_config_MA_06	2.01	- optimized valve access enabled
FB_Read_config_MA_07	2.01	- optimized valve access enabled
FB_Read_config_MA_08	2.01	- optimized valve access enabled
FB_Read_config_MA_10	2.01	- optimized valve access enabled
FB_Read_config_MA_11	2.01	- optimized valve access enabled
FB_Read_config_MA_12	2.01	- optimized valve access enabled
FB_Read_config_MA_33	2.01	- optimized valve access enabled
FB_Write_config_AI_Module	2.03	- no changes
FB_Write_config_DI_Module	2.03	- optimized valve access enabled
FB_Write_config_MA_01	2.01	- optimized valve access enabled
FB_Write_config_MA_02	2.02	- optimized valve access enabled
FB_Write_config_MA_03	2.02	- optimized valve access enabled
FB_Write_config_MA_04	2.01	- no changes
FB_Write_config_MA_05	2.01	- optimized valve access enabled
FB_Write_config_MA_06	2.01	- optimized valve access enabled
FB_Write_config_MA_07	2.01	- optimized valve access enabled
FB_Write_config_MA_08	2.02	- optimized valve access enabled
FB_Write_config_MA_10	2.02	- optimized valve access enabled
FB_Write_config_MA_11	2.01	- optimized valve access enabled
FB_Write_config_MA_12	2.01	- optimized valve access enabled
FB_Write_config_MA_33	2.02	- optimized valve access enabled

Version 3.6

object	version	change
eRespondingReadMalfunctionList	1.00	- no changes
eResponseTeachInRun	1.06	- no changes
eValveMode	1.02	- no changes
eValveState	1.00	- no changes

eResponseToValveModeSet	1.01	- no changes
eValveType_MA_01_02	1.00	- no changes
eMA100OperatingMedium	1.00	- no changes
eSavingResponse	1.01	- no changes
eTransferResponse	1.03	- no changes
stMalfunction	1.01	- no changes
stTimeStamp	1.00	- no changes
stTransferPackage	1.00	- no changes
FB_Read_Bits_From_Byte	1.01	- removed AT construct - optimized valve access enabled
FB_Read_Bits_From_Word	1.02	- removed AT construct - optimized valve access enabled
FB_Write_Value_To_Word	1.01	- removed AT construct - function block numbering set to Automatic - optimized valve access enabled
FB_AppOptionGenerator_MA_02	1.00	- no changes
FB_TeachInRun	1.02	- optimized valve access enabled
FB_ValveControl	2.04	- optimized valve access enabled
FB_StateInterpreter_MA_01	1.01	- no changes
FB_StateInterpreter_MA_02	1.02	- no changes
FB_StateInterpreter_MA_03	1.01	- no changes
FB_StateInterpreter_MA_04	1.01	- no changes
FB_StateInterpreter_MA_05	1.02	- no changes
FB_StateInterpreter_MA_06	1.01	- no changes
FB_StateInterpreter_MA_07	1.01	- no changes
FB_StateInterpreter_MA_08	1.01	- no changes
FB_StateInterpreter_MA_10	1.00	- no changes
FB_StateInterpreter_MA_11	1.02	- no changes
FB_StateInterpreter_MA_12	1.01	- no changes
FB_StateInterpreter_MA_33	1.00	- no changes
FB_StateInterpreter_TeachInRun	1.01	- no changes
FB_Download_Multiple_Parameters	2.02	- function block numbering set to Automatic - optimized valve access enabled
FB_Download_Single_Parameter	2.02	- optimized valve access enabled
FB_ReadMalfunctionList	1.01	- no changes
FB_Save_Configuration_Persistent	2.03	- optimized valve access enabled
FB_Upload_Multiple_Parameters	2.01	- no changes
FB_Upload_Single_Parameter	2.01	- no changes
FB_Read_config_AI_Module	2.00	- no changes
FB_Read_config_DI_Module	2.00	- no changes
FB_Read_config_MA_01	2.00	- no changes
FB_Read_config_MA_02	2.00	- no changes
FB_Read_config_MA_03	2.00	- no changes
FB_Read_config_MA_04	2.00	- no changes
FB_Read_config_MA_05	2.00	- no changes
FB_Read_config_MA_06	2.00	- no changes
FB_Read_config_MA_07	2.00	- no changes

FB_Read_config_MA_08	2.00	- no changes
FB_Read_config_MA_10	2.00	- no changes
FB_Read_config_MA_11	2.00	- no changes
FB_Read_config_MA_12	2.00	- no changes
FB_Read_config_MA_33	2.00	- no changes
FB_Write_config_AI_Module	2.03	- optimized valve access enabled
FB_Write_config_DI_Module	2.02	- no changes
FB_Write_config_MA_01	2.00	- no changes
FB_Write_config_MA_02	2.01	- no changes
FB_Write_config_MA_03	2.01	- no changes
FB_Write_config_MA_04	2.01	- function block numbering set to Automatic - optimized valve access enabled
FB_Write_config_MA_05	2.00	- no changes
FB_Write_config_MA_06	2.00	- no changes
FB_Write_config_MA_07	2.00	- no changes
FB_Write_config_MA_08	2.01	- no changes
FB_Write_config_MA_10	2.01	- no changes
FB_Write_config_MA_11	2.00	- no changes
FB_Write_config_MA_12	2.00	- no changes
FB_Write_config_MA_33	2.01	- no changes

Version 3.5

object	version	change
eRespondingReadMalfunctionList	1.00	- no changes
eResponseTeachInRun	1.06	- no changes
eValveMode	1.02	- no changes
eValveState	1.00	- no changes
eResponseToValveModeSet	1.01	- no changes
eValveType_MA_01_02	1.00	- no changes
eMA100OperatingMedium	1.00	- no changes
eSavingResponse	1.01	- no changes
eTransferResponse	1.03	- added value 31 (#3872358)
stMalfunction	1.01	- added timestamp (#4417523)
stTimeStamp	1.00	- created (#4417523)
stTransferPackage	1.00	- no changes
FB_Read_Bits_From_Byte	1.00	- no changes
FB_Read_Bits_From_Word	1.01	- no changes
FB_Write_Value_To_Word	1.00	- no changes
FB_AppOptionGenerator_MA_02	1.00	- no changes
FB_TeachInRun	1.01	- no changes
FB_ValveControl	2.03	- no changes
FB_StateInterpreter_MA_01	1.01	- no changes
FB_StateInterpreter_MA_02	1.02	- no changes
FB_StateInterpreter_MA_03	1.01	- no changes
FB_StateInterpreter_MA_04	1.01	- no changes

FB_StateInterpreter_MA_05	1.02	- no changes
FB_StateInterpreter_MA_06	1.01	- no changes
FB_StateInterpreter_MA_07	1.01	- no changes
FB_StateInterpreter_MA_08	1.01	- no changes
FB_StateInterpreter_MA_10	1.00	- no changes
FB_StateInterpreter_MA_11	1.02	- no changes
FB_StateInterpreter_MA_12	1.01	- no changes
FB_StateInterpreter_MA_33	1.00	- no changes
FB_StateInterpreter_TeachInRun	1.01	- no changes
FB_Download_Multiple_Parameters	2.01	- no changes
FB_Download_Single_Parameter	2.01	- no changes
FB_ReadMalfunctionList	1.01	- added timestamp information (#4417523)
FB_Save_Configuration_Persistent	2.02	- no changes
FB_Upload_Multiple_Parameters	2.01	- no changes
FB_Upload_Single_Parameter	2.01	- no changes
FB_Read_config_AI_Module	2.00	- created (#3872358)
FB_Read_config_DI_Module	2.00	- created (#3872358)
FB_Read_config_MA_01	2.00	- created (#3872358)
FB_Read_config_MA_02	2.00	- created (#3872358)
FB_Read_config_MA_03	2.00	- created (#3872358)
FB_Read_config_MA_04	2.00	- created (#3872358)
FB_Read_config_MA_05	2.00	- created (#3872358)
FB_Read_config_MA_06	2.00	- created (#3872358)
FB_Read_config_MA_07	2.00	- created (#3872358)
FB_Read_config_MA_08	2.00	- created (#3872358)
FB_Read_config_MA_10	2.00	- created (#3872358)
FB_Read_config_MA_11	2.00	- created (#3872358)
FB_Read_config_MA_12	2.00	- created (#3872358)
FB_Read_config_MA_33	2.00	- created (#3872358)
FB_Write_config_AI_Module	2.02	- no changes
FB_Write_config_DI_Module	2.02	- no changes
FB_Write_config_MA_01	2.00	- no changes
FB_Write_config_MA_02	2.01	- added application parameter iOperatingMode (#4414285)
FB_Write_config_MA_03	2.01	- no changes
FB_Write_config_MA_04	2.00	- no changes
FB_Write_config_MA_05	2.00	- no changes
FB_Write_config_MA_06	2.00	- no changes
FB_Write_config_MA_07	2.00	- no changes
FB_Write_config_MA_08	2.01	- added tuning parameters (#4510540)
FB_Write_config_MA_10	2.01	- no changes
FB_Write_config_MA_11	2.00	- no changes
FB_Write_config_MA_12	2.00	- no changes
FB_Write_config_MA_33	2.01	- no changes

Version 3.4

object	version	change
eRespondingReadMalfunctionList	1.00	- no changes
eResponseTeachInRun	1.06	- added new values 10 and 11 (#3800496)
eValveMode	1.02	- added value 10 (#3873894)
eValveState	1.00	- no changes
eResponseToValveModeSet	1.01	- no changes
eValveType_MA_01_02	1.00	- created (#3920106)
eMA100operatingMedium	1.00	- created (#3873393)
eSavingResponse	1.01	- no changes
eTransferResponse	1.02	- no changes
stMalfunction	1.00	- no changes
stTransferPackage	1.00	- no changes
FB_Read_Bits_From_Byte	1.00	- no changes
FB_Read_Bits_From_Word	1.01	- no changes
FB_Write_Value_To_Word	1.00	- no changes
FB_AppOptionGenerator_MA_02	1.00	- created (#3919901)
FB_TeachInRun	1.01	- integrated new app state return values #10 and #11 (#3800496)
FB_ValveControl	2.03	- added code to reset iErrorCode if ValveState is not „Failure“ or „NotReady“ (#2245880)
FB_StateInterpreter_MA_01	1.01	- no changes
FB_StateInterpreter_MA_02	1.02	- added outputs for indicating meaning of actual values (#3799739)
FB_StateInterpreter_MA_03	1.01	- no changes
FB_StateInterpreter_MA_04	1.01	- no changes
FB_StateInterpreter_MA_05	1.02	- added outputs for indicating meaning of actual values (#3799739)
FB_StateInterpreter_MA_06	1.01	- no changes
FB_StateInterpreter_MA_07	1.01	- no changes
FB_StateInterpreter_MA_08	1.01	- no changes
FB_StateInterpreter_MA_10	1.00	- created (#3805731)
FB_StateInterpreter_MA_11	1.02	- no changes
FB_StateInterpreter_MA_12	1.01	- no changes
FB_StateInterpreter_MA_33	1.00	- no changes
FB_StateInterpreter_TeachInRun	1.01	- added new values 10 and 11 (#3800496)
FB_Download_Multiple_Parameters	2.01	- no changes
FB_Download_Single_Parameter	2.01	- no changes
FB_ReadMalfunctionList	1.00	- no changes
FB_Save_Configuration_Persistent	2.02	- no changes
FB_Upload_Multiple_Parameters	2.01	- no changes
FB_Upload_Single_Parameter	2.01	- no changes
FB_Write_config_AI_Module	2.02	- added new sensor types within comments (#3787601)
FB_Write_config_DI_Module	2.02	- no changes
FB_Write_config_MA_01	2.00	- no changes
FB_Write_config_MA_02	2.00	- no changes

FB_Write_config_MA_03	2.01	- included new application parameters (#3800158)
FB_Write_config_MA_04	2.00	- no changes
FB_Write_config_MA_05	2.00	- no changes
FB_Write_config_MA_06	2.00	- no changes
FB_Write_config_MA_07	2.00	- no changes
FB_Write_config_MA_08	2.00	- no changes
FB_Write_config_MA_10	2.01	- created (#3898591)
FB_Write_config_MA_11	2.00	- no changes
FB_Write_config_MA_12	2.00	- no changes
FB_Write_config_MA_33	2.01	- included new application parameters (#4102822)

Version 3.3

object	version	change
eRespondingReadMalfunctionList	1.00	- no changes
eResponseTeachInRun	1.05	- no changes
eValveMode	1.01	- no changes
eValveState	1.00	- no changes
eResponseToValveModeSet	1.01	- no changes
eSavingResponse	1.01	- no changes
eTransferResponse	1.02	- no changes
stMalfunction	1.00	- no changes
stTransferPackage	1.00	- no changes
FB_Read_Bits_From_Byte	1.00	- no changes
FB_Read_Bits_From_Word	1.01	- no changes
FB_Write_Value_To_Word	1.00	- no changes
eResponseToValveModeSet	1.01	- no changes
FB_TeachInRun	1.00	- no changes
FB_ValveControl	2.02	- no changes
FB_StateInterpreter_MA_01	1.01	- no changes
FB_StateInterpreter_MA_02	1.01	- no changes
FB_StateInterpreter_MA_03	1.01	- no changes
FB_StateInterpreter_MA_04	1.01	- no changes
FB_StateInterpreter_MA_05	1.01	- no changes
FB_StateInterpreter_MA_06	1.01	- no changes
FB_StateInterpreter_MA_07	1.01	- no changes
FB_StateInterpreter_MA_08	1.01	- no changes
FB_StateInterpreter_MA_11	1.02	- no changes
FB_StateInterpreter_MA_12	1.01	- no changes
FB_StateInterpreter_MA_33	1.00	- created (#3625800)
FB_StateInterpreter_TeachInRun	1.00	- no changes
FB_Download_Multiple_Parameters	2.01	- no changes
FB_Download_Single_Parameter	2.01	- no changes
FB_ReadMalfunctionList	1.00	- no changes

FB_Save_Configuration_Persistent	2.02	- no changes
FB_Upload_Multiple_Parameters	2.01	- no changes
FB_Upload_Single_Parameter	2.01	- no changes
FB_Write_config_AI_Module	2.02	- no changes
FB_Write_config_DI_Module	2.02	- no changes
FB_Write_config_MA_01	2.00	- no changes
FB_Write_config_MA_02	2.00	- no changes
FB_Write_config_MA_03	2.00	- no changes
FB_Write_config_MA_04	2.00	- no changes
FB_Write_config_MA_05	2.00	- no changes
FB_Write_config_MA_06	2.00	- no changes
FB_Write_config_MA_07	2.00	- no changes
FB_Write_config_MA_08	2.00	- no changes
FB_Write_config_MA_11	2.01	- no changes
FB_Write_config_MA_12	2.01	- no changes
FB_Write_config_MA_33	1.00	- created (#3625895)

Version 3.2

object	version	change
eRespondingReadMalfunctionList	1.00	- created (#2256314)
eResponseTeachInRun	1.05	- created (#2256460)
eValveMode	1.01	- created
eValveState	1.00	- no changes
eResponseToValveModeSet	1.01	- no changes
eSavingResponse	1.01	- added value 25 „FB_needs_rising_edge_xExecute“ (#2487339)
eTransferResponse	1.02	- added value 8 (#3104465)
stMalfunction	1.00	- created (#2256314)
stTransferPackage	1.00	- no changes
FB_Read_Bits_From_Byte	1.00	- no changes
FB_Read_Bits_From_Word	1.01	- no changes
FB_Write_Value_To_Word	1.00	- no changes
eResponseToValveModeSet	1.01	- no changes
FB_TeachInRun	1.00	- created (#2256460)
FB_ValveControl	2.02	- FB detect a falling edge of xManualAcknowledge while input xEnable is FALSE
FB_StateInterpreter_MA_01	1.01	- no changes
FB_StateInterpreter_MA_02	1.01	- no changes
FB_StateInterpreter_MA_03	1.01	- no changes
FB_StateInterpreter_MA_04	1.01	- no changes
FB_StateInterpreter_MA_05	1.01	- no changes
FB_StateInterpreter_MA_06	1.01	- no changes
FB_StateInterpreter_MA_07	1.01	- no changes
FB_StateInterpreter_MA_08	1.01	- no changes

FB_StateInterpreter_MA_11	1.02	- no changes
FB_StateInterpreter_MA_12	1.01	- no changes
FB_StateInterpreter_TeachInRun	1.00	- created
FB_Download_Multiple_Parameters	2.01	- adjusted range for iChannel from 255 to 31 (#2515542) - validity check shows first wrong element and stops the procedure (#2526252) - adapted functionality for transfer mode on slots without valves (#2957510) - added new enum value (#3104465)
FB_Download_Single_Parameter	2.01	- adjusted range for iChannel from 255 to 31 (#2515542) - validity check shows first wrong element and stops the procedure (#2526252) - adapted functionality for transfer mode on slots without valves (#2957510) - added new enum value (#3104465)
FB_ReadMalfunctionList	1.00	- created
FB_Save_Configuration_Persistent	2.02	- redesign to state machine model - corrected iResponse when calling the FB for the first time with xExecute = FALSE (#2510159) - set output xSavingDone and iResponse when the FB stops sending busdata (#2534862)
FB_Upload_Multiple_Parameters	2.01	- adjusted range for iChannel from 255 to 31 (#2515542) - validity check shows first wrong element and stops the procedure (#2526252) - reintegrated sourcecode (#2521445) - adapted functionality for transfer mode on slots without valves (#2957510) - added new enum value (#3104465)
FB_Upload_Single_Parameter	2.01	- adjusted range for iChannel from 255 to 31 (#2515542) - reintegrated sourcecode (#2521445) - adapted functionality for transfer mode on slots without valves (#2957510) - added new enum value (#3104465)
FB_Write_config_AI_Module	2.02	- corrected iResponse when calling the FB for the first time with xExecute = FALSE (#2510159) - iResponse changes to „FB_needs_rising_Edge_xExecute“ when changing the inputs while the FB is running (#2515568) - iResponse changes always to „invalid_module_number“ when the input iInputModuleNr is wrong(#2515421)
FB_Write_config_DI_Module	2.02	- corrected iResponse when calling the FB for the first time with xExecute = FALSE (#2510159) - iResponse changes to „FB_needs_rising_Edge_xExecute“ when changing the inputs while the FB is running (#2515568) - iResponse changes always to „invalid_module_number“ when the input iInputModuleNr is wrong(#2515421)
FB_Write_config_MA_01	2.00	- no changes
FB_Write_config_MA_02	2.00	- no changes
FB_Write_config_MA_03	2.00	- no changes
FB_Write_config_MA_04	2.00	- no changes
FB_Write_config_MA_05	2.00	- no changes
FB_Write_config_MA_06	2.00	- no changes

FB_Write_config_MA_07	2.00	- no changes
FB_Write_config_MA_08	2.00	- no changes
FB_Write_config_MA_11	2.01	- no changes
FB_Write_config_MA_12	2.01	- no changes

