PCI-1245/1265 Series

DSP-Based SoftMotion
PCI Controller

User Manual

Copyright

Acknowledgments

PC-LabCard is a trademark of Advantech Co., Ltd.

IBM and PC are trademarks of International Business Machines Corporation.

MS-DOS, Windows®, Microsoft® Visual C++ and Visual BASIC are trademarks of Microsoft® Corporation.

Intel® and Pentium® are trademarks of Intel Corporation.

Delphi and C++Builder are trademarks of Inprise Corporation.

CE Notification

The PCI-1245/1245V/1245E/1265, developed by Advantech CO., LTD., has passed the CE test for environmental specifications when shielded cables are used for external wiring. We recommend the use of shielded cables. This kind of cable is available from Advantech. Contact your local supplier for ordering information.

Part No. 2003126512 Edition 3

Printed in Taiwan January 2013

Product Warranty (2 years)

Advantech warrants to you, the original purchaser, that each of its products will be free from defects in materials and workmanship for two years from the date of purchase.

This warranty does not apply to any products which have been repaired or altered by persons other than repair personnel authorized by Advantech, or which have been subject to misuse, abuse, accident or improper installation. Advantech assumes no liability under the terms of this warranty as a consequence of such events.

Because of Advantech's high quality-control standards and rigorous testing, most of our customers never need to use our repair service. If an Advantech product is defective, it will be repaired or replaced at no charge during the warranty period. For out-of-warranty repairs, you will be billed according to the cost of replacement materials, service time and freight. Consult your dealer for more details.

If you think you have a defective product, follow these steps:

1. Collect all the information about the problem encountered. (For example, CPU speed, Advantech products used, other hardware and software used, etc.) Note anything abnormal and list any onscreen messages you get when the problem occurs.
2. Call your dealer and describe the problem. Have your manual, product, and any helpful information readily available.
3. If your product is diagnosed as defective, obtain an RMA (return merchandize authorization) number from your dealer. This allows us to process your return more quickly.
4. Carefully pack the defective product, a fully-completed Repair and Replacement Order Card and a photocopy proof of purchase date (such as your sales receipt) in a shippable container. A product returned without proof of the purchase date is not eligible for warranty service.
5. Write the RMA number visibly on the outside of the package and ship it prepaid to your dealer.

Technical Support and Assistance

1. Visit the Advantech web site at **www.Advantech.com/support** where you can find the latest information about the product.
2. Contact your distributor, sales representative, or Advantech's customer service center for technical support if you need additional assistance. Have the following information ready before you call:
   - Product name and serial number
   - Description of your peripheral attachments
   - Description of your software (operating system, version, application software, etc.)
   - A complete description of the problem
   - The exact wording of any error messages

Packing List

Before setting up the system, check that the items listed below are included and in good condition. If any item does not accord with the table, Contact your dealer immediately.

- PCI-1245/1245V/1245E/1265
- Companion CD-ROM (DLL driver included)
- Startup Manual

Safety Precaution - Static Electricity

Follow these simple precautions to protect yourself from harm and the products from damage.

1. To avoid electrical shock, always disconnect the power from your PC chassis before you work on it. Don't touch any components on the CPU card or other cards while the PC is on.
2. Disconnect power before making any configuration changes. The sudden rush of power as you connect a jumper or install a card may damage sensitive electronic components.

Contents

2

CHAPTER 1

Introduction

This chapter introduces PCI-1245/1245V/1245E/1265 and lists their special features and detailed specifications.

Chapter 1 Introduction

PCI-1245/1245V/1245E/1265 series are DSP-based SoftMotion PCI bus controller boards which are designed for electrical machine automation and traditional machine automation wide applications. The board is equipped with high-performance DSP with SoftMotion algorithm inside to perform the motion trajectory and timing control to meet the synchronization in precise movement.

The Advantech SoftMotion features synchronization control in gantry, electronic gear and electronic CAM; interpolation in linear, circular and helical (spiral) curve; continuous movement in buffering piecewise trajectory to realize; cutting movement in tangential following to ensure the Z-axes is tangent to X-Y curve; high-speed position compare and triggering with any 3rd party machine vision solution.

All Advantech motion controllers are applied to "Common Motion API" architecture which is an unified user programming interface. Programmer can benefit from integrating any Advantech SoftMotion controller without changing the application code in large scale. This architecture can save the effort of application maintenance and upgrade.

1.1 Features

PCI-1245/1245V/1245E/1265 are featured by the following points (The features are listed, but varied by product model name)

- Encoder input is 10 MHz for 4xAB mode, 2.5 MHz for CW/CCW mode
- Pulse output up to 5 Mpps
- Memory buffer (10 K points) for trajectory planning which is designed in DSP
- Supports E-Gear, and helical interpolation
- Supports E-CAM providing 256 points to describe the CAM profiles which buffers located in DSP
- Hardware emergency input
- Watchdog timer
- Position latch via ORG & index signal
- Position compare triggering up to 100 KHz, and memory buffer is up to 100 K points in DSP

- Programmable interrupt
- Supports gantry mode by semi-closed loop pulse train control
- RDY/LTC-dedicated input channels & SVON/CMP/CAM-DO/ ERC-dedicated output channels are switchable for general input and output purposes

1.2 Applications

- Precise X-Y-Z position control
- Precise rotation control
- Semi-conductor packaging, assembly equipment and high-speed pick-and-place testing machine

1.3 Installation Guide

Before you install the card, make sure you have the following necessary components:

- PCI-1245/1245V/1245E/1265 card
- User manual
- Driver and software
- Utility
- PCL-10251 wiring cable between PCI board and terminal board
- ADAM-3955 terminal boards
- Any PCL-10153MJ3/PCL-10153YS5/PCL-10153PA5/PCL-10153PA5LS/PCL-10153DA2 cable between terminal board and servo drive (Supports Mitsubishi J3, Yaskawa Sigma V, Panasonic A4/A5/MINAS A and Delta A2)
- Industrial-grade PC with PCI bus slot

1.4 Accessories

Advantech offers a complete set of accessory products. These accessories include:

Wiring Cables to Wiring Board

- PCL-10251 - PCL-10251 is a 100-pin shielded cable. To achieve a better signal quality, the signal wires are twisted in such away as to form a "twisted-pair cable", reducing cross talk and noise from other signal sources.
- PCL-10152 - PCL-10152 is a 50-pin shielded cable. To achieve a better signal quality, the signal wires are twisted in such away as to form a "twisted-pair cable", reducing cross talk and noise from other signal sources.

Wiring Cable Table	
PCI-1245/ 1245V/1245E	PCL-10251
PCI-1265	PCL-10251 (100-pin) and PCL-10152 (50-pin)

Wiring Board

- ADAM-3955 - ADAM-3955 is specially designed for servo drive connection in a convenient way. The wiring board features 2-axis design. For instance, if you use PCI-1245 board, two wiring boards are necessary for 4-axis control. The fast-to-connect transfer cable are available for Panasonic A4/A5/MINAS A, Yaskawa Sigma V, Mitsubishi J3 and Delta A2 servo.
- ADAM-3956 - ADAM-3956 is specially designed for servo drive connection in a convenient way. The wiring board features 4-axis design. For instance, if you use PCI-1245 board, only one wiring board are necessary for 4-axis control. The fast-to-connect transfer cables are available for Panasonic A4/A5/MINAS, Yaskawa Sigma V, Mitsubishi J3 and Delta A2 servo.

Transfer Cables to Servo

- PCL-10153PA5 - PCL-10153PA5 is a 50-pin cable connecting ADAM-3955/ADAM-3956 to Panasonic A4 and A5 servo.
- PCL-10153PA5LS - PCL-10153PA5LS is a 50-pin cable connecting ADAM-3955/ADAM-3956 to Panasonic MINAS A servo.

- PCL-10153YS5 - PCL-10153YS5 is a 50-pin cable connecting ADAM-3955/ADAM-3956 to Yaskawa Sigma V servo.
- PCL-10153MJ3 - PCL-10153MJ3 is a 50-pin cable connecting ADAM-3955/ADAM-3956 to Mitsubishi J3 servo.
- PCL-10153DA2 - PCL-10153DA2 is a 50-pin cable connecting ADAM-3955/ADAM-3956 to Delta A2 servo.

2

CHAPTER 2

Installation

This chapter instructs users how to proceed step-by-step process for driver and hardware installation.

