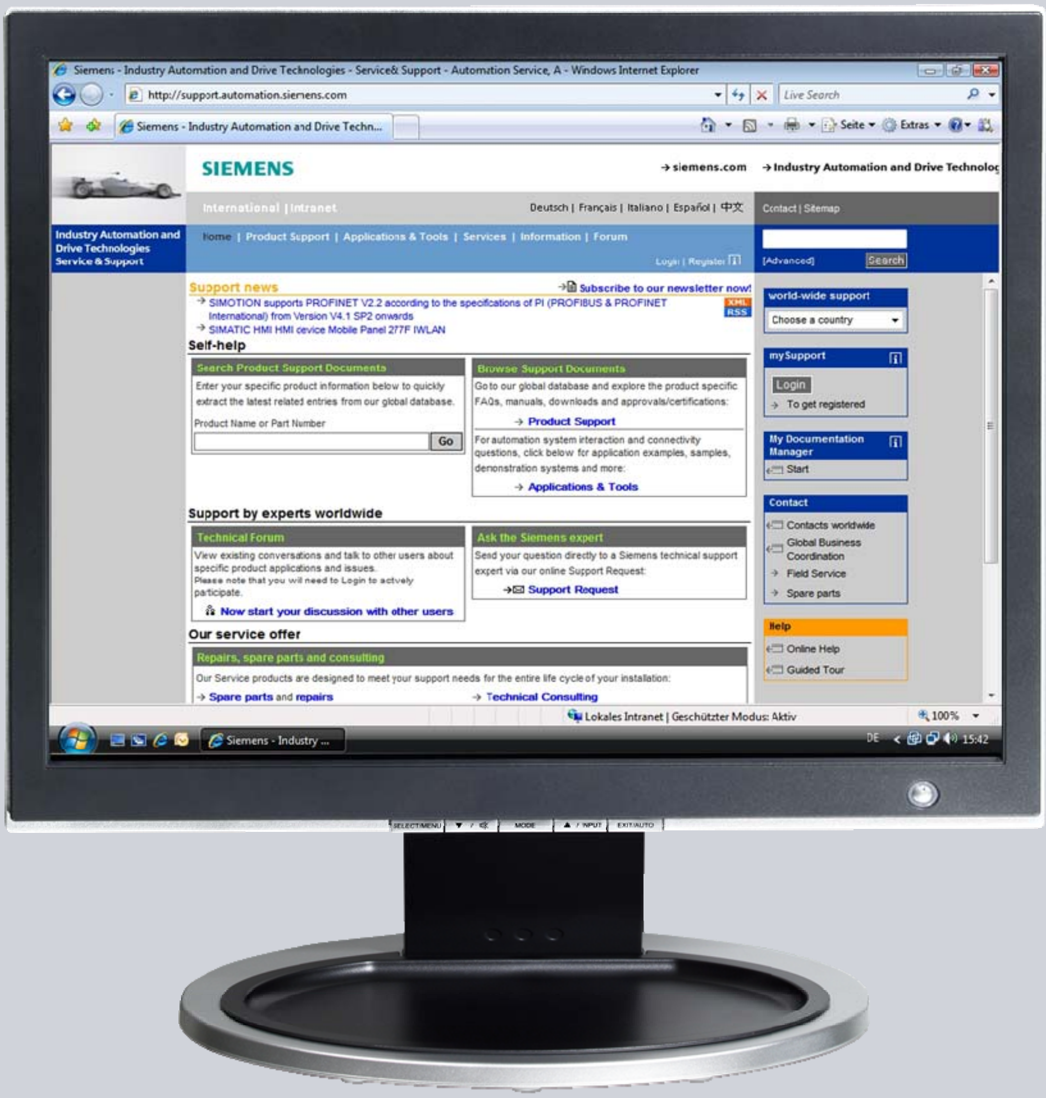
How can a Simple Palletizer be realized with the FB 488 “MC_MovePath” Technology Function?

Technology CPU

FAQ • March 2011



Service & Support

Answers for industry.

SIEMENS

Clicking the link below directly displays the download page of this document.
http://support.automation.siemens.com/WW/view/en/48206062

Question

How can a simple Palletizer be realized in the technology CPU with the FB 488 “MC_MovePath” technology function?

Answer

The instructions and notes listed in this document provide a detailed answer to this question.

Table of content

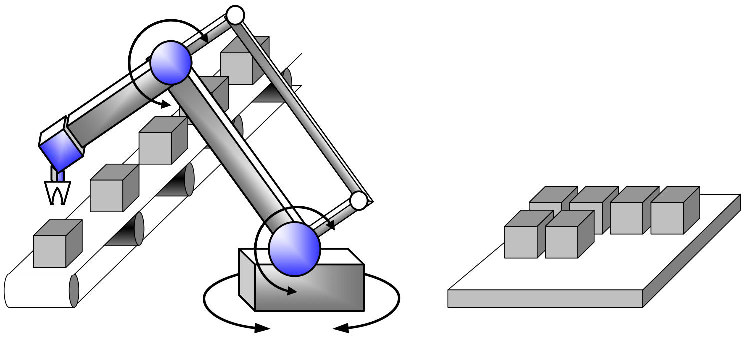
1 Introduction

1.1 Automation Task

A program for a simple palletizing process is to be developed with the help of which…

- …products can be gripped by gripper from a fixed pickup position.
- …products can be placed in different but previously fixed target positions.
- …the products can be brought to individual but previously fixed paths to the target position.
- …the path motion of the palletizer can be interrupted at previously fixed points (stop or wait points).
-
- …the gripper control for products can be realized.

Figure 1-1 Simple palletizing process



The palletizing process during the processing of the palletizing job should be…

- …able to be interrupted and continued.
- …able to be stopped before the end of the palletizing process.

1.2 Automation Solution

The FB 488 “MC_MovePath” technology function of the technology CPU already provides all functionalities for the realization of a palletizing process.

This FAQ is to demonstrate how you can realize a simple palletizer with the help of the FB 488 “MC_MovePath” technology function and what additional technology functions have to be used for this purpose.

This FAQ furthermore shows you the configuration options of the FB 488 “MC_MovePath” technology function for the realization of own applications with this technology function.

2 Program Principle

2.1 Available technology functions

The technology functions listed in the following chapters can be used for the realization of a simple palletizer.

The most important blocks are the calls of the technology functions:

- FB 488 “MC_MovePath” to execute the desired motion.
- FB 489 “MC_PathSelect” to define the motion path.

Note The chapters below only describe selected parameters of the listed technology functions that are helpful for the realization of a simple palletizer.

2.1.1 FB 488 “MC_MovePath”

With the help of this technology function, a path object can be moved by using an interpolation point table.

This technology function is the central element for the realization of the simple palletizer in this FAQ.

Figure 2-1 FB 488 “MC_MovePath” technology function

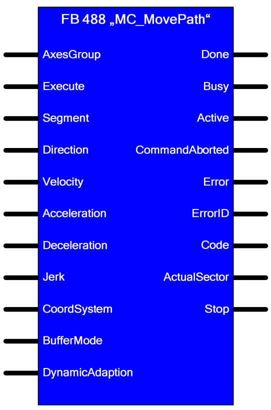


Table 2-1 Description of selected parameters

Parameters	Description
Input parameters	
AxesGroup	Number of the technology data block of the path object that represents the corresponding kinematic.
Segment	Selection of the desired path segment in which the defined motion path is stored. Up to 5 path segments can be defined in the technology.

Parameters	Description
Direction	Selection of the travel direction of the motion path. This parameter specifies whether the interpolation point table is traversed from the first to the last interpolation point or in reverse order. **Note:** to be able to select the respective travel direction here, the path segment already had to have been created in this travel direction when the interpolation point table was read-in through the FB 489 “MC_PathSelect”.
CoordSystem	Selection of the coordination system to which the positions of the travel direction relate. It can be selected between the basic coordinate system (BCS) and the object coordinate system (OCS).
BufferMode	Definition of the transition between a running motion and a subsequent motion that replaces the current motion. Due to the subsequent job, the current motion can … • …be interrupted. As a result the current motion will come to a stop. • …be supplemented at the end of the current motion. The current motion will also come to a stop. • …be supplemented seamlessly. The subsequent motion is blended by adding it to the current motion without the axis to stop.
DynamicAdaption	Adapting the dynamic of the path movement to the configured limit values of the individual path axes. A deviation of the axes motion from the configured path due to dynamic limits of the individual axis can therefore be avoided.
Output parameters	
Active	The path object is currently controlled by the technology function, i.e. the configured motion path is currently moved by kinematic.
Code	Output of the M function as defined in the interpolation point table. In doing so, the integer value entered on the previous interpolation point is output during the motion along a path segment. Once the next interpolation point has been reached, the output is ended or it is switched over to the next integer value. This is how simple path-related additional functions (e.g. opening/closing gripper) can be selected and deselected.
ActualSector	Output of index of the interpolation point belonging to the current path segment, the positive travel direction of the interpolation point can be viewed as the start of the path segment.
Stop	Information that the function has been moved to a stop point defined by the user. A stop point can be defined by indicating a negative corner distance (CornerDistancs) in the interpolation point table.

2.1.2 FB 489 “MC_PathSelect”

With the help of this technology function, the interpolation points of a point table that are present in a data block, are loaded in a segment of a path object.

Figure 2-2 FB 489 “MC_PathSelect” technology function

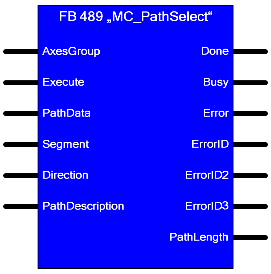


Table 2-2 Description of selected parameters

Parameters	Description
Input parameters	
AxesGroup	Number of the technology data block of the path object that represents the corresponding kinematic and which is to be allocated the motion path to be created.
PathData	Transfer of ANY pointer to an interpolation point table that is structured according to the UDT 120 “PathPoint” and that contains all interpolation points of the desired motion path. 2 up to 240 interpolation points can be stored in the interpolation point table.
Segment	Selection of the desired path segment in which the defined motion path is to be stored. Up to 5 path segments can be defined in the technology.
Direction	Specification of the preparation direction of the interpolation points in the interpolation point table for the subsequent path motion. If the interpolation point table is only prepared in one direction, the path motion can also only be carried out in this direction by means of this table. A preparation in both motion directions is also possible. However, it requires a higher calculation time for the path calculation.
PathDescription	Determination of the execution of the change of direction between the two straight lines at an interpolation point. Within the CornerDistance, the transition can be executed with the help of an arc or a jolt-limited polynomial. The polynomial is created after the maximum possible jerk of the participating individual axes and therefore no longer conforms to an ideal circular path.

Parameters	Description
Output parameters	
ErrorID ErrorID2 / ErrorID3	Output of error information of the technology function. **ErrorID2 (only for selected ErrorIDs):** Output of the index of the initial interpolation point at which the error occurred under ErrorID or of the index of the interpolation point for which an invalid REAL value was entered. **ErrorID3 (only for selected ErrorIDs):** Output of the index of the terminal support at which the error occurred under the ErrorID.
PathLength	Output of the length of the motion path created by means of interpolation point table. The value output here can be used for the definition of measuring sensors, cams and cam tracks at the path-synchronous slave axis. The path-synchronous slave axis moves from position 0.000 to the value output here when moving along the motion path.

Note When using the **UDT 121 "PathData"** for the definition of the interpolation point table, the **UDT 120 "PathPoint"** also has to be available in the program!

UDT 121 "PathData"

With the help of this UDT, the interpolation point table can be defined in a data block for a path motion.

The UDT and therefore the data block have the following structure:

Figure 2-3 UDT 121 "PathData"

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	Point	ARRAY[1..240]		Collection of points defining a segment
*22.0		"PathPoint"		
=5280.0		END_STRUCT		

UDT 120 “PathPoint”

With the help of this UDT the individual interpolation points for the interpolation point table are defined.

