

### **Servo Press Kit YJKP - Description of host function blocks in Allen Bradley (Studio 5000 Logix Designer V26.01)**

This application note describes how you use the host function blocks of the servo press kit YJKP in Allen Bradley (Studio 5000 Logix Designer V26.01).

YJKP

Supported systems:

- Tested with CompactLogix (1769-L24ER-QB1B)
- All compatible devices

Title .. Servo Press Kit YJKP - Description of host function blocks in Allen Bradley Rockwell Studio 5000 (Logix Designer V26.01)  
Version ..... 1.10  
Document no. .... 100161  
Original .....en  
Author .....Festo  
  
Last saved ..... 17.11.2017

## Copyright Notice

This documentation is the intellectual property of Festo AG & 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 AG & Co. KG.

Festo AG & 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 AG & Co. KG.

Festo AG & 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 AG & Co. KG, which can be found at [www.festo.com](http://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](http://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

<b>1</b>	<b>Components/Software used .....</b>	<b>5</b>
<b>2</b>	<b>Application description.....</b>	<b>6</b>
<b>3</b>	<b>Elements in the library .....</b>	<b>7</b>
3.1	Overview .....	7
3.2	Workflow .....	8
3.3	Enumerations .....	9
3.3.1	EN_ComMode .....	9
3.3.2	EN_AccessType .....	9
3.3.3	EN_AccessOption.....	9
3.3.4	EN_PosMode.....	9
3.3.5	EN_Acknowledge .....	10
3.3.6	EN_DataType .....	10
3.3.7	EN_MotionMode .....	11
3.3.8	EN_ServiceID .....	11
3.4	Structures.....	12
3.4.1	ST_SystemDataRef .....	12
3.4.2	ST_ConnectionSettings.....	13
3.4.3	ST_AcyclicReadRequest.....	14
3.4.4	ST_AcyclicReadResponse .....	15
3.4.5	ST_AcyclicWriteRequest .....	16
3.4.6	ST_AcyclicWriteResponse .....	17
3.4.7	ST_CyclicRequest.....	18
3.4.8	ST_CyclicResponse .....	20
3.5	Function blocks .....	22
3.5.1	FB_MapInputData .....	22
3.5.2	FB_MapOutputData .....	23
3.5.3	FB_Connect.....	24
3.5.4	FB_ReadWriteObject.....	27
3.5.5	FB_SystemSettings .....	32
3.5.6	FB_Manual .....	37
3.5.7	FB_PressControl.....	48
3.5.8	FB_Status.....	59



## 1 Components/Software used

Type/Name	Version Software/Firmware	Date of manufacture
Servo press kit YJKP	general	--
Application software YJKP (GSAY-A4-F0-Z4-1.2.1)	V1.2.1	--
Firmware controller (CECC-X)	V3.4.6	--
Firmware motor controller (CMMP-AS)	V4.0.1501.2.4	--
Allen Bradley (Studio 5000 Logix Designer)	V26.01	--

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 Allen Bradley (Studio 5000 Logix Designer V26.01).

Supported systems:

- Tested with CompactLogix (1769-L24ER-QB1B)
- All compatible devices

Supported fieldbus:

- EtherNet/IP

Following descriptions are part of the application note:

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

Content of the download package:

- [Link to the Application Notes](#)
- Example
- Source code

### 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.

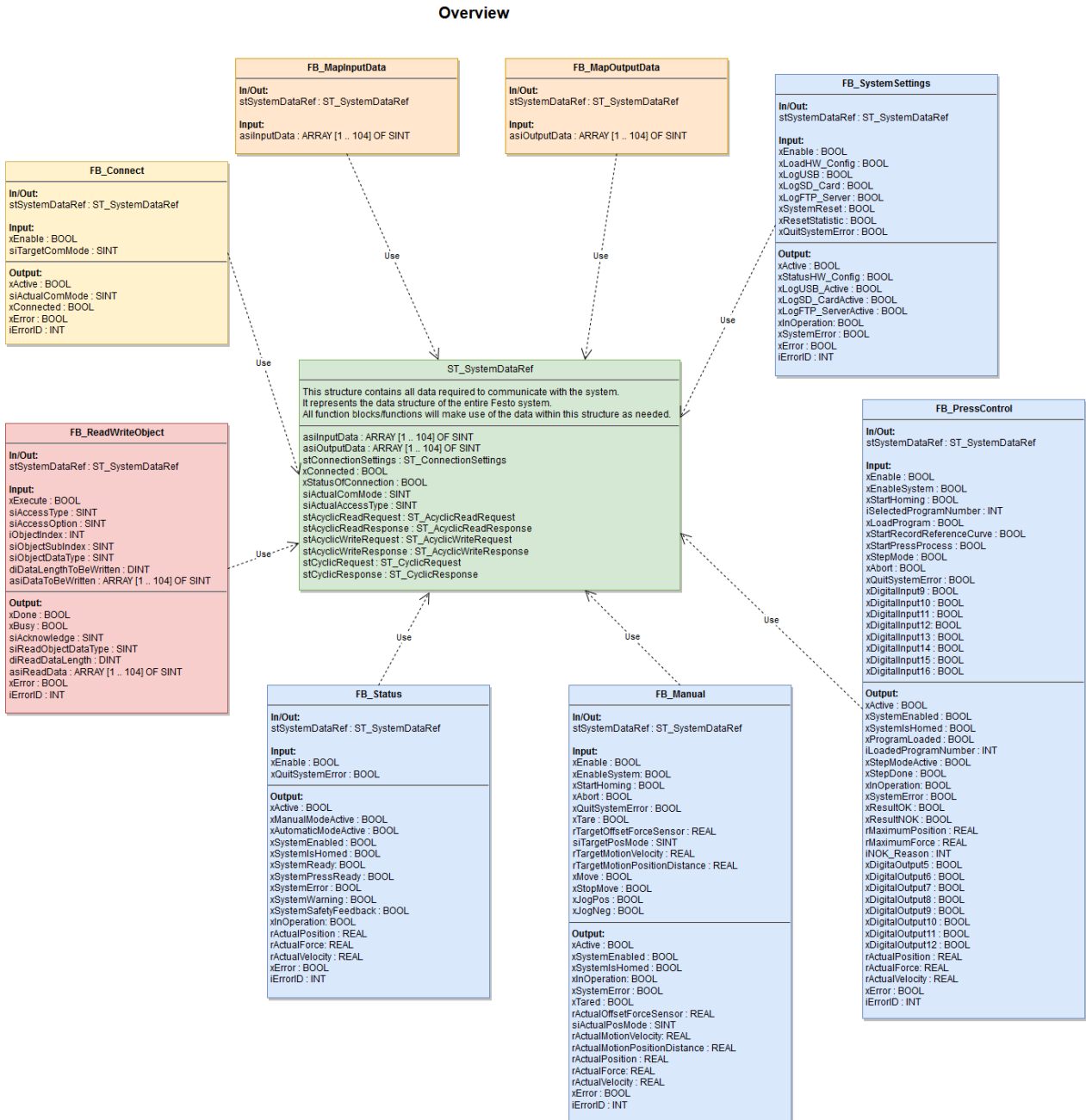


Figure 3-1: Overview

### 3.2 Workflow

The Figure 3-2: Workflow shows the required workflow of the host function blocks.

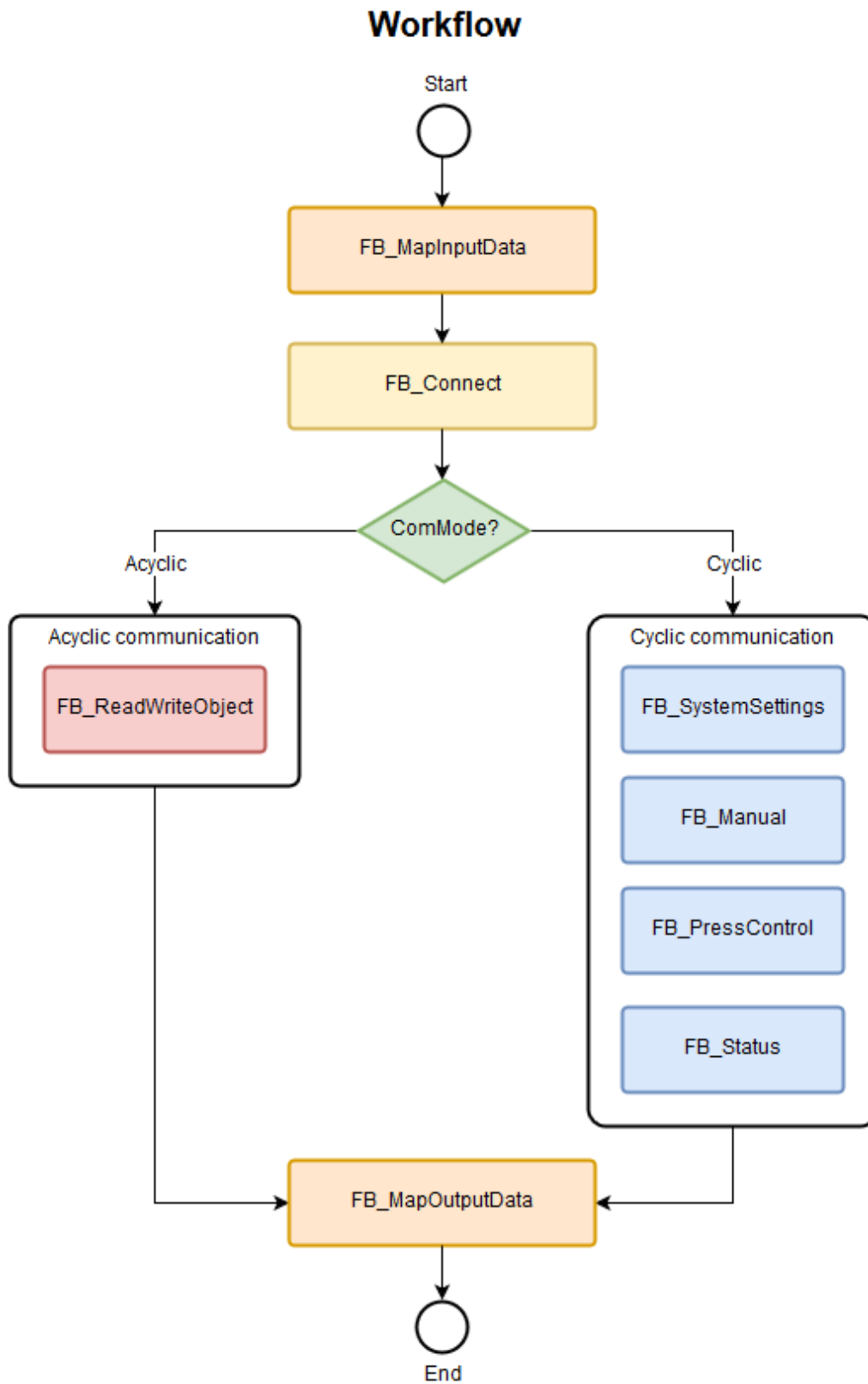


Figure 3-2: Workflow



### 3.3 Enumerations

#### 3.3.1 EN\_ComMode

This enumeration contains the possible communication modes.

