PISO-PS400 Function Reference

(Version 2.3)

ICP DAS CO., LTD.

泓格科技股份有限公司

Warranty

All products manufactured by ICPDAS Inc. are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

Warning

ICPDAS Inc. assumes no liability for damages consequent to the use of this product. ICPDAS Inc. reserves the right to change this manual at any time without notice. The information furnished by ICPDAS Inc. is believed to be accurate and reliable. However, no responsibility is assumed by ICPDAS Inc. for its use, or for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Trademark

The names used for identification only maybe registered trademarks of their respective companies.

License

The user can use, modify and backup this software on a single machine. The user may not reproduce, transfer or distribute this software, or any copy, in whole or in part.

Content

1 Forward

1.1 Introduction

- **This documentation provides the detailed information of PISO-PS400 functions, including the declarartion and parameters of PISO-PS400 functions.**
- **There are seven chapters in this documentaion.**
  - **Chapter 1 - Introduction**
  - **Chapter 2 - Basic function Setting**
  - **Chapter 3 - Status setting and read back**
  - **Chapter 4 - Interrup setting and handling**
  - **Chapter 5 - Operation of FR-Net**
  - **Chapter 6 - Auto Homing**
  - **Chapter 7 - Axis Control**

1.2 Function Structure explanation

● **Function name(argument 1*, argument 2,* …….)**
Function : Basic Function Explanation。

Argument: Defintion of associated argument in this function。

Return: Return Value。

Smaple: Simple Demo Program。(Sample Program in this manual is coded with VC++)

Remark: Remakr and Explanation。

1.3 API List

Function Name	Brief Description	Section
Basic Setting Function		
PS400_Card_Init	Initialize all PISO-PS400 cards.	2.2
PS400_Card_Close	Before Re-start PS400_Card_Init, the all PISO-PS400 cards must be released with PS400_Card_Close.	2.2
PS400_Total_Card	Number of successfully registerd cards	2.2
PS400_Get_CardNo	Get Card ID set with onboard DIP-Switch..	2.2
PS400_Reset_Card	Reset card	2.3
PS400_Set_PulseMode	Set Pulse output mode	2.4
PS400_Set_MaxSpeed	Set max. speed limit in pps.	2.5
PS400_Set_Limit	Set logic of limit switch	2.6
PS400_Set_Home	Set logic of ORG	2.7
PS400_Set_SoftLimit	Set software limit or compare register	2.8
PS400_Disable_SoftLimit	Disable software limit	2.8
PS400_Set_EncoderMode	Set Encoder Mode	2.9
PS400_Set_Servo_ON	Switch Servo ON/OFF	2.10
PS400_Set_Alm	Set Alarm pin	2.11
PS400_Set_Inp	Set INP pin	2.12
PS400_Set_Filter	Set digital filter	2.13
PS400_Set_Vring	Set ring counter	2.14
PS400_Set_AvTri	Enable triangle velocity profile	2.15
PS400_Set_Compare	Set compare counter	2.16
PS400_Set_ManualPulsar	Set manual pulse generator	2.17
PS400_Set_ManualPulsarEx	Extension function of PS400_Set_ManualPulsar( )	2.17
PS400_Set_Input	Set IN3	2.18
Status Function		
PS400_Set_Command	Set current command position	3.1
PS400_Get_Command	Get current command position	3.1
PS400_Set_Position	Set current ENCODER position	3.2
PS400_Get_Position	Get current ENCODER position	3.2
PS400_Get_Speed	Get current speed	3.3
PS400_Get_Acceleration	Get current acceleration	3.4
PS400_Get_DI_Status	Get DI status	3.5

PS400_Get_Error_status	Get error status	3.6
PS400_Get_Error_Code	Get error code	3.6
Interrupt Function		
PS400_Enable_INT	Enable interrupt service routine	4.1
PS400_Disable_INT	Disable interrupt	4.1
PS400_Set_INT_Factor	Set interrupt factor	4.3
PS400_Get_INT_Status	Get interrupt status	4.4
PS400_Attach_INT	Attach the occurrence of Interrupt with EVENT objects	4.5
PS400_Dettach_INT	Release the EVENT objects revelant to Interrupt	4.5
I/O of FRnet		
PS400_Read_FRnet	Read DI input from FRnet	5.1
PS400_Write_FRnet	Write DO output to FRnet	5.1
PS400_Set_FRnetDO	Turn ON/OFF the DO functionality	5.2
PS400_Read_FRnetStatus	Check the available DI modules	5.3
PS400_Reset_FRnet	Reset the DO functionality	5.4
Auto Homing		
PS400_Set_HomeSpeed	Set Homing Speed	6.1
PS400_Set_HomeLimit	Set Limit as ORG	6.2
PS400_Set_HomeMode	Set Homing mode	6.3
PS400_Home _Start	Start Homing	6.4
PS400_Home_Done	Homing is finished	6.5
Axis Control		
PS400_Const_Move	Single/Multiple Axes constant Speed	7.1
PS400_T_Move	Single Axis T-curve move	7.2
PS400_T_As_Move	Single Axis T-curve move	7.2
PS400_S_Move	Single Axis S-curve move	7.3
PS400_S_As_Move	Single Axis S-curve move	7.3
PS400_Conti_Move	Single Axis S-curve continuous move	7.4
PS400_Line2_Move	2-axis line interpolation	7.5
PS400_Line3_Move	3-axis line interpolation	7.5
PS400_Arc2_Move	2-axis CW arc interpolation	7.6
PS400_Set_SyncMotion	Set Sync. Factor	7.7
PS400_Set_Latch	Set Position Latch	7.7
PS400_Get_Latch	Get Position Latch value	7.7
PS400_Sync_Preset	Set 2-axis up sync. condition	7.7
PS400_Preset_Data	Get preset value after sync.	7.7

PS400_Rectamgle	**2-axis rectangle continuous interpolation**	**7.8.1**
PS400_Set_Line2	**Set 2-axis line interpolation**	**7.8.2**
PS400_Line2_Start	**Start 2-axis line interpolation**	**7.8.2**
PS400_Set_Line3	**Set 3-axis line interpolation**	**7.8.3**
PS400_Line3_Start	**Start 3-axis line interpolation**	**7.8.3**
PS400_Set_Mix2	**Set 2-axis mixed interpolation**	**7.8.4**
PS400_Multi_Intp_Move	**Multi-point continuous interpolation**	**7.8.5**
PS400_Helix3_Move	**Helical interpolation**	**7.8.6**
PS400_Set_Ratio2	**Set command ratio**	**7.8.7**
PS400_Ratio2_Start	**Start ratio motion**	**7.8.7**
PS400_Drv_Hold	**Pause Axis move**	**7.9.1**
PS400_Drv_Start	**Start Axis move**	**7.9.2**
PS400_Set_SdStop	**Set slow-down point**	**7.9.3**
PS400_Set_EmgStop	**Set Emg-stop point**	**7.9.3**
PS400_Clear_Stop_Status	**Clear Stop status**	**7.9.4**
PS400_Motion_Done	**Motion is completed**	**7.9.5**
PS400_CmpTrig_Config	**configure the Compare/Trigger functionality**	**7.9.6**
PS400_CmpTrig_Reset	**reset the Compare/Trigger configuration that is set with PS400_CmpTrig_Config**	**7.9.7**

2 Basic Setting Function

2.1 Code numbers for axes

Card of each PISO-PS400 motion card is defined by the on-board swith SW1.Up to16 cards can be supported in the same system. Each axis is m apped in the following talbe Tab 2-1. The axis is represented by bit in the axis parameter which is word type variable.

bit3	bit2	bit1	bit0
U	Z	Y	X

Axsi mapping table :

Table 2-1

Axis	X	Y	Z	U	XY	XZ	XU	YZ
Code	0x1	0x2	0x4	0x8	0x3	0x5	0x9	0x6
Var.	AXIS_X	AXIS_Y	AXIS_Z	AXIS_U	AXIS_XY	AXIS_XZ	AXIS_XU	AXIS_YZ
Axis	YU	ZU	XYZ	XYU	XZU	YZU	XYZU	
Code	0xa	0xc	0x7	0xb	0xd	0xe	0xf	
Var.	AXIS_YU	AXIS_ZU	AXIS_XYZ	AXIS_XYU	AXIS_XZU	AXIS_YZU	AXIS_XYZU	

2.2 Card initialization

- short PS400_Card_Init(void)

Fun.: User can use this function to initialize the card on the PCI bus.After successful initialization the card can activitate its functions

Para.: None

Return: SUCCESS_NO_ERROR
NO_CARD_FOUND：no available PISO-PS400 is found on system

Sample :

```
# define MaxCards 16
short CardID[MaxCards];
PS400_Card_Init ();
short card_num = PS400_Total_Card();
for (BYTE i = 0; i < card_num; i++)
{
    CardID[i] = PS400_Get_CardNo(i);
}
```

```
short CardNo = CardID[0];
//Initialize all cards, and assign the CardNo with the 1st CardID
```

- **short PS400_Card_Close(void)**

Func.: Release the resource of the card allocated by the Windows OS. User should apply this function to terminate or exit its application program.

Para.: None

Return: SUCCESS_NO_ERROR

Sample:

```
# define MaxCards 16
short CardID[MaxCards];
// Initialize all cards, and assign the CardNo
PS400_Card_Init();
short card_num = PS400_Total_Card();
for (BYTE i = 0; i < card_num; i++)
{
    CardID[i] = PS400_Get_CardNo(i);
}
short CardNo = CardID[0];

//PS400_Card_Init() can be re-initialize only after calling PS400_Card_Close()
PS400_Card_Close();

//re-initialize all cards again
PS400_Card_Init();
```

- **short PS400_Total_Card(void)**

Func.: Get the total no. of successfully initialized cards.

Para.: None

Return: Number of cards

Sample:

```
# define MaxCards 16
short CardID[MaxCards];
short card_num = PS400_Total_Card();
for (BYTE i = 0; i < card_num; i++)
{
    CardID[i] = PS400_GET_CardNo(i);
}
//find all available cards and assign the CardNo
```

- **short PS400_Get_CardNo(BYTE *index*)**

Func.: Get the Card ID based on the order that the cards were found

Para.: ***index***: card ID on the card
Return.: card ID.

Sample:
```
//Initialze all cards and get card ID。
# define MaxCards 16
short CardID[MaxCards];
short card_num;
PS400_Card_Init();
card_num = PS400_Total_Card();
for (BYTE i = 0; i < card_num; i++)
{
    CardID[i] = PS400_Get_CardNo(i);
}
BYTE CardNo = (BYTE)CardID[0];          //Assign the CardID with index=0
PS400_Set_Command(CardNo, AXIS_XYZU, 0);
PS400_Set_Position(CardNo, AXIS_XYZU, 0);
```

2.3 Card Reset

- **short PS400_Reset_Card(BYTE *cardNo*)**

Func.: Use this function to reset the card without exiting the application program。

Para.: ***cardNo***: assigned card ID

Return: SUCCESS_NO_ERROR
CARD_NUMBER_ERROR: the *cardNo* is invalid
INVALID_DEVICE_ERROR: no active device is related to the cardNo

Sample:
```
PS400_Reset_Card(1);
//Reset card 1 。
```

2.4 Set Pulse output mode

- **short PS400_Set_PulseMode(BYTE *cardNo*, WORD *axis*, BYTE *Mode*)**

Func.: **Set Pulse mode，include CW/CCW , PULSE/DIR，and rising/falling edge。**

Para.: ***cardNo*:** **card ID**
***axis*:** **axis no(Table3-1)**
***Mode*:** **mode(Table3-2)**

Pulse output mode (Table 3-2)

Type	Mode	Dir.	Output Pulse	
			nPP	nPM
CW / CCW	0		CW(rising)	CCW(rising)
	1		CW(falling)	CCW(falling)
PULSE / DIR	2	+	PULSE(rising)	DIR(LOW)
	3		PULSE(falling)	DIR(LOW)
	4	−	PULSE(rising)	DIR(LOW)
	5		PULSE(falling)	DIR(LOW)

Return: **SUCCESS_NO_ERROR**
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid
MOTION_STATUS_ERROR : some error occurred in internal Motion-ASIC,
please call PS400_GET_ERROR_CODE() for detailed information

Sample: **PS400_Set_PulseMode(1, AXIS_XYZ, 2);**
// Set card 1 axis X/Y/Z as mode 2。

2.5 Set Maximum Speed range

- **short PS400_Set_MaxSpeed(BYTE *cardNo*, WORD *axis*, DWORD Data)**

Func.: **Set Maximum pulse output rate**

Para.: ***cardNo*:** **card ID**
***axis*:** **Axis no.**
Data **Set Max. Speed from 8,000 to 4,000,000**

Return: **SUCCESS_NO_ERROR**
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis* parameter

CARD_NUMBER_ERROR: the *cardNo* is invalid

Sample： **PS400_Set_MaxSpeed(1, AXIS_XYZU, 200000)**
// Set card 1 4-Axes max. speed as 200K PPS。

2.6 Set Limit Swith (EL) logic and action

- **short PS400_Set_Limit(BYTE *cardNo*, WORD *axis*, BYTE Logic, BYTE StopMode)**

Func.: **Set the logic and action of limit switches "±EL ", logic High/Low , slow-down stop / immediate-stop。**

Para.: ***cardNo*: card ID**
***axis*: Axis no.(Table3-1)**
***Logic*: 0=Low active, 1=High active**
***StopMode*： 0=immediate stop,1=slow-down stop**

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid
LOGIC_SETTING_ERROR: the *Logic* parameter is invalid

Sample: **PS400_Set_Limit(1, AXIS_XYZU, 0, 0);**
// Set card 1 X Y Z U low-active and immediate stop

2.7 Set ORG and NORG logic

- **short PS400_Set_Home(BYTE *cardNo*, WORD *axis*, BYTE *HLogic,* BYTE *NHLogic,* BYTE *ZLlogic*)**

Func.: **Set ORG/NORG logic。**

Para.: ***cardNo*: card ID**
***axis*: axis no.(Table3-1)**
***HLogic*: ORG: 0=low active, 1=high active**
***NLlogic*: NORG: 0=low active, 1=high active**
***ZLogic* : Z-Phae: 0=low active, 1=high active**

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid a*xis* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid
LOGIC_SETTING_ERROR: the *Logic* parameter is invalid

Sample: **PS400_Set_Home(1, AXIS_XYZU, 1, 1);**
//Set card 1 X Y Z U ORG and NORG as high active。

2.8 Set software limit (±SEL）

- **short PS400_Set_SoftLimit(BYTE *cardNo*, WORD *axis*, long PLimit, long NLimit, BYTE *Type*)**

Func.: Set software limit SEL or comparator (C±) function。

Para.:

	cardNo:	card ID
	axis:	axis no.(Table3-1)
	Plimit:	Positive software limit range(-2,147,483,648 ~ +2,147,483,647)
	NLimit:	Negative software limit range(-2,147,483,648 ~ +2,147,483,647)
	Type:	comparator source: 0=Command,1=Encoder

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid
LOGIC_SETTING_ERROR: the *Logic* parameter is invalid
MODE_SETTING_ERROR: the *Type* parameter is invalid

Sample:
```
PS400_Set_SoftLimit(1, AXIS_XYZU, 20000, -3000, 0);
// Set card 1 X Y Z U axis PSEL=20000，NSEL=-3000 and comparator source as command position。
```

Notice:

When SEL is enabled, the CMP INT factor MUST be disabled. Please use PS400_Set_INT_Factor() to reset the condition to 0.