Chapter 2 Installation

2.1 Unpacking

After receiving your PCI-1245/1245V/1245E/1265 package, inspect the contents first. The package should include the following items:

- PCI-1245/1245V/1245E/1265 card
- CD-ROM (DLL driver & user manual included)

The PCI-1245/1245V/1245E/1265 card has certain electronic components vulnerable to electrostatic discharge (ESD). ESD could easily damage the integrated circuits and certain components if preventive measures are not carefully taken.

Before removing the card from the antistatic plastic bag, you should take following precautions to ward off possible ESD damage:

- Touch the metal part of your computer chassis with your hand to discharge static electricity accumulated on your body. Or one can also use a grounding strap.
- Touch the antistatic bag to a metal part of your computer chassis before opening the bag.
- Hold of the card only by the metal bracket when taking it out of the bag.

After taking out the card, you should first:

- Inspect the card for any possible signs of external damage (loose or damaged components, etc.). If the card is visibly damaged, notify our service department or the local sales representative immediately. Avoid installing a damaged card into your system.

Also pay extra attention to the followings to ensure a proper installation:

- Avoid physical contact with materials that could hold static electricity such as plastic, vinyl and Styrofoam.
- Whenever you handle the card, grasp it only by its edges. DO NOT TOUCH the exposed metal pins of the connector or the electronic components.

2.2 Driver Installation

We recommend you to install the driver before you install the PCI-1245/1245V/1245E/1265 card into your system.

The DLL driver setup program for the card is included on the companion CD-ROM that is shipped with package. Follow the steps below to install the driver software:

1. Insert the companion CD-ROM into your CD-ROM drive.
2. The setup program will be launched automatically if you have the autoplay function enabled on your system.

Note: If the autoplay function is not enabled on your computer, use Windows Explorer or Windows Run command to execute SETUP.EXE on the companion CD-ROM.

3. Select the proper Windows OS option according to your operating system. Just follow the installation instructions step by step to complete your DLL driver setup.
4. Then setup the PCI-1245/1245V/1245E/1265 Motion Utility automatically.

For further information on driver-related issues, an online version of the Device Drivers Manual is available by accessing the following path:

Start\Advantech Automation\Motion \(Board Name)

The example source codes could be found under the corresponding installation folder, such as the default installation path:

\Program Files\Advantech\ Motion \(Board Name)\Examples

2.3 Hardware Installation

Note: Make sure you have installed the driver first before you install the card (refer to 2.2 Driver Installation)

After the DLL driver installation is completed, you can now go on to install the PCI-1245/1245V/1245E/1265 card in any PCI slot on your computer. But it is suggested that you should refer to the computer's user manual or related documentations if you have any doubt. Follow the steps below to install the card on your system.

1. Turn off your computer and remove any accessories connected to the computer.
   Warning! CUT OFF power supply of your computer whenever you install or remove any card, or connect and disconnect cables.
2. Disconnect the power cord and any other cables from the back of the computer.
3. Remove the cover of the computer.
4. Select an empty +3.3/+5 V PCI slot. Remove the screws that secures the expansion slot cover to the system unit. Save the screws to secure the retaining bracket of interface card.
5. Carefully grasp the upper edge of the PCI-1245/1245V/1245E/1265. Align the hole in the retaining bracket with the hole on the expansion slot and align the gold striped edge connector with the expansion slot socket. Press the card into the socket gently but firmly. Make sure the card fits the slot tightly. Use of excessive force must be avoided; otherwise the card might be damaged.
6. Fasten the bracket of the PCI card on the back panel rail of the computer with screws.
7. Connect appropriate accessories (100-pin cable, wiring terminals, etc. if necessary) to the PCI card.
8. Replace the cover of your computer and connect the cables you removed in step 2.
9. Turn on your computer.

2

CHAPTER 3

Signal Connections

This chapter provides information about how to connect input and output signals.

Chapter 3 Signal Connections

3.1 I/O Connector Pin Assignments

The I/O connector on the PCI-1245/1245V/1245E/1265 is a 100-pin connector that enables you to connect to accessories via the PCL-10251 shielded cable.

Figure 3.1 and figure 3.2 show the pin assignments for the 100-pin I/O connector on the PCI-1245/1245V/1245E/1265, and table 3-1 shows its I/O connector signal description.

Note: *The PCL-10251 shielded cable is especially designed for the PCI-1245/1245V/1245E/1265 series to reduce noise in the analog signal lines. Refer to section 1.4 Accessories.*

VEX	1	51	VEX
EMG	2	52	NC
X_LMT+	3	53	Z_LMT+
X_LMT-	4	54	Z_LMT-
X_IN1 / LTC	5	55	Z_IN1 / LTC
X_IN2 / RDY	6	56	Z_IN2 / RDY
X_ORG	7	57	Z_ORG
Y_LMT+	8	58	U_LMT+
Y_LMT-	9	59	U_LMT-
Y_IN1 / LTC	10	60	U_IN1 / LTC
Y_IN2 / RDY	11	61	U_IN2 / RDY
Y_ORG	12	62	U_ORG
X_INP	13	63	Z_INP
X_ALM	14	64	Z_ALM
X_ECA+	15	65	Z_ECA+
X_ECA-	16	66	Z_ECA-
X_ECB+	17	67	Z_ECB+
X_ECB-	18	68	Z_ECB-
X_ECZ+	19	69	Z_ECZ+
X_ECZ-	20	70	Z_ECZ-
Y_INP	21	71	U_INP
Y_ALM	22	72	U_ALM
Y_ECA+	23	73	U_ECA+
Y_ECA-	24	74	U_ECA-
Y_ECB+	25	75	U_ECB+
Y_ECB-	26	76	U_ECB-
Y_ECZ+	27	77	U_ECZ+
Y_ECZ-	28	78	U_ECZ-
X_IN4 / JOG+	29	79	Z_IN4
X_IN5 / JOG-	30	80	Z_IN5
Y_IN4	31	81	U_IN4
Y_IN5	32	82	U_IN5
EGND	33	83	EGND
X_OUT4 / CAM-DO	34	84	Z_OUT4 / CAM-DO
X_OUT5 / CMP	35	85	Z_OUT5 / CMP
X_OUT6 / SVON	36	86	Z_OUT6 / SVON
X_OUT7 / ERC	37	87	Z_OUT7 / ERC
X_CW+ / PULS+	38	88	Z_CW+ / PULS+
X_CW- / PULS-	39	89	Z_CW- / PULS-
X_CCW+ / DIR+	40	90	Z_CCW+ / DIR+
X_CCW- / DIR-	41	91	Z_CCW- / DIR-
EGND	42	92	EGND
Y_OUT4 / CAM-DO	43	93	U_OUT4 / CAM-DO
Y_OUT5 / CMP	44	94	U_OUT5 / CMP
Y_OUT6 / SVON	45	95	U_OUT6 / SVON
Y_OUT7 / ERC	46	96	U_OUT7 / ERC
Y_CW+ / PULS+	47	97	U_CW+ / PULS+
Y_CW- / PULS-	48	98	U_CW- / PULS-
Y_CCW+ / DIR+	49	99	U_CCW+ / DIR+
Y_CCW- / DIR-	50	100	U_CCW- / DIR-

Figure 3.1: I/O Connector Pin Assignments for PCI-1245/1245V/1245E/1265

Signal	Pin	Pin	Signal
VEX	1	26	NC
V_LMT+	2	27	V_LMT+
V_IN1 / LTC	3	28	V_IN2 / RDY
V_ORG	4	29	Y_LMT+
W_LMT-	5	30	W_IN1 / LTC
W_IN2 / RDY	6	31	W_ORG
V_INP	7	32	V_ALM
V_ECA+	8	33	V_ECA-
V_ECB+	9	34	V_ECB-
V_ECZ+	10	35	V_ECZ-
W_INP	11	36	W_ALM
W_ECA+	12	37	W_ECA-
W_ECB+	13	38	W_ECB-
W_ECZ+	14	39	W_ECZ-
V_IN4	15	40	V_IN5
W_IN4	16	41	W_IN5
EGND	17	42	V_OUT4 / CAM-DO
V_OUT5 / CMP	18	43	V_OUT6 / SVON
V_OUT7 / ERC	19	44	V_CW+ / PULS+
V_CW- / PULS-	20	45	V_CW- / PULS-
V_CCW- / DIR-	21	46	EGND
W_OUT4 / CAM-DO	22	47	W_OUT5 / CMP
W_OUT6 / SVON	23	48	W_OUT7 / ERC
W_CW+ / PULS+	24	49	W_CW- / PULS-
W_CCW+ / DIR+	25	50	W_CCW- / DIR-

