

Servo Press Kit YJKP - Description of host function blocks in Mitsubishi GX Works2

This application note describes how you use the host function blocks of the servo press kit YJKP in Mitsubishi (GX Works2).

YJKP

Supported systems:

- Tested with Q03UDVCPU
- All compatible devices

Title Servo Press Kit YJKP - Description of host function blocks in Mitsubishi GX Works2
Version 1.10
Document no. 100352
Originalen
AuthorFesto

Last saved 30.09.2021

Copyright Notice

This documentation is the intellectual property of Festo SE & Co. KG, which also has the exclusive copyright. Any modification of the content, duplication or reprinting of this documentation as well as distribution to third parties can only be made with the express consent of Festo SE & Co. KG.

Festo SE & Co KG reserves the right to make modifications to this document in whole or in part. All brand and product names are trademarks or registered trademarks of their respective owners.

Legal Notice

Hardware, software, operating systems and drivers may only be used for the applications described and only in conjunction with components recommended by Festo SE & Co. KG.

Festo SE & Co. KG does not accept any liability for damages arising from the use of any incorrect or incomplete information contained in this documentation or any information missing therefrom.

Defects resulting from the improper handling of devices and modules are excluded from the warranty.

The data and information specified in this document should not be used for the implementation of safety functions relating to the protection of personnel and machinery.

No liability is accepted for claims for damages arising from a failure or functional defect. In other respects, the regulations with regard to liability from the terms and conditions of delivery, payment and use of software of Festo SE & Co. KG, which can be found at www.festo.com and can be supplied on request, shall apply.

All data contained in this document do not represent guaranteed specifications, particularly with regard to functionality, condition or quality, in the legal sense.

The information in this document serves only as basic information for the implementation of a specific, hypothetical application and is in no way intended as a substitute for the operating instructions of the respective manufacturers and the design and testing of the respective application by the user.

The operating instructions for Festo products can be found at www.festo.com/sp.

Users of this document (application note) must verify that all functions described here also work correctly in the application. By reading this document and adhering to the specifications contained therein, users are also solely responsible for their own application.

Table of contents

1	Components/Software used	4
2	Application description.....	5
3	Elements in the library	6
3.1	Overview	6
3.2	Function blocks	7
3.2.1	FB_ControlChannel	7
3.2.2	FB_ParameterChannel	19
3.3	Enumerations	23
3.3.1	EN_AccessType	23
3.3.2	EN_AccessOption.....	23
3.3.3	EN_PosMode.....	24
3.3.4	EN_Acknowledge	24
3.3.5	EN_DataType	24

1 Components/Software used

Type/Name	Version Software/Firmware	Date of manufacture
Servo press kit YJKP	general	--
Application software YJKP (GSAY-A4-F0-Z4-1.3.5)	V1.4.1	--
Firmware controller (CECC-X)	V3.8.14	--
Firmware motor controller (CMMP-AS)	V4.0.1501.2.4	--
CODESYS (CODESYS V3)	V3.5 SP15	--
Library / function blocks	V1.1.1	--

Table 1.1: 1 Components/Software used

2 Application description

This application note describes how you use the host function blocks of the servo press kit YJKP in Mitsubishi GX Works2.



Note

It refers to the functionality of the application software YJKP version GSAY-1.4.1
For lower version use other appropriate application note, which is also available on the Festo website.

Supported systems:

- Tested with Q03UDVCPU
- All compatible devices

Supported fieldbus:

- Modbus TCP

Following descriptions are part of the application note:

- Elements in the library
 - Overview
 - Enumerations
 - Function blocks

Content of the download package:

- Link to the Application Notes
- Example
- Library
- Release Note
- Old versions

3 Elements in the library

3.1 Overview

The Figure 3-1: Overview gives a short overview and dependency of the included host function blocks.

