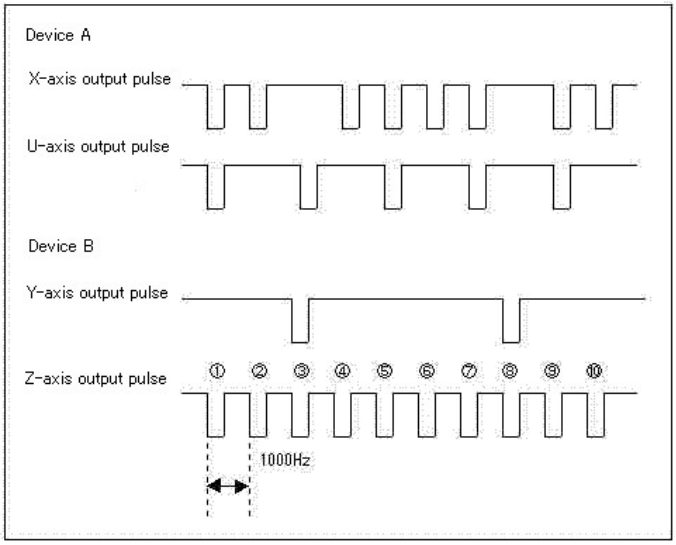
Continuous Linear Interpolation 2

Configurable number of axes	Two or more axes among multiple controllers can be configured.

Description		Operates the linear interpolation motion by specifying the moving quantity for multiple axes among multiple controllers. **The motion operates continuously in the ratio of configured moving quantity.**
		The motion is used to operate the linear interpolation motion of more than 5 axes using two or more motion controller boards (up to 8 boards). **To start this motion, use the synchronous start signal. (Refer to"Synchronous start.)**
備考	1	The maximum number of the controllable axes is 32 axes.
	2	The axis whose moving velocity is largest is the master axis and the others are slave axes. The configurations of the slave axes are the same as the master axis except the moving quantity. The moving direction is depend on the code of the moving quantity.
	3	When acceleration and deceleration is performed, the velocity is decelerated to the start-up velocity or the deceleration stops by inputting the deceleration signal (SD signal) of moving direction.
	4	The deceleration stops or immediately stops by the stop signal (EL signal) input of moving direction.
	5	The deceleration stops or immediately stops by the alarm signal (ALM signal) input.
	6	If any one axis in the interpolation axes stops in error, all interpolation axes stops as well. The interrupt at the error stop is generated for the axis which stops in the error.
	7	Configure "Constant velocity start" for the start-up motion.
	8	Even if the origin signal (ORG signal) is input, it is disregarded.
	9	**Specify the "output pulse cycle completion " to the pulse output completion timing.**
設定例		The continuous linear interpolation 2 operates in the following configuration. It outputs pulses consecutively in the following ratio; X-axis : U-axis : Y-axis : Z-axis = 8 : 5 : 2 : 10.

	Device A		Device B	
	X-axis	U-axis	Y-axis	Z-axis
Mode	continuous linear interpolation 2	continuous linear interpolation 2	continuous linear interpolation 2	continuous linear interpolation 2
Moving pulse count	800	500	200	1000
Maximum moving pulse count	1000	1000	1000	1000
Moving velocity	1000 pps	1000 pps	1000 pps	1000 pps
Startup motion	Constant velocity start	Constant velocity start	Constant velocity start	Constant velocity start
Master/slave axis	slave axis	slave axis	slave axis	master axis

The pulses are output in the following timing.



2 **Multi-Function DLL**

```
HANDLE hDeviceHandle1, hDeviceHandle2;
int nRet;
WORD wSync[4];
MTNLINE Line;

hDeviceHandle1 = MtnOpen("FBIMTN1", MTR_FLAG_NORMAL );
hDeviceHandle2 = MtnOpen("FBIMTN2", MTR_FLAG_NORMAL );
:

wSync[0] = MTR_SYNC_EXT;
wSync[1] = MTR_SYNC_EXT;
wSync[2] = MTR_SYNC_EXT;
wSync[3] = MTR_SYNC_EXT;
nRet = MtnSetSync( hDeviceHandle1, 0x09, MTR_START_MODE, wSync );
nRet = MtnSetSync( hDeviceHandle2, 0x06, MTR_START_MODE, wSync );

Line.wAxis = 0x09;
Line.wClock = 299;
Line.wMode = MTR_LINE;
Line.wAccMode = MTR_ACC_NORMAL;
Line.fLowSpeed = 100;
Line.fSpeed = 1000;
Line.dwAcc = 100;
Line.dwDec = 100;
Line.fSAccSpeed = 0;
Line.fSDecSpeed = 0;
Line.lStep[0] = 800;
Line.lStep[3] = 500;
nRet = MtnSetSyncLine( hDeviceHandle1, 1000, &Line );
if(nRet != MTR_ERROR_SUCCESS) return -1;

nRet = MtnStartMotion( hDeviceHandle1, 0x09, MTR_CONST, MTR_SYNC_LINE );

Line.wAxis = 0x06;
Line.wClock = 299;
Line.wMode = MTR_LINE;
Line.wAccMode = MTR_ACC_NORMAL;
Line.fLowSpeed = 100;
Line.fSpeed = 1000;
Line.dwAcc = 100;
Line.dwDec = 100;
Line.fSAccSpeed = 0;
Line.fSDecSpeed = 0;
Line.lStep[1] = 200;
Line.lStep[2] = 1000;
nRet = MtnSetSyncLine( hDeviceHandle2, 1000, &Line );
if(nRet != MTR_ERROR_SUCCESS) return -1;
```

```
nRet = MtnStartMotion( hDeviceHandle2, 0x06, MTR_CONST, MTR_SYNC_LINE );

/* Synchronous start signal output */
nRet = MtnOutputSync(hDeviceHandle1, MTR_EXT_START );
```

* Operating continuous linear interpolation 2 motion for X and U axes of hDeviceHandle1 and Y and Z axes of hDeviceHandle2.

[Top]