

Festo_Positioning_Basic library – FB_PICK_AND_PLACE

Commissioning of a selected system with the Festo_Positioning_Basic library is explained step by step, so that it's successfully integrated into the user program.

FPosB

Title Festo_Positioning_Basic library – FB_PICK_AND_PLACE
Version 1.30
Document no. 100102
Originalen
AuthorFesto

Last saved 11.06.2021

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 **FB_PICK_AND_PLACE_CYCLIC..... 5**
- 1.1 Activate the function block 5
- 1.2 FB_PICK_AND_PLACE_CYCLIC main features 7
- 1.3 Teaching Pick, Place and Park position 9
- 1.4 Controll Pick- and PlacePos 10

1 FB_PICK_AND_PLACE_CYCLIC

1.1 Activate the function block

The description of all inputs and outputs can be found in the separate documentation. This is an example how to use the basic functions of this function block.

Basically, this function block is controlled with the internal CNC mode of the FPosB library.

This function block has an output called pCNC, which must be assigned to a program place of the internal CNC mode.

In this example, it's in the 4th place of the CNC list.

```
fbPick_n_Place_EVO3(
  xGotoParkPos      := ,
  pstDynamic        := ADR(gstDynPnP),
  pstDynamicAddAxes := ADR(gstDynPnP_AddAxes),
  xRepeat           := ,
  xAlreadyPicked    := ,
  pstPickPos        := ADR(gstPickPos),
  pstPlacePos       := ADR(gstPlacePos),
  pstParkPos        := ADR(gstParkPos),
  plrPrePickZ_Offset := ,
  plrPrePlaceZ_Offset := ,
  plrPrePickZ_OffsetVel := ,
  plrPrePlaceZ_OffsetVel := ,
  xContinuousPrePick := ,
  xContinuousPrePlace := ,
  xTeachPickPos      := ,
  xTeachPlacePos     := ,
  xTeachParkPos      := ,
  xAckPickOrPlace    := ,
  lrUpPosZ           := ,
  lrRadius           := ,
  udiPickTimer       := ,
  udiPlaceTimer      := ,
  xPickPosReady      := ,
  xPlacePosReady     := ,
  pCNC               => g_stIn.parCNCUser[4],
  xPrePick           => ,
  xPick              => ,
  xPrePlace          => ,
  xPlace             => ,
  iPickCounter       => ,
  iPlaceCounter      => ,
  udiCycle           => ,
  udiCycleTime       => ,
  tCycleTimePickPlace => ,
  stKineDataRef      := gstGantryDataRef);
```

Calling the 4th place of the CNCUser array, it will load the data from the function block.

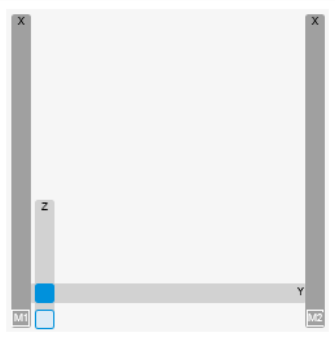
Festo_Positioning_Basic_3

Operation Mode
No Mode
Configuration Mode
Homing Mode
Jogging Mode
Stepping Mode
PTP Mode
Internal CNC Mode
External CNC Mode

Enable System
Reset System
Ack Error

Select Coordinate System
MCS
UCS
WPCS

Override 100



UCS			
	Origin	Pos On X	Pos On Y
X	0.000	1.000	0.000
Y	0.000	0.000	1.000
Z	0.000	0.000	0.000

WPCS			
	Origin	Pos On X	Pos On Y
X	0.000	1.000	-1.000
Y	0.000	1.000	1.000
Z	0.000	0.000	0.000