- **short PS400_Disable_SoftLimit(BYTE *cardNo*, WORD *axis*)**

Func.: Disable software limit (±SEL）function。

Para.:

	cardNo:	card ID
	axis:	axis no.(Table3-1)

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid

Sample:
```
PS400_Disable_SoftwareLimit(1, AXIS_XYZU);
//Disable card1  X Y Z U axis software limit。
```

2.9 Set Encoder input mode

- **short PS400_Set_EncoderMode(BYTE *cardNo*, WORD *axis*, BYTE *Mode*)**

Func.: **Set Eencoder input mode（1AB/2AB/4AB or CW/CCW）。**

Para.: ***cardNo*:** **card ID**
***axis*:** **axis no.(Table3-1)**
***Mode*:** **encoder mode : 0 = 1AB,**
1 = 2AB,
2 = 4AB,
3 = CW/CCW

Return: **SUCCESS_NO_ERROR**
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid
MODE_SETTING_ERROR: the *Mode* parameter is invalid

Sample: **PS400_Set_EncoderMode(1, AXIS_XYZU, 0, 0);**
//set card1 X Y Z U axiswith AB phase mode input。

2.10 Set Servo_ON

- **short PS400_Set_Servo_ON(BYTE *cardNo*, WORD *axis, BYTE Mode*)**

Func.: **Set DO to switch Servo ON/OFF.**

Para.: ***cardNo*:** **card ID**
***axis*:** **axis(Table3-1)**
Mode*：** ***0=ON, 1=Off

Return: **SUCCESS_NO_ERROR**
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid
MODE_SETTING_ERROR: the *Mode* parameter is invalid

Sample: **PS400_Set_Servo_ON(1, AXIS_XYZU);**
//Set card1 X Y Z U axis Servo_ON。

2.11 Set ALM logic and action

- **short PS400_Set_Alm(BYTE *cardNo*, WORD *axis*, BYTE *Mode*, BYTE *Logic*)**

Func.: **Set ALM logic and action for servo dirver/motor。**

Para.:
	***cardNo*:**	**card ID**
	***axis*:**	**axis no(Table3-1)**
	***Mode*:**	**mode: 0=disable,1=enable**
	***Logic*:**	**0=low active, 1=high active**

Return: **SUCCESS_NO_ERROR**
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid
LOGIC_SETTING_ERROR: the *Logic* parameter is invalid
MODE_SETTING_ERROR: the *Mode* parameter is invalid

Sample:
```
PS400_Set_Alm(1, AXIS_XY, 1, 0);
//Set card 1XY axis，ALARM) enable with low active。
```

2.12 Set INP logic and action

- **short PS400_Set_Inp(BYTE *cardNo*, WORD *axis*, BYTE *Mode*, BYTE *Logic*)**

Func.: **Set INP Input logic and action for servo driver/motor。**

Para.:
	***cardNo*:**	**card ID**
	***axis*:**	**axis no(Table3-1)**
	***Mode*:**	**mode: 0=disable,1=enable**
	***Logic*:**	**0=low active, 1=high active**

Return: **SUCCESS_NO_ERROR**
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid
LOGIC_SETTING_ERROR: the *Logic* parameter is invalid
MODE_SETTING_ERROR: the *Mode* parameter is invalid

Sample:
```
PS400_Set_Inp(1, AXIS_X, 1, 0);
//Set card 1 X axis，INP enable with low active。
```

Notice: **Please check the wiring。**

2.13 Set filter for DI input

short PS400_Set_Filter(BYTE *cardNo*, WORD *axis*, WORD *Mode*, WORD *TC*)

Func.: Set filter for DI input 。

Para.: ***cardNo*:** card ID

***axis*:** axis no.(Table3-1)

***Mode*:** Filter index code (0~31）in the following table:

code	Function
1	EMG,PEL,NEL,ORG,NORG
2	Encoder Z phae
4	INP,ALM
8	nEXPP, nEXPM, EXPLSN
16	(IN3)

***TC*:** Filter time parameter (0~7) in the following table:

code	Noise width	delay
0	1.75μSEC	2μSEC
1	224μSEC	256μSEC
2	448μSEC	512μSEC
3	896μSEC	1.024mSEC
4	1.792mSEC	2.048mSEC
5	3.584mSEC	4.096mSEC
6	7.168mSEC	8.192mSEC
7	14.336mSEC	16.384mSEC

Return: SUCCESS_NO_ERROR

INVALID_DEVICE_ERROR: no active device is related to the cardNo

INVALID_AXIS_ERROR: invalid *axis* parameter

CARD_NUMBER_ERROR: the *cardNo* is invalid

MODE_SETTING_ERROR: the *Mode* parameter is invalid

Sample: PS400_Set_Filter(1, AXIS_XYZU, 21, 3);

//Set card1 X Y Z U axis，(21=1+4+16) 1→EMG,PEL,NEL,NORG,ORG，4→INP,ALM，16→IN3 enable with filter time = 1.024mSEC。

2.14 Set counter as Variable-Ring

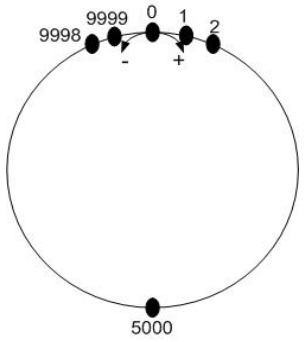
- **short PS400_Set_Vring(BYTE *cardNo*, WORD *axis*, BYTE Mode, DWORD *Value*)**

Func.: **Set ring counter (see attached fig.)。**

Para.: ***cardNo*:** **card ID**
***axis*:** **axis no.(Table3-1)**
Mode： **0:(Disable), 1:(Enable)**
***Alue*:** **ring counter range：-2,147,483,648 ~ +2,147,483,647**

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid
MODE_SETTING_ERROR: the *Mode* parameter is invalid

Sample: PS400_Set_ Vring(1, AXIS_X, 10000);
//Set card 1 X axis，ring counter with 10000 Pulse/Revolution。



Ring counter=9999

Notice:

This function will set both Pulse/Encoder couters as Ring-Counter.

This function is mutual-exclusive to PS400_Set_SoftLimit ( ).

2.15 Set Triangle velocity profile

- **short PS400_Set_AvTri(BYTE *cardNo*, WORD *axis, BYTE Mode*)**

Func.: **Enable or disable triangle velocity profile action。**

Para.: ***cardNo*:** **card ID**
***axis*:** **axis no (Table3-1)**
Mode： ***0=Disable, 1=Enable***

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis* parameter

CARD_NUMBER_ERROR: the *cardNo* is invalid
MODE_SETTING_ERROR: the *Mode* parameter is invalid

Sample: **PS400_ Set_AvTri(1, AXIS_X);**
//Set card1 X axis，enable triangle velocity profile protection。

2.16 Set comparator

- **short PS400_Set_Compare(BYTE *cardNo,* WORD *axis,* BYTE *CmpSource,*BYTE *CmpMode,* long *CmpValue*)**

Func.: **Set comparator parameter , include Compare source、Mode and Value。**

Para.: ***cardNo*: card ID**
***axis*: axis no**
***CmpSource* ： Compare source：0=Command, 1=Encoder**
***CmpMode* ： Compare mode：0=increase, 1=decrease**
***CmpValue* ： Comapre Value：-2,147,483,648 ~ +2,147,483,648**

Return: **SUCCESS_NO_ERROR**
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid
MODE_SETTING_ERROR: the *CmpMode* parameter is invalid

Sample: **PS400_ Set_Compare(1, AXIS_X, 0, 0, -50000);**
//Set card1 X axis，comparator with increase mode and value =50000。
PS400_Set_Compare(1, AXIS_Y, 1, 1, 50000);
//Set card 1 X axis，comparator with decreas mode and value =-50000。

2.17 Set Manual Pulser

- **short PS400_Set_ManualPulsar(BYTE *cardNo*, WORD *axis*, BYTE *Mode*, DWORD *Command*)**

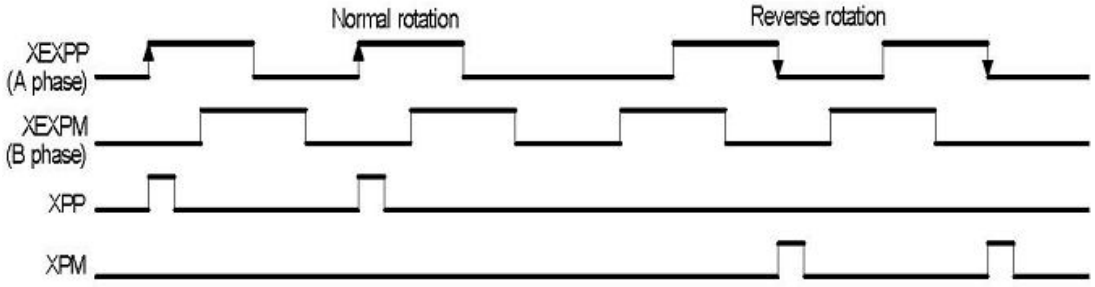
Func.: **Set the operation of Manual Pulse Generator.**

Para.: ***cardNo*: card ID**
***axis*: axis no.(1 or 2 or 4 or 8)**
***Mode*: 0 : disable**
1 : AB phase input ，specified pulse output
2 : CW/CCW ，specified pulse output
3 : CW/CCW continuous mode。

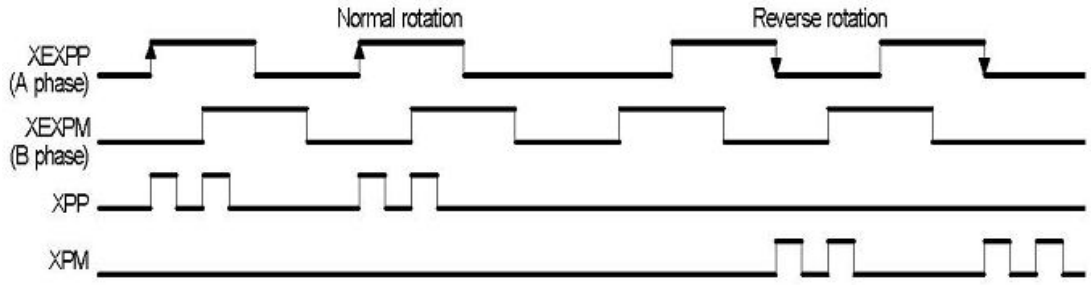
***Command*:** **Specified (mode 1/2) or continuous(mode 3) mode**

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid
MODE_SETTING_ERROR: the *Mode* parameter is invalid

Sample: PS400_Set_ManualPulsar(1, AXIS_X, 1, 1);
//Set card1 X axis，MPG 1 Pulse。



PS400_Set_ManualPulsar(1, AXIS_X, 1, 2);
//Set card1 X axis，MPG 2 Pusle。



2.18 Set IN3 action

- **short PS400_Set_Input(BYTE *cardNo*, WORD *axis*, BYTE *Mode*, BYTE *Logic*)**

Func.: Set IN3 input enable/disable function。

Para.: ***cardNo*:** **card ID**
***axis*:** **axis no. (1 or 2 or 4 or 8)**
***Mode*:** **0 : Disable，1 : Enable**
Logic **0=low active, 1=high active**

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo

INVALID_AXIS_ERROR: invalid *axis* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid
LOGIC_SETTING_ERROR: the *Logic* parameter is invalid
MODE_SETTING_ERROR: the *Mode* parameter is invalid

Sample: **PS400_Set_Input(1, AXIS_X, 1, 1);**
//Set card 1 X axis，IN3 Enable with high active logic。

3 Status Setting and readback

3.1 Set and Get Command Pulse

- **short PS400_Set_Command(BYTE *cardNo*, WORD *axis*, long *Command*)**

Func.: Set command pulse。

Para.: *cardNo*: card ID
axis: axis no.(Table3-1)
Command: range：-2,147,483,648 ~ +2,147,483,647

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid

Sample:
```
PS400_Set_Command(1, AXIS_XYZU, 0);
//Set card1 X Y Z U axis，current position as 0。
```

- **short PS400_Get_Command(BYTE *cardNo*, WORD *axis,* long* *pCommand*)**

Func.: Read command pulse。

Para.: *cardNo*: card ID
axis: axis no(1 or 2 or 4 or 8)
pCommand: point to the memory of the returned command.

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid

Sample:
```
long X_LP;
PS400_Get_Command(1, AXIS_X, &X_LP);
//Get card1 X axis，current command position。
```

3.2 Set and Get Encoder Pulse

- **short PS400_Set_Position(BYTE *cardNo*, WORD *axis*, long *Position*)**

Func.: Set Encoder Pusle。

Para.: *cardNo*: card ID

axis: axis no(Table3-1)
Position: Set Encoder pulse（range -2,147,483,648 ~ +2,147,483,647）

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid

Sample:
```
PS400_Set_Position(1, AXIS_XYZU, 0);
//Set card1 X Y Z U axis，current ENCODER as 0。
```

- **short PS400_Get_Position(BYTE *cardNo*, WORD *axis,* long* *pPosition*)**

Func.: Get Encoder feedback pulse。

Para.: *cardNo*: card ID
axis: axis no(1 or 2 or 4 or 8)
pPosition: point to the memory of the returned encoder position.

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid

Sample:
```
long X_EP;
PS400_Get_Position(1, AXIS_X, &X_EP);
//Get card1 X axis，current ENCODER feedback pulse。
```

3.3 Get current speed

- **short PS400_Get_Speed(BYTE *cardNo*, WORD *axis,* long* *pSpeed*)**

Func.: Set current speed。

Para.: *cardNo*: card ID
axis: axis no (1 or 2 or 4 or 8)
pSpeed: point to the memory of the returned speed.

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid

Sample:
```
long X_Speed;
PS400_Get_Speed(1, AXIS_X, &X_Speed);
//Get card1 X axis，current speed。
```

3.4 Get current acceleration

- **short PS400_Get_Acceleration(BYTE *cardNo*, WORD *axis,* long* *pAcc*)**

Func.: Get current acceleration。

Para.:
- ***cardNo***: card ID
- ***axis***: axis no(1 or 2 or 4 or 8)
- ***pAcc***: point to the memory of the returned acceleration.