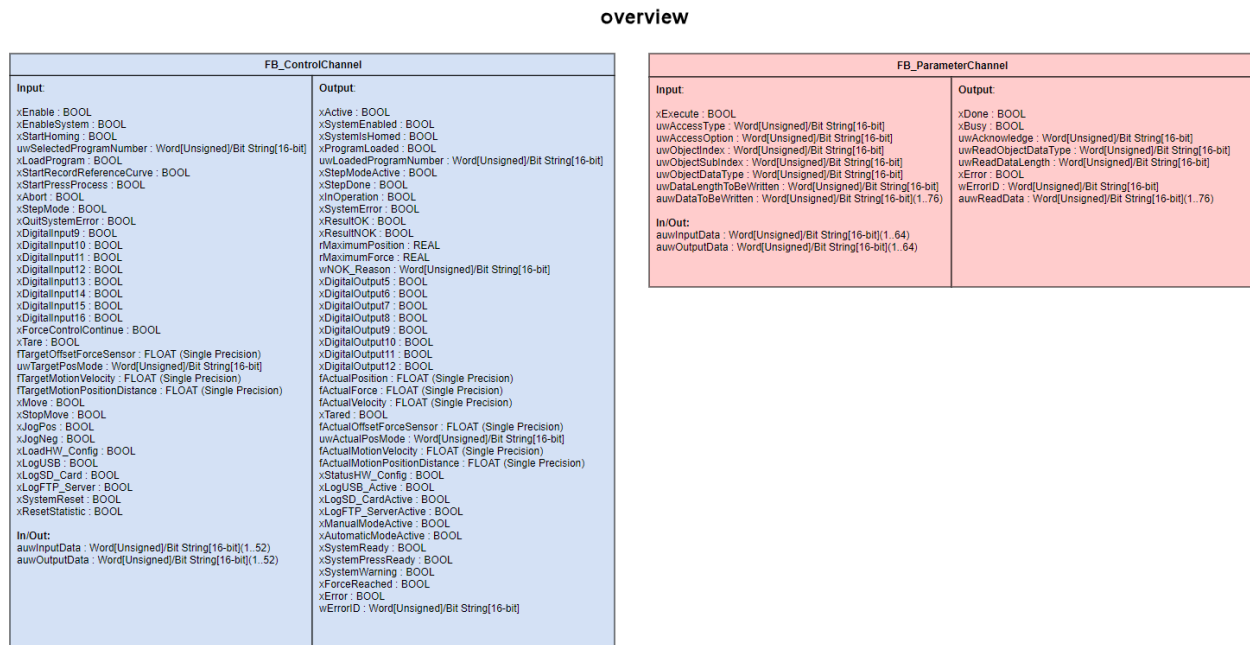


Figure 3-1: Overview

3.2 Function blocks

3.2.1 FB_ControlChannel

This function block provides press control functions for the Festo system.

Pre-Action:

- General:
 - Map the array of the input data (hardware ID of the hardware I/O module) to the function block input "auwInputData"
 - Map the array of the output data (hardware ID of the hardware I/O module) to the function block input "auwOutputData"

FB_ControlChannel	
— xEnable	xActive
— xEnableSystem	xSystemEnabled
— xStartHoming	xSystemIsHomed
— uwSelectedProgramNumber	xManualModeActive
— xLoadProgram	xAutomaticModeActive
— xStartRecordReferenceCurve	xSystemReady
— xStartPressProcess	xSystemPressReady
— xAbort	xSystemWarning
— xStepMode	xForceReached
— xQuitSystemError	xProgramLoaded
— xForceControlContinue	uwLoadedProgramNumber
— xTare	xStepModeActive
— fTargetOffsetForceSensor	xStepDone
— uswTargetPosMode	xInOperation
— fTargetMotionVelocity	xSystemError
— fTargetMotionPositionDistance	xResultOK
— xMove	xResultNOK
— xJogPos	fMaximumPosition
— xJogNeg	fMaximumForce
— xLoadHW_Config	uwNOK_Reason
— xLogUSB	xDigitalOutput5
— xLogSD_Card	xDigitalOutput6
— xLogFTP_Server	xDigitalOutput7
— xSystemReset	xDigitalOutput8
— xResetStatistic	xDigitalOutput9
— xDigitalInput9	xDigitalOutput10
— xDigitalInput10	xDigitalOutput11
— xDigitalInput11	xDigitalOutput12
— xDigitalInput12	fActualPosition
— xDigitalInput13	fActualForce
— xDigitalInput14	fActualVelocity
— xDigitalInput15	fActualOffsetForceSensor
— xDigitalInput16	uwActualPosMode
	fActualMotionVelocity
	fActualMotionPositionDistance
	xTared
	xStatusHW_Config
	xLogUSB_Active
	xLogSD_CardActive
	xLogFTP_ServerActive
	xError
	uwErrorID
— auwInputData	auwInputData
— auwOutputData	auwOutputData

Table 3-1: FB_ControlChannel

In/Out		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
Inputs		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
xEnable	BIT	Activate press control functions for the Festo system FALSE: Action stopped, resets outputs TRUE: Action running
xEnableSystem	BIT	Enable system command Falling edge: Disable system Rising edge: Enable system
xStartHoming	BIT	Homing command Falling edge: No action Rising edge: Start homing movement
uwSelectedProgramNumber	Word[Un- signed]/Bit String[16-bit]	Selected program number
xLoadProgram	BIT	Load program command Falling edge: No action Rising edge: Load selected program number
xStartRecordReferenceCurve	BIT	Start record reference curve command Falling edge: No action Rising edge: Start record reference curve (Only in window "Step 2/4: Record / loading reference curve(s)" possible)
xStartPressProcess	BIT	Start press process command Falling edge: No action Rising edge: Start press process
xAbort	BIT	Abort command FALSE: Abort press process or homing TRUE: No action
xStepMode	BIT	Step mode command FALSE: Deactivate step mode TRUE: Activate step mode
xQuitSystemError	BIT	Quit system error command Falling edge: No action Rising edge: Quit upcoming system error
xDigitalInput9	BIT	Virtual digital input 9 for step enabling conditions in the sequencer of the press process
xDigitalInput10	BIT	Virtual digital input 10 for step enabling conditions in the sequencer of the press process
xDigitalInput11	BIT	Virtual digital input 11 for step enabling conditions in the sequencer of the press process
xDigitalInput12	BIT	Virtual digital input 12 for step enabling conditions in the sequencer of the press process
xDigitalInput13	BIT	Virtual digital input 13 for step enabling conditions in the sequencer of the press process
xDigitalInput14	BIT	Virtual digital input 14 for step enabling conditions in the sequencer of the press process
xDigitalInput15	BIT	Virtual digital input 15 for step enabling conditions in the sequencer of the press process
xDigitalInput16	BIT	Virtual digital input 16 for step enabling conditions in the sequencer of the press process