Enumeration	SINT	
<i>Name</i>	<i>Value</i>	<i>Description</i>
ACYCLIC	0x00	Acyclic communication mode
CYCLIC	0x01	Cyclic communication mode

Table 3-1: EN\_ComMode

#### 3.3.2 EN\_AccessType

This enumeration contains the possible access type to a object.

Enumeration	SINT	
<i>Name</i>	<i>Value</i>	<i>Description</i>
READ	0x00	Read object
WRITE	0x01	Write object

Table 3-2: EN\_AccessType

#### 3.3.3 EN\_AccessOption

This enumeration contains the possible access option to a object.

Enumeration	SINT	
<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

#### 3.3.4 EN\_PosMode

This enumeration contains the possible position modes.

Enumeration	SINT	
<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

### 3.3.5 EN\_Acknowledge

This enumeration contains the possible acknowledge of a response.

Enumeration	SINT	
<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

### 3.3.6 EN\_DataType

This enumeration contains the possible data type of a object.

Enumeration	SINT	
<i>Name</i>	<i>Value</i>	<i>Description</i>
UNKNOWN	0x00	Unknown data type
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

### 3.3.7 EN\_MotionMode

This enumeration contains the possible motion modes.

Enumeration	SINT	
<i>Name</i>	<i>Value</i>	<i>Description</i>
JOG	0x00	Jog
ABSOLUTE	0x01	Move absolute
RELATIVE	0x02	Move relative

Table 3-7: EN\_MotionMode

### 3.3.8 EN\_ServiceID

This enumeration contains the possible service ID for a request.

Enumeration	SINT	
<i>Name</i>	<i>Value</i>	<i>Description</i>
UNKNOWN	0x00	Unknown
ACYCLIC_READ	0x10	Acyclic read request
ACYCLIC_WRTIE	0x11	Acyclic write request
CYCLIC	0x12	Cyclic request

Table 3-8: EN\_ServiceID

### 3.4 Structures

#### 3.4.1 ST\_SystemDataRef

This structure contains all data required to communicate with the system. It represents the data structure of the entire Festo system. All function blocks/functions will make use of the data within this structure as needed.

<b>Structure</b>		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
asiInputData	ARRAY [1 .. 104] OF SINT	This array contains the mapped raw input data
asiOutputData	ARRAY [1 .. 104] OF SINT	This array contains the mapped raw output data
stConnectionSettings	ST_ConnectionSettings	This structure contains all data required for the connection
xConnected	BOOL	Connection established FALSE = No connection established TRUE = Connection established
xStatusOfConnection	BOOL	Status of the communication module FALSE = No connection established TRUE = Connection established
siActualComMode	SINT	Actual communication mode 0x00 = Acyclic communication mode 0x01 = Cyclic communication mode
siActualAccessType	SINT	Actual access type 0x00 = Read object 0x01 = Write object
stAcyclicReadRequest	ST_AcyclicReadRequest	This structure contains all data required to create a acyclic read request
stAcyclicReadResponse	ST_AcyclicReadResponse	This structure contains all data required to parse a acyclic read response
stAcyclicWriteRequest	ST_AcyclicWriteRequest	This structure contains all data required to create a acyclic write request
stAcyclicWriteResponse	ST_AcyclicWriteResponse	This structure contains all data required to parse a acyclic write response
stCyclicRequest	ST_CyclicRequest	This structure contains all data required to create a cyclic request
stCyclicResponse	ST_CyclicResponse	This structure contains all data required to parse a cyclic response

Table 3-9: ST\_SystemDataRef

### 3.4.2 ST\_ConnectionSettings

This structure contains all data required for the connection.

<b>Structure</b>		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
tTimeout	DINT	Timeout Timeout of the connection Unit [ms]

Table 3-10: ST\_ConnectionSettings

### 3.4.3 ST\_AcyclicReadRequest

This structure contains all data required to create a acyclic read request.

<b>Structure</b>		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
siServiceID	SINT	Service ID 0x10 = Read object acyclic
diMessageID	DINT	Message ID Free usable message ID of the application Can be set from the client, in the related answer the same message ID will be send to the client A unique assignment of the request and response is possible TCP/IP: Can be set to zero in this request. (Handshake: Automatically) Modbus TCP: Toogle or count for handshake (Handshake: Manual) EtherNet/IP: Toogle or count for handshake (Handshake: Manual) PROFINET IO: Toogle or count for handshake (Handshake: Manual)
diDataLength	DINT	Data length Must be 0x04 in this request (Object index, Sub index, Reserved) (Data length without header)
siAcknowledge	SINT	Acknowledge Can be set to 0x00 in this request
diReadType	DINT	Read type 0x00 = Single read object Read object with one sub index (e.g. 0x00) -> Read object value Read object with more sub index (e.g. 0x00-0x02) and read sub index is 0x00 -> Read object value (Object value is data size of all objects) Read object with more sub index (e.g. 0x00-0x02) and read sub index is unequal 0x00 -> Read object value 0x01 = Multi read object Read object with one sub index (e.g. 0x00) -> Read object value Read object with more sub index (e.g. 0x00-0x02) and read sub index is 0x00 -> Read all object values Read object with more sub index (e.g. 0x00-0x02) and read sub index is unequal 0x00 -> Read all object values
iIndex	INT	Index Index of the object to be read
siSubIndex	SINT	Sub index Sub index of the object to be read
siReserved	SINT	Reserved Can be set to 0x00 in this request

Table 3-11: ST\_AcyclicReadRequest

### 3.4.4 ST\_AcyclicReadResponse

This structure contains all data required to parse a acyclic read response.

<b>Structure</b>		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
siServiceID	SINT	Service ID Service ID related to the request sent by the client
diMessageID	DINT	Message ID Message ID related to the request sent by the client
diDatalength	DINT	Data length 0x04 (Object index, Sub index, Data type) + data length of the Object to be read. (Data length without header) e.g. for reading a object UINT (2 Bytes), 0x04 + 0x02 = 0x06 will be sent
siAcknowledge	SINT	Acknowledge 0x00 if the access was OK or value unequal 0x00 if not See acknowledge description
diReadType	DINT	Read type Read type related to the request sent by the client
ilIndex	INT	Index Index of the object to be read
siSubIndex	SINT	Sub index Sub index of the object to be read
siDataType	SINT	Data type Data type of the object to be read

Table 3-12: ST\_AcyclicReadResponse

### 3.4.5 ST\_AcyclicWriteRequest

This structure contains all data required to create a acyclic write request.

<b>Structure</b>		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
siServiceID	SINT	Service ID 0x11 = Write object acyclic
diMessage ID	DINT	Message ID Free usable message ID of the application Can be set from the client, in the related answer the same message ID will be send to the client A unique assignment of the request and response is possible TCP/IP: Can be set to zero in this request. (Handshake: Automatically) Modbus TCP: Toogle or count for handshake (Handshake: Manual) EtherNet/IP: Toogle or count for handshake (Handshake: Manual) PROFINET IO: Toogle or count for handshake (Handshake: Manual)
diDataLength	DINT	Data length 0x04 (Object index, Sub index, Data type) + data length of the Object to be written (Data length without header) e.g. for writing a object UINT (2 Bytes), 0x04 + 0x02 = 0x06 will be sent
siAcknowledge	SINT	Acknowledge Can be set to 0x00 in this request
diWriteType	DINT	Write type 0x00 = Single write object Write object with one sub index (e.g. 0x00) -> Write object value Write object with more sub index (e.g. 0x00-0x02) and write sub index is 0x00 -> Read-only object, not writeable (Object value is data size of all objects) Write object with more sub index (e.g. 0x00-0x02) and write sub index is unequal 0x00 -> Write object value 0x01 = Multi write object Write object with one sub index (e.g. 0x00) -> Write object value Write object with more sub index (e.g. 0x00-0x02) and write sub index is 0x00 -> Write all object values Write object with more sub index (e.g. 0x00-0x02) and write sub index is unequal 0x00 -> Write all object values
iIndex	INT	Index Index of the object to be written
siSubIndex	SINT	Sub index Sub Index of the object to be written
siDataType	SINT	Data type Data type of the object to be written
asiDataByte	ARRAY [1 .. 76] OF SINT	Data bytes Data bytes of the value to be written

Table 3-13: ST\_AcyclicWriteRequest



### 3.4.6 ST\_AcyclicWriteResponse

This structure contains all data required to parse a acyclic write response.

<b>Structure</b>		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
siService ID	SINT	Service ID Service ID related to the request sent by the client
diMessage ID	DINT	Message ID Message ID related to the request sent by the client
diData length	DINT	Data length Must be 0x04 in this response (Object index, Sub index, Data type) (Data length without header)
siAcknowledge	SINT	Acknowledge 0x00 if the access was OK or value unequal 0x00 if not See acknowledge description
diWriteType	DINT	Write type Write type related to the request sent by the client
iIndex	INT	Index Index of the object to be written
siSubIndex	SINT	Sub Index Sub Index of the object to be written
siDataType	SINT	Data type Data type of the object to be written

Table 3-14: ST\_AcyclicWriteResponse

### 3.4.7 ST\_CyclicRequest

This structure contains all data required to create a cyclic request.