Signal	Pin	Pin	Signal
DI0	1	2	EGND
DI2	3	4	DI1
DI4	5	6	DI3
DI6	7	8	DI5
DO0	9	10	DI7
DO2	11	12	DO1
DO4	13	14	DO3
DO6	15	16	DO5
AIN0	17	18	DO7/VEX
AIN1	19	20	AGND

Figure 3.2: I/O Connector Pin Assignments for PCI-1265 (Daughter Board)

Note: *DO7 / VEX can be chosen by jumper settings. For detailed information, please refer to Chapter 3.2*

Table 3.1: I/O Connector Signal Description			
Signal Name	**Reference**	**Direction**	**Description**
VEX	-	Input	External Power (12~24V_{DC})
EMG	-	Input	Emergency Stop (for all axes)
LMT+	-	Input	+ Direction Limit
LMT-	-	Input	- Direction Limit
LTC	-	Input	Position Latch
RDY	-	Input	Servo Ready
ORG	-	Input	Home Position
INP	-	Input	In-Position Input
ALM	-	Input	Servo Error
ECA+	-	Input	Encoder Phase A+
ECA-	-	Input	Encoder Phase A -
ECB+	-	Input	Encoder Phase B +
ECB-	-	Input	Encoder Phase B -
ECZ+	-	Input	Encoder Phase Z +
ECZ-	-	Input	Encoder Phase Z -
EGND	-	-	Ground
DI	EGND	Input	General-purposed digital input
DO	EGND	Output	General-purposed digital output
CAM-DO	EGND	Output	DO during assigned position interval and vice versa
CMP	EGND	Output	Compare to Trigger Output
SVON	EGND	Output	Servo ON
ERC	EGND	Output	Error Counter Clear
CW+ / PULS+	EGND	Output	Output pulse CW/Pulse+
CW- / PULS-	EGND	Output	Output pulse CW/Pulse-
CCW+ / DIR+	EGND	Output	Output pulse CCW/DIR+
CCW- / DIR-	EGND	Output	Output pulse CCW/DIR-
AI	AGND	Input	Analog input

Note:

1. *X, Y, Z,U,V and W represent for ID of each axis.*
2. *RDY & LTC dedicated input channels are designed to be switchable and support general purpose input channel usage.*
3. *SVON, CMP, CAM-DO and ERC dedicated output channels are designed to be switchable and support general purpose output channel usage.*
4. *For easy to note, DO4/CAM-DO, DO5/CMP, DO6/SVON and DO7/ERC will be used and expressed in motion utility.*
5. *X_IN4 has three switchable functions - general purpose input, JOG+ and MPG+ (Manual Pulser).*
6. *X_IN5 has three switchblade functions - general purpose input, JOG- and MPG-(Manual Pulser).*

3.2 Location of DIP switch

Figure 3.2 shows the names and locations of DIP switch on the PCI-1245/1245V/1245E/1265. The switch is used to set board ID.

BoardID Switch

PCI-1245/1245V/1245E/1265 have a built-in DIP switch (SW1), which is used to define each card's unique identifier for Motion Utility. You can determine the BoardID identifier on the register as shown in table 3.2. When there are multiple cards in the same chassis, this BoardID setting is useful for identifying each card's unique device number.

We set the BoardID switch to 0 at the factory. If you need to adjust it to another number, set SW1 by referring to table 3.2.

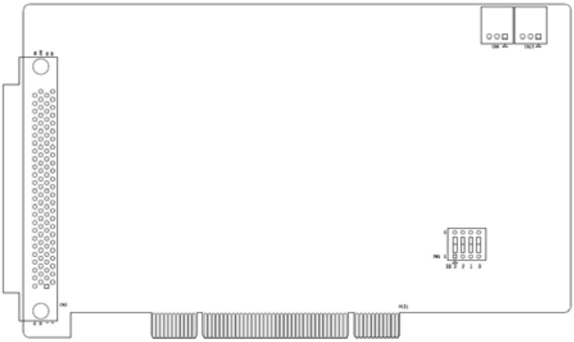


Figure 3.3: Location of Jumpers & DIP Switch

Table 3.2: BoardID Setting				
Board ID Setting (SW1)				
Board ID (Dec.)	**Switch Position**			
	ID3 (1)	ID2 (2)	ID1 (3)	ID0 (4)
*0	●	●	●	●
1	●	●	●	❍
:				
14	❍	❍	❍	●
15	❍	❍	❍	❍
❍= Off	●= On	* = default		

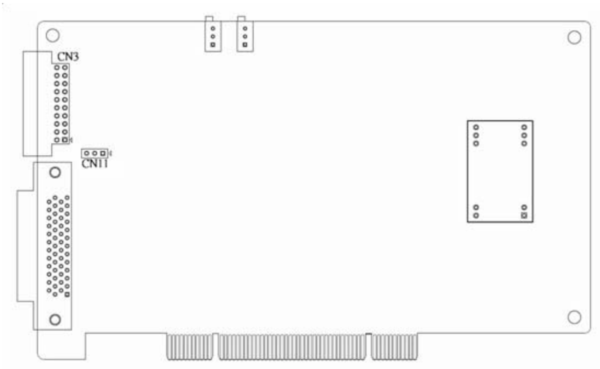


Figure 3.4: Location of Jumpers (Daughter Board)

Figure 3.4: Location of Jumpers & DIP Switch

Table 3.3: Jumper Settings	
	Sets CN3 (PIN 18) to DO7 (default)
	Sets CN3 (PIN 18) to VEX

3.3 Output Pulse [CW± / PULS±,CCW± / DIR±]

The pulse command has two types: One is in clockwise/ counter-clockwise mode; the other is in pulse/direction mode. CW+ / PULS+ and CW- / PULS- are differential signal pairs and CCW+ / DIR+ and CCW- / DIR- are differential signal pairs. Default setting of pulse output mode is pulse/direction. User can change the output mode by programming.

Figure 3.5: Photocoupler Interface

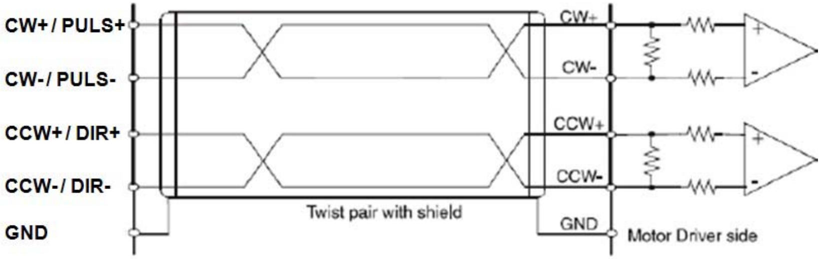


Figure 3.6: Line Drive Interface

3.4 Over Traveling Limit Switch Input [LMT+/-]

Over traveling limit switches are used for system protection. This input signal is connected through the connection of photo coupler and RC filter. When the limit switch is applied, the external power VEX DC 12 ~ 24 V will be the source of the photo coupler. This enables the over traveling function.

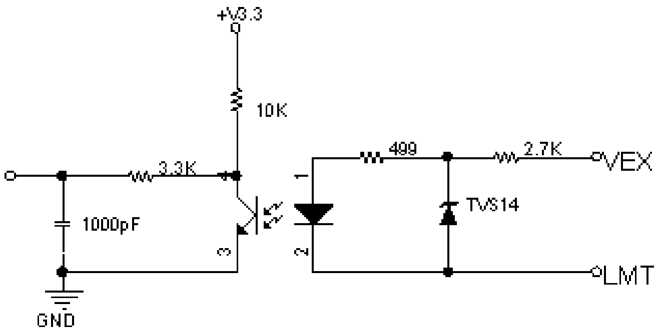


Figure 3.7: Circuit Diagram for Limit Input Signals

3.5 Position Latch [LTC]

It is a general purpose input pin which is used to latch the simultaneous position information. Users can read the position counter by programming. For detailed information, refer to chapter 6.

3.6 Servo Ready Signal [RDY]

It is a general purpose digital input which is used to check the servo ready status from servo drive connection. For example, you can check the status before any command is issued. Users can also use this RDY as general purpose input for other usages.

3.7 Home Position [ORG]

Home position is to define the original position or home signal for each axis. refer to chapter 6 for programming settting.

3.8 In-Position Singal [INP]

The In-Position range (or deviation) is usually defined by servo drive. When the motor moves and converges within this range (or deviation), the servo driver will send the signal out to indicate that the motor is in the defined position.

3.9 Servo Error & Alarm [ALM]

This input is from servo drive which will generate the alarm signal to indicate any operation error.