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid

Sample:
```
long X_ACC;
PS400_Get_Acceleration (1, AXIS_X, &X_ACC);
//Get card1 X axis，current acceleration。
```

3.5 Get DI status

- **short PS400_Get_DI_Status(BYTE *cardNo*, WORD *axis*, WORD* *pStatus*)**

Func.: Get DI status。

Para.:
- ***cardNo***: card ID
- ***axis***: axis no (1 or 2 or 4 or 8)
- ***pStatus***: point to the memory of the returned DI status.
- ****pStatus***: the each bit of variable that pStatus points to means:
  - Bit0 → DRIVING (Check Pusle output)
  - Bit1 → LIMIT+ (Check Limit switch)
  - Bit2 → LIMIT- (Check Limit swithc)
  - Bit3 → EMERGENCY (Check EMG)
  - Bit4 → ALARM (Check ALM)
  - Bit5 → HOME (Check ORG)
  - Bit6 → NEAR HOME (Chenck NORG)
  - Bit7 → IN3 (Check IN3)
  - Bit8 → INPOS (Check INP)

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid

Sample:
```
WORD Status;
PS400_Get_DI_Status(1, AXIS_X, &Status);
```

```
if ( Status&0x02 )
 {
      //Get card1   X axis current limit switch.
 }
```

3.6 Get or Clear error status

● **short PS400_Get_Error_Status(BYTE *cardNo*)**

Func.: Get error status。

Para.: ***cardNo***: card ID

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
CARD_NUMBER_ERROR: the *cardNo* is invalid
MOTION_STATUS_ERROR: some error occurred in internal Motion-ASIC,
please call PS400_GET_ERROR_CODE() for detailed information

Sample:

```
if (PS400_Get_Error_Status (1) == MOTION_STATUS_ERROR)
{
     //Get card1 error status。
}
```

● **short PS400_Get_Error_Code(BYTE *cardNo*, WORD *axis*, BYTE* *pErrCode*)**

Func.: Get error code。

Para.:
cardNo: card ID
axis: axis no(1 or 2 or 4 or 8)
pErrCode: point to the memory of the returned Error Code.

***pErrCode* will be the combination of the following values:**

code	cause	description
0x1	SOFT LIMIT+	Positive Software Limit
0x2	SOFT LIMIT-	Negative Software Limit
0x4	LIMIT+	Positive limit
0x8	LIMIT-	Negative limit
0x10	ALARM	Alarm
0x20	EMERGENCY	Emergency
0x40	Reserved	Reserve
0x80	HOME	Z-phase and HOME on

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid

Sample:

```
BYTE ErrCode;
PS400_Get_Error_Code(1, AXIS_X, &ErrCode);

if ( (ErrCode & 0x01 == 0x1) || (ErrCode & 0x2) )
{
    //Get card1 Xaxis，positive or negative limit.
}
```

4 Interrupt Handling

4.1 Enable / Disable interrupt function

- **short PS400_Enable_INT(BYTE *cardNo*)**

Func.: **Enable interrupt function。**

Para.: ***cardNo*:** **card ID**

Return: **SUCCESS_NO_ERROR**
INVALID_DEVICE_ERROR: no active device is related to the cardNo
CARD_NUMBER_ERROR: the *cardNo* is invalid
IOCTL_FAIL_ERROR: the DeviceIoControl( ) failed, please call GetLastError( ) for further information

Sample:
```
PS400_Enable_INT(1);
//Set card1 interrupt enable。
```

- **short PS400_Disable_INT(BYTE *cardNo*)**

Func.: **Disable interrupt function。**

Para.: ***cardNo*:** **card ID**

Return: **SUCCESS_NO_ERROR**
INVALID_DEVICE_ERROR: no active device is related to the cardNo
CARD_NUMBER_ERROR: the *cardNo* is invalid
IOCTL_FAIL_ERROR: the DeviceIoControl( ) failed, please call GetLastError( ) for further information

Sample:
```
PS400_Disbale_INT(1);
//Set card1 interrupt enable。
```

4.2 Set interrupt factor

- **short PS400_Set_INT_Factor(BYTE *cardNo*, WORD *axis*, WORD *Factor*)**

Func.: **Set interrupt factor。**

Para.: ***cardNo*:** **card ID**
***axis*:** **axis no(Table3-1)**
***Factor*:** **interrupt factor in the following table**

no	Code	Description
0	RST	Reset
1	PULSE	First pulse happens
2	P>=C-	Counter is greater than Comparator in negative

		direction
3	P<C-	Counter is smaller than Comparator in negative direction
4	P>=C+	Counter is greater than comparator in positive direction
5	P<C+	Counter is smaller than comparator in positive direction
6	C-END	Constant speed is ended
7	C-STA	Constant speed starts
8	D-END	Deceleration is ended
9	H-END	Homing is ended
10	SYNC	Synchronization is active

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid
MODE_SETTING_ERROR: the *Factor* parameter is invalid

Sample: PS400_Set_INT_Factor(1, AXIS_XY, 8);
// Set card 1 Interrupt active when deceleration is ended

Notice:

When PS400_Set_INT_Factor() configures interrupt with CMP revelant factors, the Software Limit be disabled. Please call PS400_Disable_SoftLimit( ) to disable software interrupt.

4.3 Get interrupt status

- short PS400_Get_INT_Status(BYTE *cardNo,* WORD *axis,* WORD* *pINTSts*)

Func.: Get interrupt status。

Para.: *cardNo*: card ID
axis: axis no (1 or 2 or 4 or 8)
pINTSts: point to the memory of the returned Inetrrupt-Status.

***pINTSts* will be the combination of the following values:**

code	Cond.	description
0x0001	PULSE	First pulse happens
0x0002	P>=C-	Counter is greater than Comparator in negative direction
0x0004	P<C-	Counter is smaller than Comparator in negative direction
0x0008	P<C+	Counter is greater than comparator in positive direction
0x0010	P<=C-	Counter is smaller than comparator in negative

		direction
0x0020	C-END	Constant speed is ended
0x0040	C-STA	Constant speed starts
0x0080	D-END	Deceleration is ended

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid
IOCTL_FAIL_ERROR: the DeviceIoControl( ) failed, please call GetLastError( ) for further information

Sample:
```
WORD wIntStatus;
PS400_Get_INT_Status(0, AXIS_X, &wIntStatus);
//Get card 0  Xaxis，interrupt status
```

Notice:

This function will get the correct status after enabling Interrupt with PS400_Enable_INT( ).

4.4 Attach/DettachEVENT objects for Interrupt notification

- short PS400_Attach_INT(BYTE *cardNo,* HANDLE **eHandle_List*)

Func.: Attache the EVENT objects to the related Interrupt. The EVENT object will be raised when the Interrupt occurs. WaitForSingleObjcet( ) is need, and usually applied in independent Thread.

Para.: *cardNo*: card ID
eHandle_List: the start-address of EVENT-HANDLE array. The array with 4-EVENT handles must be decalared in user's program.

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
EVENT_CREATE_ERROR: the CreateEvent( ) function failed.
CARD_NUMBER_ERROR: the *cardNo* is invalid
IOCTL_FAIL_ERROR: the DeviceIoControl( ) failed, please call GetLastError( ) for further information.

Sample:
```
HANDLE Event_Handle_List[4];
PS400_Attach_INT (1, Event_Handle_List);
//attach the Event Handle array to Card 1
WaitForSingleObject( Event_Handle_List[0], 5000 );
//wait the EVENT of AXIS_X in 5000 ms
```

- **short PS400_Dettach_INT(BYTE *cardNo*)**

Func.: **Dttache the EVENT objects to the related Interrupt and release the created resource.**

Para.: ***cardNo*:** **card ID**

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
CARD_NUMBER_ERROR: the *cardNo* is invalid
IOCTL_FAIL_ERROR: the DeviceIoControl( ) failed, please call GetLastError( ) for further information.

Sample:

```
PS400_Dettach_INT (1);
//deattch the EVENT-HANDLE and release the allocated resources
```

5 Read&Write from&to FRnet

5.1 Read or Wirde data

- **short PS400_Read_FRnet(BYTE *cardNo*, WORD *RA*, WORD* *pRAData*, BOOL *bDirectAccess* = TRUE)**

Func.: Read FRnet input

Para.: *cardNo*: card ID
RA: RA8~RA15
pRAData: point to the memory of the returned FRnet input.
bDirectAccess:
TRUE: Direct access the FRnet modelus (default)
FALSE: Read the digital input that saved by periotic Timer-ISR.

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
CARD_NUMBER_ERROR: the *cardNo* is invalid
INVALID_MODE_ERROR: the *RA* parameter is invalid
IOCTL_FAIL_ERROR: the DeviceIoControl( ) failed, please call GetLastError( ) for further information.

Sample:
```
WORD IN_Data;
PS400_Read_FRnet(1, 8, &IN_Data);
//Set card1，RA = 8，IN_Data in 16-bits。
```

- **short PS400_Write_FRnet(BYTE *cardNo, WORD SA, WORD data*)**

Func.: Write FRnet output。

Para.: *cardNo*: card ID
SA: groupSA0~SA7
dara: 16-bits data

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
CARD_NUMBER_ERROR: the *cardNo* is invalid
INVALID_MODE_ERROR: the *SA* parameter is invalid
IOCTL_FAIL_ERROR: the DeviceIoControl( ) failed, please call GetLastError( ) for further information.

Sample:
```
PS400_Write_FRnet(1, 0, 0xffff);
//Set card1，SA = 0，16-bits data is 0xffff。
```

5.2 Enable/Disable FRnet DO functionality

- **short PS400_Set_FRnetDO(BYTE cardNo, BYTE *Mode*)**

Func.: **Enable/Disable the DO functionality.**

Para.: ***cardNo*:** **card ID**
***Mode:*:** **0:Disable, 1:Enable**

Return: **SUCCESS_NO_ERROR**
INVALID_DEVICE_ERROR: no active device is related to the cardNo
CARD_NUMBER_ERROR: the *cardNo* is invalid
IOCTL_FAIL_ERROR: the DeviceIoControl( ) failed, please call GetLastError( ) for further information.

Sample:
```
// Enable card1 FRnet DO functionality
PS400_Set_FRnetDO(1, 1);
```

5.3 Ckect available FRnet DI modules

- **short PS400_Get_FRnetStatus(BYTE *cardNo, WORD* pStatus*)**

Func.: **check the on-line FRnet DI modules.**

Para.: ***cardNo*:** **card ID**
pStatus: **point to the memory of the returned available DI modules. The bit 0~bit7 of *pStatus stands for every FRnet DI module. For instance, 0x42 means two available FRnet DI modules set with RA9 and RA14.**

Return: **SUCCESS_NO_ERROR**
INVALID_DEVICE_ERROR: no active device is related to the cardNo
CARD_NUMBER_ERROR: the *cardNo* is invalid
IOCTL_FAIL_ERROR: the DeviceIoControl( ) failed, please call GetLastError( ) for further information.

Sample:
```
// check all on-line FRnet DI modules of card1
WORD FRnet_DI_Status;
short Error_Code;
Error_Code = FRnet_DI_Status=PS400_Get_FRnetStatus(1, &FRnet_DI_Status);
```

5.4 Reset FRnet DO

- **short PS400_Reset_FRnet(BYTE *cardNo*)**

Func.: **reset all attached FRnet DO modules.**

Para.: ***cardNo*:** **card ID**

Return: **SUCCESS_NO_ERROR**
INVALID_DEVICE_ERROR: no active device is related to the cardNo
CARD_NUMBER_ERROR: the *cardNo* is invalid
IOCTL_FAIL_ERROR: the DeviceIoControl( ) failed, please call GetLastError( ) for further information.

Sample:
```
// reset all FRnet DO modules of card1
PS400_Reset_FRnet(1);
```

5.5 Enable periodic FRnet DI

- **short PS400_Enable_FRnet_Scan(BYTE *cardNo,* BYTE *PeriodFactor*)**

Func.: **enable internal Timer of FRNet, the ISR will read FRnet DI module periodically.**

Para.: ***cardNo*:** **card ID**
***PeriodFactor*:** **assign the period of FRnet Time (0~255).**
T = 2.88 ms * (***PeriodFactor*** + 1)

Return: **SUCCESS_NO_ERROR**
INVALID_DEVICE_ERROR: no active device is related to the cardNo
CARD_NUMBER_ERROR: the *cardNo* is invalid
IOCTL_FAIL_ERROR: the DeviceIoControl( ) failed, please call GetLastError( ) for further information.

Sample:
```
// assign card1 the period of 0.0288s
PS400_Enable_FRnet_Scan(1, 9);
```

5.6 Disable periodic FRnet DI

- **short PS400_Disable_FRnet_Scan(BYTE *cardNo*)**

Func: **disable internal Timer of FRNet**

Para.: ***cardNo*:** **card ID**

Return: **SUCCESS_NO_ERROR**
INVALID_DEVICE_ERROR: no active device is related to the cardNo
CARD_NUMBER_ERROR: the *cardNo* is invalid
IOCTL_FAIL_ERROR: the DeviceIoControl( ) failed, please call GetLastError( ) for further information.

Sample:
```
// disable FRnet internal Timer
PS400_Disable_FRnet_Scan(1);
```

6 Auto-Homing

PS400 supports hardware auto-homing function,please follow the following steps to complete the related setting.:

- **high-speed to search NORG**
- **low-speed to search ORG**
- **low-speed to get Z-phase of encoder**
- **high-speed to move to assigned offset point**

Not all the above is necessary in the setting, some steps can skipped if necessary。

6.1 Set Auto-homing parameter

- **short PS400_Set_HomeSpeed(BYTE *cardNo*, WORD *axis*, *DWORD SV* ,DWORD *V,* DWORD *A,* DWORD *HV*)**

Func.: Set homing speed。

Para.:

	***cardNo*:**	**card IDd**
	axis:	**axis no(Table3-1)**
	***SV*：**	**start-speed**
	***V*：**	**constant speed**
	***A*：**	**acceleration**
	***HV*:**	**speed after NORG (Vmin~Vmax PPS)**

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
CARD_NUMBER_ERROR: the *cardNo* is invalid

Sample:
```
PS400_ Set_HomeSpeed(1, AXIS_X, 500, 20000, 10000, 5000);
//Set card1 X axis，start speed is 500PPS，constant speed is 20000PPS，
  accelerationis 10000PPS/Sec, speed after NORG is 5000 PPS。
```

6.2 Set limit switch as ORG

- **short PS400_Set_LimitHome(BYTE *cardNo*,WORD *axis*, BYTE *Mode*)**

Func.: Set limit switch EL as ORG。

Para.:

	***cardNo*:**	**card IDd**
	***axis*:**	**axis no (Table3-1)**
	***Mode*:**	**0=cancel ,1= enable**

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
CARD_NUMBER_ERROR: the *cardNo* is invalid

Sample: PS400_ Set_LimitHome(1, AXIS_X, 0);
//Set card1 X axis，cancel Limit as ORG。

6.3 Set homing mode

- short PS400_Set_HomeMode(BYTE *cardNo*, WORD *axis*, WORD *Step1*, WORD *Step2*, WORD *Step3*, WORD *Step4* ,long *Offset*)

Func.: Set homing mode and related parameter。

Para.:
cardNo: card ID
axis: axis no. (Table3-1)
Step1: 0= skip,1=positive direction search,2=negative direction search
Step2: 0= skip,1= positive direction search,2= negative direction search
Step3: 0= skip,1= positive direction search,2= negative direction search
Step4: 0= skip,1= positive direction search,2= negative direction search
Offset: offset (range：0 ~ 2,147,483,647)

The Homing Steps are defined as follows:

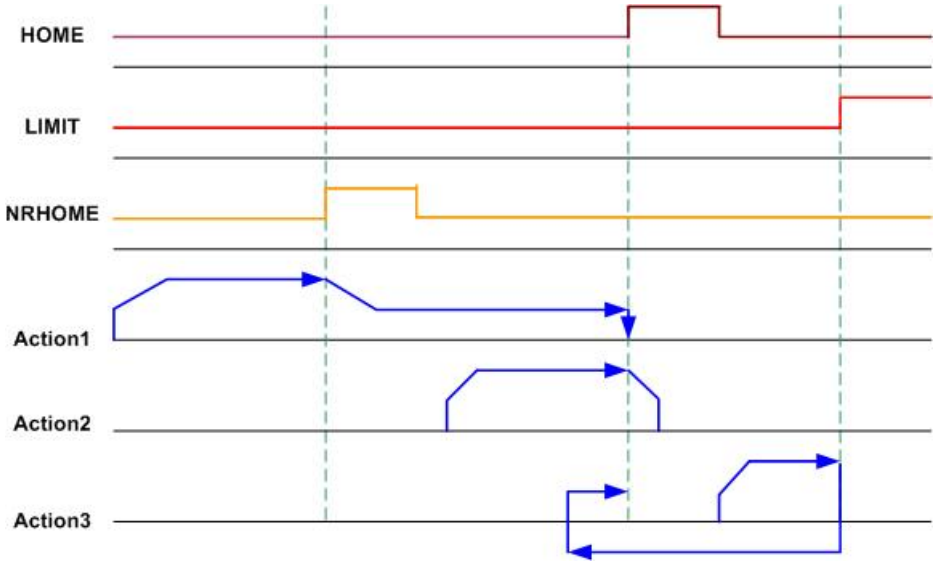
Step	Action	Speed	Sensor input
Step 1	High-speed to NORG	(V)	NORG(IN0)
Step 2	Low-speed to ORG	(HV)	ORG(IN1)
Step 3	Low-speed to Z-phase	(HV)	Z-phase(IN2)
Step 4	High-speed to offset	(V)	

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
CARD_NUMBER_ERROR: the *cardNo* is invalid

Sample:
```
BYTE bDone;
PS400_Set_MaxSpeed(1, AXIS_X, 200000);
// Set card1，4-axis max. speed as 200K PPS。
PS400_Set_HomeSpeed(1, AXIS_X, 5000, 20000, 10000, 500);
PS400_Set_HomeMode(1, AXIS_X, 2, 2, 0, 1, 0);
PS400_Home_Start(1, AXIS_X);
PS400_Home_Done(1, AXIS_X, &bDone);
//Set card1 X axis with the following setting:
```

	Input	Direction	Speed
step 1	NORG (IN0) Low active	－	20000 (PPS) (V)
step 2	ORG (IN1) Low active	－	500 (PPS) (HV)
step 3	Z-phase (IN2) High active	None	500 (PPS) (HV)
step 4	0 pulse (offset)	＋	20000 (PPS) (V)

IN3 is reserved for user-defined input



The diagram illustrates the above homing sequence

6.4 Start homing process

- **short PS400_Home _Start(BYTE *cardNo*, WORD *axis*)**

Func.: **Start homing process after finish the related setting。**

Para.: ***cardNo*:** **card ID**
***axis*:** **axis no (Table3-1)**

Return: **SUCCESS_NO_ERROR**
INVALID_DEVICE_ERROR: no active device is related to the cardNo
CARD_NUMBER_ERROR: the *cardNo* is invalid

Sample: **PS400_ Home_Start(1, AXIS_X);**
//Set card1 X axis，starts homing process。

6.5 Wait for homing completion

- **BYTE PS400_Home_Done(BYTE *cardNo*, WORD *axis,* BYTE* *pDone*)**

Func.: **uses this to wait for homing completion。**

Para.: ***cardNo*:** **card ID**
***axis*:** **axis no (Table3-1)**
***pDone*:** **point to the memory that indidates the completion of Homing.**
YES -- completed
NO -- not-completed

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
CARD_NUMBER_ERROR: the *cardNo* is invalid

Sample:
```
BYTE bDone;
PS400_Home_Done(1, AXIS_X, &bDone);
If ( bDone == NO)
{
    //Check card1 X axis , homing is not completed。
}
```

7 Axis control

- **Multiple axes can be commanded for single axis motion。**
- **T-curve and S-curve velocity profile are supported**
- **2~3 axis linear-interpolation。**
- **2-axis circular-interpolation。**
- **3-axis helical-interpolation。**
- **2-axis proportional movement。**

7.1 Fixed pulse with constant speed

- **short PS400_Const_Move(BYTE *cardNo*, WORD *axis*, DWORD *V*, long *Command*)**

Func.: **Set fixed pulse with constand speed movement。**

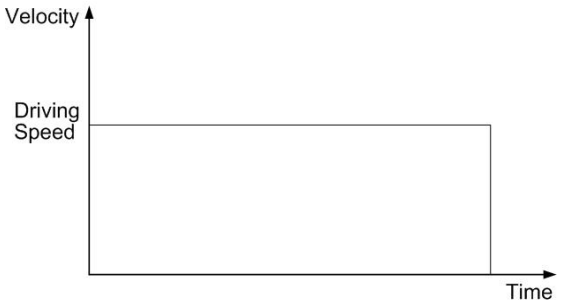
Para.:

	***cardNo*:**	**card ID**
	***axis*:**	**axis no (1 or 2 or 4 or 8)**
	***V*:**	**speed in (PPS)**
	***Command*:**	**Pusle (range：-2,147,483,648 ~ +2,147,483,647)**

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
CARD_NUMBER_ERROR: the *cardNo* is invalid
MOTION_STATUS_ERROR: some error occurred in internal Motion-ASIC, please call PS400_GET_ERROR_CODE() for detailed information

Sample:

```
BYTE cardNo=0;
DWORD v=10000;
DWORD p=10000;
PS400_Set_MaxSpeed(cardNo, AxIS_XYZU, 200000)
// Set card 0, 4 axes max. speed as 200K PPS.
PS400_Const_Move(0, AXIS_X, v, p)
//Set card 0, X-axis 10000 pulse movement with constant speed as 10000 PPS.
```



7.2 Fixed pulse with T-curve velocity profile

- **short PS400_T_Move(BYTE *cardNo*, WORD *axis*, DWORD *SV*, DWORD *V*, DWORD *A*, short *AO*, long *Command*)**

Func.: **Set fixed pulse with symmetric T-curve velocity profile。**

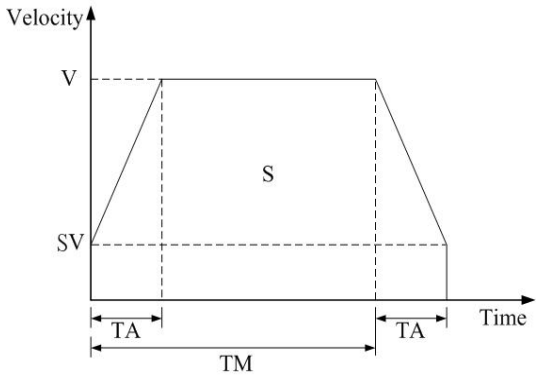
Para.:

	cardNo:	**card ID**
	axis:	**axis no (1 or 2 or 4 or 8)**
	SV:	**start speed (range：1~4M PPS)**
	V:	**Max. speed (range：1~4M PPS)**
	A:	**Acceleration (PPS/Sec) (range：(PS400_Set_MaxSpeed data)(÷64 ~ x125 PPS/Sec)**
	AO:	**Offset pulse (-32,768 ~ +32,767)**
	Command:	**pulse (-2,147,483,648 ~ +2,147,483,647)**

Return: **SUCCESS_NO_ERROR**
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_SPEED_ERROR: invalid *SV* or *V* parameter
INAVLID_ACCELERATION_ERROR: invalid *A* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid
MOTION_STATUS_ERROR: some error occurred in internal Motion-ASIC, please call PS400_GET_ERROR_CODE() for detailed information

Sample:

```
BYTE cardNo=0;
DWORD sv=500;
DWORD v=10000;
DWORD a=5000;
short ao=0;
DWORD p=10000;
PS400_Set_MaxSpeed(cardNo, AXIS_XYZU, 200000);
// Set card0, 4 axes Max. speed as 200K PPS.
PS400_T_Move(cardNo, AXIS_X, sv, v, a, ao, p);
//Set card 0, X-axis with T-curve.
```



- **short PS400_T_As_Move(BYTE *cardNo*, WORD *axis*, DWORD *SV*, DWORD *V*, DWORD *A*, WORD *D*, short *AO*, long *Command*)**

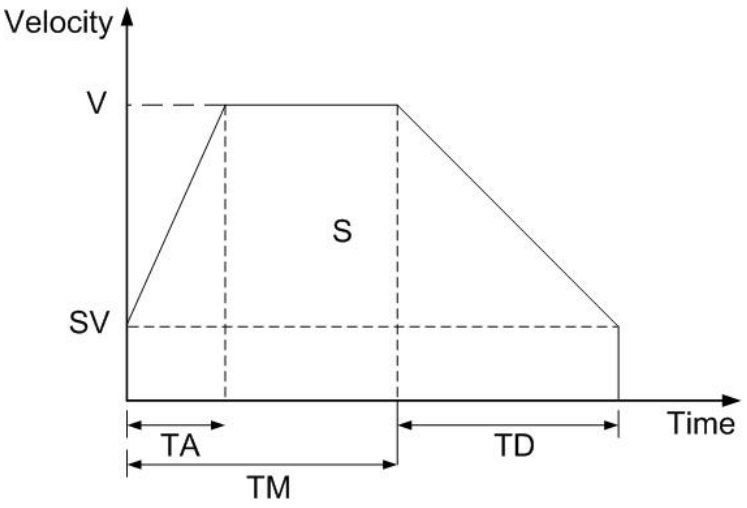
Func.: **Set fixed pulse non-symmetric T-curve velocity profile。**

Para.:

***cardNo*:**	**card ID**
***axis*:**	**axis no (1 or 2 or 4 or 8)**
***SV*:**	**start speed (range：1~4M PPS)**
***V*:**	**Max. speed (PPS)**
***A*:**	**Acceleration (PPS/Sec)**
***D*:**	**Deceleration(PPS/Sec)**
***AO*:**	**Offset Pulse (-32,768 ~ +32,767)**
***Command*:**	**pulse (-2,147,483,648 ~ +2,147,483,647)**

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_SPEED_ERROR: invalid *SV* or *V* parameter
INAVLID_ACCELERATION_ERROR: invalid *A* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid
MOTION_STATUS_ERROR: some error occurred in internal Motion-ASIC, please call PS400_GET_ERROR_CODE() for detailed information

Sample:
```
BYTE cardNo=0;
DWORD sv=500;
DWORD v=10000;
DWORD a=8000;
short ao=0;
DWORD d=2000;
DWORD p=10000;
PS400_Set_MaxSpeed(cardNo, AxIS_XYZU, 200000)
// Set card 0, 4-axes max. speed as 200K PPS.
PS400_T_As_Move(cardNo, AXIS_X, sv, v, a, d, ao, p);
//Set card0, X-axis with non-symmetric T-curve profile.
```



7.3 Fixed Pusle with S-curve velocity profile

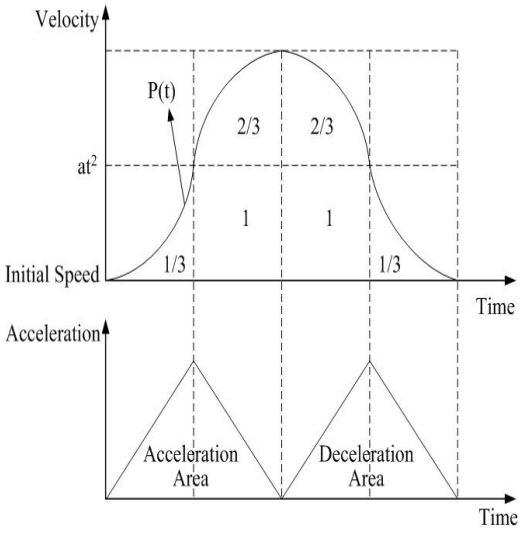
- **short PS400_S_Move(BYTE *cardNo*, WORD *axis*, DWORD *SV*, DWORD *V*, DWORD *K*, short *AO*, long *Command*)**

Func.: **Set fixed pulse with S-curve velocity profile。**

Para.:
- ***cardNo*: card ID**
- ***axis*: axis no(1 or 2 or 4 or 8)**
- ***SV*: start speed (range 1~4M PPS)**
- ***V*: speed (PPS)**
- ***K*: jerk (Jerk PPS/ Sec^2)**
- ***AO:* *Offset Pulse (-32,768 ~ +32,767)***
- ***Command*: pulse(range:-2,147,483,648 ~ +2,147,483,647)**

Return: **SUCCESS_NO_ERROR**
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_SPEED_ERROR: invalid *SV* parameter
INAVLID_ACCELERATION_ERROR: invalid *K* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid
MOTION_STATUS_ERROR: some error occurred in internal Motion-ASIC, please call PS400_GET_ERROR_CODE() for detailed information

Sample:
```
BYTE cardNo=0;
DWORD sv=500;
DWORD v=10000;
DWORD k=5000;
short ao=0;
DWORD p=10000;
PS400_Set_MaxSpeed(cardNo, AXIS_XYZU, 200000);
// Set card0, 4 axes mas. Speed as 200K PPS.
PS400_S_Move(cardNo, AXIS_X, sv, v, k, ao, p);
//Set card0, X-axis with symmetric  S-Curve profile.
```



Symmetric S-curve profile

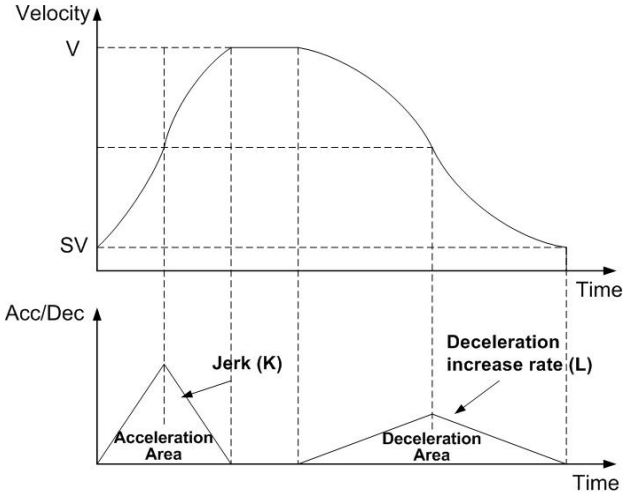
- **short PS400_S_As_Move(BYTE *cardNo*, WORD *axis*, DWORD *SV*, DWORD *V*, DWORD *K*, DWORD *L*, short *AO*, long *Command*)**

Func.: **Fixed pulse with non-symmetric S-curve velocity profile。**

Para.: ***cardNo*: card ID**
***axis*: axis no(1 or 2 or 4 or 8)**
***SV*: start speed (range：1~4M PPS)**
***V*: Max. speed(PPS)**
***K*: Acceleration Jerk (Jerk PPS/ Sec^2)**
***L*: Deceleration Jerk (Jerk PPS/ Sec^2)**
***AO*: Offset Pulse (-32,768 ~ +32,767)**
***Command*: pulse(-2,147,483,648 ~ +2,147,483,647)**

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_SPEED_ERROR: invalid *SV* parameter
INAVLID_ACCELERATION_ERROR: invalid *K* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid
MOTION_STATUS_ERROR: some error occurred in internal Motion-ASIC, please call PS400_GET_ERROR_CODE() for detailed information