In/Out		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
xForceControlContinue	BIT	Continue force control command Falling edge: No action Rising edge: Stop force control and move to next step without NOK evaluation although force control step is not finished
xTare	BIT	Tare of the force sensor command Falling edge: Reset tare of the force sensor Rising edge: Tare of the force sensor
fTargetOffsetForceSensor	FLOAT (Single Precision)	Target offset of the force sensor Unit [N]
uwTargetPosMode	Word[Un-signed]/Bit String[16-bit]	Target position mode 0x00 = Jog 0x01 = Move absolute 0x02 = Move relative
fTargetMotionVelocity	FLOAT (Single Precision)	Target motion velocity for target position mode (Jog, Move absolute/relative) Unit [mm/s]
fTargetMotionPositionDistance	FLOAT (Single Precision)	Target motion position/distance for target position mode (Move absolute/relative) Unit [mm]
xMove	BIT	Move command Falling edge: No action Rising edge: Start movement depending on target position mode (Move absolute/relative), target motion velocity and target motion position/distance
xStopMove	BIT	Stop move command Falling edge: No action Rising edge: Stop movement depending on target position mode (Move absolute/relative)
xJogPos	BIT	Jog positive command Falling edge: Stop movement depending on target position mode (Jog) Rising edge: Start movement in positive direction depending on target position mode (Jog) and target motion velocity
xJogNeg	BIT	Jog negative command Falling edge: Stop movement depending on target position mode (Jog) Rising edge: Start movement in negative direction depending on target position mode (Jog) and target motion velocity
xLoadHW_Config	BIT	Load hardware configuration command Falling edge: No action Rising edge: Load hardware configuration (Configured hardware in window "Hardware configuration")
xLogUSB	BIT	Log USB command False: Deactivate log on USB-Stick True: Activate log on USB-Stick
xLogSD_Card	BIT	Log SD-Card command False: Deactivate log on SD-Card True: Activate log on SD-Card
xLogFTP_Server	BIT	Log FTP-Server command False: Deactivate log on FTP-Server True: Activate log on FTP-Server

In/Out		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
xSystemReset	BIT	System reset command Falling edge: No action Rising edge: System reset
xResetStatistic	BIT	Reset statistic command Falling edge: No action Rising edge: Reset statistic (Reset statistic in window "Process diagnosis")
In/Out		
auwInputData	Word[Un- signed]/Bit String[16- bit](1..64)	Start address of input data Hardware ID of the module from which the data is to be read
auwOutputData	Word[Un- signed]/Bit String[16- bit](1..64)	Start address of output data Hardware ID of the module to which the data is to be written
Outputs		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
xActive	BIT	Function block is active FALSE: Function block is inactive TRUE: Function block is active
xSystemEnabled	BIT	System is enabled FALSE: System is disabled TRUE: System is enabled
xSystemIsHomed	BIT	System is homed FALSE: System is not homed TRUE: System is homed
xProgramLoaded	BIT	Program is loaded Falling edge: No action Rising edge: Program is loaded
uwLoadedProgramNumber	Word[Un- signed]/Bit String[16-bit]	Loaded program number
xStepModeActive	BIT	Step mode is active FALSE: Step mode is inactive TRUE: Step mode is active
xStepDone	BIT	Step is done Falling edge: No action Rising edge: Step is done
xInOperation	BIT	System is in operation FALSE: System is not in operation TRUE: System is in operation
xSystemError	BIT	System error FALSE: No error TRUE: Error occurred
xResultOK	BIT	OK result of the press process
xResultNOK	BIT	NOK result of the press process For further information see NOK reason
fMaximumPosition	FLOAT (Single Pre- cision)	Maximum position of the press process Unit [mm]

In/Out		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
fMaximumForce	FLOAT (Single Precision)	Maximum force of the press process Unit [N]
wNOK_Reason	Word[Un-signed]/Bit String[16-bit]	NOK reason of the press process
xDigitalOutput5	BIT	Virtual digital output 5 in the sequencer of the press process
xDigitalOutput6	BIT	Virtual digital output 6 in the sequencer of the press process
xDigitalOutput7	BIT	Virtual digital output 7 in the sequencer of the press process
xDigitalOutput8	BIT	Virtual digital output 8 in the sequencer of the press process
xDigitalOutput9	BIT	Virtual digital output 9 in the sequencer of the press process
xDigitalOutput10	BIT	TVirtual digital output 10 in the sequencer of the press process
xDigitalOutput11	BIT	Virtual digital output 11 in the sequencer of the press process
xDigitalOutput12	BIT	Virtual digital output 12 in the sequencer of the press process
fActualPosition	FLOAT (Single Precision)	Actual position Unit [mm]
fActualForce	FLOAT (Single Precision)	Actual force Unit [N]
fActualVelocity	FLOAT (Single Precision)	Actual velocity Unit [mm/s]
XTared	BIT	Status of tare of the force sensor FALSE: Force sensor not tared TRUE: Force sensor tared
fActualOffsetForceSensor	FLOAT (Single Precision)	Actual offset of the force sensor Unit [N]
uwActualPosMode	Word[Un-signed]/Bit String[16-bit]	Actual position mode 0x00 = Jog 0x01 = Move absolute 0x02 = Move relative
fActualMotionVelocity	FLOAT (Single Precision)	Actual motion velocity for actual position mode (Jog, Move absolute/relative) Unit [mm/s]
fActualMotionPositionDistance	FLOAT (Single Precision)	Actual motion position/distance for actual position mode (Move absolute/relative) Unit [mm]
xStatusHW_Config	BIT	Status of the hardware configuration FALSE: Hardware configuration invalid TRUE: Hardware configuration valid
xLogUSB_Active	BIT	Log on USB is active FALSE: Log on USB-Stick is inactive TRUE: Log on USB-Stick is active