The UDT for the individual interpolation points of the interpolation point table has the following structure:

Figure 2-4 UDT 120 “PathPoint”

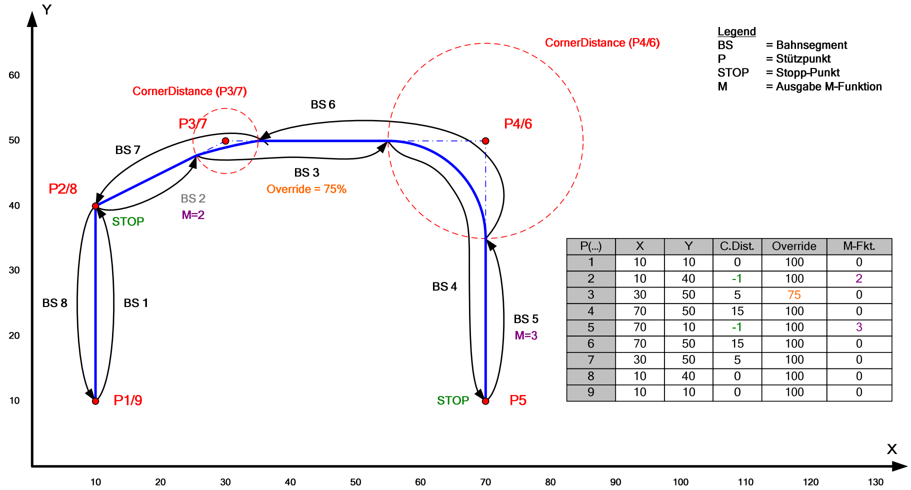
Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	X	REAL	0.000000e+000	X-Coordinate
+4.0	Y	REAL	0.000000e+000	Y-Coordinate
+8.0	Z	REAL	0.000000e+000	Z-Coordinate
+12.0	CornerDistance	REAL	0.000000e+000	Corner distance for transition start.
+16.0	Override	REAL	1.000000e+002	Velocity override for path velocity.
+20.0	M_Function	INT	0	Spezial function code for path point.
=22.0		END_STRUCT		

Table 2-3 Description of the parameters

Parameters	Description
X / Y / Z	Description of the coordinates of the respective interpolation point. For two-dimensional kinematics, the coordinate not used (coordinate outside of the process level) must not be defined.
CornerDistance	Corner distance (spheric around the defined interpolation point), within which the transition between two straight lines and the motion path takes place. **Stop point** If a negative corner distance is entered, the path motion is stopped at this interpolation point. The movement can be continued with the FB 483 “MC_GroupContinue” technology function. If the job at FB 483 “MC_GroupContinue” is started directly before reaching the stop, the interruption of the path motion can be stopped until the next stop point, i.e. there is no interruption of the motion.
Override	Through the override parameter, the travel speed of the each path segment following the interpolation point can be influenced. If the table is traversed from the first to the last interpolation point, each override will have an effect on the path segment following the interpolation point. If the table is followed in reverse direction (last till first interpolation point), the each override setting of the previous interpolation point will have an effect on the path segment.
M_Function	Interpolation point detection defined by the user. As soon as the interpolation point was passed by the motion, the value defined at the code output of FB 488 “MC_MovePath” is output until reaching the next interpolation point. Through this parameter, additional functions (e.g. opening/closing gripper) that are to be executed at certain path positions can be, for example, selected and deselected.

The following graphic is to serve as example for the definition of a path motion through individual interpolation points.

Figure 2-5 Example for the definition of a path motion through individual interpolation points

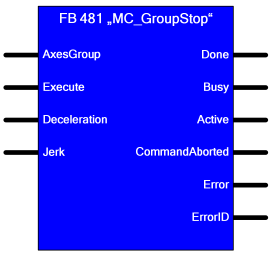


P(...)	X	Y	C.Dist.	Override	M-Fkt.
1	10	10	0	100	0
2	10	40	-1	100	2
3	30	50	5	75	0
4	70	50	15	100	0
5	70	10	-1	100	3
6	70	50	15	100	0
7	30	50	5	100	0
8	10	40	0	100	0
9	10	10	0	100	0

2.1.3 FB 481 “MC_GroupStop”

With the help of this technology function a path motion of a path object can be interrupted, so that the current motion comes to a standstill on a defined path. A continuation of the motion is no longer possible after executing this technology function.

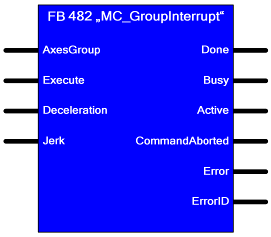
Figure 2-6 FB 481 “MC_GroupStop” technology function



2.1.4 FB 482 “MC_GroupInterrupt”

With the help of this technology function a path motion of a path object can be interrupted on the currently defined path. It is possible to continue the motion via the FB 483 “MC_GroupContinue” technology function.

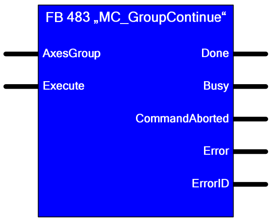
Figure 2-7 FB 482 “MC_GroupInterrupt” technology function



2.1.5 FB 483 “MC_GroupContinue”

With the help of this technology function, a path motion of a path object that was stopped through a stop point (intermediate stop) or interrupted with the FB 482 “MC_GroupInterrupt” technology function, can be continued on the currently defined path.

Figure 2-8 FB 483 “MC_GroupContinue” technology function



2.2 Path objects

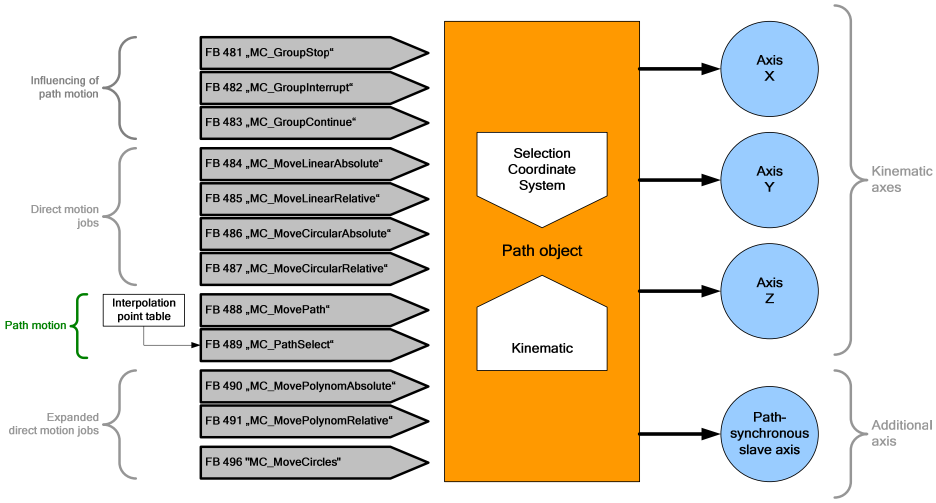
In most cases the handling axes of a machine consist of several individual axes that need to be moved together in a coordinate way. However, the realization of this coordination in the user program requires significant efforts.

The “path object” technology takes on this coordination of the individual axes and therefore also simplifies the use of sophisticated kinematics with the technology CPU.

2.2.1 Function

The principle function of the path object or the switching of the axes is shown in the figure below.

Figure 2-9 Principle function of a path object



Control of the path object

The path object is controlled through various technology functions via which the defined motion jobs can be transmitted:

- **direct motion jobs**
  Through these technology functions, motions in the plane (2D) or in space (3D) can be executed on lines and arcs.
- **path motion**
  Through these technology functions, motions in the plane (2D) or in space (3D) can be executed on a motion path, that was defined through an interpolation point table.
  This functionality is used in the realization of a simple palletizer described here.
- **expanded direct motion jobs**
  Through these technology functions, motions in the plane (2D) or in space (3D) can be executed, …
  - …whose path is predefined through a polynomial.
  - …that are located on an arc and which are defined by specifying an angle segment. This also enables circular motions through an angle > 360° (e.g. several revolutions).

Interpolation point table

Through the interpolation point table there is the option to read various motion paths in the path object that are described through interpolation points. For this purpose, the interpolation point table has to be prepared for the use in the path object through the FB 489 “MC_PathSelect” technology function. The path motion can be executed through the FB 488 “MC_MovePath” technology function.

Axis X / axis Y / axis Z

These axes represent the individual axes of the kinematic and have to be created for the use in connection with a path object as "path axis" technology object in S7T Config.

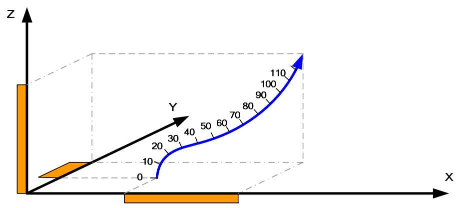
For two-dimensional path objects the assignment of the direction of the axis (X, Y, Z) can be individually selected for the definition of the kinematic. Through this selection the processing level of the machine (XY, XZ, YZ) can be specified.

Path-synchronous slave axis

This axis follows the path motion synchronous (not the individual axes) of the kinematic and can be used for the following functions:

- auxilliary axis for the use of measuring sensors, cams and cam tracks to execute switching procedures on certain positions of the motion path.
- gripper alignment for the "Scara" kinematic

Figure 2-10 Path-synchronous slave axis



Note

> When using the "gripper alignment" function with the "Scara" kinematic, the rotary axis of the gripper can be directly connected to the path object as path-synchronous slave axis. The axis is then directly controlled through the kinematic or the path object.

Path object – kinematic

Through the selection of the kinematic for the path object, the coordination of the individual axes in relation to the movement path is specified.

The kinematics available for the path object are introduced in the next chapter of this document.

Path object – selection coordinate system

Through the selection of the coordinate system, the relation of the path coordinates in the machine between basic (BCS) and object coordinate system (OCS) can be specified.

The relationships of the individual coordinate systems to each other are introduced in the chapter after the next of this document.

2.2.2 Available kinematics

The following kinematics are available for the use in the technology CPU in connection with a path object.

Table 2-4

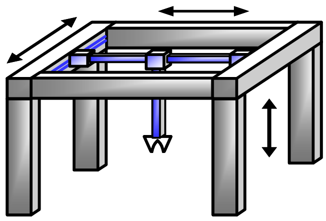
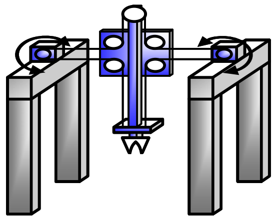
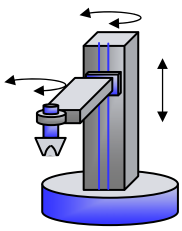
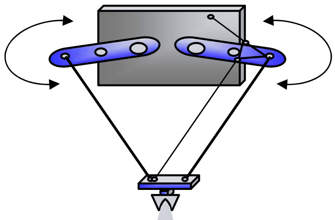
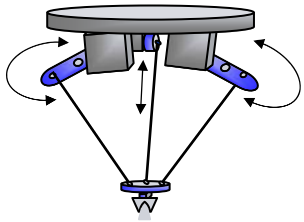
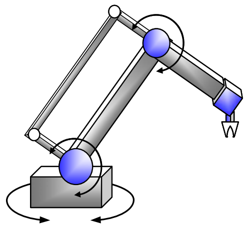
Figure	Dimension	Description
	2D / 3D	**Cartesian gantry** Linear axis movements through linear axes in the plane (2D) and in space (3D).
	2D	**Roll picker (cross slide)** Linear axis movements through two fixed drive axes in the plane (2D) with the help of rotating belt. The rotary direction of the drive axes determines the direction of the linear movement in the plane.
	3D	**SCARA** Axis movements in space (3D) around a central pivot point. The arm of the SCARA kinematic can be equipped with two pivot points to increase the working area of the kinematic.
	2D	**Delta 2 picker** Axis movements in the plane (2D) with high dynamic realized with the help of two rotary axes and mechanical coupler elements. The tool carrier (e.g. gripper) always remains in the same orientation with the work level when moving the Delta 2 picker.

Figure	Dimension	Description
	3D	**Delta 3 picker** Axis movements in space (3D) with high dynamic realized with the help of three rotary axes and mechanical coupler elements. The tool carrier (e.g. gripper) always remains in the same orientation with the work level when moving the Delta 3 picker.
	3D	**Articulated arm (top loader)** Axis movements in space (3D) through the pivotal axis in the base point of the kinematic and the two rotary axes of the articulated arms.

2.2.3 Coordinate systems

The travel motions of a path object can be executed in two different coordinate systems, to which the position details, e.g. of the interpolation point table relate.

- **Basic coordinate system (BCS)**
- **Object coordinate system (OCS)**

The desired coordinate system can each be selected when calling the desired technology function.

Basic coordinate system (BCS)

The basic coordinate system describes a fixed relationship of the kinematic in space. The origin of the basic coordinate system is in the kinematic zero point of the machine.

The position of the basic coordinate system can be moved in relation to the kinematic through offset values when configuring in S7T Config.

Object coordinate system (OCS)

The object coordinate system can be created by translating and rotating the basic coordinate system and serves as a simple and flexible description of a processing procedure on the product, for example, the target position of the palletizer on the pallete.

Machine coordinate system (MCS)

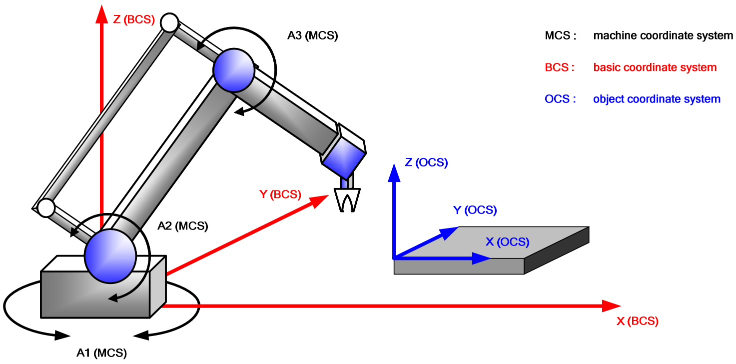
The positions of the individual axes that are controlled through the selected kinematic are summarized in the axis coordinate system (MCS). The individual axes are not aligned in a Cartesian way and therefore do not correspond to the reference axes of the basic coordinate system (BCS).