M-Function K-Param
H-Function L-Param
G-Code Line M Active ☐

Internal CNC No **Pick_and_Place_FPosB**

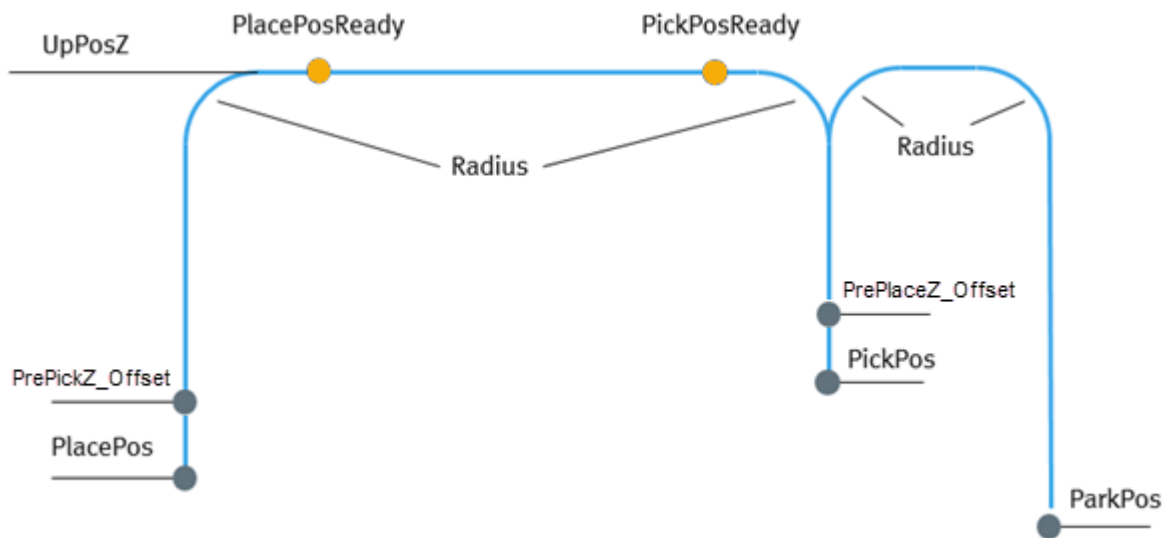
Start
Halt
Quick Stop
Abort
Ack M Function
Ack All M Functions
Enable Step Mode
Next Step
EnablePathCorrection
AvoidAbortByError

CPX_CEC.Application.GVL_Motion_INOUT		
Expression	Type	Value
vOrigin1	SMC_VECTOR3D	
vPointOnX1	SMC_VECTOR3D	
vPointOnY1	SMC_VECTOR3D	
vOrigin2	SMC_VECTOR3D	
vPointOnX2	SMC_VECTOR3D	
vPointOnY2	SMC_VECTOR3D	
garSingleVar	ARRAY [0..3] OF SMC_CNC.SMC_SingleVar	
g_stIn	FPosB.FPB_INPUT_DATA	
eMOP	OP_MODE	eInternalCncMode
stCW	CONTROL_WORD	
eSetCS	COORDINATE_SYSTEM	eMCS
stUCS	CS_ORIGIN	
stWPCS	CS_ORIGIN	
stDynPtp	DYNAMIC	
stPosPtp	CART_POS	
stDynJogStep	DYNAMIC	
lrStepWidth	LREAL	5
eSelectAxis	AXIS	eX
sCNCFilePath	STRING	'ffx/codesys/ TestCNCs/*.*cnc'
uiUserCNCNo	UINT	4
stCNC CorrParameters	CNC_CORRECTION_PARAMETERS	

With the functions from the control word, like “Start”, “Halt” or “Abort”, the movement of the function block is controlled.

1.2 FB_PICK_AND_PLACE_CYCLIC main features

The following image describes the path of the FB_PICK_AND_PLACE in its main features:



PlacePos (pstPlacePos): Position to place the part

PickPos (pstPickPos): Position to pick the part

UpPosZ (lrUpPosZ): Up-position of the z-axis, while moving between pick and place position

Radius (lrRadius): Radius, which is used to smoothen the path between Z- and X/Y- movement (for all positioning). The radius is limited by the shortest distance of a part of the whole movement.

Example: If the shortest distance, for instance in z, is 30mm, the radius has a maximum radius of the half of the distance, so 15mm in this example.

ParkPos (pstParkPos): Position to park the gantry at a certain waiting position. If activated, the gantry will move with a U-move after the PlacePos to this ParkPos. Also positioning from ParkPos to PickPos is done with a U-move.

To activate this movement, a separate input is needed: xGotoParkPos.

PlacePosReady (xPickPosReady): fixed waiting position before placing the part. Need to be acknowledged by this input to continue the movement (state controlled).

PickPosReady (xPlacePosReady): fixed waiting position before picking the part. Need to be acknowledged by this input to continue the movement (state controlled)

PrePickZ_Offset, PrePlaceZ_Offset: This pointer to lreal defined the offset before Pick or Place Pos
Offset value from Pick/Place Position or absolute Position?
Any limitations?

plrPrePickZ_OffsetVel, plrPrePlaceZ_OffsetVel: Velocity for the movement from the offset-position until the pick or place position.