3.10 Encoder Input [ECA+/-, ECB+/-, ECZ+/-]

When the feedback encoder signals arrive, connect ECA+/ECA- to phase A of encoder output. It is a differential pair. The same rule is for ECB+/- and ECZ+/-. The default setting of PCI-1245/1245V/1245E/1265 is quadrature input (4xAB phase). The following diagram shows the interface circuit for one channel:

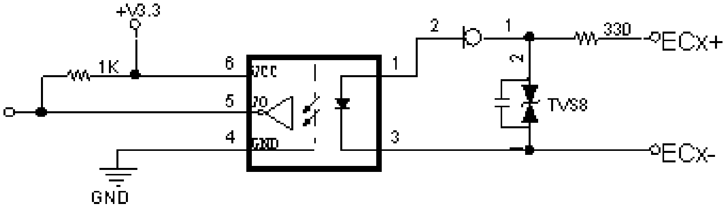


Figure 3.8: Circuit Diagram of Encoder Feedback

In the circuit diagram above, PCI-1245/1245V/1245E/1265 use high speed photo coupler for isolation. The source's encoder output can be differential mode or open-collector mode. And the maximum acceptable 4xAB phase feedback frequency is about 10 MHz.

3.11 Emergency Stop Input (EMG)

When emergency stop input signal is enabled, the output of the drive pulse for all axes will be stopped.

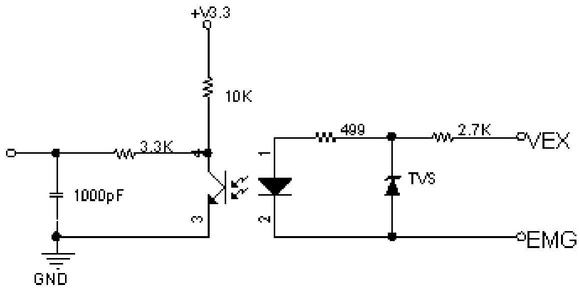


Figure 3.9: Circuit Diagram of Emergency Stop Input Signal

This signal should be used in combination with external power DC 12 ~ 24 V. The response time of circuitry should take about 0.25 msec because of the delay of photo coupled and RC filter.

3.12 External Power Input (VEX)

External power is necessary for all input signals of each axis. Apply DC 12 ~ 24 V voltage as required.

3.13 Position Window Output [CAM-DO]

As the following figure shows, users can define the interval and level to generate a digital output with a defined duration.

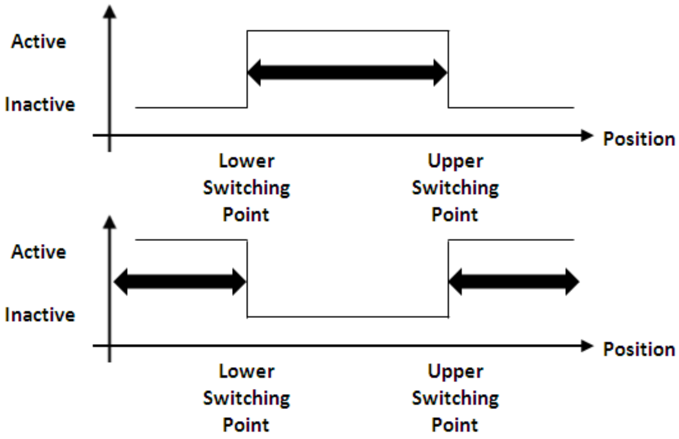


Figure 3.10: Circuit Diagram of Position Window Output

3.14 Activate Servo ON [SVON]

This SVON is to generate a digital output to activate the servo drive to be ready for move status.

3.15 Servo Error Counter Clear [ERC]

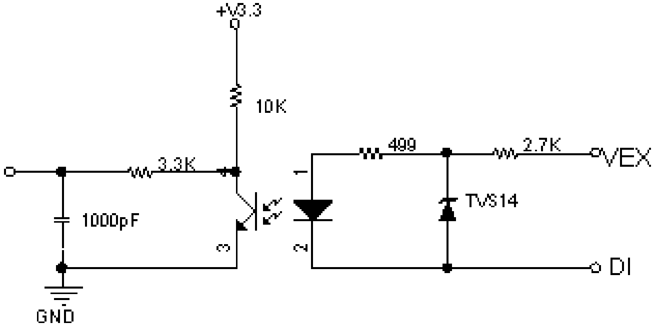
The deviation counter clear is generated by servo drive and the board can receive it as a general purpose input. The counter will be cleared by some instances: homing, emergency stop case, servo alarm and over travelling limit activated.

3.16 Position Compare Output [CMP]

This is specially designed for the customers who can use the position compare output to synchronize with other 3rd party vision devices. For PCI-1245/1245V/1245E/1265, the position compare output channel is determined by pin definition - CMP.

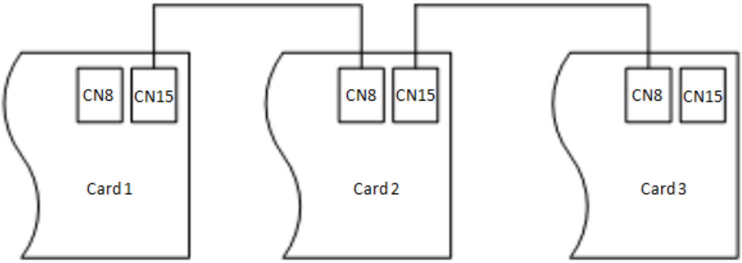
3.17 JOG and MPG

The JOG and MPG mode could be supported by pin assignment - X_IN4 & X_IN5. These two pins could be switchable. X_IN4 has three functions: general purpose digital input, JOG+ and MPG_A. X_IN5 also has three functions: general purpose digital input, JOG- and MPG_B. The circuit is as follow:



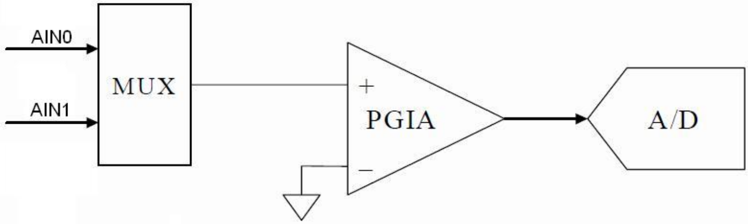
3.18 Simultaneous Start and Stop within Multiple Cards

Simultaneous start and stop within multiple cards is supported by connecting the CN8 and CN15 on each card one by one. For the function call of simultaneous start and stop, refer to chapter 6.

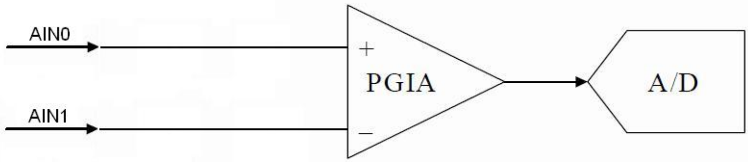


3.19 Analog Input

Two analog input channels are only supported for PCI-1265 cards. For connections, refer to the following diagram. For API, you can refer to 6.3.3.2 for more information.



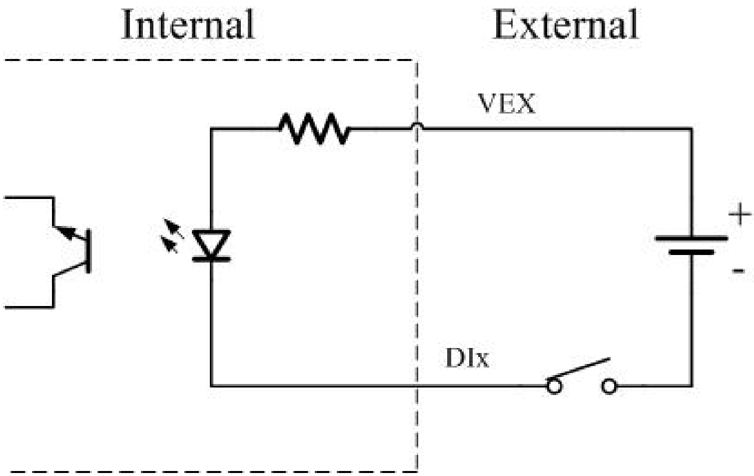
Single ended

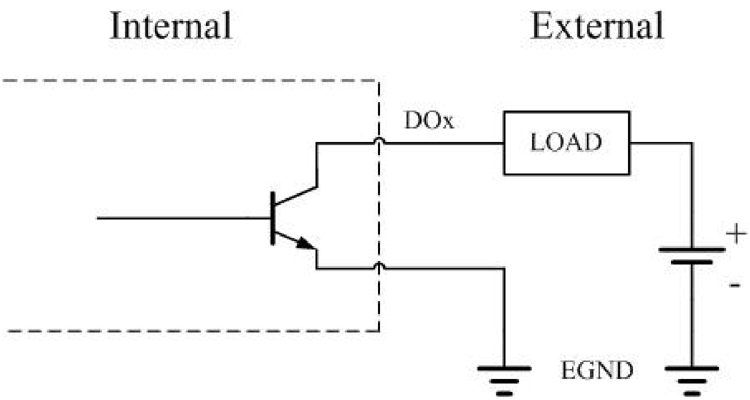


Differential

3.20 Digital Input and Output

PCI-1265 supports general purposed digital inputs and digital outputs. For connections, refer to the following diagram. For API, you can refer to 6.3.3.1 for more information.





