PISO-PS400 RTSS DLL Function Reference

(Version 1.0)

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.

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.

Contents

CHAPTER 1

Introduction

This software package is developed for PISO-PS400 pulse-based motion controller. It includes the RTSS DLL for Ardence® RTX 7.1. This RTSS DLL proves the similar functions with Win32 Library (Version 3.0) .

One unique Card ID will be referred by each function in Library. This Card ID is configured with on-board Dip-Switch, and helps to identify multiple PISO-PS400 cards in your system. In other words, you no longer worry about the order that Operating System scans PISO-PS400 cards; the only thing you must take care is the correct relationship between the terminal-boards and PISO-PS400 cards.

There are samples that are provided for Microsoft® Visual C++ 6.0 to demonstrate the functions of PISO-PS400 RTSS DLL. Some samples need the Generic-Type terminal-board (DN-8468G) to connect the external sensors, output pulse and encoder-input.

This documentation provides the detailed information of RTSS DLL functions, including the function-decelerations, definitions of both parameters and return codes. These functions will be cataloged and described in the following chapters:

- CHAPTER 2 – System Initialization
- CHAPTER 3 – Automatic Home Search
- CHAPTER 4 – Independent Moving Functions
- CHAPTER 5 – Interpolation Moving Functions
- CHAPTER 6 – Other Motion Functions
- CHAPTER 7 – Advanced Motion Configurations
- CHAPTER 8 – Miscellaneous Functions
- CHAPTER 9 – Status
- CHAPTER 10 – FRnet I/O extension

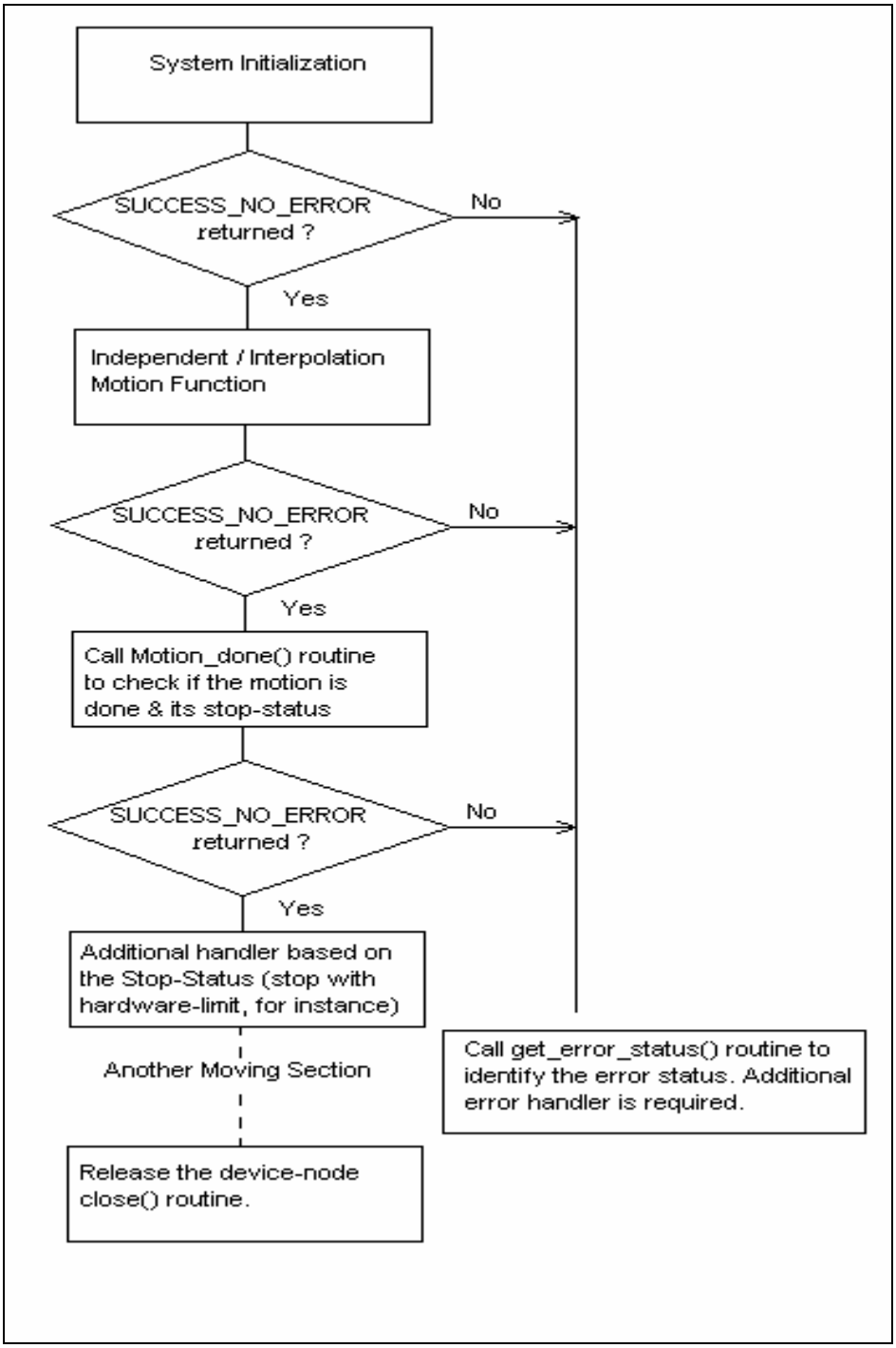

Figure 1 - typical programming following-chart

CHAPTER 2

System Initialization

2.1 Operating-System Configuration

The functions in this chapter provide the interface to Operating-System. By calling these functions, your applications can scan all active PISO-PS400 cards in your system, and get the specific Card-IDs configured with the on-board Dip-Switch. Open the card and access the internal Motion-Control ASIC with the other functions in PISO-PS400 Library.

2.1.1 ps400_scan

short ps400_scan(short* pCardNum, BYTE* pAvailCards = NULL)

Description:

This function scans all active PISO-PS400 cards in your system. The pCardNum saves the numbers of active PISO-PS400 cards. The optional user-provided Array, pAvailCards, indicates the presence of active PISO-PS400 card. (1: present, 0: absent)

Parameters:

pCardNum: The pointer to the memory that stores the numbers of active PISO-PS400 cards.

pAvailCard: The address of user-provided ***BYTE***-Array. Based on the Card ID, each element indicates the presence of active PISO-PS400 card. The user must prepare one ***BYTE***-Array with ***PS400_MaxCards*** elements.

For instance, there are three active PISO-PS400 cards with Card ID 3, 5 and 7. The content of pAvailCard Array will be

{ 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0 }

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_NO_CARD_FOUND: There is no active card available in your system.

ERROR_CARD_ID_DUPLICATED: There are multiple cards are assigned the same Card ID, please check the settings of on-board Dip-Switch.

ERROR_ACCESS_VIOLATION_DATA_COPY: Some system exception occurs while copying memory, please check the pointer-type parameter you assign to this function.

2.1.2 ps400_get_cardinfo

ps400_get_cardinfo(ByVal ScannedIndex As Interger, ByRef pCardID As Byte) As Integer

Description:

This function returns the Card ID based on the scanned-index. This routine will get the Card ID configured with on-board Dip-Switch.

Parameters:

ScannedIndex: The index that the active PISO-PS400 card is scanned. This index begins from 0, and is less than the active PISO-PS400 cards.

pCardID: The pointer to the memory that stores the specific Card ID.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_NO_CARD_FOUND: There is no active card available in your system.

ERROR_INVALID_SCANNED_INDEX: Indicates the ScannedIndex is not less than the numbers of active PISO-PS400 cards.

ERROR_ACCESS_VIOLATION_DATA_COPY: Some system exception occurs while copying memory, please check the pointer-type parameter you assign to this function.

2.1.3 ps400_open

short ps400_open(BYTE bCardID)

Description:

This function opens the device node of PISO-PS400 based on the specific Card ID. If this function returns successfully, the process that calls this function owns the device until ps400_close() is called. The device node of PISO-PS400 is ought to be owned before accessing the Motion-Control ASIC with the other functions. It's recommended to call ps400_scan() and ps400_get_cardinfo() to get the Card ID.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_DEVICE_OPEN: Fail to open the device-node of PISO-PS400. Please make sure no other process owns that PISO-PS400 card.

2.1.4 ps400_close

VC6 / BCB6

short ps400_close(BYTE bCardID)

VB6

ps400_close(ByVal bCardID As Byte) As Integer

Description:

This function closes the device node of PISO-PS400 based on the specific Card ID. After calling this function, the PISO-PS400 card will be released, and other process can open it.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no opened PISO-PS400 card with assigned Card ID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_DEVICE_CLOSE: Fail to close the device-node of PISO-PS400.

2.1.5 ps400_reset

short ps400_reset(BYTE bCardID)

Description:

This function re-sets the internal Motion-Control ASIC and re-configures the basic registers with default value. After calling this function, all configuration set before will be ignored. This function terminates the current motion, too.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no opened PISO-PS400 card with assigned Card ID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_CARD_RESET: Cannot reset the Motion-Control ASIC, please call GetLastError() for further system information.

2.1.6 ps400_open_all

short ps400_open_all(void)

Description:

This function opens the all active PISO-PS400 cards. If this function returns successfully, the process that calls this function owns all devices until ps400_close_all() is called.

Parameters:

None

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_DEVICE_OPEN: Fail to open the device-node of PISO-PS400. Please make sure no other process occupies that PISO-PS400 card.

2.1.7 ps400_close_all

short ps400_close_all(void)

Description:

This function closes all PISO-PS400 cards that are opened by one application. After calling this function, the all PISO-PS400 cards will be released, and other process can open them.
If the RTX process forgets to call ps400_close( ) to release the specific card, this occupied device won't be accessed. The ps400_close_all( ) can be called to release any occupied devices.

Parameters:

None

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.
ERROR_DEVICE_CLOSE: Fail to close the device-node of PISO-PS400.

2.1.8 ps400_reset_all

short ps400_reset_all(void)

Description:

This function re-sets the internal Motion-Control ASIC of all PISO-PS400 cards that are opened by one application, and re-configures the basic registers with default value. After calling this function, all configuration set before will be ignored. This function terminates the current motion of all active PISO-PS400 cards, too.

Parameters:

None

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_CARD_RESET: Cannot reset the Motion-Control ASIC, please call GetLastError() for further system information.

2.2 Hardware Configuration

After the driver is loaded, the pre-defined configurations are assigned to the relative registers of Motion-Control ASIC. The functions in this chapter configure/change the default-settings and polarities of output-pulse, input-encoder and hardware-limit sensors.

2.2.1 ps400_set_pls_cfg

short ps400_set_pls_cfg(BYTE bCardID, WORD wAxis, WORD wPulseMode, WORD wPulseLogic, WORD wDirectionLogic)

Description:

This function configures the output-pulse mode of PISO-PS400.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

wPulseMode: PULSE_MODE_CW_CCW or PULSE_MODE_PULSE_DIRECTION.

wPulseLogic: PULSE_LOGIC_ACTIVE_HIGH or PULSE_LOGIC_ACTIVE_LOW

wDirectionLogic: PULSE_FORWARD_ACTIVE_HIGH or PULSE_FORWARD_ACTIVE_LOW. This parameter will be ignored if the parameter ***wPulseMode*** is assigned to PULSE_MODE_CW_CCW.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_INVALID_PULSE_MODE: Neither PULSE_MODE_CW_CCW nor PULSE_MODE_PULSE_DIRECTION is assigned to parameter ***wPulseMode***.

ERROR_INVALID_PULSE_LEVEL: Neither PULSE_LOGIC_ACTIVE_HIGH nor PULSE_LOGIC_ACTIVE_LOW is assigned to parameter ***wPulseLogic***.

ERROR_INVALID_PULSE_DIRECTION: Neither PULSE_FORWARD_ACTIVE_HIGH nor PULSE_FORWARD_ACTIVE_LOW is assigned to parameter ***wDirectionLogic***.

ERROR_PULSE_MODE_SET: Cannot change the output pulse mode, please call GetLastError() for further system information.

2.2.2 ps400_set_enc_cfg

short ps400_set_enc_cfg(BYTE bCardID, WORD wAxis, WORD wEncoderMode, BYTE bCounterSource = 0)

Description:

This function configures the input-encoder mode of PISO-PS400.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

wEncoderMode: ENCODER_MODE_AB, ENCODER_MODE_AB_DIVID_2, ENCODER_MODE_AB_DIVID_4 or ENCODER_MODE_CW_CCW.

bCounterSource: The optional parameter that is reserved for future.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_INVALID_ENCODER_MODE: No valid encoder mode is assigned to parameter ***wEncoderMode***.

ERROR_ENCODER_MODE_SET: Cannot change the input encoder mode, please call GetLastError() for further system information.

2.2.3 ps400_set_limit

short ps400_set_limit(BYTE bCardID, WORD wAxis, WORD wLimitLogic, WORD wStopMode = LIMIT_STOP_SUDDEN)

Description:

This function configures the polarity and stop-mode of hardware-limit sensor.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

wLimitLogic: LIMIT_LOGIC_ACTIVE_HIGH or LIMIT_LOGIC_ACTIVE_LOW.

wStopMode: LIMIT_STOP_SUDDEN or LIMIT_STOP_SLOWDOWN. This optional parameter is set as LIMIT_STOP_SUDDEN by default.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_INVALID_LIMIT_LOGIC: Neither LIMIT_LOGIC_ACTIVE_HIGH nor LIMIT_LOGIC_ACTIVE_LOW is assigned to parameter ***wLimitLogic***.

ERROR_INVALID_STOP_MODE: Neither LIMIT_STOP_SUDDEN nor LIMIT_STOP_SLOWDOWN is assigned to parameter ***wStopMode***.

ERROR_LIMIT_SENSOR_SET: Cannot configure the hardware-limit sensor, please call GetLastError() for further system information.

2.3 Hardware Configuration (optional)

The functions in this chapter enable/disable the additional signals, including INP and ALARM. The signal-filtering feature is built in Motion-Control ASIC, and can be configured with ps400_set_filter(). The ***Range*** register of Motion-Control ASIC can be configured with ps400_set_range() function. Assigning different value to the ***Range*** register, the accuracy and valid-range of speed, acceleration/deceleration and jerk/deceleration-rate will be changed.
The Motion-Control ASIC also provides the software-limit feature, and is enabled/configured with ps400_set_softlimit(). Another helpful function, ps400_load_config(), configures all PISO-PS400 cards with the pre-defined configuration file, PS400_Config.ini.

2.3.1 ps400_set_range

short ps400_set_range(BYTE bCardID, WORD wAxis, DWORD dwRange)

Description:

This function changes the ***Range*** register to change the accuracy and valid-range of speed, acceleration/deceleration and jerk/deceleration-rate. The relationship between ***Range*** register and Speed/Acceleration/Jerk is illustrated in Figure 2. Another function, ps400_get_range_settings(), gets the current valid-range of speed, acceleration/deceleration and jerk/deceleration-rate.

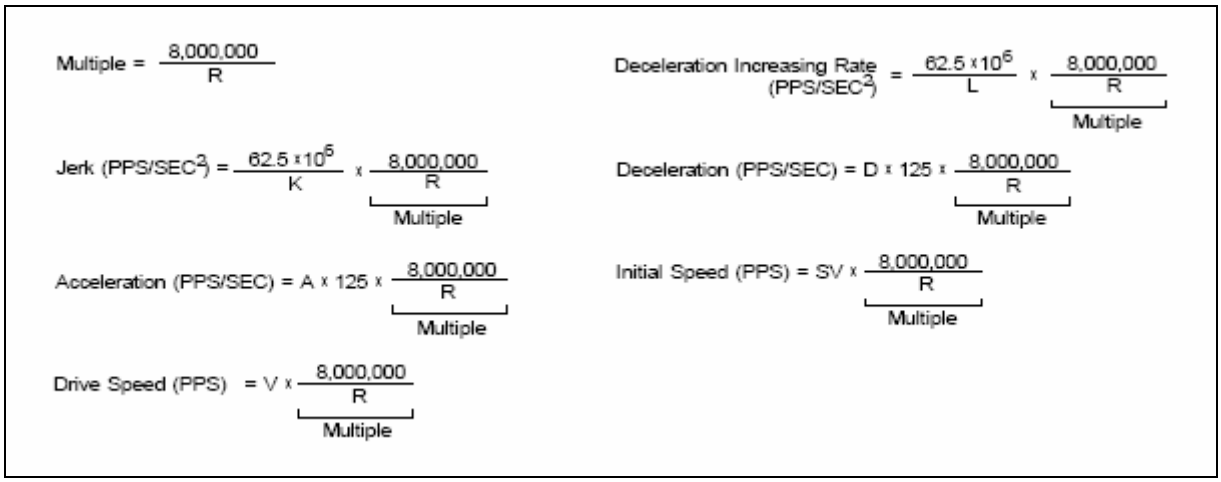

Figure 2 – relationship between ***Range*** register and Speed/Acceleration/Jerk

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

dwRange: The value to be assigned to the ***Range*** register (16,000 ~ 8,000,000)

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_INVALID_RANGE: The value to be assigned to ***Range*** register is invalid.

ERROR_RANGE_CHANGE: Cannot change the content of ***Range*** register, please call GetLastError() for further system information.

2.3.2 ps400_get_range_settings

short ps400_get_range_settings(BYTE bCardID, WORD wAxis, AXIS_RANGE_SETTINGS* pAxisRangeSetting)

Description:

This function gets the valid-range of Speed, Acceleration/Deceleration and Jerk/Deceleration-Increasing-Rate based on the setting of ***Range*** register. Please refer to the 'Set_Range' sample.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

pAxisRangeSetting: The pointer to the data structure that stores the valid-range of Speed, Acceleration/Deceleration and Jerk/Deceleration-Increasing-Rate.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_ACCESS_VIOLATION_DATA_COPY: Some system exception occurs while copying memory, please check the pointer-type parameter you assign to this function.

2.3.3 ps400_set_inp

short ps400_set_inp(BYTE bCardID, WORD wAxis, WORD wINPEnable, WORD wINPLogic)

Description:

This function enables/disables INP feature and configures its polarity. This feature is active until calling ps400_set_inp() with INP_DISABLE_FEATURE.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

wINPEnable: INP_ENABLE_FEATURE or INP_DISABLE_FEATURE.

wINPLogic: INP_LOGIC_ACTIVE_HIGH or INP_LOGIC_ACTIVE_LOW.