The "path object" technology object connects the basic coordinate system (BCS) through a "kinematic transformation" with the machine coordinate system (MCS).

Example of the correlation of the coordinate systems

The correlation of the individual coordinate systems is summarized in the figure below.

Figure 2-11 Correlation of the coordinate systems



Note The axes of the **object coordinate system (OCS)** can be moved in relation to the **basic coordinate system (BCS)** and they can also be rotated around the axes of the **basic coordinate system (BCS)**.

Specification of the object coordinate system through FB 480 “MC_SetCartesianTransform”

With the help of this technology function the object coordinate system (OCS) can be created by rotating and translating the basic coordinate system (BCS).

The programming of the interpolation points in the object coordinate system (OCS) can be simplified through the direct reference to the part.

The object coordinate system (OCS) can furthermore also be linked to a tracking axis and it can be moved synchronously with this axis through this technology function. This facilitates the programming of the processes on a moved part enormously.

Figure 2-12 FB 480 “MC_SetCartesianTransform” technology function

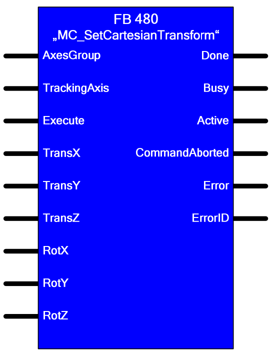


Table 2-5 Description of selected parameters

Parameters	Description
Input parameters	
TrackingAxis	Specification of the number of a tracking axis (e.g. conveyor belt) with which the coordinate system is to be moved synchronously. If the value <> 0 is entered, the CPU accepts the object coordinate system (OCS) as moved. Motion jobs in the object coordinate system (OCS) can only be executed when the object coordinate system is synchronized with the conveyor belt axis. Synchronization can be carried out through the FB 494 "MC_GroupSyncConveyorBelt" technology function.
TransX / TransY / Trans Z	Specifying the translation of the object coordinate system (OCS) in the individual axis direction starting from the basic coordinate system (BCS). The translation of the coordinate system is performed before rotating the coordinate system.
RotX / RotY / RotZ	Indicating the rotation of the translated object coordinate system (OCS) around the individual axes. The rotations around the individual axes are performed in the following sequence: • rotation around X axis • rotation around Y axis • rotation around Z axis When determining the necessary rotation angles, note the sequence of the rotation processes, otherwise you might receive a different target coordinate system.

Note When the interpolation points are programmed for the palletizer in the object coordinate system (OCS), the pickup position of the products which usually presents a fixed position in the basic coordinate system (BCS), also has to be adjusted for each new specification of the object coordinate system (OCS), i.e. it has to be transformed in this coordinate system.

2.3 Program sequence

2.3.1 Preparations

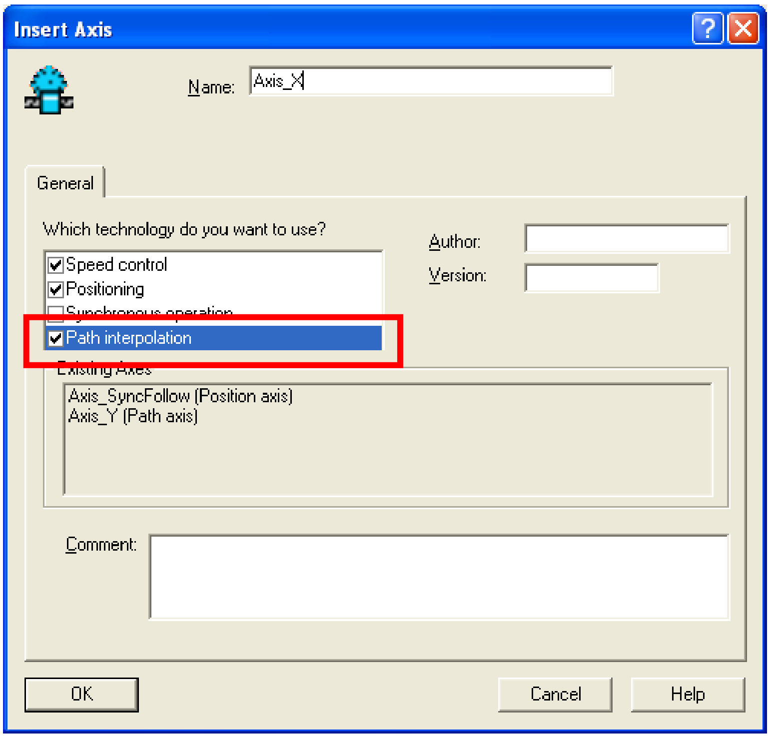
This chapter briefly explains what technology objects have to be created and interconnected in the technology CPU to create a simple palletizer.

Note To keep the presentation as simple and understandable as possible, a 2D kinematic with 2 kinematic axes and a path-synchronous slave axis was chosen as sample program.

Creating the kinematic axes

When creating the kinematic axes (**Axis_X** and **Axis_Y** of processing level XY) the "path interpolation" technology has to be selected. Otherwise the respective axis cannot be used as kinematic axis for a path object.

Figure 2-13 Creating the kinematic axes



The other axis settings are performed according to the respective parameters of the present axis.

Creating a path-synchronous slave axis

If a path-synchronous **Axis_SyncFollow** slave axis is to be used on the path object, it is sufficient to create this axis as "positioning axis" in S7T Config.

Creating the path object

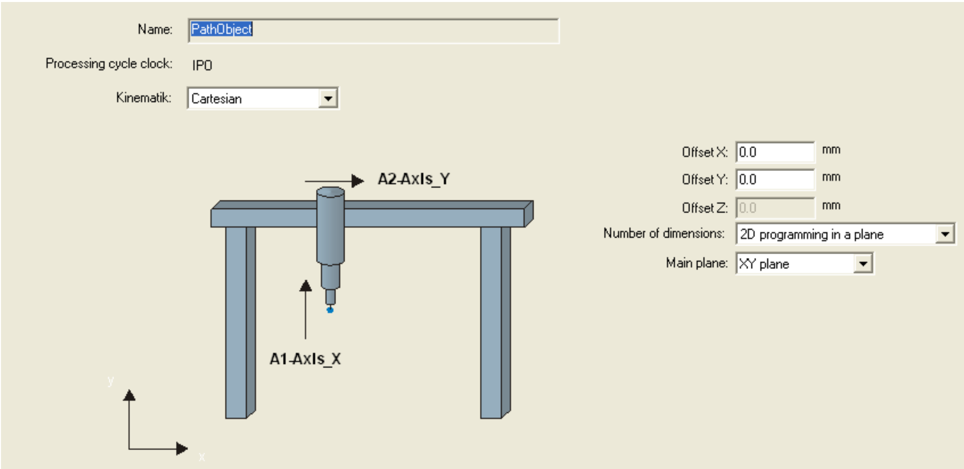
When creating the path object, the following configuration steps have to be performed:

- configuration of the desired kinematic of the path object
- setting the default settings of the dynamic values of the path object
- setting of the dynamic limits of the path object
- interconnection of already created axes (kinematic axes and path-synchronous slave axis) with the path object

When configuring the desired kinematic of the path object, the following settings always have to be configured:

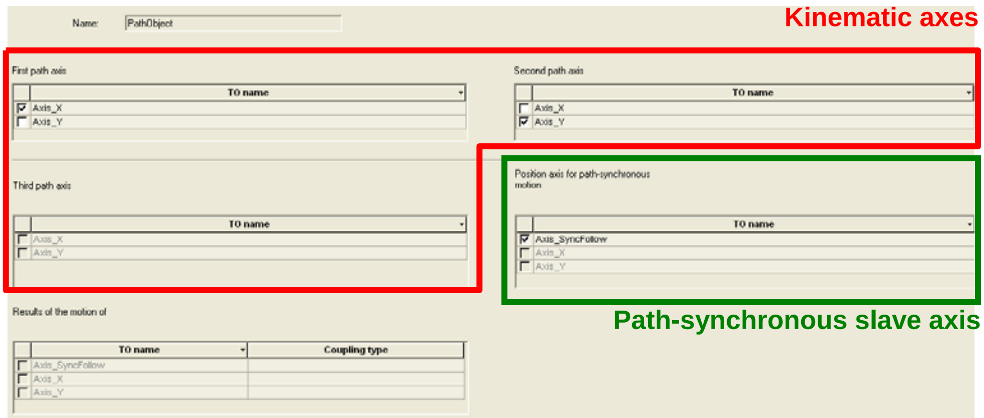
- selection of the desired kinematic, including the specification of various setting values to determine the kinematic characteristics and the kinematic dimensions (2D or 3D).
- definition of an offset to adjust the kinematic origin to the origin of the machine coordinate system (MCS).
- specification of the processing level

Figure 2-14 Configuration of the kinematic of the path object



With the interconnection of the kinematic axes, the axes already created in S7T Config or the virtual axes are assigned to the path object. The path-synchronous slave axis is also defined at the same time.

Figure 2-15 Interconnection of already created axes



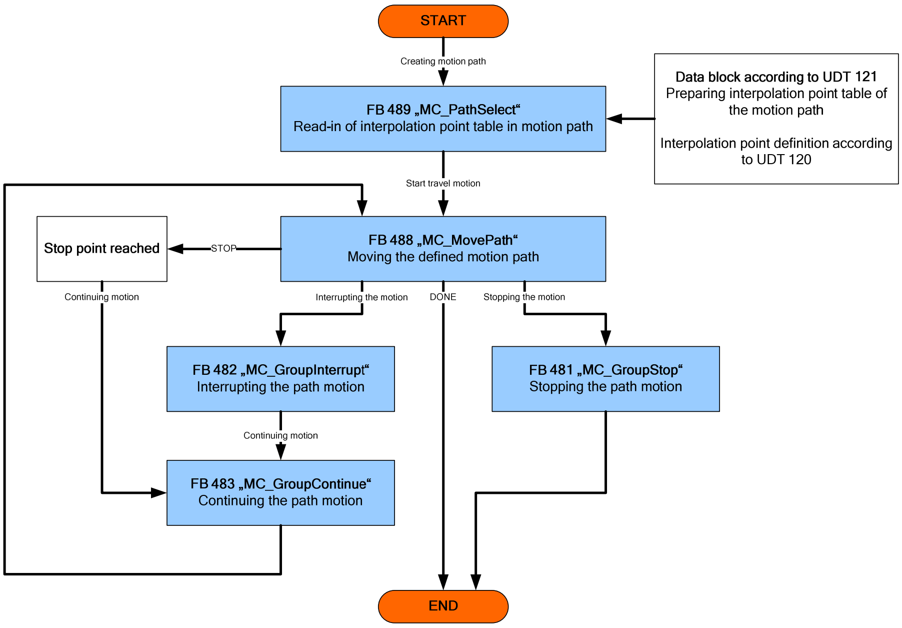
Note Additionally, an axis (tracking axis), for example, a conveyor belt axis can also be defined here, to which the object coordinate system (OCS) of the path object can be synchronized and which can be moved together with this axis.
The conveyor belt axis (tracking axis) has to be aligned in a way so that it is positioned in the direction of the X axis of the object coordinate system.

2.3.2 Programming the palletizer

The palletizing process can be realized with the FB 488 “MC_MovePath” technology function. The following steps are necessary:

- defining the interpolation point table
- read-in of the interpolation point table through FB 489 “MC_PathSelect”
- starting travel motion through FB 488 “MC_MovePath”
- if necessary, cancel, interrupt and continue travel motion.

Figure 2-16 Program sequence of the palletizing process



2.3.3 Useful tools to realize the program

When realizing the sample program some useful tools were used, that are briefly introduced here. These blocks can also be used independently from this program example in correlation with the FB 488 “MC_MovePath” technology function.

Creating the ANY pointer on the interpolation point table

The transfer of the interpolation points, stored in a data block, from the interpolation point table to the FB 489 “MC_PathSelect” is performed through an ANY pointer.

The block introduced here will help you to create this ANY pointer. Through the block introduced here, an ANY pointer can furthermore also be realized quickly and easily on a sub-area of the data block, for example, when only sub-areas of the interpolation point table stored in the data block are to be moved.