Version 3.1

object	version	change
eValveState	1.00	- no changes
FB_Read_Bits_From_Byte	1.00	- no changes
FB_Read_Bits_From_Word	1.01	- Changes iValue, xError from InOut to Out (#2222801)
FB_Write_Value_To_Word	1.00	- no changes
eResponseToValveModeSet	1.01	- added new Value „FB_not_enabled := 1051“ (#2417793)
FB_ValveControl	2.01	- integrated new Value „FB_not_enabled := 1051“ (#2417793) - updated enable = FALSE behaviour - changed output awBusDataToVTEM to VAR_IN_OUT (#2406661) - added new Output sResponseToValveModeSet (#2475841)
FB_StateInterpreter_MA_01	1.01	- no changes
FB_StateInterpreter_MA_02	1.01	- no changes
FB_StateInterpreter_MA_03	1.01	- no changes
FB_StateInterpreter_MA_04	1.01	- no changes
FB_StateInterpreter_MA_05	1.01	- no changes
FB_StateInterpreter_MA_06	1.01	- no changes
FB_StateInterpreter_MA_07	1.01	- no changes
FB_StateInterpreter_MA_08	1.01	- no changes
FB_StateInterpreter_MA_11	1.02	- added outputs xPistonDetectRet and xPistonDetectAdv (#2248584)
FB_StateInterpreter_MA_12	1.01	- no changes
eSavingResponse	1.00	- no changes
eTransferResponse	1.01	- added items 27, 28, 29, 30
stTransferPackage	1.00	- no changes
FB_Write_config_MA_01	2.00	- created (#2256987)
FB_Write_config_MA_02	2.00	- added parameters for the sensor port mapping (#2256299) - changed awBusDataToVTEM into VAR_IN_OUT (#2406661)
FB_Write_config_MA_03	2.00	- added parameters for the sensor port mapping (#2256299) - changed awBusDataToVTEM into VAR_IN_OUT (#2406661)
FB_Write_config_MA_04	2.00	- added parameters for the sensor port mapping (#2256299) - changed awBusDataToVTEM into VAR_IN_OUT (#2406661)
FB_Write_config_MA_05	2.00	- added parameters for the sensor port mapping (#2256299) - changed awBusDataToVTEM into VAR_IN_OUT (#2406661)
FB_Write_config_MA_06	2.00	- created (#2408095)
FB_Write_config_MA_07	2.00	- added parameters for moment of inertia used for swivel drives (#2307857)

		- added parameters for the sensor port mapping (#2256299) - changed awBusDataToVTEM into VAR_IN_OUT (#2406661)
FB_Write_config_MA_08	2.00	- added parameters for moment of inertia used for swivel drives (#2307857) - added parameters for the sensor port mapping (#2256299) - changed awBusDataToVTEM into VAR_IN_OUT (#2406661)
FB_Write_config_MA_11	2.00	- added parameters for the sensor port mapping (#2256299) - removed tuning parameters (#2408215) - changed awBusDataToVTEM into VAR_IN_OUT (#2406661)
FB_Write_config_MA_12	2.00	- added parameters for moment of inertia used for swivel drives (#2307857) - changed awBusDataToVTEM into VAR_IN_OUT (#2406661)
FB_Download_Multiple_Parameters	2.00	- changed awBusDataToVTEM into VAR_IN_OUT (#2406661) - fixed bug that caused „exit transfer mode“ even if xExecute is FALSE (#2246210) - added value 7 to list of invalid commands (#2318899)
FB_Download_Single_Parameter	2.00	- changed awBusDataToVTEM into VAR_IN_OUT (#2406661) - fixed bug that caused „exit transfer mode“ even if xExecute is FALSE (#2246210) - added value 7 to list of invalid commands (#2406621)
FB_Save_Configuration_Persistent	2.01	- changed awBusDataToVTEM into VAR_IN_OUT (#2406661) - fixed bug that caused „exit transfer mode“ even if xExecute is FALSE (#2458152)
FB_Upload_Multiple_Parameters	2.00	- changed awBusDataToVTEM into VAR_IN_OUT (#2406661) - fixed bug that caused „exit transfer mode“ even if xExecute is FALSE (#2246210) - added value 7 to list of invalid commands (#2406621) - fixed a bug regarding the stability of the function block (#2262058)
FB_Upload_Single_Parameter	2.00	- changed awBusDataToVTEM into VAR_IN_OUT (#2406661) - fixed bug that caused „exit transfer mode“ even if xExecute is FALSE (#2246210) - added value 7 to list of invalid commands (#2406621)
FB_Write_config_AI_Module	2.01	- created
FB_Write_config_DI_Module	2.01	- created

Version 2.0

-- First official release --

=====

Urheberrechtshinweis

Diese Unterlagen sind geistiges Eigentum der Festo SE & Co. KG, der auch das ausschließliche Urheberrecht daran zusteht. Eine inhaltliche Änderung, die Vervielfältigung oder der Nachdruck dieser Unterlagen sowie deren Weitergabe an Dritte ist nur mit der ausdrücklichen Erlaubnis der Festo SE & Co. KG gestattet.

Festo SE & Co. KG behält sich das Recht vor, dieses Dokument vollständig oder teilweise zu ändern. Alle Marken- und Produktnamen sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Titelhälter.

Rechtliche Hinweise

Hardware, Software, Betriebssysteme und Treiber dürfen nur für die beschriebenen Einsatzfälle und nur in Verbindung mit den von Festo SE & Co. KG empfohlenen Komponenten verwendet werden.

Festo SE & Co. KG lehnt jede Haftung für Schäden ab, die durch die Anwendung von allenfalls falschen bzw. unzureichenden Informationen oder aufgrund fehlender Informationen in diesen Unterlagen entstehen.

Defekte, die durch unsachgemäße Behandlung von Geräten und Baugruppen entstehen, sind von der Gewährleistung ausgeschlossen.

Sicherheitsrelevante Funktionen, im Sinne von Personen- und Maschinenschutz, dürfen mit Angaben und Informationen aus diesem Dokument nicht realisiert werden.

Für Folgeschäden, die durch einen Ausfall oder eine Funktionsstörung entstehen, wird dann jede Haftung abgelehnt. Im Übrigen gelten die Regelungen bzgl. Haftung aus den Liefer-, Zahlungs- und Softwarenutzungsbedingungen der Festo SE & Co. KG, welche Sie unter www.festo.com finden, welche wir Ihnen aber auch auf Anforderung gerne zukommen lassen.

Alle in diesem Dokument angegebenen Daten sind keine zugesicherten Eigenschaften, insbesondere nicht für Funktionalität, Zustand oder Qualität im rechtlichen Sinn.

Die jeweiligen Bedienungsanleitungen der Festo Produkte sind unter www.festo.com/sp zu finden.

Der Benutzer dieses Dokuments (Funktion und Anwendung) muss selbst sicherstellen, dass jede Funktion die hier beschrieben ist, auch in seiner Applikation ordnungsgemäß funktioniert. Der Benutzer bleibt auch durch das Studium dieses Dokuments sowie der Nutzung der darin genannten Angaben weiterhin allein verantwortlich für die eigene Anwendung.

Inhaltsverzeichnis

1	Wichtige Hinweise	16
1.1	Bestimmungsgemäße Verwendung	16
1.1.1	Zielgruppe	16
1.1.2	Service	16
1.2	Sicherheitshinweise	16
1.3	Kennzeichnung spezieller Informationen	16
1.3.1	Piktogramme	16
1.3.2	Textkennzeichnungen	17
1.3.3	Weitere Konventionen	17
2	Einführung	18
2.1	Grundlagen	18
2.1.1	Aufbau der Bibliothek	19
2.1.1.1	Datentypen	19
2.2	Voraussetzungen	20
2.2.1	Anbindung des CPX-Terminals an die Steuerung	20
2.2.2	Einbinden der Bibliothek	20
2.3	Allgemeine Hinweise zum Motion Terminal VTEM	21
2.3.1	Dokumentationen zum Motion Terminal VTEM	21
2.4	Kommunikationsprotokoll des Motion Terminals VTEM	21
2.5	E/A-Daten mit den Bausteinen verbinden	23
3	FB_ValveControl – Steuerung eines Ventils durch den Betrieb einer Motion App	25
3.1	Funktionsbeschreibung	25
3.2	Ein- und Ausgänge	26
3.2.1	Der Datentyp eResponseToValveModeSet	27
3.2.2	Der Datentyp eValveType_MA_01_02	28
4	FB_AppOptionGenerator_MA_02 – Erzeugung des App-Option-Werts für MA#02	29
4.1	Funktionsbeschreibung	29
4.2	Ein- und Ausgänge	29
5	FB_StateInterpreter_... – Interpretation des App-Status	30
5.1	Funktionsbeschreibung	30
5.2	Ein- und Ausgänge	30
5.2.1	Motion App #01 (Wegeventilfunktionen)	31
5.2.2	Motion App #02 (Proportional-Wegeventil)	31
5.2.3	Motion App #05 (Zu- und Abluftdrosselung)	32
5.2.4	Motion App #10 (Durchflussregelung)	32
5.2.5	Motion App #11 (Soft Stop)	32
5.2.6	Motion App #12 (Diagnose Leckage)	33
5.2.7	Motion App #33 (Positionieren)	33
5.2.8	Lernfahrt	33
6	FB_Read_config_MA_... – Parametrierung einer Motion App lesen	34
6.1	Funktionsbeschreibung	34
6.2	Ein- und Ausgänge	34
6.2.1	Der Datentyp eTransferResponse	35
7	FB_Write_config_MA_... – Parametrierung einer Motion App schreiben	37
7.1	Funktionsbeschreibung	37
7.2	Ein- und Ausgänge	37
7.2.1	Beispiel zur Zuweisung der Eingänge	38

8	FB_Read_config_DI/AI_Module – Parametrierung der Eingangsmodule lesen	40
8.1	Funktionsbeschreibung	40
8.2	Ein- und Ausgänge	41
9	FB_Write_config_DI/AI_Module – Parametrierung der Eingangsmodule schreiben	43
9.1	Funktionsbeschreibung	43
9.2	Ein- und Ausgänge	44
10	FB_ReadMalfunctionList – Störungsliste auslesen	46
10.1	Funktionsbeschreibung	46
10.2	Ein- und Ausgänge	46
10.2.1	Der Datentyp eResponseReadMalfunctionList.....	47
10.2.2	Der Datentyp stMalfunction.....	48
10.2.3	Der Datentyp stTimeStamp.....	48
11	FB_TeachInRun – Lernfahrt parametrieren und ausführen	49
11.1	Funktionsbeschreibung	49
11.2	Ein- und Ausgänge	49
11.2.1	Der Datentyp eResponseTeachInRun.....	50
12	FB_Save_Configuration_Persistent – Parametrierung persistent speichern	52
12.1	Funktionsbeschreibung	52
12.2	Ein- und Ausgänge	52
12.2.1	Der Datentyp eSavingResponse	52
13	FB_Download_Single_Parameter – Einzelne Parameter zum Motion Terminal übertragen	54
13.1	Funktionsbeschreibung	54
13.2	Ein- und Ausgänge	54
14	FB_Download_Multiple_Parameters – Mehrere Parameter zum Motion Terminal übertragen	55
14.1	Funktionsbeschreibung	55
14.2	Ein- und Ausgänge	55
14.2.1	Der Datentyp stTransferPackage	56
15	FB_Upload_Single_Parameter – Einzelnen Wert vom Motion Terminal auslesen	57
15.1	Funktionsbeschreibung	57
15.2	Ein- und Ausgänge	57
16	FB_Upload_Multiple_Parameters – Mehrere Werte vom Motion Terminal auslesen.....	58
16.1	Funktionsbeschreibung	58
16.2	Ein- und Ausgänge	58
17	Hilfbausteine	60
17.1	FB_Read_Bits_From_Byte – Bit-Bereich aus Byte auslesen	60
17.1.1	Funktionsbeschreibung	60
17.1.2	Ein- und Ausgänge.....	60
17.2	FB_Read_Bits_From_Word – Bit-Bereich aus Wort auslesen	61
17.2.1	Funktionsbeschreibung	61
17.2.2	Ein- und Ausgänge.....	61
17.3	FB_Write_Value_To_Word – Wert innerhalb eines Wortes schreiben	61
17.3.1	Funktionsbeschreibung	61
17.3.2	Ein- und Ausgänge.....	61
A	Technischer Anhang.....	62
A.1	Technische Daten	62
A.1.1	Allgemein	62
B	Index.....	63

1 Wichtige Hinweise

1.1 Bestimmungsgemäße Verwendung

Die Funktionsbausteine (FBs) der hier beschriebenen Bibliothek dienen zur Steuerung und Parametrierung des Motion Terminals VTEM von Festo.

Mit den Funktionsbausteinen können die vielfältigen Funktionen dieses Geräts komfortabel in das SPS-Programm eingebunden werden. Dabei werden von den Funktionsbausteinen keine zusätzlichen Funktionen realisiert, sondern die Bedienung der geräteseitig implementierten Funktionen wesentlich erleichtert.

Die Funktionsbausteine werden für jedes Ventil des Motion Terminals mit einer separaten Instanz zyklisch aufgerufen. Die gleichzeitige Ausführung mehrerer Funktionsbausteine zur Steuerung desselben Ventils ist unzulässig. Es wird daher empfohlen, die Implementierung über eine Zustandsmaschine zu realisieren, um den Zustand des gleichzeitigen Zugriffs ausschließen zu können.



Unsicherer Zustand der Anlage bei parallelem Aufruf von Bausteinen

Bei gleichzeitigem Zugriff von mehreren Funktionsbausteinen auf die Ausgangsdaten eines Ventils (wBusDataToVTEM...) kann es zu unvorhersehbarem Verhalten des Motion Terminals sowie der angeschlossenen Peripherie kommen.

- Sicherstellen, dass niemals mehr als eine Bausteininstanz zur selben Zeit auf die Ausgangsdaten eines Ventils (wBusDataToVTEM...) zugreift.

Um eine anwenderseitige Anpassung der Funktionsbausteine zu ermöglichen, werden die Funktionsbausteine ungeschützt im Quelltext zur Verfügung gestellt. Änderungen des Quelltextes der Funktionsbausteine sollten nur durch fachkundiges Personal durchgeführt werden und können Auswirkungen auf die Supportleistungen zur Folge haben.

Die „Sicherheitstechnischen Hinweise“ sowie der bestimmungsgemäße Gebrauch der jeweiligen Geräte, Baugruppen und Module ist zu beachten. Beim Anschluss handelsüblicher Zusatzkomponenten, wie Sensoren und Aktuatoren, sind die angegebenen Grenzwerte für Drücke, Temperaturen, elektrische Daten, Momente usw. einzuhalten.

1.1.1 Zielgruppe

Diese Beschreibung wendet sich ausschließlich an ausgebildete Fachleute der Steuerungs- und Automatisierungstechnik, die Erfahrungen mit der Installation, Inbetriebnahme, Programmierung und Diagnose von Positioniersystemen und den relevanten Feldbussen besitzen.

1.1.2 Service

Bei technischen Problemen kann der lokale Service von Festo oder der zentrale Service über folgende E-Mail-Adresse helfen:

service_international@festo.com

1.2 Sicherheitshinweise

Bei der Inbetriebnahme und Programmierung von pneumatischen Systemen sind unbedingt die in den Beschreibungen und Bedienungsanleitungen zu den eingesetzten Komponenten gegebenen Sicherheitsvorschriften zu beachten. Der Anwender hat dafür Sorge zu tragen, dass sich niemand im Einflussbereich der angeschlossenen Aktuatoren aufhält. Der mögliche Gefahrenbereich muss durch geeignete Maßnahmen wie Absperrungen oder Warnhinweise gesichert werden.

1.3 Kennzeichnung spezieller Informationen

1.3.1 Piktogramme

Folgende Piktogramme kennzeichnen Textstellen, die spezielle Informationen enthalten:



Informationen

Empfehlungen, Tipps und Verweise auf andere Informationsquellen

1.3.2 Textkennzeichnungen

1. Ziffern kennzeichnen Tätigkeiten, die in der angegebenen Reihenfolge durchzuführen sind.
- Der Auflistungspunkt kennzeichnet Tätigkeiten, die in beliebiger Reihenfolge durchgeführt werden können.
- Der Spiegelstrich kennzeichnet allgemeine Aufzählungen.

1.3.3 Weitere Konventionen

FB	Die Abkürzung „FB“ wird für den Begriff „Funktionsbaustein“ an einigen Stellen genutzt (auch „FBs“)
„OK“	Namen von Fenstern, Dialogen und Schaltflächen, z. B. „Meldefenster“, „Projekt dearchivieren“, „OK“ sowie Bezeichnungen werden in Anführungszeichen dargestellt.
STRG	Namen von Tasten auf der PC-Tastatur werden im Text mit Großbuchstaben dargestellt (z. B. EINGABETASTE, STRG, C, F1, usw.). Beispiel: Die STRG-Taste gemeinsam mit der Taste „C“ erzeugt das Zeichen STRG+C.
STRG+C	Für einige Funktionen müssen zwei Tasten gleichzeitig gedrückt werden. Beispiel: Die STRG-Taste gemeinsam mit der Taste „C“ erzeugt das Zeichen STRG+C.
	Wird einfach von „Klicken“ oder „Doppelklick“ gesprochen, so ist immer die linke Maustaste gemeint. Falls die rechte Maustaste benutzt werden soll, wird ausdrücklich darauf hingewiesen.

2 Einführung

2.1 Grundlagen

Die Bibliothek Festo_VTEM_DevCon ist eine interne Bibliothek für Siemens TIA Portal ab Version 19.

Die Bausteine der Bibliothek unterstützen den Anwender bei der Parametrierung und dem Betrieb von Motion Apps auf dem Motion Terminal VTEM von Festo. Das Motion Terminal VTEM kann dabei über eine passende Feldbus-Schnittstelle des CPX-Terminals gesteuert und parametriert werden.



Informationen

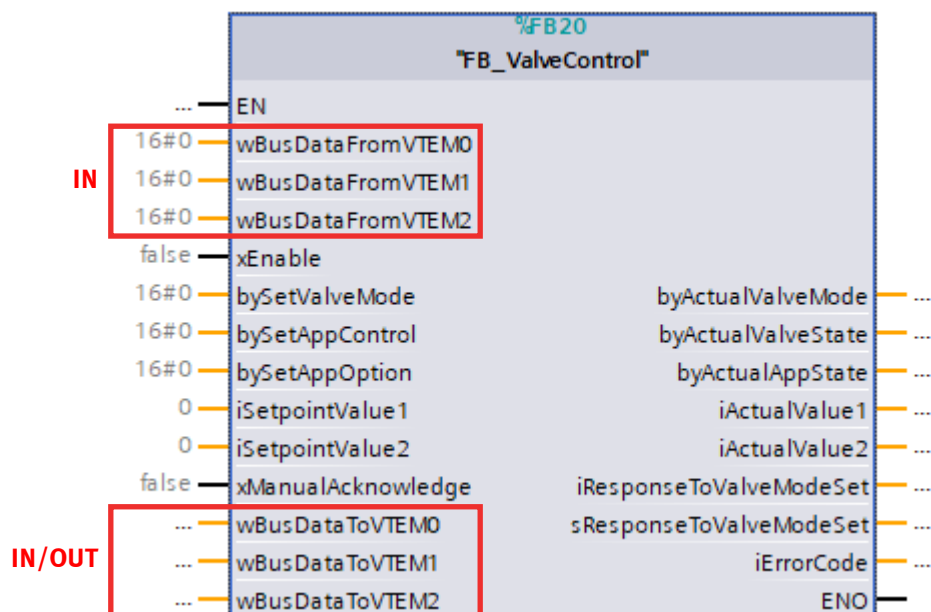
Die Bausteine stellen eine vereinfachte Bedienung des Protokolls dar, das in der Anwenderdokumentation des Motion Terminals VTEM beschrieben ist. Alle hier beschriebenen Funktionen der Bausteine können mit entsprechendem Aufwand auch direkt über die zyklischen Prozessdaten des Motion Terminals im Anwenderprogramm implementiert werden.

Detaillierte Informationen über die Funktion, Bedienung und Parametrierung sind der Anwenderdokumentation des Motion Terminals VTEM zu entnehmen (→ [Dokumentationen zum Motion Terminal VTEM](#)).