<b>Structure</b>		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
siServiceID	SINT	Service ID 0x12 = Write/Read objects cyclic
iDataLength	INT	Data length Must be 0x15 in this request (Service ID, Data length, Acknowledge, Message ID) (Data length without header)
siAcknowledge	SINT	Acknowledge Can be set to 0x00 in this request
siMessageID	SINT	Message ID Free usable message ID of the application Can be set from the client, in the related answer the same message ID will be send to the client A unique assignment of the request and response is possible TCP/IP: Can be set to zero in this request. (Handshake: Automatically) Modbus TCP: Toogle or count for handshake (Handshake: Manual) EtherNet/IP: Toogle or count for handshake (Handshake: Manual) PROFINET IO: Toogle or count for handshake (Handshake: Manual)
diControlWord	DINT	Control word Bit 0 = Manual mode Bit 1 = Automatic mode Bit 2 = Start homing Bit 3 = Start press process Bit 4 = Abort press process (low active) Bit 5 = Load program Bit 6 = Quit error Bit 7 = Enable press Bit 8 = Tare Bit 9 = Move Bit 10 = Stop move Bit 11 = Jog pos Bit 12 = Jog neg Bit 13 = Load hardware configuration Bit 14 = Logging USB Bit 15 = Logging SD-Card Bit 16 = System reset Bit 17 = Reset statistic Bit 18 = Step mode Bit 19 = Logging FTP-Server Bit 20-31 = Reserved
siComMode	SINT	Communication mode 0x00 = Acyclic communication mode 0x01 = Cyclic communication mode
diOffsetForceSensor	DINT	Offset of the force sensor Unit [N] $1 * 10^{-2}$ (2 decimal places, e.g. 1 = 0,01 N)
siMotionMode	SINT	Motion mode (Jog, Move absolute/relative) 0x00 = Jog 0x01 = Move absolute 0x02 = Move relative

<b>Structure</b>		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
diMotionVelocity	DINT	Motion velocity (Jog, Move absolute/relative) Unit [mm/s] $1 * 10^{-2}$ (2 decimal places, e.g. 1 = 0,01 mm/s)
diMotionPositionDistance	DINT	Motion position/distance (Move absolute/relative) Unit [mm] $1 * 10^{-2}$ (2 decimal places, e.g. 1 = 0,01 mm)
iSelectedProgramNumber	INT	Selected program number
siDigitalInputs	SINT	Digital inputs (Step enabling conditions in the sequencer of the press process) Bit 0 = Input 9 Bit 1 = Input 10 Bit 2 = Input 11 Bit 3 = Input 12 Bit 4 = Input 13 Bit 5 = Input 14 Bit 6 = Input 15 Bit 7 = Input 16

Table 3-15: ST\_CyclicRequest

### 3.4.8 ST\_CyclicResponse

This structure contains all data required to parse a cyclic response.

<b>Structure</b>		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
siServiceID	SINT	Service ID Service ID related to the request sent by the client
iDataLength	INT	Data length Must be 0x2B in this response (Service ID, Data length, Acknowledge, Message ID) (Data length without header)
siAcknowledge	SINT	Acknowledge 0x00 if the access was OK or value unequal 0x00 if not See acknowledge description
siMessageID	SINT	Message ID Message ID related to the request sent by the client
diStatusWord	DINT	Status word Bit 0 = Manual mode Bit 1 = Automatic mode Bit 2 = Homing required Bit 3 = Program loaded Bit 4 = Step mode Bit 5 = In operation Bit 6 = Step done Bit 7 = OK Bit 8 = NOK Bit 9 = Press enabled Bit 10 = Tared Bit 11 = Status ready Bit 12 = Status servo press ready Bit 13 = Status error Bit 14 = Status warning Bit 15 = Status safety Bit 16 = Status hardware configuration Bit 17 = Status logging USB Bit 18 = Status logging SD-Card Bit 19 = Status logging FTP-Server Bit 20-31 = Reserved
siComMode	SINT	Communication mode 0x00 = Acyclic communication mode 0x01 = Cyclic communication mode
diOffsetForceSensor	DINT	Offset of the force sensor Unit [N] $1 * 10^{-2}$ (2 decimal places, e.g. 1 = 0,01 N)
siMotionMode	SINT	Motion mode (Jog, Move absolute/relative) 0x00 = Jog 0x01 = Move absolute 0x02 = Move relative
diMotionVelocity	DINT	Motion velocity (Jog, Move absolute/relative) Unit [mm/s] $1 * 10^{-2}$ (2 decimal places, e.g. 1 = 0,01 mm/s)
diMotionPositionDistance	DINT	Motion position/distance (Move absolute/relative) Unit [mm] $1 * 10^{-2}$ (2 decimal places, e.g. 1 = 0,01 mm)
iLoadedProgramNumber	INT	Loaded program number

<b>Structure</b>		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
siDigitalOutputs	SINT	Digital outputs (Configurable as output signals in the sequencer of the press process) Bit 0 = Output 5 Bit 1 = Output 6 Bit 2 = Output 7 Bit 3 = Output 8 Bit 4 = Output 9 Bit 5 = Output 10 Bit 6 = Output 11 Bit 7 = Output 12
diActualPosition	DINT	Actual position Unit [mm] $1 * 10^{-2}$ (2 decimal places, e.g. 1 = 0,01 mm)
diActualForce	DINT	Actual force Unit [N] $1 * 10^{-2}$ (2 decimal places, e.g. 1 = 0,01 N)
diActualVelocity	DINT	Actual velocity Unit [mm/s] $1 * 10^{-2}$ (2 decimal places, e.g. 1 = 0,01 mm/s)
diMaximumPosition	DINT	Maximum position in the press process Unit [mm] $1 * 10^{-2}$ (2 decimal places, e.g. 1 = 0,01 mm)
diMaximumForce	DINT	Maximum force in the press process Unit [N] $1 * 10^{-2}$ (2 decimal places, e.g. 1 = 0,01 N)
iNOK_Reason	INT	NOK reason of the press process More detailed information see online help of the YJKP

Table 3-16: ST\_CyclicResponse

### 3.5 Function blocks

#### 3.5.1 FB\_MapInputData

This function block maps the input data "asiInputData" from the addressed hardware I/O module (CECC-X-M1) depending on the actual communication mode and access type to the depending structures.

Communication mode -> Acyclic:

- Access type -> Read object:
  - Structure "stSystemDataRef" -> stSystemDataRef.stAcyclicReadResponse
- Access type -> Write object:
  - Structure "stSystemDataRef" -> stSystemDataRef.stAcyclicWriteResponse

Communication mode -> Cyclic:

- Structure "stSystemDataRef" -> stSystemDataRef.stCyclicResponse

Pre-Action:

- General:
  - Map an instance of the structure "stSystemDataRef" to the function block input "stSystemDataRef"
  - Map the input data from the addressed hardware I/O module (CECC-X-M1) to the function block input "asiInputData"

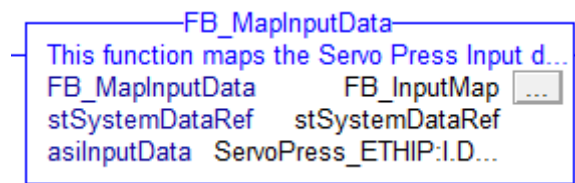


Figure 3-3: FB\_MapInputData

In/Out		
Name	Data type	Description
stSystemDataRef	ST_SystemDataRef	This data structure contains all data required to communicate with the system
asiInputData	ARRAY [1 .. 104] OF SINT	This array contains the mapped raw input data

Table 3-17: FB\_MapInputData

### 3.5.2 FB\_MapOutputData

This function block maps depending on the actual communication mode and access type the depending structures to the output data "asiOutputData" to the addressed hardware I/O module (CECC-X-M1).

Communication mode -> Acyclic:

- Access type -> Read object:
  - Structure "stSystemDataRef" -> stSystemDataRef.stAcyclicReadRequest
- Access type -> Write object:
  - Structure "stSystemDataRef" -> stSystemDataRef.stAcyclicWriteRequest

Communication mode -> Cyclic:

- Structure "stSystemDataRef" -> stSystemDataRef.stCyclicRequest

Pre-Action:

- General:
  - Map an instance of the structure "stSystemDataRef" to the function block input "stSystemDataRef"
  - Map the output data from the addressed hardware I/O module (CECC-X-M1) to the function block input "asiOutputData"

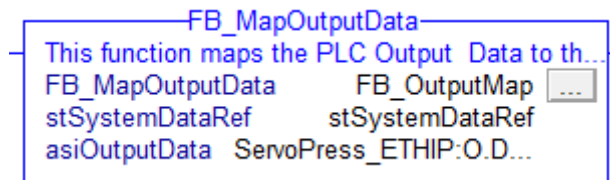


Figure 3-4: FB\_MapOutputData

In/Out		
Name	Data type	Description
stSystemDataRef	ST_SystemDataRef	This data structure contains all data required to communicate with the system
asiOutputData	ARRAY [1 .. 104] OF SINT	This array contains the mapped raw output data

Table 3-18: FB\_MapOutputData

### 3.5.3 FB\_Connect

This function block establishes a connection between the host system and the Festo system and controls the communication mode for the connection.

