

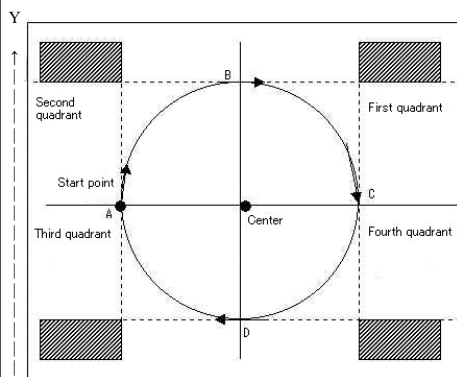
Circular Interpolation

Configurable number of axes Two to four axes in the same controller can be configured.

Description	Operates the CW (clockwise rotation)/CCW (counterclockwise rotation) circular interpolation between any two axes in the same controller. * Any other circular interpolation at the other axes are not available during the circular interpolation motion.
Comments	<ol style="list-style-type: none">1 Configure the moving velocity to the interpolation control axes, and configure the end and center points.2 The end point of the both axes is the same position as the start point, the circle is a true circle.3 The circular interpolation cannot be operated for only one axis or 3 or more axes.4 Regardless of the counter value, the current position is the start point (coordinate 0, 0) in the circular interpolation.5 The motion immediately stops by the stop signal (EL signal) input of moving direction.6 The motion immediately stops by the alarm signal (ALM signal) input.7 If any one axis in the interpolation axes stops in error, all interpolation axes stops as well. Confirm the error stop in "MTR_FINISH_STATUS" of the MtnGetStatus function. (except 0) For the axis which is the error stop factor, the factor bit is "1" and for the other interpolation axes, bit 15 is "1".8 The acceleration and deceleration motion is not available in the circular interpolation motion.9 Even if the origin signal (ORG signal) is input, it is disregarded.10 The synthesis rate constant control is available.11 Specify the "output pulse cycle completion" to the pulse output completion timing.12 If the ending coordinate is not on the circumference, the circular interpolation motion is completed when either axis reaches the end point in the end point quadrant and does not end at the specified end-point coordinate. If the motion is resumed after such circular interpolation motion is completed, the point moves from the point where the last circular interpolation is completed to the end-point coordinate. (End-point drawing motion)

Configuration Example Configure the moving velocity, end point, and center point as the table below to draw arc to each point of A to D in the figure below.
When the end point of the circular interpolation is configured in the shaded area in the figure below, the motion does not stop. (permanent rotation)

	Configuration 1		Configuration 2		Configuration 3		Configuration 4	
	X-axis	Y-axis	X-axis	Y-axis	X-axis	Y-axis	X-axis	Y-axis
Moving direction	CW direction		CW direction		CW direction		CW direction	
Synthesis rate constant control	OFF		OFF		OFF		OFF	
Moving velocity	100 pps		100 pps		100 pps		100 pps	
End point	0	0	100	100	200	0	100	-100
Center point	100	0	100	0	100	0	100	0
Result	True circle (A→B→C →D→A)		90 degrees circle (A→B)		180 degrees circle (A→B→C)		270 degrees circle (A→B→C→D)	



Center: (100, 0)
A: (0, 0) B: (100, 100)
C: (200, 0) D: (100, -100)

The motion operate circular interpolation within the same controller.

Configured axis	X-axis	Y-axis
Motion	Circular interpolation	Circular interpolation
Moving velocity	500	
Frequency rate	299	299
Moving direction	CW (clockwise rotation)	CW (clockwise rotation)
Synthesis rate constant control	No	No
central coordinate	100	0

Ending coordinate	0	0
-------------------	---	---

The motion sets the current position at start-up as the start point (0, 0) and draws an relative arc.
Set the ending coordinate to (0, 0) to draw a true circle.
The acceleration and deceleration motion is not available in the circular interpolation motion. Start at a constant velocity.

2 Multi-Function DLL

```
HANDLE hDeviceHandle;  
int nRet;  
MTNARC Arc;  
  
hDeviceHandle = MtnOpen("FBIMTN1", MTR_FLAG_NORMAL );  
;  
  
Arc.wAxis = 0x03;  
Arc.wClock = 299;  
Arc.wMode = MTR_ARC_CW;  
Arc.fSpeed = 100;  
Arc.lCenterX = 100;  
Arc.lCenterY = 0;  
Arc.lEndX = 0;  
Arc.lEndY = 0;  
  
nRet = MtnSetMotionArc( hDeviceHandle, MTR_ARC_NORMAL, &Arc );  
if(nRet != MTR_ERROR_SUCCESS) return -1;  
  
nRet = MtnStartMotion( hDeviceHandle, 0x03, MTR_CONST, MTR_ARC );  
  
* Operating the circular interpolation which passes through the center point (100, 0) and end point (0, 0) on the X and Y axes of hDeviceHandle to draw a true circle.
```