Architektur

Die Steuerung des Motion Terminals VTEM geschieht auf Ventilebene über die Motion Apps und mit Hilfe der Bausteine. Das bedeutet, die Bausteine steuern bzw. parametrieren jeweils eine Motion App auf einem Ventil. Die Verbindung der Bausteine zu den Ventilen geschieht über die Ein- und Ausgangsdaten des CPX-Terminals.

Alle Steuerungs- bzw. Parametrierbausteine besitzen jeweils einen Eingang und einen Ein-Ausgang zur Zuordnung der E/A-Daten des Ventils:



- wBusDataFromVTEM...: Eingangsdaten des Ventils (3 Worte)
- wBusDataToVTEM...: Ausgangsdaten für das Ventil (3 Worte)



Unsicherer Zustand der Anlage bei parallelem Aufruf von Bausteinen

Bei gleichzeitigem Zugriff von mehreren Funktionsbausteinen auf die Ausgangsdaten eines Ventils (wBusDataToVTEM...) kann es zu unvorhersehbarem Verhalten des Motion Terminals sowie der angeschlossenen Peripherie kommen.

- Sicherstellen, dass niemals mehr als eine Bausteininstanz zur selben Zeit auf die Ausgangsdaten eines Ventils (wBusDataToVTEM...) zugreift.



Es wird empfohlen, für jedes Ventil eine eigene Instanz der Bausteine zu verwenden.

2.1.1 Aufbau der Bibliothek

Die Bibliothek umfasst verschiedene Datentypen und Bausteine mit unterschiedlichen Funktionen und Inhalten. Einige Datentypen werden innerhalb der Bausteine als Liste numerischer Werte verwendet, so dass diese Werte über einen Bezeichner-Namen adressiert werden können („Enumeration“).

Beispiel für die Verwendung der Datentypen `eValveState` und `eResponseToValveModeSet`

```
IF #byActualValveState = INT_TO_BYTE(#eValveState.Failure) THEN
    #iResponseToValveModeSet := #eResponseToValveModeSet.failure;
```

Bausteine

Die Bausteine werden in den folgenden Kapiteln detailliert beschrieben.

2.1.1.1 Datentypen

`eValveMode`

Der Datentyp wird als Enumeration verwendet und enthält die Bezeichnungen der einzelnen Ventil-Betriebsmodi. Die Integer-Werte zwischen 1 und 59 entsprechen der jeweiligen Motion-App-ID.

`eValveState`

Der Datentyp wird als Enumeration verwendet und enthält die möglichen Ventilzustände. Die Integer-Werte entsprechend dem Dezimalwert der entsprechenden Bits in den Prozessdaten.

`eResponseToValveModeSet`

Der Datentyp wird als Enumeration verwendet und enthält Rückmeldungen zu den aktuell anliegenden Werten in den Prozess-Ausgangsdaten. Die Integer-Werte entsprechen den Dezimalwerten im Bereich „App State“, erweitert um zusätzliche Rückmeldungen des Bausteins „FB_ValveControl“.

`eValveType_MA_01_02`

Der Datentyp wird als Enumeration verwendet und enthält die Werte für die Ventiltypen der Motion Apps #01 und #02, die über den Bereich „App Option“ in den Prozess-Ausgangsdaten übertragen werden.

`eMA10OperatingMedium`

Der Datentyp wird als Enumeration verwendet und enthält die Werte für den Parameter „Operating Medium“ für die Motion App #10 (Durchflussregelung).

`eSavingResponse`

Der Datentyp wird als Enumeration verwendet und enthält Rückmeldungen zum Befehl „Persistent speichern“, der unter anderem über den Baustein „FB_Save_Configuration_Persistent“ gestartet werden kann.

`eTransferResponse`

Der Datentyp wird als Enumeration verwendet und enthält Rückmeldungen zu gestarteten Transfermodus-Befehlen und wird von den Bausteinen zur Steuerung des Transfermodus verwendet.

`eResponseReadMalfunctionList`

Der Datentyp wird als Enumeration verwendet und enthält die vom Baustein „FB_ReadMalfunctionList“ verwendeten Rückmeldungen.

`eResponseTeachInRun`

Der Datentyp wird als Enumeration verwendet und enthält sowohl die Rückmeldungen des Bausteins „FB_TeachInRun“ als auch die Rückmeldungen der Lernfahrt selbst.

`stTransferPackage`

Der Datentyp wird zur Aufnahme von Parametrierungen in eine Liste genutzt, die mit den Bausteinen „FB_upload_Multiple_Parameters“ und „FB_download_Multiple_Parameters“ übertragen werden können.

`stMalfunction`

Der Datentyp bildet den Aufbau eines Eintrags in der Störungsliste ab und enthält den Störungscode, den Störungs-Subcode und die Einstufung der Störung.

2.2 Voraussetzungen

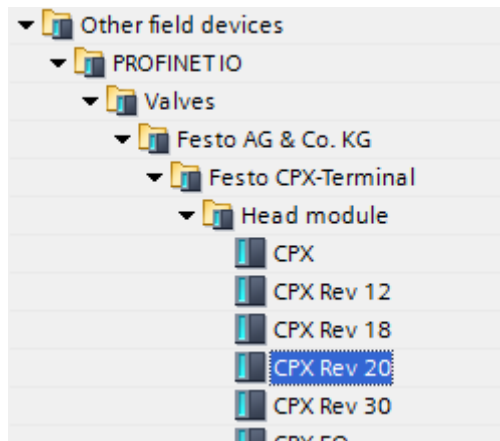
2.2.1 Anbindung des CPX-Terminals an die Steuerung



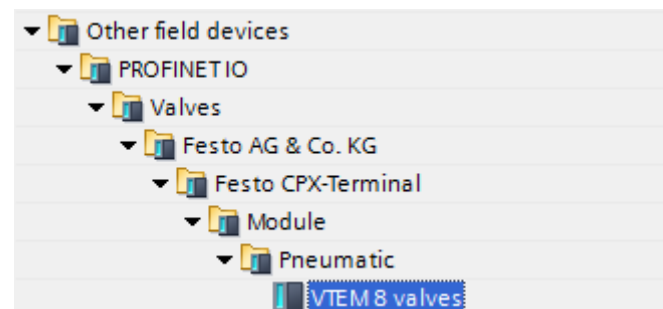
Gerätebeschreibungsdatei

Zur Nutzung des Motion Terminals VTEM muss eine aktuelle Gerätebeschreibungsdatei des CPX-Systems für Profinet (Busknoten CPX-FB33) in das Steuerungsprojekt eingebunden werden.

Die Gerätebeschreibungsdatei kann im Support Portal von Festo geladen werden (→ www.festo.com/sp).

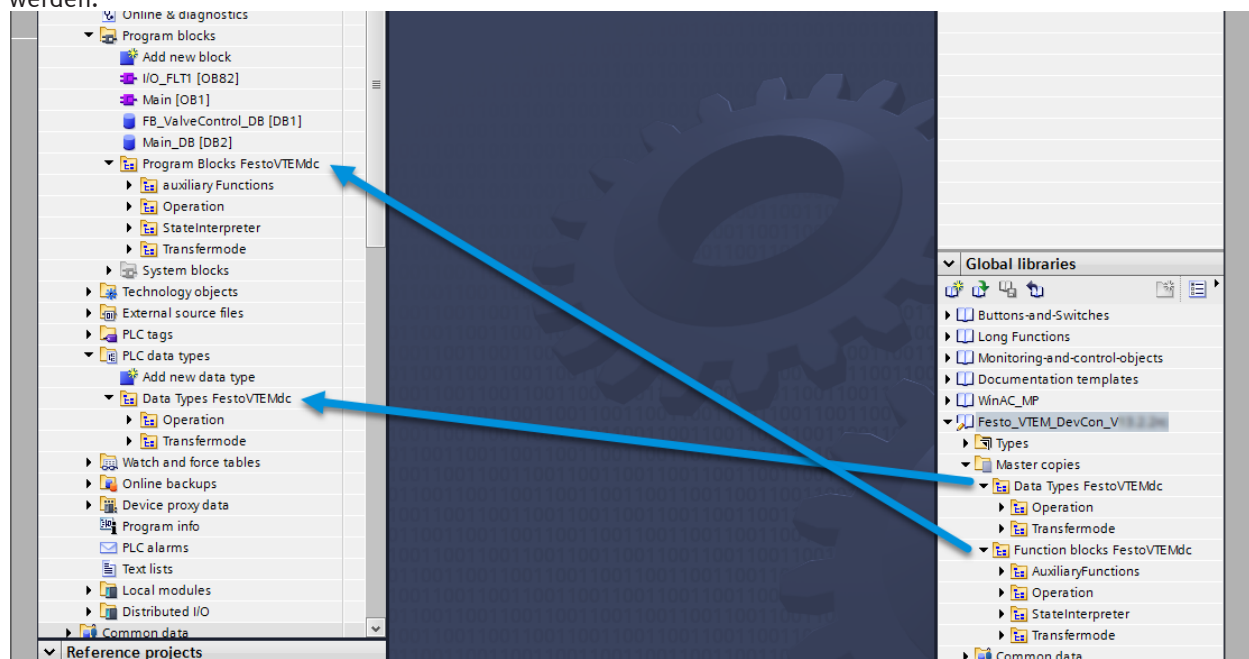


Das Motion Terminal VTEM wird als Submodul des CPX-Terminals eingebunden:



2.2.2 Einbinden der Bibliothek

Die benötigten Bausteine müssen aus der Bibliothek „Festo_VTEM_DevCon“ in das Steuerungsprojekt kopiert werden.



2.3 Allgemeine Hinweise zum Motion Terminal VTEM

Für die Inbetriebnahme und die Handhabung des Motion Terminals ist ein sicheres Verständnis der Funktion des Produkts unbedingt erforderlich.

Folgende zusätzliche Beschreibungen sind zum Verständnis notwendig:

2.3.1 Dokumentationen zum Motion Terminal VTEM

Dokumentation	Inhalt
Quick Start Guide	Beispielhafte Beschreibung der WebConfig-Oberfläche
Quick Reference PLC programming	Kompakte Übersicht aller Funktionen und Parameter des Motion Terminals, die per SPS genutzt werden können.
„Anleitung Gebrauch Motion Terminal VTEM“ (VTEM)	Anleitung zum sicheren Gebrauch und wichtige Hinweise zu Handhabung, Installation, Inbetriebnahme und Wartung des Motion Terminals VTEM.
„Beschreibung Motion Terminal VTEM, Funktion und Parametrierung“ (VTEM-BES-...)	Ausführliche Beschreibung des Motion Terminals VTEM sowie dessen Funktion und Parametrierung
„Beschreibung Motion Terminal VTEM, Motion App ...“ (VTEM-MA...-... / GAMM-A...-...)	Ausführliche Beschreibung der Motion Apps für das Motion Terminal VTEM
„CPX Systembeschreibung“ (CPX-SYS-...)	Informationen zum Gesamtsystem des CPX-Terminals

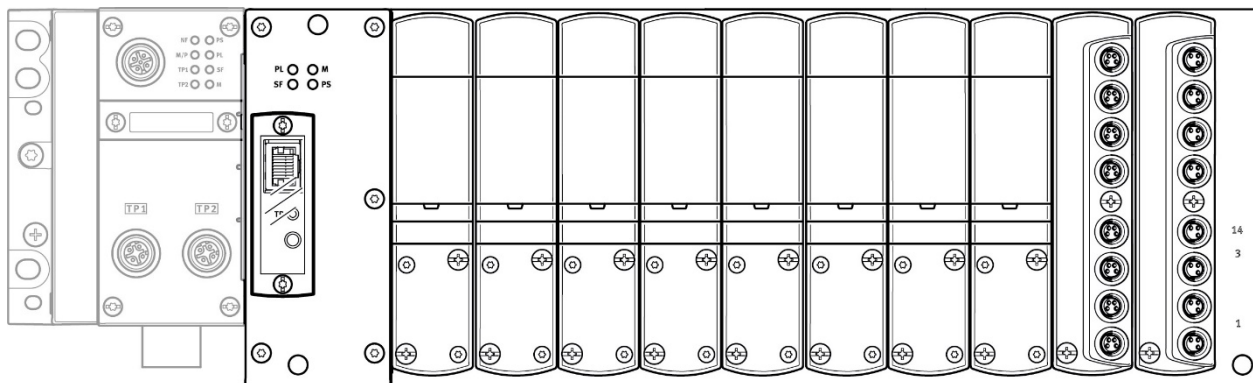


Alle verfügbaren Dokumente zum Produkt → www.festo.com/pk

Die in den Dokumenten aufgeführten Informationen und Sicherheitshinweise sind zu beachten!

2.4 Kommunikationsprotokoll des Motion Terminals VTEM

Die Kommunikation zwischen der übergeordneten Steuerung (SPS) und dem Controller des Motion Terminals basiert auf je 8 × 6 Byte Eingangs- und Ausgangsdaten des CPX-Terminals. Jedem der maximal 8 Ventilsteckplätze eines Motion Terminals sind dabei 6 Byte Eingangsdaten (Process Data Input, PDI) und 6 Byte Ausgangsdaten (Process Data Output, PDO) zugeordnet, unabhängig von der tatsächlich vorhandenen Anzahl an Ventilen.



Ventil auf Steckplatz 0	Ventil auf Steckplatz 1	Ventil auf Steckplatz 2	Ventil auf Steckplatz 3	Ventil auf Steckplatz 4	Ventil auf Steckplatz 5	Ventil auf Steckplatz 6	Ventil auf Steckplatz 7
6 Byte PDI	6 Byte PDI	6 Byte PDI	6 Byte PDI	6 Byte PDI	6 Byte PDI	6 Byte PDI	6 Byte PDI
6 Byte PDO	6 Byte PDO	6 Byte PDO	6 Byte PDO	6 Byte PDO	6 Byte PDO	6 Byte PDO	6 Byte PDO



Der genau Aufbau der Eingangs- und Ausgangsdaten ist in der Beschreibung „Motion Terminal VTEM Funktion und Parametrierung“ ausführlich beschrieben.

Diese Informationen werden in der vorliegenden Beschreibung als bekannt vorausgesetzt.

2.5 E/A-Daten mit den Bausteinen verbinden

Im Programm, das die Bausteine dieser Bibliothek verwendet, müssen die 6 Byte Eingangsdaten und 6 Byte Ausgangsdaten je Ventil als 3 Variablen vom Typ WORD vorgehalten werden.

Es wird außerdem empfohlen, für jedes Ventil eine eigene Instanz der verwendeten Bausteine anzulegen.



Die angegebene Startadresse für die Variablen der Eingangs- und Ausgangsdaten muss zum Ventil passen, mit dem kommuniziert werden soll (→ [Kommunikationsprotokoll des Motion Terminals VTEM](#)). Dabei müssen Module des CPX-Terminals berücksichtigt werden.

Die Beispiele unten beziehen sich auf ein Motion Terminal mit einem Busknoten CPX-FB33 als Master im CPX-Terminal.

Device overview								
Module	...	Rack	Slot	I address	Q address	Type	Article number	Firmware
fb33atvtem1of8		0	0	2042*		CPX Rev 20	TN 197330	V3.2.24
PN-IO Interface		0	0 X1	2041*		CPX		
VTEM 8 Valves		0	2	2...49	2...49	VTEM 8 valves	TN 8047502	

In der Standardvariablen-Tabelle können die Ein- und Ausgangsvariablen für die Bausteine angelegt werden:

Standard tag table							
	Name	Data type	Address	Retain	Visible in HMI	Accessible from ...	Comment
1	Valve0_IN0	Word	%IW2		☐	☐	Data from Valve 0
2	Valve0_IN1	Word	%IW4		☐	☐	Data from Valve 0
3	Valve0_IN2	Word	%IW6		☐	☐	Data from Valve 0
4	Valve0_OUT0	Word	%QW2		☐	☐	Data to Valve 0
5	Valve0_OUT1	Word	%QW4		☐	☐	Data to Valve 0
6	Valve0_OUT2	Word	%QW6		☐	☐	Data to Valve 0
7	Valve1_IN1	Word	%IW8		☐	☐	Data from Valve 1
8	Valve1_IN2	Word	%IW10		☐	☐	Data from Valve 1
9	Valve1_IN3	Word	%IW12		☐	☐	Data from Valve 1
10	Valve1_OUT1	Word	%QW8		☐	☐	Data to Valve 1
11	Valve1_OUT2	Word	%QW10		☐	☐	Data to Valve 1
12	Valve1_OUT3	Word	%QW12		☐	☐	Data to Valve 1
13	Valve2_IN1	Word	%IW14		☐	☐	Data from Valve 2
14	Valve2_IN2	Word	%IW16		☐	☐	Data from Valve 2
15	Valve2_IN3	Word	%IW18		☐	☐	Data from Valve 2
40	Valve6_OUT1	Word	%QW38		☐	☐	Data to Valve 6
41	Valve6_OUT2	Word	%QW40		☐	☐	Data to Valve 6
42	Valve6_OUT3	Word	%QW42		☐	☐	Data to Valve 6
43	Valve7_IN1	Word	%IW44		☐	☐	Data from Valve 7
44	Valve7_IN2	Word	%IW46		☐	☐	Data from Valve 7
45	Valve7_IN3	Word	%IW48		☐	☐	Data from Valve 7
46	Valve7_OUT1	Word	%QW44		☐	☐	Data to Valve 7
47	Valve7_OUT2	Word	%QW46		☐	☐	Data to Valve 7
48	Valve7_OUT3	Word	%QW48		☐	☐	Data to Valve 7

Beim Aufruf eines Bausteins werden die Variablen mit den Eingangsdaten den Bausteineingängen „wBusDataFromVTEM...“ übergeben. Die Daten am Ein-Ausgang des Bausteins „wBusDataToVTEM...“ werden den Variablen mit den Ausgangsdaten zugewiesen.