Pre-Action:

- General:
  - Map an instance of the structure "stSystemDataRef" to the function block input "stSystemDataRef"
  - Set timeout of the connection -> Unit [ms] in the structure "stSystemDataRef" -> stSystemDataRef.stConnectionSettings.tTimeout
  - Map the status of the connection to the structure "stSystemDataRef" -> stSystemDataRef.xStatusOfConnection (FALSE: No connection established | TRUE: Connection established)
- EtherNet/IP:
  - Map the input data from the addressed hardware I/O module (CECC-X-M1) to the function block input "asiInputData" of function block "FB\_MapInputData"
  - Map the output data from the addressed hardware I/O module (CECC-X-M1) to the function block input "asiOutputData" of function block "FB\_MapOutputData"

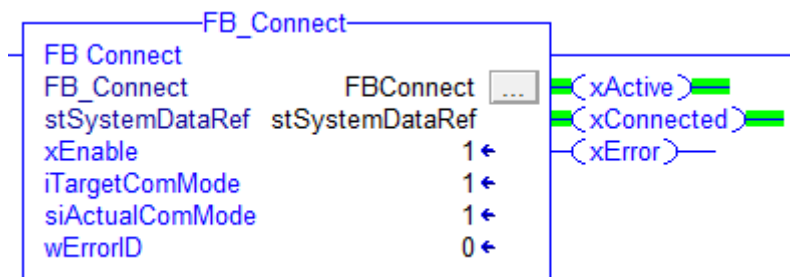


Figure 3-5: FB\_Connect

In/Out		
Name	Data type	Description
stSystemDataRef	ST_SystemDataRef	This data structure contains all data required to communicate with the system
<b>Inputs</b>		
Name	Data type	Description
xEnable	BOOL	Activate establish a connection between the host system and the Festo system and control the communication mode for the connection FALSE: Action stopped, resets outputs TRUE: Action running
siTargetComMode	SINT	Target communication mode 0x00 = Acyclic communication mode 0x01 = Cyclic communication mode
<b>Outputs</b>		
Name	Data type	Description
xActive	BOOL	Function block is active FALSE: Function block is inactive TRUE: Function block is active
siActualComMode	SINT	Actual communication mode 0x00 = Acyclic communication mode 0x01 = Cyclic communication mode



In/Out		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
xConnected	BOOL	Connection established FALSE: No connection established TRUE: Connection established
xError	BOOL	Error occurred during action FALSE: No error TRUE: Error occurred, function block aborts action
iErrorID	INT	Contains detailed error information in case of an error For further information see error table

Table 3-19: FB\_Connect

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	No connection established
0x0004	Unknown communication mode
0x0005	Timeout is exceeded
0x0006	Receive/parse acyclic write response to switch to cyclic communication mode
0x0007	Receive/parse cyclic response to switch to acyclic communication mode
0x0008	Receive/parse cyclic response

Table 3-20: Error table FB\_Connect

**Examples:****System connection (Acyclic communication mode)**

1. Set input "siTargetComMode" to 0x00 (Acyclic communication mode)
2. Set input "xEnable" to TRUE
3. Check if output "xActive" is TRUE
4. Check if output "xConnected" is TRUE
5. Check if output "siActualComMode" is 0x00 (Acyclic communication mode)
6. Check if output "xError" is FALSE

**System connection (Cyclic communication mode)**

1. Set input "siTargetComMode" to 0x01 (Cyclic communication mode)
2. Set input "xEnable" to TRUE
3. Check if output "xActive" is TRUE
4. Check if output "xConnected" is TRUE
5. Check if output "siActualComMode" is 0x01 (Cyclic communication mode)
6. Check if output "xError" is FALSE

### **System connected and switch from acyclic communication mode to cyclic communication mode**

1. Precondition:
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x00 (Acyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x00 (Acyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Set input "siTargetComMode" to 0x01 (Cyclic communication mode)
3. Check if output "xActive" is TRUE
4. Check if output "xConnected" is TRUE
5. Check if output "siActualComMode" is 0x01 (Cyclic communication mode)
6. Check if output "xError" is FALSE

### **System connected and switch from cyclic communication mode to acyclic communication mode**

1. Precondition:
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x01 (Cyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x01 (Cyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Set input "siTargetComMode" to 0x00 (Acyclic communication mode)
3. Check if output "xActive" is TRUE
4. Check if output "xConnected" is TRUE
5. Check if output "siActualComMode" is 0x00 (Acyclic communication mode)
6. Check if output "xError" is FALSE

### **Function block error reaction**

1. Precondition:
  1. Input "xEnable" is TRUE
  2. Output "xActive" is FALSE
  3. Output "xError" is TRUE
  4. Output "iErrorID" contains detailed error information  
For further information see error table
2. Set input "xEnable" to FALSE
3. Reset all other active function blocks
4. Wait more than 2s to establish a new connection to the Festo system (After internal timeout of 2s the Festo system reset internally to acyclic communication mode)
5. Proceed with system connection

### 3.5.4 FB\_ReadWriteObject

This function block provides read/write access to the specified objects of the Festo system. For detailed information of the host interface and the available objects use the application note “Servo press kit – Host interface”.

Pre-Action:

- General:
  - Map an instance of the structure "stSystemDataRef" to the function block input "stSystemDataRef"
  - Status of connection must be connection established in the function block "FB\_Connect" -> FB\_Connect.xConnected
  - Status of actual communication mode must be acyclic communication mode in the function block "FB\_Connect" -> FB\_Connect.byActualComMode

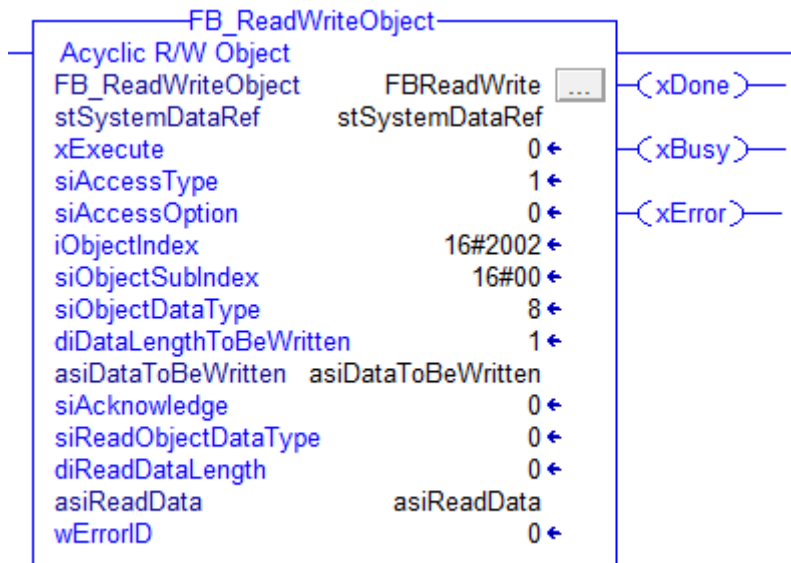


Figure 3-6: FB\_ReadWriteObject

In/Out		
Name	Data type	Description
stSystemDataRef	ST_SystemDataRef	This data structure contains all data required to communicate with the system
Inputs		
Name	Data type	Description
xExecute	BOOL	Start action for read/write object Falling edge: Resets outputs Rising edge: Action start
siAccessType	SINT	Access type 0x00 = Read object 0x01 = Write object
siAccessOption	SINT	Access option 0x00 = Single read/write object 0x01 = Multi read/write object
iObjectIndex	INT	Object index
siObjectSubIndex	SINT	Object sub index
siObjectDataType	SINT	Object data type

<b>In/Out</b>		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
diDataLengthToBeWritten	DINT	Data length of the value to be written
asiDataToBeWritten	ARRAY [1 .. 76] OF SINT SINT	Data bytes of the value to be written
<b>Outputs</b>		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
xDone	BOOL	Function block is done FALSE: Function block is not done TRUE: Function block is done
xBusy	BOOL	Function block is busy FALSE: Function block is not busy TRUE: Function block is busy
siAcknowledge	SINT	Acknowledge
siReadObjectDataType	SINT	Read object data type
diReadDataLength	DINT	Data length of the read value
asiyReadData	ARRAY [1 .. 76] OF SINT	Data bytes of the read value
xError	BOOL	Error occurred during action FALSE: No error TRUE: Error occurred, function block aborts action
iErrorID	INT	Contains detailed error information in case of an error For further information see error table

Table 3-21: FB\_ReadWriteObject

<b>Error table</b>	
<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	Unknown access type
0x0004	No connection established
0x0005	Invalid communication mode
0x0006	Timeout is exceeded
0x0007	Receive/parse acyclic read response
0x0008	Receive/parse acyclic write response

Table 3-22: Error table FB\_ReadWriteObject

**Examples:****Write variable (Single)**

1. Precondition of function block "FB\_Connect":
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x00 (Acyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x00 (Acyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Set input "siAccessType" to 0x01 (Write)
3. Set input "siAccessOption" to 0x00 (Single)
4. Set input "iObjectIndex" 0x2300 (Variables 1-10)
5. Set input "siObjectSubIndex" 0x01 (Variable 1)
6. Set input "siObjectDataType" 0x06 (DINT)
7. Set input "diDataLengthToBeWritten" 0x04 (4 Bytes)
8. Set data to be written to input "asiDataToBeWritten" (Data to be written starts at input "asiDataToBeWritten[1]" with a length of 4 Bytes)
9. Condition of write variable done
  1. Check if output "xDone" is TRUE
  2. Check if output "xBusy" is FALSE
  3. Check if output "siAcknowledge" is 0x00
  4. Check if output "xError" is FALSE

**Read variable (Single)**

1. Precondition of function block "FB\_Connect":
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x00 (Acyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x00 (Acyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Set input "siAccessType" to 0x00 (Read)
3. Set input "siAccessOption" to 0x00 (Single)
4. Set input "iObjectIndex" 0x2300 (Variables 1-10)
5. Set input "siObjectSubIndex" 0x01 (Variable 1)
6. Set input "siObjectDataType" 0x06 (DINT)
7. Condition of read variable done
  1. Check if output "xDone" is TRUE
  2. Check if output "xBusy" is FALSE
  3. Check if output "siAcknowledge" is 0x00
  4. Check if output "xError" is FALSE
  5. Check if output "siReadObjectDataType" is 0x06 (DINT)
  6. Check if output "diReadDataLength" is 0x04 (4 Bytes)
  7. Read data is in output "asiReadData" (Read data starts at asiReadData[1] with a length of 4 Bytes)

### Write variable (Multi)

1. Precondition of function block "FB\_Connect":
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x00 (Acyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x00 (Acyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Set input "siAccessType" to 0x01 (Write)
3. Set input "siAccessOption" to 0x01 (Multi)
4. Set input "iObjectIndex" 0x2300 (Variables 1-10)
5. Set input "siObjectSubIndex" 0x00 (If you use a object sub index unequal zero the object data type "si-ObjectDataType" must be 0x06 (DINT))
6. Set input "siObjectDataType" 0x07 (UDINT)
7. Set input "diDataLengthToBeWritten" 0x2C (44 Bytes)
8. Set data to be written to input "asiDataToBeWritten"
  1. The first four bytes of the array are the data length and are not used
  2. Variable 1 starts at asiDataToBeWritten[5] with a length of 4 Bytes
  3. ...
  4. Variable 10 starts at asiDataToBeWritten[41] with a length of 4 bytes
9. Condition of write variable done
  1. Check if output "xDone" is TRUE
  2. Check if output "xBusy" is FALSE
  3. Check if output "siAcknowledge" is 0x00
  4. Check if output "xError" is FALSE