Limit in velocity? Acc/Dec used from Dynamic setting of Pick_Place_FB?

This pointer to lreal defined the offset velocity for **plrPrePickZ_Offset, plrPrePlaceZ_Offset**

xAlreadyPicked: by activate this Bit, it will directly goto PlacePos and skip the Pick Pos.

xRepeat: This will automatically repeat the Pick and Place Cycle, depends on the conditions of the following Inputs:

xContinuousPrePick,
xContinuousPrePlace,
xPickPosReady,
xPlacePosReady,
udiPickTimer,
udiPlaceTimer,

```
fbPick_n_Place_EVO3(
  xGotoParkPos      := ,
  pstDynamic        := ADR(gstDynPnP),
  pstDynamicAddAxes := ADR(gstDynPnP_AddAxes),
  xRepeat           := ,
  xAlreadyPicked    := ,
  pstPickPos        := ADR(gstPickPos),
  pstPlacePos       := ADR(gstPlacePos),
  pstParkPos        := ADR(gstParkPos),
  plrPrePickZ_Offset := ,
  plrPrePlaceZ_Offset := ,
  plrPrePickZ_OffsetVel := ,
  plrPrePlaceZ_OffsetVel := ,
  xContinuousPrePick := ,
  xContinuousPrePlace := ,
  xTeachPickPos      := ,
  xTeachPlacePos     := ,
  xTeachParkPos      := ,
  xAckPickOrPlace    := ,
  lrUpPosZ           := ,
  lrRadius           := ,
  udiPickTimer       := ,
  udiPlaceTimer      := ,
  xPickPosReady      := ,
  xPlacePosReady     := ,
  pCNC               => g_stIn.parCNCUser[4],
  xPrePick           => ,
  xPick              => ,
  xPrePlace          => ,
  xPlace             => ,
  iPickCounter       => ,
  iPlaceCounter      => ,
  udiCycle           => ,
  udiCycleTime       => ,
  tCycleTimePickPlace => ,
  stKineDataRef      := gstGantryDataRef);
```

FB_PICK_AND_PLACE_CYCLIC :		fbPick_n_Place_EVO3		
xRepeat		xPrePick		
xGotoParkPos		xPick		
UpperPosZ : 10.000		xPrePlace		
lrRad : 20.000		xPlace		
PickTimer: 1000		iPickCounter:0		
PlaceTimer: 1000		iPlaceCounter:0		
xAlreadyPicked		udiCycle :0		
xAckPickOrPlace		CycleTime 0 [ms]		
xPickPosReady (Tab)		lrPickRatePerMin: 0.000		
xPlacePosReady (Tab)		Park	Pick	Place
xTeachPickPos		X	10.000	30.000
		Y	10.000	50.000
xTeachParkPos		Z	10.000	100.000
		U	10.000	180.000
xTeachPlacePos		PrePosZoffs	0.000	0.000
xPickPosReady(Toggle)		PrePosZ VEL	0.000	0.000
xPlacePosReady(Toggle)		Dynamic	XYZ	U/A
		Vel	100.00	100.00
xContinousPrePick		Acc	1000.00	1000.00
		Dec	-1000.00	-1000.00
xContinousPrePlace				

1.3 Teaching Pick, Place and Park position

It's possible to move the gantry to a certain position (in Jogging, Step or PTP mode) and to apply these position values as one of the position of the Pick and Place function block.