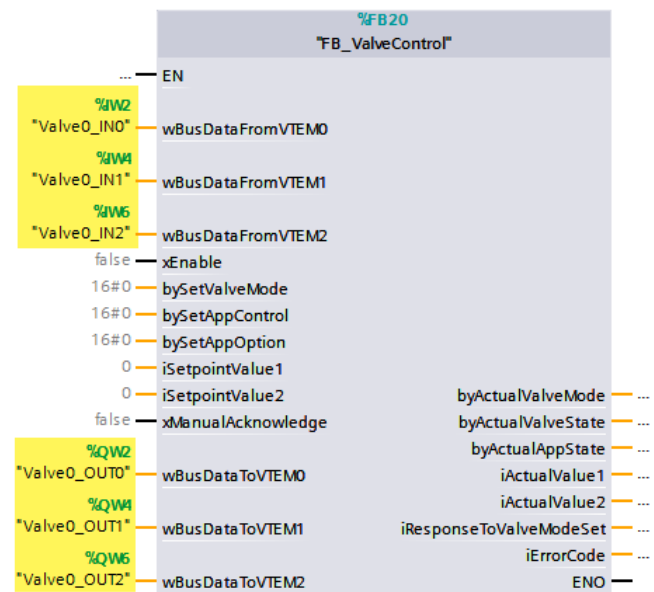
Beispiel in SCL

```

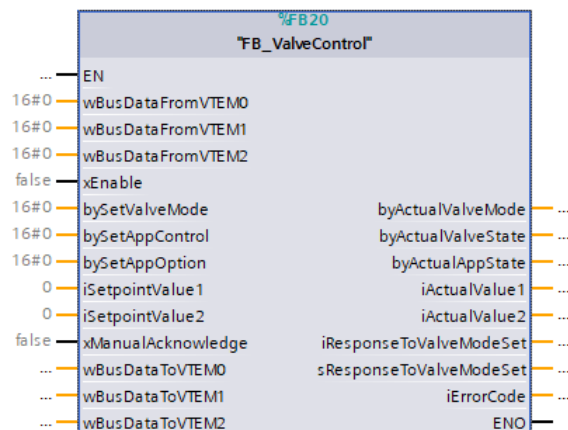
"FB_ValveControl_DB"(wBusDataFromVTEM0:="Valve0_IN0",
wBusDataFromVTEM1:="Valve0_IN1",
wBusDataFromVTEM2:="Valve0_IN2",
xEnable:=,
bySetValveMode:=,
bySetAppControl:=,
bySetAppOption:=,
iSetpointValue1:=,
iSetpointValue2:=,
xManualAcknowledge:=,
byActualValveMode=>,
byActualValveState=>,
byActualAppState=>,
iActualValue1=>,
iActualValue2=>,
iResponseToValveModeSet=>,
iErrorCode=>,
wBusDataToVTEM0:="Valve0_OUT0",
wBusDataToVTEM1:="Valve0_OUT1",
wBusDataToVTEM2:="Valve0_OUT2");

```

Beispiel in FUP

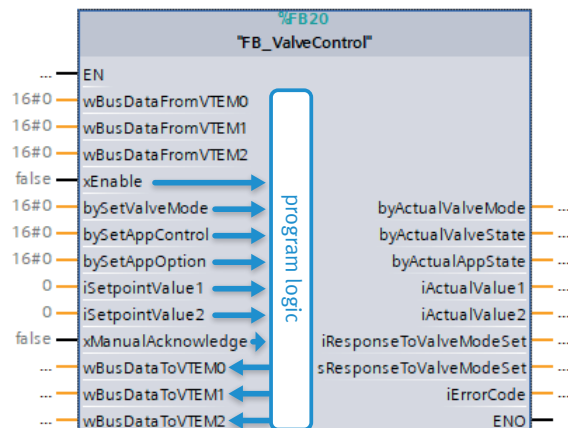


3 FB_ValveControl – Steuerung eines Ventils durch den Betrieb einer Motion App

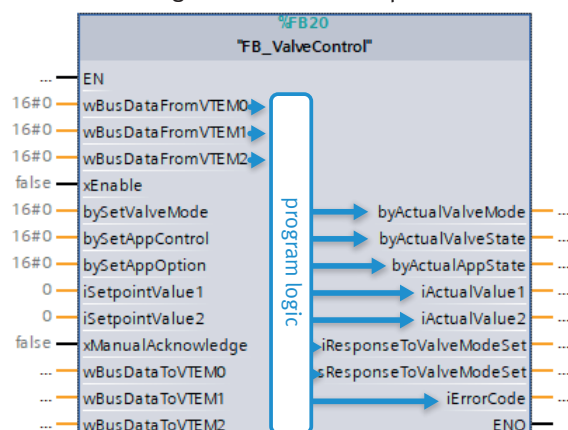


3.1 Funktionsbeschreibung

Der Baustein FB_ValveControl dient zur Bedienung und Steuerung der Motion Apps. Hierzu interpretiert der Baustein die an den Eingängen anliegenden Werte und generiert dementsprechend die Daten an den Eingängen, die zum Motion Terminal gesendet werden.



Parallel interpretiert der Baustein die vom Motion Terminal empfangenen Eingangsdaten eines Ventils, schlüsselt die darin enthaltenen Informationen auf und gibt sie an den entsprechenden Ausgängen aus.



Der Zustand der Bausteinausgänge erfolgt daher mindestens einen Zyklus nach der Änderung der Eingangswerte.

3.2 Ein- und Ausgänge

Die folgende Tabelle zeigt die Ein- und Ausgangsvariablen des Funktionsbausteins.

Bezeichnung	Variablentyp	Datentyp	Beschreibung
wBusDataFromVTEM0 wBusDataFromVTEM1 wBusDataFromVTEM2	VAR_INPUT	WORD	Die vom Motion Terminal empfangenen 6 Byte Eingangsdaten müssen an diesen Eingängen angelegt werden (→ E/A-Daten mit den Bausteinen verbinden).
wBusDataToVTEM0 wBusDataToVTEM1 wBusDataToVTEM2	VAR_IN_OUT	WORD	Die 6 Byte Ausgangsdaten, die zum Motion Terminal gesendet werden müssen, liegen an diesen Ausgängen (→ E/A-Daten mit den Bausteinen verbinden).
xEnable	VAR_INPUT	BOOL	Eingang muss TRUE sein, um den Baustein nutzen zu können. Bei einem Wechsel zu FALSE stoppt der Baustein alle auf dem Ventil laufenden Aktionen (Valve Mode = „valve inactive“). Nach erfolgreichem Beenden des Bausteins erhält der Ausgang iResponseToValveModeSet den Wert 1051 = „FB_not_enabled“.
bySetValveMode	VAR_INPUT	BYTE	Eingang zum Setzen des Ventilmodus. Zulässige Werte sind: 1 ... 59 (ID Motion App) 60 (Lernfahrt) 61 (Motion App beenden) 62 (inaktive Fehler quittieren), alternativ Eingang „xManualAcknowledge“ nutzen
bySetAppControl	VAR_INPUT	BYTE	Eingang zum Setzen der Motion-App-Steuerung. Die Bedeutung ist Motion-App-spezifisch.
bySetAppOption	VAR_INPUT	BYTE	Eingang zum Setzen der Motion-App-Einstellung. Die Bedeutung ist Motion-App-spezifisch. Für die Motion Apps #01 und #02 kann der Datentyp eValveType_MA_01_02 verwendet werden. Für Motion App #02 gibt es einen Baustein für das Zusammensetzen der App Option (FB_AppOptionGenerator_MA_02)
iSetPointValue1	VAR_INPUT	INT	Eingang für den Sollwert 1. Die Bedeutung ist Motion-App-spezifisch.
iSetPointValue2	VAR_INPUT	INT	Eingang für den Sollwert 2. Die Bedeutung ist Motion-App-spezifisch.
xManualAcknowledge	VAR_INPUT	BOOL	Mit einer steigenden Flanke an diesem Eingang können inaktive Fehler im Ventilstatus „not ready“ quittiert werden (entspricht Vorgabe „bySetValveMode := 62“ --> Ventilstatus wird „configurable“).
byActualValveMode	VAR_OUTPUT	BYTE	Rückgemeldeter Modus des Ventils
byActualValveState	VAR_OUTPUT	BYTE	Rückgemeldeter Status des Ventils
iActualAppState	VAR_OUTPUT	INT	Rückgemeldeter Status der Motion App
iActualValue1	VAR_OUTPUT	INT	Rückgemeldeter Istwert 1
iActualValue2	VAR_OUTPUT	INT	Rückgemeldeter Istwert 2
iResponseToValveModeSet	VAR_OUTPUT	INT	Rückmeldung zum gesendeten ValveMode (Eingang bySetValveMode). Bedeutung: → Der Datentyp eResponseToValveModeSet .
sResponseToValveModeSet	VAR_OUTPUT	STRING	Text zum Wert an iResponseToValveModeSet.

iErrorCode	VAR_OUTPUT	INT	Im Fehlerfall (Valve state = „Failure“ oder „Not Ready“) liegt an diesem Ausgang der Störungscode des zuletzt aufgetretenen Fehlers an.
------------	------------	-----	---

Die Verwendung der Eingänge hängt von der aktuell betriebenen Motion App ab.

3.2.1 Der Datentyp eResponseToValveModeSet

Der Datentyp wird als Enumeration verwendet und dient zur Ausgabe von Rückmeldungen zum gesendeten ValveMode (Wert am Eingang „SetValveMode“). Ein Teil der hinterlegten Rückgabewerte entspricht dem Wert des Ausgangs „AppState“ im Fall von fehlerhaften Ausgangsdaten (wBusDataToVTEM...). Zusätzlich gibt es bausteinspezifische Rückgabewerte (grau hinterlegt), die nicht über den Ausgang „AppState“ abgebildet werden (z. B. über den Status der Acknowledge-Funktion).

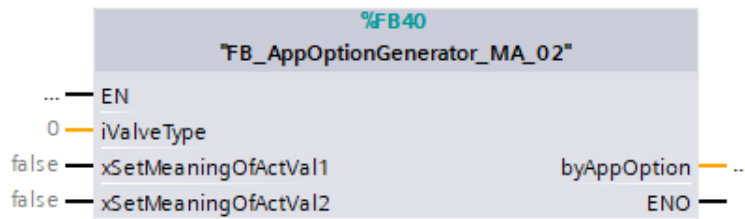
Bedeutung	Datentyp	Rückgabewert
invalid_SetValveMode	INT	1
invalid_SetAppControl	INT	2
invalid_SetAppOption	INT	3
invalid_SetPointValue1	INT	4
invalid_SetPointValue2	INT	5
invalid_configuration	INT	6
access_denied_WebConfig	INT	10
access_denied_saving_parmetrisation	INT	11
no_license_for_this_MA	INT	12
all_licenses_in_use	INT	13
invalid_license_file	INT	14
no_valve_detected	INT	15
valve_self_calibration_running	INT	16
no_TransferMode_with_this_FB	INT	1020
preparing_for_start	INT	1031
starting	INT	1032
MA_running	INT	1033
Transfermode_running	INT	1034
stopping	INT	1035
all_MAs_stopped	INT	1036
acknowledge_needed	INT	1037
failure	INT	1038
preparing_for_acknowledge	INT	1040
acknowledging	INT	1041
all_errors_acknowledged	INT	1042
failure_still_active	INT	1043
no_Communication	INT	1050
FB_not_enabled	INT	1051

3.2.2 Der Datentyp eValveType_MA_01_02

Der Datentyp wird als Enumeration verwendet und enthält die Werte für die Ventiltypen, die für Motion App #01 bzw. Motion App #02 ausgewählt werden können.

Bedeutung		Datentyp	Rückgabewert
Valve_4_3_G	4/3 G (normally closed)	INT	1
Valve_4_3_B	4/3 B (normally open)	INT	2
Valve_4_3_E	4/3 E (normally exhausted)	INT	3
Valve_2x3_2_O	2 × 3/2 O (normally open)	INT	4
Valve_2x3_2_G	2 × 3/2 G (normally closed)	INT	5
Valve_3_2_O_and_3_2_G	3/2 O + 3/2 G	INT	6
Valve_4_2_bistable	4/2 bistable	INT	7
Valve_2x2_2_G	2 × 2/2 G (normally closed)	INT	8
Valve_4_2_Mono	4/2 mono	INT	9
Valve_4_3_G_MA02	4/3 G (normally closed)	INT	14
Valve_2x3_3_G_MA02	2 × 3/3 G (normally closed)	INT	15

4 FB_AppOptionGenerator_MA_02 – Erzeugung des App-Option-Werts für MA#02



4.1 Funktionsbeschreibung

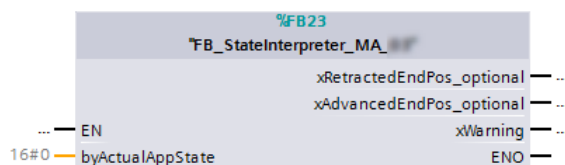
Der Baustein FB_AppOptionGenerator_MA_02 übernimmt die Erzeugung des App-Option-Werts für die Motion App #02 (Proportional-Wegeventil), der in Falle dieser App mehrere Informationen enthält. Die entsprechenden Informationen können an jeweils einem Eingang des Bausteins angelegt werden. Der Baustein generiert daraus den passenden App-Option-Wert am Ausgang.

4.2 Ein- und Ausgänge

Die folgende Tabelle zeigt die Ein- und Ausgangsvariablen des Funktionsbausteins.

Bezeichnung	Variablentyp	Datentyp	Beschreibung
iValveType	VAR_INPUT	INT	Wert für den Ventiltyp, der von der Motion App simuliert werden soll (→).
xSetMeaningOfActVal1	VAR_INPUT	BOOL	Einstellung für die Bedeutung von Istwert 1 0: Istwert zeigt Ventilstellung 1: Istwert zeigt Druck (relativ)
xSetMeaningOfActVal2	VAR_INPUT	BOOL	Einstellung für die Bedeutung von Istwert 2 0: Istwert zeigt Ventilstellung 1: Istwert zeigt Druck (relativ)
byAppOption	VAR_OUTPUT	BYTE	Byte zur Weitergabe an den Baustein FB_ValveControl

5 FB_StateInterpreter_... – Interpretation des App-Status

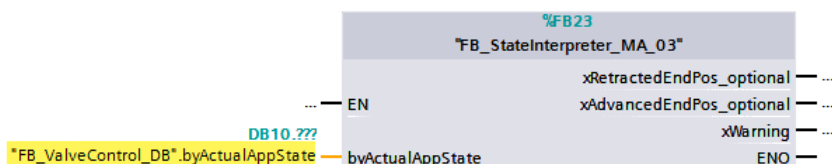


5.1 Funktionsbeschreibung

Die State-Interpreter-Bausteine stellen die Informationen zum Status einer laufenden Motion App oder der Lernfahrt aus den Eingangsdaten (Byte 1) an separaten Ausgängen zur Verfügung. Hierzu wird der Eingang „byActualAppState“ direkt mit dem Ausgang „byActualAppState“ des Bausteins „FB_ValveControl“ verbunden.

Beispiel

```
"FB_StateInterpreter_MA_03_DB_1" (byActualAppState:="FB_ValveControl_DB".byActualAppState,
xRetractedEndPos_optional=>,
xAdvancedEndPos_optional=>,
xWarning=>);
```



Die State-Interpreter-Bausteine dürfen nur verwendet werden, solange die passende Motion App bzw. die Lernfahrt auf dem betreffenden Ventil läuft (der Status des Ventils (Valve State) = 2 („running“)). Andernfalls sind die Werte an den Ausgängen nicht korrekt.

5.2 Ein- und Ausgänge

Die State-Interpreter-Bausteine sind alle nach demselben Prinzip aufgebaut (Aufteilung des App-State-Bereichs in die Einzelinformationen), die Ausgänge sind aber Motion-App-spezifisch. Die in der folgenden Tabelle beschriebenen Ein- und Ausgangsvariablen sind bei den meisten Bausteinen vorhanden.

Bezeichnung	Variablentyp	Datentyp	Beschreibung
byActualAppState	VAR_INPUT	BYTE	Eingang für den Bereich „App State“ (Byte 1) in den Eingangsdaten, die vom Motion Terminal empfangen werden. Der Baustein „FB_ValveControl“ gibt diesen Bereich an einem separaten Ausgang aus (byActualValveState).
xRetractedEndPos_optional	VAR_OUTPUT	BOOL	Ausgang entspricht dem Zustand am optional zugewiesenen Eingang des digitalen Eingangsmoduls für die eingefahrene Endposition (→ Beschreibung Motion Terminal VTEM, Funktion und Parametrierung). Die Zuweisung erfolgt mit Hilfe der Parametrierbausteine der Motion Apps (→ Beispiel zur Zuweisung der Eingänge).
xAdvancedEndPos_optional	VAR_OUTPUT	BOOL	Ausgang entspricht dem Zustand am optional zugewiesenen Eingang des digitalen Eingangsmoduls für die ausgefahrene Endposition (→ Beschreibung Motion Terminal VTEM, Funktion und Parametrierung). Die Zuweisung erfolgt mit Hilfe der Parametrierbausteine der Motion Apps (→ Beispiel zur Zuweisung der Eingänge).

Bezeichnung	Variablentyp	Datentyp	Beschreibung
xRetractedEndPos	VAR_OUTPUT	BOOL	Ausgang entspricht dem Zustand am zugewiesenen Eingang des Eingangsmoduls für die eingefahrene Endposition (→ Beschreibung Motion Terminal VTEM, Funktion und Parametrierung). Die Zuweisung erfolgt mit Hilfe der Parametrierbausteine der Motion Apps (→ Beispiel zur Zuweisung der Eingänge).
xAdvancedEndPos	VAR_OUTPUT	BOOL	Ausgang entspricht dem Zustand am zugewiesenen Eingang des Eingangsmoduls für die ausgefahrene Endposition (→ Beschreibung Motion Terminal VTEM, Funktion und Parametrierung). Die Zuweisung erfolgt mit Hilfe der Parametrierbausteine der Motion Apps (→ Beispiel zur Zuweisung der Eingänge).
xWarning	VAR_OUTPUT	BOOL	Ausgang ist TRUE, solange die laufende Motion App eine Warnung meldet.

Die folgenden Abschnitte beschreiben, falls vorhanden, die Motion-App-spezifischen Ausgangsvariablen der entsprechenden Bausteine:

5.2.1 Motion App #01 (Wegeventilfunktionen)

Bezeichnung	Variablentyp	Datentyp	Beschreibung
bySwitchingPositionAt2	VAR_OUTPUT	BYTE	Schaltstellung am Arbeitsluftanschluss (2): 0: Geschlossen 1: Belüftet 2: Entlüftet 3: Fehler
bySwitchingPositionAt4	VAR_OUTPUT	BYTE	Schaltstellung am Arbeitsluftanschluss (4): 0: Geschlossen 1: Belüftet 2: Entlüftet 3: Fehler

5.2.2 Motion App #02 (Proportional-Wegeventil)

Bezeichnung	Variablentyp	Datentyp	Beschreibung
xMeaningOfActVal1	VAR_OUTPUT	BOOL	Information zum Inhalt von Istwert 1 (Arbeitsluftanschluss (2)): 0: Ventilposition (1 digit = 0,01 %) 1: Druck (relativ) (1 digit = 1 mbar)
xMeaningOfActVal2	VAR_OUTPUT	BOOL	Information zum Inhalt von Istwert 2 (Arbeitsluftanschluss (4)): 0: Ventilposition (1 digit = 0,01 %) 1: Druck (relativ) (1 digit = 1 mbar)

5.2.3 Motion App #05 (Zu- und Abluftdrosselung)

Bezeichnung	Variablentyp	Datentyp	Beschreibung
xMeaningOfActVal1	VAR_OUTPUT	BOOL	Information zum Inhalt von Istwert 1: 0: Öffnungsgrad Entlüftung (1 digit = 0,01 %) 1: Druck (relativ) an Arbeitsluftanschluss (2) (1 digit = 1 mbar)
xMeaningOfActVal2	VAR_OUTPUT	BOOL	Information zum Inhalt von Istwert 2: 0: Öffnungsgrad Belüftung (1 digit = 0,01 %) 1: Druck (relativ) an Arbeitsluftanschluss (4) (1 digit = 1 mbar)

5.2.4 Motion App #10 (Durchflussregelung)

Bezeichnung	Variablentyp	Datentyp	Beschreibung
x2or4outOfSpec	VAR_OUTPUT	BOOL	Ausgang ist TRUE, wenn der Betrieb von (2) oder (4) außerhalb der Spezifikation erfolgt. Die Regelgenauigkeit ist möglicherweise eingeschränkt.
xFlow2FromMeasurement	VAR_OUTPUT	BOOL	Ausgang ist TRUE, wenn die Istwert-Angabe zum Durchfluss an (2) auf Messung basiert (höhere Genauigkeit).
xFlow4FromMeasurement	VAR_OUTPUT	BOOL	Ausgang ist TRUE, wenn die Istwert-Angabe zum Durchfluss an (4) auf Messung basiert (höhere Genauigkeit).
xRatioSetToMaxOverLimit2	VAR_OUTPUT	BOOL	Ausgang ist TRUE, wenn das Verhältnis von Solldurchfluss an (2) zu maximal stellbarem Durchfluss größer ist als die spezifizierte Grenze (95% + Wert des Parameters „Offset Info Solldurchfluss-Beschränkung für (2)“)
xRatioSetToMaxOverLimit4	VAR_OUTPUT	BOOL	Ausgang ist TRUE, wenn das Verhältnis von Solldurchfluss an (4) zu maximal stellbarem Durchfluss größer ist als die spezifizierte Grenze (95% + Wert des Parameters „Offset Info Solldurchfluss-Beschränkung für (4)“)

5.2.5 Motion App #11 (Soft Stop)

Bezeichnung	Variablentyp	Datentyp	Beschreibung
xContactPressure	VAR_OUTPUT	BOOL	Ausgang ist TRUE, wenn der intern durch die Motion App berechnete Anpressdruck in der Endlage erreicht wurde.
xPistonDetectRet	VAR_OUTPUT	BOOL	Ausgang ist TRUE, wenn sich der Kolben innerhalb des Erfassungsbereichs des zugewiesenen Teilhubsensors für die eingefahrene Endposition befindet. Die Zuweisung erfolgt mit Hilfe der Parametrierbausteine der Motion Apps (→ Beispiel zur Zuweisung der Eingänge).
xPistonDetectAdv	VAR_OUTPUT	BOOL	Ausgang ist TRUE, wenn sich der Kolben innerhalb des Erfassungsbereichs des zugewiesenen Teilhubsensors für die ausgefahrene Endposition befindet. Die Zuweisung erfolgt mit Hilfe der Parametrierbausteine der Motion Apps (→ Beispiel zur Zuweisung der Eingänge).