2

CHAPTER 4

Common Motion API

This chapter introduces common motion API architecture & concept.

Chapter 4 Common Motion API

4.1 Introduction of Common Motion Architecture

In order to unify user interfaces of all Advantech motion devices, new software architecture is designed for all Advantech motion devices which is called “Common Motion Architecture”. This architecture defines all user interfaces and all motion functions that are implemented, including single axis and multiple axes. This unified programming platform enables users to operate devices in the same manner.

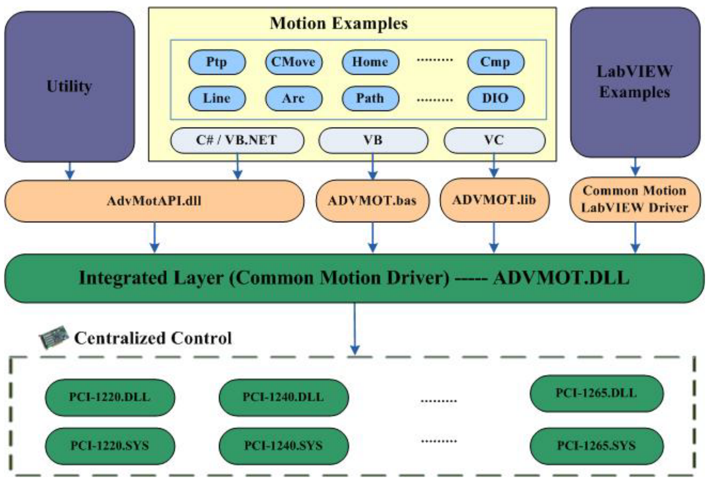
There are three layers in this architecture: Device Driver Layer, Integrated Layer and Application Layer. Users do not need to know how to operate the specific driver of a specify device, but only to know the Common Motion Driver. Even though the device which supports this architecture has changed, the application does not need to be modified.

Advantech Common Motion (ACM) Architecture defines three types of operation objects: Device, Axis and Group. Each type has its own methods, properties and states.

To start single axis motion, you have to follow the following steps:

Open device->open one axis of this device->configure instance of this axis->start motion.

All operations can be done by calling corresponding ACM APIs. General calling flows of Device, Axis, Group are specified by Common Motion Architecture. For detailed information, refer to the **Calling Flow section**.



4.2 Device Number

Device number is composed of 32 bits:

4th byte	3rd byte	2nd H byte	2nd L byte	1st byte
Master/device type ID	Master/device board ID (or BaseAddr)		Ring	Slave Board ID

- 4^{th} byte
  Master/device type ID (refer to master device type ID table))
- 3^{rd} & 2^{nd} H byte:
  Master/device board ID (or base address)
- 2^{nd} L byte:
  Master ring number, used by remote device, use 0 as default value for local device
- 1^{st} byte:
  Slave board ID, used by remote device, use 0 as default value for local device.

Local Device Number

4th byte	3rd byte	2nd H byte	2nd L byte	1st byte
Master type ID	Board ID (or BaseAddr)		0	0

For example, one BoardID of PCI-1245 is 1, the device number (Hexadecimal) is:

27	001	0	0

So the device number is 0x27001000.

4.3 Naming Rules of API and Properties

The naming rule is based on three objects: Device Object, Axis Object and Group Object. User will find many abbreviations in APIs. Table of abbreviations and their meanings is as follow:

Table 4.1: Abbreviations and Their Meanings

Abbreviations	Full Name	Comments
PPU	Pulse Per Unit	A virtual unit of motion
Dev	Device	
Ax	Axis	
Gp	Group	Multiple axes
Mas	Master	Master Axis or Master Board of device based on communicating mechanism
Daq		Common name of AI/AO/DI/DO
Rel	Relative	
Abs	Absolute	
Cmd	Command	
Vel	Velocity	
Acc	Accelerate	
Dec	Decelerate	
Emg	Emergency	Emergency stop
Sd	Slow down	
Info	Information	
Cmp	Compare	
Inp	In position	
EZ	Encode Z	
El	Hardware Limit	
Mel	Negative Limit	
Pel	Positive Limit	
Org	Origin	
Ext	External	
FT	Feature	Feature properties
CFG	Configuration	Configuration properties
PAR	Parameter	Parameter properties

Table 4.1: Abbreviations and Their Meanings		
Ipo	Interpolation	
Chan	Channel	

Naming Rules of API

The naming rules of API are as follows:

- Acm_DevXXX: Represents this API will implement function for device, such as device properties setting. Eg.Acm_DevSetProperty.
- Acm_DaqXXX: Represents this API will implement the function of DI, DO, AI or AO. Eg.Acm_DaqDiGetByte.
- Acm_AxXXXX: Represents this API will implement function for axis, such as single axis motion, homing. Eg. Acm_AxHome.
- Acm_GpXXXX: Represents this API will implement function for multiple axes. Such as interpolation motion. Eg. Acm_GpMoveLinearRel.

Naming Rules of Property

The properties have three types: feature, configuration and parameter.

Feature: Feature properties are related to the hardware features. The naming rules are as follows:

- FT_DevXXX: For device. Eg. FT_DevAxisCount.
- FT_DaqXXX: For DI, DO, AI, and AO. Eg. FT_DaqDiMaxChan.
- FT_AxXXX: For axis object. Eg. FT_Ax
- FT_GpXXX: For group object.

Configuration: The values of configuration properties may change, but not frequently.

- CFG_DevXXX: For device. Eg. CFG_DevBoardID.
- CFG_AxXXXX: For axis. Eg. CFG_AxMaxVel.
- CFG_DaqXXX: For DI, DO, AI and AO. Eg. CFG_DaqDiMaxChan.
- CFG_GpXXXX: For group object. Eg. CFG_GpAxisInGroup.

Parameter: The values of parameter properties may change frequently.

- PAR_DevXXX: For device.
- PAR_AxXXXX: For axis. Eg. PAR_AxVelLow.
- PAR_DaqXXX: For DI, DO, AI and AO.
- PAR_GpXXXX: For group. Eg. PAR_GpGroupID.

2

CHAPTER 5

Utility

This chapter is to describe the comprehensive & graphical utility

Chapter 5 Utility

5.1 Introduction

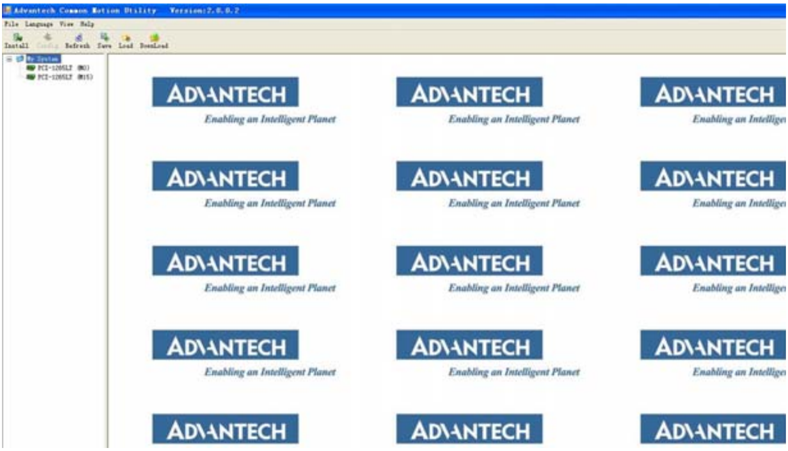
The utility is developed with .Net control library according to Common Motion API architecture. The .Net control library includes control - Device, Axis, Group and component - AxisSetupView, AxisScopeView, AxisDiagView, GroupPathView and GroupSpeedView. The new utility is consistent and compatible with old AdvMotionUtility. The new utility supports PCI-1220U, PCI-1240U, PCI-1245/1245V/1245E/1265 series products.