FB_PICK_AND_PLACE_CYCLIC :		fbPick_n_Place_EVO3		
xRepeat	xPrePick			
xGotoParkPos	xPick			
UpperPosZ : 10.000	xPrePlace			
lrRad : 20.000	xPlace			
PickTimer: 1000	iPickCounter:0			
PlaceTimer: 1000	iPlaceCounter:0			
xAlreadyPicked	udiCycle :0			
xAckPickOrPlace	CycleTime 0 [ms]			
xPickPosReady (Tab)	lrPickRatePerMin: 0.000			
xPlacePosReady (Tab)				
xTeachPickPos		Park	Pick	Place
	X	10.000	30.000	110.000
	Y	10.000	30.000	50.000
	Z	10.000	100.000	100.000
xTeachParkPos	U	10.000	90.000	180.000
xTeachPlacePos	PrePosZoffs		0.000 0.000	
	PrePosZ VEL		0.000 0.000	
xPickPosReady(Toggle)	Dynamic		XYZ	U/A
xPlacePosReady(Toggle)	Vel	100.00		100.00
	Acc	1000.00		1000.00
xContinousPrePick	Dec	-1000.00		-1000.00
xContinousPrePlace				

```

fbPick_n_Place_EVO3(
  xGotoParkPos      := ,
  pstDynamic        := ADR(gstDynPnP),
  pstDynamicAddAxes := ADR(gstDynPnP_AddAxes),
  xRepeat           := ,
  xAlreadyPicked    := ,
  pstPickPos        := ADR(gstPickPos),
  pstPlacePos       := ADR(gstPlacePos),
  pstParkPos        := ADR(gstParkPos),
  plrPrePickZ_Offset := ,
  plrPrePlaceZ_Offset := ,
  plrPrePickZ_OffsetVel := ,
  plrPrePlaceZ_OffsetVel := ,
  xContinuousPrePick := ,
  xContinuousPrePlace := ,
  xTeachPickPos      := ,
  xTeachPlacePos     := ,
  xTeachParkPos      := ,
  xAckPickOrPlace    := ,
  lrUpPosZ           := ,
  lrRadius           := ,
  udiPickTimer       := ,
  udiPlaceTimer      := ,
  xPickPosReady      := ,
  xPlacePosReady     := ,
  pCNC               => g_stIn.parCNCUser[4],
  xPrePick           => ,
  xPick              => ,
  xPrePlace          => ,
  xPlace             => ,
  iPickCounter       => ,
  iPlaceCounter      => ,
  udiCycle            => ,
  udiCycleTime       => ,
  tCycleTimePickPlace => ,
  stKineDataRef      := gstGantryDataRef);
  
```

1.4 Controll Pick- and PlacePos

You have 2 possibilities to control the hold time in both positions:

1. Using a timer => PickTimer and PlaceTimer

FB_PICK_AND_PLACE_CYCLIC :		fbPick_n_Place_EVO3		
xRepeat		xPrePick		
xGotoParkPos		xPick		
UpperPosZ : 10.000		xPrePlace		
lrRad : 20.000		xPlace		
PickTimer: 1000		iPickCounter:0		
PlaceTimer: 1000		iPlaceCounter:0		
xAlreadyPicked		udiCycle :0		
xAckPickOrPlace		CycleTime 0 [ms]		
xPickPosReady (Tab)		lrPickRatePerMin: 0.000		
xPlacePosReady (Tab)		Park	Pick	Place
xTeachPickPos	X	10.000	30.000	110.000
xTeachParkPos	Y	10.000	30.000	50.000
xTeachPlacePos	Z	10.000	100.000	100.000
	U	10.000	90.000	180.000
	PrePosZoffs		0.000	0.000
xPickPosReady(Toggle)	PrePosZ VEL		0.000	0.000
xPlacePosReady(Toggle)	Dynamic	XYZ	U/A	
	Vel	100.00	100.00	
xContinousPrePick	Acc	1000.00	1000.00	
xContinousPrePlace	Dec	-1000.00	-1000.00	

```

fbPick_n_Place_EVO3(
  xGotoParkPos      := ,
  pstDynamic         := ADR(gstDynPnP),
  pstDynamicAddAxes := ADR(gstDynPnP_AddAxes),
  xRepeat           := ,
  xAlreadyPicked    := ,
  pstPickPos        := ADR(gstPickPos),
  pstPlacePos       := ADR(gstPlacePos),
  pstParkPos        := ADR(gstParkPos),
  plrPrePickZ_Offset := ,
  plrPrePlaceZ_Offset := ,
  plrPrePickZ_OffsetVel := ,
  plrPrePlaceZ_OffsetVel := ,
  xContinuousPrePick := ,
  xContinuousPrePlace := ,
  xTeachPickPos      := ,
  xTeachPlacePos     := ,
  xTeachParkPos      := ,
  xAckPickOrPlace    := ,
  lrUpPosZ           := ,
  lrRadius           := ,
  udiPickTimer       := ,
  udiPlaceTimer      := ,
  xPickPosReady      := ,
  xPlacePosReady     := ,
  pCNC               => g_stIn.parCNCUser[4],
  xPrePick           => ,
  xPick              => ,
  xPrePlace          => ,
  xPlace             => ,
  iPickCounter       => ,
  iPlaceCounter      => ,
  udiCycle           => ,
  udiCycleTime       => ,
  tCycleTimePickPlace => ,
  stKineDataRef      := gstGantryDataRef);
  