5.2.6 Motion App #12 (Diagnose Leakage)

Bezeichnung	Variablentyp	Datentyp	Beschreibung
byStateLeakageAt2	VAR_OUTPUT	BYTE	Status der Leakage am Arbeitsluftanschluss (2): 0: Defekt (rot) 1: Kritisch (orange) 2: Warnung (gelb) 3: Gut (grün)
byStateLeakageAt4	VAR_OUTPUT	BYTE	Status der Leakage am Arbeitsluftanschluss (4): 0: Defekt (rot) 1: Kritisch (orange) 2: Warnung (gelb) 3: Gut (grün)
byStateProcedure	VAR_OUTPUT	BYTE	Status der Diagnose Leakage: 0: Diagnose inaktiv 1: Diagnose aktiv 2: Diagnose abgebrochen 3: Diagnose teilweise abgeschlossen ((2) valide) 4: Diagnose teilweise abgeschlossen ((4) valide) 5: Diagnose abgeschlossen

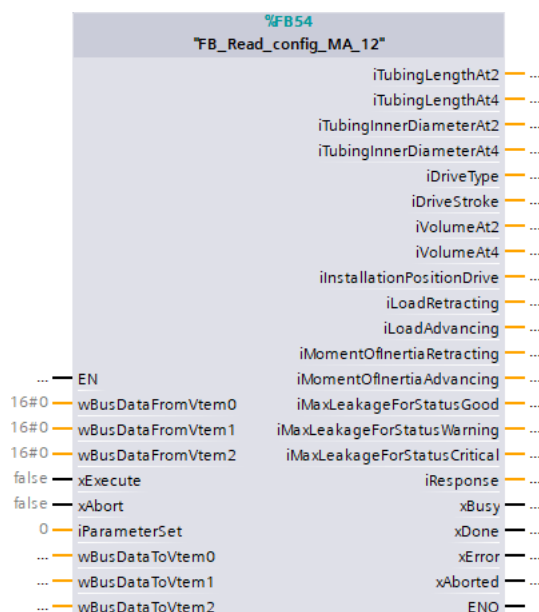
5.2.7 Motion App #33 (Positionieren)

Bezeichnung	Variablentyp	Datentyp	Beschreibung
xTargetPositionReached	VAR_OUTPUT	BOOL	Ausgang ist TRUE, wenn die absolute Abweichung der Istposition von der Sollposition den im Parameter „Toleranz für ‚Zielposition erreicht““ definierten Wert für mindestens die im Parameter „Beruhigungszeit bis ‚Zielposition erreicht““ definierte Dauer nicht überschreitet.
xOvershootInPlannedPath	VAR_OUTPUT	BOOL	Ausgang ist TRUE, wenn der geplante Bewegungspfad ein Überfahren der Zielposition enthält.
xControlledStop	VAR_OUTPUT	BOOL	Ausgang ist TRUE, wenn ein kontrollierter Stopp durch eine Positionsverletzung ausgelöst wurde (nur aktiv, wenn die Endlagenüberwachung nicht durch den entsprechenden Applikationsparameter deaktiviert wurde).

5.2.8 Lernfahrt

Bezeichnung	Variablentyp	Datentyp	Beschreibung
iTeachInRunState	VAR_OUTPUT	INT	Aktueller Zustand der Lernfahrt als Zahlenwert (→ Der Datentyp eResponseTeachInRun)
sTeachInRunState	VAR_OUTPUT	STRING	Aktueller Zustand der Lernfahrt als Text

6 FB_Read_config_MA... – Parametrierung einer Motion App lesen



6.1 Funktionsbeschreibung

Die Bausteine dienen zum Auslesen der Motion-App-spezifischen Parameterwerte und stellen jeden Wert an einem entsprechenden Ausgang bereit. Das Parameter-Set, dessen Werte gelesen werden sollen, wird über einen eigenen Eingang definiert. Die Übertragung der Parameterwerte auf die Steuerung erfolgt bei einer steigenden Flanke am Eingang „xExecute“. Das entsprechende Ventil wird in den Transfermodus versetzt und die Parameterwerte werden nacheinander übertragen. Der Status der Übertragung wird an den Ausgängen dargestellt.



Intern rufen die Bausteine den Baustein „FB_Upload_Multiple_Parameters“ auf und übergeben die Werte des Arrays des Datentyps „stTransferPackage“ an die Ausgänge.

Nach Abschluss der Übertragung (Ausgang „xDone“ = TRUE) bleiben die Werte an den Ausgängen stehen, bis der Baustein durch Setzen des Eingangs „xExecute“ auf FALSE zurückgesetzt wird. Der Transfermodus wird beendet und das entsprechende Ventil in den Modus „inaktiv“ versetzt (valve mode = inactive).

6.2 Ein- und Ausgänge

Die folgende Tabelle zeigt die Ein- und Ausgangsvariablen der Funktionsbausteine.

Bezeichnung	Variablentyp	Datentyp	Beschreibung
wBusDataFromVTEM0 wBusDataFromVTEM1 wBusDataFromVTEM2	VAR_INPUT	WORD	Die vom Motion Terminal empfangenen 6 Byte Eingangsdaten müssen an diesen Eingängen angelegt werden (→ E/A-Daten mit den Bausteinen verbinden).
wBusDataToVTEM0 wBusDataToVTEM1 wBusDataToVTEM2	VAR_IN_OUT	WORD	Die 6 Byte Ausgangsdaten, die zum Motion Terminal gesendet werden müssen, liegen an diesen Ausgängen (→ E/A-Daten mit den Bausteinen verbinden).
xExecute	VAR_INPUT	BOOL	Steigende Flanke am Eingang startet die Übertragung. Bei Wegnahme des „TRUE“-Signals wird der Baustein nach Abschluss der Übertragung zurückgesetzt und das Ventil in den Zustand „inaktiv“ versetzt (valve mode = 61).

Bezeichnung	Variablentyp	Datentyp	Beschreibung
xAbort	VAR_INPUT	BOOL	TRUE bricht die Übertragung ab. Nach Setzen von xAbort und xExecute auf „FALSE“ wird der Baustein nach Abschluss der Übertragung zurückgesetzt und das Ventil in den Zustand „inaktiv“ versetzt (valve mode = 61).
iParameterSet	VAR_INPUT	INT	Eingang zum Auswählen des Parameter-Sets, aus dem die Werte übernommen werden sollen.
<ParameterName>	VAR_OUTPUT	<Datentyp>	Ausgänge für die übertragenen Parameterwerte. Die Ausgänge sind Motion-App-spezifisch. Ihre Funktion ist in der Dokumentation der jeweiligen Motion App und in den Kommentaren der Ausgänge beschrieben.
iResponse	VAR_OUTPUT	eTransfer Response	Rückmeldung zum Status der Übertragung. Bedeutung: → Der Datentyp eTransferResponse .
xBusy	VAR_OUTPUT	BOOL	Ausgang ist TRUE, solange ein Vorgang im Baustein aktiv ist.
xDone	VAR_OUTPUT	BOOL	Ausgang wird TRUE, wenn die Übertragung ohne Fehler abgeschlossen werden konnte.
xEError	VAR_OUTPUT	BOOL	Ausgang wird TRUE, wenn während des Übertragungsprozesses ein Fehler auftritt.
xAborted	VAR_OUTPUT	BOOL	Ausgang ist TRUE, wenn die Übertragung durch den Eingang „xAAbort“ abgebrochen wurde. Ausgang wird zurückgesetzt, wenn xExecute und xAbort gleichzeitig FALSE sind.

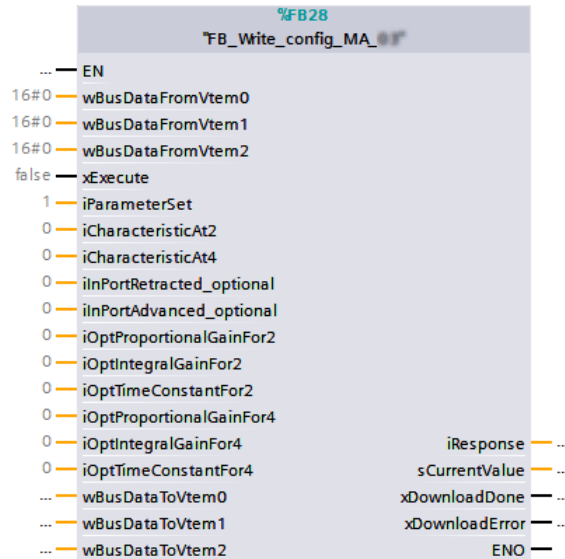
6.2.1 Der Datentyp eTransferResponse

Der Datentyp wird als Enumeration verwendet und dient zur Ausgabe von Rückmeldungen zum aktuellen Status der Übertragung.

Bedeutung	Datentyp	Rückgabewert
invalid_Channel	INT	1
invalid_TransferControl	INT	2
invalid_AddressedTarget	INT	3
invalid_Index	INT	4
invalid_TransferValue	INT	5
invalid_Combination	INT	6
access_denied_DataInUse	INT	7
no_data_empty_slot	INT	8
invalid_ParameterSet	INT	9
access_denied_WebConfig	INT	10
access_denied_saving	INT	11
no_valve_detected	INT	15
reading_transfermode	INT	20
ready_for_new_transfer	INT	21
preparing_transfer	INT	22
transferring	INT	23
transfer_successful	INT	24
FB_needs_rising_edge_xExecute	INT	25
Valve_Mode_is_not_61_inactive	INT	26

Bedeutung	Datentyp	Rückgabewert
Invalid_firmware_version	INT	27
communication_failed	INT	28
invalid_module_number	INT	29
resetting	INT	30
aborting	INT	31

7 FB_Write_config_MA... – Parametrierung einer Motion App schreiben



7.1 Funktionsbeschreibung

Die Bausteine zum Übertragen der Parameter, die für den Betrieb einer Motion App benötigt werden, stellen für jeden Parameter einen Eingang zur Verfügung, an den der gewünschte Parameterwert angelegt werden kann. Über einen weiteren Eingang kann das Parameter-Set festgelegt werden, in das die Werte geschrieben werden sollen.

Die Übertragung der Parameterwerte auf das Motion Terminal erfolgt bei einer steigenden Flanke am Eingang „xExecute“. Das entsprechende Ventil wird in den Transfermodus versetzt und die Parameterwerte werden nacheinander übertragen. Der Status der Übertragung wird an den Ausgängen dargestellt.

Beim Beenden des Bausteins durch eine fallende Flanke am Eingang „xExecute“ wird der Transfermodus beendet und das Ventil in den Modus „inaktiv“ versetzt (valve mode = inactive).



Intern rufen die Bausteine den Baustein „FB_Download_Multiple_Parameters“ auf und übergeben die Werte an den Eingängen als Array des Datentyps „stTransferPackage“.

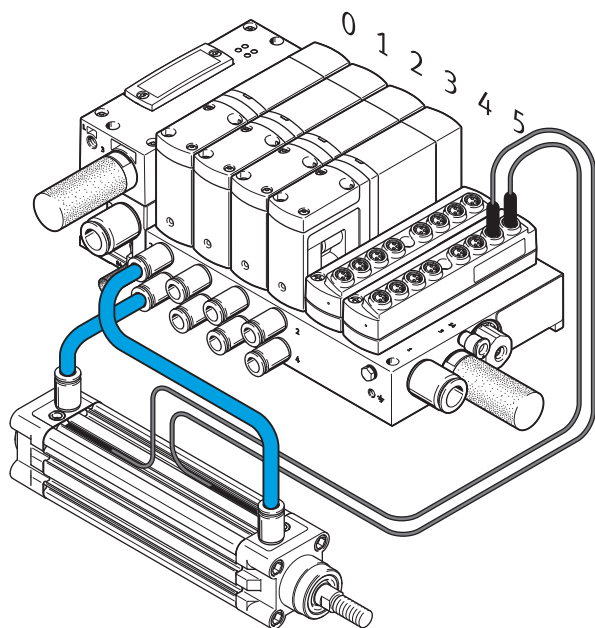
7.2 Ein- und Ausgänge

Die folgende Tabelle zeigt die Ein- und Ausgangsvariablen der Funktionsbausteine.

Bezeichnung	Variablentyp	Datentyp	Beschreibung
wBusDataFromVTEM0 wBusDataFromVTEM1 wBusDataFromVTEM2	VAR_INPUT	WORD	Die vom Motion Terminal empfangenen 6 Byte Eingangsdaten müssen an diesen Eingängen angelegt werden (→ E/A-Daten mit den Bausteinen verbinden).
wBusDataToVTEM0 wBusDataToVTEM1 wBusDataToVTEM2	VAR_IN_OUT	WORD	Die 6 Byte Ausgangsdaten, die zum Motion Terminal gesendet werden müssen, liegen an diesen Ausgängen (→ E/A-Daten mit den Bausteinen verbinden).
xExecute	VAR_INPUT	BOOL	Steigende Flanke am Eingang startet die Übertragung. Bei Wegnahme des „TRUE“-Signals wird der Transfermodus beendet und ein gegebenenfalls nicht abgeschlossener Transfer abgebrochen.
iParameterSet	VAR_INPUT	INT	Eingang zum Auswählen des Parameter-Sets, in das die Werte übernommen werden sollen.

Bezeichnung	Variablentyp	Datentyp	Beschreibung
<ParameterName>	VAR_INPUT	<Datentyp>	Eingänge für die Parameter-Werte, die übertragen werden sollen. Die Werte müssen für die Dauer der Übertragung konstant sein. Die Eingänge sind Motion-App-spezifisch. Ihre Funktion ist in der Dokumentation der jeweiligen Motion App und in den Kommentaren der Eingänge beschrieben.
iResponse	VAR_OUTPUT	eTransfer Response	Rückmeldung zum Status der Übertragung. Bedeutung: → Der Datentyp eTransferResponse .
sCurrentValue	VAR_OUTPUT	STRING	Information, welcher Wert aktuell übertragen wird. Diese Information kann bei Abbruch der Übertragung zur Fehleridentifikation genutzt werden.
xDownloadDone	VAR_OUTPUT	BOOL	Ausgang wird TRUE, wenn die Übertragung ohne Fehler abgeschlossen werden konnte.
xDownloadError	VAR_OUTPUT	BOOL	Ausgang wird TRUE, wenn während des Übertragungsprozesses ein Fehler auftritt.

7.2.1 Beispiel zur Zuweisung der Eingänge



Für dieses Beispiel soll auf dem Ventil auf Steckplatz 0 die Motion App 6 (ECO-Fahrt) parametrierung werden.

An den Arbeitsluftanschlüssen des Steckplatzes 0 ist ein Kolbenstangenzyylinder angeschlossen.

Die Näherungsschalter SMT-8M-A-PS-24V-... an den beiden Endlagen des Kolbenstangenzyinders sind an den Eingängen 0 und 1 des digitalen Eingangsmoduls angeschlossen.

Für die Konfiguration der Eingangsmodule (angeschlossener Sensortyp und Ausrichtung des Sensors bei analogen Eingangsmodulen) können die Funktionsbausteine FB_Write_config_AI_Module und FB_Write_config_DI_Module verwendet werden (→ [VTEM_AOI_Write_config_DI/AI_Module – Parametrierung der Eingangsmodule schreiben](#)).

Die Zuordnung der Eingänge zum Ventil muss über die Systemparameter 70 und 71 des Ventils parametrierung werden (→ Beschreibungen der Motion Apps).

Bei Verwendung der Funktionsbausteine „FB_Write_config_MA...“ werden hierfür die Eingänge „ilnPortRetracted“ bzw. „ilnPortRetracted_optional“ und „ilnPortAdvanced“ bzw. „ilnPortAdvanced_optional“ genutzt. (Die Eingänge „..._optional“ sind Bestandteil der Funktionsbausteine für Motion Apps, die für den Betrieb keine Sensoren benötigen.)

Verwendung des Funktionsbausteins FB_Write_config_MA_06

Die Werte für die Parameter errechnen sich aus der Position des Eingangsmoduls (Steckplatz), multipliziert mit dem Faktor 10, addiert zur Nummer des Eingangs auf diesem Eingangsmodul (0 ... 7).