Caveat:

If the incorrect setting is assigned to parameter ***wINPLogic***, the ps400_motion_done() will report MOTION_NOT_DONE always.

It's recommended to run ***PCEzGo.exe*** to check the correct settings.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_INVALID_INP_ENABLE: Neither INP_ENABLE_FEATURE nor INP_DISABLE_FEATURE is assigned to parameter ***wINPEnable***.

ERROR_INVALID_INP_LOGIC_LEVEL: Neither INP_LOGIC_ACTIVE_HIGH nor INP_LOGIC_ACTIVE_LOW is assigned to parameter ***wINPLogic***.

ERROR_INP_SIGNAL_SET: Cannot set the INP configuration, please call GetLastError() for further system information.

2.3.4 ps400_set_alarm

short ps400_set_alarm(BYTE bCardID, WORD wAxis, WORD wAlarmEnable, WORD wAlarmLogic)

Description:

This function enables/disables ALARM feature and configures its polarity. This feature is active until calling ps400_set_alarm() with ALARM_DISABLE_FEATURE.

Parameters:

bCardID: the specific Card ID that is configured with the on-board Dip-Switch.

wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

wAlarmEnable: ALARM_ENABLE_FEATURE or ALARM_DISABLE_FEATURE.

wAlarmLogic: ALARM_LOGIC_ACTIVE_HIGH or ALARM_LOGIC_ACTIVE_LOW.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_INVALID_ALARM_ENABLE: Neither ALARM_ENABLE_FEATURE nor ALARM_DISABLE_FEATURE is assigned to parameter ***wAlarmEnable***.

ERROR_INVALID_ALARM_LOGIC_LEVEL: Neither ALARM_LOGIC_ACTIVE_HIGH nor ALARM_LOGIC_ACTIVE_LOW is assigned to parameter ***wAlarmLogic***.

ERROR_ALARM_SIGNAL_SET: Cannot set the ALARM configuration, please call GetLastError() for further system information.

2.3.5 ps400_set_filter

short ps400_set_filter(BYTE bCardID, WORD wAxis, WORD wFilterEnable, WORD wFilterCfg, WORD wDelayTime)

Description:

This function enables/disables the signal filter built in Motion-Control ASIC, and configures the signal-sources and delay-time. This feature is active until calling ps400_set_filter() with FILTER_ DISABLE_FEATURE.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

wFilterEnable: FILTER_ENABLE_FEATURE or FILTER_DISABLE_FEATURE.

wFilterCfg: The following signal-sources can be combined with OR (|) operator.

FILTER_CFG_EMG_EL_ORG_NORG (for EMG, hardware-limit, Home and Near-Home),
FILTER_CFG_ENCODER_Z_PHASE (for Z-Phase/INDEX),
FILTER_CFG_INP_ALARM (for INP and ALARM),
FILTER_CFG_EXP_EXPLSN (for manual-pulse-generator),
FILTER_CFG_IN3 (for digital-input IN3)

wDelayTime: Can be one of the following delay-time settings (unit: micro-second):

FILTER_DELAY_2us,
FILTER_DELAY_256us,
FILTER_DELAY_512us,
FILTER_DELAY_1024us,
FILTER_DELAY_2048us,
FILTER_DELAY_4096us,
FILTER_DELAY_8192us,
FILTER_DELAY_16384us

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_INVALID_FILTER_ENABLE: Neither FILTER_ENABLE_FEATURE nor FILTER_DISABLE_FEATURE is assigned to parameter ***wFilterEnable***.

ERROR_INVALID_FILTER_CONFIGURATION: Invalid Filter-Source combination is assigned to parameter ***wFilterCfg***.

ERROR_INVALID_FILTER_DELAY_TIME: Invalid delay-time is assigned to parameter ***wDelayTime***.

ERROR_FILTER_SET: Cannot set the Filter configuration, please call GetLastError() for further system information.

2.3.6 ps400_set_softlimit

short ps400_set_softlimit(BYTE bCardID, WORD wAxis, WORD wSWLimitEnable, WORD wCmpSource, long LimitPositive, long LimitNegaitive)

Description:

The internal comparators of Motion-Control ASIC provide the software-limit feature. The Motion-Control ASIC will monitor either Logic-Command counter or Encoder-Position counter. Once the content of these counters exceeds the pre-defined value, the deceleration stop will be started. This feature is active until calling ps400_set_softlimit() with SW_LIMIT_DISABLE_FEATURE.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.
wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.
wSWLimitEnable: SW_LIMIT_ENABLE_FEATURE or SW_LIMIT_DISABLE_FEATURE.
wCmpSource: CMP_SRC_LOGIC_COMMAND or CMP_SRC_ENCODER_POSITION.
LimitPositive: The pre-defined value for the comparator in forward direction.
LimitNegaitive: The pre-defined value for the comparator in reverse direction.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.
ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).
ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.
ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.
ERROR_INVALID_SOFTWARE_LIMIT_ENABLE: Neither SW_LIMIT_ENABLE_FEATURE nor SW_LIMIT_DISABLE_FEATURE is assigned to parameter ***wSWLimitEnable***.
ERROR_INVALID_SOFTWARE_LIMIT_COMPARATOR_SOURCE: Neither CMP_SRC_LOGIC_COMMAND nor CMP_SRC_ENCODER_POSITION is assigned to parameter ***wCmpSource***.
ERROR_CONFLICT_WITH_VRING: Indicates that the ASIC-Internal Comparators are used for Variable-Ring counter. Please disable Variable-Ring counter with ps400_set_vring().
ERROR_CONFLICT_WITH_SYNCH_ACTION: Indicates that the ASIC-Internal Comparators are used as the condition of Synchronous-Action. Please disable synchronous-condition with ps400_set_synch().
ERROR_SW_LIMIT_SET: Cannot configure the software-limit settings, please call GetLastError() for further system information.

2.3.7 ps400_servo_on

short ps400_servo_on(BYTE bCardID, WORD wAxis, BYTE bServoON, BYTE bAutoOFF)

Description:

This function turns on/off the Servo.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

bServoON: SERVO_ON or SERVO_OFF.

bAutoOFF: SERVO_MANUAL_OFF: turn off Servo manually.

SERVO_AUTO_OFF: turn off Servo when the ps400_close() or ps400_close_all() is called.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_INVALID_SERVO_SETTING: Neither SERVO_ON nor SERVO_OFF is assigned to parameter ***bServoON***.

ERROR_SERVO_ON_SET: Cannot set the Servo output, please call GetLastError() for further system information.

CHAPTER 3

Automatic Home Search

With the external Near-Home (NORG), Home (ORG) and Z-Phase/INDEX sensors, the auto-homing feature provided by Motion-Control ASIC will helps to search the Home (ORG) automatically.
The typical Automatic Home Search is illustrated in Figure 3.

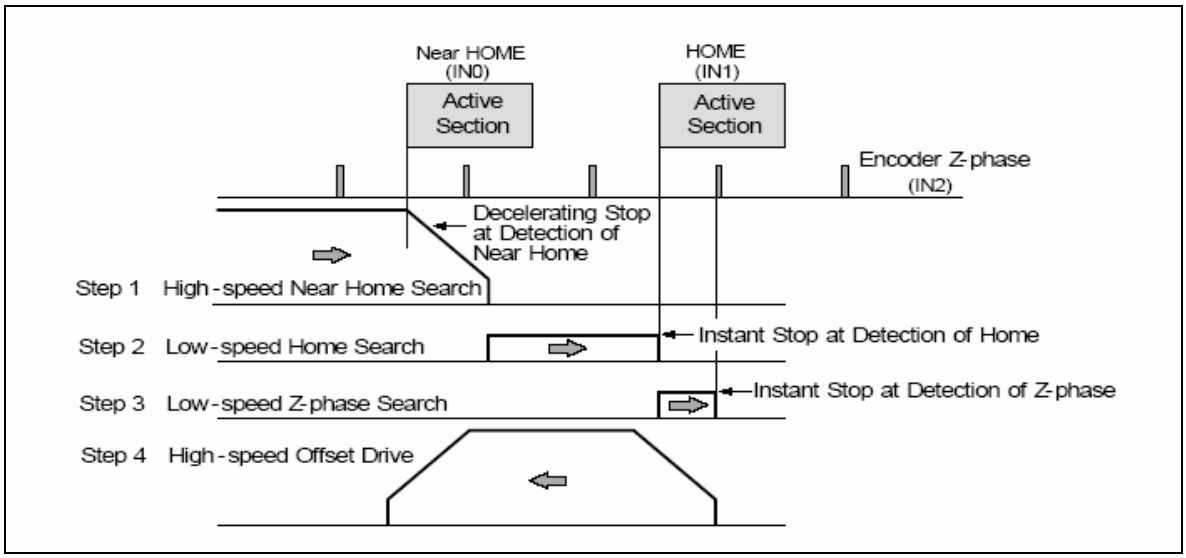

Figure 3 - typical Automatic Home Search

3.1 ps400_set_home_cfg

short ps400_set_home_cfg(BYTE bCardID, WORD wAxis, WORD wHomeLogic, WORD wNHomeLogic, WORD wIndexLogic, WORD wHomeSteps, DWORD dwStep4Offset)

Description:

This function configures the polarities of Near-Home(NORG), Home(ORG) and Z-Phase/INDEX sensors. The searching-steps of Automatic-Home-Search are configured in this function, too.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

wHomeLogic: HOME_LOGIC_ACTIVE_HIGH or HOME_LOGIC_ACTIVE_LOW.

wNHomeLogic: NHOME_LOGIC_ACTIVE_HIGH or NHOME_LOGIC_ACTIVE_LOW

wIndexLogic: INDEX_LOGIC_ACTIVE_HIGH or INDEX_LOGIC_ACTIVE_LOW

wHomeSteps: The combination of Automatic-Home-Search 4-Steps. The configurations are:

Step-1:
AUTO_HOME_STEP1_FORWARD, AUTO_HOME_STEP1_REVERSE and AUTO_HOME_STEP1_DISABLE

Step-2:
AUTO_HOME_STEP2_FORWARD, AUTO_HOME_STEP2_REVERSE and AUTO_HOME_STEP2_DISABLE

Step-3:
AUTO_HOME_STEP3_FORWARD, AUTO_HOME_STEP3_REVERSE and AUTO_HOME_STEP3_DISABLE

Step-4:
AUTO_HOME_STEP4_FORWARD, AUTO_HOME_STEP4_REVERSE and AUTO_HOME_STEP4_DISABLE

Notice: Based on external sensor, It's recommended to include either AUTO_HOME_STEP1_FORWARD / AUTO_HOME_STEP1_REVERSE or AUTO_HOME_STEP2_FORWARD / AUTO_HOME_STEP2_REVERSE in wHomeSteps.

dwStep4Offset: The offset driving in Step-4 of Automatic Home Search.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_INVALID_HOME_LOGIC_LEVEL: Neither HOME_LOGIC_ACTIVE_HIGH nor HOME_LOGIC_ACTIVE_LOW is assigned to parameter ***wHomeLogic***.

ERROR_INVALID_NEAR_HOME_LOGIC_LEVEL: Neither NHOME_LOGIC_ACTIVE_HIGH nor NHOME_LOGIC_ACTIVE_LOW is assigned to parameter ***wNHomeLogic***.

ERROR_INVALID_INDEX_LOGIC_LEVEL: Neither INDEX_LOGIC_ACTIVE_HIGH nor INDEX_LOGIC_ACTIVE_LOW is assigned to parameter ***wIndexLogic***.

ERROR_INVALID_AUTO_HOME_STEP: The Automatic-Home-Search Steps are out of pre-defined configurations.

ERROR_HOME_CFG_SET: Cannot change the configuration of Automatic-Home-Search, please call GetLastError() for further system information.

3.2 ps400_home_start

short ps400_home_start(BYTE bCardID, WORD wAxis, DWORD dwStartSpeed, DWORD dwAcceleration, DWORD dwDeceleration, DWORD dwNHomeSearchSpeed, DWORD dwHomeSearchSpeed, WORD wSyncMode)

Description:

This function starts Automatic-Home-Search with the Start-Speed, Acceleration/Deceleration, Near-Home Searching Speed and Home Searching Speed.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

dwStartSpeed: The Start Speed in Step-1 of Automatic-Home-Search.

dwAcceleration: The Acceleration in Step-1 of Automatic-Home-Search motion.

dwDeceleration: The Deceleration in Step-1 of Automatic-Home-Search motion.

dwNHomeSearchSpeed: The Near-Home Search Speed(Driving Speed) in Step-1 of Automatic-Home-Search motion.

dwHomeSearchSpeed: The Home Search Speed in Step-2 of Automatic-Home-Search motion. This speed is recommended to be lower than dwStartSpeed.

wSyncMode: In RTX RTSS DLL, only DISABLE_BLOCK_OPEARTION is supported.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_INVALID_BLOCK_OPEARTION_MODE: Neither DISABLE_BLOCK_OPEARTION nor ENABLE_BLOCK_OPEARTION is assigned to parameter ***wSyncMode***.

ERROR_INVALID_HOME_SEARCH_SPEED: The value assigned to parameter ***dwHomeSearchSpeed*** is out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_START_SPEED_EXCEED_DRIVING_SPEED: The ***dwStartSpeed*** is larger than ***dwNHomeSearchSpeed***.

ERROR_INVALID_START_SPEED: The value assigned to parameter ***dwStartSpeed*** is out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_DRIVING_SPEED: The value assigned to parameter ***dwNHomeSearchSpeed*** is out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_ACCELERATION: The value assigned to parameter ***dwAcceleration*** is out of range of Acceleration. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_DECELERATION: The value assigned to parameter ***dwDeceleration*** is out of range of Deceleration. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_CONFIG_IS_NEEDED: The Automatic-Home-Search had not been configured. Please configure the Automatic-Home-Search with ps400_set_home_cfg() first.

ERROR_OCCURS_IN_AXIS_X, ERROR_OCCURS_IN_AXIS_Y, ERROR_OCCURS_IN_AXIS_Z ERROR_OCCURS_IN_AXIS_U:

Indicates that some error happens to AXIS_X, AXIS_Y, AXIS_Z or AXIS_U. Please call ps400_get_error_status() for detailed information.

ERROR_MOTION_NOT_COMPLETE: Indicates the previous motion is not completed. Please wait for completion of motion, or stop motion with ps400_stop_move().

ERROR_CONFLICT_WITH_MPG: Indicates the Automatic-Home-Search cannot support Manual-Pulse-Generator, please call ps400_set_mpg() to disable MPG mode.

ERROR_RTX_UNSUPPORT_MODE: Indicates some specific mode is not supported in RTX RTSS DLL.

ERROR_START_HOME: Cannot start Automatic-Home-Search, please call GetLastError() for further system information.

CHAPTER 4

Independent Moving Functions

These functions in this chapter start the independent motion, including velocity-move, constant-speed move, trapezoidal-profile and S-curve move.

4.1 ps400_velocity_move

short ps400_velocity_move(BYTE bCardID, WORD wAxis, DWORD dwStartSpeed, DWORD dwDriveSpeed, DWORD dwAcceleration, BYTE bDirection)

Description:

This function starts velocity-move with ***dwDriveSpeed*** driving-speed continuously. The trapezoidal-profile moving will be applied to Acceleration. Calling ps400_stop_move() to terminate the velocity-move.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.
wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.
dwStartSpeed: The Start-Speed in trapezoidal-profile move.
dwDriveSpeed: The Drive-Speed in trapezoidal-profile move.
dwAcceleration: The Acceleration in trapezoidal-profile move.
bDirection: MOVE_DIRECTION_FORWARD or MOVE_DIRECTION_REVERSE.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.
ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).
ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.
ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_INVALID_MOVE_DIRECTION: Neither MOVE_DIRECTION_FORWARD nor MOVE_DIRECTION_REVERSE is assigned to parameter ***bDirection***.

ERROR_START_SPEED_EXCEED_DRIVING_SPEED: The ***dwStartSpeed*** is larger than ***dwDriveSpeed***.

ERROR_INVALID_START_SPEED: The value assigned to parameter ***dwStartSpeed*** is out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_DRIVING_SPEED: The value assigned to parameter ***dwDriveSpeed*** is out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_ACCELERATION: The value assigned to parameter ***dwAcceleration*** is out of range of Acceleration. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_OCCURS_IN_AXIS_X, ERROR_OCCURS_IN_AXIS_Y, ERROR_OCCURS_IN_AXIS_Z ERROR_OCCURS_IN_AXIS_U:

Indicates that some error happens to AXIS_X, AXIS_Y, AXIS_Z or AXIS_U. Please call ps400_get_error_status() for detailed information.

ERROR_MOTION_NOT_COMPLETE: Indicates the previous motion is not completed.

ERROR_CONFLICT_WITH_MPG: Indicates the velocity-move cannot support Manual-Pulse-Generator, please call ps400_set_mpg() to disable MPG mode.

ERROR_CONTI_MOVE_START: Cannot start velocity-move, please call GetLastError() for further system information.

4.2 ps400_const_move

short ps400_const_move(BYTE bCardID, WORD wAxis, DWORD dwDriveSpeed, long FixedPulse)

Description:

This function starts constant-speed, fixed-pulse motion. No acceleration/deceleration is applied in this motion.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

dwDriveSpeed: The Drive-Speed in constant-speed moving.

FixedPulse: The total numbers of output pulse. This parameter is a signed 32-bits variable, the negative value indicates motion in reverse-direction

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_INVALID_DRIVING_SPEED: The value assigned to parameter ***dwDriveSpeed*** is out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_OCCURS_IN_AXIS_X, ERROR_OCCURS_IN_AXIS_Y, ERROR_OCCURS_IN_AXIS_Z ERROR_OCCURS_IN_AXIS_U:

Indicates that some error happens to AXIS_X, AXIS_Y, AXIS_Z or AXIS_U. Please call ps400_get_error_status() for detailed information.

ERROR_MOTION_NOT_COMPLETE: Indicates the previous motion is not completed.

ERROR_CONFLICT_WITH_MPG: Indicates the velocity-move cannot support Manual-Pulse-Generator, please call ps400_set_mpg() to disable MPG mode.

ERROR_CONST_MOVE_START: Cannot start constant-speed motion, please call GetLastError() for further system information.