Sample:
```
BYTE cardNo=0;
DWORD sv=500;
DWORD v=10000;
DWORD k=8000;
DWORD l=2000;
Short ao=0;
DWORD p=10000;
PS400_Set_MaxSpeed(cardNo, AXIS_XYZU, 200000);
// Set card 0, 4 axes max. speed as 200K PPS.
PS400_S_As_Move(cardNo, AXIS_X, sv, v, k, l, ao, p);
//Set card0, X-axis with non-symmetric S-Curve.
```



Non-symmetric S-curve

7.4 Continuous Movement

- **short PS400_Conti_move(BYTE *cardNo*, WORD *axis*, DWORD *SV*, long *V*, DWORD *A*, BYTE Dir)**

Func.: **Start continuous motion until the following situation occurs:**
- **calling PS400_Set_SdStop() or PS400_Set_EmgStop()**
- **satisifying the configuration of hardware/software limit**

Para.:

	***cardNo*:**	**card ID**
	***axis*:**	**axis no (1 or 2 or 4 or 8)**
	***SV*:**	**start speed (PPS) (range：1~4M PPS)**
	***V*:**	**max. speed (PPS)**
	***A*:**	**acceleration (PPS/Sec)**
	Dir	**Direction : 0 : Positive, 1 : Negative**

Return: **SUCCESS_NO_ERROR**
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_SPEED_ERROR: invalid *SV* parameter
INAVLID_ACCELERATION_ERROR: invalid *A* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid
MOTION_STATUS_ERROR: some error occurred in internal Motion-ASIC, please call PS400_GET_ERROR_CODE() for detailed information

Sample:
```
BYTE cardNo=0;
DWORD sv=500;
DWORD v=10000;
DWORD a=8000;
PS400_Set_MaxSpeed(cardNo, AXIS_XYZU, 200000);
// Set card0, 4 axes max. speed as 200K PPS.
PS400_Conti_Move(cardNo, AXIS_X, sv, v, a, 0);
//Set card 0, X-axis with T-curve profile.
```

7.5 Linear interpolation

- **short PS400_Line2_Move(BYTE *cardNo*, WORD *axis1*, WORD *axis2*, BYTE *Mode*, DWORD *SV*, DWORD *V*, DWORD *A*, DWORD *K*,short *AO*, long *FP1*, long *FP2*)**

Func.: **2-axis linear interpolation in the same card。**

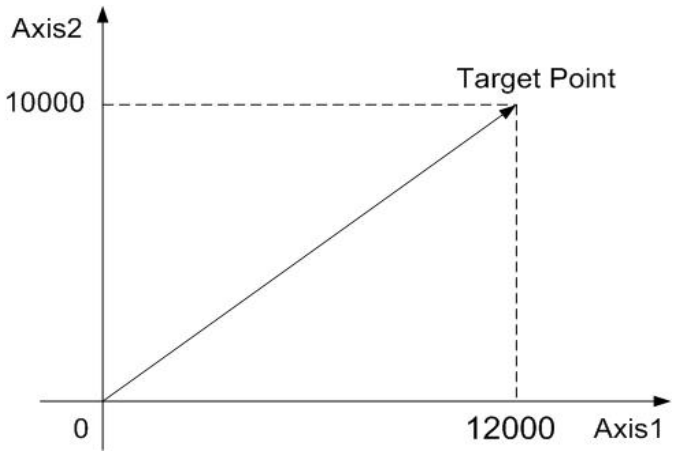
Para.:

	***cardNo*:**	**card ID**
	***axis1*:**	**first axis no.: X、Y、Z、U (1、2、4、8)**
	***axis2*:**	**second axis no: X、Y、Z、U (1、2、4、8)**
	***Mode*:**	**0 → fixed speed (V)**
		1 → Symmetric T-profile (SV、V、A、AO)
		2 → Symmetric S-curve (SV、V、K、AO)

***SV*:**	**Start speed (range：1~4M PPS)**
***V*:**	**Max. speed(PPS)**
***A*:**	**Acceleration**
***K*:**	**Jerk (Jerk PPS/ Sec^2)**
***AO*:**	**Offset Pulse (-32,768 ~ +32,767)**
***FP1*:**	**first axis Pulse (-2,147,483,648 ~ +2,147,483,647)**
***FP2*:**	**second axis Pulse(-2,147,483,648 ~ +2,147,483,647)**

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis1* or *axis2* parameter
INVALID_MODE_ERROR: invalid *Mode* parameter
INVALID_SPEED_ERROR: invalid *SV* parameter
INAVLID_ACCELERATION_ERROR: invalid *A or K* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid
MOTION_STATUS_ERROR: some error occurred in internal Motion-ASIC, please call PS400_GET_ERROR_CODE() for detailed information

Sample:
```
BYTE cardNo=0;
BYTE Mode=1;
DWORD sv=500;
DWORD v=10000;
DWORD a=8000;
DWORD k=0; // k 可為任意值
short ao=0;
DWORD fp1=10000;
DWORD fp2=10000;
PS400_Set_MaxSpeed(cardNo, AXIS_XYZU, 200000);
// Set card1. 4 axes max. speed with 200K PPS.
PS400_Line2_Move(cardNo, AXIS_X, AXIS_Y, Mode, sv, v, a, k, ao, fp1, fp2);
//Set card0, XY axis with symmetric T-curve profile.
```



2-axis linear interpolation

- **short PS400_Line2_As_Move(BYTE *cardNo*, WORD *axis1*, WORD *axis2*, BYTE *Mode*, DWORD *SV*, DWORD *V*, DWORD A, DWORD *D*, DWORD *K*, DWORD *L*, short *AO*, long *FP1*, long *FP2*)**

Func.: 2-axis non-symmetric linear interpolation。

Para.:

cardNo:	card ID
axis1:	first axis: X、Y、Z、U (1、2、4、8)
axis2:	second axis: X、Y、Z、U (1、2、4、8)
Mode:	0 → 2-axis constant speed (V)
	1 → non-symmetric T-curve (SV、V、A、AO)
	2 → non-symmetric S-curve (SV、V、A、D、AO)
SV:	Start speed (range：1~4M PPS)
V:	Max. speed(PPS)
A:	Acceleration Jerk (PPS/Sec)
D:	Deceleration Jerk (PPS/Sec)
K:	Jerk (Jerk PPS/ Sec^2)
L:	Deceleration Jerk (Jerk PPS/ Sec^2)
AO:	Offset Pulse (-32,768 ~ +32,767)
FP1:	first axis Pulse(-2,147,483,648 ~ +2,147,483,647)
FP2:	second axis Pulse(-2,147,483,648 ~ +2,147,483,647)

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis1* or *axis2* parameter
INVALID_MODE_ERROR: invalid *Mode* parameter
INVALID_SPEED_ERROR: invalid *SV* parameter
INAVLID_ACCELERATION_ERROR: invalid *A, D, K or L* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid
MOTION_STATUS_ERROR: some error occurred in internal Motion-ASIC, please call PS400_GET_ERROR_CODE() for detailed information

Sample:

```
BYTE cardNo=0;
DWORD sv=500;
DWORD v=10000;
DWORD a=8000;
DWORD d=2000;
DWORD k=0;// k can be any value
DWORD l=0;// l can be any value
 short ao=0;
DWORD fp1=10000;
DWORD fp2=10000;
PS400_Set_MaxSpeed(cardNo, AXIS_XYZU, 200000);
// Set card1, 4 axes max. speed as  200K PPS。
PS400_Line2_As_Move(cardNo, AXIS_X, AXIS_Y, 1, sv, v, a, d, k, l, ao, fp1, fp2);
//Set card0, XY axes with symmetric T-curve profile。
```

- **short PS400_Line3_Move(BYTE *cardNo*, WORD *axis1*, WORD *axis2*, WORD *axis3*,**

BYTE *Mode*, DWORD *SV*, DWORD *V*, DWORD *A*,
DWORD *K*, short *AO*, long *FP1*, long *FP2,* long *FP3*)

Func.: **3-axis symmetric linear interpolation in the same card。**

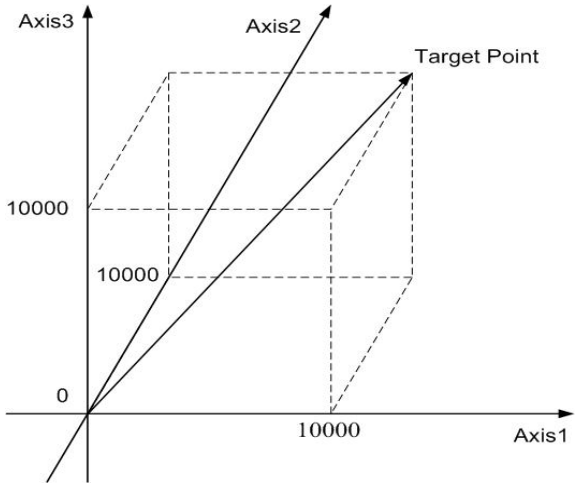
Para.:

***cardNo*:**	**card ID**
***axis1*:**	**first axis: X、Y、Z、U (1、2、4、8)**
***axis2*:**	**second axis: X、Y、Z、U (1、2、4、8)**
***axis3*:**	**third axis: X、Y、Z、U (1、2、4、8)**
***Mode*:**	**0 → fixed constant speed (V)**
	1 → symmetrci T-curve (SV、V、A、AO)
	2 → symmetric S-curve (SV、V、K、AO)
***SV*:**	**Start speed (PPS) (range：1~4M PPS)**
***V*:**	**Max. speed(PPS) (range：1~4M PPS)**
***A*:**	**Acceleration (PPS/Sec)**
***K*:**	**Acceleration Jerk (Jerk PPS/ Sec^2)**
***AO*:**	**Offset Pulse (-32,768 ~ +32,767)**
***FP1*:**	**first axis Pulse(-2,147,483,648 ~ +2,147,483,647)**
***FP2*:**	**second axis Pulse(-2,147,483,648 ~ +2,147,483,647)**
***FP3*:**	**third axis Pulse (-2,147,483,648 ~ +2,147,483,647)**

Return: **SUCCESS_NO_ERROR**
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis1, axis2* or *axis3* parameter
INVALID_MODE_ERROR: invalid *Mode* parameter
INVALID_SPEED_ERROR: invalid *SV or V* parameter
INAVLID_ACCELERATION_ERROR: invalid *A or K* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid
MOTION_STATUS_ERROR: some error occurred in internal Motion-ASIC,
please call PS400_GET_ERROR_CODE() for detailed information

Sample:

```
BYTE cardNo=0;
BYTE mode=2;
DWORD sv=500;
DWORD v=10000;
DWORD a=0;//a can be any value
DWORD k=8000;
 short ao=0;
DWORD fp1=10000;
DWORD fp2=10000;
DWORD fp3=10000;
PS400_Set_MaxSpeed(cardNo, AXIS_XYZU, 200000);
// Set card1, 4 axes with max. speed  200K PPS。
PS400_Line3_Move(cardNo, AXIS_X, AXIS_Y, AXIS_Z, mode, sv, v, a, k, ao, fp1, fp2,
fp3);
//Set card0, XYZ axes symmetric S-Curve。
```



3-axis linear interpolation

- **short PS400_Line3_As_Move(BYTE *cardNo*, WORD *axis1*, WORD *axis2*, WORD *axis3*, BYTE *Mode*, DWORD *SV*, DWORD *V*, DWORD *A*, DWORD *D*, DWORD *K*, DWORD *L*, short *AO*, long *FP1*, long *FP2,* long *FP3*)**

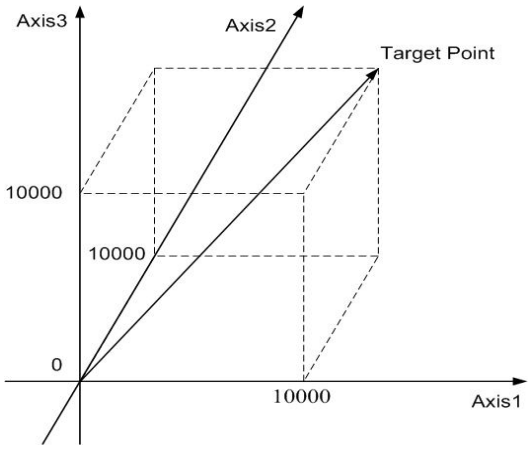
Func.: **3-axis non-symmetric linear interpolation in the same card。**

Para.: ***cardNo*: card ID**
***axis1*: first axis: X、Y、Z、U (1、2、4、8)**
***axis2*: second axis: X、Y、Z、U (1、2、4、8)**
***axis3*: third axis: X、Y、Z、U (1、2、4、8)**
***Mode*: 0 → fixed constant speed (V)**
1 → non-symmetric T-curve (SV、V、A、AO)
2 → non-symmetric S-curve (SV、V、A、D、AO)
***SV*: Start speed (range：1~4M PPS)**
***V*: Max. speed (range：1~4M PPS)**
***A*: Acceleration**
***D*: Deceleration**
***K*: Acceleratin Jerk (Jerk PPS/ Sec^2)**
***L*: 設定向量減速度變化率值 (Jerk PPS/ Sec^2)**
***AO*: Offset Pulse (-32,768 ~ +32,767)**
***FP1*:first axis Pulse(-2,147,483,648 ~ +2,147,483,647)**
***FP2*:second axis Pulse(-2,147,483,648 ~ +2,147,483,647)**
***FP3*:third axis Pulse(-2,147,483,648 ~ +2,147,483,647)**

Return: **SUCCESS_NO_ERROR**
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis1, axis2* or *axis3* parameter
INVALID_MODE_ERROR: invalid *Mode* parameter
INVALID_SPEED_ERROR: invalid *SV or V* parameter
INAVLID_ACCELERATION_ERROR: invalid *A, D, K or L* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid
MOTION_STATUS_ERROR: some error occurred in internal Motion-ASIC, please call PS400_GET_ERROR_CODE() for detailed information

Sample: **BYTE cardNo=0;**

```
BYTE mode=2;
DWORD sv=500;
DWORD v=10000;
DWORD a=0;//a can be any value
DWORD d=0;//d can be any value
DWORD k=8000;
DWORD l=4000;
short ao=0;
DWORD fp1=10000;
DWORD fp2=10000;
DWORD fp3=10000;
PS400_Set_MaxSpeed(cardNo, AXIS_XYZU, 200000);
// Set card1, 4 axes with max. speed as 200K PPS.
PS400_Line3_As_Move(cardNo, AXIS_X, AXIS_Y, AXIS_Z, mode, sv, v, a, d, k, l, ao, fp1, fp2,
fp3);
//Set card 0, XYZ axes as non-symmetric S-Curve.
```



3-axis non-symmetric linear interpolation

7.6 Circular interpolation

- **short PS400_Arc2_Move(BYTE *cardNo*, WORD *axis1*, WORD *axis2*, BYTE *Mode*, BYTE Dir, DWORD *SV*, DWORD *V*, DWORD *A*, long *CP1*, long *CP2* , long *FP1*, long *FP2*)**