5.1.1 Contents

Mainly according to the order of operations, the following interfaces will be introduced:

1. Main Form: includes Main Menu, Toolbar and Device Tree.
2. Single-axis Motion: focuses on the I/O and attribute configuration, and status and movement operations (P to P/ Continue/ Homing) of single axis.
3. Multi-axis Motion: focuses on multi-axis (Group) interpolation operation, including the basic interpolation (Line/Arc /Helix), continuous interpolation (Path) and tangent follow motion.
4. Synchronized Motion: focuses on synchronized motion operations, including electronic CAM (E-CAM), electronic gear (E-Gear) and gantry (Gantry) movement.
5. Digital Input: displays device's input status.
6. Digital Output: displays device's output status.
7. Analog Input: displays device's analog input status.

5.2 Main Form



5.2.1 Main Form

5.2.1.1 File

Click [Exit] to terminate this process.

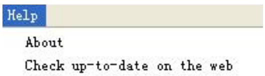
5.2.1.2 Language

Through this menu, language in Utility can be switched. This utility supports three languages: English, simplified Chinese and traditional Chinese. After you select a language, the corresponding menu item will be checked. When you close the Utility, the language you selected will be saved to register. When opened next time, the utility's language will be last used one.

5.2.1.3 View

This menu allow users to display/hide the toolbar, status bar and device tree. If Toolbar/Status Bar/Device Tree is visible, the corresponding menu item will be checked.

5.2.1.4 Help

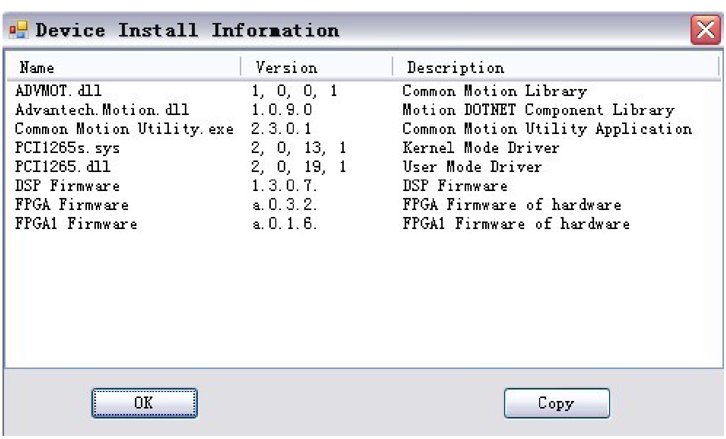


The [About] menu item supports the copyright notice of the driver and utility for device. Click [Check up-to-date on the web], you can link to company's website to check whether the firmware, driver and utility are the latest ones by comparing version information of Install interface.

5.2.2 Toolbar

5.2.2.1 Install

Click [Install], a new window will pop up as below, which shows the version information of driver, hardware, firmware and utility.



Click "Copy", the first two raw information will be copied. You can paste the information to editor, such as word or text editor.

First column is name, the second column is version number and the third column is description. ADVMOT.dll is the common interface for development. Advantech.Motion.dll is the .NET motion control library. Common Motion Utility.exe is the utility which is running now. The fourth and fifth lines are driver files (Kernal-Mode and User-Mode), which depends on device type; the sixth line is DSP firmware and the seventh line is FPGA of the hardware.

Note: Only DSP-based motion control card will show the DSP and FPGA information.

5.2.2.2 Refresh

This button supports refresh function. Click [Refresh], Device Tree will re-load the Device. No device is selected by default after operation.

5.2.2.3 Save

This button can save all properties of the axes of the selected device.

5.2.2.4 Load

This button can import configurations of all axes of the selected device. After the device is selected, click the button, an Open Dialog box will appear. Select the previously exported configuration file and click [OK], you can import the configuration file into the Device hardware.

5.2.2.5 Download

For PCI-1245/1245V/1245E/1265 series, it's DSP-based motion controller. After clicking device, you can see the interface as follows.

The tool is to upgrade the DSP and FPGA firmware. FPGA firmware is for footprint, U2 and U7 respectively (U2 and U7 are printed in PCB). Both download and upgrade procedure are the same, but you shall be aware that the PC is necessary to reboot after FPGA firmware upgraded. Then, the new FPGA firmware will be truly updated.

The top of this dialog shows the current device type, name and firmware version. Click [Open File] to select lastest firmware file you have acquired. Clicking [Start Download] will activate the downloading procedure to hardware and progress bar will show the task process.

Note: 1. *After clicking [Start Download], the dialog cannot be terminated when downloading the firmware to hardware.*

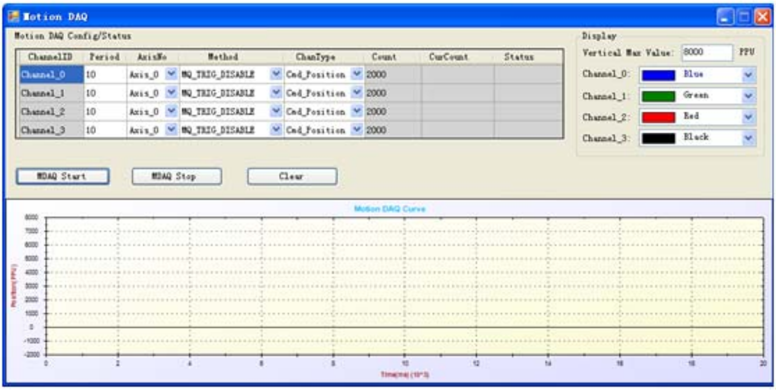
2. *While downloading, due to power outages or other problems, if download process is not complete, the hardware needs to be sent back to Advantech for firmware update.*

5.2.2.6 Motion DAQ

The tool is mainly to show Motion Data Acquisition function. In PCI-1245/1245V/1245E/1265, four channels are provided for motion data acquisition. Each of them can acquire Command/Actual/Lag (the difference between Command and Actual) motion data of any axis, with the max. data count of 2000.

Note: *So far, only PCI-1245, PCI-1245V, PCI-1245E and PCI-1265 support this function and the tool button is available.*

After you click the button, you'll see the interface as below:



This interface consists of the following parts:

1. Motion DAQ Config/Status
   You can directly configure acquisition data of each channel in DataGridView. ChannelID, Count, CurCount and Status row are read-only (the background is grey).
   ChannelID: It means a channel ID. Four channels (Channel_0, Channel_1, Channel_2 and Channel_3) are provided.
   AxisNo: Any axis of the device can be selected.
   Period: Acquisition period, which means the interval between each data is acquired. The range is 1 to 255 ms. In order to unify the max value of horizontal ordinate of Curve window, Period value of each channel will adopt the same value. Therefore, if Period value of one channel has been changed, that of other channels will change accordingly.
   Method: Trigger mode. Trigger modes of data acquisition are as followings:
   0: MQ_TRIG_DISABLE: Disable data acquisition function;
   1: MQ_TRIG_SW: Trigger by software (Click MDAQ Start to trigger);
   2: MQ_TRIG_DI£½Trigger by DI (reserved);
   3: MQ_TRIG_AX0_START: Trigger when axis 0 starts to move;
   4: MQ_TRIG_AX1_START: Trigger when axis 1 starts to move;
   5: MQ_TRIG_AX2_START: Trigger when axis 2 starts to move;
   6: MQ_TRIG_AX3_START: Trigger when axis 3 starts to move;
   7: MQ_TRIG_AX4_START: Trigger when axis 4 starts to move;
   8: MQ_TRIG_AX5_START: Trigger when axis 5 starts to move;
   9: MQ_TRIG_AX6_START: Trigger when axis 6 starts to move;
   10: MQ_TRIG_AX7_START: Trigger when axis 7 starts to move;
   11: MQ_TRIG_AX8_START: Trigger when axis 8 starts to move;
   12: MQ_TRIG_AX9_START: Trigger when axis 9 starts to move;
   13: MQ_TRIG_AX10_START: Trigger when axis 10 starts to move;
   14: MQ_TRIG_AX11_START: Trigger when axis 11 starts to move;

   So far in PCI-1245, PCI-1245V, only 0-6 trigger methods are supported; In PCI-1265, only 0-8 trigger modes are supported. Moreover, trigger by DI is reserved.
   ChanType: The source of data acquisition, the available values of which are:
   0: Cmd_Position: Command Position;

1: Act_Position: Actual Position;
2: Lag_Position: Lag Position, which means the difference between Command Position and Actual Position;
3: Cmd_Velocity (reserved): Command Velocity.
Count: Count of acquired data, the range of which is 0-2000. The max value is 2000 by default in Utility.
CurCount: Acquired data count will be returned after motion data is acquired.
Status: Display current acquisition status:
0: Ready: Data acquisition function is not started yet;
1: Wait Trigger: Data acquisition function is started, but waits for trigger; 2: Started: Motion data is being acquired.