Figure 2-17 FC 530 “Path_IdxToAny” block

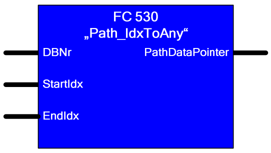


Table 2-6 Parameter of the FC 530 “Path_IdxToAny”

Parameters	Function	Note
Input parameters		
DBNr	Number of the data block in which the interpolation point table is stored for the travel motion.	The data block has to be structured based on UDT 121 or UDT 120.
StartIdx	Number of the index of the first interpolation point of the travel motion.	1 ≤ StartIdx ≤ EndIdx
EndIdx	Number of the index of the last interpolation point of the travel motion.	StartIdx ≤ EndIdx ≤ 240
Output parameters		
PathDataPointer	ANY pointer, which can be transferred to FB 489 “MC_PathSelect” to read-in the desired path segment.	

Enables of the path-synchronous slave axis

To use the path-synchronous slave axis configured on the path object, it has to be enabled through parameter **5710** (userdefault.w.mode) of the expert list of the path object with the help of the FB 407 “MC_WriteParameter” technology function.

The block introduced here helps you to enable the path-synchronous axis in the style of the FB 401 “MC_Power” technology function.

Figure 2-18 Block FB 525 “AxisSyncFollowEnable”

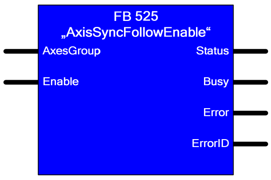


Table 2-7 Parameter of FB 525 “AxisSyncFollowEnable”

Parameters	Function	Note
Input parameters		
AxesGroup	Number of the path object whose path synchronous slave axis is to be enabled.	
Enable	By setting this input, the path-synchronous slave axis is enabled. By resetting the input, the enable of the axis is reversed.	The setting or resetting of the enable of the path-synchronous slave axis has to be performed **before** the start of the path motion through the FB 488 “MC_MovePath”. Otherwise the state of this input has no influence on the path-synchronous slave axis.
Output parameters		
State	Internal status of the blocks, whether the enable was set or reset through the block via the FB 407 “MC_WriteParameter” technology function.	If the parameter of the path object was manipulated, for example, through an additional write access, the output parameter will not correspond to the actual status of the path-synchronous slave axis.
Busy	Display of the write process of the parameter through the FB 407 "MC_WriteParameter” technology function.	
Error	When writing the parameter through the FB 407 “MC_WriteParameter” an error occurred.	
ErrorID	Output of the error code of FB 407 “MC_WriteParameter”.	

DANGER

The setting or resetting of the enables of the path-synchronous slave axis during an active path motion (even when the axes are stopped at a stop point) has no influence on the axis.

The setting or resetting of the enables of the path-synchronous slave axis is only performed through the FB 488 “MC_MovePath” when starting the path motion.

Conversion of code output in individual bits

The “Code” (M function) output as integer value at FB 488 “MC_MovePath” can be separated in bit by bit information with the help of this block.

Figure 2-19 Block FC 531 “CodeToBit”

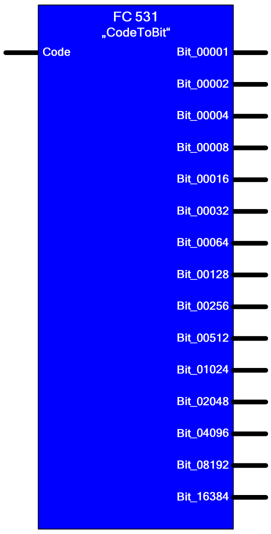


Table 2-8 Parameter of the FC 531 “CodeToBit”

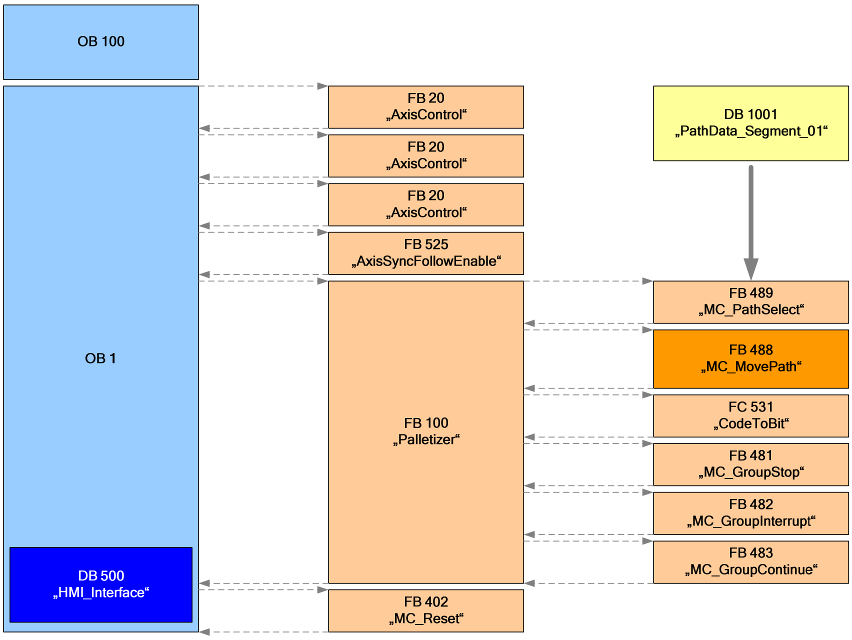
Parameters	Function	Note
Input parameters		
Code	The “code” value (M function) output at FB 488 “MC_MovePath” as integer value.	
Output parameters		
Bit_00001 Bit_00002 Bit_00004 Bit_00008 Bit_00016 Bit_00032 Bit_00064 Bit_00128 Bit_00256 Bit_00512 Bit_01024 Bit_02048 Bit_04096 Bit_08192 Bit_16384	Output of bits that can be represented through the input parameters.	The name of the outputs represents the integer values that have to be indicated on the input for the control of the output. By summarizing the indicated values, several outputs can be controlled.

3 Sample Program

3.1 Realization

The sample program is realized in the following figure through the displayed program structure.

Figure 3-1 Program structure



The individual program blocks take on the following functions:

Table 3-1 Function of the program blocks

Block	Function	Note
OB 100	Initialization of the HMI user interface and the function blocks of the program with the default values of the sample program.	
OB 1	Call of the function blocks of the program, according to the program structure shown above.	
DB 500	Central interface block for the data transfer of sample programs for the HMI user interface	
FB 20	This block is used with the respective instance for enabling, resetting and referencing of the axes of the sample program.	The block uses the technology functions: • "MC_Power" • "MC_Reset" • "MC_Home"
FB 100	Central block to realize the entire functionality of the palletizer.	In this block, the necessary technology functions for the palletizer process are

Block	Function	Note
		called.
FB 402	Block for resetting the path object.	
FB 489	Generating the segment of the path object from the interpolation point table stored in the data block.	The interpolation point table is stored in **DB 1001** of the sample program.
FB 488	Realizing the palletizing process by means of the motion path copied in the path segment from the interpolation point table.	
FC 531	Transmission of the code information (M function) in individual bits output by FB 488 “MC_MovePath”.	The block serves the simplification of the control of switching functions through the M functions of the interpolation point table.
FB 481	Stopping of the path motion.	
FB 482	Interrupting the path motion with the option to continue.	
FB 483	Continuing the path motion after an interruption or after moving to a stop point.	
DB 1001	Interpolation point table of the complete travel motion of the palletizing process.	A complete palletizing process can be achieved by a single call of the “MC_MovePath”.

Note

The technology functions of the palletizing process that are called in the FB 100 “Palletizer” are realized as multi-instances of this block.

3.2 Description of the program networks

The following chapter only deals with the networks of the **FB 100 “Palletizer”,** that contain the technology function calls necessary for the palletizing process.

FB 100 “Palletizer” - Network 1 to 3

The first three networks of the block serve for the initialization of the sample program of the simple palletizer.

In **Network 1** the path object with the selected kinematic is assigned to the multi-instances of the FB 488 “MC_MovePath” and FB 489 “MC_PathSelect” technology function blocks. This means that the motion paths can be prepared by means of the interpolation point table for the kinematic axes of the path object.

In **Network 2** the path object is also transferred to the multi-instances of the FB 481 “MC_GoupStop”, FB 482 “MC_GoupInterrupt” und FB 483 “MC_GoupContinue” technology function blocks to influence the path motion.

To store the motion path by means of the interpolation point table, segment 1 is set by default in the sample program in **network 3**.

Figure 3-2 FB 100 “Palletizer” - Network 1

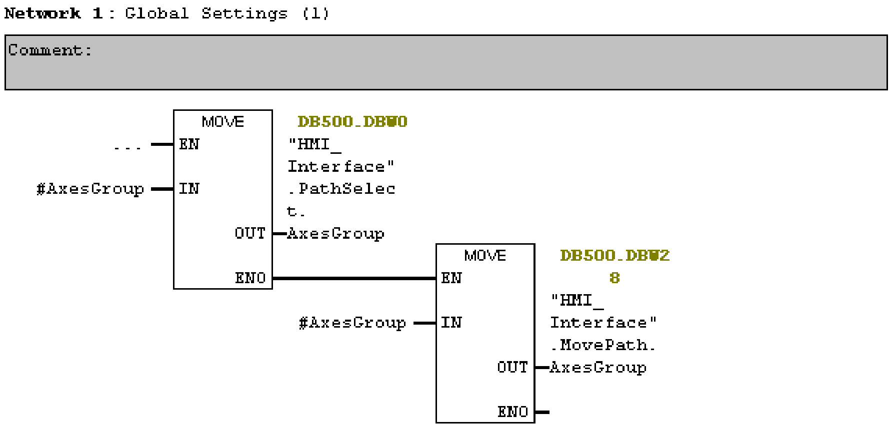


Figure 3-3 FB 100 “Palletizer” - Network 2

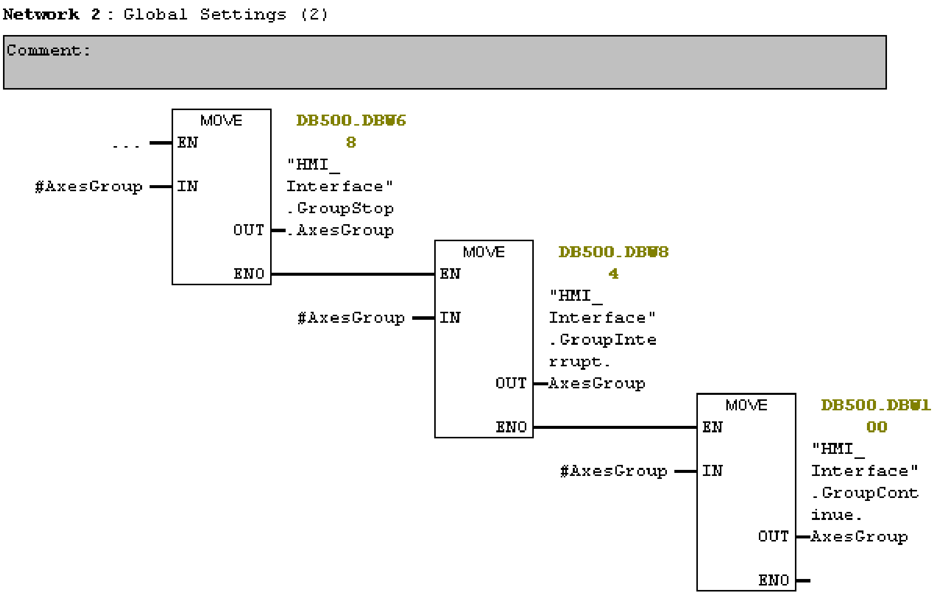
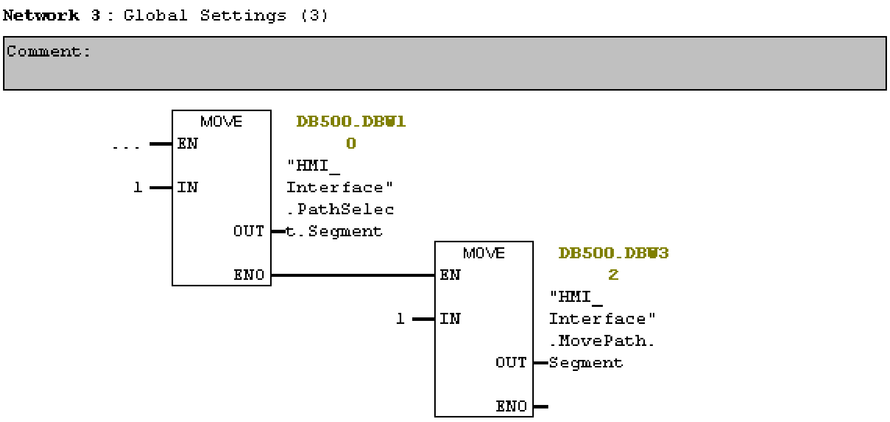