Func.: 2-axis circular interpolation。

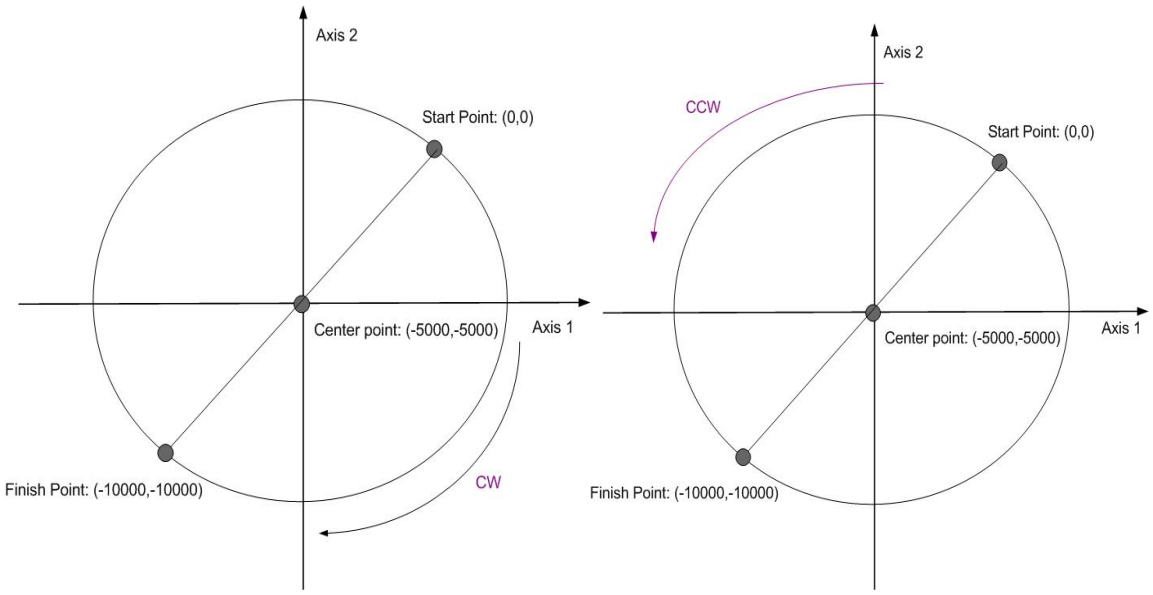
Para.:

***cardNo*:**	**card ID**
***axis1*:**	**first axis: X、Y、Z、U (1、2、4、8)**
***axis2*:**	**second axis: X、Y、Z、U (1、2、4、8)**
***Mode*:**	**0 → fixed constant speed (V)**
	1 → symmetric T-curve (SV、V、A)
Dir	**0: CW，clockwise；1:CCW，counter-clockwise**
***SV*:**	**Start velocity (PPS) (range：1~4M PPS)**
***V*:**	**Max. speed(PPS) (range：1~4M PPS)**
***A*:**	**Acceleration (PPS/Sec)**
***CP1*:**	**first axis center point (-2,147,483,648 ~ +2,147,483,647)**
***CP2*:**	**second axis center point (-2,147,483,648 ~ +2,147,483,647)**
***FP1*:**	**first axis Pulse(-2,147,483,648 ~ +2,147,483,647)**
***FP2*:**	**second axis Pulse(-2,147,483,648 ~ +2,147,483,647)**

Return: **SUCCESS_NO_ERROR**
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis1* or *axis2* parameter
INVALID_MODE_ERROR: invalid *Mode* parameter
INVALID_SPEED_ERROR: invalid *SV or V* parameter
INAVLID_ACCELERATION_ERROR: invalid *A* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid
MOTION_STATUS_ERROR: some error occurred in internal Motion-ASIC, please call PS400_GET_ERROR_CODE() for detailed information

Sample:

```
BYTE cardNo=0;
BYTE mode=1;
DWORD sv=500;
DWORD v=10000;
DWORD a=8000;
DWORD cp1=10000;
DWORD cp2=0;
DWORD fp1=0;
DWORD fp2=0;
PS400_Set_MaxSpeed(cardNo, AXIS_XYZU, 200000);
// Set card1 4axes with mas. Speed 200K PPS.
PS400_Arc2_Move(cardNo, AXIS_X, AXIS_Y, mode, 0, sv, v, a, cp1, cp2, fp1, fp2);
//Set card=0, XY axes with T-curve.
```



CW interpolation **CCW interpolation**

7.7 Synchronized Moton

- **short PS400_Set_SyncMotion(BYTE *cardNo*, WORD *axis1*, WORD *axis2*, BYTE *Sync*, BYTE *Drv*)**

Func.: **Set synchronized motion in the same card。**

Para.: ***cardNo*:** **card ID**
***axis1*:** **main axis no (1 or 2 or 4 or 8)**
***axis2*:** **sync. Axis no. in the following table**

axis1 / axis2	X	Y	Z	U
0	none	none	none	none
1	Y	Z	U	X
2	Z	U	X	Y
3	YZ	ZU	UX	XY
4	U	X	Y	Z
5	YU	ZX	UY	XZ
6	ZU	UX	XY	YZ
7	YZU	ZUX	UXY	XYZ

***Sync*:** **Sync. Condition in the following table**

code	Cond.	Description
0x0000		Cancel Sync. Condition
0x0001	P≥C+	Pulse counter is greater than or equal to COMP+
0x0002	P<C+	Pulse counter is smaller than COMP+
0x0004	P<C-	Pusle counter is smaller than COMP-
0x0008	P≥C-	Pulse counter is greater than or equal to COMP-
0x0010	C-STA	Constant speed starts

0x0020	D-END	Deceleration is ended
0x0040	IN3↑	nIN3 is in rising edge
0x0080	IN3↓	nIN3 is in falling edge

For instance: P≥C+ and IN3↑ (0x0001 + 0x0040 = 0x0041)

Drv: Syn. Action is in the following table

code	Cond.	Description
0		Cancel Sync. Condition
1	FDRV+	Positive dir. Fixed pulse movement, please use PS400_Sync_Preset() to set new position
2	FDRV-	Negative dir. Fixed pulse movement, please use PS400_Sync_Preset() to set new position
3	CDRV+	Positive continuous movement
4	CDRV-	Negative continuous movement
5	SSTOP	Deceleration stop
6	ISTOP	Immediate stop

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis1* or *axis2* parameter
INVALID_MODE_ERROR: invalid *Drv* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid

Sample:
```
BYTE cardNo=0;
PS400_Set_Compare(cardNo, AXIS_X, 0, 0, 20000);
//Set CP+ as 20000.
PS400_Set_MaxSpeed(cardNo, AXIS_XYZU, 200000);
// Set card 1, 4 axes max. speed as 200K PPS.
PS400_Set_SyncMotion(cardNo, AXIS_X, 1, 1, 3);
PS400_T_Move(cardNo, AXIS_X, 500, 10000, 5000, 0, 50000);
//Set X-axis with T-curve for 50000 pulses, when X-axis reaches 20000 pulse, Y axis
  starts continuous movement.
```

- **short PS400_Set_Latch(BYTE *cardNo,* WORD *axis1,* WORD *axis2,* BYTE *Sync,* BYTE *Latch*)**

Func.: Set position latch function in the same card。

Para.: *cardNo*: card ID
axis1: Main axis no. (1 or 2 or 4 or 8)
axis2: Sync. Axis no. in the following table

axis1 / axis2	X	Y	Z	U
0	None	None	None	None
1	Y	Z	U	X
2	Z	U	X	Y
3	YZ	ZU	UX	XY
4	U	X	Y	Z
5	YU	ZX	UY	XZ
6	ZU	UX	XY	YZ

7	YZU	ZUX	UXY	XYZ

Sync: Sync. Condition in the following table

code	Cond.	Description
0x0000		Cancel Sync. Action
0x0001	P≥C+	Pulse counter is greater than or equal to COMP+
0x0002	P<C+	Pulse counter is smaller than COMP+
0x0004	P<C-	Pusle counter is smaller than COMP-
0x0008	P≥C-	Pulse counter is greater than or equal to COMP-
0x0010	C-STA	Constant speed starts
0x0020	D-END	Deceleration is ended
0x0040	IN3↑	nIN3 is in rising edge
0x0080	IN3↓	nIN3 is in falling edge

For instance: P≥C+ and IN3↑ (0x0001 + 0x0040 = 0x0041)

Latch: Sync. Action in the following table

code	Cond.	Description
0		Cancel sync. action
1	LPSAV	Stors currnet position，[Command→BR]
2	EPSAV	Stors feedback position [Position→BR]

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis1* or *axis2* parameter
INVALID_MODE_ERROR: invalid *Latch* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid

Sample:
```
PS400_Set_MaxSpeed(1, AXIS_XYZU, 200000);
// Set card1, 4 axes max. speed as 200K PPS.
PS400_Set_Latch(1, AXIS_X, 0, 0x0040, 2);
PS400_T_Move(1, AXIS_X, 500, 20000, 10000, 0, 100000);
// Set X axis with T-Curve, Latch position when IN3 is in rising edge.
```

● **long PS400_Get_Latch(BYTE *cardNo,* WORD *axis,* long* *pLatch*)**

Func.: Get latched data。

Para.: *cardNo*: card ID
axis: axis no. (1 or 2 or 4 or 8)
pLatch: point to the memory of the latched data.

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis1* or *axis2* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid

Sample:
```
long data;
PS400_Get_Latch(1, AXIS_Y, &data);
//Get card1, Y axis , latched position
```

- **short PS400_Sync_Preset(BYTE *cardNo*, WORD *axis1*, WORD *axis2*, BYTE *SYNC*, BYTE *Preset*)**

Func.: Set Sync. Axis motion condition in the same card。

Para.: *cardNo*: card ID
axis1: main axis no. X, Y, Z or U (1 or 2 or 4 or 8)
axis2: sync. Axis no. in the following table

axis1 / axis2	X	Y	Z	U
0	none	none	none	none
1	Y	Z	U	X
2	Z	U	X	Y
3	YZ	ZU	UX	XY
4	U	X	Y	Z
5	YU	ZX	UY	XZ
6	ZU	UX	XY	YZ
7	YZU	ZUX	UXY	XYZ

Sync: Sync. Condition in the following table

code	Cond.	Description
0x0000		Cancel sync. condition
0x0001	P≥C+	Pulse counter is greater than or equal to COMP+
0x0002	P<C+	Pulse counter is smaller than COMP+
0x0004	P<C-	Pusle counter is smaller than COMP-
0x0008	P≥C-	Pulse counter is greater than or equal to COMP-
0x0010	D-STA	Constant speed starts
0x0020	D-END	Deceleration is ended
0x0040	IN3↑	nIN3 is in rising edge
0x0080	IN3↓	nIN3 is in falling edge

For instance, P≥C+ and IN3↑ (0x0001 + 0x0040 = 0x0041)

Preset: Sync. Data is in the following table

code	Cond.	Description
0		Cancel Sync. data
1	LPSET	Set new command counter，[Command←PRESET_DATA]
2	EPSET	Set new feedback counter (Position)，[Position←PRESET_DATA]
3	OPSET	Set new positin (P)，[P←PRESET_DATA]
4	VLSET	Set new velocity (V)，[V←PRESET_DATA]

To be used together with PS400_Preset_Data。

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis1* or *axis2* parameter
INVALID_MODE_ERROR: invalid *Preset* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid

- **short PS400_Preset_Data(BYTE *cardNo*, WORD *axis*, long *Data*)**

Func.: Set Sync. Motion data。

Para.:

	cardNo:	card ID
	axis:	Sync. axis(Table3-1)
	Data:	Command: -2,147,483,648 ~ +2,147,483,647
		Position: -2,147,483,648 ~ +2,147,483,647
		P : 0 ~ 4,294,967,295

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_MODE_ERROR: PS400_Sync_Preset( ) is not called
CARD_NUMBER_ERROR: the *cardNo* is invalid

Sample:
```
PS400_Disable_SoftLimit(1, AXIS_X);
//Disable X axis software limit.
PS400_Sync_Preset(1, AXIS_X, AXIS_Y, 1, 2);
//Set card1 and enable X-axis sync. Condition as P >= C+, change Y-axis speed.
PS400_Preset_Data(1, AXIS_Y, 10000);
PS400_Set_Compare(1, AXIS_X, 1, 0, 1000);
//Set card1 X-axis，P as feedback position(EP)，C+ is 1K
//When X-axis EP >= 1,000, Set Y-axis speed as 10K
```

7.8 Continuous interpolation

7.8.1 2-axis rectangle continuous interpolation

- **short PS400_Rectangle(BYTE *cardNo*, WORD *axis1*, WORD *axis2*, BYTE *Mode*, WORD *SP*, BYTE *Dir*, long *LP*, long *WP*, long *RP*, DWORD *RSV*, DWORD *RV*, DWORD *RA*, DWORD *RD*)**

Func.: 2-axis rectangle continuous interpolation。

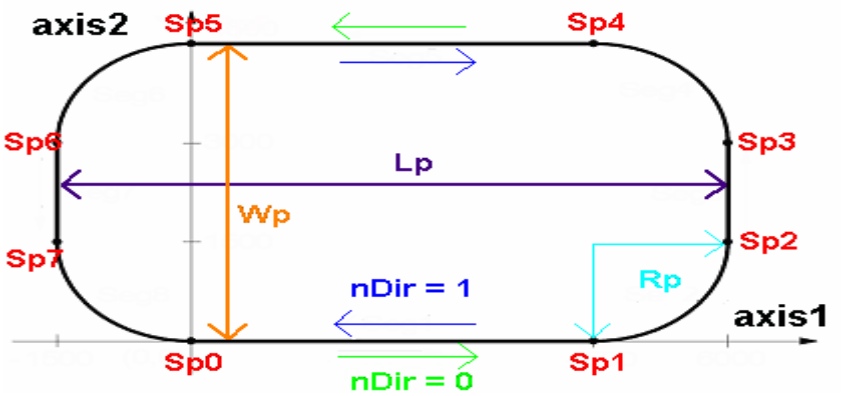
Para.:

	cardNo:	card ID
	axis1:	first axis: X、Y、Z、U (1、2、4、8)
	axis2:	second axis: X、Y、Z、U (1、2、4、8)
	Mode:	0 → constant speed
		1 → symmetric T-curve
	SP:	Start point 0 ~ 7 (Sp0 ~ Sp7 in the following fig.)
	Dir:	Direction with 0、1 (CCW or CW)
	LP:	Length Pulse no. (1 ~ 2,147,483,647)
	WP:	Width Pulseno.(1 ~ 2,147,483,647)
	RP:	Arc Pulse no.(1 ~ 2,147,483,647)
	RSV:	start velocity (range：1~4M PPS)
	RV:	max. velocity (range：1~4M PPS)
	RA:	acceleration
	RD:	deceleration

Return: SUCCESS_NO_ERROR

INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis1* or *axis2* parameter
INVALID_PULSE_ERROR: invalid *LP, WP* or *RP* parameter
INAVLID_ACCELERATION_ERROR: invalid R*A or RD* parameter
INVALID_SPEED_ERROR: invalid *RSV* or *RV* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid
MOTION_STATUS_ERROR: some error occurred in internal Motion-ASIC, please call PS400_GET_ERROR_CODE() for detailed information

Sample:
```
BYTE cardNo=1; // Set card 1.
Long sv=1000; // Set start velocity 1000 PPS.
Long v=10000; // Set max. velocity 10000 PPS.
Long a=5000; // Set acceleration 5000 PPS/s.
long d=5000; //Set deceleration 5000 PPS/s.
PS400_Set_MaxSpeed(cardNo, AxIS_XYZU, 200000)
// Set card1, 4 axes with max. speed  200K PPS.
PS400_Rectangle(
cardNo, AXIS_X, AXIS_Y, 1, 0, 0, 20000, 10000, 1000, sv, v, a, d);
//Set card 1, 2-axis rectangle interpolation with auto deceleration.
```