In/Out		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
xLogSD_CardActive	BIT	Log on SD-Card is active FALSE: Log on SD-Card is inactive TRUE: Log on SD-Card is active
xLogFTP_ServerActive	BIT	Log on FTP-Server is active FALSE: Log on FTP-Server is inactive TRUE: Log on FTP-Server is active
xManualModeActive	BIT	Manual mode active FALSE: Manual mode is inactive TRUE: Manual mode is active
xAutomaticModeActive	BIT	Automatic mode active FALSE: Automatic mode is inactive TRUE: Automatic mode is active
xSystemReady	BIT	System is ready FALSE: System is not ready TRUE: System is ready
xSystemPressReady	BIT	System servo press is ready FALSE: System servo press is not ready TRUE: System servo press is ready
xSystemWarning	BIT	System warning FALSE: No warning TRUE: Warning occurred
xForceReached	BIT	Force reached in force control FALSE: Force not reached in force control TRUE: Force reached in force control
xError	BIT	Error occurred during action FALSE: No error TRUE: Error occurred, function block aborts action
wErrorID	Word[Un- signed]/Bit String[16-bit]	Contains detailed error information in case of an error For further information see error table

Table 3-6: FB_ControlChannel

Error table	
<i>Error ID</i>	<i>Description</i>
0x0000	No error No error information available
0x0001	Invalid state of the main state machine
0x0002	Invalid state of the in operation state machine
0x0003	Connection timeout

Table 3-7: Error table FB_ControlChannel

Examples:**Load hardware configuration**

1. Precondition:
 1. Input "xEnable" is TRUE
 2. Output "xActive" is TRUE
 3. Output "xSystemError" is FALSE
 4. Output "xInOperation" is FALSE

5. Output "xError" is FALSE
2. You have to be in the window "Hardware configuration" in the WebVisu of the Festo system
3. Set in the WebVisu your desired configuration
4. Set input "xLoadHW_Config" to TRUE
5. During operation output "xInOperation" is TRUE
6. Condition of load hardware configuration done
 1. Check if output "xActive" is TRUE
 2. Check if output "xInOperation" is FALSE
 3. Check if output "xSystemError" is FALSE
 4. Check if output "xError" is FALSE
7. Save the loaded hardware configuration via the button "Save" in the WebVisu

Activate logging (For all logging functions the same)

1. Precondition:
 1. Input "xEnable" is TRUE
 2. Output "xActive" is TRUE
 3. Output "xSystemError" is FALSE
 4. Output "xInOperation" is FALSE
 5. Output "xError" is FALSE
2. Set input "xLogSD_Card" to TRUE
3. Condition of logging on the SD-Card active
 1. Check if output "xActive" is TRUE
 2. Check if output "xLogSD_CardActive" is TRUE

System reset

1. Precondition:
 1. Input "xEnable" is TRUE
 2. Output "xActive" is TRUE
 3. Output "xSystemError" is FALSE
 4. Output "xInOperation" is FALSE
 5. Output "xError" is FALSE
2. Set input "xSystemReset" to TRUE
3. A reboot of the controller of the Festo system will be performed

Reset statistic

1. Precondition:
 1. Input "xEnable" is TRUE
 2. Output "xActive" is TRUE
 3. Output "xSystemError" is FALSE
 4. Output "xInOperation" is FALSE
 5. Output "xError" is FALSE
2. Set input "xResetStatistic" to TRUE
3. The statistic in the window "Process diagnosis" in the WebVisu of the Festo system will be reset

Enable system

1. Precondition:
 1. Input "xEnable" is TRUE
 2. Output "xActive" is TRUE
 3. Output "xInOperation" is FALSE
 4. Output "xSystemError" is FALSE
 5. Output "xError" is FALSE
 6. Output "xSystemPressReady" is TRUE
2. Set input "xEnableSystem" to TRUE
3. During operation output "xInOperation" is TRUE
4. Condition of system enabled
 1. Check if output "xActive" is TRUE
 2. Check if output "xInOperation" is FALSE
 3. Check if output "xSystemError" is FALSE
 4. Check if output "xError" is FALSE

5. Check if output "xSystemIsEnabled" is TRUE

Disable system

1. Precondition:
 1. Input "xEnable" is TRUE
 2. Output "xActive" is TRUE
 3. Output "xInOperation" is FALSE
 4. Output "xSystemEnabled" is TRUE
 5. Output "xSystemError" is FALSE
 6. Output "xError" is FALSE
 7. Output "xSystemPressReady" is TRUE
2. Set input "xEnableSystem" to FALSE
3. During operation output "xInOperation" is TRUE
4. Condition of system disabled
 1. Check if output "xActive" is TRUE
 2. Check if output "xInOperation" is FALSE
 3. Check if output "xSystemError" is FALSE
 4. Check if output "xError" is FALSE
 5. Check if output "xSystemIsEnabled" is FALSE

Tare force sensor



Caution

Not used for the press process, for the press process use the function inside the sequencer to tare the force sensor!!!

1. Precondition:
 1. Input "xEnable" is TRUE
 2. Output "xActive" is TRUE
 3. Output "xInOperation" is FALSE
 4. Output "xSystemError" is FALSE
 5. Output "xError" is FALSE
2. Set input "xTare" to TRUE
3. Condition of force sensor tared
 1. Check if output "xActive" is TRUE
 2. Check if output "xTared" is TRUE

Add offset to the force sensor



Caution

Not used for the press process, for the press process use the function inside the sequencer to add an offset to the force sensor!!!