Figure 3-4 FB 100 “Palletizer” - Network 3

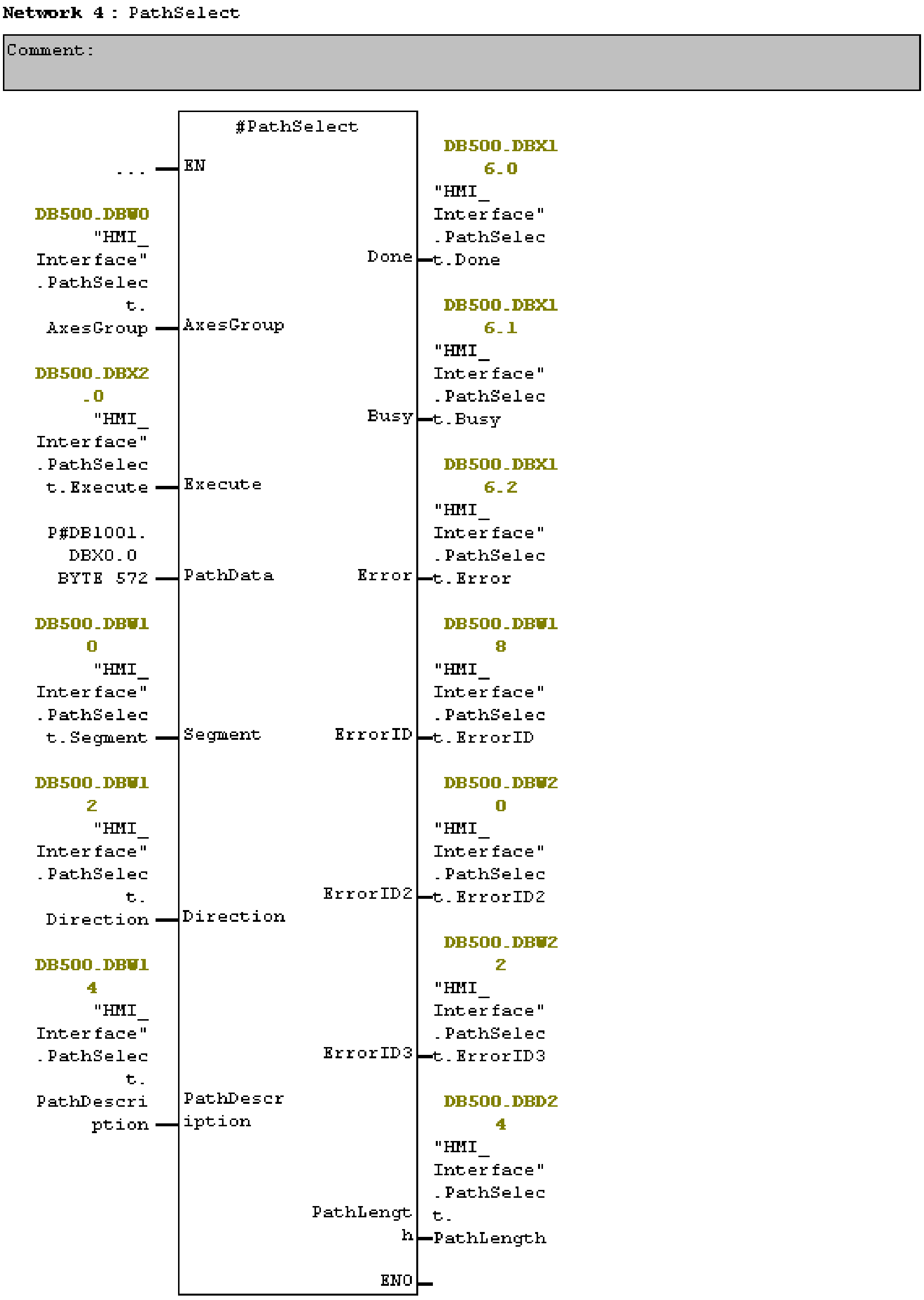


FB 100 "Palletizer" - Network 4

In this network, the interpolation point table is prepared as the basis for the travel motion. The interpolation points from **DB 1001** are transferred to the block with the aid of ANY pointers. This block subsequently prepares the travel motion for the respective kinematic of the path object and stores it in the non-adjustable segment 1 of the technology CPU.

The preparation is started through the **Execute** input of the block. After successful preparation of the travel motion, the travel length of the path-synchronous slave axis is given back during the path motion on the **PathLength** output.

Figure 3-5 FB 100 "Palletizer" - Network 4

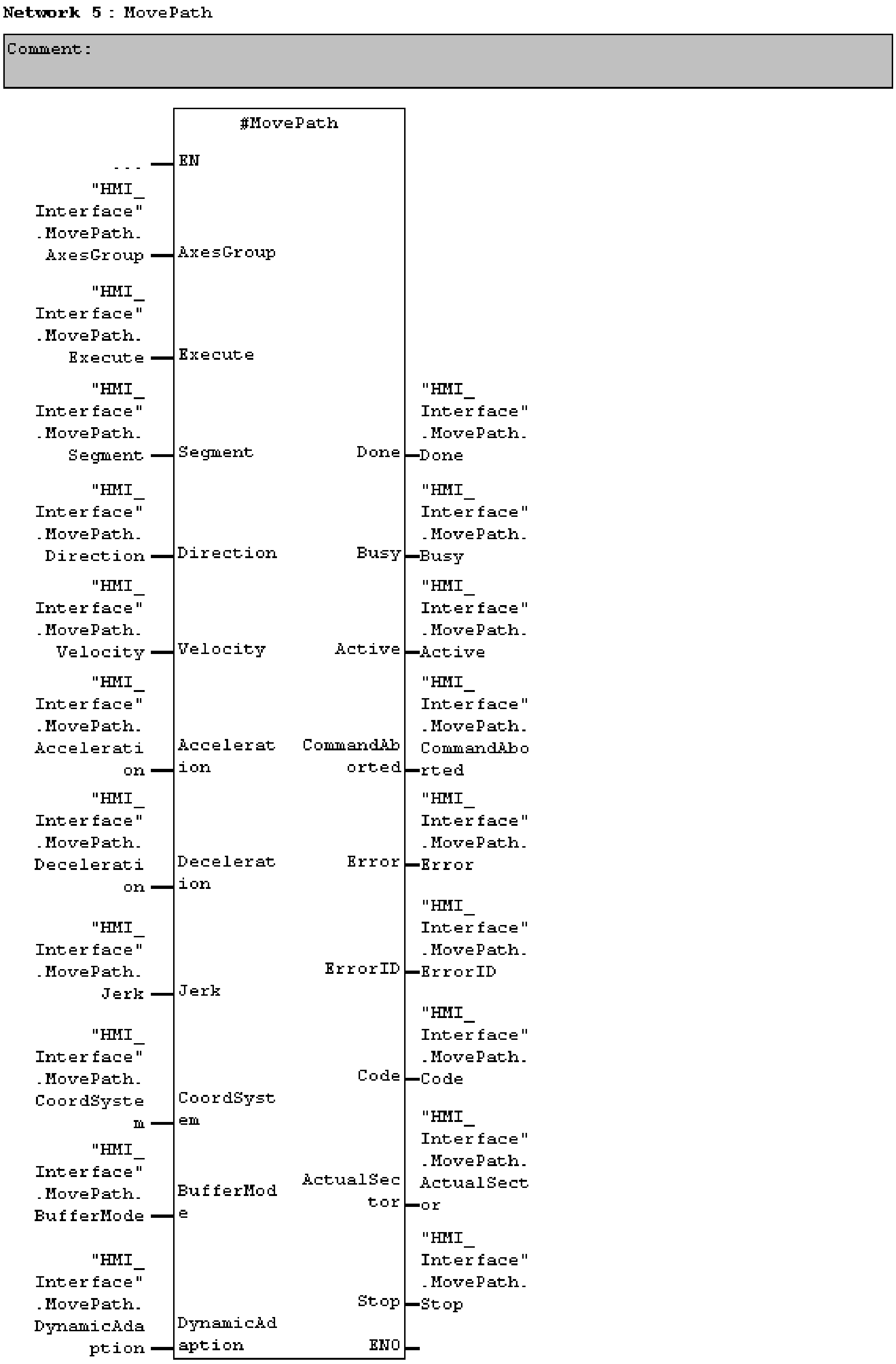


FB 100 “Palletizer” - Network 5

In this network the travel motion, i.e. the actual palletizing process is triggered.

Through the **Execute** input, the travel motion is started by means of the interpolation point table stored in the **Segment** with the preset dynamic (**Acceleration**, **Deceleration**, **Jerk**).

Figure 3-6 FB 100 “Palletizer” - Network 5



The **Busy** output of the block is set, as long as the block for the motion control of the path object is active.

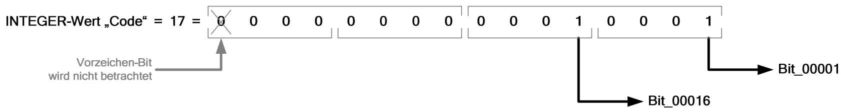
Through the **Active** output, the block furthermore shows that the path object is currently controlled by the technology function. This means, the axes of the path object are in motion or are at a stop point. In this case, the **Stop** output of the block is set additionally.

The currently processed element of the interpolation point table (Index = ActualSector) is displayed by the block at the **ActualSector** output.

The code of the M function, configured in the interpolation point table is output on the **Code** output of the block.

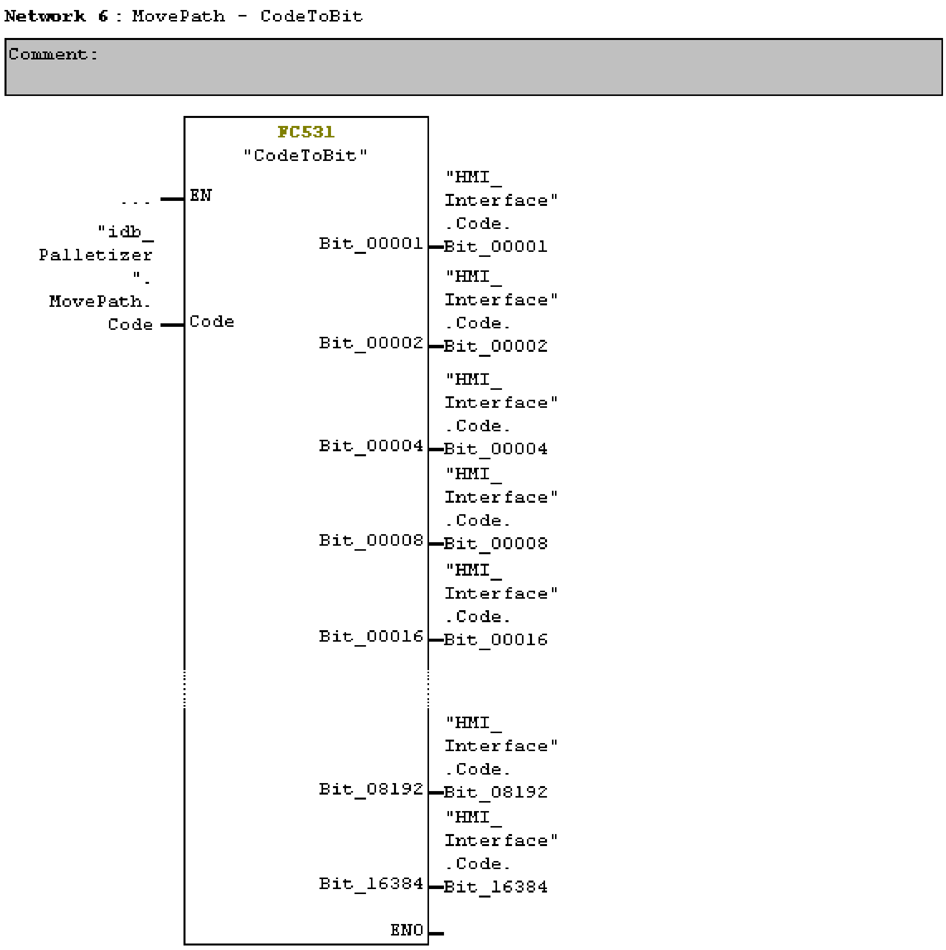
FB 100 “Palletizer” - Network 6

In network 6 the code of the M function from the interpolation point table that is output on the FB 488 “MovePath” technology function is transferred to the FC 531 “CodeToBit” to separate the integer value into individual bits.