After the mouse moves away from the edit box, the setting values will take effect. You can check current configurations of each channel in DataGridView, which is shown as below:

Motion DAQ Config/Status

ChannelID	Period	AxisNo	Method	ChanType	Count	CurCount	Status
Channel_0	10	Axis_0	MQ_TRIG_DISABLE	Cmd_Position	2000		
Channel_1	10	Axis_0	MQ_TRIG_DISABLE	Cmd_Position	2000		
Channel_2	10	Axis_0	MQ_TRIG_DISABLE	Cmd_Position	2000		
Channel_3	10	Axis_0	MQ_TRIG_DISABLE	Cmd_Position	2000		

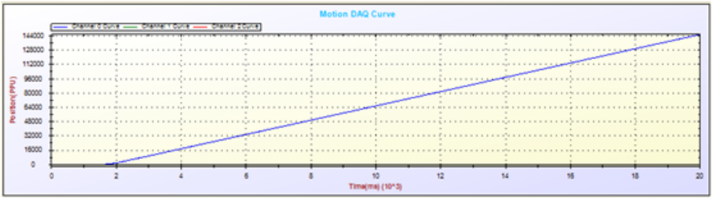
2. Function Opeartions
MDAQ Start: Start motion data acquisition function. When trigger conditions are met, the acquisition of motion data will be started;
MDAQ Stop: Stop motion data acquisition;
Clear: Clear curve of each channel.

3. Curve Display
When MDAQ Start is started, data acquisition will be started when trigger conditions are met. You can check current sample count the status of each channel, which is shown as below:

Motion DAQ Config/Status

ChannelID	Period	AxisNo	Method	ChanType	Count	CurCount	Status
Channel_0	10	Axis_0	MQ_TRIG_SW	Cmd_Position	2000	337	Started
Channel_1	10	Axis_0	MQ_TRIG_SW	Act_Position	2000	338	Started
Channel_2	10	Axis_0	MQ_TRIG_SW	Lag_Position	2000	338	Started
Channel_3	10	Axis_0	MQ_TRIG_DISABLE	Cmd_Position	2000	0	Ready.

When data acquisition is finished, the curve of corresponding acquisition data of each channel will be displayed in the below picture box, which is shown as bleow:



4. Display
   Display area in the uppper right corner is to configure the color of each channel curve and the max value of vertical coordinate of picture box. Select a color from combo box, then the color of corresponding curve will be changed.

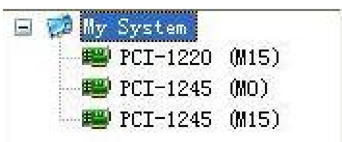
5.2.2.7 Hide Tree

This button is provided to hide/show Device Tree.

If Device Tree is currently shown, click the button to hide it and the text on the button will change to "Show Tree".

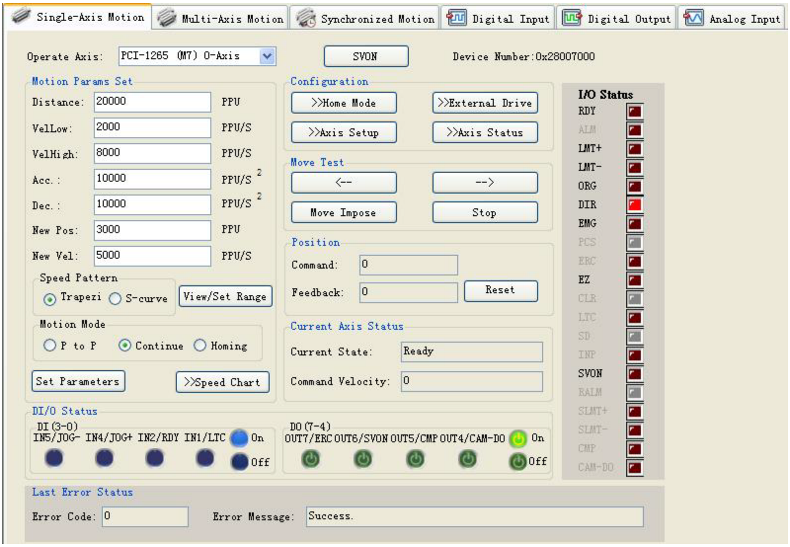
If Device Tree is currently hided, click the button to show it and the text on the button will change to "Hide Tree".

5.2.3 Device Tree



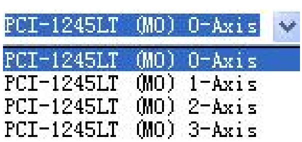
Click any device of tree view; you will see the operation interface.

5.3 Single-Axis Motion



5.3.1 Operate Axis

Select the operating axis. Click the check box drop-down symbol, all axes of the selected device will display as follows:



5.3.2 Motion Params Set

After finishing the parameter setting for operation, click [Set Parameters] to save the values to device.

5.3.2.1 Basic Parameter Setup

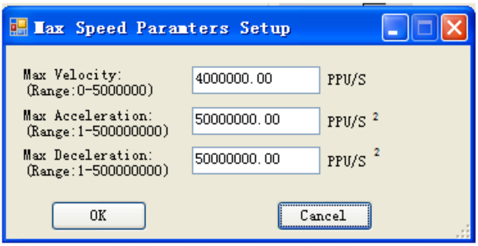
It's mainly about the settings of distance(Distance) in point to point movement, initial velocity (VelLow), movement velocity (VelHigh), acceleration (Acc.) and deceleration (Dec.) in single-axis motion, movement distance (New Pos.) and velocity (New Vel) in superimposed movement (Move Impose).

5.3.2.2 Speed Pattern

Set the speed pattern of movement, which can be trapezoidal pattern (Trapezi) or S-type (S-curve).

5.3.2.3 View/Set Range

Click [View/Set Range] to check or set the maximum velocity, acceleration and deceleration. The dialog will show as follow.



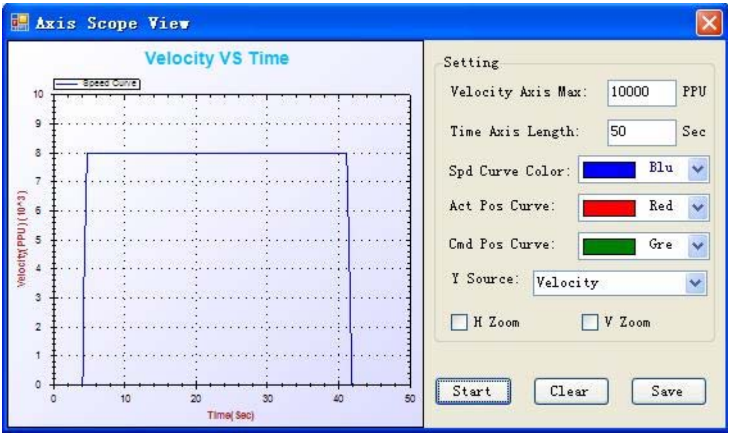
Note: *VelHigh in Single-axis Motion can not be greater than the Max Velocity; Acc. can not be greater than the Max Acceleration and Dec. can not be greater than the Max Deceleration.*

5.3.2.4 Move Mode

Select Move Mode. There are three move modes in single-axis motion: P to P (point to point motion), Continue (constant-speed continuous motion), Homing (homing motion).

5.3.2.5 Speed Chart

By clicking [Speed Chart], you can see the velocity curve.

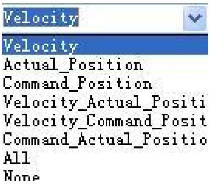


Wherein, on the right there are setting and operating buttons, on the left there is movement/speed curve in single-axis motion.

5.3.2.5.1 Setting

Setting items are as follows:

1. Vertical Max Value: sets maximum vertical coordinate.
2. Time Length Value: sets maximum horizontal coordinate.
3. Spd Curve Color: setsthe color for speed curve.
4. Act Pos Curve: sets the color for actual position curve.
5. Cmd Pos Curve: sets the color for command position curve.
6. Y Source: data source for vertical coordinate. You can select any one or any combination of velocity, command position and actual position as below.