### Read variable (Multi)

1. Precondition of function block "FB\_Connect":
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x00 (Acyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x00 (Acyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Set input "siAccessType" to 0x00 (Read)
3. Set input "siAccessOption" to 0x01 (Multi)
4. Set input "iObjectIndex" 0x2300 (Variables 1-10)
5. Set input "siObjectSubIndex" 0x00 (If you use a object sub index unequal zero the object data type "si-ObjectDataType" and "siReadObjectDataType" must be 0x06 (DINT))
6. Set input "siObjectDataType" 0x07 (UDINT)
7. Condition of read variable done
  1. Check if output "xDone" is TRUE
  2. Check if output "xBusy" is FALSE
  3. Check if output "siAcknowledge" is 0x00
  4. Check if output "xError" is FALSE
  5. Check if output "siReadObjectDataType" is 0x07 (UDINT)
  6. Check if output "siReadDataLength" is 0x2C (44 Bytes)
  7. Read data is in output "asiReadData"
    1. The first four bytes of the array are the data length
    2. Variable 1 starts at asiReadData [5] with a length of 4 Bytes
    3. ...
    4. Variable 10 starts at asiReadData [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 "iErrorID" contains detailed error information  
For further information see error table
2. Set input " xExecute " to FALSE
3. Fix the error

### 3.5.5 FB\_SystemSettings

This function block provides system settings for the Festo system.

Pre-Action:

- General:
  - Map an instance of the structure "stSystemDataRef" to the function block input "stSystemDataRef"
  - Status of connection must be connection established in the function block "FB\_Connect" -> FB\_Connect.xConnected
  - Status of actual communication mode must be cyclic communication mode in the function block "FB\_Connect" -> FB\_Connect.siActualComMode
  - For the input "xQuitSystemError" all function blocks (FB\_Manual, FB\_PressControl, FB\_SystemSettings and FB\_Status) have to use the same variable when they are active or all function blocks (FB\_Manual, FB\_PressControl, FB\_SystemSettings and FB\_Status) must be interlocked. Because they are all mapped to the same internal variable. If there is no mechanism implemented described before it could be that the input is overwritten by a call of another active function block (FB\_Manual, FB\_PressControl, FB\_SystemSettings and FB\_Status).

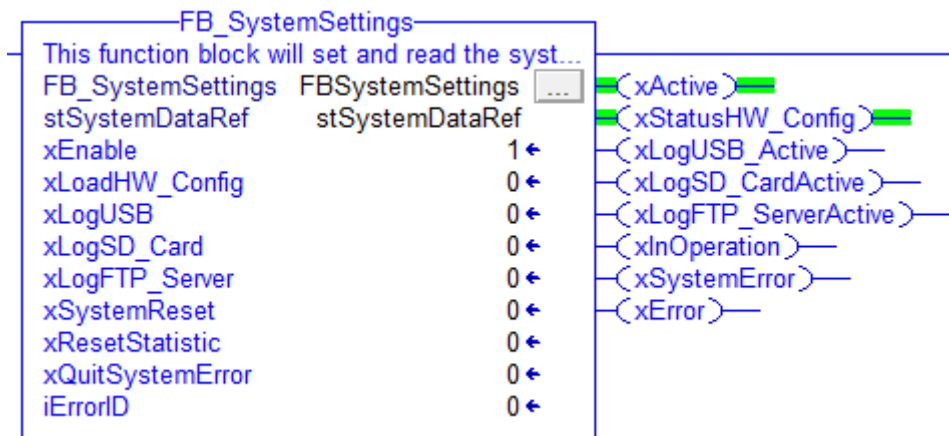


Figure 3-7: FB\_SystemSettings

In/Out		
Name	Data type	Description
stSystemDataRef	ST_SystemDataRef	This data structure contains all data required to communicate with the system
Inputs		
Name	Data type	Description
xEnable	BOOL	Activate system settings for the Festo system FALSE: Action stopped, resets outputs TRUE: Action running
xLoadHW_Config	BOOL	Load hardware configuration command Falling edge: No action Rising edge: Load hardware configuration (Configured hardware in window "Hardware configuration")
xLogUSB	BOOL	Log USB command False: Deactivate log on USB-Stick True: Activate log on USB-Stick
xLogSD_Card	BOOL	Log SD-Card command False: Deactivate log on SD-Card True: Activate log on SD-Card



<b>In/Out</b>		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
xLogFTP_Server	BOOL	Log FTP-Server command False: Deactivate log on FTP-Server True: Activate log on FTP-Server
xSystemReset	BOOL	System reset command Falling edge: No action Rising edge: System reset
xResetStatistic	BOOL	Reset statistic command Falling edge: No action Rising edge: Reset statistic (Reset statistic in window "Process diagnosis")
xQuitSystemError	BOOL	Quit system error command Falling edge: No action Rising edge: Quit upcoming system error
<b>Outputs</b>		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
xActive	BOOL	Function block is active FALSE: Function block is inactive TRUE: Function block is active
xStatusHW_Config	BOOL	Status of the hardware configuration FALSE: Hardware configuration invalid TRUE: Hardware configuration valid
xLogUSB_Active	BOOL	Log on USB is active FALSE: Log on USB-Stick is inactive TRUE: Log on USB-Stick is active
xLogSD_CardActive	BOOL	Log on SD-Card is active FALSE: Log on SD-Card is inactive TRUE: Log on SD-Card is active
xLogFTP_ServerActive	BOOL	Log on FTP-Server is active FALSE: Log on FTP-Server is inactive TRUE: Log on FTP-Server is active
xInOperation	BOOL	System is in operation FALSE: System is not in operation TRUE: System is in operation
xSystemError	BOOL	System error FALSE: No error TRUE: Error occurred
xError	BOOL	Error occurred during action FALSE: No error TRUE: Error occurred, function block aborts action
iErrorID	INT	Contains detailed error information in case of an error For further information see error table

Table 3-23: FB\_SystemSettings

<b>Error table</b>	
<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	No connection established
0x0004	Invalid communication mode
0x0005	Service is not supported
0x0006	Data length invalid
0x0007	Object index invalid
0x0008	Access type invalid
0x0009	Data type invalid
0x000A	Value invalid
0x000B	Read type invalid
0x000C	Write type invalid
0x000D	Values invalid

Table 3-24: Error table FB\_SystemSettings

**Examples:**

**Read system settings**

1. Precondition of function block "FB\_Connect":
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x01 (Cyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x01 (Cyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Set input "xEnable " to TRUE
3. All outputs will be updated according to the status of the Festo system

### Load hardware configuration

1. Precondition of function block "FB\_Connect":
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x01 (Cyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x01 (Cyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Precondition of function block "FB\_SystemSettings":
  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
3. You have to be in the window "Hardware configuration" in the WebVisu of the Festo system
4. Set in the WebVisu your desired configuration
5. Set input "xLoadHW\_Config" to TRUE
6. During operation output "xInOperation" is TRUE
7. 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
8. Save the loaded hardware configuration via the button "Save" in the WebVisu

### Activate logging (For all logging functions the same)

1. Precondition of function block "FB\_Connect":
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x01 (Cyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x01 (Cyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Precondition of function block "FB\_SystemSettings":
  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
3. Set input "xLogSD\_Card" to TRUE
4. 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 of function block "FB\_Connect":
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x01 (Cyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x01 (Cyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Precondition of function block "FB\_SystemSettings":
  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
3. Set input "xSystemReset" to TRUE
4. A reboot of the controller of the Festo system will be performed

### Reset statistic

1. Precondition of function block "FB\_Connect":
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x01 (Cyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x01 (Cyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Precondition of function block "FB\_SystemSettings":
  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
3. Set input "xResetStatistic" to TRUE
4. The statistic in the window "Process diagnosis" in the WebVisu of the Festo system will be reset

### Quit system error

1. Precondition of function block "FB\_Connect":
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x01 (Cyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x01 (Cyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Precondition of function block "FB\_SystemSettings":
  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
3. 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 "iErrorID" contains detailed error information  
For further information see error table
2. Set input "xEnable" to FALSE
3. Fix the error

### 3.5.6 FB\_Manual

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

Pre-Action:

- General:
  - Map an instance of the structure "stSystemDataRef" to the function block input "stSystemDataRef"
  - Status of connection must be connection established in the function block "FB\_Connect" -> FB\_Connect.xConnected
  - Status of actual communication mode must be cyclic communication mode in the function block "FB\_Connect" -> FB\_Connect.siActualComMode
  - For the input "xQuitSystemError" all function blocks (FB\_Manual, FB\_PressControl, FB\_SystemSettings and FB\_Status) have to use the same variable when they are active or all function blocks (FB\_Manual, FB\_PressControl, FB\_SystemSettings and FB\_Status) must be interlocked. Because they are all mapped to the same internal variable. If there is no mechanism implemented described before it could be that the input is overwritten by a call of another active function block (FB\_Manual, FB\_PressControl, FB\_SystemSettings and FB\_Status).
  - For the inputs "xEnableSystem", "xStartHoming", "xAbort" the function blocks (FB\_Manual and FB\_PressControl) have to use the same variable when they are active or the function blocks (FB\_Manual and FB\_PressControl) must be interlocked. Because they are mapped to the same internal variable. If there is no mechanism implemented described before it could be that the input is overwritten by a call of the other active function block (FB\_Manual and FB\_PressControl).