-

Figure 3-7 FB 100 “Palletizer” - Network 6



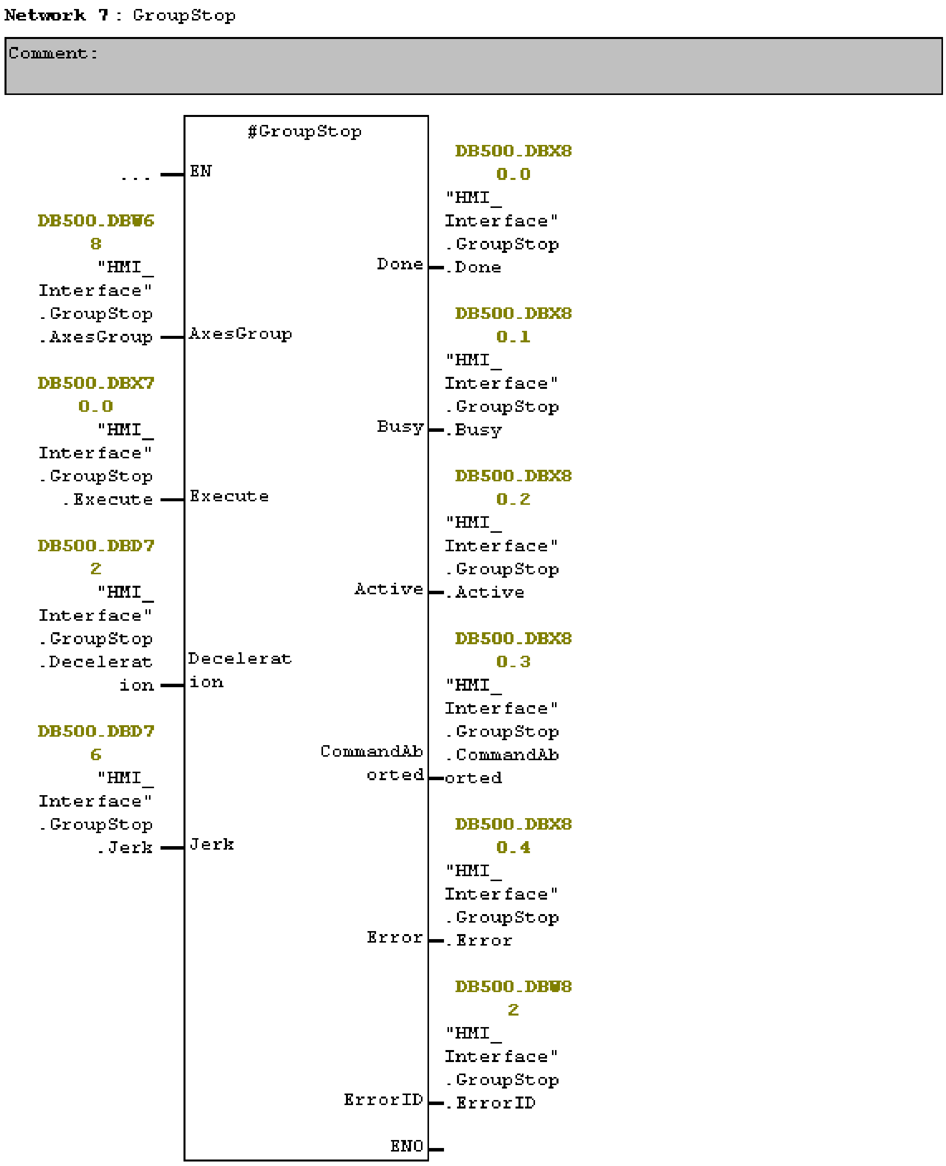
This is how the individual bits can be directly used on the palletizer as control information, for example, for the gripper control or to request new materials.

FB 100 "Palletizer" - Network 7

In this network an active travel motion can be interrupted through the FB 481 "MC_GroupStop" technology function without being able to continue, even if the path object is at a stop point.

When setting the **Execute** input, the currently active path motion is decelerated whilst taking the preset **Deceleration** and **Jerk** dynamic values into consideration and the travel motion on FB 488 "MC_MovePath" is interrupted.

In the process, the **Active** output signals that the path object is currently controlled by the technology function.



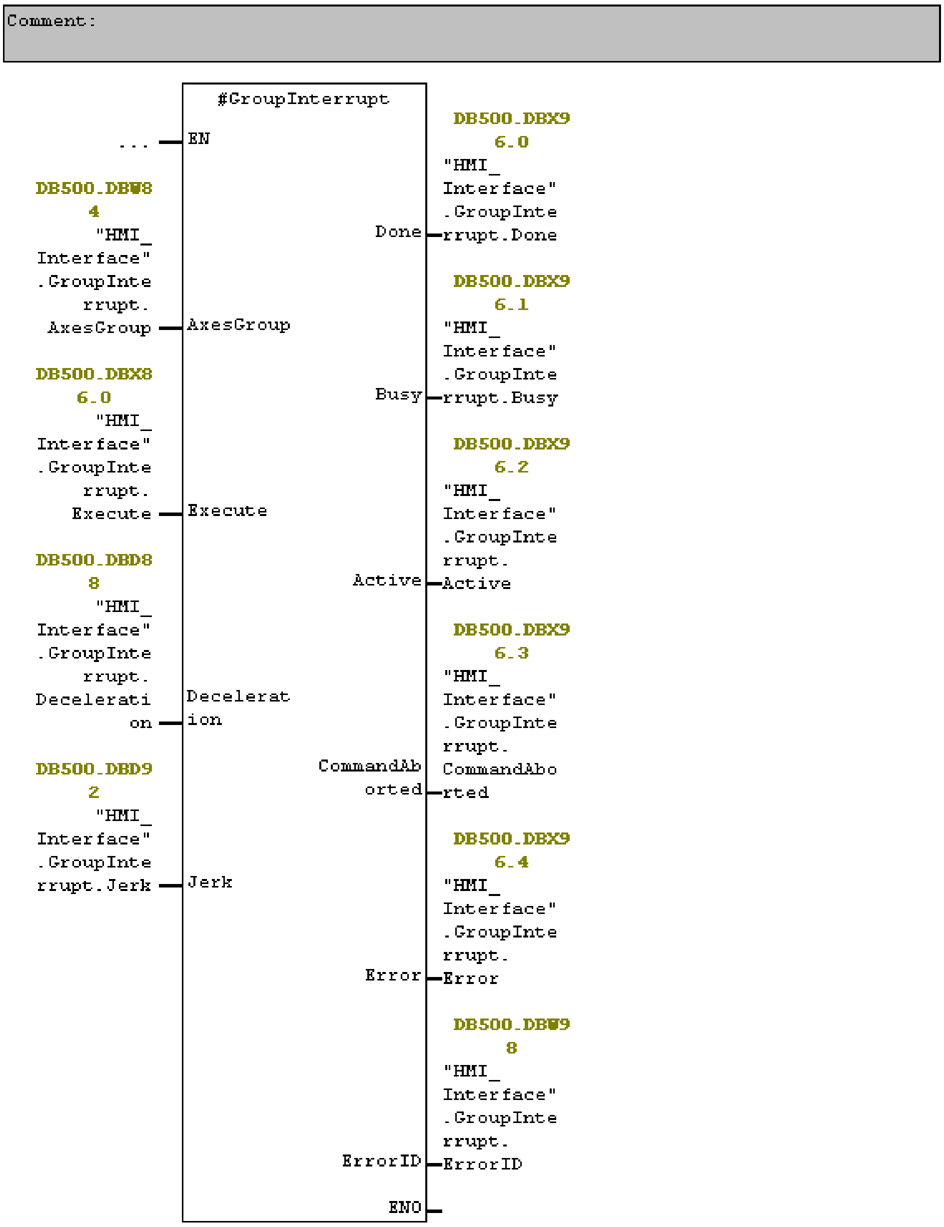
FB 100 “Palletizer” - Network 8

In this network an active travel motion can be interrupted through the FB 482 “MC_GroupInterrupt” technology function and continued later, even if the path object is at a stop point.

When setting the **Execute** input the currently active travel motion is decelerated whilst taking the preset **Deceleration** and **Jerk** dynamic values into consideration. The travel motion at FB 488 “MC_MovePath” is interrupted and can be continued through the FB 483 “MC_GroupContinue” technology function.

During the execution of the technology function, the **Active** output signals that the path object is currently controlled by the technology function.

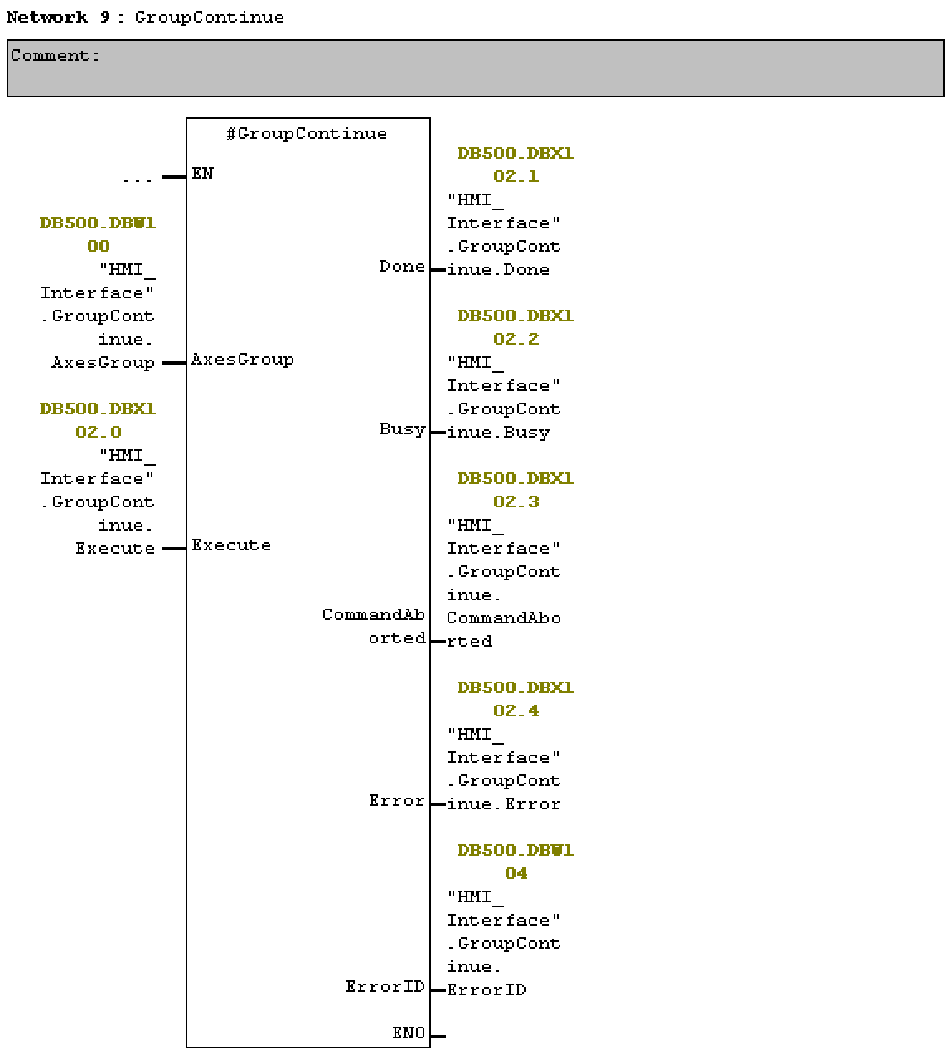
Network 8 : GroupInterrupt



FB 100 “Palletizer” - Network 9

In this network a travel motion can be continued that is at stop point or that was interrupted through the FB 482 “MC_GroupInterrupt” technology function.

When setting the **Execute** input, the FB 488 “MC_MovePath” instantly continues with the processing of the travel motion defined through the interpolation point table.



3.3 HMI user interface of the sample program

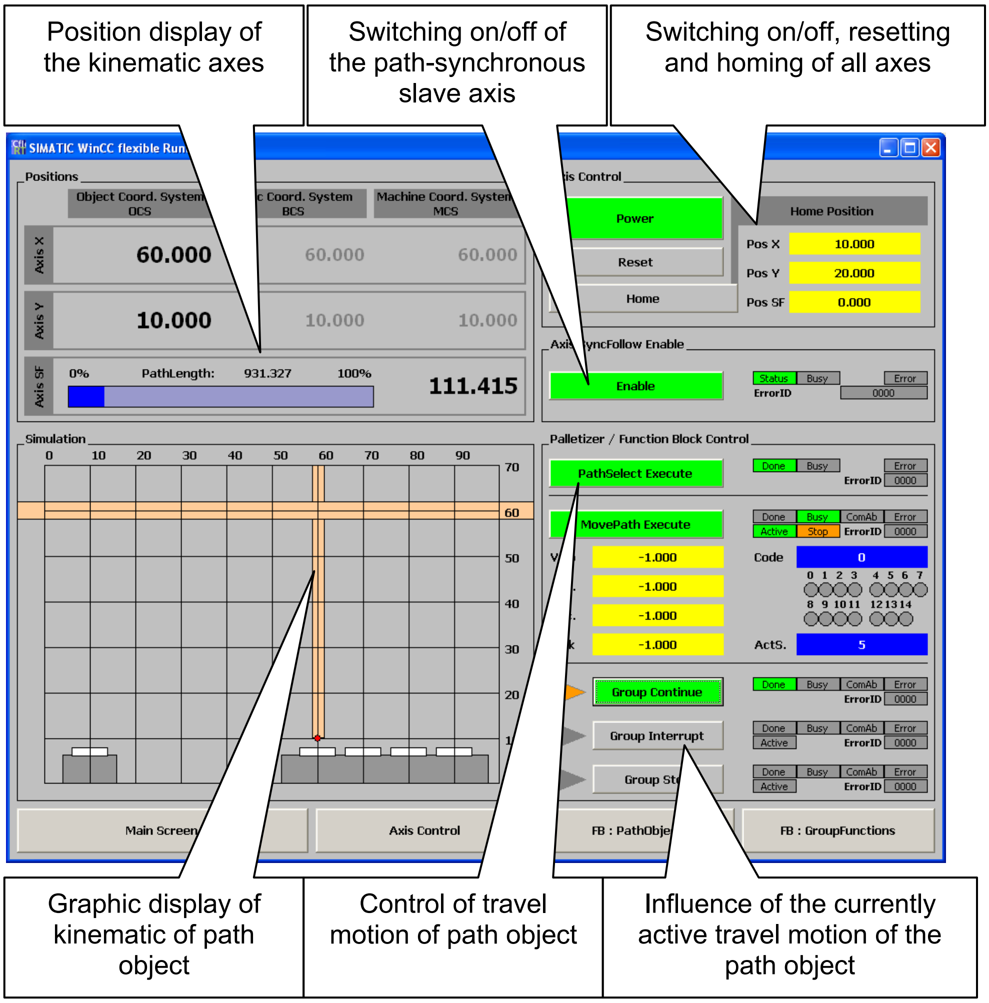
The sample program introduced here, has a HMI user interface through which the individual blocks described here can be conveniently operated.