```

2. Using an input to acknowledge it depending on other processes.

- a. With the input xAlreadyPicked, which, however, has a different function (see 1.2)

FB_PICK_AND_PLACE_CYCLIC : fbPick_n_Place_EVO3

Repeat	xPrePick
GotoParkPos	xPick
UpperPosZ : 10.000	xPrePlace
Rad : 20.000	xPlace
PickTimer: 1000	iPickCounter:0
PlaceTimer: 1000	iPlaceCounter:0
AlreadyPicked	udiCycle :0
AckPickOrPlace	CycleTime 0 [ms]
PickPosReady (Tab)	lrPickRatePerMin: 0.000
PlacePosReady (Tab)	
TeachPickPos	
TeachParkPos	
TeachPlacePos	
PickPosReady(Toggle)	
PlacePosReady(Toggle)	
ContinuousPrePick	
ContinuousPrePlace	

```

DEFAULT_FB_PICK_AND_PLACE_CYCLIC
xGotoParkPos := ,
pstDynamic := ADR(gstDynPnP),
pstDynamicAddAxes := ADR(gstDynPnP_AddAxes),
xRepeat := ,
xAlreadyPicked := ,
pstPickPos := ADR(gstPickPos),
pstPlacePos := ADR(gstPlacePos),
pstParkPos := ADR(gstParkPos),
plrPrePickZ_Offset := ,
plrPrePlaceZ_Offset := ,
plrPrePickZ_OffsetVel := ,
plrPrePlaceZ_OffsetVel := ,
xContinuousPrePick := ,
xContinuousPrePlace := ,
xTeachPickPos := ,
xTeachPlacePos := ,
xTeachParkPos := ,
xAckPickOrPlace := ,
lrUpPosZ := ,
lrRadius := ,
udiPickTimer := ,
udiPlaceTimer := ,
xPickPosReady := ,
xPlacePosReady := ,
pCNC => g_stIn.parCNCUser[4],
xPrePick => ,
xPick => ,
xPrePlace => ,
xPlace => ,
iPickCounter => ,
iPlaceCounter => ,
udiCycle => ,
udiCycleTime => ,
tCycleTimePickPlace => ,

```

- b. With xAckPickOrPlace. PLC sets this input as soon as a part is picked or placed safely. The cycle will continue.

FB_PICK_AND_PLACE_CYCLIC : fbPick_n_Place_EVO3

Repeat	xPrePick
GotoParkPos	xPick
UpperPosZ : 10.000	xPrePlace
Rad : 20.000	xPlace
PickTimer: 1000	iPickCounter:0
PlaceTimer: 1000	iPlaceCounter:0
AlreadyPicked	udiCycle :0
AckPickOrPlace	CycleTime 0 [ms]
PickPosReady (Tab)	lrPickRatePerMin: 0.000
PlacePosReady (Tab)	
TeachPickPos	
TeachParkPos	
TeachPlacePos	
PickPosReady(Toggle)	
PlacePosReady(Toggle)	
ContinuousPrePick	
ContinuousPrePlace	

```

DEFAULT_FB_PICK_AND_PLACE_CYCLIC
xGotoParkPos := ,
pstDynamic := ADR(gstDynPnP),
pstDynamicAddAxes := ADR(gstDynPnP_AddAxes),
xRepeat := ,
xAlreadyPicked := ,
pstPickPos := ADR(gstPickPos),
pstPlacePos := ADR(gstPlacePos),
pstParkPos := ADR(gstParkPos),
plrPrePickZ_Offset := ,
plrPrePlaceZ_Offset := ,
plrPrePickZ_OffsetVel := ,
plrPrePlaceZ_OffsetVel := ,
xContinuousPrePick := ,
xContinuousPrePlace := ,
xTeachPickPos := ,
xTeachPlacePos := ,
xTeachParkPos := ,
xAckPickOrPlace := ,
lrUpPosZ := ,
lrRadius := ,
udiPickTimer := ,
udiPlaceTimer := ,
xPickPosReady := ,
xPlacePosReady := ,
pCNC => g_stIn.parCNCUser[4],
xPrePick => ,
xPick => ,
xPrePlace => ,
xPlace => ,
iPickCounter => ,
iPlaceCounter => ,
udiCycle => ,
udiCycleTime := ,
tCycleTimePickPlace => ,

```