4.3 ps400_const_moveall

short ps400_const_moveall(BYTE bCardID, WORD wAxis, DWORD dwDriveSpeed[], long FixedPulse[])

Description:

This function starts multiple axes the constant-speed, fixed-pulse moving simultaneously. No acceleration/deceleration is applied in this motion.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxes: Can be any combination of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

dwDriveSpeed[]: The pointer to the ***DWORD***-Array that contains Driving-Speed of multiple axes. Different speed can be assigned to each axis.

FixedPulse[]: The pointer to the ***long***-Array that contains Fixed-Pulse of multiple axes. Different pulse number can be assigned to each axis. The element of this ***long***-Array is a signed 32-bits variable, the negative value indicates motion in reverse-direction

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_INVALID_DRIVING_SPEED: The value assigned to some elements of ***dwDriveSpeed[]*** are out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_ACCESS_VIOLATION_DATA_COPY: Some system exception occurs while copying memory, please check the pointer-type parameter you assign to this function.

ERROR_OCCURS_IN_AXIS_X, ERROR_OCCURS_IN_AXIS_Y, ERROR_OCCURS_IN_AXIS_Z, ERROR_OCCURS_IN_AXIS_U, ERROR_OCCURS_IN_AXIS_XY, ERROR_OCCURS_IN_AXIS_XZ, ERROR_OCCURS_IN_AXIS_YZ, ERROR_OCCURS_IN_AXIS_XU, ERROR_OCCURS_IN_AXIS_YU, ERROR_OCCURS_IN_AXIS_ZU, ERROR_OCCURS_IN_AXIS_XYZ, ERROR_OCCURS_IN_AXIS_XYU, ERROR_OCCURS_IN_AXIS_XZU, ERROR_OCCURS_IN_AXIS_YZU, ERROR_OCCURS_IN_AXIS_XYZU:
Indicates that some error happens to AXIS_xxxx. Please call ps400_get_error_status() for detailed information.

ERROR_MOTION_NOT_COMPLETE: Indicates the previous motion is not completed.

ERROR_AXES_MOVE_CHECK: Cannot forward the Axes-checking command to system, please call GetLastError() for further system information.

ERROR_HOLD_AXES_NOT_RELEASE: Indicates the hold-axes had not been release, please call ps400_drv_start() to release the hold-axes.

ERROR_CONST_MOVE_START: Cannot start constant-speed motion, please call GetLastError() for further system information.

4.4 ps400_t_move

short ps400_t_move(BYTE bCardID, WORD wAxis, DWORD dwStartSpeed, DWORD dwDriveSpeed, DWORD dwAcceleration, DWORD dwDeceleration, long FixedPulse, short wAccCntOffset = 8)

Description:

This function starts trapezoidal-profile, fixed-pulse motion.

Parameters:

bCardID: the specific Card ID that is configured with the on-board Dip-Switch.

wAxis: can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

dwStartSpeed: The Start-Speed in trapezoidal-profile moving.

dwDriveSpeed: The Drive-Speed in trapezoidal-profile moving.

dwAcceleration: The Acceleration in trapezoidal-profile moving.

dwDeceleration: The Deceleration in trapezoidal-profile moving.

FixedPulse: The total numbers of output pulse. This parameter is a signed 32-bits variable, the negative value indicates motion in reverse-direction

wAccCntOffset: This optional parameter to configure the offset for Acceleration/Deceleration driving. The default setting of ***wAccCntOffset*** is 8.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_START_SPEED_EXCEED_DRIVING_SPEED: The ***dwStartSpeed*** is larger than ***dwDriveSpeed***.

ERROR_INVALID_START_SPEED: The value assigned to parameter ***dwStartSpeed*** is out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_DRIVING_SPEED: The value assigned to parameter ***dwDriveSpeed*** is out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_ACCELERATION: The value assigned to parameter ***dwAcceleration*** is out of range of Acceleration. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_DECELERATION: The value assigned to parameter ***dwDeceleration*** is out of range of Deceleration. Please refer to ps400_set_range() and ps400_get_range_settings().

Notice: In the case, ***dwAcceleration*** > ***dwDeceleration*** , the following formula should be satisfied, too.

dwDeceleration > (dwAcceleration x dwDriveSpeed) / 4,000,000.

ERROR_OCCURS_IN_AXIS_X, ERROR_OCCURS_IN_AXIS_Y, ERROR_OCCURS_IN_AXIS_Z ERROR_OCCURS_IN_AXIS_U:

Indicates that some error happens to AXIS_X, AXIS_Y, AXIS_Z or AXIS_U. Please call ps400_get_error_status() for detailed information.

ERROR_MOTION_NOT_COMPLETE: Indicates the previous motion is not completed.

ERROR_CONFLICT_WITH_MPG: Indicates the trapezoidal-move cannot support Manual-Pulse-Generator, please call ps400_set_mpg() to disable MPG mode.

ERROR_T_MOVE_START: Cannot start trapezoidal moving, please call GetLastError() for further system information.

4.5 ps400_t_moveall

short ps400_t_moveall(BYTE bCardID, WORD wAxes, DWORD dwStartSpeed[], DWORD dwDriveSpeed[], DWORD dwAcceleration[], DWORD dwDeceleration[], long FixedPulse[], short wAccCntOffset[] = NULL)

Description:

This function starts multiple axes the trapezoidal-profile, fixed-pulse moving simultaneously.

Parameters:

bCardID: the specific Card ID that is configured with the on-board Dip-Switch.

wAxes: can be any combination of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

dwStartSpeed[]: The pointer to the ***DWORD***-Array that contains Start-Speed of multiple axes. Different value can be assigned to each axis.

dwDriveSpeed[]: The pointer to the ***DWORD***-Array that contains Driving-Speed of multiple axes. Different value can be assigned to each axis.

dwAcceleration[]: The pointer to the ***DWORD***-Array that contains Acceleration of multiple axes. Different value can be assigned to each axis.

dwDeceleration[]: The pointer to the ***DWORD***-Array that contains Deceleration of multiple axes. Different value can be assigned to each axis.

FixedPulse[]: The pointer to the ***long***-Array that contains Fixed-Pulse of multiple axes. Different pulse number can be assigned to each axis. The element of this parameter is a signed 32-bits variable, the negative value indicates motion in reverse-direction

wAccCntOffset[]: This optional pointer to the ***short***-Array that contains offset of multiple axes. The default setting of ***wAccCntOffset[]*** is NULL.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_START_SPEED_EXCEED_DRIVING_SPEED: Some elements in ***dwStartSpeed[]*** are larger than relative element of ***dwDriveSpeed[]***.

ERROR_INVALID_START_SPEED: The value assigned to some elements of ***dwStartSpeed[]*** are out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_DRIVING_SPEED: The value assigned to some elements of ***dwDriveSpeed[]*** are out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_ACCELERATION: The value assigned to some elements of ***dwAcceleration[]*** are out of range of Acceleration. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_DECELERATION: The value assigned to some elements of ***dwDeceleration[]*** are out of range of Deceleration. Please refer to ps400_set_range() and ps400_get_range_settings().

Notice: In the case, ***dwAcceleration*** > ***dwDeceleration*** , the following formula should be satisfied, too.

dwDeceleration > (dwAcceleration x dwDriveSpeed) / 4,000,000.

ERROR_ACCESS_VIOLATION_DATA_COPY: Some system exception occurs while copying memory, please check the pointer-type parameter you assign to this function.

ERROR_OCCURS_IN_AXIS_X, ERROR_OCCURS_IN_AXIS_Y, ERROR_OCCURS_IN_AXIS_Z, ERROR_OCCURS_IN_AXIS_U, ERROR_OCCURS_IN_AXIS_XY, ERROR_OCCURS_IN_AXIS_XZ, ERROR_OCCURS_IN_AXIS_YZ, ERROR_OCCURS_IN_AXIS_XU, ERROR_OCCURS_IN_AXIS_YU, ERROR_OCCURS_IN_AXIS_ZU, ERROR_OCCURS_IN_AXIS_XYZ, ERROR_OCCURS_IN_AXIS_XYU, ERROR_OCCURS_IN_AXIS_XZU, ERROR_OCCURS_IN_AXIS_YZU, ERROR_OCCURS_IN_AXIS_XYZU:

Indicates that some error happens to AXIS_xxxx. Please call ps400_get_error_status() for detailed information.

ERROR_MOTION_NOT_COMPLETE: Indicates the previous motion is not completed.

ERROR_AXES_MOVE_CHECK: Cannot forward the Axes-checking command to system, please call GetLastError() for further system information.

ERROR_HOLD_AXES_NOT_RELEASE: Indicates the hold-axes had not been release, please call ps400_drv_start() to release the hold-axes.

ERROR_T_MOVE_START: Cannot start trapezoidal moving, please call GetLastError() for further system information.

4.6 ps400_s_move

short ps400_t_move(BYTE bCardID, WORD wAxis, DWORD dwStartSpeed, DWORD dwDriveSpeed, DWORD dwAccelerationRate, DWORD DWORD dwDecelerationRate, long FixedPulse, short wAccCntOffset = 8)

Description:

This function starts S-curve, fixed-pulse motion.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

dwStartSpeed: The Start-Speed in S-curve moving.

dwDriveSpeed: The Drive-Speed in S-curve moving.

dwAccelerationRate: The Acceleration-Increasing-Rate in S-curve moving. The Acceleration will be assigned to maximum value automatically.

dwDecelerationRate: The Deceleration-Increasing-Rate in S-curve moving. The Deceleration will be assigned to maximum value automatically.

FixedPulse: This parameter is a signed 32-bits variable, the negative value indicates motion in reverse-direction

wAccCntOffset: This optional parameter to configure the offset for Acceleration/Deceleration driving. The default setting of ***wAccCntOffset*** is 8.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_START_SPEED_EXCEED_DRIVING_SPEED: The ***dwStartSpeed*** is larger than or equal to ***dwDriveSpeed***.

ERROR_INVALID_START_SPEED: The value assigned to parameter ***dwStartSpeed*** is out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_DRIVING_SPEED: The value assigned to parameter ***dwDriveSpeed*** is out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_JERK: The value assigned to parameter ***dwAccelerationRate*** is out of range of Acceleration-Increasing-Rate. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_DECELERATION_RATE: The value assigned to parameter ***dwDecelerationRate*** is out of range of Deceleration-Increasing-Rate. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_OCCURS_IN_AXIS_X, ERROR_OCCURS_IN_AXIS_Y, ERROR_OCCURS_IN_AXIS_Z ERROR_OCCURS_IN_AXIS_U:

Indicates that some error happens to AXIS_X, AXIS_Y, AXIS_Z or AXIS_U. Please call ps400_get_error_status() for detailed information.

ERROR_MOTION_NOT_COMPLETE: Indicates the previous motion is not completed.

ERROR_CONFLICT_WITH_MPG: Indicates the S-curvel moving cannot support Manual-Pulse-Generator, please call ps400_set_mpg() to disable MPG mode.

ERROR_S_MOVE_START: Cannot start S-curvel moving, please call GetLastError() for further system information.

4.7 ps400_s_moveall

short ps400_s_moveall(BYTE bCardID, WORD wAxes, DWORD dwStartSpeed[], DWORD dwDriveSpeed[], DWORD dwAccelerationRate[], DWORD dwDecelerationRate[], long FixedPulse[], short wAccCntOffset[] = NULL)

Description:

This function starts multiple axes the S-curve, fixed-pulse moving simultaneously.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxes: Can be any combination of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

dwStartSpeed[]: The pointer to the ***DWORD***-Array that contains Start-Speed of multiple axes. Different value can be assigned to each axis.

dwDriveSpeed[]: The pointer to the ***DWORD***-Array that contains Driving-Speed of multiple axes. Different value can be assigned to each axis.

dwAccelerationRate[]: The pointer to the ***DWORD***-Array that contains Acceleration-Increasing-Rate of multiple axes. Different value can be assigned to each axis.

dwDecelerationRate[]: The pointer to the ***DWORD***-Array that contains Deceleration-Increasing-Rate of multiple axes. Different value can be assigned to each axis.

FixedPulse[]: The pointer to the ***long***-Array that contains Fixed-Pulse of multiple axes. Different pulse number can be assigned to each axis. The element of this parameter is a signed 32-bits variable, the negative value indicates motion in reverse-direction

wAccCntOffset[]: This optional pointer to the ***short***-Array that contains offset of multiple axes. The default setting of ***wAccCntOffset[]*** is NULL.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_START_SPEED_EXCEED_DRIVING_SPEED: Some elements in ***dwStartSpeed[]*** are larger than relative element of ***dwDriveSpeed[]***.

ERROR_INVALID_START_SPEED: The value assigned to some elements of ***dwStartSpeed[]*** are out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_DRIVING_SPEED: The value assigned to some elements of ***dwDriveSpeed[]*** are out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_JERK: The value assigned to some elements of ***dwAccelerationRate[]*** are out of range of Acceleration-Increasing-Rate. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_DECELERATION_RATE: The value assigned to some elements of ***dwDecelerationRate[]*** are out of range of Deceleration-Increasing-Rate. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_ACCESS_VIOLATION_DATA_COPY: Some system exception occurs while copying memory, please check the pointer-type parameter you assign to this function.

ERROR_OCCURS_IN_AXIS_X, ERROR_OCCURS_IN_AXIS_Y, ERROR_OCCURS_IN_AXIS_Z, ERROR_OCCURS_IN_AXIS_U, ERROR_OCCURS_IN_AXIS_XY, ERROR_OCCURS_IN_AXIS_XZ, ERROR_OCCURS_IN_AXIS_YZ, ERROR_OCCURS_IN_AXIS_XU, ERROR_OCCURS_IN_AXIS_YU, ERROR_OCCURS_IN_AXIS_ZU, ERROR_OCCURS_IN_AXIS_XYZ, ERROR_OCCURS_IN_AXIS_XYU, ERROR_OCCURS_IN_AXIS_XZU, ERROR_OCCURS_IN_AXIS_YZU, ERROR_OCCURS_IN_AXIS_XYZU:

Indicates that some error happens to AXIS_xxxx. Please call ps400_get_error_status() for detailed information.

ERROR_MOTION_NOT_COMPLETE: Indicates the previous motion is not completed.

ERROR_AXES_MOVE_CHECK: Cannot forward the Axes-checking command to system, please call GetLastError() for further system information.

ERROR_HOLD_AXES_NOT_RELEASE: Indicates the hold-axes had not been release, please call ps400_drv_start() to release the hold-axes.

ERROR_S_MOVE_START: Cannot start S-curve moving, please call GetLastError() for further system information.

CHAPTER 5

Interpolation Moving Functions

5.1 Individual Interpolation Moving

The functions in this chapter provide both trapezoidal and S-curve acceleration/deceleration in 2D/3D linear interpolation moving. And only trapezoidal acceleration/deceleration can be applied to circular interpolation moving.

5.1.1 ps400_t_line2_move

short ps400_t_line2_move(BYTE bCardID, WORD wMainAxis, WORD wSlaveAxis, DWORD dwStartSpeed, DWORD dwDriveSpeed, DWORD dwAcceleration, DWORD dwDeceleration, long MainAxisFinishPoint, long SlaveAxisFinishPoint, short wAccCntOffset = 8, WORD wSyncMode)

Description:

This function starts the trapezoidal-profile, 2-dimension linear interpolation moving.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wMainAxis: The main-axis of Interpolation moving, can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

wSlaveAxis: The slave-axis of Interpolation moving, can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U (cannot be the same as ***wMainAxis***).

dwStartSpeed: The Start-Speed in trapezoidal-profile moving. This Start-Speed will be applied to main-axis.

dwDriveSpeed: The Drive-Speed in trapezoidal-profile moving. This Drive-Speed will be applied to main-axis.

dwAcceleration: The Acceleration in trapezoidal-profile moving. This Acceleration will be applied to main-axis.

dwDeceleration: The Deceleration in trapezoidal-profile moving. This Deceleration will be applied to main-axis.

MainAxisFinishPoint: The finish point of main-axis. This parameter is the relative offset to the current position. And the negative value indicates that the finish point is in reverse-direction.

SlaveAxisFinishPoint: The finish point of slave-axis. This parameter is the relative offset to the current position. And the negative value indicates that the finish point is in reverse-direction.

wAccCntOffset: This optional parameter to configure the offset for Acceleration/Deceleration driving. The default setting of ***wAccCntOffset*** is 8.

wSyncMode: In RTX RTSS DLL, only DISABLE_BLOCK_OPEARTION is supported.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wMainAxis*** or ***wSlaveAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wMainAxis*** or ***wSlaveAxis***.

ERROR_INVALID_INTERPOLATION_SLAVE_AXES: The parameter ***wSlaveAxis*** includes the axis ID assigned to ***wMainAxis***.

ERROR_INVALID_BLOCK_OPEARTION_MODE: Neither DISABLE_BLOCK_OPEARTION nor ENABLE_BLOCK_OPEARTION is assigned to parameter ***wSyncMode***.

ERROR_START_SPEED_EXCEED_DRIVING_SPEED: The ***dwStartSpeed*** is larger than ***dwDriveSpeed***.

ERROR_INVALID_START_SPEED: The value assigned to parameter ***dwStartSpeed*** is out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_DRIVING_SPEED: The value assigned to parameter ***dwDriveSpeed*** is out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_ACCELERATION: The value assigned to parameter ***dwAcceleration*** is out of range of Acceleration. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_DECELERATION: The value assigned to parameter ***dwDeceleration*** is out of range of Deceleration. Please refer to ps400_set_range() and ps400_get_range_settings().

Notice: In the case, ***dwAcceleration*** > ***dwDeceleration*** , the following formula should be satisfied, too.

dwDeceleration > (dwAcceleration x dwDriveSpeed) / 4,000,000.

ERROR_INTERPOLATION_NOT_COMPLETE: The interpolation moving started before had not completed.

ERROR_REASSIGN_SYNCH_MODE_COMMAND: The previous Synchronous Operation is not returned.

ERROR_OCCURS_IN_AXIS_X, ERROR_OCCURS_IN_AXIS_Y, ERROR_OCCURS_IN_AXIS_Z, ERROR_OCCURS_IN_AXIS_U, ERROR_OCCURS_IN_AXIS_XY, ERROR_OCCURS_IN_AXIS_XZ, ERROR_OCCURS_IN_AXIS_YZ, ERROR_OCCURS_IN_AXIS_XU, ERROR_OCCURS_IN_AXIS_YU, ERROR_OCCURS_IN_AXIS_ZU:

Indicates that some error happens to AXIS_xxxx. Please call ps400_get_error_status() for detailed information.