3.3.1 HMI user interface – main screen

Through the main screen of the HMI user interface, all function blocks of the sample programs that are necessary for the realization of the palletizing process can be controlled.

In addition, the selected kinematic (gantry) of the path object is displayed in animated form, so that the travel motion of the kinematic axes can be optically controlled through the user interface.

Figure 3-8 HMI user interface – main screen



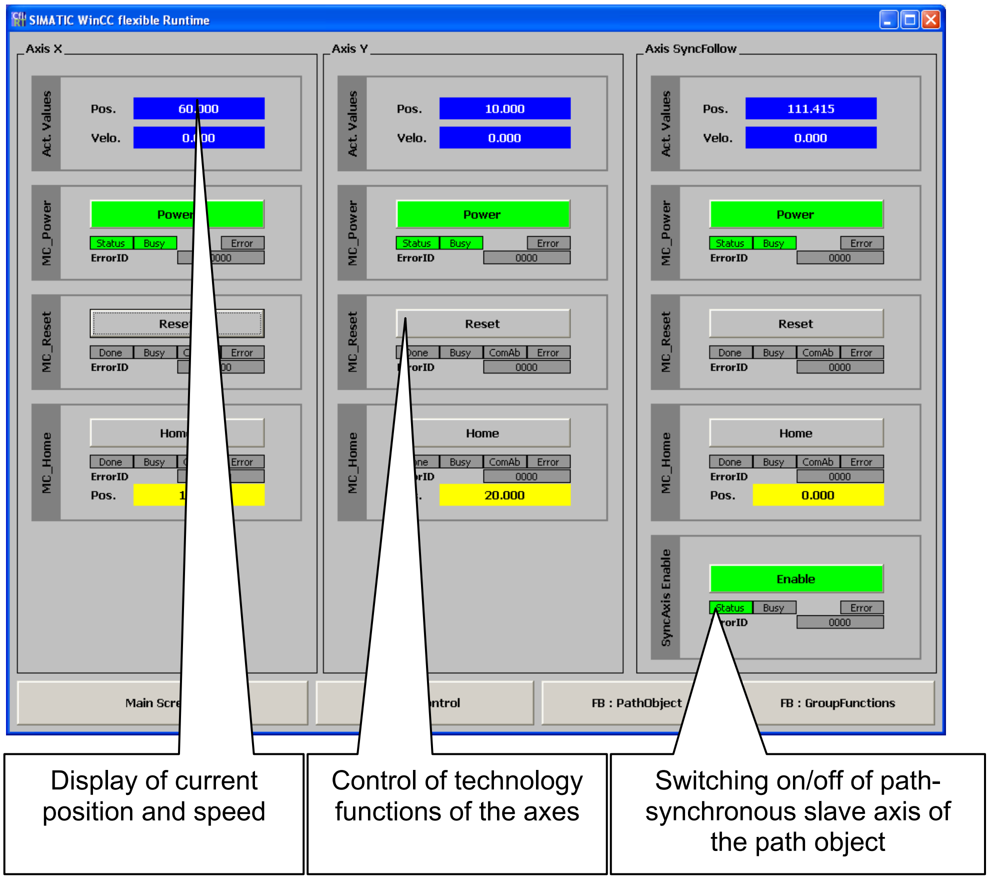
Note

> Through the **"Axis Control"** operator display of the HMI user interface, all axes of the sample program are influenced, i.e. the kinematic axes of the path object and the path-synchronous slave axis (switching on/off, resetting and referencing).

3.3.2 HMI user interface – axis control

With the “Axis Control” screen, all axes of the sample program can be influenced and monitored separately. The FB 401 “MC_Power”, FB 402 “MC_Reset” and FB 403 “MC_Home” technology functions of the axes can be controlled and monitored separately here.

Figure 3-9 HMI user interface – axis control



Note

The switching on/off of the path-synchronous slave axis of the path object is performed with the aid of the **FB 525 “AxisSyncFollowEnable”** tool block with the help of which the **Parameter 5710** (userdefault.w.mode) of the path object is described through the **FB 407 “MC_WriteParameter”** technology function.

3.3.3 HMI user interface – FB PathObject

With the “FB PathObject” screen, the main functions can be controlled and monitored to realize the palletizer.

Through this screen the parameters of the FB 489 “MC_PathSelect” and FB 488 “MC_MovePath” technology functions can also be changed which are no longer listed on the main screen of the HMI user interface.

For the parameters that can be called by specifying a fixed parameter value, an input help is available that can be reached through the “question mark” button. The available functions are mentioned in plain text. By selecting the desired function, the respective numeric value is entered at the input of the block.

Figure 3-10 HMI user interface – FB PathObject

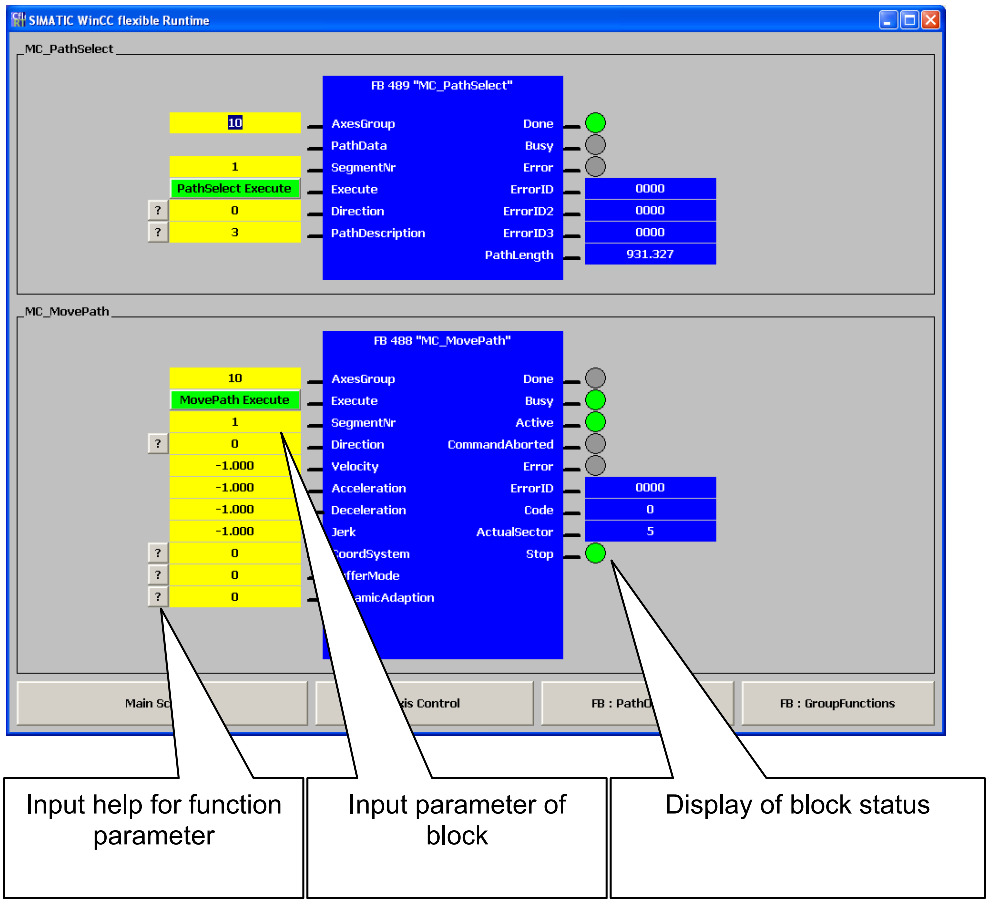
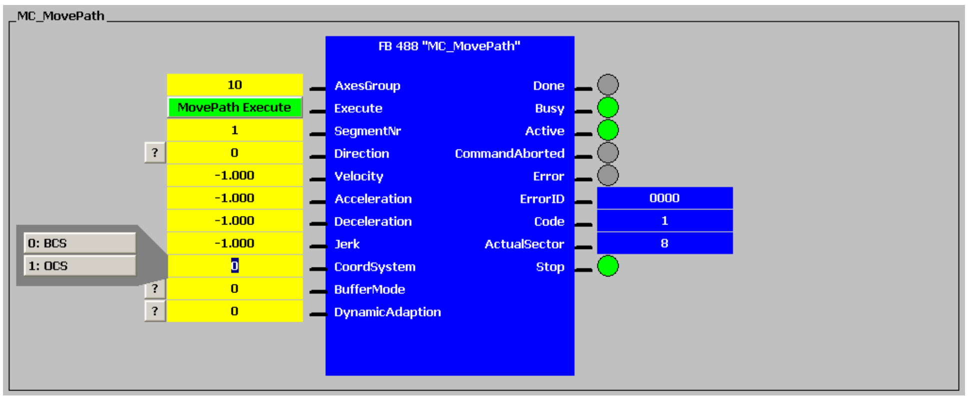


Figure 3-11 HMI user interface – FB PathObject – use of the input help

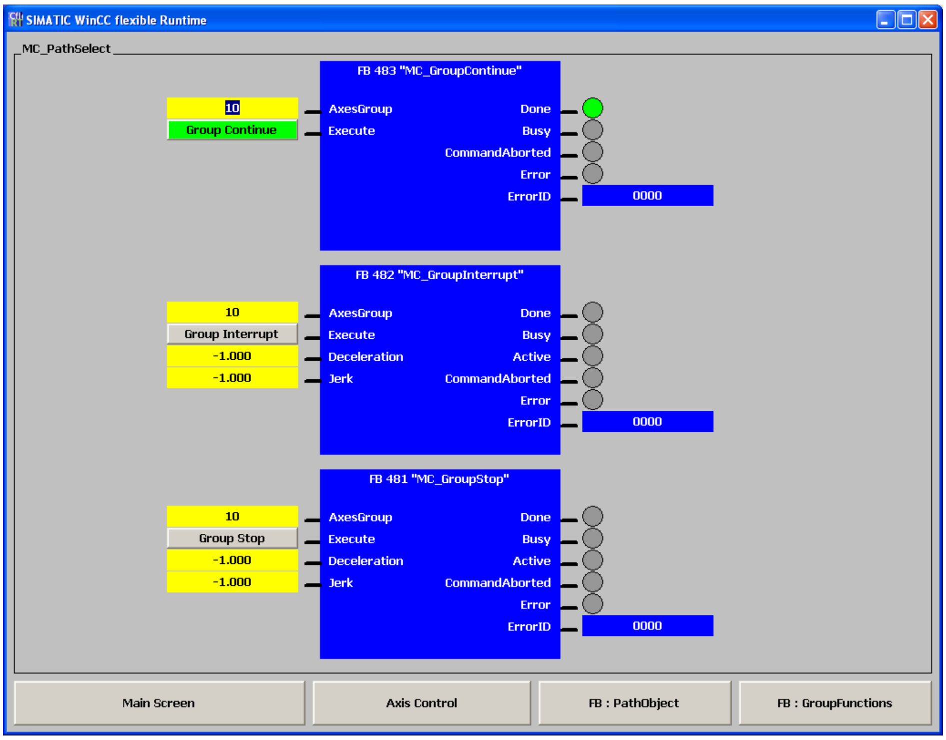


3.3.4 HMI user interface – FB GroupFunctions

With the “FB GroupFunctions” screen, the technology functions to influence the travel motion of the palletizer can be controlled and monitored.

Through this screen the parameters of the FB 481 “MC_GroupStop” and FB 482 “MC_GroupInterrupt” technology functions can also be changed which are no longer listed on the main screen of the HMI user interface.