7.8.2 2-axis linear interpolation

- **short PS400_Set_Line2(BYTE *cardNo*, WORD *axis1*, WORD *axis2*, DWORD *SV* , DWORD *V* , DWORD *A*)**

Func.: **Set 2-axis linear interpolation with symmetric T-curve。**

Para.:

***cardNo*:**	**card ID**
***axis1*:**	**first axis: X、Y、Z、U (1、2、4、8)**
***axis2*:**	**second axis: X、Y、Z、U (1、2、4、8)**
***SV*:**	**start velocity (PPS) (range：1~4M PPS)**
***V*:**	**Max. velocity (PPS) (range：1~4M PPS)**
***A*:**	**Acceleration (PPS/Sec)**

Return: SUCCESS_NO_ERROR

INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis1* or *axis2* parameter
INAVLID_ACCELERATION_ERROR: invalid *A* parameter
INVALID_SPEED_ERROR: invalid *SV* or *V* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid

Sample:

```
BYTE cardNo=1; // Set card 1.
long sv=300; // set start velocity as 300 PPS.
long v=18000; // set max. velocity as 18000 PPS.
long a=500000; // set acc. As 500000 PPS/s.
long loop1;
PS400_Set_MaxSpeed(cardNo, AxIS_XYZU, 200000)
// set card1, 4 axes with max. velocity as 200K PPS.
PS400_Set_Line2(cardNo, AXIS_X, AXIS_Y, sv, v, a);
for (loop1 = 0; loop1 < 10000; loop1++)
{
    PS400_ Line2_Start (cardNo, 0, 100, 100);
    PS400_ Line2_Start (cardNo, 0, -100, -100);
}
PS400_ Line2_Start (cardNo, 1, 100, 100);
// set card 1, X and Y axes linear interpolation
```

- **short PS400_Line2_Start(BYTE *cardNo*, BYTE *Mode*, long *FP1*, long *FP2*)**

Func.: **Start 2-axis linear interpolation in the same card。**

Para.:

***cardNo*:**	**card ID**
***Mode*:**	**0 → 2 axis linear interpolation**
	1 → 2 axis linear interpolation end
***FP1*:**	**first axis Pulse(-2,147,483,648 ~ +2,147,483,647)**
***FP2*:**	**second axis Pulse(-2,147,483,648 ~ +2,147,483,647)**

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
CARD_NUMBER_ERROR: the *cardNo* is invalid
MOTION_STATUS_ERROR: some error occurred in internal Motion-ASIC, please call PS400_GET_ERROR_CODE() for detailed information

Sample:

```
BYTE cardNo=1; // set card 1
long sv=300; // set start velocity as 300 PPS
long v=18000; // set max. velocity as 18000 PPS
long a=500000; // set acc. As 500000 PPS/s
```

```
long loop1;
PS400_Set_MaxSpeed(1, AxIS_XYZU, 200000)
// set card1, 4 axes max. velocity as 200K PPS
PS400_Set_Line2(cardNo, AXIS_X, AXIS_Y, sv, v, a);
for (loop1 = 0; loop1 < 10000; loop1++)
{
    PS400_ Line2_Start (cardNo, 0, 100, 100);
    PS400_ Line2_Start (cardNo, 0, -100, -100);
}
PS400_ Line2_Start (cardNo, 1, 100, 100);
//set card1, X and Y axes linear interpolation.
```

7.8.3 3-axis linear interpolation

- **short PS400_Set_Line3(BYTE *cardNo*, WORD *axis1*, WORD *axis2*, WORD *axis3*, DWORD *SV* , DWORD *V* , DWORD *A*)**

Func.: Set 3-axis linear interpolationo in the same card

Para.:

cardNo:	card ID
axis1:	first axis no. : X、Y、Z、U (1、2、4、8)
axis2:	second axis no. : X、Y、Z、U (1、2、4、8)
axis3:	third axis no. : X、Y、Z、U (1、2、4、8)
SV:	start velocity (PPS) (range：1~4M PPS)
V:	max. velocity (PPS) (range：1~4M PPS)
A:	acceleration (PPS/Sec)

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis1, axis2* or *axis3* parameter
INAVLID_ACCELERATION_ERROR: invalid *A* parameter
INVALID_SPEED_ERROR: invalid *SV* or *V* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid
MOTION_STATUS_ERROR: some error occurred in internal Motion-ASIC, please call PS400_GET_ERROR_CODE() for detailed information

Sample:

```
BYTE cardNo=1; // set card 1
long sv=300; // set start velocity as 300 PPS
long v=18000; // set max. velocity as 18000 PPS
long a=500000; // set acc. As 500000 PPS/s
long loop1;
PS400_Set_Line3(cardNo, AXIS_X, AXIS_Y, AXIS_Z, sv, v, a);
for (loop1 = 0; loop1 < 10000; loop1++)
{
```

```
        PS400_ Line 3_Start(cardNo, 0, 100, 100, 100);
        PS400_ Line3_Satrt(cardNo, 0, -100, -100, -100);
    }
    PS400_ Line3_Start(cardNo, 1, 100, 100, 100);
    // set card1 , X、Y、Z   3-axis linear interpolation
```

- **short PS400_Line3_Start(BYTE *cardNo*, BYTE *Mode*, long *FP1*, long *FP2*, long *FP3*)**

Func.: start 3-axis linear interpolation

Para.:

cardNo:	card ID
Mode:	0 → 3-axis linear interpolation
	1 → 3-axis linear interpolation end
FP1:	first axis Pulse(-2,147,483,648 ~ +2,147,483,647)
FP2:	second Pulse(-2,147,483,648 ~ +2,147,483,647)
FP3:	third axis Pulse (-2,147,483,648 ~ +2,147,483,647)

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
CARD_NUMBER_ERROR: the *cardNo* is invalid
MOTION_STATUS_ERROR: some error occurred in internal Motion-ASIC, please call PS400_GET_ERROR_CODE() for detailed information

Sample:

```
BYTE cardNo=1; //set card 1
long sv=300; //set start velocity as 300 PPS
long v=18000; //set max. velocity as 18000 PPS
long a=500000; //set acc. As 500000 PPS/s
long loop1;
PS400_Set_MaxSpeed(cardNo, AxIS_XYZU, 200000)
// set card1, 4 axes max. velocity as 200K PPS
PS400_Set_Line3(cardNo, AXIS_X, AXIS_Y, AXIS_Z, sv, v, a);
for (loop1 = 0; loop1 < 10000; loop1++)
{
    PS400_ Line 3_Start(cardNo, 0, 100, 100, 100);
    PS400_ Line3_Satrt(cardNo, 0, -100, -100, -100);
}
PS400_ Line3_Start(cardNo, 1, 100, 100, 100);
//set card1, X, Y and Z 3-axis linear interpolation
```

7.8.4 2-axis mixed interpolation

- **short PS400_Set_Mix2(BYTE *cardNo*, WORD *axis1*, WORD *axis2*, BYTE *Acc*, DWORD *SV* , DWORD *V* , DWORD *A*)**

Func.: 2-axis mixed interpolation in the same card。

Para.:

cardNo:	card ID
axis1:	first axis : X、Y、Z、U (1、2、4、8)
axis2:	second axis : X、Y、Z、U (1、2、4、8)
nAcc:	0 → constant velocity (V)
	1 → symmetric T-curve (SV、V、A)
SV:	start velocity (PPS) (range：1~4M PPS)
V:	max. velocity(PPS) (range：1~4M PPS)
A:	acceleration (PPS/Sec)

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis1* or *axis2* parameter
INAVLID_ACCELERATION_ERROR: invalid *A* parameter
INVALID_SPEED_ERROR: invalid *SV* or *V* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid

Sample:

```
BYTE cardNo=1; //set card1
unsigned short sv=300; //set start velocity as 300 PPS
unsigned short v=18000; //set max. velocity as 18000 PPS
unsigned long a=500000; //set acc. As 500000 PPS/s
unsigned short loop1;
PS400_Set_MaxSpeed(cardNo, AxIS_XYZU, 200000)
// set card1, 4 axes max. velocity as 200K PPS
PS400_Set_Mix2(cardNo, AXIS_X, AXIS_Y, 1, sv, v, a);
for (loop1 = 0; loop1 < 10000; loop1++)
{
    PS400_Mix2_Start (cardNo, 0, 1, 0, 0, 100, 100);
    PS400_Mix2_Start (cardNo, 0, 2, 100, 0, 100, 100);
}
PS400_Mix2_Start (cardNo, 1, 4, 100, 100, 0, 0);
//set card1, X and Y 2-axis linear interpolation
```

- **short PS400_Mix2_Start(BYTE *cardNo*, BYTE *Acc*, BYTE *Mode*, long *CP1,* long *CP2*, long *FP1*, long *FP2*)**

Func.: Start 2-axis mixed interpolation
間 Func.。

Para.:

cardNo:	card ID

Acc: 0 → continuous interpolation
1 → continuous interpolation end

Mode: 1： PS400_Line2( )
2： PS400_Arc2_Move()CW
3： PS400_Arc2_Move()CCW

CP1 first axis arc center (-2,147,483,648 ~ +2,147,483,647)
CP2 second axis arc center(-2,147,483,648 ~ +2,147,483,647)
FP1: first axis Pulse(-2,147,483,648 ~ +2,147,483,647)
FP2: second axis Pulse(-2,147,483,648 ~ +2,147,483,647)

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INAVLID_MODE_ERROR: invalid *Mode* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid
MOTION_STATUS_ERROR: some error occurred in internal Motion-ASIC,
please call PS400_GET_ERROR_CODE() for detailed information

Sample:

```
BYTE cardNo=1; //set card 1
unsigned short sv=300; //set start velocity as 300 PPS
unsigned short v=18000; //set max. velocity as 18000 PPS
unsigned long a=500000; //set acc. As 500000 PPS/s
unsigned short loop1;
PS400_Set_MaxSpeed(cardNo, AxIS_XYZU, 200000)
// set card1, 4 axes max. velocity as 200K PPS
PS400_Set_Mix2(cardNo, AXIS_X, AXIS_Y, 1, sv, v, a);
for (loop1 = 0; loop1 < 10000; loop1++)
{
    PS400_Mix2_Start (cardNo, 0, 1, 0, 0, 100, 100);
    PS400_Mix2_Start (cardNo, 0, 2, 100, 0, 100, 100);
}
PS400_Mix2_Start (cardNo, 1, 4, 100, 100, 0, 0);
//set card 1, X and Y 2-axis linear interpolation
```

7.8.5 Multi-point Interpolation

- short PS400_Muti_Intp_Move(BYTE *cardNo*, WORD *axis1*, WORD *axis2*, WORD *axis3*, BYTE *Acc*, DWORD *SV*, DWORD *V*, DWORD *A*, DWORD *D*, BYTE Mode*[ ]*, long CP*1[ ]*, long CP*2[ ]*, long FP*1[ ]*, long FP*2[ ]*,long FP*3[ ]*)

Func.: Multi-point interpolation in the same card。
Para.: *cardNo*: card ID

axis1: first axis no. : X, Y, Z or U (1、2、4、8)
axis2: second axis no. : X, Y, Z or U (1、2、4、8)
axis3: third axis no. : X, Y, Z or U (1、2、4、8)
Acc: 0 → constant velocity (V)
1 → symmetric T-curve (SV、V、A、D)
SV: start velocity (PPS) (range：1~4M PPS)
V: max. velocity (PPS) (range：1~4M PPS)
A: acceleration(PPS/Sec)
D: deceleration(PPS/Sec)
Mode[ ]: Max. points: 1024 (0 ~ 1023)，with the following mode：

1：PS400_Line2( )

2： PS400_Arc2_Move()CW direction

3： PS400_Arc2_Move()CCW direction

4： PS400_Line3D()

5： interpolation end

CP1[ ]: first axis arc center (-2,147,483,648 ~ +2,147,483,647)
CP2[ ]: second axis arc center(-2,147,483,648 ~ +2,147,483,647)
FP1[ ]: first axis Pulse(-2,147,483,648 ~ +2,147,483,647)

FP2[ ]: second axis Pulse(-2,147,483,648 ~ +2,147,483,647)

FP3[ ]: third axis Pulse(-2,147,483,648 ~ +2,147,483,647)

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis1, axis2* or *axis3* parameter
INAVLID_ACCELERATION_ERROR: invalid *A* or *D* parameter
INVALID_SPEED_ERROR: invalid *SV* or *V* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid
MOTION_STATUS_ERROR: some error occurred in internal Motion-ASIC, please call PS400_GET_ERROR_CODE() for detailed information

Sample:
```
BYTE cardNo=1; //set card 1
long sv=100; //set start velocity as 100 PPS
long v=3000; //set max. velocity as 3000 PPS
long a=2000; //set acc. As  2000 PPS/s
long d=2000; //set deceleration as 2000 PPS/s
BYTE Mode[10]=  {     1,     2,     1,     2,     1,7,0,0,0,0};
long cp1[10]=   {     0, 10000,     0,     0,     0,0,0,0,0,0};
long cp2[10]=   {     0,     0,     0,-10000,     0,0,0,0,0,0};
long fp1[10]=   { 10000, 10000,  1000, 10000,-31000,0,0,0,0,0};
long fp2[10]=   { 10000, 10000,     0,-10000,-10000,0,0,0,0,0};
long fp3[10]=   {     0,     0,     0,     0,     0,0,0,0,0,0};
PS400_Set_MaxSpeed(1, AxIS_XYZU, 200000)
// set card1, 4 axes max. velocity as 200K PPS
PS400_Multi_Intp_Move(
cardNo, AXIS_X, AXIS_Y, 0, 1, sv, v, a, d, Mode,cp1, cp2, fp1, fp2, fp3);
//set card1, multi-point interpolation.
```

7.8.6 3-axis helical interpolation

- **short PS400_Helix3_Move(BYTE *cardNo*, WORD *axis1*, WORD *axis2*, WORD *axis3*, BYTE *Dir*, DWORD *V* , long *CP1*, long *CP2*, long *Cycle*, long *Pitch*)**

Func.: 3-axis helical interpolation

Para.:

cardNo:	card ID
axis1:	first arc axis no. : X, Y, Z or U (1、2、4、8)
axis2:	second arc axis no. : X, Y, Z or U (1、2、4、8)
axis3:	Sync. Axis no.: X, Y, Z or U (1、2、4、8)
Dir:	0 → CW 1 → CCW
V:	velocity (PPS) (range：1~4M PPS)
CP1:	first arc axis center(-2,147,483,648 ~ +2,147,483,647)
CP2:	second ard axis center(-2,147,483,648 ~ +2,147,483,647)
Cycle:	circular cycles
Pitch:	helical pitch (-2,147,483,648 ~ +2,147,483,647)

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis1, axis2* or *axis3* parameter
INVALID_SPEED_ERROR: invalid *V* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid
MOTION_STATUS_ERROR: some error occurred in internal Motion-ASIC, please call PS400_GET_ERROR_CODE() for detailed information