ERROR_MOTION_NOT_COMPLETE: Indicates the previous motion is not completed.

ERROR_AXES_MOVE_CHECK: Cannot forward the Axes-checking command to system, please call GetLastError() for further system information.

ERROR_CONFLICT_WITH_MPG: Indicates the trapezoidal 2D interpolation moving cannot support Manual-Pulse-Generator, please call ps400_set_mpg() to disable MPG mode.

ERROR_RTX_UNSUPPORT_MODE: Indicates some specific mode is not supported in RTX RTSS DLL.

ERROR_T_LINE2_START: Cannot start trapezoidal 2D interpolation moving, please call GetLastError() for further system information.

5.1.2 ps400_s_line2_move

short ps400_s_line2_move(BYTE bCardID, WORD wMainAxis, WORD wSlaveAxis, DWORD dwStartSpeed, DWORD dwDriveSpeed, DWORD dwAccelerationRate, DWORD dwDecelerationRate, long MainAxisFinishPoint, long SlaveAxisFinishPoint, short wAccCntOffset = 8, WORD wSyncMode)

Description:

This function starts the S-curve, 2-dimension linear interpolation moving.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wMainAxis: The main-axis of Interpolation moving, can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

wSlaveAxis: The slave-axis of Interpolation moving, can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U (cannot be the same as ***wMainAxis***).

dwStartSpeed: The Start-Speed in S-curve moving. This Start-Speed will be applied to main-axis.

dwDriveSpeed: The Drive-Speed in S-curve moving. This Drive-Speed will be applied to main-axis.

dwAccelerationRate: The Acceleration-Increasing-Rate in S-curve moving. The Acceleration will be assigned to maximum value automatically. This Acceleration-Increasing-Rate will be applied to main-axis.

dwDecelerationRate: The Deceleration-Increasing-Rate in S-curve moving. The Deceleration will be assigned to maximum value automatically. This Acceleration-Increasing-Rate will be applied to main-axis.

MainAxisFinishPoint: The finish point of main-axis. This parameter is the relative offset to the current position. And the negative value indicates that the finish point is in reverse-direction.

SlaveAxisFinishPoint: The finish point of slave-axis. This parameter is the relative offset to the current position. And the negative value indicates that the finish point is in reverse-direction

wAccCntOffset: This optional parameter to configure the offset for Acceleration/Deceleration driving. The default setting of ***wAccCntOffset*** is 8.

wSyncMode: In RTX RTSS DLL, only DISABLE_BLOCK_OPEARTION is supported.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wMainAxis*** or ***wSlaveAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wMainAxis*** or ***wSlaveAxis***.

ERROR_INVALID_INTERPOLATION_SLAVE_AXES: The parameter ***wSlaveAxis*** includes the axis ID assigned to ***wMainAxis***.

ERROR_INVALID_BLOCK_OPEARTION_MODE: Neither DISABLE_BLOCK_OPEARTION nor ENABLE_BLOCK_OPEARTION is assigned to parameter ***wSyncMode***.

ERROR_START_SPEED_EXCEED_DRIVING_SPEED: The ***dwStartSpeed*** is larger than or equal to ***dwDriveSpeed***.

ERROR_INVALID_START_SPEED: The value assigned to parameter ***dwStartSpeed*** is out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_DRIVING_SPEED: The value assigned to parameter ***dwDriveSpeed*** is out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_JERK: The value assigned to parameter ***dwAccelerationRate*** is out of range of Acceleration Increasing Rate. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_DECELERATION_RATE: The value assigned to parameter ***dwDecelerationRate*** is out of range of Deceleration Increasing Rate. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INTERPOLATION_NOT_COMPLETE: The interpolation moving started before had not completed.

ERROR_REASSIGN_SYNCH_MODE_COMMAND: The previous Synchronous Operation is not returned.

ERROR_OCCURS_IN_AXIS_X, ERROR_OCCURS_IN_AXIS_Y, ERROR_OCCURS_IN_AXIS_Z,
ERROR_OCCURS_IN_AXIS_U, ERROR_OCCURS_IN_AXIS_XY,
ERROR_OCCURS_IN_AXIS_XZ, ERROR_OCCURS_IN_AXIS_YZ,
ERROR_OCCURS_IN_AXIS_XU, ERROR_OCCURS_IN_AXIS_YU,
ERROR_OCCURS_IN_AXIS_ZU:

Indicates that some error happens to AXIS_xxxx. Please call ps400_get_error_status() for detailed information.

ERROR_MOTION_NOT_COMPLETE: Indicates the previous motion is not completed.

ERROR_AXES_MOVE_CHECK: Cannot forward the Axes-checking command to system, please call GetLastError() for further system information.

ERROR_CONFLICT_WITH_MPG: Indicates the S-curve 2D interpolation moving cannot support Manual-Pulse-Generator, please call ps400_set_mpg() to disable MPG mode.

ERROR_RTX_UNSUPPORT_MODE: Indicates some specific mode is not supported in RTX RTSS DLL.

ERROR_S_LINE2_START: Cannot start S-curve 2D interpolation moving, please call GetLastError() for further system information.

5.1.3 ps400_t_line3_move

short ps400_t_line3_move(BYTE bCardID, WORD wMainAxis, WORD wSecondAxis, WORD wThirdAxis, DWORD dwStartSpeed, DWORD dwDriveSpeed, DWORD dwAcceleration, DWORD dwDeceleration, long MainAxisFinishPoint, long SecondAxisFinishPoint, long ThirdAxisFinishPoint, short wAccCntOffset = 8, WORD wSyncMode = DISABLE_BLOCK_OPEARTION)

Description:

This function starts the trapezoidal-profile, 3-dimension linear interpolation moving.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wMainAxis: The main-axis of interpolation moving, can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

wSecondAxis: The second-axis of Interpolation moving, can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U (cannot be the same as ***wMainAxis***).

wThirdAxis: The third-axis of Interpolation moving, can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U (neither ***wMainAxis*** nor ***wSecondAxis*** can be assigned to ***wThirdAxis***)

dwStartSpeed: The Start-Speed in trapezoidal-profile moving. This Start-Speed will be applied to main-axis.

dwDriveSpeed: The Drive-Speed in trapezoidal-profile moving. This Drive-Speed will be applied to main-axis.

dwAcceleration: The Acceleration in trapezoidal-profile moving. This Acceleration will be applied to main-axis.

dwDeceleration: The Deceleration in trapezoidal-profile moving. This Deceleration will be applied to main-axis.

MainAxisFinishPoint: The finish point of main-axis. This parameter is the relative offset to the current position. And the negative value indicates that the finish point is in reverse-direction.

SecondAxisFinishPoint: The finish point of second-axis. This parameter is the relative offset to the current position. And the negative value indicates that the finish point is in reverse-direction.

ThirdAxisFinishPoint: The finish point of third-axis. This parameter is the relative offset to the current position. And the negative value indicates that the finish point is in reverse-direction.

wAccCntOffset: This optional parameter to configure the offset for Acceleration/Deceleration driving. The default setting of ***wAccCntOffset*** is 8.

wSyncMode: In RTX RTSS DLL, only DISABLE_BLOCK_OPEARTION is supported.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wMainAxis, wSecondAxis*** or ***wThirdAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wMainAxis, wSecondAxis*** or ***wThirdAxis***.

ERROR_INVALID_INTERPOLATION_SLAVE_AXES: Either ***wSecondAxis*** or ***wThirdAxis*** includes the axis ID assigned to ***wMainAxis***.

ERROR_INTERPOLATION_SLAVE_AXES_DUPLICATED: The axis ID assigned to ***wSecondAxis*** and ***wThirdAxis*** is the same.

ERROR_INVALID_BLOCK_OPEARTION_MODE: Neither DISABLE_BLOCK_OPEARTION nor ENABLE_BLOCK_OPEARTION is assigned to parameter ***wSyncMode***.

ERROR_START_SPEED_EXCEED_DRIVING_SPEED: The ***dwStartSpeed*** is larger than ***dwDriveSpeed***.

ERROR_INVALID_START_SPEED: The value assigned to parameter ***dwStartSpeed*** is out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_DRIVING_SPEED: The value assigned to parameter ***dwDriveSpeed*** is out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_ACCELERATION: The value assigned to parameter ***dwAcceleration*** is out of range of Acceleration. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_DECELERATION: The value assigned to parameter ***dwDeceleration*** is out of range of Deceleration. Please refer to ps400_set_range() and ps400_get_range_settings().

Notice: In the case, ***dwAcceleration*** > ***dwDeceleration*** , the following formula should be satisfied, too.

dwDeceleration > (dwAcceleration x dwDriveSpeed) / 4,000,000.

ERROR_INTERPOLATION_NOT_COMPLETE: The interpolation moving started before had not completed.

ERROR_REASSIGN_SYNCH_MODE_COMMAND: The previous Synchronous Operation is not returned.

ERROR_OCCURS_IN_AXIS_X, ERROR_OCCURS_IN_AXIS_Y, ERROR_OCCURS_IN_AXIS_Z,
ERROR_OCCURS_IN_AXIS_U, ERROR_OCCURS_IN_AXIS_XY,
ERROR_OCCURS_IN_AXIS_XZ, ERROR_OCCURS_IN_AXIS_YZ,
ERROR_OCCURS_IN_AXIS_XU, ERROR_OCCURS_IN_AXIS_YU,
ERROR_OCCURS_IN_AXIS_ZU, ERROR_OCCURS_IN_AXIS_XYZ,

ERROR_OCCURS_IN_AXIS_XYU, ERROR_OCCURS_IN_AXIS_XZU,
ERROR_OCCURS_IN_AXIS_YZU,:
Indicates that some error happens to AXIS_xxxx. Please call ps400_get_error_status() for detailed information.

ERROR_MOTION_NOT_COMPLETE: Indicates the previous motion is not completed.

ERROR_AXES_MOVE_CHECK: Cannot forward the Axes-checking command to system, please call GetLastError() for further system information.

ERROR_CONFLICT_WITH_MPG: Indicates the Trapezoidal 3D interpolation moving cannot support Manual-Pulse-Generator, please call ps400_set_mpg() to disable MPG mode.

ERROR_RTX_UNSUPPORT_MODE: Indicates some specific mode is not supported in RTX RTSS DLL.

ERROR_T_LINE3_START: Cannot start Trapezoidal 3D interpolation moving, please call GetLastError() for further system information.

5.1.4 ps400_s_line3_move

short ps400_s_line3_move(BYTE bCardID, WORD wMainAxis, WORD wSecondAxis, WORD wThirdAxis, DWORD dwStartSpeed, DWORD dwDriveSpeed, DWORD dwAccelerationRate, DWORD dwDecelerationRate, long MainAxisFinishPoint, long SecondAxisFinishPoint, long ThirdAxisFinishPoint, short wAccCntOffset = 8, WORD wSyncMode = DISABLE_BLOCK_OPEARTION)

Description:

This function starts the S-curve, 3-dimension linear interpolation moving.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wMainAxis: The main-axis of interpolation moving, can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

wSecondAxis: Thhe second-axis of Interpolation moving, can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U (cannot be the same as ***wMainAxis***).

wThirdAxis: The third-axis of Interpolation moving, can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U (neither ***wMainAxis*** nor ***wSecondAxis*** can be assigned to ***wThirdAxis***)

dwStartSpeed: The Start-Speed in S-curve moving. This Start-Speed will be applied to main-axis.

dwDriveSpeed: The Drive-Speed in S-curve moving. This Drive-Speed will be applied to main-axis.

dwAccelerationRate: The Acceleration-Increasing-Rate in S-curve moving. The Acceleration will be assigned to maximum value automatically. This Acceleration-Increasing-Rate will be applied to main-axis.

dwDecelerationRate: The Deceleration-Increasing-Rate in S-curve moving. The Deceleration will be assigned to maximum value automatically. This Acceleration-Increasing-Rate will be applied to main-axis.

MainAxisFinishPoint: The finish point of main-axis. This parameter is the relative offset to the current position. And the negative value indicates that the finish point is in reverse-direction.

SecondAxisFinishPoint: The finish point of second-axis. This parameter is the relative offset to the current position. And the negative value indicates that the finish point is in reverse-direction.

ThirdAxisFinishPoint: The finish point of third-axis. This parameter is the relative offset to the current position. And the negative value indicates that the finish point is in reverse-direction.

wAccCntOffset: This optional parameter to configure the offset for Acceleration/Deceleration driving. The default setting of ***wAccCntOffset*** is 8.

wSyncMode: In RTX RTSS DLL, only DISABLE_BLOCK_OPEARTION is supported.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wMainAxis,*** ***wSecondAxis*** or ***wThirdAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wMainAxis,*** ***wSecondAxis*** or ***wThirdAxis***.

ERROR_INVALID_INTERPOLATION_SLAVE_AXES: Either ***wSecondAxis*** or ***wThirdAxis*** includes the axis ID assigned to ***wMainAxis***.

ERROR_INTERPOLATION_SLAVE_AXES_DUPLICATED: The axis ID assigned to ***wSecondAxis*** and ***wThirdAxis*** is the same.

ERROR_INVALID_BLOCK_OPEARTION_MODE: Neither DISABLE_BLOCK_OPEARTION nor ENABLE_BLOCK_OPEARTION is assigned to parameter ***wSyncMode***.

ERROR_START_SPEED_EXCEED_DRIVING_SPEED: The ***dwStartSpeed*** is larger than or equal to ***dwDriveSpeed***.

ERROR_INVALID_START_SPEED: The value assigned to parameter ***dwStartSpeed*** is out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_DRIVING_SPEED: The value assigned to parameter ***dwDriveSpeed*** is out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_JERK: The value assigned to parameter ***dwAccelerationRate*** is out of range of Acceleration Increasing Rate. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_DECELERATION_RATE: The value assigned to parameter ***dwDecelerationRate*** is out of range of Deceleration Increasing Rate. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INTERPOLATION_NOT_COMPLETE: The interpolation moving started before had not completed.

ERROR_REASSIGN_SYNCH_MODE_COMMAND: The previous Synchronous Operation is not returned.

ERROR_OCCURS_IN_AXIS_X, ERROR_OCCURS_IN_AXIS_Y, ERROR_OCCURS_IN_AXIS_Z,
ERROR_OCCURS_IN_AXIS_U, ERROR_OCCURS_IN_AXIS_XY,
ERROR_OCCURS_IN_AXIS_XZ, ERROR_OCCURS_IN_AXIS_YZ,
ERROR_OCCURS_IN_AXIS_XU, ERROR_OCCURS_IN_AXIS_YU,
ERROR_OCCURS_IN_AXIS_ZU, ERROR_OCCURS_IN_AXIS_XYZ,
ERROR_OCCURS_IN_AXIS_XYU, ERROR_OCCURS_IN_AXIS_XZU,

ERROR_OCCURS_IN_AXIS_YZU:

Indicates that some error happens to AXIS_xxxx. Please call ps400_get_error_status() for detailed information.

ERROR_MOTION_NOT_COMPLETE: Indicates the previous motion is not completed.

ERROR_AXES_MOVE_CHECK: Cannot forward the Axes-checking command to system, please call GetLastError() for further system information.

ERROR_CONFLICT_WITH_MPG: Indicates the S-curve 3D linear interpolation moving cannot support Manual-Pulse-Generator, please call ps400_set_mpg() to disable MPG mode.

ERROR_RTX_UNSUPPORT_MODE: Indicates some specific mode is not supported in RTX RTSS DLL.

ERROR_S_LINE3_START: Cannot start S-curve 3D linear interpolation moving, please call GetLastError() for further system information.

5.1.5 ps400_t_arc2_move

short ps400_t_arc2_move(BYTE bCardID, WORD wMainAxis, WORD wSlaveAxis, DWORD dwStartSpeed, DWORD dwDriveSpeed, DWORD dwAcceleration, WORD wArcDirection, long MainAxisCenterPoint, long SlaveAxisCenterPoint, long MainAxisFinishPoint, long SlaveAxisFinishPoint, short wAccCntOffset = 8, WORD wSyncMode = DISABLE_BLOCK_OPEARTION)

Description:

This function starts the trapezoidal-profile, 2-dimension circular interpolation moving. Only symmetric trapezoidal Acceleration/Deceleration is applied to circular interpolation. The start-point will be the *Origin* of circular-interpolation motion. The ***MainAxisCenterPoint*** & ***SlaveAxisCenterPoint*** are center coordinates related to *Origin*; and ***MainAxisFinishPoint*** & ***SlaveAxisFinishPoint*** are finish coordinates related to *Origin*. The position tolerance for the specified circular curve is ±1 within the interpolation range. When the value of finish-point reaches the coordinate of *short-axis*, the circular interpolation will be completed. Figure 4 illustrates the finish-point checking of circular interpolation.