1. Precondition:
 1. Input "xEnable" is TRUE
 2. Output "xActive" is TRUE
 3. Output "xInOperation" is FALSE
 4. Output "xSystemError" is FALSE
 5. Output "xError" is FALSE
2. Set input "fTargetOffsetForceSensor" to your desired offset (N)
3. Condition of offset added to the force sensor
 1. Check if output "xActive" is TRUE
 2. Check if output "rActualOffsetForceSensor" is old value plus "rTargetOffsetForceSensor"

Homing

1. Precondition:
 1. Input "xEnable" is TRUE
 2. Output "xActive" is TRUE
 3. Output "xInOperation" is FALSE

4. Output "xSystemError" is FALSE
5. Output "xError" is FALSE
6. Output "xStatusHW_Config" is TRUE
2. Set input "xAbort" to TRUE
3. Set input "xStartHoming" to TRUE
4. During operation output "xInOperation" is TRUE
5. Homing can be aborted via set input "xAbort" to FALSE
6. Condition of homing aborted
 1. Check if output "xActive" is TRUE
 2. Check if output "xInOperation" is FALSE
 3. Check if output "xSystemError" is FALSE
 4. Check if output "xError" is FALSE
 5. Check if output "xSystemIsHomed" is FALSE
7. Condition of homing done
 1. Check if output "xActive" is TRUE
 2. Check if output "xInOperation" is FALSE
 3. Check if output "xSystemError" is FALSE
 4. Check if output "xError" is FALSE
 5. Check if output "xSystemIsHomed" is TRUE

Jog

1. Precondition:
 1. Input "xEnable" is TRUE
 2. Output "xActive" is TRUE
 3. Output "xSystemEnabled" is TRUE
 4. Output "xInOperation" is FALSE
 5. Output "xSystemError" is FALSE
 6. Output "xError" is FALSE
 7. Output "xSystemPressReady" is TRUE
2. Set input "uwTargetPosMode" to 0x00 (Jog)
3. Set input "fTargetMotionVelocity" to your desired velocity (mm/s)
4. Set input "xJogPos" or "xJogPos" to TRUE
5. During operation output "xInOperation" is TRUE
6. Jogging can be stopped via set input "xJogPos" or "xJogPos" to FALSE
7. Condition of jogging stopped or done
 1. Check if output "xActive" is TRUE
 2. Check if output "xInOperation" is FALSE
 3. Check if output "xSystemError" is FALSE
 4. Check if output "xError" is FALSE

Move absolute

1. Precondition:
 1. Input "xEnable" is TRUE
 2. Output "xActive" is TRUE
 3. Output "xSystemEnabled" is TRUE
 4. Output "xInOperation" is FALSE
 5. Output "xSystemError" is FALSE
 6. Output "xError" is FALSE
 7. Output "xSystemPressReady" is TRUE
2. Set input "uwTargetPosMode" to 0x01 (Move absolute)
3. Set input "fTargetMotionVelocity" to your desired velocity (mm/s)
4. Set input "fTargetMotionPositionDistance" to your desired position (mm)
5. Set input "xStopMove" to FALSE
6. Set input "xMove" to TRUE
7. During operation output "xInOperation" is TRUE
8. Move absolute can be stopped via set input "xStopMove" to TRUE
9. Condition of move absolute stopped
 1. Check if output "xActive" is TRUE
 2. Check if output "xInOperation" is FALSE
 3. Check if output "xSystemError" is FALSE
 4. Check if output "xError" is FALSE
 5. Check if output "fActualPosition" is unequal to input "fTargetMotionPositionDistance"

10. Condition of move absolute done
 1. Check if output "xActive" is TRUE
 2. Check if output "xInOperation" is FALSE
 3. Check if output "xSystemError" is FALSE
 4. Check if output "xError" is FALSE
 5. Check if output "fActualPosition" is equal to input "fTargetMotionPositionDistance"

Move relative

1. Precondition:
 1. Input "xEnable" is TRUE
 2. Output "xActive" is TRUE
 3. Output "xSystemEnabled" is TRUE
 4. Output "xInOperation" is FALSE
 5. Output "xSystemError" is FALSE
 6. Output "xError" is FALSE
 7. Output "xSystemPressReady" is TRUE
2. Set input "uwTargetPosMode" to 0x02 (Move relative)
3. Set input "fTargetMotionVelocity" to your desired velocity (mm/s)
4. Set input "fTargetMotionPositionDistance" to your desired distance (mm)
5. Set input "xStopMove" to FALSE
6. Set input "xMove" to TRUE
7. During operation output "xInOperation" is TRUE
8. Move relative can be stopped via set input "xStopMove" to TRUE
9. Condition of move relative stopped
 1. Check if output "xActive" is TRUE
 2. Check if output "xInOperation" is FALSE
 3. Check if output "xSystemError" is FALSE
 4. Check if output "xError" is FALSE
 5. Check if output "fActualPosition" is unequal to output "fActualPosition" plus input "fTargetMotionPositionDistance"
10. Condition of move relative done
 1. Check if output "xActive" is TRUE
 2. Check if output "xInOperation" is FALSE
 3. Check if output "xSystemError" is FALSE
 4. Check if output "xError" is FALSE
 5. Check if output "fActualPosition" is equal to output "fActualPosition" plus input "fTargetMotionPositionDistance"