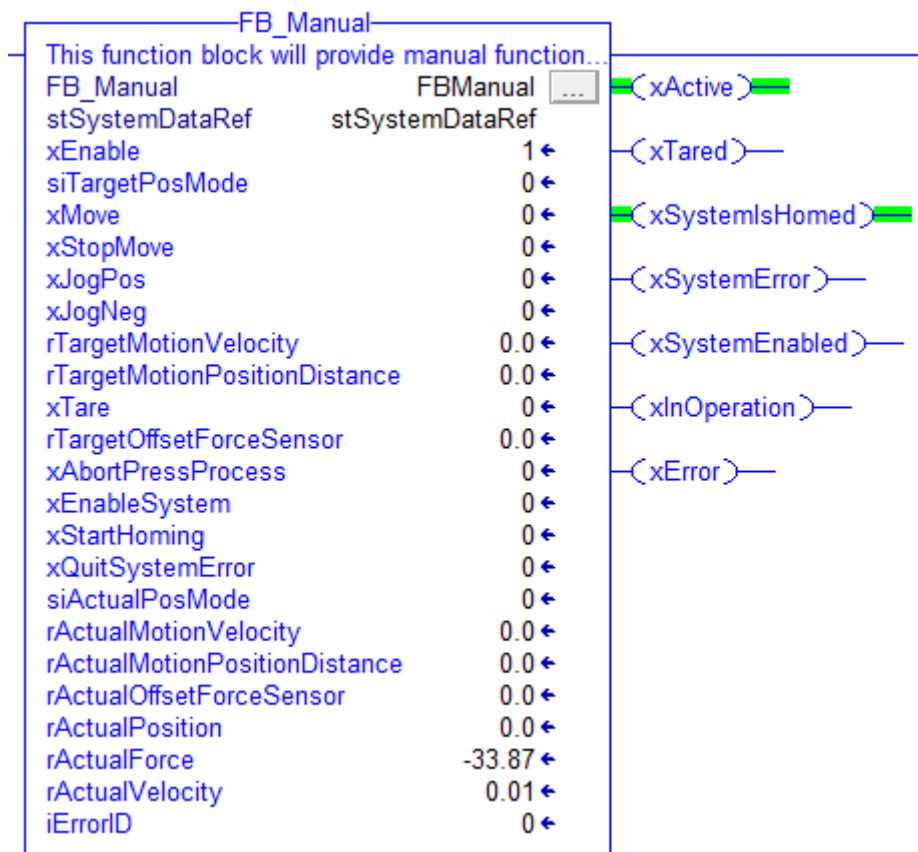


Table 3-25: FB\_Manual

<b>In/Out</b>		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
stSystemDataRef	ST_SystemDataRef	This data structure contains all data required to communicate with the system
<b>Inputs</b>		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
xEnable	BOOL	Activate manual control functions for the Festo system FALSE: Action stopped, resets outputs TRUE: Action running
xEnableSystem	BOOL	Enable system command Falling edge: Disable system Rising edge: Enable system
xStartHoming	BOOL	Homing command Falling edge: No action Rising edge: Start homing movement
xAbort	BOOL	Abort command False: Abort press process or homing True: No action
xQuitSystemError	BOOL	Quit system error command Falling edge: No action Rising edge: Quit upcoming system error
xTare	BOOL	Tare of the force sensor command Falling edge: Reset tare of the force sensor Rising edge: Tare of the force sensor
rTargetOffsetForceSensor	REAL	Target offset of the force sensor Unit [N]
siTargetPosMode	SINT	Target position mode 0x00 = Jog 0x01 = Move absolute 0x02 = Move relative
rTargetMotionVelocity	REAL	Target motion velocity for target position mode (Jog, Move absolute/relative) Unit [mm/s]
rTargetMotionPositionDistance	REAL	Target motion position/distance for target position mode (Move absolute/relative) Unit [mm]
xMove	BOOL	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	BOOL	Stop move command Falling edge: No action Rising edge: Stop movement depending on target position mode (Move absolute/relative)
xJogPos	BOOL	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

<b>In/Out</b>		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
xJogNeg	BOOL	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
<b>Outputs</b>		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
xActive	BOOL	Function block is active FALSE: Function block is inactive TRUE: Function block is active
xSystemEnabled	BOOL	System is enabled FALSE: System is disabled TRUE: System is enabled
xSystemIsHomed	BOOL	System is homed FALSE: System is not homed TRUE: System is homed
xInOperation	BOOL	System is in operation FALSE: System is not in operation TRUE: System is in operation
xSystemError	BOOL	System error FALSE: No error TRUE: Error occurred
xTared	BOOL	Status of tare of the force sensor FALSE: Force sensor not tared TRUE: Force sensor tared
rActualOffsetForceSensor	REAL	Actual offset of the force sensor Unit [N]
siActualPosMode	SINT	Actual position mode 0x00 = Jog 0x01 = Move absolute 0x02 = Move relative
rActualMotionVelocity	REAL	Actual motion velocity for actual position mode (Jog, Move absolute/relative) Unit [mm/s]
rActualMotionPositionDistance	REAL	Actual motion position/distance for actual position mode (Move absolute/relative) Unit [mm]
rActualPosition	REAL	Actual position Unit [mm]
rActualForce	REAL	Actual force Unit [N]
rActualVelocity	REAL	Actual velocity Unit [mm/s]
xError	BOOL	Error occurred during action FALSE: No error TRUE: Error occurred, function block aborts action
iErrorID	INT	Contains detailed error information in case of an error For further information see error table

Table 3-26: FB\_Manual

<b>Error table</b>	
<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	No connection established
0x0004	Invalid communication mode
0x0005	Service is not supported
0x0006	Data length invalid
0x0007	Object index invalid
0x0008	Access type invalid
0x0009	Data type invalid
0x000A	Value invalid
0x000B	Read type invalid
0x000C	Write type invalid
0x000D	Values invalid

Table 3-27: Error table FB\_Manual

**Examples:**

**Read manual status**

1. Precondition of function block "FB\_Connect":
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x01 (Cyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x01 (Cyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Set input "xEnable " to TRUE
3. All outputs will be updated according to the status of the Festo system



**Enable system**

1. Precondition of function block "FB\_Connect":
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x01 (Cyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x01 (Cyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Precondition of function block "FB\_Manual":
  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
3. Precondition of function block "FB\_Status":
  1. Input "xEnable" is TRUE
  2. Output "xActive" is TRUE
  3. Output "xSystemPressReady" is TRUE
  4. Output "xInOperation" is FALSE
  5. Output "xError" is FALSE
4. Set input "xEnableSystem" to TRUE
5. During operation output "xInOperation" is TRUE
6. 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 of function block "FB\_Connect":
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x01 (Cyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x01 (Cyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Precondition of function block "FB\_Manual":
  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
3. Precondition of function block "FB\_Status":
  1. Input "xEnable" is TRUE
  2. Output "xActive" is TRUE
  3. Output "xSystemPressReady" is TRUE
  4. Output "xInOperation" is FALSE
  5. Output "xError" is FALSE
4. Set input "xEnableSystem" to FALSE
5. During operation output "xInOperation" is TRUE
6. 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



### Important

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

1. Precondition of function block "FB\_Connect":
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x01 (Cyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x01 (Cyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Precondition of function block "FB\_Manual":
  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
3. Set input "xTare" to TRUE
4. 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



### Important

**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 of function block "FB\_Connect":
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x01 (Cyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x01 (Cyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Precondition of function block "FB\_Manual":
  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
3. Set input "rTargetOffsetForceSensor" to your desired offset (N)
4. 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 of function block "FB\_Connect":
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x01 (Cyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x01 (Cyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Precondition of function block "FB\_Manual":
  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
3. Precondition of function block "FB\_SystemSettings":
  1. Input "xEnable" is TRUE
  2. Output "xActive" is TRUE
  3. Output "xStatusHW\_Config" is TRUE
  4. Output "xInOperation" is FALSE
  5. Output "xSystemError" is FALSE
  6. Output "xError" is FALSE
4. Set input "xAbort" to TRUE
5. Set input "xStartHoming" to TRUE
6. During operation output "xInOperation" is TRUE
7. Homing can be aborted via set input "xAbort" to FALSE
8. 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
9. 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 of function block "FB\_Connect":
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x01 (Cyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x01 (Cyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Precondition of function block "FB\_Manual":
  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
3. Precondition of function block "FB\_Status":
  1. Input "xEnable" is TRUE
  2. Output "xActive" is TRUE
  3. Output "xSystemPressReady" is TRUE
  4. Output "xInOperation" is FALSE
  5. Output "xError" is FALSE
4. Set input "siTargetPosMode" to 0x00 (Jog)
5. Set input "rTargetMotionVelocity" to your desired velocity (mm/s)
6. Set input "xJogPos" or "xJogPos" to TRUE
7. During operation output "xInOperation" is TRUE
8. Jogging can be stopped via set input "xJogPos" or "xJogPos" to FALSE
9. 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 of function block "FB\_Connect":
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x01 (Cyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x01 (Cyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Precondition of function block "FB\_Manual":
  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
3. Precondition of function block "FB\_Status":
  1. Input "xEnable" is TRUE
  2. Output "xActive" is TRUE
  3. Output "xSystemPressReady" is TRUE
  4. Output "xInOperation" is FALSE
  5. Output "xError" is FALSE
4. Set input "siTargetPosMode" to 0x01 (Move absolute)
5. Set input "rTargetMotionVelocity" to your desired velocity (mm/s)
6. Set input "rTargetMotionPositionDistance" to your desired position (mm)
7. Set input "xStopMove" to FALSE
8. Set input "xMove" to TRUE
9. During operation output "xInOperation" is TRUE
10. Move absolute can be stopped via set input "xStopMove" to TRUE
11. 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 "rActualPosition" is unequal to input "rTargetMotionPositionDistance"
12. 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 "rActualPosition" is equal to input "rTargetMotionPositionDistance"

### Move relative

1. Precondition of function block “FB\_Connect”:
  1. Input "xEnable" is TRUE
  2. Input “siTargetComMode” is 0x01 (Cyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output “siActualComMode” is 0x01 (Cyclic communication mode)
  5. Output “xConnected” is TRUE
  6. Output "xError" is FALSE
2. Precondition of function block “FB\_Manual”:
  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
3. Precondition of function block “FB\_Status”:
  1. Input "xEnable" is TRUE
  2. Output "xActive" is TRUE
  3. Output “xSystemPressReady” is TRUE
  4. Output “xInOperation” is FALSE
  5. Output "xError" is FALSE
4. Set input “siTargetPosMode” to 0x02 (Move relative)
5. Set input “rTargetMotionVelocity” to your desired velocity (mm/s)
6. Set input “rTargetMotionPositionDistance” to your desired distance (mm)
7. Set input “xStopMove” to FALSE
8. Set input “xMove” to TRUE
9. During operation output “xInOperation” is TRUE
10. Move relative can be stopped via set input “xStopMove” to TRUE
11. 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 "rActualPosition" is unequal to output "rActualPosition" plus input “rTargetMotionPositionDistance”
12. 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 "rActualPosition" is equal to output "rActualPosition" plus input “rTargetMotionPositionDistance”