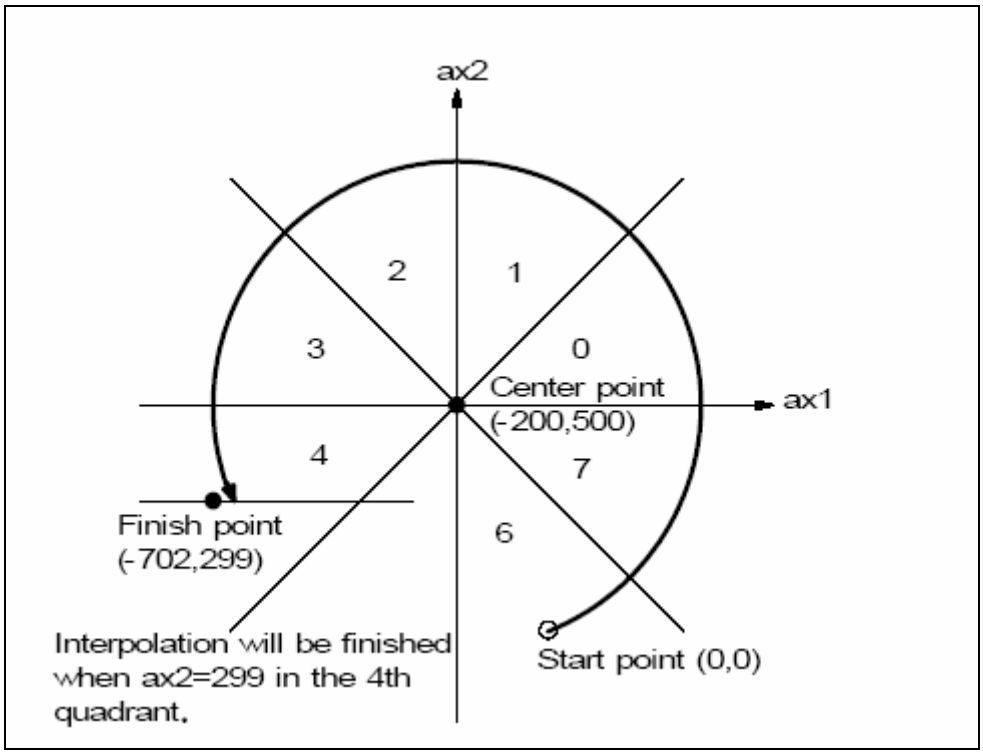

Figure 4 – finish-point checking of circular interpolation

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wMainAxis: The main-axis of Interpolation moving, can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

wSlaveAxis: The slave-axis of Interpolation moving, can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U (cannot be the same as ***wMainAxis***).

dwStartSpeed: The Start-Speed in trapezoidal-profile moving. This Start-Speed will be applied to main-axis.

dwDriveSpeed: The Drive-Speed in trapezoidal-profile moving. This Drive-Speed will be applied to main-axis.

dwAcceleration: The Acceleration in trapezoidal-profile moving. This Acceleration will be applied to main-axis.

wArcDirection: Clockwise (INTERP_ARC_DIRECTION_CLOCKWISE) or Counter-Clockwise (INTERP_ARC_DIRECTION_COUNTER_CLOCKWISE).

MainAxisCenterPoint: The center point of main-axis. This parameter is the relative offset to the current position. And the negative value indicates that the finish point is in reverse-direction.

SlaveAxisCenterPoint: The center point of slave-axis. This parameter is the relative offset to the current position. And the negative value indicates that the finish point is in reverse-direction.

MainAxisFinishPoint: The finish point of main-axis. This parameter is the relative offset to the current position. And the negative value indicates that the finish point is in reverse-direction.

SlaveAxisFinishPoint: The finish point of slave-axis. This parameter is the relative offset to the current position. And the negative value indicates that the finish point is in reverse-direction.

wAccCntOffset: This optional parameter to configure the offset for Acceleration/Deceleration driving. The default setting of ***wAccCntOffset*** is 8.

wSyncMode: In RTX RTSS DLL, only DISABLE_BLOCK_OPEARTION is supported.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wMainAxis*** or ***wSlaveAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wMainAxis*** or ***wSlaveAxis***.

ERROR_INVALID_INTERPOLATION_ARC_DIRECTION: Neither INTERP_ARC_DIRECTION_CLOCKWISE nor

INTERP_ARC_DIRECTION_COUNTER_CLOCKWISE is assigned to parameter ***wArcDirection***.

ERROR_INVALID_INTERPOLATION_SLAVE_AXES: The parameter ***wSlaveAxis*** includes the axis ID assigned to ***wMainAxis***.

ERROR_INVALID_BLOCK_OPEARTION_MODE: Neither DISABLE_BLOCK_OPEARTION nor ENABLE_BLOCK_OPEARTION is assigned to parameter ***wSyncMode***.

ERROR_START_SPEED_EXCEED_DRIVING_SPEED: The ***dwStartSpeed*** is larger than ***dwDriveSpeed***.

ERROR_INVALID_START_SPEED: The value assigned to parameter ***dwStartSpeed*** is out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_DRIVING_SPEED: The value assigned to parameter ***dwDriveSpeed*** is out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_ACCELERATION: The value assigned to parameter ***dwAcceleration*** is out of range of Acceleration. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_ARC_DECELERATION_POINT_CALCULATE: The path of circular moving is too small. Please try to increase the circular-path.

ERROR_INTERPOLATION_NOT_COMPLETE: The interpolation moving started before had not completed.

ERROR_REASSIGN_SYNCH_MODE_COMMAND: The previous Synchronous Operation is not returned.

ERROR_OCCURS_IN_AXIS_X, ERROR_OCCURS_IN_AXIS_Y, ERROR_OCCURS_IN_AXIS_Z,
ERROR_OCCURS_IN_AXIS_U, ERROR_OCCURS_IN_AXIS_XY,
ERROR_OCCURS_IN_AXIS_XZ, ERROR_OCCURS_IN_AXIS_YZ,
ERROR_OCCURS_IN_AXIS_XU, ERROR_OCCURS_IN_AXIS_YU,
ERROR_OCCURS_IN_AXIS_ZU:

Indicates that some error happens to AXIS_xxxx. Please call ps400_get_error_status() for detailed information.

ERROR_MOTION_NOT_COMPLETE: Indicates the previous motion is not completed.

ERROR_AXES_MOVE_CHECK: Cannot forward the Axes-checking command to system, please call GetLastError() for further system information.

ERROR_CONFLICT_WITH_MPG: Indicates the Circular interpolation moving cannot support Manual-Pulse-Generator, please call ps400_set_mpg() to disable MPG mode.

ERROR_RTX_UNSUPPORT_MODE: Indicates some specific mode is not supported in RTX RTSS DLL.

ERROR_T_ARC2_START: Cannot start Circular interpolation moving, please call GetLastError() for further system information.

5.2 Continuous Interpolation Moving

The continuous interpolation provides none-stop linear & circular interpolation moving. The continuous interpolation moving is combined with multiple linear & circular interpolation segments. To add arbitrary interpolation segment, only the constant Vector-Speed is applied to continuous-interpolation moving.

The continuous interpolation moving is configured with ps400_conti_interp_begin() and completed with ps400_conti_interp_end(). All settings that are configured with ps400_conti_interp_begin() will be kept in driver until ps400_conti_interp_end() being called. The interpolation segments after ps400_conti_interp_begin() will use these configurations, including axes involved in interpolation-moving and constant Vector-Speed.

To avoid the continuous interpolation to be interrupted, the configurations of next interpolation segment had better be set as soon as possible. The function, ps400_conti_interp_next_ready(), indicates the next interpolation segment is ready to be configured.

In case the continuous-interpolation moving is terminated, you could re-start the uncompleted interpolation segments with INTERP_CONTINUE_START setting in parameter ***wContiInterpMoveMode***. The typical programming following-chart is described in Figure 5. Please refer to the ‘Conti_Interp’ sample.

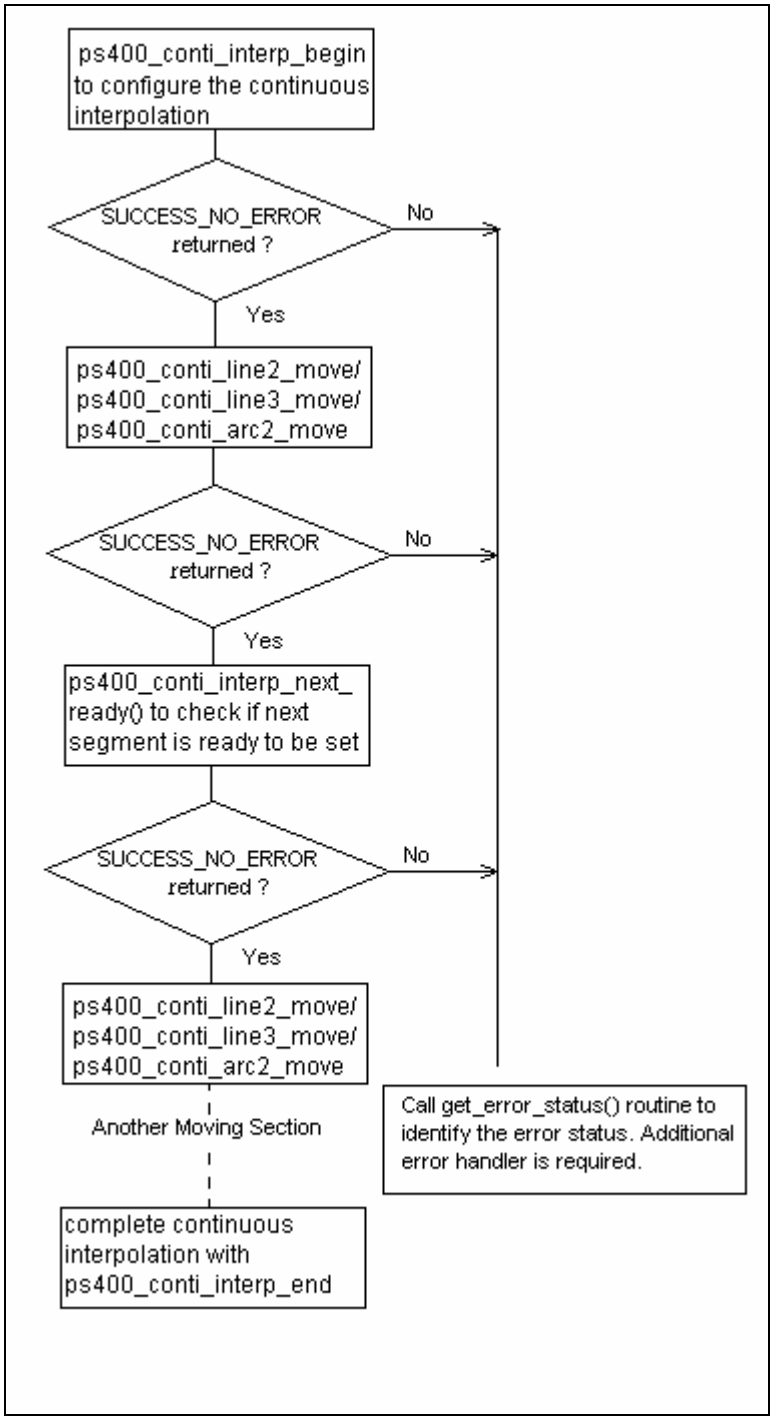

Figure 5 - typical programming following-chart of continuous-interpolation

5.2.1 ps400_conti_interp_begin

short ps400_conti_interp_begin(BYTE bCardID, WORD wMainAxis, WORD wSecondAxis, WORD wThirdAxis, DWORD dwConstSpeed)

Description:

This function configures the involved axes, the constant vector-speed in continuous interpolation moving.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wMainAxis: The main-axis of interpolation moving, can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

wSecondAxis: The second-axis of Interpolation moving, can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U (cannot be the same as ***wMainAxis***).

wThirdAxis: The third-axis of Interpolation moving, can be one of AXIS_X, AXIS_Y, AXIS_Z, AXIS_U or INVALID_AXIS_ASSIGNMENT. (neither ***wMainAxis*** nor ***wSecondAxis*** can be assigned to ***wThirdAxis***).

dwConstSpeed: The constant Vector-Speed in continuous interpolation. This parameter should be less than 2,000,000 PPS)

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wMainAxis, wSecondAxis*** or ***wThirdAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wMainAxis, wSecondAxis*** or ***wThirdAxis***.

ERROR_INVALID_INTERPOLATION_SLAVE_AXES: Either ***wSecondAxis*** or ***wThirdAxis*** includes the axis ID assigned to ***wMainAxis***.

ERROR_INTERPOLATION_SLAVE_AXES_DUPLICATED: The axis ID assigned to ***wSecondAxis*** and ***wThirdAxis*** is the same.

ERROR_INVALID_DRIVING_SPEED: The value assigned to parameter ***dwDriveSpeed*** is out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INTERPOLATION_NOT_COMPLETE: The previous interpolation-moving had not been completed.

ERROR_OCCURS_IN_AXIS_X, ERROR_OCCURS_IN_AXIS_Y, ERROR_OCCURS_IN_AXIS_Z, ERROR_OCCURS_IN_AXIS_U, ERROR_OCCURS_IN_AXIS_XY, ERROR_OCCURS_IN_AXIS_XZ, ERROR_OCCURS_IN_AXIS_YZ, ERROR_OCCURS_IN_AXIS_XU, ERROR_OCCURS_IN_AXIS_YU, ERROR_OCCURS_IN_AXIS_ZU, ERROR_OCCURS_IN_AXIS_XYZ, ERROR_OCCURS_IN_AXIS_XYU, ERROR_OCCURS_IN_AXIS_XZU, ERROR_OCCURS_IN_AXIS_YZU:

Indicates that some error happens to AXIS_xxxx. Please call ps400_get_error_status() for detailed information.

ERROR_MOTION_NOT_COMPLETE: Indicates the previous motion is not completed.

ERROR_AXES_MOVE_CHECK: Cannot forward the Axes-checking command to system, please call GetLastError() for further system information.

ERROR_INVALID_RANGE: The assigned value is invalid.

ERROR_RANGE_CHANGE: Cannot change the settings of ***Range*** register, please call GetLastError() for further system information.

ERROR_CONTI_INTERP_SET: Cannot configure continuous-interpolation moving, please call GetLastError() for further system information.

5.2.2 ps400_conti_interp_next_ready

short ps400_conti_interp_next_ready(BYTE bCardID, BYTE *pReady)

Description:

This function checks if the next interpolation segment is ready to be set.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

pReady: The pointer to the memory that stores the ready-status of next interpolation segment.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_CONFIG_IS_NEEDED: The continuous interpolation had not been configured. Please call ps400_conti_interp_begin() first.

ERROR_CONTI_INTERP_NEXT_READY: Cannot get the next-ready status, please call GetLastError() for further system information.

5.2.3 ps400_conti_line2_move

short ps400_conti_line2_move(BYTE bCardID, long MainAxisFinishPoint, long SlaveAxisFinishPoint, WORD wContiInterpMoveMode)

Description:

This function starts the constant vector-speed, 2-dimension linear interpolation moving.

Parameters:

bCardID: the specific Card ID that is configured with the on-board Dip-Switch.

MainAxisFinishPoint: The finish point of main-axis. This parameter is the relative offset to the current position. And the negative value indicates that the finish point is in reverse-way.

SlaveAxisFinishPoint: The finish point of slave-axis. This parameter is the relative offset to the current position. And the negative value indicates that the finish point is in reverse-way

wContiInterpMoveMode:

INTERP_CONTINUE_START: indicates the begin of continuous interpolation moving.

INTERP_NEXT_CONTINUOUS_MOTION: indicates the interpolation segment is one part of continuous interpolation moving, and the interrupt of motion checking in involved implicitly.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_CONFIG_IS_NEEDED: The continuous interpolation had not been configured. Please call ps400_conti_interp_begin() first.

ERROR_INVALID_CONTINUE_INTERPOLATION_MOTION: Neither INTERP_CONTINUE_START nor INTERP_NEXT_CONTINUOUS_MOTION is assigned to parameter ***wContiInterpMoveMode***.

ERROR_OCCURS_IN_AXIS_X, ERROR_OCCURS_IN_AXIS_Y, ERROR_OCCURS_IN_AXIS_Z, ERROR_OCCURS_IN_AXIS_U, ERROR_OCCURS_IN_AXIS_XY, ERROR_OCCURS_IN_AXIS_XZ, ERROR_OCCURS_IN_AXIS_YZ, ERROR_OCCURS_IN_AXIS_XU, ERROR_OCCURS_IN_AXIS_YU, ERROR_OCCURS_IN_AXIS_ZU:

Indicates that some error happens to AXIS_xxxx. Please call ps400_get_error_status() for detailed information.

ERROR_CONTI_INTERP_INTERRUPTED: Indicates the continuous interpolation moving is interrupted. This code is returned only when ***wContiInterpMoveMode*** is set as INTERP_NEXT_CONTINUOUS_MOTION.

ERROR_MOTION_NOT_COMPLETE: Indicates the previous motion is not completed. This code is returned only when ***wContiInterpMoveMode*** is set as INTERP_CONTINUE_START.

ERROR_CONTI_INTERP_NEXT_NOT_READY: Indicates the internal Motion-Control ASIC is not ready to set the next interpolation segment. Please call ps400_conti_interp_next_ready() first.

ERROR_CONTI_INTERP_LINE2_MOVE: Cannot start ps400_conti_line2_move, please call GetLastError() for further system information.

5.2.4 ps400_conti_line3_move

short ps400_conti_line3_move(BYTE bCardID, long MainAxisFinishPoint, long SecondAxisFinishPoint, long ThirdAxisFinishPoint, WORD wContiInterpMoveMode)

Description:

This function starts the constant vector-speed, 3-dimension linear interpolation moving.

Parameters:

bCardID: the specific Card ID that is configured with the on-board Dip-Switch.

MainAxisFinishPoint: The finish point of main-axis. This parameter is the relative offset to the current position. And the negative value indicates that the finish point is in reverse-way.

SecondAxisFinishPoint: The finish point of second-axis. This parameter is the relative offset to the current position. And the negative value indicates that the finish point is in reverse-way

ThirdAxisFinishPoint: The finish point of third-axis. This parameter is the relative offset to the current position. And the negative value indicates that the finish point is in reverse-way

wContiInterpMoveMode:

INTERP_CONTINUE_START: indicates the begin of continuous interpolation moving.

INTERP_NEXT_CONTINUOUS_MOTION: indicates the interpolation segment is one part of continuous interpolation moving, and the interrupt of motion checking in involved implicitly.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_CONFIG_IS_NEEDED: The continuous interpolation had not been configured. Please call ps400_conti_interp_begin() first.

ERROR_INVALID_CONTINUE_INTERPOLATION_MOTION: Neither INTERP_CONTINUE_START nor INTERP_NEXT_CONTINUOUS_MOTION is assigned to parameter ***wContiInterpMoveMode***.

ERROR_CONTI_INTERP_INCORRECT_CONFIG: Only two axes are configured with ps400_conti_interp_begin(), and ps400_conti_line3_move() failed to execute.

ERROR_OCCURS_IN_AXIS_X, ERROR_OCCURS_IN_AXIS_Y, ERROR_OCCURS_IN_AXIS_Z, ERROR_OCCURS_IN_AXIS_U, ERROR_OCCURS_IN_AXIS_XY, ERROR_OCCURS_IN_AXIS_XZ, ERROR_OCCURS_IN_AXIS_YZ, ERROR_OCCURS_IN_AXIS_XU, ERROR_OCCURS_IN_AXIS_YU,

ERROR_OCCURS_IN_AXIS_ZU, ERROR_OCCURS_IN_AXIS_XYZ,
ERROR_OCCURS_IN_AXIS_XYU, ERROR_OCCURS_IN_AXIS_XZU,
ERROR_OCCURS_IN_AXIS_YZU:

Indicates that some error happens to AXIS_xxxx. Please call ps400_get_error_status() for detailed information.