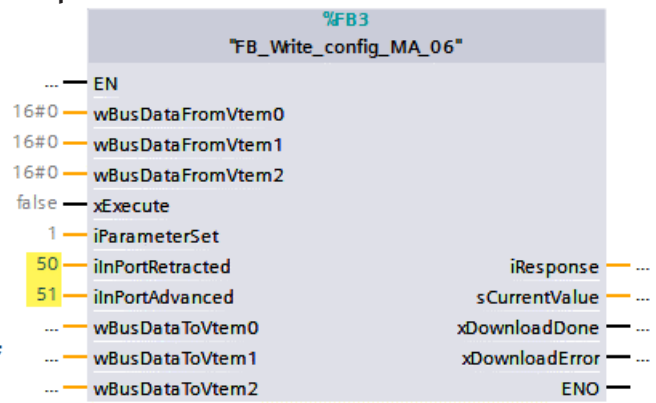
ilnPortRetracted: Nummer des Steckplatzes des digitalen Eingangsmoduls (5) × 10 + Nummer des Eingangs, an dem der Näherungsschalter für die Position „Eingefahren“ angeschlossen ist (0).
 $5 \times 10 + 0 = 50$

ilnPortAdvanced: Nummer des Steckplatzes des digitalen Eingangsmoduls (5) × 10 + Nummer des Eingangs, an dem der Näherungsschalter für die Position „Ausgefahren“ angeschlossen ist (1).
 $5 \times 10 + 1 = 51$

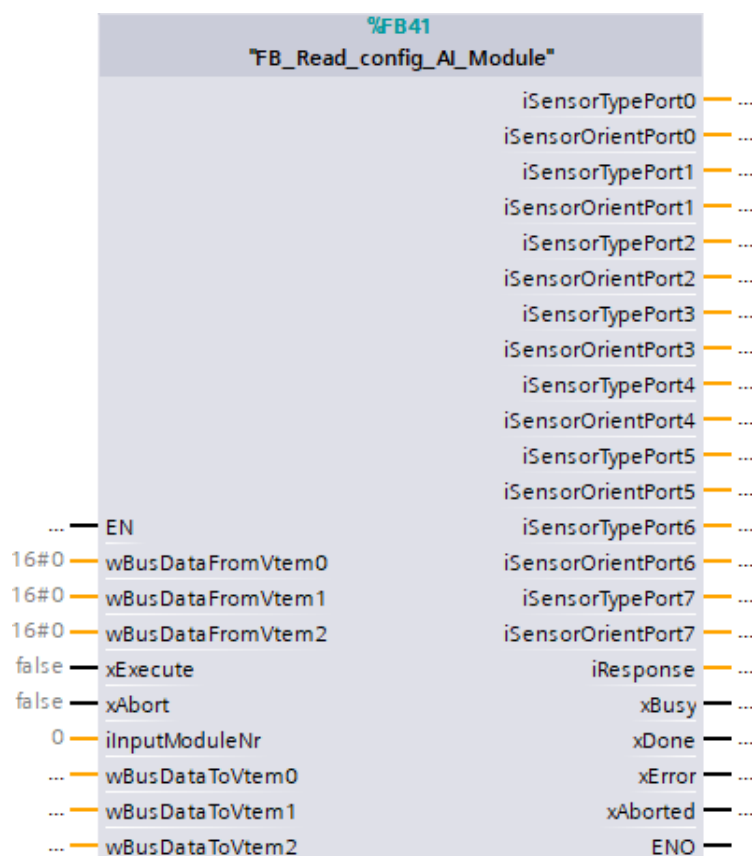
Beispiel in SCL

```
"FB_Write_config_MA_06_DB"(wBusDataFromVtem0:=16#0,
wBusDataFromVtem1:=16#0,
wBusDataFromVtem2:=16#0,
xExecute:=false,
iParameterSet:=1,
iInPortRetracted:=50,
iInPortAdvanced:=51,
iResponse=>_int_out_,
sCurrentValue=>_string_out_,
xDownloadDone=>_bool_out_,
xDownloadError=>_bool_out_,
wBusDataToVtem0:=_word_inout_,
wBusDataToVtem1:=_word_inout_,
wBusDataToVtem2:=_word_inout_);
```

Beispiel in FUP



8 FB_Read_config_DI/AI_Module – Parametrierung der Eingangsmodule lesen



8.1 Funktionsbeschreibung

Die Bausteine übertragen die Parameterwerte der digitalen und analogen Eingangsmodule und stellen jeden Wert an einem eigenen Ausgang zur Verfügung. Über einen Eingang kann die Nummer des Eingangsmoduls festgelegt werden, dessen Parameter gelesen werden sollen (beginnend von links beim ersten Eingangsmodul mit der Nummer 1).

Die Übertragung der Parameterwerte auf die Steuerung startet, wenn der Eingang „xExecute“ TRUE ist. Das entsprechende Ventil wird in den Transfermodus versetzt und die Parameterwerte werden nacheinander übertragen. Der Status der Übertragung wird an den Ausgängen des Bausteins dargestellt.

Das Ventil wird in den Modus „inaktiv“ (valve mode = 61 (inactive)) versetzt bei folgenden Ereignissen:

Ereignis	iResponse	xBusy	xDone	xError	xAborted
Erfolgreiche Übertragung aller Parameter	24 (transfer_successful)	FALSE	TRUE	FALSE	FALSE
Fehler bei der Übertragung der Parameter	Fehlerursache (→ Der Datentyp eTransferResponse)	FALSE	FALSE	TRUE	FALSE
Abbruch der Übertragung durch „xAbort“ = TRUE	31 (aborting)	FALSE	FALSE	FALSE	TRUE

Nach einem dieser Ereignisse wird der Baustein zurückgesetzt, sobald an den Eingängen „xAbort“ und „xExecute“ ein FALSE-Signal anliegt.



Intern rufen die Bausteine den Baustein „FB_Upload_Multiple_Parameters“ auf und übergeben die Werte des Arrays des Datentyps „stTransferPackage“ an die Ausgänge.

8.2 Ein- und Ausgänge

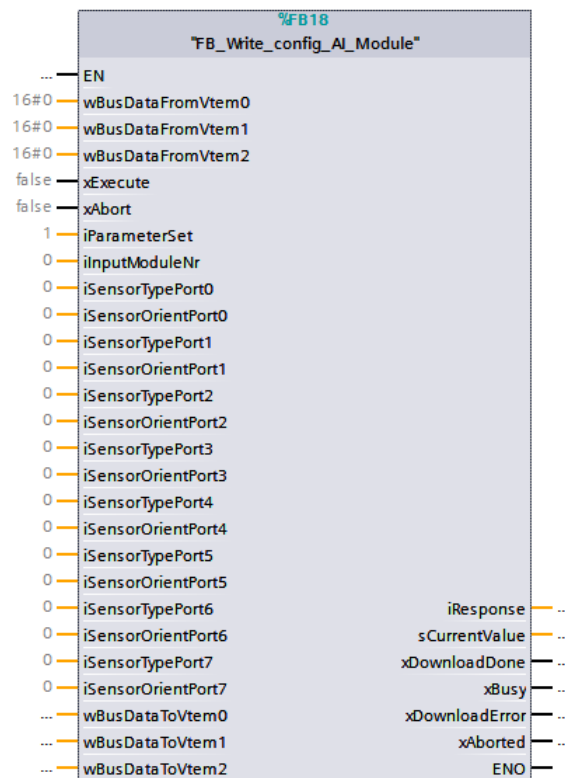
Die folgende Tabelle zeigt die Ein- und Ausgangsvariablen der Funktionsbausteine.

Bezeichnung	Variablentyp	Datentyp	Beschreibung
wBusDataFromVTEM0 wBusDataFromVTEM1 wBusDataFromVTEM2	VAR_INPUT	WORD	Die vom Motion Terminal empfangenen 6 Byte Eingangsdaten müssen an diesen Eingängen angelegt werden (→ E/A-Daten mit den Bausteinen verbinden).
wBusDataToVTEM0 wBusDataToVTEM1 wBusDataToVTEM2	VAR_IN_OUT	WORD	Die 6 Byte Ausgangsdaten, die zum Motion Terminal gesendet werden müssen, liegen an diesen Ausgängen (→ E/A-Daten mit den Bausteinen verbinden).
xExecute	VAR_INPUT	BOOL	Steigende Flanke am Eingang startet die Übertragung. Bei Wegnahme des „TRUE“-Signals wird der Baustein nach Abschluss der Übertragung zurückgesetzt und das Ventil in den Zustand „inaktiv“ versetzt (valve mode = 61).
xAbort	VAR_INPUT	BOOL	TRUE bricht die Übertragung ab. Nach Setzen von xAbort und xExecute auf „FALSE“ wird der Baustein nach Abschluss der Übertragung zurückgesetzt und das Ventil in den Zustand „inaktiv“ versetzt (valve mode = 61).
iInputModuleNr	VAR_INPUT	INT	Nummer des Eingangsmoduls, dessen Parameter gelesen werden sollen, beginnend mit 1 beim ersten Eingangsmodul von links.
iSensorTypePort0 ... 7	VAR_OUTPUT	INT	ID des Sensortyps, der am jeweiligen Eingang angeschlossen ist. 0: Kein Sensor 1001: SMT-8M-A-PS-24V... (digital) 2001: SDAP-MHS-M50 (Positionssensor) 2002: SDAP-MHS-M100 (Positionssensor) 2003: SDAP-MHS-M160 (Positionssensor) 2901: Benutzerdefinierter Positionssensor 1 ¹⁾ 2902: Benutzerdefinierter Positionssensor 2 ¹⁾ 2903: Benutzerdefinierter Positionssensor 3 ¹⁾ 3001: SFAB-50U-...-2SA-... (Durchflusssensor) 3002: SFAB-200U-...-2SA-... (Durchflusssensor) 3003: SFAB-600U-...-2SA-... (Durchflusssensor) 3004: SFAB-1000U-...-2SA-... (Durchflusssensor) 3101: SFAH-50U-...-PNVBA-... (Durchflusssensor) 3102: SFAH-100U-...-PNVBA-... (Durchflusssensor) 3103: SFAH-200U-...-PNVBA-... (Durchflusssensor) 3901: Benutzerdefinierter Durchflusssensor 1 ¹⁾ 3902: Benutzerdefinierter Durchflusssensor 2 ¹⁾ 3903: Benutzerdefinierter Durchflusssensor 3 ¹⁾
iSensorOrientPort0 ... 7 ²⁾	VAR_OUTPUT	INT	Signalverhalten des Sensors, der am jeweiligen Eingang des analogen Eingangsmoduls angeschlossen ist. Positionstransmitter 0: Ausgangssignal steigt beim Ausfahren ³⁾ 1: Ausgangssignal fällt beim Ausfahren ³⁾ Durchflusssensor 0: Positive Durchflusswerte („pustend“) 1: Negative Durchflusswerte („saugend“)

Bezeichnung	Variablentyp	Datentyp	Beschreibung
iResponse	VAR_OUTPUT	eTransfer Response	Rückmeldung zum Status der Übertragung. Bedeutung: → Der Datentyp eTransferResponse .
xBusy	VAR_OUTPUT	BOOL	Ausgang ist TRUE, solange ein Vorgang im Baustein aktiv ist.
xDone	VAR_OUTPUT	BOOL	Ausgang wird TRUE, wenn die Übertragung ohne Fehler abgeschlossen werden konnte.
xError	VAR_OUTPUT	BOOL	Ausgang wird TRUE, wenn während des Übertragungsprozesses ein Fehler auftritt.
xAborted	VAR_OUTPUT	BOOL	Ausgang ist TRUE, wenn die Übertragung durch den Eingang „xAbort“ abgebrochen wurde. Ausgang wird zurückgesetzt, wenn xExecute und xAbort gleichzeitig FALSE sind.

- 1) Benutzerdefinierte Sensoren müssen über den Transfermodus definiert werden (→ [Dokumentationen zum Motion Terminal VTEM](#)).
- 2) Nur für analoge Eingangsmodule
- 3) Ausfahren: Arbeitsluftanschluss (4) belüftet, Arbeitsluftanschluss (2) entlüftet

9 FB_Write_config_DI/AI_Module – Parametrierung der Eingangsmodule schreiben



9.1 Funktionsbeschreibung

Die Bausteine übertragen die Parameter für die digitalen und analogen Eingangsmodule und stellen für jeden Parameter einen Eingang zur Verfügung, an den der gewünschte Parameterwert angelegt werden kann. Über einen weiteren Eingang kann die Nummer des zu parametrierenden Eingangsmoduls festgelegt werden (beginnend von links beim ersten Eingangsmodul mit der Nummer 1).

Die Übertragung der Parameterwerte auf das Motion Terminal startet, wenn der Eingang „xExecute“ TRUE ist. Das entsprechende Ventil wird in den Transfermodus versetzt und die Parameterwerte werden nacheinander übertragen. Der Status der Übertragung wird an den Ausgängen des Bausteins dargestellt.

Das Ventil wird in den Modus „inaktiv“ (valve mode = 61 (inactive) versetzt bei folgenden Ereignissen:

Ereignis	iResponse	sCurrentValue	xBusy	xDownload Done	xDownload Error	xAborted
Erfolgreiche Übertragung aller Parameter	24 (transfer_successful)	„AllParametersWritten“	FALSE	TRUE	FALSE	FALSE
Fehler bei der Übertragung der Parameter	Fehlerursache (→ Der Datentyp eTransferResponse)	Name des Eingangswerts, bei dessen Übertragung der Fehler auftrat.	FALSE	FALSE	TRUE	FALSE
Abbruch der Übertragung durch „xAbort“ = TRUE	Wert bei Abbruch (→ Der Datentyp eTransferResponse)	Name des Eingangswerts, bei dessen Übertragung der Abbruch stattfand.	FALSE	FALSE	FALSE	TRUE

Nach einem dieser Ereignisse wird der Baustein zurückgesetzt, sobald an den Eingängen „xAbort“ und „xExecute“ ein FALSE-Signal anliegt.



Intern rufen die Bausteine den Baustein „FB_Download_Multiple_Parameters“ auf und übergeben die Werte an den Eingängen als Array des Datentyps „stTransferPackage“.

9.2 Ein- und Ausgänge

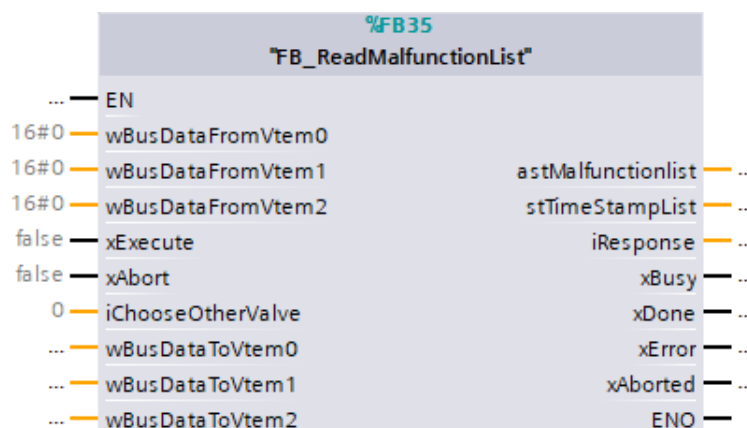
Die folgende Tabelle zeigt die Ein- und Ausgangsvariablen der Funktionsbausteine.

Bezeichnung	Variablentyp	Datentyp	Beschreibung
wBusDataFromVTEM0 wBusDataFromVTEM1 wBusDataFromVTEM2	VAR_INPUT	WORD	Die vom Motion Terminal empfangenen 6 Byte Eingangsdaten müssen an diesen Eingängen angelegt werden (→ E/A-Daten mit den Bausteinen verbinden).
wBusDataToVTEM0 wBusDataToVTEM1 wBusDataToVTEM2	VAR_IN_OUT	WORD	Die 6 Byte Ausgangsdaten, die zum Motion Terminal gesendet werden müssen, liegen an diesen Ausgängen (→ E/A-Daten mit den Bausteinen verbinden).
xExecute	VAR_INPUT	BOOL	TRUE-Signal am Eingang startet die Übertragung. Bei Wegnahme des TRUE-Signals wird der Transfermodus beendet nachdem der Transfer abgeschlossen wurde.
xAbort	VAR_INPUT	BOOL	TRUE-Signal am Eingang bricht den Vorgang im Baustein ab. Der Transfermodus wird gegebenenfalls beendet und das Ventil in den Zustand „inaktiv“ (valve mode = 61) versetzt.
iInputModuleNr	VAR_INPUT	INT	Nummer des zu parametrierenden Eingangsmoduls, beginnend mit 1 beim ersten Eingangsmodul von links.
iSensorTypePort0 ... 7	VAR_INPUT	INT	ID des Sensortyps, der am jeweiligen Eingang angeschlossen ist. 0: Kein Sensor 1001: SMT-8M-A-PS-24V-... (digital) 2001: SDAP-MHS-M50 (Positionssensor) 2002: SDAP-MHS-M100 (Positionssensor) 2003: SDAP-MHS-M160 (Positionssensor) 2901: Benutzerdefinierter Positionssensor 1 ¹⁾ 2902: Benutzerdefinierter Positionssensor 2 ¹⁾ 2903: Benutzerdefinierter Positionssensor 3 ¹⁾ 3001: SFAB-50U-...-2SA-... (Durchflusssensor) 3002: SFAB-200U-...-2SA-... (Durchflusssensor) 3003: SFAB-600U-...-2SA-... (Durchflusssensor) 3004: SFAB-1000U-...-2SA-... (Durchflusssensor) 3101: SFAH-50U-...-PNVBA-... (Durchflusssensor) 3102: SFAH-100U-...-PNVBA-... (Durchflusssensor) 3103: SFAH-200U-...-PNVBA-... (Durchflusssensor) 3901: Benutzerdefinierter Durchflusssensor 1 ¹⁾ 3902: Benutzerdefinierter Durchflusssensor 2 ¹⁾ 3903: Benutzerdefinierter Durchflusssensor 3 ¹⁾
iSensorOrientPort0 ... 7 ²⁾	VAR_INPUT	INT	Signalverhalten des Sensors, der am jeweiligen Eingang des analogen Eingangsmoduls angeschlossen ist. Positionstransmitter 0: Ausgangssignal steigt beim Ausfahren ³⁾ 1: Ausgangssignal fällt beim Ausfahren ³⁾ Durchflusssensor 0: Positive Durchflusswerte („pustend“) 1: Negative Durchflusswerte („saugend“)
iResponse	VAR_OUTPUT	eTransfer Response	Rückmeldung zum Status der Übertragung. Bedeutung: → Der Datentyp eTransferResponse .

Bezeichnung	Variablentyp	Datentyp	Beschreibung
sCurrentValue	VAR_OUTPUT	STRING	Information, welcher Wert aktuell übertragen wird. Diese Information kann bei Abbruch der Übertragung zur Fehleridentifikation genutzt werden.
xBusy	VAR_OUTPUT	BOOL	Ausgang ist TRUE, solange ein Vorgang im Baustein aktiv ist.
xDownloadDone	VAR_OUTPUT	BOOL	Ausgang wird TRUE, wenn die Übertragung ohne Fehler abgeschlossen werden konnte.
xDownloadError	VAR_OUTPUT	BOOL	Ausgang wird TRUE, wenn während des Übertragungsprozesses ein Fehler auftritt.
xAborted	VAR_OUTPUT	BOOL	Ausgang ist TRUE, wenn die Übertragung durch den Eingang „xAbort“ abgebrochen wurde. Ausgang wird zurückgesetzt, wenn xExecute und xAbort gleichzeitig FALSE sind.

- 1) Benutzerdefinierte Sensoren müssen über den Transfermodus definiert werden (→ [Dokumentationen zum Motion Terminal VTEM](#)).
- 2) Nur für analoge Eingangsmodule
- 3) Ausfahren: Arbeitsluftanschluss (4) belüftet, Arbeitsluftanschluss (2) entlüftet

10 FB_ReadMalfunctionList – Störungsliste auslesen



10.1 Funktionsbeschreibung

Der Baustein „FB_ReadMalfunctionList“ liest den Inhalt der Störungsliste eines Ventils automatisch aus und übergibt diesen am Ausgang „astMalfuctionlist“.

Der Ausgang „astMalfuctionlist“ übergibt ein Array mit 40 Elementen vom Typ „stMalfuction“. Jedes Element enthält damit einen kompletten Eintrag der Störungsliste bestehend aus „Code“, „Subcode“, „Classification“ und Zeitstempel („hours“, „minutes“, „seconds“ und „milliseconds“). Am Ausgang „stTimeStampList“ wird ein Zeitstempel im Format „stTimeStamp“ ausgegeben, der den Zeitpunkt beschreibt, zu dem die Störungsliste ausgelesen wurde. Durch Vergleichen dieses Zeitpunkts mit den Zeitpunkten der Störungslisteneinträge kann deren Ereigniszeitpunkt bestimmt werden. Die Zeitpunkte beziehen sich auf die Betriebsdauer des Motion Terminals seit dem Einschalten.

Über den Eingang „iChooseOtherValve“ kann definiert werden, von welchem Ventil die Störungsliste gelesen werden soll. So können hierfür auch die Prozessdaten eines beliebigen anderen Steckplatzes genutzt werden.

10.2 Ein- und Ausgänge

Die folgende Tabelle zeigt die Ein- und Ausgangsvariablen des Funktionsbausteins.