Load program

1. Precondition:
 1. Input "xEnable" is TRUE
 2. Output "xActive" is TRUE
 3. Output "xInOperation" is FALSE
 4. Output "xSystemError" is FALSE
 5. Output "xError" is FALSE
2. Set input "uwSelectedProgramNumber" to your desired program number
3. Set input "xLoadProgram" to TRUE
4. During operation output "xInOperation" is TRUE
5. Condition of program loaded
 1. Check if output "xActive" is TRUE
 2. Check if output "xInOperation" is FALSE
 3. Check if output "xSystemError" is FALSE
 4. Check if output "xError" is FALSE
 5. Check if output "uwLoadedProgramNumber" is input "uwSelectedProgramNumber"

Start press process

1. Precondition:
 1. Input "xEnable" is TRUE
 2. Output "xActive" is TRUE
 3. Output "xInOperation" is FALSE
 4. Output "xSystemError" is FALSE

5. Output "xError" is FALSE
 6. Output "xSystemPressReady" is TRUE
2. Check if output "uwLoadedProgramNumber" is set to your desired program number
3. Set input "xAbort" to TRUE
4. Set input "xStartPressProcess" to TRUE
5. During operation output "xInOperation" is TRUE
6. Press process can be aborted via set input "xAbort" to FALSE
7. Condition of press process aborted
 1. Check if output "xActive" is TRUE
 2. Check if output "xInOperation" is FALSE
 3. Check if output "xSystemError" is FALSE
 4. Check if output "xError" is FALSE
 5. Check if output "xResultNOK" is TRUE
8. Condition of press process done
 1. Check if output "xActive" is TRUE
 2. Check if output "xInOperation" is FALSE
 3. Check if output "xSystemError" is FALSE
 4. Check if output "xError" is FALSE
 5. Check if output "xResultOK" is TRUE or output "xResultNOK" is TRUE

Record reference curve

1. Precondition:
 1. Input "xEnable" is TRUE
 2. Output "xActive" is TRUE
 3. Output "xInOperation" is FALSE
 4. Output "xSystemError" is FALSE
 5. Output "xError" is FALSE
 6. Output "xSystemPressReady" is TRUE
2. You have to be in the window "Step 2/4: Record / loading reference curve(s)" in the Festo system
3. Set input "xAbort" to TRUE
4. Set input "xRecordReferenceCurve" to TRUE
5. During operation output "xInOperation" is TRUE
6. Record reference curve can be aborted via set input "xAbort" to FALSE
7. Condition of record reference curve aborted or done
 1. Check if output "xActive" is TRUE
 2. Check if output "xInOperation" is FALSE
 3. Check if output "xSystemError" is FALSE
 4. Check if output "xError" is FALSE

Step mode

1. Precondition:
 1. Input "xEnable" is TRUE
 2. Output "xActive" is TRUE
 3. Output "xInOperation" is FALSE
 4. Output "xSystemError" is FALSE
 5. Output "xError" is FALSE
 6. Output "xSystemPressReady" is TRUE
2. Check if output "uwLoadedProgramNumber" is set to your desired program number
3. Set input "xAbort" to TRUE
4. Set input "xStepMode" to TRUE
5. Check if output "xStepModeActive" is TRUE
6. Set input "xStartPressProcess" to TRUE
7. During operation output "xInOperation" is TRUE
8. During step processing output "xStepDone" is FALSE
9. Press process can be aborted via set input "xAbort" to FALSE
10. Condition of step done
 1. Check if output "xActive" is TRUE
 2. Check if output "xStepDone" is TRUE
 3. Check if output "xSystemError" is FALSE
 4. Check if output "xError" is FALSE
11. Set input "xStartPressProcess" to FALSE
12. Set input "xStartPressProcess" to TRUE

13. During step processing output "xStepDone" is FALSE
14. Condition of step done
 1. Check if output "xActive" is TRUE
 2. Check if output "xStepDone" is TRUE
 3. Check if output "xSystemError" is FALSE
 4. Check if output "xError" is FALSE
15. Repeat step 12 to 15 until all steps are process and condition of press process done is reached
16. Condition of press process aborted
 1. Check if output "xActive" is TRUE
 2. Check if output "xInOperation" is FALSE
 3. Check if output "xSystemError" is FALSE
 4. Check if output "xError" is FALSE
 5. Check if output "xResultNOK" is TRUE
17. Condition of press process done
 1. Check if output "xActive" is TRUE
 2. Check if output "xInOperation" is FALSE
 3. Check if output "xSystemError" is FALSE
 4. Check if output "xError" is FALSE
 5. Check if output "xResultOK" is TRUE or output "xResultNOK" is TRUE

Quit system error

1. Precondition:
 1. Input "xEnable" is TRUE
 2. Output "xActive" is TRUE
 3. Output "xSystemError" is TRUE
 4. Output "xInOperation" is FALSE
 5. Output "xError" is FALSE
2. Acknowledge the error via the input "xQuitSystemError"

Function block error reaction

1. Precondition:
 1. Input "xEnable" is TRUE
 2. Output "xActive" is FALSE
 3. Output "xError" is TRUE
 4. Output "wErrorID" contains detailed error information
For further information see error table
2. Set input "xEnable" to FALSE
3. Fix the error

3.2.2 FB_ParameterChannel

This function block provides read/write access to the specified objects of the Festo system.
For detailed information of the available objects use the application note “YJKP – Object directory”.