ERROR_CONTI_INTERP_INTERRUPTED: Indicates the continuous interpolation moving is interrupted. This code is returned only when ***wContiInterpMoveMode*** is set as INTERP_NEXT_CONTINUOUS_MOTION.

ERROR_MOTION_NOT_COMPLETE: Indicates the previous motion is not completed. This code is returned only when ***wContiInterpMoveMode*** is set as INTERP_CONTINUE_START.

ERROR_CONTI_INTERP_NEXT_NOT_READY: Indicates the internal Motion-Control ASIC is not ready to set the next interpolation segment, please call ps400_conti_interp_next_ready() first.

ERROR_CONTI_INTERP_LINE3_MOVE: Cannot start ps400_conti_line3_move, please call GetLastError() for further system information.

5.2.5 ps400_conti_arc2_move

short ps400_conti_arc2_move(BYTE bCardID, WORD wArcDirection, long MainAxisCenterPoint, long SlaveAxisCenterPoint, long MainAxisFinishPoint, long SlaveAxisFinishPoint, WORD wContiInterpMoveMode)

Description:

This function starts the constant vector-speed, 2-dimension linear interpolation moving.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wArcDirection: Clockwise (INTERP_ARC_DIRECTION_CLOCKWISE) or Counter-Clockwise (INTERP_ARC_DIRECTION_COUNTER_CLOCKWISE).

MainAxisCenterPoint: The center point of main-axis. This parameter is the relative offset to the current position. And the negative value indicates that the finish point is in reverse-way.

SlaveAxisCenterPoint: The center point of slave-axis. This parameter is the relative offset to the current position. And the negative value indicates that the finish point is in reverse-way

MainAxisFinishPoint: The finish point of main-axis. This parameter is the relative offset to the current position. And the negative value indicates that the finish point is in reverse-way.

SlaveAxisFinishPoint: The finish point of slave-axis. This parameter is the relative offset to the current position. And the negative value indicates that the finish point is in reverse-way

wContiInterpMoveMode:

INTERP_CONTINUE_START: indicates the begin of continuous interpolation moving.

INTERP_NEXT_CONTINUOUS_MOTION: indicates the interpolation segment is one part of continuous interpolation moving, and the interrupt of motion checking in involved implicitly.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_CONFIG_IS_NEEDED: The continuous interpolation had not been configured. Please call ps400_conti_interp_begin() first.

ERROR_INVALID_CONTINUE_INTERPOLATION_MOTION: Neither INTERP_CONTINUE_START nor INTERP_NEXT_CONTINUOUS_MOTION is assigned to parameter ***wContiInterpMoveMode***.

ERROR_INVALID_INTERPOLATION_ARC_DIRECTION: Neither INTERP_ARC_DIRECTION_CLOCKWISE nor INTERP_ARC_DIRECTION_COUNTER_CLOCKWISE is assigned to parameter ***wArcDirection***.

ERROR_OCCURS_IN_AXIS_X, ERROR_OCCURS_IN_AXIS_Y, ERROR_OCCURS_IN_AXIS_Z, ERROR_OCCURS_IN_AXIS_U, ERROR_OCCURS_IN_AXIS_XY, ERROR_OCCURS_IN_AXIS_XZ, ERROR_OCCURS_IN_AXIS_YZ, ERROR_OCCURS_IN_AXIS_XU, ERROR_OCCURS_IN_AXIS_YU, ERROR_OCCURS_IN_AXIS_ZU:

Indicates that some error happens to AXIS_xxxx. Please call ps400_get_error_status() for detailed information.

ERROR_CONTI_INTERP_INTERRUPTED: Indicates the continuous interpolation moving is interrupted. This code is returned only when ***wContiInterpMoveMode*** is set as INTERP_NEXT_CONTINUOUS_MOTION.

ERROR_MOTION_NOT_COMPLETE: Indicates the previous motion is not completed. This code is returned only when ***wContiInterpMoveMode*** is set as INTERP_CONTINUE_START.

ERROR_CONTI_INTERP_NEXT_NOT_READY: Indicates the internal Motion-Control ASIC is not ready to set the next interpolation segment, please call ps400_conti_interp_next_ready() first.

ERROR_CONTI_INTERP_ARC2_MOVE: Cannot start ps400_conti_arc2_move, please call GetLastError() for further system information.

5.2.6 ps400_conti_interp_end

short ps400_conti_interp_end (BYTE bCardID)

Description:

This function completes the continuous-interpolation moving, and clears the related configurations kept in driver.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_INVALID_RANGE: The assigned value is invalid.

ERROR_RANGE_CHANGE: cannot change the settings of ***Range*** register, please call GetLastError() for further system information.

ERROR_CONTI_INTERP_CLEAR: Cannot clear continuous-interpolation settings, please call GetLastError() for further system information.

CHAPTER 6

Other Motion Functions

This chapter introduces several helpful functions, including ps400_stop_move(), ps400_drv_hold() and ps400_drv_start().

6.1 ps400_stop_move

short ps400_stop_move(BYTE bCardID, WORD wAxis, WORD wStopMode)

Description:

This function stops current motion with slowdown or stop-sudden mode. Please call ps400_motion_down() to make sure the specific axis stop before starting next motion.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

wStopMode: STOP_SLOWDOWN or STOP_SUDDEN.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_INVALID_STOP_MODE: Neither STOP_SLOWDOWN nor STOP_SUDDEN is assigned to parameter ***wStopMode***.

ERROR_MOTION_STOP_SET: Cannot stop current motion, please call GetLastError() for further system information.

6.2 ps400_stop_move_all

short ps400_stop_move_all(BYTE bCardID, WORD wAxes, WORD wStopMode)

Description:

This helpful function stops motion of multiple axes. Please call ps400_motion_done() to make sure that all axes are stop before starting next motion.

Parameters:

bCardID: the specific Card ID that is configured with the on-board Dip-Switch.

wAxes: can be any combination of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

wStopMode: STOP_SLOWDOWN or STOP_SUDDEN.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Invalid axis-combination is assigned to parameter ***wAxes***.

ERROR_INVALID_STOP_MODE: Neither STOP_SLOWDOWN nor STOP_SUDDEN is assigned to parameter ***wStopMode***.

ERROR_MOTION_STOP_SET: Cannot stop current motion, please call GetLastError() for further system information.

6.3 ps400_drv_hold

short ps400_drv_hold(BYTE bCardID, WORD wAxes)

Description:

This function holds the motion-starting of the involved axes. And these involved axes will start moving simultaneously when ps400_drv_start() is called.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxes: Can be any combination of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Invalid axis-combination is assigned to parameter ***wAxes***.

ERROR_MOTION_NOT_COMPLETE: Indicates the previous motion is not completed.

ERROR_HOLD_AXES_NOT_RELEASE: Indicates the hold-axes had not been release, please call ps400_drv_start() to release the hold-axes.

ERROR_DRIVE_HOLD: Cannot hold the motion-starting, please call GetLastError() for further system information.

6.4 ps400_drv_start

short ps400_drv_start(BYTE bCardID, WORD wAxes)

Description:

This function starts multiple axes simultaneously that are held by ps400_drv_hold().

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxes: Can be any combination of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Invalid axis-combination is assigned to parameter ***wAxes***.

ERROR_HOLD_AXES_NOT_MATCH: The axes that will to be started are not match to the axes that are held by ps400_drv_hold().

ERROR_DRIVE_START: Cannot start motion of held axes, please call GetLastError() for further system information.

CHAPTER 7

Advanced Motion Configurations

This chapter introduces some advanced features, including variable-ring counter, manual-pulse-generator, compare-and-trigger and the synchronous-action between axes. The interrupt factors and the axis-related events are introduced in this chapter, too.
The advanced features are specific features, and maybe not co-exist with normal operation. It's recommended to disable these functions when they are not needed.

7.1 ps400_set_vring

short ps400_set_vring(BYTE bCardID, WORD wAxis, WORD wVRINGEnable, DWORD dwRingValue)

Description:
This function enables and configures the variable-ring feature for both logic-command and encoder-position counters. After enabling variable-ring feature, these two counters will be reset to zero automatically.

Parameters:
bCardID: The specific Card ID that is configured with the on-board Dip-Switch.
wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.
wVRINGEnable: VARIABLE_RING_ENABLE_FEATURE or
VARIABLE_RING_DISABLE_FEATURE.
dwRingValue: The total numbers that counter can count. (2 < dwRingValue).
For instance, assigning 10,000 to dwRingValue indicates the operation of ring-counter will be: increasing in forward direction … →9998→9999→0→1→…
decreasing in reserve direction …→1→0→9999→9998→…

Return Code:
SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_INVALID_RING_COUNTER: Indicates the parameter ***dwRingValue*** is less than 2.

ERROR_INVALID_FILTER_ENABLE: Neither VARIABLE_RING_ENABLE_FEATURE nor VARIABLE_RING_DISABLE_FEATURE is assigned to parameter ***wVRINGEnable***.

ERROR_CONFLICT_WITH_SOFTLIMIT: Indicates that the AXIS-Internal Comparators are used for software-limit. Please disable software-limit feature with ps400_set_softlimit().

ERROR_CONFLICT_WITH_CMPTRIG: Indicates that the ASIC-Internal Comparators are used for Compare & Trigger feature. Please disable Compare & Trigger feature with ps400_const_pitch_trig_config().

ERROR_CONFLICT_WITH_SYNCH_ACTION: Indicates that the ASIC-Internal Comparators are used as the condition of Synchronous-Action. Please disable synchronous-condition with ps400_set_synch().

ERROR_VRING_SET: Cannot enable/configure the variable-ring feature, please call GetLastError() for further system information.

7.2 ps400_set_mpg

short ps400_set_mpg(BYTE bCardID, WORD wAxis, WORD wEXPConfig, DWORD dwFixedPulse, DWORD dwSpeed, DWORD dwMaxMPGFreq)

Description:

This function enables and configures the manual-pulse-generator feature. After enabling manual-pulse-generator feature, the constant-speed motion will be started when every pulse is sent from external manual-pulse-generator.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

wEXPConfig: EXP_AB_PHASE_MPG, EXP_CW_CCW_ACTIVE_LOW_MPG or EXP_DISABLE_FEATURE.

dwFixedPulse: Indicates the numbers of pulse will be output when each pulse is sent from manual-pulse-generator. For instance, assigning 5 to this parameter, 5 pulses will be output when each pulse is sent from external manual-pulse-generator.

dwSpeed: The constant-speed of output pulse.

dwMaxMPGFreq: The maximum frequency of the manual-pulse-generator. Please check the datasheet of manual-pulse-generator.

Notice: the following formula is needed to be satisfied:

dwSpeed $\geqq$ dwMaxMPGFreq x dwFixedPulse x 2.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_INVALID_MPG_EXP_CONFIG: Indicates the invalid MPG-mode is assigned to parameter ***wEXPConfig***.

ERROR_INVALID_MPG_SPEED: The value assigned to parameter ***dwSpeed*** is out of range of Speed or is less than ***2*** **x** ***dwMaxMPGFreq*** **x** ***dwFixedPulse***.

ERROR_MPG_SET: Cannot enable/configure the manual-pulse-generator, please call GetLastError() for further system information.

7.3 ps400_const_pitch_trig_config

short ps400_const_pitch_trig_config(BYTE bCardID, WORD wAxis, WORD wCmpTrigEnable, WORD wCmpSource, WORD wOutpotLogic, WORD wPulseWidth, DWORD dwConstPitch)

Description:

This function enables and configures the Compare-and-Trigger feature. Both constant-pitch and variable-offset Compare-and-Trigger are supported. If the wCmpTrigMode is configured as CMPTRIG_VARIABLE_OFFSET, each offset will be loaded into the comparator of Motion-ASIC in ISR (Interrupt-Service-Routine). Therefore, some functions related to interrupt will not be allowed. These functions are:

ps400_set_int_factor(),ps400_config_FRnet()
and Block/Non-Block operation mode in
ps400_home_start(),ps400_t_line2_move(),ps400_t_line3_move(),ps400_s_line2_move(), ps400_s_line3_move(),ps400_t_arc2_move().

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.
wAxis: AXIS_X or AXIS_Y.
wCmpTrigEnable: CMPTRIG_ENABLE_FEATURE or CMPTRIG_DISABLE_FEATURE.
wCmpSource: CMP_SRC_LOGIC_COMMAND or CMP_SRC_ENCODER_POSITION.
wOutpotLogic: CMPTRIG_LOGIC_ACTIVE_HIGH or CMPTRIG_LOGIC_ACTIVE_LOW.
wPulseWidth: Can be one of settings:

TRIG_PULSE_WIDTH_100us,
TRIG_PULSE_WIDTH_200us,
TRIG_PULSE_WIDTH_1ms,
TRIG_PULSE_WIDTH_2ms,
TRIG_PULSE_WIDTH_10ms,
TRIG_PULSE_WIDTH_20ms.

dwConstPitch: The constant-pitch (pulse) is needed when ***wCmpTrigMode*** =
CMPTRIG_CONSTANT_PITCH. (2 $\leqq$ dwConstPitch $\leqq$ 2,147,483,647)

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.
ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).
ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_INVALID_AXIS: Neither AXIS_X nor AXIS_Y is assigned to parameter ***wAxis*** ***(wCmpTrigEnable =*** CMPTRIG_ENABLE_FEATURE***).***

ERROR_INVALID_CMPTRIG_ENABLE: Neither CMPTRIG_ENABLE_FEATURE nor CMPTRIG_DISABLE_FEATURE is assigned to parameter ***wCmpTrigEnable***.

ERROR_INVALID_COMPARE_SOURCE: Neither CMP_SRC_LOGIC_COMMAND nor CMP_SRC_ENCODER_POSITION is assigned to parameter ***wCmpSource***.

ERROR_INVALID_CMPTRIG_LOGIC_LEVEL: Neither CMPTRIG_LOGIC_ACTIVE_HIGH nor CMPTRIG_LOGIC_ACTIVE_LOW is assigned to parameter ***wOutpotLogic***.

ERROR_INVALID_CMPTRIG_PULSE_WIDTH: Indicates no valid settings is assigned to parameter ***wPulseWidth***.

ERROR_INVALID_CONST_PITCH: Indicates the parameter ***dwConstPitch*** is out of valid range (2 $\leqq$ dwConstPitch $\leqq$ 2,147,483,647)

ERROR_CONFLICT_WITH_SOFTLIMIT: Indicates that the AXIS-Internal Comparators are used for software-limit. Please disable software-limit feature with ps400_set_softlimit().

ERROR_CONFLICT_WITH_VRING: Indicates that the ASIC-Internal Comparators are used for Variable-Ring counter. Please disable Variable-Ring counter with ps400_set_vring().

ERROR_CONFLICT_WITH_SYNCH_ACTION: Indicates that the ASIC-Internal Comparators are used as the condition of Synchronous-Action. Please disable synchronous-condition with ps400_set_synch().

ERROR_CONFIG_IS_NEEDED: Indicates no compare-and-trigger feature had been configured (***wCmpTrigEnable*** = CMPTRIG_DISABLE_FEATURE).

ERROR_CMPTRIG_SET: Cannot enable/configure the compare-and-trigger feature, please call GetLastError() for further system information.

7.4 ps400_set_int_factor

short ps400_set_int_factor(BYTE bCardID, WORD wAxis, WORD wIntFactor)

Description:

This function configures the motion-related interrupt-factors. Please call ps400_get_int_status() to get the relevant interrupt-status;

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

wIntFactor: The interrupt factor can be the combination of following settings:

INT_FACTOR_DISABLE: disables all interrupt factors.

INT_FACTOR_PULSE: interrupt will be triggered when pulse is at its active level. For instance, if the PULSE_LOGIC_ACTIVE_HIGH is configured in ps400_set_pls_cfg(), the interrupt will raised at each raising-edge of output-pulse.

INT_FACTOR_EXCEED_CMP_NEGATIVE: interrupt will be triggered while the content of logic-command/encoder-position counter is larger than COMP- comparator.

INT_FACTOR_LESS_CMP_NEGATIVE: interrupt will be triggered while the content of logic-command/encoder-position counter is less than COMP- comparator.

INT_FACTOR_LESS_CMP_POSITIVE: interrupt will be triggered while the content of logic-command/encoder-position counter is less than COMP+ comparator.

INT_FACTOR_EXCEED_CMP_POSITIVE: interrupt will be triggered while the content of logic-command/encoder-position counter is larger than COMP+ comparator.

INT_FACTOR_END_CONST_SPEED_MOVE: interrupt will be triggered when the constant-speed moving is completed.

INT_FACTOR_START_CONST_SPEED_MOVE: interrupt will be triggered when the constant-speed moving is started.

INT_FACTOR_END_DRIVING: interrupt will be triggered when the motion is completed.

Notice:

If the INT_FACTOR_START_CONST_SPEED_MOVE is set, the interrupt will be triggered both at the end of Acceleration and Deceleration.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERORR_INVALID_INT_FACTOR: Indicates the invalid interrupt factor is assigned to parameter ***wIntFactor***.

ERROR_CONFLICT_WITH_CMPTRIG: Indicates the Compare-and-Trigger feature is enabled, and interrupt-factor cannot co-exist with this feature. Please call ps400_const_pitch_trig_config () to disable Compare-and-Trigger feature.

ERROR_INT_FACTOR_SET: Cannot enable/configure the specific interrupt factor, please call GetLastError() for further system information.

7.5 ps400_set_synch

short ps400_set_synch(BYTE bCardID, WORD wMainAxis, WORD wSyncEnable, WORD wSyncAxes, WORD wSyncCondition, WORD wSyncActionMainAxis, WORD wSyncActionOtherAxes, WORD wCmpSource, DWORD dwComparatorPositive, DWORD dwComparatorNegative)

Description:

This function enables and configures the synchronous condition/actions in main-axis and other axes that the synchronous action will be applied to.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wMainAxis: The main-axis of synchronous-action, this parameter can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_Y.

wSyncEnable: SYNC_ENABLE_FEATURE or SYNC_DISABLE_FEATURE.

wSyncAxes: The other invloved axes of synchronous-action, they be any combination of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U. However, this parameter cannot include ***wMainAxis***.

wSyncCondition: The conditions of main-axis to start synchronous-actions, including

SYNC_CONDITION_EXCEED_CMP_POSITIVE: the value of counter ≧ COMP+.