Bezeichnung	Variablentyp	Datentyp	Beschreibung
wBusDataFromVTEM0 wBusDataFromVTEM1 wBusDataFromVTEM2	VAR_INPUT	WORD	Die vom Motion Terminal empfangenen 6 Byte Eingangsdaten müssen an diesen Eingängen angelegt werden (→ E/A-Daten mit den Bausteinen verbinden).
wBusDataToVTEM0 wBusDataToVTEM1 wBusDataToVTEM2	VAR_IN_OUT	WORD	Die 6 Byte Ausgangsdaten, die zum Motion Terminal gesendet werden müssen, liegen an diesen Ausgängen (→ E/A-Daten mit den Bausteinen verbinden).
xExecute	VAR_INPUT	BOOL	Steigende Flanke am Eingang startet die Übertragung. Bei Wegnahme des TRUE-Signals wird der Transfermodus beendet und der Baustein zurückgesetzt, sobald der Übertragungsprozess erfolgreich (iResponse = „done“ oder „no_entries_in_malfuncrion_list“) oder mit einem Fehler (xError = TRUE) beendet wurde.
xAbort	VAR_INPUT	BOOL	TRUE-Signal am Eingang bricht den Vorgang im Baustein ab. Der Transfermodus wird gegebenenfalls beendet und das Ventil in den Zustand „inaktiv“ (valve mode = 61) versetzt.

Bezeichnung	Variablentyp	Datentyp	Beschreibung
iChooseOtherValve	VAR_INPUT	INT	Auswahl des Ventils, dessen Störungsliste gelesen werden soll. 0: Störungsliste des Ventils auslesen, dessen Prozessdaten der Baustein verwendet. 100 ... 107: Störungsliste des Ventils auf dem Steckplatz 0 (100) ... 7 (107) auslesen.
astMalfunctionlist	VAR_OUTPUT	ARRAY [1..40] OF stMalfunction	Inhalt der Störungsliste (→ Der Datentyp stMalfunction). Der Inhalt des Arrays ist vollständig, sobald xDone = TRUE ist. Der Ausgang wird zurückgesetzt, sobald xAbort = FALSE und xEnable = FALSE sind.
stTimeStampList	VAR_OUTPUT	stTimeStamp	Erstellungszeitpunkt der Störungsliste (→ Der Datentyp stTimeStamp)
iResponse	VAR_OUTPUT	INT	Rückmeldung zum Status des Bausteins oder der Übertragung. Bedeutung: → Der Datentyp eResponseReadMalfunctionList .
xBusy	VAR_OUTPUT	BOOL	Ausgang ist TRUE, solange ein Vorgang im Baustein aktiv ist.
xDownloadDone	VAR_OUTPUT	BOOL	Ausgang wird TRUE, wenn die Übertragung ohne Fehler abgeschlossen werden konnte (auch bei leerer Störungsliste).
xDownloadError	VAR_OUTPUT	BOOL	Ausgang wird TRUE, wenn während des Übertragungsprozesses ein Fehler auftritt.
xAborted	VAR_OUTPUT	BOOL	Ausgang ist TRUE, wenn die Übertragung durch den Eingang „xAbort“ abgebrochen wurde. Ausgang wird zurückgesetzt, wenn xExecute und xAbort gleichzeitig FALSE sind.

10.2.1 Der Datentyp eResponseReadMalfunctionList

Der Datentyp wird als Enumeration verwendet und dient zur Ausgabe von bausteinspezifische Rückmeldungen. Sie enthält zusätzlich die Werte von „AppState“ im Fall von fehlerhaften Ausgangsdaten mit einem Versatz von +100 und die Rückgabewerte des Transfermodus aus der Enumeration eTransferResponse mit einem Versatz von +200.

Bedeutung	Datentyp	Rückgabewert
idle	INT	0
setting_valve_to_inactive	INT	1
getting_malfunction_count	INT	2
transferring	INT	3
done	INT	4
no_entries_in_malfunction_list	INT	5
error_while_transmitting_data_to_vtem	INT	6
aborting	INT	7
resetting	INT	8
FB_needs_rising_edge_xExecute	INT	10
reserved	INT	100
invalid_SetValveMode	INT	101
invalid_SetAppControl	INT	102
invalid_SetAppOption	INT	103

Bedeutung	Datentyp	Rückgabewert
invalid_SetPointValue1	INT	104
invalid_SetPointValue2	INT	105
invalid_configuration	INT	106
access_denied_WebConfig	INT	110
access_denied_saving_parameterisation	INT	111
no_license_for_this_MA	INT	112
all_licenses_in_use	INT	113
invalid_license_file	INT	114
no_valve_detected	INT	115
valve_self_calibration_running	INT	116
invalid_Channel	INT	201
invalid_TransferControl	INT	202
invalid_iChooseOtherValve	INT	203
invalid_Index	INT	204
invalid_TransferValue	INT	205
invalid_Combination	INT	206
access_denied_DataInUse	INT	207
no_data_empty_slot	INT	208

10.2.2 Der Datentyp stMalfunction

Der Datentyp bildet den Inhalt eines Eintrags in der Störungsliste ab, bestehend aus Störungscode, Störungs-Subcode, Einstufung der Störung und Zeitstempel des Eintrags.

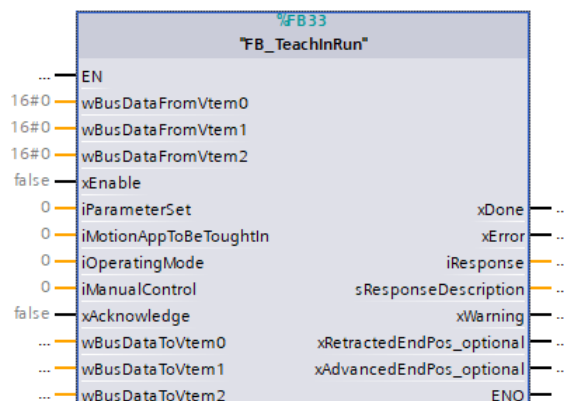
Bezeichnung	Datentyp	Beschreibung
iMalfunctionCode	INT	Code des Störungslisteneintrags
iMalfunctionSubcode	INT	Subcode des Störungslisteneintrags
iMalfunctionClassification	INT	Einstufung des Störungslisteneintrags: 1: aktiver Fehler 2: inaktiver Fehler 3: Warnung
diHours	DINT	Zeitstempel (Stunden) des Störungslisteneintrags
iMinutes	INT	Zeitstempel (Minuten) des Störungslisteneintrags
iSeconds	INT	Zeitstempel (Sekunden) des Störungslisteneintrags
iMilliseconds	INT	Zeitstempel (Millisekunden) des Störungslisteneintrags

10.2.3 Der Datentyp stTimeStamp

Der Datentyp bildet den Inhalt eines Zeitstempels ab, bestehend aus Stunden, Minuten, Sekunden und Millisekunden.

Bezeichnung	Datentyp	Beschreibung
diHours	DINT	Zeitstempel (Stunden)
iMinutes	INT	Zeitstempel (Minuten)
iSeconds	INT	Zeitstempel (Sekunden)
iMilliseconds	INT	Zeitstempel (Millisekunden)

11 FB_TeachInRun – Lernfahrt parametrieren und ausführen



11.1 Funktionsbeschreibung

Der Baustein FB_TeachInRun dient dem Erstellen von Lerndaten mithilfe der Lernfahrt (valve mode = 60).

Dabei kann über entsprechende Eingänge des Bausteins bestimmt werden, für welche Motion App die Lerndaten erstellt werden sollen (iMotionAppToBeToughtIn) und in welchem Parameter-Set sie abgelegt werden sollen (iParameterSet).

Auch der Betriebsmodus (iOperatingMode) und, bei manuellem Betrieb, die Steuerung (iManualControl) der Lernfahrt können über entsprechende Eingänge des Bausteins vorgegeben werden.

Zusätzlich bietet der Baustein einen Eingang, um inaktive Fehler zu quittieren (xAcknowledge). Das Quittieren erfolgt bei einer steigenden Flanke am Eingang.

Auf der Ausgangsseite stellt der Baustein abgesehen von den Indikatoren für die erfolgreich abgeschlossene (xDone) oder wegen eines Fehlers beendete (xError) Ausführung des Bausteins einen zweifachen Ausgang zur Verfügung für die Rückmeldungen des Bausteins als Integer-Wert (iResponse) und als String (sResponseDescription).

Auch die weiteren Informationen aus dem AppState der Lernfahrt (Status der Eingänge, aktive Warnung) werden über eigene Ausgänge zur Verfügung gestellt.

Das Ausführen des Bausteins mit iOperatingMode = 0 (Stop) führt dazu, dass die Lernfahrt zwar aktiviert wird (valve mode = 60), jedoch nicht gestartet. Das bedeutet, es werden noch keine Ventilzustände geändert. In diesem Fall enthalten die Ausgänge „iResponse“ und „sResponseDescription“ das Ergebnis der zuletzt ausgeführten Lernfahrt. Falls für die einzulernende Motion App noch keine Lernfahrt durchgeführt wurde, geben die Ausgänge den Wert „1“ bzw. „TeachInRun_unlearned“ aus.

Bei Aktivierung des Schreibzugriffs über die WebConfig-Oberfläche während des aktiven Zustands des Bausteins gibt der Baustein am Ausgang iResponse den Wert „110“ bzw. „access_denied_WebConfig“ zurück.

11.2 Ein- und Ausgänge

Die folgende Tabelle zeigt die Ein- und Ausgangsvariablen des Funktionsbausteins.

Bezeichnung	Variablentyp	Datentyp	Beschreibung
wBusDataFromVTEM0 wBusDataFromVTEM1 wBusDataFromVTEM2	VAR_INPUT	WORD	Die vom Motion Terminal empfangenen 6 Byte Eingangsdaten müssen an diesen Eingängen angelegt werden (→ E/A-Daten mit den Bausteinen verbinden).
wBusDataToVTEM0 wBusDataToVTEM1 wBusDataToVTEM2	VAR_IN_OUT	WORD	Die 6 Byte Ausgangsdaten, die zum Motion Terminal gesendet werden müssen, liegen an diesen Ausgängen (→ E/A-Daten mit den Bausteinen verbinden).
xEnable	VAR_INPUT	BOOL	TRUE: Start des Bausteins FALSE: Beenden des Baustein in jedem Zustand. Das Ventil und der Baustein werden zurückgesetzt. Zurücksetzen ist beendet, wenn iResponse = 210 („idle“) → Der Datentyp eResponseTeachInRun . Solange muss der Baustein aufgerufen werden.

Bezeichnung	Variablentyp	Datentyp	Beschreibung
iParameterSet	VAR_INPUT	INT	0: aktives Parameter Set wird verwendet 1...5: aktives Parameter Set wird für die Dauer der Lernfahrt mit dem Wert dieser Variable überschrieben und anschließend wird das vorherige Parameter Set wiederhergestellt.
iMotionAppToBeToughtIn	VAR_INPUT	INT	ID der Motion App, die eingelernt werden soll.
iOperationMode	VAR_INPUT	INT	0: Stopp 1: Automatisch 2: Manuell (nur Motion App #11)
iManualControl	VAR_INPUT	INT	0: Sperren 1: Ausfahren 2: Einfahren 3: Entlüften
xAcknowledge	VAR_INPUT	BOOL	Eine steigende Flanke quittiert inaktive Fehler in der Störungsliste des Ventils.
xError	VAR_OUTPUT	BOOL	Ausgang wird TRUE, wenn während der Ausführung des Bausteins bzw. der Lernfahrt ein Fehler auftritt.
iResponse	VAR_OUTPUT	INT	Rückmeldung zum Status des Bausteins oder der Lernfahrt als numerischer Wert. Bedeutung: → Der Datentyp eResponseTeachInRun .
sResponseDescription	VAR_OUTPUT	STRING	Rückmeldung zum Status des Bausteins oder der Lernfahrt als Textausgabe, entsprechend der Enumeration eResponseTeachInRun.
xWarning	VAR_OUTPUT	BOOL	Ausgang wird gesteuert durch Bit 7 im AppState.
xRetractedEndPos_optional	VAR_OUTPUT	BOOL	Ausgang entspricht dem Zustand am zugewiesenen Eingang des digitalen Eingangsmoduls für die eingefahrene Endposition (→ Beschreibung Motion Terminal VTEM, Funktion und Parametrierung). Die Zuweisung erfolgt mit Hilfe der Parametrierbausteine der Motion Apps (→ Beispiel zur Zuweisung der Eingänge).
xAdvancedEndPos_optional	VAR_OUTPUT	BOOL	Ausgang entspricht dem Zustand am zugewiesenen Eingang des digitalen Eingangsmoduls für die ausgefahrene Endposition (→ Beschreibung Motion Terminal VTEM, Funktion und Parametrierung). Die Zuweisung erfolgt mit Hilfe der Parametrierbausteine der Motion Apps (→ Beispiel zur Zuweisung der Eingänge).

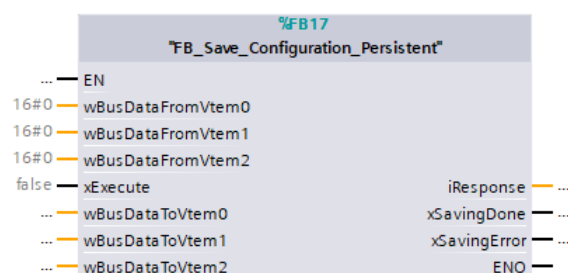
11.2.1 Der Datentyp eResponseTeachInRun

Der Datentyp wird als Enumeration verwendet und dient zur Ausgabe von bausteinspezifische Rückmeldungen (z. B. den Zustand des Bausteins) und enthält zusätzlich Rückgabewerte zum Status der Lernfahrt. Ein Teil der hinterlegten Rückgabewerte entspricht dem Wert des Ausgangs „AppState“ im Fall von fehlerhaften Ausgangsdaten (wBusDataToVTEM...) mit einem Versatz von +100.

Bedeutung	Datentyp	Rückgabewert
TeachInRun_reserved	INT	0
TeachInRun_unlearned	INT	1
TeachInRun_check_tubes	INT	2
TeachInRun_choose_control	INT	3

Bedeutung	Datentyp	Rückgabewert
TeachInRun_tune_trajectory	INT	4
TeachInRun_determine_reference_value	INT	5
TeachInRun_identification	INT	6
TeachInRun_operable_with_reduced_performance	INT	10
TeachInRun_completed_with_limitations	INT	11
TeachInRun_partially_completed_only_2	INT	12
TeachInRun_partially_completed_only_4	INT	13
TeachInRun_completed	INT	14
TeachInRun_error	INT	15
invalid_SetValveMode	INT	101
invalid_SetAppControl	INT	102
invalid_SetAppOption	INT	103
invalid_SetPointValue1	INT	104
invalid_SetPointValue2	INT	105
invalid_configuration	INT	106
access_denied_WebConfig	INT	110
access_denied_saving_parameterisation	INT	111
no_license_for_this_MA	INT	112
all_licenses_in_use	INT	113
invalid_license_file	INT	114
no_valve_detected	INT	115
valve_self_calibration_running	INT	116
idle	INT	210
storing_first_cycle_input_data	INT	215
resetting_valve_to_61	INT	220
ReadingActiveParameterSet	INT	230
ChangingActiveParameterSet	INT	232
writing_parameter_MotionAppToBeToughtIn	INT	234
running_teachIn_run	INT	240
teachin_run_completed	INT	260
error_occured_check_DiagnosticMemory	INT	270
input_data_invalid	INT	271
FB_needs_rising_edge_xEnable	INT	272
acknowledge_needed	INT	273
Invalid_MotionAppToBeToughtIn	INT	274
resetting	INT	300

12 FB_Save_Configuration_Persistent – Parametrierung persistent speichern



12.1 Funktionsbeschreibung

Der Baustein dient zum einfachen Aufruf der Funktion „Konfiguration persistent abspeichern“. Dabei werden die Daten für das aktuell zu parametrierende Ventil dauerhaft auf dem Controller des Motion Terminals abgespeichert.

12.2 Ein- und Ausgänge

Die folgende Tabelle zeigt die Ein- und Ausgangsvariablen des Bausteins.

Bezeichnung	Variablentyp	Datentyp	Beschreibung
wBusDataFromVTEM0 wBusDataFromVTEM1 wBusDataFromVTEM2	VAR_INPUT	WORD	Die vom Motion Terminal empfangenen 6 Byte Eingangsdaten müssen an diesen Eingängen angelegt werden (→ E/A-Daten mit den Bausteinen verbinden).
wBusDataToVTEM0 wBusDataToVTEM1 wBusDataToVTEM2	VAR_IN_OUT	WORD	Die 6 Byte Ausgangsdaten, die zum Motion Terminal gesendet werden müssen, liegen an diesen Ausgängen (→ E/A-Daten mit den Bausteinen verbinden).
xExecute	VAR_INPUT	BOOL	Steigende Flanke am Eingang startet einen Speichervorgang. Ein neuer Speichervorgang benötigt somit eine neue steigende Flanke. Bei Wegnahme des „TRUE“-Signals vor dem Abschluss des Speichervorgangs (xSavingDone = TRUE) wird der Transfermodus beendet.
iResponse	VAR_OUTPUT	INT	Rückmeldung zum Status des Speichervorgangs. Bedeutung: → Der Datentyp eSavingResponse .
xSavingDone	VAR_OUTPUT	BOOL	Der Ausgang wird „TRUE“ wenn der Speichervorgang erfolgreich abgeschlossen wurde.
xSavingError	VAR_OUTPUT	BOOL	Der Ausgang wird „TRUE“ wenn während des Speichervorgangs ein Fehler aufgetreten ist.

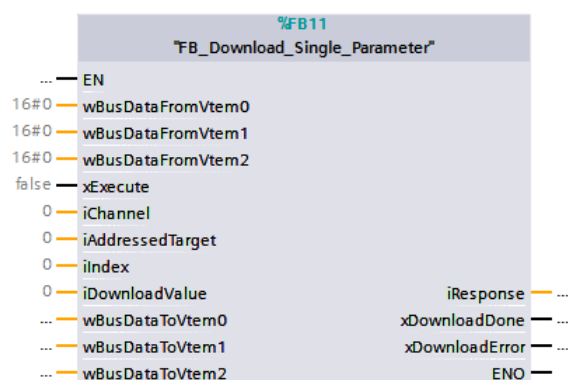
12.2.1 Der Datentyp eSavingResponse

Der Datentyp wird als Enumeration verwendet und dient zur Ausgabe von Rückmeldungen zum aktuellen Status des Speichervorgangs.

Bedeutung	Datentyp	Rückgabewert
saving_in_progress	INT	1
saving_successful	INT	2
saving_not_possible	INT	3
saving_failed	INT	4
access_denied_WebConfig	INT	10
access_denied_saving	INT	11
no_valve_detected	INT	15
preparing_to_save	INT	20

Bedeutung	Datentyp	Rückgabewert
ready_to_save	INT	21
FB_needs_rising_edge_xExecute	INT	25

13 FB_Download_Single_Parameter – Einzelne Parameter zum Motion Terminal übertragen



13.1 Funktionsbeschreibung

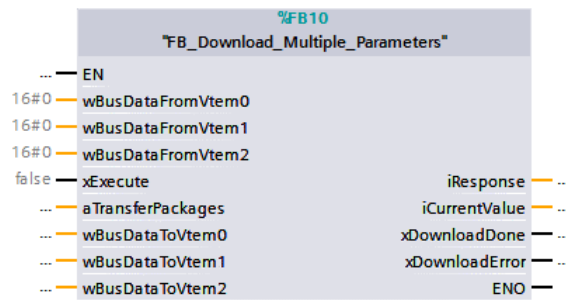
Der Baustein überträgt einen Wert über den Transfermodus von der SPS zum Motion Terminal. Er kann nur für beschreibbare Werte verwendet werden.

13.2 Ein- und Ausgänge

Die folgende Tabelle zeigt die Ein- und Ausgangsvariablen des Bausteins.