Pre-Action:

- General:
 - Map the array of the input data (hardware ID of the hardware I/O module) to the function block input "auwInputData"
 - Map the array of the output data (hardware ID of the hardware I/O module) to the function block input "auwOutputData"

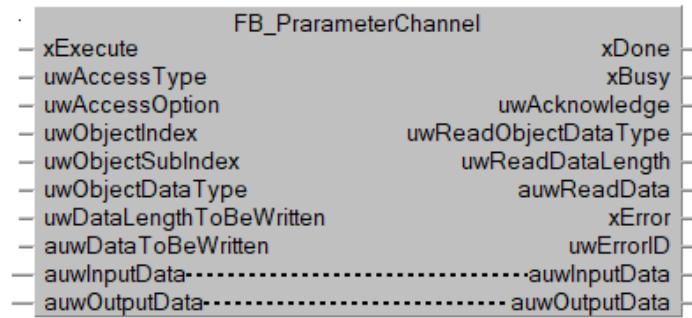


Figure 3-3: FB_ ParameterChannel

In/Out		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
Inputs		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
xExecute	BIT	Start action for read/write object Falling edge: Resets outputs Rising edge: Action start
uwAccessType	Word[Un- signed]/Bit String[16-bit]	Access type 0x00 = Read object 0x01 = Write object
uwAccessOption	Word[Un- signed]/Bit String[16-bit]	Access option 0x00 = Single read/write object 0x01 = Multi read/write object
uwObjectIndex	Word[Un- signed]/Bit String[16-bit]	Object index
uwObjectSubIndex	Word[Un- signed]/Bit String[16-bit]	Object sub index
uwObjectDataType	Word[Un- signed]/Bit String[16-bit]	Object data type
uwDataLengthToBeWritten	Word[Un- signed]/Bit String[16-bit]	Data length of the value to be written
auwDataToBeWritten	Word[Un- signed]/Bit String[16- bit](1..76)	Data bytes of the value to be written
In/Out		
auwInputData	Word[Un- signed]/Bit String[16- bit](1..64)	Start address of input data Hardware ID of the module from which the data is to be read
auwOutputData	Word[Un- signed]/Bit String[16- bit](1..64)	Start address of output data Hardware ID of the module to which the data is to be written
Outputs		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
xDone	BIT	Function block is done FALSE: Function block is not done TRUE: Function block is done
xBusy	BIT	Function block is busy FALSE: Function block is not busy TRUE: Function block is busy
uwAcknowledge	Word[Un- signed]/Bit String[16-bit]	Acknowledge
uwReadObjectDataType	Word[Un- signed]/Bit String[16-bit]	Read object data type

In/Out		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
udwReadDataLength	Double Word[Un-signed]/Bit String[32-bit]	Data length of the read value
auwReadData	Word[Un-signed]/Bit String[16-bit](1..76)	Data bytes of the read value
xError	BIT	Error occurred during action FALSE: No error TRUE: Error occurred, function block aborts action
wErrorID	Word[Un-signed]/Bit String[16-bit]	Contains detailed error information in case of an error For further information see error table

Table 3-8: FB_ ParameterChannel

Error table	
<i>Error ID</i>	<i>Description</i>
0x0000	No error No error information available
0x0001	Invalid state of the main state machine
0x0002	Invalid state of the in operation state machine
0x0003	Invalid input data
0x0004	Invalid output data

Table 3-9: Error table FB_ ParameterChannel

Examples:**Write variable (Single)**

1. Set input "uwAccessType" to 0x01 (Write)
2. Set input "uwAccessOption" to 0x00 (Single)
3. Set input "uwObjectIndex" 0x2300 (Variables 1-10)
4. Set input "uwObjectSubIndex" 0x01 (Variable 1)
5. Set input "uwObjectDataType" 0x06 (DINT)
6. Set input "uwDataLengthToBeWritten" 0x04 (4 Bytes)
7. Set data to be written to input "auwDataToBeWritten" (Data to be written starts at input "auwDataToBeWritten[1]" with a length of 4 Bytes)
8. Set input "xExecute" to True
9. Condition of write variable done
 1. Check if output "xDone" is TRUE
 2. Check if output "xBusy" is FALSE
 3. Check if output "uwAcknowledge" is 0x00
 4. Check if output "xError" is FALSE

Read variable (Single)

1. Set input "uwAccessType" to 0x00 (Read)
2. Set input "uwAccessOption" to 0x00 (Single)
3. Set input "uwObjectIndex" 0x2300 (Variables 1-10)
4. Set input "uwObjectSubIndex" 0x01 (Variable 1)
5. Set input "uwObjectDataType" 0x06 (DINT)
6. Set input "xExecute" to True
7. Condition of read variable done
 1. Check if output "xDone" is TRUE
 2. Check if output "xBusy" is FALSE
 3. Check if output "uwAcknowledge" is 0x00
 4. Check if output "xError" is FALSE
 5. Check if output "uwReadObjectDataType" is 0x06 (DINT)
 6. Check if output "uwReadDataLength" is 0x04 (4 Bytes)
 7. Read data is in output "auwReadData" (Read data starts at auwReadData[1] with a length of 4 Bytes)

Write variable (Multi)

1. Set input "uwAccessType" to 0x01 (Write)
2. Set input "uwAccessOption" to 0x01 (Multi)
3. Set input "uwObjectIndex" 0x2300 (Variables 1-10)
4. Set input "uwObjectSubIndex" 0x00 (If you use a object sub index unequal zero the object data type "uwObjectDataType" must be 0x06 (DINT))
5. Set input "uwObjectDataType" 0x07 (UDINT)
6. Set input "uwDataLengthToBeWritten" 0x2C (44 Bytes)
7. Set data to be written to input "auwDataToBeWritten"
 1. The first four bytes of the array are the data length and are not used
 2. Variable 1 starts at auwDataToBeWritten[5] with a length of 4 Bytes
 3. ...
 4. Variable 10 starts at auwDataToBeWritten[41] with a length of 4 bytes
8. Set input "xExecute" to True
9. Condition of write variable done
 1. Check if output "xDone" is TRUE
 2. Check if output "xBusy" is FALSE
 3. Check if output "uwAcknowledge" is 0x00
 4. Check if output "xError" is FALSE