### Quit system error

1. Precondition of function block “FB\_Connect”:
  1. Input "xEnable" is TRUE
  2. Input “siTargetComMode” is 0x01 (Cyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output “siActualComMode” is 0x01 (Cyclic communication mode)
  5. Output “xConnected” is TRUE
  6. Output "xError" is FALSE
2. Precondition of function block “FB\_Manual”:
  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
3. 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 "iErrorID" contains detailed error information  
For further information see error table
2. Set input " xEnable " to FALSE
3. Fix the error

### 3.5.7 FB\_PressControl

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

Pre-Action:

- General:
  - Map an instance of the structure "stSystemDataRef" to the function block input "stSystemDataRef"
  - Status of connection must be connection established in the function block "FB\_Connect" -> FB\_Connect.xConnected
  - Status of actual communication mode must be cyclic communication mode in the function block "FB\_Connect" -> FB\_Connect.siActualComMode
  - For the input "xQuitSystemError" all function blocks (FB\_Manual, FB\_PressControl, FB\_SystemSettings and FB\_Status) have to use the same variable when they are active or all function blocks (FB\_Manual, FB\_PressControl, FB\_SystemSettings and FB\_Status) must be interlocked. Because they are all mapped to the same internal variable. If there is no mechanism implemented described before it could be that the input is overwritten by a call of another active function block (FB\_Manual, FB\_PressControl, FB\_SystemSettings and FB\_Status).
  - For the inputs "xEnableSystem", "xStartHoming", "xAbort" the function blocks (FB\_Manual and FB\_PressControl) have to use the same variable when they are active or the function blocks (FB\_Manual and FB\_PressControl) must be interlocked. Because they are mapped to the same internal variable. If there is no mechanism implemented described before it could be that the input is overwritten by a call of the other active function block (FB\_Manual and FB\_PressControl).

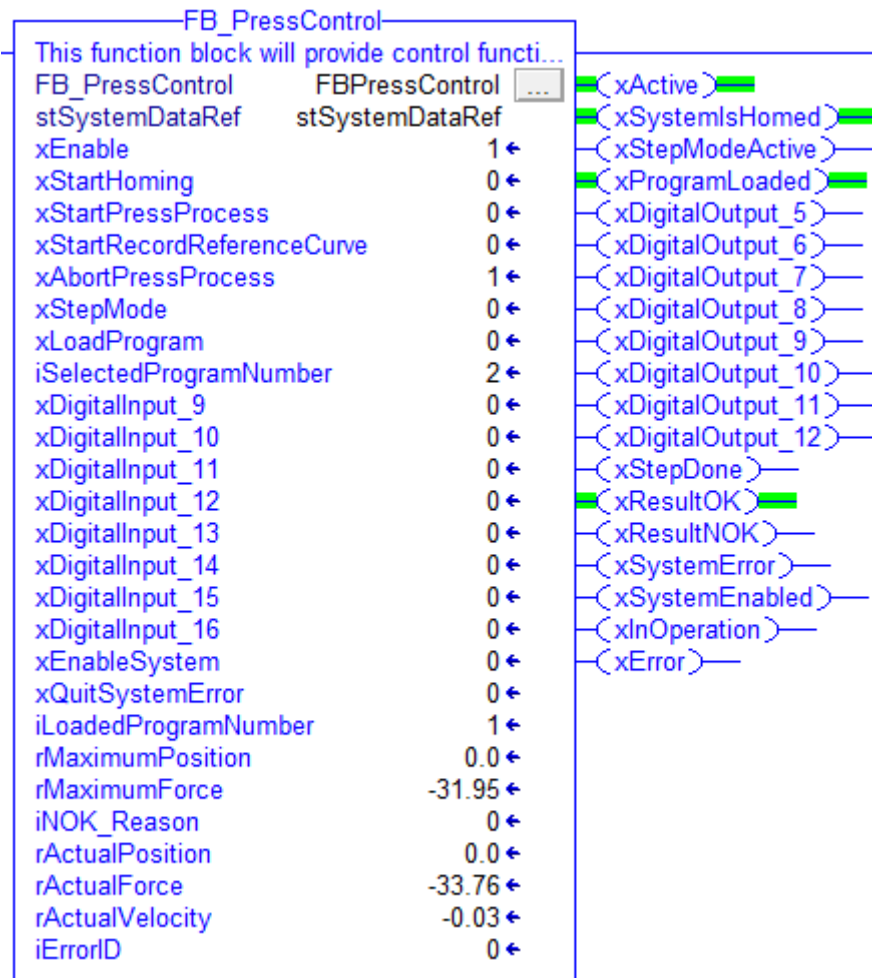


Figure 3-8: FB\_PressControl



In/Out		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
stSystemDataRef	ST_SystemDataRef	This data structure contains all data required to communicate with the system
<b>Inputs</b>		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
xEnable	BOOL	Activate press control functions for the Festo system FALSE: Action stopped, resets outputs TRUE: Action running
xEnableSystem	BOOL	Enable system command Falling edge: Disable system Rising edge: Enable system
xStartHoming	BOOL	Homing command Falling edge: No action Rising edge: Start homing movement
iSelectedProgramNumber	INT	Selected program number
xLoadProgram	BOOL	Load program command Falling edge: No action Rising edge: Load selected program number
xStartRecordReferenceCurve	BOOL	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	BOOL	Start press process command Falling edge: No action Rising edge: Start press process
xAbort	BOOL	Abort command False: Abort press process or homing True: No action
xStepMode	BOOL	Step mode command False: Deactivate step mode True: Activate step mode
xQuitSystemError	BOOL	Quit system error command Falling edge: No action Rising edge: Quit upcoming system error
xDigitalInput9	BOOL	Virtual digital input 9 for step enabling conditions in the sequencer of the press process
xDigitalInput10	BOOL	Virtual digital input 10 for step enabling conditions in the sequencer of the press process
xDigitalInput11	BOOL	Virtual digital input 11 for step enabling conditions in the sequencer of the press process
xDigitalInput12	BOOL	Virtual digital input 12 for step enabling conditions in the sequencer of the press process
xDigitalInput13	BOOL	Virtual digital input 13 for step enabling conditions in the sequencer of the press process
xDigitalInput14	BOOL	Virtual digital input 14 for step enabling conditions in the sequencer of the press process
xDigitalInput15	BOOL	Virtual digital input 15 for step enabling conditions in the sequencer of the press process
xDigitalInput16	BOOL	Virtual digital input 16 for step enabling conditions in the sequencer of the press process

<b>In/Out</b>		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
<b>Outputs</b>		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
xActive	BOOL	Function block is active FALSE: Function block is inactive TRUE: Function block is active
xSystemEnabled	BOOL	System is enabled FALSE: System is disabled TRUE: System is enabled
xSystemIsHomed	BOOL	System is homed FALSE: System is not homed TRUE: System is homed
xProgramLoaded	BOOL	Program is loaded Falling edge: No action Rising edge: Program is loaded
iLoadedProgramNumber	INT	Loaded program number
xStepModeActive	BOOL	Step mode is active FALSE: Step mode is inactive TRUE: Step mode is active
xStepDone	BOOL	Step is done Falling edge: No action Rising edge: Step is done
xInOperation	BOOL	System is in operation FALSE: System is not in operation TRUE: System is in operation
xSystemError	BOOL	System error FALSE: No error TRUE: Error occurred
xResultOK	BOOL	OK result of the press process
xResultNOK	BOOL	NOK result of the press process For further information see NOK reason
rMaximumPosition	REAL	Maximum position of the press process Unit [mm]
rMaximumForce	REAL	Maximum force of the press process Unit [N]
iNOK_Reason	INT	NOK reason of the press process
xDigitalOutput5	BOOL	Virtual digital output 5 in the sequencer of the press process
xDigitalOutput6	BOOL	Virtual digital output 6 in the sequencer of the press process
xDigitalOutput7	BOOL	Virtual digital output 7 in the sequencer of the press process
xDigitalOutput8	BOOL	Virtual digital output 8 in the sequencer of the press process
xDigitalOutput9	BOOL	Virtual digital output 9 in the sequencer of the press process
xDigitalOutput10	BOOL	Virtual digital output 10 in the sequencer of the press process

In/Out		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
xDigitalOutput11	BOOL	Virtual digital output 11 in the sequencer of the press process
xDigitalOutput12	BOOL	Virtual digital output 12 in the sequencer of the press process
rActualPosition	REAL	Actual position Unit [mm]
rActualForce	REAL	Actual force Unit [N]
rActualVelocity	REAL	Actual velocity Unit [mm/s]
xError	BOOL	Error occurred during action FALSE: No error TRUE: Error occurred, function block aborts action
iErrorID	INT	Contains detailed error information in case of an error For further information see error table

Table 3-28: FB\_PressControl

<b>Error table</b>	
<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	No connection established
0x0004	Invalid communication mode
0x0005	Service is not supported
0x0006	Data length invalid
0x0007	Object index invalid
0x0008	Access type invalid
0x0009	Data type invalid
0x000A	Value invalid
0x000B	Read type invalid
0x000C	Write type invalid
0x000D	Values invalid

Table 3-29: Error table FB\_PressControl

**Examples:****Read press control status**

1. Precondition of function block "FB\_Connect":
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x01 (Cyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x01 (Cyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Set input "xEnable" to TRUE