SYNC_CONDITION_LESS_CMP_POSITIVE: the value of counter < COMP+.

SYNC_CONDITION_LESS_CMP_NEGATIVE: the value of counter < COMP-.

SYNC_CONDITION_EXCEED_CMP_NEGATIVE: the value of counter ≧ COMP-.

SYNC_CONDITION_START_DRIVING: moving is started.

SYNC_CONDITION_END_DRIVING: moving is completed or terminated.

wSyncActionMainAxis: When the synchronous-condition is satisfied, the synchronous action that can be applied to main-axis, including:

SYNC_ACTION_NONE: no synchronous action is applied to main-axis.

SYNC_ACTION_LOGIC_CMD_LATCH: stores the content of logic-command counter into ***Buffer*** Register.

SYNC_ACTION_ENCODER_POS_LATCH: stores the content of encoder-position counter into ***Buffer*** Register.

wSyncActionOtherAxes: When the synchronous-condition is satisfied, the synchronous action that can be applied to other axes, including:

SYNC_ACTION_NONE: no synchronous action is applied to other axes.

SYNC_ACTION_FIXED_FORWARD_DRIVE: starts the fixed-pulse moving in forward direction.

SYNC_ACTION_FIXED_REVERSE_DRIVE: starts the fixed-pulse moving in reverse direction.

SYNC_ACTION_CONTINUE_FORWARD_DRIVE: starts the velocity moving in forward direction.

SYNC_ACTION_CONTINUE_REVERSE_DRIVE: starts the velocity moving in reverse direction.

SYNC_ACTION_SLOWDOWN_STOP: stops the current motion with slowdown mode.

SYNC_ACTION_SUDDEN_STOP: stops the current motion immediately.

SYNC_ACTION_LOGIC_CMD_LATCH: stores the content of logic-command counter into ***Buffer*** Register.

SYNC_ACTION_ENCODER_POS_LATCH: stores the content of encoder-position counter into ***Buffer*** Register.

wCmpSource: CMP_SRC_LOGIC_COMMAND or CMP_SRC_ENCODER_POSITION

dwComparatorPositive: the value to be set into COMP+.

dwComparatorNegative: the value to be set into COMP-.

Notice: please refer to ps400_get_latch() to read the ***Buffer*** register.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wMainAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wMainAxis***.

ERROR_INVALID_SYNCHRONOUS_AXES: Indicates invalid axes are assigned to parameter ***wSyncAxes***.

ERROR_INVALID_SYNCH_ENABLE: Neither SYNC_ENABLE_FEATURE nor SYNC_DISABLE_FEATURE is assigned to parameter ***wSyncEnable***.

ERROR_INVALID_SYNCH_CONDITION: Indicates no valid synchronous-condition is assigned to parameter ***wSyncCondition***.

ERROR_INVALID_SYNCH_ACTION: Indicates invalid synchronous-action is assigned to parameter ***wSyncActionMainAxis*** or ***wSyncActionOtherAxes***.

ERROR_INVALID_COMPARE_SOURCE: Neither CMP_SRC_LOGIC_COMMAND nor CMP_SRC_ENCODER_POSITION is assigned to parameter ***wCmpSource***.

ERROR_CONFLICT_WITH_SOFTLIMIT: Indicates that the AXIS-Internal Comparators are used for software-limit. Please disable software-limit feature with ps400_set_softlimit().

ERROR_CONFLICT_WITH_VRING: Indicates that the ASIC-Internal Comparators are used for Variable-Ring counter. Please disable Variable-Ring counter with ps400_set_vring().

ERROR_CONFLICT_WITH_CMPTRIG: Indicates that the ASIC-Internal Comparators are used for Conpare & Trigger feature. Please disable Compare & Trigger feature with ps400_const_pitch_trig_config ().

ERROR_SYNCH_SET: Cannot enable/configure the synchronous condition & actions, please call GetLastError() for further system information.

7.6 ps400_synch_t_move_cfg

short ps400_synch_t_move_cfg(BYTE bCardID, WORD wAxis, DWORD dwStartSpeed, DWORD dwDriveSpeed, DWORD dwAcceleration, DWORD dwDeceleration, long FixedPulse)

Description:

This function configures the necessary parameters of trapezoidal-profile moving. This function is helpful when the synchronous-action is set as SYNC_ACTION_XXXX_XXXXXXXX_DRIVE.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: This axis should be one axis of the parameter ***wSyncAxes*** that is assigned to ps400_set_synch().

dwStartSpeed: The Start-Speed in trapezoidal-profile moving.

dwDriveSpeed: The Drive-Speed in trapezoidal-profile moving.

dwAcceleration: The Acceleration in trapezoidal-profile moving.

dwDeceleration: The Deceleration in trapezoidal-profile moving.

FixedPulse: The total numbers of output pulse. This parameter is a signed 32-bits variable, the negative value indicates motion in reverse-direction

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_START_SPEED_EXCEED_DRIVING_SPEED: The ***dwStartSpeed*** is larger than ***dwDriveSpeed***.

ERROR_INVALID_START_SPEED: The value assigned to parameter ***dwStartSpeed*** is out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_DRIVING_SPEED: The value assigned to parameter ***dwDriveSpeed*** is out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_ACCELERATION: The value assigned to parameter ***dwAcceleration*** is out of range of Acceleration. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_DECELERATION: The value assigned to parameter ***dwDeceleration*** is out of range of Deceleration. Please refer to ps400_set_range() and ps400_get_range_settings().

Notice: In the case, ***dwAcceleration*** > ***dwDeceleration*** , the following formula should be satisfied, too.

dwDeceleration > (dwAcceleration x dwDriveSpeed) / 4,000,000.

ERROR_OCCURS_IN_AXIS_X, ERROR_OCCURS_IN_AXIS_Y, ERROR_OCCURS_IN_AXIS_Z ERROR_OCCURS_IN_AXIS_U:

Indicates that some error happens to AXIS_X, AXIS_Y, AXIS_Z or AXIS_U. Please call ps400_get_error_status() for detailed information.

ERROR_MOTION_NOT_COMPLETE: Indicates the previous motion is not completed.

ERROR_T_MOVE_START: Cannot configure the parameters of trapezoidal moving, please call GetLastError() for further system information.

7.7 ps400_synch_s_move_cfg

short ps400_synch_s_move_cfg(BYTE bCardID, WORD wAxis, DWORD dwStartSpeed, DWORD dwDriveSpeed, DWORD dwAccelerationRate, DWORD dwDecelerationRate, long FixedPulse)

Description:

This function configures the necessary parameters of S-curve moving. This function is helpful when the synchronous-action is set as SYNC_ACTION_FIXED_XXXXXXXX_DRIVE.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: This axis should be one axis of the parameter ***wSyncAxes*** that is assigned to ps400_set_synch().

dwStartSpeed: The Start-Speed in S-curve moving.

dwDriveSpeed: The Drive-Speed in S-curve moving.

dwAccelerationRate: The Acceleration-Increasing-Rate in S-curve moving. The Acceleration will be assigned to maximum value automatically.

dwDecelerationRate: The Deceleration-Increasing-Rate in S-curve moving. The Deceleration will be assigned to maximum value automatically.

FixedPulse: The total numbers of output pulse. This parameter is a signed 32-bits variable, the negative value indicates motion in reverse-direction

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_START_SPEED_EXCEED_DRIVING_SPEED: The ***dwStartSpeed*** is larger than or equal to ***dwDriveSpeed***.

ERROR_INVALID_START_SPEED: The value assigned to parameter ***dwStartSpeed*** is out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_DRIVING_SPEED: The value assigned to parameter ***dwDriveSpeed*** is out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_JERK: The value assigned to parameter ***dwAccelerationRate*** is out of range of Acceleration Increasing Rate. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_INVALID_DECELERATION_RATE: The value assigned to parameter ***dwDecelerationRate*** is out of range of Deceleration Increasing Rate. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_OCCURS_IN_AXIS_X, ERROR_OCCURS_IN_AXIS_Y, ERROR_OCCURS_IN_AXIS_Z ERROR_OCCURS_IN_AXIS_U:

Indicates that some error happens to AXIS_X, AXIS_Y, AXIS_Z or AXIS_U. Please call ps400_get_error_status() for detailed information.

ERROR_MOTION_NOT_COMPLETE: Indicates the previous motion is not completed.

ERROR_S_MOVE_START: Cannot configure the parameters of S-curve moving, please call GetLastError() for further system information.

CHAPTER 8

Miscellaneous Functions

This chapter introduces some functions that are hard to be cataloged, including setting the logic-command counter and encoder-position counter, triangle prevention of trapezoidal-profile fixed-pulse driving, changing driving-speed while trapezoidal-profile moving and updating the total numbers of output pulse.

8.1 ps400_t_change_v

short ps400_t_change_v(BYTE bCardID, WORD wAxis, DWORD dwDriveSpeed)

Description:

This function changes the Drive-Speed during trapezoidal-profile moving.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

dwDriveSpeed: The Drive-Speed in trapezoidal-profile moving.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_INVALID_DRIVING_SPEED: The value assigned to parameter ***dwDriveSpeed*** is out of range of Speed. Please refer to ps400_set_range() and ps400_get_range_settings().

ERROR_SPEED_CHANGE_FAIL_IN_ACC_DEC: Indicates the Drive-Speed cannot be changed during Acceleration/Deceleration.

ERROR_INVALID_OPERATION_IN_S_CURVE: Indicates the Drive-Speed cannot be applied to S-curve moving.

ERROR_NOT_CONSTANT_SPEED_IN_T_MOVE: Indicates the Drive-Speed cannot be changed in non-constant speed area of trapezoidal-profile moving.

ERROR_T_DRIVING_SPEED_CHANGE: Cannot change the Drive-Speed, please call GetLastError() for further system information.

8.2 ps400_t_set_avtri

short ps400_t_set_avtri(BYTE bCardID, WORD wAxis, WORD wAvTriCfg)

Description:

This function enables the triangle prevention of fixed-pulse, trapezoidal-profile moving. After enabling this feature, the Motion-Control ASIC will determine the deceleration-point by the following formula: *Numbers of output pulse* $\geqq$ *2 x (pulse number at Acceleration + pulse number at Deceleration).*

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

wAvTriCfg: AVOID_TRIANGLE_ENABLE_FEATURE or AVOID_TRIANGLE_DISABLE_FEATURE.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_INVALID_AVOID_TRIANGLE_CONFIG: Neither AVOID_TRIANGLE_ENABLE_FEATURE nor AVOID_TRIANGLE_DISABLE_FEATURE is assigned to parameter ***wAvTriCfg***.

ERROR_T_AVOID_TRIANGLE_SET: Cannot set the avoid-triangle feature, please call GetLastError() for further system information.

8.3 ps400_change_p

short ps400_change_p(BYTE bCardID, WORD wAxis, DWORD dwP)

Description:

This function changes the total numbers of output pulse during moving.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

dwP: The total numbers of output pulse.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_CONFLICT_WITH_INTERPOLATION_MOVE: Indicates the finish-point of interpolation moving cannot be changed dynamically.

ERROR_OUTPUT_PULSE_CHANGE: Cannot change total number of output pulse, please call GetLastError() for further system information.

8.4 ps400_set_cmdcounter

short ps400_set_cmdcounter(BYTE bCardID, WORD wAxis, long lData)

Description:

This function set the content of logic-command counter.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

lData: The value to be set to logic-command counter.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_CMD_COUNTER_SET: Cannot set the logic-command counter, please call GetLastError() for further system information.

8.5 ps400_set_position

short ps400_set_position(BYTE bCardID, WORD wAxis, DWORD lData)

Description:

This function set the content of encoder-position counter.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

lData: The value to be set to encoder-position counter.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_POS_COUNTER_SET: Cannot set the encoder-position counter, please call GetLastError() for further system information.

CHAPTER 9

Status

This chapter introduces functions to get the status of PISO-PS400, including ps400_motion_done(), ps400_get_cmdcounter(), ps400_get_position(), ps400_get_speed(), ps400_get_acc(), ps400_get_latch(),ps400_get_mdi_status(),ps400_get_in3(),ps400_get_int_status() and ps400_get_error_status().

9.1 ps400_motion_done

short ps400_motion_done(BYTE bCardID, WORD wAxis, BYTE* pDone, WORD* pStopStatus)

Description:

This function checks the completion of motion and reports the cause of motion-completion.

Parameters:

bCardID: the specific Card ID that is configured with the on-board Dip-Switch.

wAxis: can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

pDone: The pointer to the memory that stores the motion-status. The motion-status will be

MOTION_DONE: the specific axis is stop.

MOTION_NOT_DONE: the specific axis is driving.

pStopStatus: The pointer to the memory that stores the cause of motion-completion, including

DRIVE_FINISH_WITH_SW_LIMIT_POSITIVE: reaches software limit in forward direction and stops

DRIVE_FINISH_WITH_SW_LIMIT_NEGATIVE: reaches software limit in reverse direction and stops

DRIVE_FINISH_WITH_STOP_COMMAND: the stop command is executed.

DRIVE_FINISH_OUTPUT_FIXED_PULSE: completion of fixed-pulse moving.

DRIVE_FINISH_WITH_AUTO_HOME: completion of automatic-home-search.

DRIVE_FINISH_WITH_LIMIT_POSITIVE: reaches hardware limit in forward direction and stops

DRIVE_FINISH_WITH_LIMIT_NEGATIVE: reaches hardware limit in reverse direction and stops

DRIVE_FINISH_WITH_ALARM: the ALARM feature is enabled and is active to stop driving.

DRIVE_FINISH_WITH_EMG: the driving is stopped when EMG is active.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_ACCESS_VIOLATION_DATA_COPY: Some system exception occurs while copying memory, please check the pointer-type parameter you assign to this function.

ERROR_MOTION_DONE_GET: Cannot get motion status, please call GetLastError() for further system information.

9.2 ps400_get_cmdcounter

short ps400_get_cmdcounter(BYTE bCardID, WORD wAxis, long* pData)

Description:

This function gets the content of logic-command counter.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

pData: The pointer to the memory that stores logic-command counter.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_ACCESS_VIOLATION_DATA_COPY: Some system exception occurs while copying memory, please check the pointer-type parameter you assign to this function.

ERROR_CMD_COUNTER_GET: Cannot get the logic-command counter, please call GetLastError() for further system information.

9.3 ps400_get_position

short ps400_get_position(BYTE bCardID, WORD wAxis, long* pData)

Description:

This function gets the content of encoder-position counter.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

pData: The pointer to the memory that stores encoder-position counter.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_ACCESS_VIOLATION_DATA_COPY: Some system exception occurs while copying memory, please check the pointer-type parameter you assign to this function.

ERROR_POS_COUNTER_GET: Cannot get the encoder-position counter, please call GetLastError() for further system information.

9.4 ps400_get_speed

short ps400_get_speed(BYTE bCardID, WORD wAxis, DWORD* pSpeed)

Description:

This function gets the speed of current motion.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

pSpeed: The pointer to the memory that stores speed of current motion.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_ACCESS_VIOLATION_DATA_COPY: Some system exception occurs while copying memory, please check the pointer-type parameter you assign to this function.

ERROR_SPEED_GET: Cannot get the current speed, please call GetLastError() for further system information.

9.5 ps400_get_acc

short ps400_get_acc(BYTE bCardID, WORD wAxis, DWORD* pAcc)

Description:

This function gets the acceleration of current motion.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

pAcc: The pointer to the memory that stores acceleration of current motion.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_ACCESS_VIOLATION_DATA_COPY: Some system exception occurs while copying memory, please check the pointer-type parameter you assign to this function.

ERROR_ACCELERATION_GET: cannot get the current acceleration, please call GetLastError() for further system information.

9.6 ps400_get_latch

short ps400_get_latch(BYTE bCardID, WORD wAxis, long* pLatchData)

Description:

This function gets the content of ***Buffer*** register.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

pLatchData: The pointer to the memory that stores data latched in ***Buffer*** register. Please refer to ps400_set_synch() for to latched data into ***Buffer*** register.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_ACCESS_VIOLATION_DATA_COPY: Some system exception occurs while copying memory, please check the pointer-type parameter you assign to this function.

ERROR_LATCH_GET: Cannot get the latched data, please call GetLastError() for further system information.

9.7 ps400_get_mdi_status

short ps400_get_mdi_status(BYTE bCardID, WORD wAxis, WORD* pDIStatus)

Description:

This function checks the status of motion-related digital inputs.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

pDIStatus: The pointer to the memory that stores the motion-related digital inputs. The value stored in this parameter may be the combination of following status:

DI_STATUS_ACTIVE_DRIVING: indicates the specific axis is driving.

DI_STATUS_ACTIVE_LMTP: the hardware limit in forward direction is active.

DI_STATUS_ACTIVE_LMTM: the hardware limit in reverse direction is active.

DI_STATUS_ACTIVE_EMG: the EMG signal is active.

DI_STATUS_ACTIVE_ALARM: the ALARM signal is enabled and active.

DI_STATUS_ACTIVE_HOME: the Home (ORG) signal is active.

DI_STATUS_ACTIVE_NEARHOME: the Near-Home (NORG) signal is active.

DI_STATUS_ACTIVE_INP: the INP signal is enabled and active.

DI_STATUS_ACTIVE_INDEX: The Z-Phase/INDEX signal is active.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_ACCESS_VIOLATION_DATA_COPY: Some system exception occurs while copying memory, please check the pointer-type parameter you assign to this function.

ERROR_DI_STATUS_GET Cannot get motion-related digital inputs, please call GetLastError() for further system information.

9.8 ps400_get_in3

short ps400_get_in3(BYTE bCardID, WORD wAxis, BYTE* pIN3Status)

Description:

This function gets the status of digital input ***IN3***.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

pIN3Status: The pointer to the memory that stores the input of IN3. The value stored in this parameter will be 0x01 or 0x00.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_ACCESS_VIOLATION_DATA_COPY: Some system exception occurs while copying memory, please check the pointer-type parameter you assign to this function.

ERROR_IN3_GET: Cannot get the digital input ***IN3***, please call GetLastError() for further system information.