Sample:

```
BYTE cardNo=1; //set card 1
PS400_Set_MaxSpeed(cardNo, AxIS_XYZU, 200000)
// set card1, 4 axes max. velocity as 200K PPS
//=============================================
long v=50000;
//set max. velocity as 50000 PPS.
PS400_HELIX_3D(cardNo, AXIS_Y, AXIS_Z, AXIS_X, 1, v, 0, 1000, 5, -2000);
//set card1, Y and Z circular interpolation, X axis sync.
//=============================================
long v=100000;
//set max. velocity as 100000 PPS
PS400_Helix3_Move(cardNo, AXIS_Y, AXIS_Z, AXIS_U, 1, v, 0, 25000, 50, 3600);
//set card1, Y and Z circular interpolation, X axis sync.
```

7.8.7 2-axis ratio movement

- **short PS400_Set_Ratio2(BYTE *cardNo*, WORD *axis1*, WORD *axis2*, DWORD *SV* , DWORD *V* , DWORD *A*, float R*atio*)**

Func.: 2-axis ratio movement。

Para.:

	cardNo:	card ID
	axis1:	first axis no. : X、Y、Z、U (1、2、4、8)
	axis2:	second axis no. : X、Y、Z、U (1、2、4、8)
	SV:	start velocity (PPS) (range：1~4M PPS)
	V:	max. velocity(PPS) (range：1~4M PPS)
	A:	acceleration(PPS/Sec)，
	Ratio:	2-axis ratio

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
INVALID_AXIS_ERROR: invalid *axis1 or axis2* parameter
INAVLID_ACCELERATION_ERROR: invalid *A* parameter
INVALID_SPEED_ERROR: invalid *SV* or *V* parameter
CARD_NUMBER_ERROR: the *cardNo* is invalid

Sample:

```
BYTE cardNo=1; //set card1
long sv=300; // set start velocity as 300 PPS
long v=18000; //set max. velocity as 18000 PPS
long a=500000; //set acc. As 500000 PPS/s
long loop1;
long loop2;
PS400_Set_MaxSpeed(cardNo, AxIS_XYZU, 200000)
// set card1, 4 axes max. velocity 200K PPS
PS400_Set_Ratio(cardNo, AXIS_U, AXIS_X, sv, v, a, 0.36f);
for (loop2 = 0; loop2 < 5; loop2++)
{
    for (loop1 = 0; loop1 < 5; loop1++)
    {
        PS400_Ratio2_Start(cardNo, 0, 3600, 0);
        PS400_Ratio2_Start(cardNo, 0, 3600, 1);
    }
    PS400_Ratio2_Start(cardNo, 0, 7200, 0);
    PS400_Ratio2_Start(cardNo, 0, 3600, 1);
}
PS400_Ratio2_Start(cardNo, 1, 7200, 0);
//set card1, start U and X 2-axis ratio movement.
```

- **short PS400_Ratio2_Start(BYTE *cardNo*, BYTE *nType*, long *data*, BYTE *nDir*)**

Func.: start 2-axis ratio movement

Para.:

cardNo:	card ID
Mode:	0 → ratio movement
	1 → ratio movement end
Data:	first axis Pulse(-2,147,483,648 ~ +2,147,483,647)
Dir:	second axis direction:
	0 → CW
	1 → CCW

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
CARD_NUMBER_ERROR: the *cardNo* is invalid
MOTION_STATUS_ERROR: some error occurred in internal Motion-ASIC, please call PS400_GET_ERROR_CODE() for detailed information

Sample:

```
BYTE cardNo=1; //set card1
long sv=300; // set start velocity as 300 PPS
long v=18000; //set max. velocity as 18000 PPS
long a=500000; //set acc. As 500000 PPS/s
long loop1;
long loop2;
PS400_Set_MaxSpeed(cardNo, AxIS_XYZU, 200000)
// set card1, 4 axes max. velocity as 200K PPS
PS400_Set_Ratio(cardNo, AXIS_U, AXIS_X, sv, v, a, 0.36f);
for (loop2 = 0; loop2 < 5; loop2++)
{
    for (loop1 = 0; loop1 < 5; loop1++)
    {
        PS400_Ratio2_Start(cardNo, 0, 3600, 0);
        PS400_Ratio2_Start(cardNo, 0, 3600, 1);
    }
    PS400_Ratio2_Start(cardNo, 0, 7200, 0);
    PS400_Ratio2_Start(cardNo, 0, 3600, 1);
}
PS400_Ratio2_Start(cardNo, 1, 7200, 0);
// set card1, starts U and X 2-axis ratio movement
```

7.9 Other functions

7.9.1 Set Axis Hold

- **short PS400_Drv_Hold(BYTE *cardNo*, WORD *axis*)**

Func.: this is used to hold axis movement before multi-axis simultaneous start is used

Para.: *cardNo*: card ID
axis: axis no. (table 3-1)

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
CARD_NUMBER_ERROR: the *cardNo* is invalid

Sample:
```
PS400_Drv_Hold(1, AXIS_XY);
PS400_Set_MaxSpeed(1, AxIS_XYZU, 200000)
// set card1, 4 axes max. velocity as 200K PPS
PS400_T_Move(1, AXIS_X, 500, 50000, 10000, 0, 50000);
PS400_S_Move(1, AXIS_Y, 500, 50000, 15000, 0, 50000);
//set card1  X as T-curve, Y as S-Curve，use PS400_Drv_Satrt() to start these 2 axes
  simulatneously
PS400_Drv_Start(1, AXIS_XY);
```

7.9.2 Set Axis Start

- **short PS400_Drv_Start(BYTE *cardNo*, WORD *axis*)**

Func.: to issue multiple axes simultaneous start movement

Para.: *cardNo*: card ID
axis: axis no. (Table3-1)

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
CARD_NUMBER_ERROR: the *cardNo* is invalid

Sample:
```
/PS400_Drv_Hold(1, AXIS_XY);
PS400_Set_MaxSpeed(1, AxIS_XYZU, 200000)
// set card1, 4 axes max. velocity as 200K PPS
PS400_T_Move(1, AXIS_X, 500, 50000, 10000, 0, 50000);
PS400_S_Move(1, AXIS_Y, 500, 50000, 15000, 0, 50000);
PS400_Drv_Start(1, AXIS_XY);
// set card1, X as T-curve, Y as S-Curve, use PS400_Drv_Satrt() to start these 2 axes
simulatneously
```

7.9.3 Set Axis Stop

- **short PS400_Set_SdStop(BYTE *cardNo*, WORD *axis*)**

Func.: set slow-down to stop the axis。

Para.: *cardNo*: card ID
axis: axis no. (table 3-1)

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
CARD_NUMBER_ERROR: the *cardNo* is invalid

Sample:
```
PS400_Set_SdStop(1, AXIS_XY);
//set card1, X Y axes slow-down to stop
```

- **short PS400_Set_EmgStop(BYTE *cardNo*, WORD *axis*)**

Func.: set EMG to stop the axis。

Para.: *cardNo*: card ID
axis: axis no. (table 3-1)

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
CARD_NUMBER_ERROR: the *cardNo* is invalid

Sample:
```
PS400_Set_EmgStop(1, AXIS_ZU);
//set card1, Z and U axes EMG stop。
```

7.9.4 Clear stop status

- **short PS400_Clear_Stop_Status(BYTE *cardNo*)**

Func.: Clear stop status. This function helps to clear the stop-status that is caused by Motion-Error or calling PS400_Set_SdStop() / PS400_Set_EmgStop(). It's recommended to remove the root-cause of Motion-Error and call this routine to clear the Stop-Status.

Para.: *cardNo*: card ID

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
CARD_NUMBER_ERROR: the *cardNo* is invalid

Sample:
```
PS400_Clear_Stop_Status(1);
//Clear card1 stop status。
```

7.9.5 Motion done check

- **short PS400_Move_Done(BYTE *cardNo*, WORD *axis,* BYTE* *pDone*)**

Func.: to check the motion done status。

Para.: *cardNo*: card ID
axis: axis no. (table 3-1)
pDone: point to the memory of returned status
YES: completed
NO : not completed

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
CARD_NUMBER_ERROR: the *cardNo* is invalid

Sample:
```
BYTE cardNo=1, Done; //set card1
PS400_T_Move(cardNo, AXIS_XYZU, 100, 10000, 5000, 0, 50000);
// XYZU axes move 50000 Pulse

PS400_Motion_Done(cardNo, AXIS_X, &Done);
if ( Done == NO)
{
    // check X axis motion done
}
```

7.9.6 Compare-and-Trigger Configuration

- **short PS400_CmpTrig_Config(BYTE *cardNo*, WORD *axis, BYTE Type, BYTE OutputLogic, BYTE PulseWidth, BYTE bDirection, long Pitch = 0, long* pOffset = NULL, DWORD dwOffLen = 0*)**

Func.: configure the Compare/Trigger functionality.

Para.: *cardNo*: card ID
axis: AXIS_X or AXIS_Y
Type: the operation mode is combined by two set configurations.
Counter for Comparator:
LOGIC_POSITION – using Pulse Command/Logic Position counter
ENCODER_POSITION – using Encoder counter
Compare mode :
CONSTANT_PITCH – constant pitch
VARIABLE_OFFSET – variable offset
OutputLogic: Output signal polarity
DCC_ACTIVE_HIGH: Normal Low, Active High
DCC_ACTIVE_LOW: Normal High, Active Low
PulseWidth: width of trigger signal
DCC_PULSE_WIDTH_100us: 100 micro-second

DCC_PULSE_WIDTH_200u: 200 micro-second
DCC_PULSE_WIDTH_1ms: 1 mini-second
DCC_PULSE_WIDTH_2ms: 2 mini-second
DCC_PULSE_WIDTH_10ms: 10 mini-second
DCC_PULSE_WIDTH_20ms: 20 mini-second

bDirection: motion direction
MOVE_DIRECTION_PLUS: forward moving (+ direction)
MOVE_DIRECTION_MINUS: reversemoving (- direction)

Pitch: the constant pulse number for CONSTANT_PITCH mode

pOffset: the array that stores the variable trigger offsets
(only for VARIABLE_OFFSET mode)

dwOffLen: indicate the number of variable offsets (only for VARIABLE_OFFSET)

Return: SUCCESS_NO_ERROR
INVALID_DEVICE_ERROR: no active device is related to the cardNo
CARD_NUMBER_ERROR: the *cardNo* is invalid
INVALID_MODE_ERROR: invalid *OutputLogic* or *PulseWidth* parameter
INVALID_DIRECTION_ERROR: invalid *bDirection* parameter
INVALID_PULSE_ERROR: invalid *Pitch* parameter
INVALID_SCAN_OFFSET_DATA: the offset in *pOffset* are not absolute increasing/decreasing
INVALID_SCAN_OFFSET_LEN: invalid *dwOffLen* 的 parameter (valid value: 1~2048)
MOTION_STATUS_ERROR: some error occurred in internal Motion-ASIC, please call PS400_GET_ERROR_CODE() for detailed information

Sample:

```
BYTE cardNo=1; //set card1
 short wError;
 long lOffset[2048], curPos;
 int i;

//set Compare-Trigger with X axis Command/Pulse counter,
    Constant-Picth (1000 pulse),
    Output signal polarity: Active-High
    Trigger signal width: 200 micro-second
    Forward motion
wError = PS400_CompTrig_Config(cardNo, AXIS_X, (CONSTANT_PITCH |
          LOGIC_POSITION), DCC_ACTIVE_HIGH, DCC_PULSE_WIDTH_200u,
          MOVE_DIRECTION_PLUS, 1000, NULL, 0);
:
:
//reset Compare-Trigger related configuration of X-axis
PS400_CmpTrig_Reset(cardNo, AXIS_X);

//read current Command/Pulse counter
PS400_Get_Command(cardNo, AXIS_X,&curPos);

//configure the variable offsets
curPos = curPos + 100; // make sure the 1st trigger occurs in next motion
for( i=0; i<100; i++)
    lOffset[i] = curPos + (i*i);
```

```
// set Compare-Trigger with X axis Command/Pulse counter,
    Variable-Offset Trigger (100 offsets)
    Output signal polarity: Active-High
    Trigger signal width: 200 micro-second
Forward motion
wError = PS400_CompTrig_Config(cardNo, AXIS_X, (VARIABLE_OFFSET |
            LOGIC_POSITION), DCC_ACTIVE_HIGH, DCC_PULSE_WIDTH_200u,
            MOVE_DIRECTION_PLUS, 0, lOffset, 100);
:
:
// reset Compare-Trigger related configuration of X-axis
PS400_CmpTrig_Reset(cardNo, AXIS_X);
```

Notice: **1. On terminal board, the X-DCC/Y-DCC (X/Y axis Deviation Counter Clear) and *Driving status* use the same output pin. After calling PS400_CmpTrig_Config() to enable the Compare/Trigger functionality, this output pin will switch to DCC.**

2. The CONSTANT_PITCH mode will change both Logic and Encoder couters to Ring counters automatically.

3. When VARIABLE_OFFSET is set, the two internal comparators are used to chech the Logic/Encoder counter. The compare-condition is updated in ISR (Interrupt Service Routine). Therefore, the other functions that need ISR will be disabled, including::

PS400_Enable_FRnet_Scan(),PS400_Set_INT_Factor(),PS400_Attach_INT()

The limitions of Variable-Offset are:

(i) must occurs in next motion (cannot at the start-point)

(ii) must be absolute increasing/decreasing

(iii) maximum offsets is 2048

7.9.7 Reset the Compare-Trigger configuration

- **short PS400_CmpTrig_Reset(BYTE *cardNo*, WORD *axis)***

Func.: **reset the Compare/Trigger configuration that is set with PS400_CmpTrig_Config.**

Para.: ***cardNo*:** **card ID**
***axis*:** **AXIS_X or AXIS_Y**

Return: **SUCCESS_NO_ERROR**
INVALID_DEVICE_ERROR: no active device is related to the cardNo
CARD_NUMBER_ERROR: the *cardNo* is invalid
MOTION_STATUS_ERROR: some error occurred in internal Motion-ASIC, please call PS400_GET_ERROR_CODE() for detailed information

Appendix：error code table

code	Description
0	SUCCESS_NO_ERROR
-1	GET_CARDNO_ERROR
-2	INITIAL_SETTING_ERROR
-3	NO_CARDS_FOUND
-4	INSTALL_DRIVER_ERROR
-5	CARD_NUMBER_ERROR
-6	MODE_SETTING_ERROR
-7	LOGIC_SETTING_ERROR
-8	MOTION_INT_ERROR
-9	PCI_INT_ERROR
-10	MOTION_STATUS_ERROR
-11	GETDI_SETTING_ERROR
-12	FILE_LOADING_ERROR
-13	CARD_INITIAL_EEROR
-14	IOCTL_FAIL_ERROR
-15	INn_RANGE_ERROR
-16	NO_SUPPORT_ERROR
-17	INVALID_DEVICE_ERROR
-18	INVALID_AXIS_ERROR
-19	INVALID_MODE_ERROR
-20	EVENT_CREATE_ERROR
-21	INVALID_SPEED_ERROR
-22	INVALID_ACCELERATION_ERROR
-23	INVALID_PULSE_ERROR
-24	DEVICE_OCCUPIED_ERROR
-25	CONFIG_NO_MATCH_ERROR
-26	INVALID_DECELERATION_ERROR
-27	INVALID_SCAN_OFFSET_DATA
-28	INVALID_SCAN_OFFSET_LEN
-29	INVALID_DIRECTION_ERROR
-30	INVALID_MAPPED_VIEW