3. All outputs will be updated according to the status of the Festo system

**Enable system**

1. Precondition of function block "FB\_Connect":
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x01 (Cyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x01 (Cyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Precondition of function block "FB\_PressControl":
  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
3. Precondition of function block "FB\_Status":
  1. Input "xEnable" is TRUE
  2. Output "xActive" is TRUE
  3. Output "xSystemPressReady" is TRUE
  4. Output "xInOperation" is FALSE
  5. Output "xError" is FALSE
4. Set input "xEnableSystem" to TRUE
5. During operation output "xInOperation" is TRUE
6. 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 of function block "FB\_Connect":
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x01 (Cyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x01 (Cyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Precondition of function block "FB\_PressControl":
  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
3. Precondition of function block "FB\_Status":
  1. Input "xEnable" is TRUE
  2. Output "xActive" is TRUE
  3. Output "xSystemPressReady" is TRUE
  4. Output "xInOperation" is FALSE
  5. Output "xError" is FALSE
4. Set input "xEnableSystem" to FALSE
5. During operation output "xInOperation" is TRUE
6. 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

## Homing

1. Precondition of function block "FB\_Connect":
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x01 (Cyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x01 (Cyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Precondition of function block "FB\_PressControl":
  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
3. Precondition of function block "FB\_SystemSettings":
  1. Input "xEnable" is TRUE
  2. Output "xActive" is TRUE
  3. Output "xStatusHW\_Config" is TRUE
  4. Output "xInOperation" is FALSE
  5. Output "xSystemError" is FALSE
  6. Output "xError" is FALSE
4. Set input "xAbort" to TRUE
5. Set input "xStartHoming" to TRUE
6. During operation output "xInOperation" is TRUE
7. Homing can be aborted via set input "xAbort" to FALSE
8. 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
9. 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

## Load program

1. Precondition of function block "FB\_Connect":
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x01 (Cyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x01 (Cyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Precondition of function block "FB\_PressControl":
  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
3. Set input "iSelectedProgramNumber" to your desired program number
4. Set input "xLoadProgram" to TRUE
5. During operation output "xInOperation" is TRUE
6. 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 "iLoadedProgramNumber" is input "iSelectedProgramNumber"

## Start press process

1. Precondition of function block "FB\_Connect":
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x01 (Cyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x01 (Cyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Precondition of function block "FB\_PressControl":
  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
3. Precondition of function block "FB\_Status":
  1. Input "xEnable" is TRUE
  2. Output "xActive" is TRUE
  3. Output "xSystemPressReady" is TRUE
  4. Output "xInOperation" is FALSE
  5. Output "xError" is FALSE
4. Check if output "iLoadedProgramNumber" is set to your desired program number
5. Set input "xAbort" to TRUE
6. Set input "xStartPressProcess" to TRUE
7. During operation output "xInOperation" is TRUE
8. Press process can be aborted via set input "xAbort" to FALSE
9. 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
10. 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 of function block "FB\_Connect":
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x01 (Cyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x01 (Cyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Precondition of function block "FB\_PressControl":
  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
3. Precondition of function block "FB\_Status":
  1. Input "xEnable" is TRUE
  2. Output "xActive" is TRUE
  3. Output "xSystemPressReady" is TRUE
  4. Output "xInOperation" is FALSE
  5. Output "xError" is FALSE
4. You have to be in the window "Step 2/4: Record / loading reference curve(s)" in the Festo system
5. Set input "xAbort" to TRUE
6. Set input "xRecordReferenceCurve" to TRUE
7. During operation output "xInOperation" is TRUE
8. Record reference curve can be aborted via set input "xAbort" to FALSE
9. 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 of function block "FB\_Connect":
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x01 (Cyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x01 (Cyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Precondition of function block "FB\_PressControl":
  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
3. Precondition of function block "FB\_Status":
  1. Input "xEnable" is TRUE
  2. Output "xActive" is TRUE
  3. Output "xSystemPressReady" is TRUE
  4. Output "xInOperation" is FALSE
  5. Output "xError" is FALSE
4. Check if output "iLoadedProgramNumber" is set to your desired program number
5. Set input "xAbort" to TRUE
6. Set input "xStepMode" to TRUE
7. Check if output "xStepModeActive" is TRUE
8. Set input "xStartPressProcess" to TRUE
9. During operation output "xInOperation" is TRUE
10. During step processing output "xStepDone" is FALSE
11. Press process can be aborted via set input "xAbort" to FALSE
12. 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
13. Set input "xStepMode" to FALSE
14. Set input "xStepMode" to TRUE
15. During step processing output "xStepDone" is FALSE
16. 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
17. Repeat step 12 to 15 until all steps are process and condition of press process done is reached
18. 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
19. 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 of function block "FB\_Connect":
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x01 (Cyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x01 (Cyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Precondition of function block "FB\_PressControl":
  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
3. 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 "iErrorID" contains detailed error information  
For further information see error table
2. Set input " xEnable " to FALSE
3. Fix the error

### 3.5.8 FB\_Status

This function block provides status information of the Festo system.

Pre-Action:

- General:
  - Map an instance of the structure "stSystemDataRef" to the function block input "stSystemDataRef"
  - Status of connection must be connection established in the function block "FB\_Connect" -> FB\_Connect.xConnected
  - Status of actual communication mode must be cyclic communication mode in the function block "FB\_Connect" -> FB\_Connect.siActualComMode
  - FB\_Manual, FB\_PressControl, FB\_SystemSettings and FB\_Status must be interlocked
  - For the input "xQuitSystemError" all function blocks (FB\_Manual, FB\_PressControl, FB\_SystemSettings and FB\_Status) have to use the same variable when they are active or all function blocks (FB\_Manual, FB\_PressControl, FB\_SystemSettings and FB\_Status) must be interlocked. Because they are all mapped to the same internal variable. If there is no mechanism implemented described before it could be that the input is overwritten by a call of another active function block (FB\_Manual, FB\_PressControl, FB\_SystemSettings and FB\_Status).

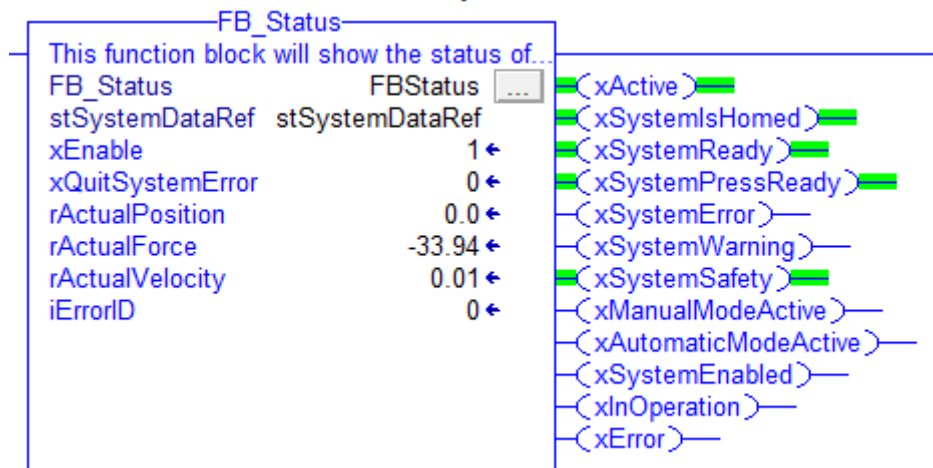


Figure 3-9: FB\_Status

<b>In/Out</b>		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
stSystemDataRef	ST_SystemDataRef	This data structure contains all data required to communicate with the system
<b>Inputs</b>		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
xEnable	BOOL	Activate get status information of the Festo system FALSE: Action stopped, resets outputs TRUE: Action running
xQuitSystemError	BOOL	Quit system error command Falling edge: No action Rising edge: Quit upcoming system error
<b>Outputs</b>		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
xActive	BOOL	Function block is active FALSE: Function block is inactive TRUE: Function block is active
xManualModeActive	BOOL	Manual mode active FALSE: Manual mode is inactive TRUE: Manual mode is active
xAutomaticModeActive	BOOL	Automatic mode active FALSE: Automatic mode is inactive TRUE: Automatic mode is active
xSystemEnabled	BOOL	System is enabled FALSE: System is disabled TRUE: System is enabled
xSystemIsHomed	BOOL	System is homed FALSE: System is not homed TRUE: System is homed
xSystemReady	BOOL	System is ready FALSE: System is not ready TRUE: System is ready
xSystemPressReady	BOOL	System servo press is ready FALSE: System servo press is not ready TRUE: System servo press is ready
xSystemError	BOOL	System error FALSE: No error TRUE: Error occurred
xSystemWarning	BOOL	System warning FALSE: No warning TRUE: Warning occurred
xSystemSafetyFeedback	BOOL	System safety acknowledgement FALSE: No safety acknowledgement TRUE: Safety acknowledgement
xInOperation	BOOL	System is in operation FALSE: System is not in operation TRUE: System is in operation
rActualPosition	REAL	Actual position Unit [mm]
rActualForce	REAL	Actual force Unit [N]

<b>In/Out</b>		
<i>Name</i>	<i>Data type</i>	<i>Description</i>
rActualVelocity	REAL	Actual velocity Unit [mm/s]
xError	BOOL	Error occurred during action FALSE: No error TRUE: Error occurred, function block aborts action
iErrorID	INT	Contains detailed error information in case of an error For further information see error table

Table 3-30: FB\_Status

<b>Error table</b>	
<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	No connection established
0x0004	Invalid communication mode
0x0005	Service is not supported
0x0006	Data length invalid
0x0007	Object index invalid
0x0008	Access type invalid
0x0009	Data type invalid
0x000A	Value invalid
0x000B	Read type invalid
0x000C	Write type invalid
0x000D	Values invalid

Table 3-31: Error table FB\_Status

## Examples:

### Read system status

1. Precondition of function block "FB\_Connect":
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x01 (Cyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x01 (Cyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Set input "xEnable " to TRUE
3. All outputs will be updated according to the status of the Festo system

### Quit system error

1. Precondition of function block "FB\_Connect":
  1. Input "xEnable" is TRUE
  2. Input "siTargetComMode" is 0x01 (Cyclic communication mode)
  3. Output "xActive" is TRUE
  4. Output "siActualComMode" is 0x01 (Cyclic communication mode)
  5. Output "xConnected" is TRUE
  6. Output "xError" is FALSE
2. Precondition of function block "FB\_Status":
  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
3. 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 "iErrorID" contains detailed error information  
For further information see error table
2. Set input " xEnable " to FALSE
3. Fix the error