Bezeichnung	Variablentyp	Datentyp	Beschreibung
wBusDataFromVTEM0 wBusDataFromVTEM1 wBusDataFromVTEM2	VAR_INPUT	WORD	Die vom Motion Terminal empfangenen 6 Byte Eingangsdaten müssen an diesen Eingängen angelegt werden (→ E/A-Daten mit den Bausteinen verbinden).
wBusDataToVTEM0 wBusDataToVTEM1 wBusDataToVTEM2	VAR_IN_OUT	WORD	Die 6 Byte Ausgangsdaten, die zum Motion Terminal gesendet werden müssen, liegen an diesen Ausgängen (→ E/A-Daten mit den Bausteinen verbinden).
xExecute	VAR_INPUT	BOOL	Steigende Flanke am Eingang startet die Übertragung. Bei Wegnahme des „TRUE“-Signals vor dem Abschluss der Übertragung (xDownloadDone = TRUE) wird der Transfermodus beendet.
iChannel	VAR_INPUT	INT	Kanal des Transfermodus
iAddressedTarget	VAR_INPUT	INT	Kanalspezifische Adresse des Ziels, in das der zu übertragende Wert geschrieben werden soll.
iIndex	VAR_INPUT	INT	Index des Parameters, der auf den zu übertragenden Wert gesetzt werden soll.
iDownloadValue	VAR_INPUT	INT	Wert, auf den der Parameter gesetzt werden soll.
iResponse	VAR_OUTPUT	INT	Rückmeldung zum Status der Übertragung. Bedeutung: → Der Datentyp eTransferResponse im Abschnitt VTEM_AOI_Write_config_MA ... – Parametrierung einer Motion App schreiben .
xDownloadDone	VAR_OUTPUT	BOOL	Ausgang wird TRUE, wenn die Übertragung ohne Fehler abgeschlossen werden konnte.
xDownloadError	VAR_OUTPUT	BOOL	Ausgang wird TRUE, wenn bei der Übertragung Fehler aufgetreten sind.

14 FB_Download_Multiple_Parameters – Mehrere Parameter zum Motion Terminal übertragen



14.1 Funktionsbeschreibung

Der Baustein überträgt eine Liste mit Werten über den Transfermodus von der SPS zum Motion Terminal. Die Liste besteht aus einem eindimensionalen Array mit Elementen des Datentyps „stTransferPackage“. Es können nur beschreibbare Werte übertragen werden.



Ein einziger unzulässiger Wert in der Liste führt zu einem Abbruch des Übertragungsprozesses.

14.2 Ein- und Ausgänge

Bezeichnung	Variablentyp	Datentyp	Beschreibung
wBusDataFromVTEM0 wBusDataFromVTEM1 wBusDataFromVTEM2	VAR_INPUT	WORD	Die vom Motion Terminal empfangenen 6 Byte Eingangsdaten müssen an diesen Eingängen angelegt werden (→ E/A-Daten mit den Bausteinen verbinden).
wBusDataToVTEM0 wBusDataToVTEM1 wBusDataToVTEM2	VAR_IN_OUT	WORD	Die 6 Byte Ausgangsdaten, die zum Motion Terminal gesendet werden müssen, liegen an diesen Ausgängen (→ E/A-Daten mit den Bausteinen verbinden).
xExecute	VAR_INPUT	BOOL	Steigende Flanke am Eingang startet die Übertragung. Bei Wegnahme des „TRUE“-Signals vor dem Abschluss der Übertragung (xDownloadDone = TRUE) wird der Transfermodus beendet.
aTransferPackages	VAR_INPUT	ARRAY [0..100] OF stTransferPackage	Liste der Werte, die übertragen werden sollen (inklusive Kanal, Adresse des Ziels und Index) (→ Der Datentyp stTransferPackage). Nach dem letzten gültigen TransferPackage muss ein Eintrag folgen, der vollständig „0“ ist. Damit wird dem Baustein signalisiert, dass keine weiteren Transfers nötig sind. Es können somit bis zu 100 Werte übertragen werden. In diesem Fall muss der 101. Eintrag [100] in allen Elementen des Datentyps den Wert „0“ enthalten.
iResponse	VAR_OUTPUT	INT	Rückmeldung zum Status der Übertragung. Bedeutung: → Der Datentyp eTransferResponse im Abschnitt VTEM_AOI_Write_config_MA ... – Parametrierung einer Motion App schreiben .

Bezeichnung	Variablentyp	Datentyp	Beschreibung
iCurrentValue	VAR_OUTPUT	INT	Nummer des Array-Elements, das aktuell übertragen wird. Diese Information kann bei Abbruch der Übertragung zur Identifikation des Abbruchzeitpunkts genutzt werden.
xDownloadDone	VAR_OUTPUT	BOOL	Ausgang wird TRUE, wenn die Übertragung ohne Fehler abgeschlossen werden konnte.
xDownloadError	VAR_OUTPUT	BOOL	Ausgang wird TRUE, wenn bei der Übertragung Fehler aufgetreten sind.

14.2.1 Der Datentyp stTransferPackage

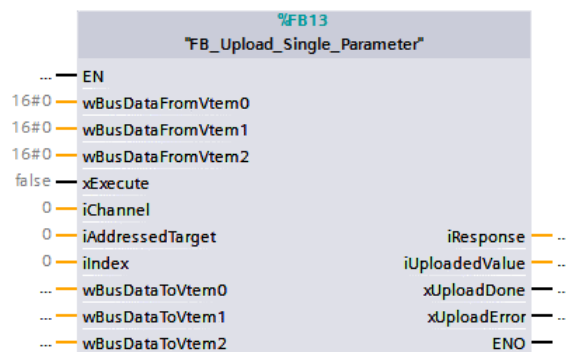
Der Datentyp bildet das Muster für einen Datensatz der Liste, die mit den Bausteinen „FB_Download_Multiple_Parameters“ und „FB_Upload_Multiple_Parameters“ übertragen wird.

Der Datentyp enthält jeweils einen Integer-Wert für den Kanal, die Adresse des Ziels, den Index und den zu übertragenden Wert.

Die folgende Tabelle zeigt die Elemente der Struktur.

Bezeichnung	Datentyp	Beschreibung
iChannel	INT	Kanal des Transfermodus
iAddressedTarget	INT	Kanalspezifische Adresse des Ziels, dessen Wert geschrieben bzw. gelesen werden soll.
iIndex	INT	Index des zu übertragenden Werts
iTransferValue	INT	Wert, auf den der Parameter gesetzt werden soll (Download) bzw. gelesener Wert (Upload).

15 FB_Upload_Single_Parameter – Einzelnen Wert vom Motion Terminal auslesen



15.1 Funktionsbeschreibung

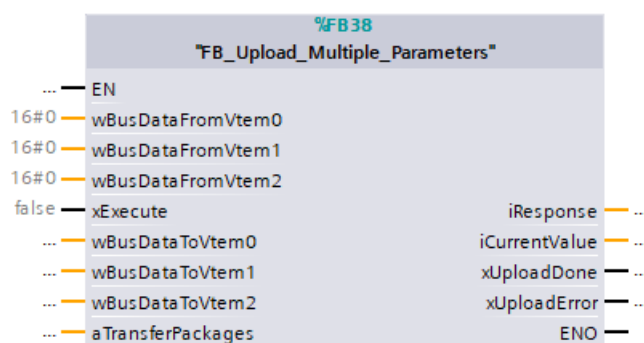
Der Baustein überträgt einen Wert über den Transfermodus vom Motion Terminal zur SPS.

15.2 Ein- und Ausgänge

Die folgende Tabelle zeigt die Ein- und Ausgangsvariablen des Bausteins.

Bezeichnung	Variablentyp	Datentyp	Beschreibung
wBusDataFromVTEM0 wBusDataFromVTEM1 wBusDataFromVTEM2	VAR_INPUT	WORD	Die vom Motion Terminal empfangenen 6 Byte Eingangsdaten müssen an diesen Eingängen angelegt werden (→ E/A-Daten mit den Bausteinen verbinden).
wBusDataToVTEM0 wBusDataToVTEM1 wBusDataToVTEM2	VAR_IN_OUT	WORD	Die 6 Byte Ausgangsdaten, die zum Motion Terminal gesendet werden müssen, liegen an diesen Ausgängen (→ E/A-Daten mit den Bausteinen verbinden).
xExecute	VAR_INPUT	BOOL	Steigende Flanke am Eingang startet die Übertragung. Bei Wegnahme des „TRUE“-Signals vor dem Abschluss der Übertragung (xUploadDone = TRUE) wird der Transfermodus beendet.
iChannel	VAR_INPUT	INT	Kanal des Transfermodus
iAddressedTarget	VAR_INPUT	INT	Kanalspezifische Adresse des Ziels, von dem der zu übertragende Wert gelesen werden soll.
iIndex	VAR_INPUT	INT	Index, für den der hinterlegte Wert ausgelesen werden soll.
iResponse	VAR_OUTPUT	INT	Rückmeldung zum Status der Übertragung. Bedeutung: → Der Datentyp eTransferResponse im Abschnitt VTEM_AOI_Write_config_MA_... – Parametrierung einer Motion App schreiben .
iUploadedValue	VAR_OUTPUT	INT	Ausgelesener Wert
xUploadDone	VAR_OUTPUT	BOOL	Ausgang wird TRUE, wenn die Übertragung ohne Fehler abgeschlossen werden konnte.
xUploadError	VAR_OUTPUT	BOOL	Ausgang wird TRUE, wenn bei der Übertragung Fehler aufgetreten sind.

16 FB_Upload_Multiple_Parameters – Mehrere Werte vom Motion Terminal auslesen



16.1 Funktionsbeschreibung

Der Baustein überträgt eine Liste mit Werten über den Transfermodus vom Motion Terminal zur SPS. Die Liste besteht aus einem eindimensionalen Array mit Elementen des Datentyps „stTransferPackage“.

16.2 Ein- und Ausgänge

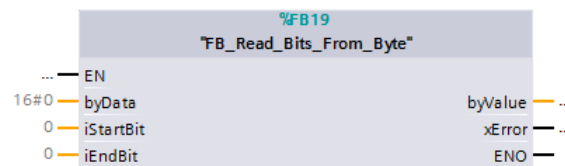
Die folgende Tabelle zeigt die Ein- und Ausgangsvariablen des Bausteins.

Bezeichnung	Variablentyp	Datentyp	Beschreibung
wBusDataFromVTEM0 wBusDataFromVTEM1 wBusDataFromVTEM2	VAR_INPUT	WORD	Die vom Motion Terminal empfangenen 6 Byte Eingangsdaten müssen an diesen Eingängen angelegt werden (→ E/A-Daten mit den Bausteinen verbinden).
wBusDataToVTEM0 wBusDataToVTEM1 wBusDataToVTEM2	VAR_IN_OUT	WORD	Die 6 Byte Ausgangsdaten, die zum Motion Terminal gesendet werden müssen, liegen an diesen Ausgängen (→ E/A-Daten mit den Bausteinen verbinden).
xExecute	VAR_INPUT	BOOL	Steigende Flanke am Eingang startet die Übertragung. Bei Wegnahme des „TRUE“-Signals vor dem Abschluss der Übertragung (xUploadDone = TRUE) wird der Transfermodus beendet.
aTransferPackages	VAR_IN_OUT	ARRAY [0..100] OF stTransferPackage	Liste der Werte, die übertragen werden sollen (inklusive Kanal, Adresse des Ziels und Index) (→ Der Datentyp stTransferPackage). Nach dem letzten gültigen TransferPackage muss ein Eintrag folgen, der vollständig „0“ ist. Damit wird dem Baustein signalisiert, dass keine weiteren Transfers nötig sind. Es können somit bis zu 100 Werte übertragen werden. In diesem Fall muss der 101. Eintrag [100] in allen Elementen des Datentyps den Wert „0“ enthalten.
iResponse	VAR_OUTPUT	INT	Rückmeldung zum Status der Übertragung. Bedeutung: → Der Datentyp eTransferResponse im Abschnitt VTEM_AOI_Write_config_MA_... – Parametrierung einer Motion App schreiben .
iCurrentValue	VAR_OUTPUT	INT	Nummer des Array-Elements, das aktuell übertragen wird. Diese Information kann bei Abbruch der Übertragung zur Identifikation des Abbruchzeitpunkts genutzt werden.
xUploadDone	VAR_OUTPUT	BOOL	Ausgang wird TRUE, wenn die Übertragung ohne Fehler abgeschlossen werden konnte.

Bezeichnung	Variablentyp	Datentyp	Beschreibung
xUploadError	VAR_OUTPUT	BOOL	Ausgang wird TRUE, wenn bei der Übertragung Fehler aufgetreten sind.

17 Hilfbausteine

17.1 FB_Read_Bits_From_Byte – Bit-Bereich aus Byte auslesen



17.1.1 Funktionsbeschreibung

Der Baustein vereinfacht das Auslesen von Bereichen innerhalb eines Bytes. Dazu wird dem Baustein ein Byte übergeben sowie das Startbit und Endbit des Bereiches, dessen Wert als Byte ausgegeben werden soll.

Beispiel

Aus dem Ausgangsbyte „byActualAppState“ der Instanz „fbvalvecontrol_V0“ des Bausteins „FB_ValveControl“ soll das Bit 7 (Warning) ausgelesen und in die Variable „byWarning“ geschrieben werden.



17.1.2 Ein- und Ausgänge

Die folgende Tabelle zeigt die Ein- und Ausgangsvariablen des Bausteins.

Bezeichnung	Variablentyp	Datentyp	Beschreibung
byData	VAR_INPUT	BYTE	Byte, aus dem der Wert des Bereichs zwischen Startbit und Endbit ausgelesen werden soll.
iStartBit	VAR_INPUT	INT	Erstes Bit (0 ... 7) des Bereichs, dessen Wert als Byte ausgegeben werden soll.
iEndBit	VAR_INPUT	INT	Letztes Bit (0 ... 7) des Bereichs, dessen Wert als Byte ausgegeben werden soll.
byValue	VAR_OUTPUT	BYTE	Wert des angegebenen Bereichs als Byte.
xError	VAR_OUTPUT	BOOL	Ausgang wird TRUE, wenn bei der Übertragung Fehler aufgetreten sind.

17.2 FB_Read_Bits_From_Word – Bit-Bereich aus Wort auslesen



17.2.1 Funktionsbeschreibung

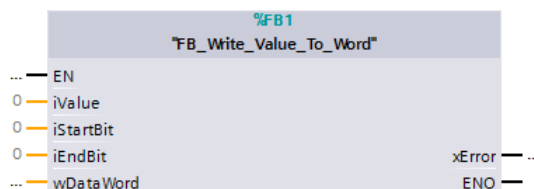
Der Baustein vereinfacht das Auslesen von Bereichen innerhalb eines Wortes. Dazu wird dem Baustein ein Wort übergeben sowie das Startbit und Endbit des Bereiches, dessen Wert als Integer ausgegeben werden soll.

17.2.2 Ein- und Ausgänge

Die folgende Tabelle zeigt die Ein- und Ausgangsvariablen des Bausteins.

Bezeichnung	Variablentyp	Datentyp	Beschreibung
wData	VAR_INPUT	WORD	Wort, aus dem der Wert des Bereichs zwischen Startbit und Endbit ausgelesen werden soll.
iStartBit	VAR_INPUT	INT	Erstes Bit (0 ... 15) des Bereichs, dessen Wert als Integer ausgegeben werden soll.
iEndBit	VAR_INPUT	INT	Letztes Bit (0 ... 15) des Bereichs, dessen Wert als Integer ausgegeben werden soll.
iValue	VAR_OUTPUT	INT	Wert des angegebenen Bereichs als Integer.
xError	VAR_OUTPUT	BOOL	Ausgang wird TRUE, wenn bei der Übertragung Fehler aufgetreten sind.

17.3 FB_Write_Value_To_Word – Wert innerhalb eines Wortes schreiben



17.3.1 Funktionsbeschreibung

Der Baustein vereinfacht das Schreiben von Werten in einen Bereich eines Wortes. Dazu wird dem Baustein das entsprechende Wort übergeben sowie das Startbit und Endbit des Bereiches, in den der angelegte Integer-Wert geschrieben werden soll.

17.3.2 Ein- und Ausgänge

Die folgende Tabelle zeigt die Ein- und Ausgangsvariablen des Bausteins.

Bezeichnung	Variablentyp	Datentyp	Beschreibung
iValue	VAR_INPUT	INT	Wert, der in den angegebenen Bereich geschrieben werden soll.
iStartBit	VAR_INPUT	INT	Erstes Bit (0 ... 15) des Bereiches, in den der übergebene Integer-Wert geschrieben werden soll.
iEndBit	VAR_INPUT	INT	Letztes Bit (0 ... 15) des Bereiches, in den der übergebene Integer-Wert geschrieben werden soll.
wDataWord	VAR_IN_OUT	WORD	Wort, in das der übergebene Wert zwischen Startbit und Endbit geschrieben werden soll.
xError	VAR_OUTPUT	BOOL	Ausgang wird TRUE, wenn bei der Übertragung Fehler aufgetreten sind.

A Technischer Anhang

A.1 Technische Daten

A.1.1 Allgemein

Technische Daten	
Version der Erstellungssoftware	Siemens TIA Portal V19

B Index

A

Anbindung an die Steuerung.....	20
App-Status	
interpretieren.....	30

B

Bestimmungsgemäße Verwendung.....	16
Bibliothek	
Architektur.....	18
Aufbau.....	19
einbinden.....	20
Voraussetzungen.....	20
Bit-Bereich	
auslesen aus Byte.....	60
auslesen aus Wort.....	61
schreiben innerhalb eines Wortes.....	61

D

Datentyp	
eMA100OperatingMedium.....	19
eResponseReadMalfunctionList.....	19, 47
eResponseTeachInRun.....	19, 50
eResponseToValveModeSet.....	19, 27
eSavingResponse.....	19, 52
eTransferResponse.....	19, 35
eValveMode.....	19
eValveState.....	19
eValveType_MA_01_02.....	19
stMalfunction.....	19, 48
stTransferPackage.....	19, 56
Dokumentationen.....	21

E

E/A-Daten.....	23
Eingänge	
Zuweisung.....	38
Erstellungssoftware.....	62

F

FB_Download_Multiple_Parameters.....	55
FB_Download_Single_Parameter.....	54
FB_Read_Bits_From_Byte.....	60
FB_Read_Bits_From_Word.....	61
FB_ReadMalfunctionList.....	46
FB_Save_Configuration_Persistent.....	52
FB_StateInterpreter.....	30
FB_TeachInRun.....	49
FB_Upload_Multiple_Parameters.....	58
FB_Upload_Single_Parameter.....	57

FB_ValveControl.....	25
FB_Write_config_DI/AI_Module.....	43
FB_Write_config_MA.....	37
FB_Write_Value_To_Word.....	61

G

Gerätebeschreibungsdatei.....	20
Grundlagen.....	18

H

Historie.....	2
---------------	---

K

Kommunikationsprotokoll.....	21
------------------------------	----

L

Lernfahrt ausführen.....	49
--------------------------	----

M

Motion App	
betreiben.....	25

P

Parameter vom Motion Terminal auslesen	
einzeln.....	57
mehrere.....	58
Parameter zum Motion Terminal übertragen	
einzeln.....	54
mehrere.....	55
Parametrierung	
Eingangsmodule	
lesen.....	40
schreiben.....	43
Lernfahrt.....	49
Motion App.....	37
Parametrierung persistent speichern.....	52
persistent speichern.....	52

S

Service.....	16
Sicherheitshinweise.....	16
speichern	
persistent.....	52
State-Interpreter-Bausteine.....	30
Störungsliste auslesen.....	46

T

Technische Daten.....	62
-----------------------	----

V

Ventilsteckplatz.....	21
Versions-Info	2

Z

Zielgruppe.....	16
-----------------	----