9.9 ps400_get_int_status

short ps400_get_int_status(BYTE bCardID, WORD wAxis, WORD* pIntStatus)

Description:

This function gets the status of interrupt factors.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

pIntStatus: The pointer to the memory that stores the status of interrupt factors, including

INT_STATUS_EXCEED_CMP_NEGATIVE,

INT_STATUS_LESS_CMP_NEGATIVE,

INT_STATUS_LESS_CMP_POSITIVE,

INT_STATUS_EXCEED_CMP_POSITIVE,

INT_STATUS_END_CONST_SPEED_MOVE,

INT_STATUS_START_CONST_SPEED_MOVE,

INT_STATUS_END_DRIVING

Please refer to ps400_set_int_factor() for the setting of relative interrupt factors.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_ACCESS_VIOLATION_DATA_COPY: Some system exception occurs while copying memory, please check the pointer-type parameter you assign to this function.

ERROR_INT_STATUS_GET: Cannot get the status of interrupt, please call GetLastError() for further system information.

9.10 ps400_get_error_status

short ps400_get_error_status(BYTE bCardID, WORD wAxis, WORD* pErrorStatus)

Description:

This function gets the error-status of specific axis.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wAxis: Can be one of AXIS_X, AXIS_Y, AXIS_Z or AXIS_U.

pErrorStatus: The pointer to the memory that stores the error status, including

DRIVE_ERROR_STATUS_SLMTP: error caused by software limit in forward direction.

DRIVE_ERROR_STATUS_SLMTM: error cause by software limit in reverse direction.

DRIVE_ERROR_STATUS_LMTP: error caused by hardware limit in forward direction.

DRIVE_ERROR_STATUS_LMTM: error caused by hardware limit in reverse direction.

DRIVE_ERROR_STATUS_ALARM: error caused by ALARM signal.

DRIVE_ERROR_STATUS_EMG: error caused by EMG signal.

DRIVE_ERROR_STATUS_HOME: error caused by execution of automatic-home-search. The Z-Phase/INDEX is already active at the start of Automatic-Home-Search Step-3. Please refer to ps400_set_home_cfg() for detailed information.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_MULTI_AXES_ASSIGNED: Multiple axes are assigned to parameter ***wAxis***.

ERROR_NO_VALID_AXIS_ASSIGNED: No valid axis ID is assigned to parameter ***wAxis***.

ERROR_ACCESS_VIOLATION_DATA_COPY: Some system exception occurs while copying memory, please check the pointer-type parameter you assign to this function.

ERROR_ERROR_STATUS_GET: Cannot get the error status, please call GetLastError() for further system information.

CHAPTER 10

FRnet I/O extension

The PISO-PS400 equips the FRnet ASIC to connect the remote FRnet I/O extension. The FRnet ASIC provides the real-time status-updating of its remote I/O modules. This chapter introduces the FRnet functions, including ps400_scan_FRnet_DI(),ps400_reset_FRnet(),ps400_config_FRnet(),ps400_get_FRnet_DI() and ps400_set_FRnet_DO().

10.1 ps400_scan_FRnet_DI

short ps400_scan_FRnet_DI(BYTE bCardID, WORD *pDIModules)

Description:

This function scans the active *FRnet* DI modules.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

pDIModules: The pointer to the memory that indicates the active *FRnet* DI modules. Each bit of this parameter stands for the presence of active *FRnet* DI module.

B7							B0
SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_ACCESS_VIOLATION_DATA_COPY: Some system exception occurs while copying memory, please check the pointer-type parameter you assign to this function.

ERROR_FRNET_DI_MODULE_GET: Cannot scan the active *FRnet* DI modules, please call GetLastError() for further system information.

10.2 ps400_reset_FRnet

short ps400_reset_FRnet(BYTE bCardID)

Description:

This function reset the onboard *FRnet* master controller.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_FRNET_RESET: Cannot reset *FRnet* controller, please call GetLastError() for further system information.

10.4 ps400_get_FRnet_DI

short ps400_get_FRnet_DI(BYTE bCardID, WORD wSA, WORD *pStatus, WORD wEnableDirectAccess)

Description:

This function get the digital-inputs of *FRnet* DI module.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wSA: The Group-Address of *FRnet* DI module. One of following Group-Address can be assigned to this parameter:

FRNET_SA8,
FRNET_SA9,
FRNET_SA10,
FRNET_SA11,
FRNET_SA12,
FRNET_SA13,
FRNET_SA14,
FRNET_SA15.

pStatus: The pointer to the WORD that indicates each digital-input of *FRnet* DI module.

B15							B8
DI_15	DI_14	DI_13	DI_12	DI_11	DI_10	DI_9	DI_8
B7							B0
DI_7	DI_6	DI_5	DI_4	DI_3	DI_2	DI_1	DI_0

wEnableDirectAccess: In RTX RTSS DLL, only FRNET_ENABLE_DIRECT_ACCESS is supported.

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_INVALID_FRNET_ACCESS_MODE: Neither FRNET_ENABLE_DIRECT_ACCESS nor FRNET_DISABLE_DIRECT_ACCESS is assigned to parameter ***wEnableDirectAccess***.

ERROR_INVALID_FRNET_SA_GROUP_ADDRESS: Indicates the invalid Group-Address of *FRnet* DI module is assigned to parameter ***wSA***.

ERROR_CONFIG_IS_NEEDED: Indicates the periodic-updating had not been configured when FRNET_DISABLE_DIRECT_ACCESS is assigned to the parameter ***wEnableDirectAccess***. Please enable periodic-updating with ps400_config_FRnet() function.

ERROR_ACCESS_VIOLATION_DATA_COPY: Some system exception occurs while copying memory, please check the pointer-type parameter you assign to this function.

ERROR_RTX_UNSUPPORT_MODE: Indicates some specific mode is not supported in RTX RTSS DLL.

ERROR_FRNET_INPUT: Cannot get the digital-inputs of *FRnet* DI module, please call GetLastError() for further system information.

10.5 ps400_set_FRnet_DO

short ps400_set_FRnet_DO(BYTE bCardID, WORD wRA, WORD wDOData)

Description:

This function set the digital-outputs of *FRnet* DO module.

Parameters:

bCardID: The specific Card ID that is configured with the on-board Dip-Switch.

wRA: The Group-Address of *FRnet* D0 module. One of following Group-Address can be assigned to this parameter:

FRNET_RA0,

FRNET_RA1,

FRNET_RA2,

FRNET_RA3,

FRNET_RA7,

FRNET_RA5,

FRNET_RA6,

FRNET_RA7.

wDOData: The 16-bits data to be set to *FRnet* DO module.

B15							B8
DO_15	DO_14	DO_13	DO_12	DO_11	DO_10	DO_9	DO_8
B7							B0
DO_7	DO_6	DO_5	DO_4	DO_3	DO_2	DO_1	DO_0

Return Code:

SUCCESS_NO_ERROR: The function returns successfully.

ERROR_INVALID_CARD_ID: There is no active PISO-PS400 card configured with bCardID, or the given Card ID is invalid (for instance, Card ID is assigned to 254).

ERROR_INVALID_FRNET_RA_GROUP_ADDRESS: Indicates the invalid Group-Address of *FRnet* DO module is assigned to parameter ***wRA***.

ERROR_FRNET_OUTPUT: Cannot set the digital-outputs of *FRnet* DO module, please call GetLastError() for further system information.

APPENDIX A

Error Code

The Error Codes are divided into tree parts: System Error, Parameter Error and Runtime Error.

SUCCESS_NO_ERROR	0

System Error:

ERROR_ROUTINE_FAIL_BASE	-100
ERROR_GET_CARD_ID	-101
ERROR_DEVICE_OPEN	-102
ERROR_DEVICE_CLOSE	-103
ERROR_CARD_RESET	-104
ERROR_RANGE_CHANGE	-105
ERROR_PULSE_MODE_SET	-106
ERROR_ENCODER_MODE_SET	-107
ERROR_LIMIT_SENSOR_SET	-108
ERROR_INP_SIGNAL_SET	-109
ERROR_ALARM_SIGNAL_SET	-110
ERROR_SERVO_ON_SET	-111
ERROR_IN3_SET	-112
ERROR_IN3_GET	-113
ERROR_FILTER_SET	-114
ERROR_SW_LIMIT_SET	-115
ERROR_HOME_CFG_SET	-116
ERROR_HOME_LIMIT_SET	-117
ERROR_START_HOME	-118
ERROR_DI_STATUS_GET	-119
ERROR_ERROR_STATUS_GET	-120
ERROR_CMD_COUNTER_SET	-121
ERROR_CMD_COUNTER_GET	-122
ERROR_POS_COUNTER_SET	-123

ERROR_POS_COUNTER_GET	-124
ERROR_MOTION_DONE_GET	-125
ERROR_SPEED_GET	-126
ERROR_ACCELERATION_GET	-127
ERROR_LATCH_GET	-128
ERROR_MOTION_STOP_SET	-129
ERROR_MOTION_STOP_ALL_SET	-130
ERROR_DRIVE_START	-131
ERROR_DRIVE_HOLD	-132
ERROR_VRING_SET	-133
ERROR_MPG_SET	-134
ERROR_CMPTRIG_SET	-135
ERROR_SYNCH_SET	-136
ERROR_INT_FACTOR_SET	-137
ERROR_INT_STATUS_GET	-138
ERROR_CONTI_MOVE_START	-139
ERROR_CONST_MOVE_START	-140
ERROR_T_MOVE_START	-141
ERROR_S_MOVE_START	-142
ERROR_T_LINE2_START	-143
ERROR_T_LINE3_START	-144
ERROR_S_LINE2_START	-145
ERROR_S_LINE3_START	-146
ERROR_T_ARC2_START	-147
ERROR_CONTI_INTERP_SET	-148
ERROR_CONTI_INTERP_CLEAR	-149
ERROR_CONTI_INTERP_NEXT_READY	-150
ERROR_CONTI_INTERP_LINE2_MOVE	-151
ERROR_CONTI_INTERP_LINE3_MOVE	-152
ERROR_CONTI_INTERP_ARC2_MOVE	-153
ERROR_T_DRIVING_SPEED_CHANGE	-154
ERROR_T_AVOID_TRIANGLE_SET	-155
ERROR_OUTPUT_PULSE_CHANGE	-156
ERROR_OUT1_GET	-157
ERROR_FRNET_DI_MODULE_GET	-158
ERROR_FRNET_FREQUENCY_SET	-159
ERROR_FRNET_INPUT	-160
ERROR_FRNET_OUTPUT	-161

ERROR_FRNET_RESET	-162

Parameter Error:

ERROR_INVALID_PARAMETER_BASE	-200
ERROR_INVALID_CARD_ID	-201
ERROR_INVALID_SCANNED_INDEX	-202
ERROR_CARD_ID_DUPLICATED	-203
ERROR_INVALID_RANGE	-204
ERROR_INVALID_PULSE_MODE	-205
ERROR_INVALID_PULSE_LEVEL	-206
ERROR_INVALID_PULSE_DIRECTION	-207
ERROR_INVALID_ENCODER_MODE	-208
ERROR_INVALID_LIMIT_LOGIC	-209
ERROR_INVALID_STOP_MODE	-210
ERROR_INVALID_INP_ENABLE	-211
ERROR_INVALID_INP_LOGIC_LEVEL	-212
ERROR_INVALID_ALARM_ENABLE	-213
ERROR_INVALID_ALARM_LOGIC_LEVEL	-214
ERROR_INVALID_SERVO_SETTING	-215
ERROR_INVALID_IN3_ENABLE	-216
ERROR_INVALID_IN3_LOGIC_LEVEL	-217
ERROR_INVALID_FILTER_ENABLE	-218
ERROR_INVALID_FILTER_CONFIGURATION	-219
ERROR_INVALID_FILTER_DELAY_TIME	-220
ERROR_INVALID_SOFTWARE_LIMIT_ENABLE	-221
ERROR_INVALID_SOFTWARE_LIMIT_COMPARATOR_SOURCE	-222
ERROR_INVALID_MOVE_DIRECTION	-223
ERROR_INVALID_HOME_LOGIC_LEVEL	-224
ERROR_INVALID_NEAR_HOME_LOGIC_LEVEL	-225
ERROR_INVALID_INDEX_LOGIC_LEVEL	-226
ERROR_INVALID_AUTO_HOME_STEP	-227
ERROR_INVALID_BLOCK_OPEARTION_MODE	-228
ERROR_INVALID_AVOID_TRIANGLE_CONFIG	-229
ERROR_INVALID_MPG_EXP_CONFIG	-230
ERROR_INVALID_NHOME_SEARCH_SPEED	-231
ERROR_INVALID_HOME_SEARCH_SPEED	-232
ERROR_INVALID_ACCELERATION	-233
ERROR_INVALID_DECELERATION	-234

ERROR_INVALID_JERK	-235
ERROR_INVALID_DECELERATION_RATE	-236
ERROR_INVALID_RING_COUNTER	-237
ERROR_INVALID_RING_ENABLE	-238
ERROR_INVALID_AXIS	-239
ERROR_INVALID_CONST_PITCH	-240
ERROR_INVALID_OFFSET_BUFFER	-241
ERROR_INVALID_OFFSET_LEN	-242
ERROR_INVALID_OFFSET_DATA	-243
ERROR_INVALID_START_SPEED	-244
ERROR_INVALID_DRIVING_SPEED	-245
ERROR_INVALID_MANUAL_DECELERATION_POINT	-246
ERROR_START_SPEED_EXCEED_DRIVING_SPEED	-247
ERROR_MULTI_AXES_ASSIGNED	-248
ERROR_NO_VALID_AXIS_ASSIGNED	-249
ERROR_INVALID_INTERPOLATION_SLAVE_AXES	-250
ERROR_INTERPOLATION_SLAVE_AXES_DUPLICATED	-251
ERROR_INVALID_SYNCHRONOUS_AXES	-252
ERROR_INVALID_INTERPOLATION_ARC_DIRECTION	-253
ERROR_INVALID_CONTINUE_INTERPOLATION_MOTION	-254
ERROR_INVALID_FRNET_PERIODIC_ENABLE	-255
ERROR_INVALID_FRNET_PERIODIC_FACTOR	-256
ERROR_INVALID_FRNET_SA_GROUP_ADDRESS	-257
ERROR_INVALID_FRNET_RA_GROUP_ADDRESS	-258
ERROR_INVALID_FRNET_ACCESS_MODE	-259
ERROR_INVALID_COMPARE_SOURCE	-260
ERROR_INVALID_MPG_SPEED	-261
ERROR_INVALID_CMPTRIG_ENABLE	-262
ERROR_INVALID_CMPTRIG_TRIGGER_MODE	-263
ERROR_INVALID_CMPTRIG_LOGIC_LEVEL	-264
ERROR_INVALID_CMPTRIG_PULSE_WIDTH	-265
ERROR_INVALID_SYNCH_ENABLE	-266
ERROR_INVALID_SYNCH_CONDITION	-267
ERROR_INVALID_SYNCH_ACTION	-268
ERROR_INVALID_EVENT_ENABLE	-269
ERORR_INVALID_INT_FACTOR	-270
ERORR_INVALID_ COMPARATOR	-271

Runtime Error:

ERROR_RUNTIME_BASE	-300
ERROR_OCCURS_IN_AXIS_X	-301
ERROR_OCCURS_IN_AXIS_Y	-302
ERROR_OCCURS_IN_AXIS_XY	-303
ERROR_OCCURS_IN_AXIS_Z	-304
ERROR_OCCURS_IN_AXIS_XZ	-305
ERROR_OCCURS_IN_AXIS_YZ	-306
ERROR_OCCURS_IN_AXIS_XYZ	-307
ERROR_OCCURS_IN_AXIS_U	-308
ERROR_OCCURS_IN_AXIS_XU	-309
ERROR_OCCURS_IN_AXIS_YU	-310
ERROR_OCCURS_IN_AXIS_XYU	-311
ERROR_OCCURS_IN_AXIS_ZU	-312
ERROR_OCCURS_IN_AXIS_XZU	-313
ERROR_OCCURS_IN_AXIS_YZU	-314
ERROR_OCCURS_IN_AXIS_XYZU	-315
ERROR_NO_CARD_FOUND	-316
ERROR_MEMORY_MAP	-317
ERROR_MEMORY_UNMAP	-318
ERROR_ACCESS_VIOLATION_DATA_COPY	-319
ERROR_VARIABLE_PITCH_SET	-320
ERROR_INT_EVENT_ATTACH	-321
ERROR_INT_EVENT_DETTACH	-322
ERROR_INT_EVENT_CREATE	-323
ERROR_CONFIG_IS_NEEDED	-324
ERROR_MOTION_NOT_COMPLETE	-325
ERROR_CONFLICT_WITH_SOFTLIMIT	-326
ERROR_CONFLICT_WITH_CMPTRIG	-327
ERROR_CONFLICT_WITH_VRING	-328
ERROR_CONFLICT_WITH_SYNCH_ACTION	-329
ERROR_ARC_DECELERATION_POINT_CALCULATE	-330
ERROR_REASSIGN_SYNCH_MODE_COMMAND	-331
ERROR_OVERLAP_EVENT_CREATE	-332
ERROR_INTERPOLATION_NOT_COMPLETE	-333
ERROR_CONTI_INTERP_INTERRUPTED	-334
ERROR_CONTI_INTERP_INCORRECT_CONFIG	-335
ERROR_CONTI_INTERP_NEXT_NOT_READY	-336

ERROR_SPEED_CHANGE_FAIL_IN_ACC_DEC	-337
ERROR_INVALID_OPERATION_IN_S_CURVE	-338
ERROR_NOT_CONSTANT_SPEED_IN_T_MOVE	-339
ERROR_MOTION_IS_COMPLETED	-340
ERROR_CONFLICT_WITH_INTERPOLATION_MOVE	-341
ERROR_HOLD_AXES_NOT_MATCH	-342
ERROR_BLOCK_OP_CONFLICT_WITH_CMPTRIG	-343
ERROR_CONFLICT_WITH_MPG	-344
ERROR_HOLD_AXES_NOT_RELEASE	-345
ERROR_ZPHASE_ACTIVE_AT_STEP3	-346
ERROR_AXES_MOVE_CHECK	-360
ERROR_IOCTL_FAILED	-361
ERROR_UNDEFINED_EXCEPTION	-362
ERROR_RTX_UNSUPPORT_MODE	-363
ERROR_CONFIG_FILE_LOAD	-370
ERROR_CONFLICT_IN_CONFIG_FILE	-371
ERROR_INVALID_FILE_HANDLE	-372