Figure 3-12 HMI user interface – FB GroupFunctions



4 Additional Notes

4.1 Handling of additional motion paths

4.1.1 Creating and using other segments

Within the technology CPU for the interpolation, up to 5 motion paths, the so called "segments", with up to a max. of 240 interpolation points can be managed simultaneously.

Whilst a travel motion (segment) is performed through the FB 488 “MC_MovePath” technology function, other interpolation point tables can be read into the segments in parallel through the FB 489 “MC_PathSelect” technology function.

Note During the read-in of additional interpolation point tables, please note the load of the technology CPU during this process.

The additionally created segments can later simply be executed by calling the FB 488 “MC_MovePath” technology function.

4.1.2 Moving along the paths in forward and backward direction

The travel motion of the interpolation point table has to be specified during the creation of the motion path through the FB 489 “MC_PathSelect” technology function.

The travel motion of the interpolation point table can be set through the **Direction** input parameter via the FB 488 “MC_MovePath” technology function:

- Direction = 0 : preparing path segment in forward direction
- Direction = 1 : preparing path segment in backward direction
- Direction = 2 : preparing path segment in both directions

To move the prepared segment, the desired travel direction also has to be indicated on the FB 488 “MC_MovePath” technology function. However, only the direction for which the segment was prepared through the “MC_PathSelect” technology function can be performed.

Note It is usually recommended to configure the entire palletizing process as a continuous interpolation point table (travel direction – forward), as long as the start and target positions of the process are already known when starting the application.

In this case, stops can also be overshot with the help of the FB 483 “MC_GroupContinue” technology function without stopping the kinematic axes.

4.2 Adding motion paths

With the help of the FB 488 “MC_MovePath” technology function an active travel motion can be added to another travel motion. However, a second instance of the FB 488 “MC_MovePath” technology function is necessary for this purpose.

Through the **BufferMode** input of the FB 488 “MC_MovePath” technology function, the following override modes of the current motion can be selected:

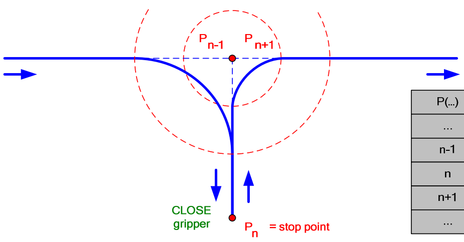
- **BufferMode=0: override of the active motion**
  The currently active motion is stopped and the reactivated motion is started.
- **BufferMode=1: adding the new motion**
  The currently active motion is finished. The kinematic axes of the path object are brought to a halt. Afterwards the newly activated motion is started.
- **BufferMode=2: blending of the new motion**
  At the end of the currently active motion the newly activated motion is added without stopping the kinematic axes, i.e. with continuous transmission of speed (=blending).

4.3 Gripper control through FB 488 “MC_MovePath”

Through the **Code** output of the FB 488 “MC_MovePath” technology function a gripper control can be realized whilst using individual bits of the integer variables. Block FC 531 “CodeToBit” can be used for this purpose. This is explained in more detail in chapter 2.3.3.

The code or the M function has to be already entered in the interpolation point table of the travel motion. The code of the M function, defined in the interpolation point table, is output as soon as the respective interpolation point is **approached** or overshot.

Figure 4-1 Gripper control



P(..)	X	Y	C.Dist.	Override	M fct.
...	...	...	...	...	...
n-1	PXn-1	PYn-1	15	100	0
n	PXn	PYn	-1	100	3
n+1	PXn+1	PYn+1	5	100	0
...	...	...	...	...	...

4.4 Use of zone monitoring

Through the FB 492 “MC_ZoneCheck” technology function, zone monitoring can be performed during the motion process.

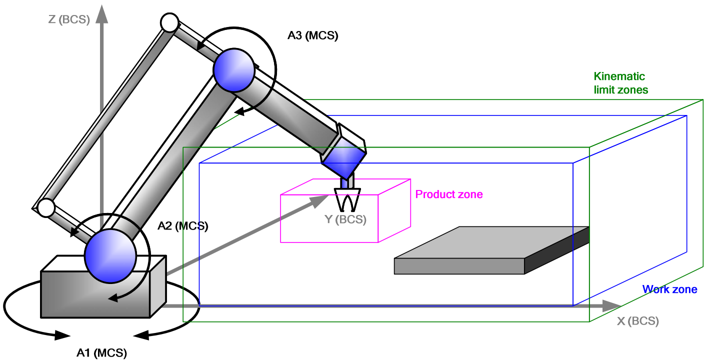
4.4.1 Monitoring zones

The following zones can be monitored with the FB 492 “MC_ZoneCheck” technology function:

Table 4-1 Zones of the zone monitoring

Zone	Max. number	Description
Product zone	5	The product zone is coupled to the kinematic end point (tool pickup point) and moves together with it. If no product zone is defined, the kinematic end point is used as “product zone”. Through the product zone a collision with the work zone can be monitored. Through the different product zones, for example, different grippers or a gripper with or without product can be monitored.
Work zone	1	The work zone defines the work area of the kinematic. If the product zone leaves the work zone a collision is reported.
Kinematic limit zone	1	The kinematic limit zone defines the maximum limit within which the product zone can be. Normally the work zone is within the kinematic limit zone. If the product zone enters the kinematic limit zone, a collision is reported.
Stop zone	5	Obstacles in the work zone are modeled through stop zones. If the product zone enters the stop zone, a collision is reported.
Report zone	5	The reaching of a defined position in the work area can be reported through report zones. If the product zone enters the report zone, a collision is reported.

Figure 4-2 Example regarding zones of the zone monitoring



The definition of individual zones is performed according to the following principle through the UDT 44 “ZoneDefinition” and UDT 43 “SingleZoneDef” data types:

- each zone is defined as a cuboid
- a zone start point with the coordinates in X, Y and Z direction is defined which relates to the basic coordinate system (BCS).
- the extension of the zone in X, Y and Z direction is defined.

UDT 44 “ZoneDefinition”

In this UDT various monitoring zones of the FB 492 “MC_ZoneCheck” technology function are listed.

Figure 4-3 UDT 44 “ZoneDefinition”

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	KinematicLimit	"SingleZoneDef"		Limitation zone of the kinematics.
+26.0	WorkingLimit	"SingleZoneDef"		Permissible working area.
+52.0	BlockLimit	ARRAY[1..5]		Blocked areas for handling device
*26.0		"SingleZoneDef"		
+182.0	ProductZone	ARRAY[1..5]		Product protection areas
*26.0		"SingleZoneDef"		
+312.0	WatchZone	ARRAY[1..5]		Watch zones
*26.0		"SingleZoneDef"		
=442.0		END_STRUCT		

UDT 43 “SingleZoneDef”

In this UDT the structure of the necessary data for the description of a monitoring zone is stored.

Figure 4-4 UDT 43 “SingleZoneDef”

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	PosX	REAL	0.000000e+000	X component start point
+4.0	PosY	REAL	0.000000e+000	Y component start point
+8.0	PosZ	REAL	0.000000e+000	Z component start point
+12.0	Width	REAL	0.000000e+000	X dimension
+16.0	Depth	REAL	0.000000e+000	Y dimension
+20.0	Height	REAL	0.000000e+000	Z dimension
+24.0	Active	BOOL	FALSE	Zone monitoring active
=26.0		END_STRUCT		

Through the **Active** parameter the respective zone can be enabled or disabled.

4.4.2 Monitoring of the zones

The monitoring of the defined zones of the zone monitoring can be switched on and off through the **Mode** parameter of the FB 492 “MC_ZoneCheck” technology function:

- **Mode=0: zone monitoring is switched on.**
  When a defined monitoring zone is violated, the path object is stopped with maximum deceleration of the path axes. If the work zone, the kinematic limit zone or the stop zone is violated, the technology function is terminated; otherwise the technology function to monitor the zones will stay active.
- **Mode=1: The zone monitoring is switched on (only report).**
  When a defined monitoring zone is violated, only a message will be output at the technology function block and the path object is no longer influenced.
- **Mode=2: zone monitoring is switched off.**
  A zone monitoring which is active through the FB 492 “MC_ZoneCheck” technology function is switched off.

Note When monitoring the individual zones, forward collision detection is carried out.

4.4.3 Zone monitoring as work area limit

To limit the work area of a kinematic, usually the limit switches of the axes are used. However, for some kinematics there is no relation between Cartesian coordinate system and machine axes. In this case, the zone monitoring can be used with the work zone or the kinematic limit zone to limit the work zone.

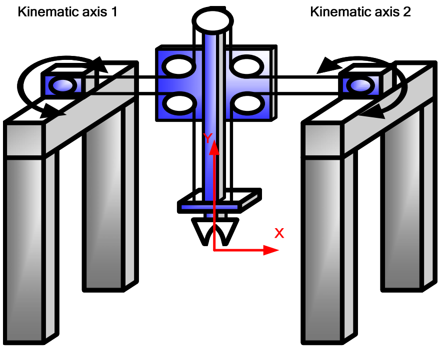
Limit of the travel zone for the "roll picker" kinematic

For the “roll picker” the motion of the reference point in the basic coordinate system depends roughly on the motion direction of the two kinematic axes:

- to move in **X direction** the two kinematic axes have to be moved in the **same direction (right+right/left+left)**.
- to move in Y direction the two kinematic axes have to be moved in the **opposite direction (right+left/left+right)**.

However, this means that the limit switches in X and Y direction in the basic coordinate system can not be assigned to the kinematic axes, i.e. the real axes of the path object in the machine coordinate system.

Figure 4-5 Limit of the travel zone for the “roll picker” kinematic



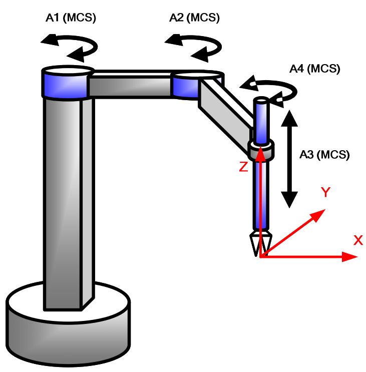
To limit the work zone, in this case it is recommended to use the FB 492 “MC_ZoneCheck” technology function in **Mode 0**. In this case, the technology function delivers the violation of the respective limit zone. However, the evaluation of the travel direction and the affected axis has to be triggered within the application.

Note When using the FB 492 “MC_ZoneCheck” technology function in **Mode 0** the technology function has to be restarted before starting a travel motion after a zone violation. Otherwise the zone monitoring is then no longer active.

Limit of the travel zone for the "SCARA" kinematic

For the “SCARA” kinematic the same point in the basic coordinate system can be reached through different positions of the kinematic axes. This is why it is not possible to limit the work zone through the limit switch of the kinematic axes.

Figure 4-6 Limit of the travel zone for the “SCARA” kinematic



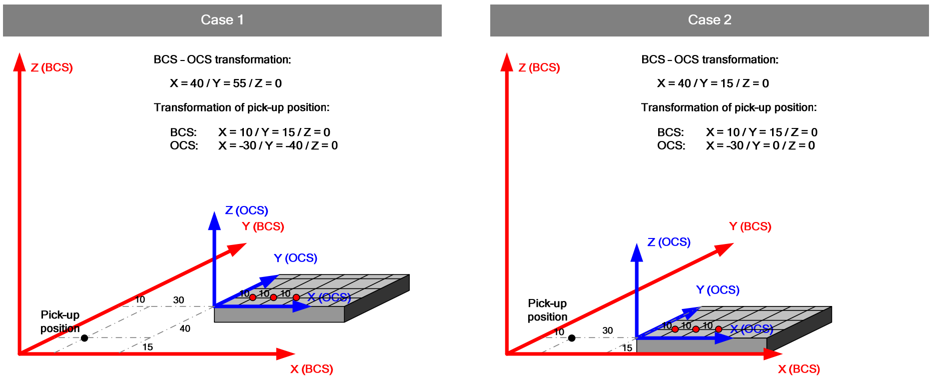
Here, it is also recommended to use the the FB 492 “MC_ZoneCheck” technology function in **Mode 0** to limit the work zone.

4.5 Transformation of the coordinate system

To be able to use uniform palletizing strategies on the pallete, it is recommended to describe the palletizing process on a fixed point in relation to the pallete and to translate the object coordinate system (OCS) each time with the help of the FB 480 “MC_SetCartesianTransform” technology function to this point.

The programming of the interpolation point table is carried out in the object coordinate system (OCS), whose origin is located on the respectively fixed point on the pallete.

Figure 4-7 Use of the object coordinate system (OCS)



Note The pick-up position of the products to be palletized is usually on a fixed point in the basic coordinate system (BCS). When translating the points of the interpolation point table through the transformation of the object coordinate system, this point (in the object coordinate system) has to be respecified every time (e.g. through retransformation) and has to be entered at the respective place in the interpolation point table.