Read variable (Multi)

1. Set input "uwAccessType" to 0x00 (Read)
2. Set input "uwAccessOption" to 0x01 (Multi)
3. Set input "uwObjectIndex" 0x2300 (Variables 1-10)

4. Set input "uwObjectSubIndex" 0x00 (If you use a object sub index unequal zero the object data type "uwObjectDataType" and "uwReadObjectDataType" must be 0x06 (DINT))
5. Set input "uwObjectDataType" 0x07 (UDINT)
6. Set input "xExecute" to True
7. Condition of read variable done
 1. Check if output "xDone" is TRUE
 2. Check if output "xBusy" is FALSE
 3. Check if output "uwAcknowledge" is 0x00
 4. Check if output "xError" is FALSE
 5. Check if output "uwReadObjectDataType" is 0x07 (UDINT)
 6. Check if output "uwReadDataLength" is 0x2C (44 Bytes)
 7. Read data is in output "auwReadData"
 1. The first four bytes of the array are the data length
8. Variable 1 starts at auwReadData [5] with a length of 4 Bytes
9. ...
10. Variable 10 starts at auwReadData [41] with a length of 4 bytes

Function block error reaction

1. Precondition:
 1. Input "xExecute" is TRUE
 2. Output "xBusy" is FALSE
 3. Output "xError" is TRUE
 4. Output "wErrorID" contains detailed error information
For further information see error table
2. Set input " xExecute " to FALSE
3. Fix the error

3.3 Enumerations

3.3.1 EN_AccessType

This enumeration contains the possible access type to a object.

Enumera- tion	Word[Un- signed]/Bit String[16- bit]	
<i>Name</i>	<i>Value</i>	<i>Description</i>
READ	0x00	Read object
WRITE	0x01	Write object

Table 3-2: EN_AccessType

3.3.2 EN_AccessOption

This enumeration contains the possible access option to a object.

Enumera- tion	Word[Un- signed]/Bit String[16- bit]	
<i>Name</i>	<i>Value</i>	<i>Description</i>
SINGLE	0x00	Single read/write object
MULTI	0x01	Multi read/write object

Table 3-3: EN_AccessOption

3.3.3 EN_PosMode

This enumeration contains the possible position modes.

Enumeration	Word[Unsigned]/Bit String[16-bit]	
<i>Name</i>	<i>Value</i>	<i>Description</i>
JOG	0x00	Jog
ABSOLUTE	0x01	Move absolute
RELATIVE	0x02	Move relative

Table 3-4: EN_PosMode

3.3.4 EN_Acknowledge

This enumeration contains the possible acknowledge of a response.

Enumeration	Word[Unsigned]/Bit String[16-bit]	
<i>Name</i>	<i>Value</i>	<i>Description</i>
OK	0x00	OK
SERVICE_IS_NOT_SUPPORTED	0x01	Service is not supported -> Check service ID in the request data
DATA_LENGTH_INVALID	0x02	Data length invalid -> Check data length in the request data
OBJECT_INDEX_INVALID	0x03	Object index invalid -> Check index and sub index of the object in the request data
ACCESS_TYPE_INVALID	0x04	Access type invalid -> Check access type of the object in the request data
DATA_TYPE_INVALID	0x05	Data type invalid -> Check data type of the object in the request data
VALUE_INVALID	0x06	Value invalid -> Check range of values of the object in the request data
READ_TYPE_INVALID	0x07	Read type invalid -> Check read type in the request data
WRITE_TYPE_INVALID	0x08	Write type invalid -> Check write type in the request data
VALUES_INVALID	0x09	Values invalid -> Check range of values of the object in the request data

Table 3-5: EN_Acknowledge

3.3.5 EN_DataType

This enumeration contains the possible data type of a object.

Enumeration	Word[Unsigned]/Bit String[16-bit]	
<i>Name</i>	<i>Value</i>	<i>Description</i>
UNKNOWN	0x00	Unknown data type

Enumeration	Word[Un- signed]/Bit String[16- bit]	
<i>Name</i>	<i>Value</i>	<i>Description</i>
TYPE_BOOL	0x01	8 bit boolean (0 (FALSE) ... 1 (TRUE))
TYPE_SINT	0x02	8 bit signed short integer (-128 ... 127)
TYPE_USINT	0x03	8 bit unsigned short integer (0 ... 255)
TYPE_INT	0x04	16 bit signed integer (-32768 ... 32767)
TYPE_UINT	0x05	16 bit unsigned integer (0 ... 65535)
TYPE_DINT	0x06	32 bit signed long integer (-2147483648 ... 2147483647)
TYPE_UDINT	0x07	32 bit unsigned long integer (0 ... 4294967295)
TYPE_BYTE	0x08	8 bit unsigned short integer (0 ... 255)
TYPE_WORD	0x09	16 bit unsigned integer (0 ... 65535)
TYPE_DWORD	0x0A	32 bit unsigned long integer (0 ... 4294967295)
TYPE_REAL	0x0B	32 bit float (1.400e-45 ... 3.403e+38)

Table 3-6: EN_DataType