7. H Zoom: if it is checked, it indicates horizontal zoom is enabled, you can select appropriate region by the mouse to zoom in.
8. V Zoom: if it is checked, it indicates vertical zoom is enabled, you can select appropriate region by the mouse to zoom in.

After the setting item is edited, the value will become effective as the mouse leaves the edit box.

5.3.2.5.2 Start

Click [Start], the graphic box will be ready to draw the curve, if the axis is in motion, you can see the trajectory. After clicked, the text on [Start] button will change into "Stop"; click [Stop], drawing the curve will stop and the text will back to "Start".

5.3.2.5.3 Clear

Click [Clear], the current curve in graphic box will be cleared.

5.3.2.5.4 Save

Click [Save], the specified path curve will be saved as .png, .gif, .jpg, .tif or .bmp format.

5.3.3 SVON

Click [SVON], the servos of axes will be turned on and the text on it will change into "SVOFF"; click [SVOFF], the servos of axes will be turned off and the text on it will be back to "SVON".

5.3.4 Configuration

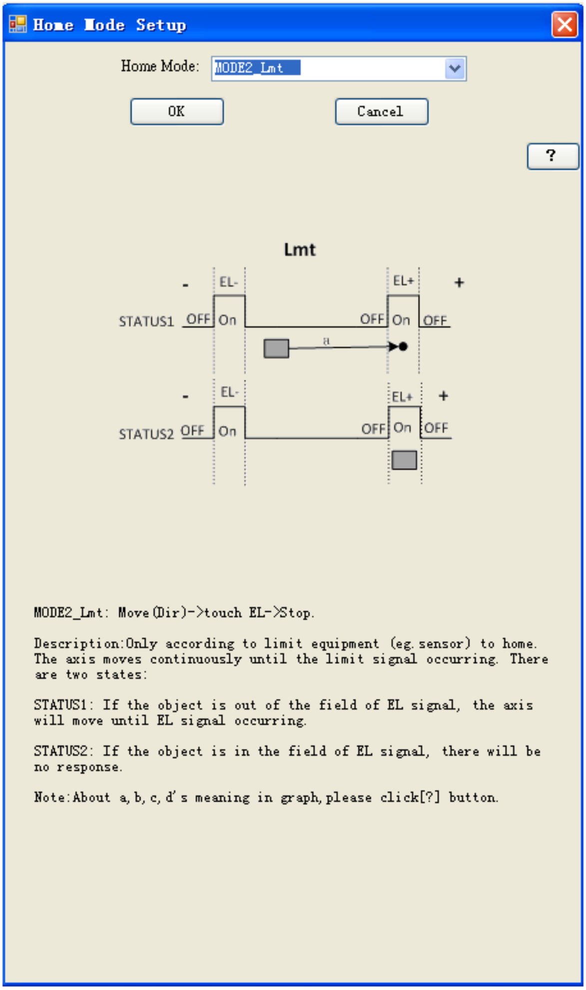
It includes Home Mode configuration, External Drive mode, the property configuration and I/O status of the axis.

5.3.4.1 Home Mode

Before performing home movement, you need to select the mode first. Board offers 16 modes, which are any one or combination of the ORG (back to the origin), Lmt (back to the limit point) and EZ (to find Z-phase).

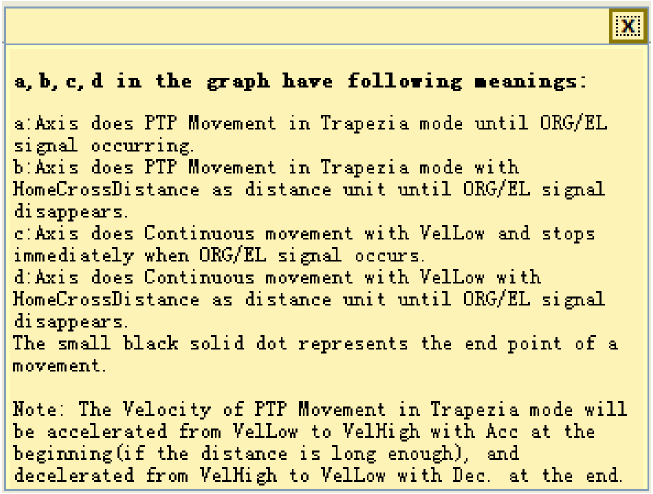
For detailed information, refer to the description about Home Mode in Common API of Programming guide.

Click [Home Mode], a new form appears as below:



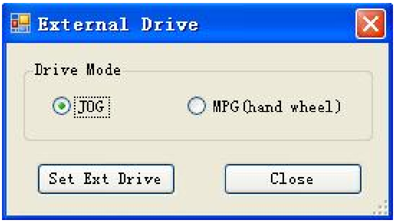
You can select any mode listed in the comobox, there is corresponding illustration below. You can click [OK] to select the mode in the HomeMode combobox, or click [Cancle] to cancle the operation. The default setting is "Mode1_Abs".

Click “?”, the pop-up dialog will show up. the dialog will give the explanation for the parameters in the home mode. The example figure is as follows:



5.3.4.2 External Drive

Click [External Dirve], a new form will appear as below, you can select an external drive mode (JOG/MPG) to operate external drive.

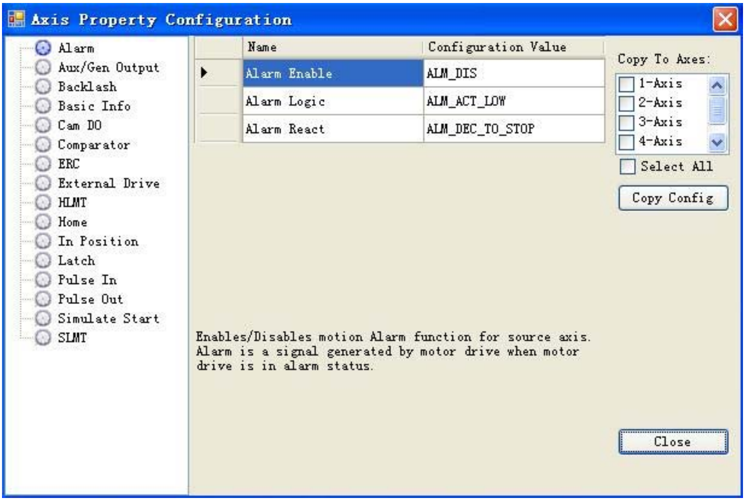


Select JOG or MPG and click [Set Ext Drive], the external drive mode will be set and you can operate external drive then. Click [Close], the form will be closed and the external drive is set to “Disable”.

Note: *For PCI-1245/1245V/1245E/1265series, only axis 0 is available for external drive as master axis.*

5.3.4.3 Axis Setup

Click the button to check/set the axis's attributes and I/O as follows:



The left tree view shows the classification of axis's properties, when you click the corresponding item, the right side, Data View, will list the properties and corresponding property values in the category. For detail, refer to the description about Feature, Configuration and Parameter of axis which are listed in property list of Programming guide.The attributes are classified as follows:

Classification	Name	Brief Introduction
Alarm	Alarm Enable	Enables/Disables motion Alarm function for source axis.
	Alarm Logic	Sets the active logic for alarm signal.
	Alarm React	Sets the reacting mode for alarm signal.

Aux/Gen Output	AuxOut Enable	Enables/Disables axis's Aux-Output in group's AddPathDwell() for source axis.
	AuxOut Time	Sets axis's Aux-Output on time in group's AddPathDwell() for source axis.
	GenDo Enable	Enables/Disables axis DO as general DO function for source axis.
Backlash	Backlash Enable	Enables/Disables corrective backlash for source axis.
	Backlash Pulses	Sets the compensation pulse numbers for source axis.Whenever direction change occurs, the axis outputs backlash corrective pulses before sending commands.
	Backlash Velocity	Sets the velocity for backlash signal.
Basic Info	PhyID	The physical ID of source axis.
	PPU	The pulse per unit(PPU) of source axis.It is a virtual unit.You can set PPU according to actual motor.This can mask the different precision of different motors.
	ModuleRange	Sets the module range for this axis.
Cam DO	CamDO Enable	Enables/Disables CAM DO function for source axis.
	CamDO Logic	Sets the active logic for CAM DO signal.
	CamDO Compare Source	Sets the compare source for CAM DO signal.
	CamDO Mode	Sets the mode for CAM DO signal.
	CamDO Direction	Sets the direction for CAM DO.
	CamDO Low Limit	Sets the low limit for CAM DO signal.
	CamDO High Limit	Sets the high limit for CAM DO signal.

Comparator	Compare Enable	Enables/Disables axis comparator for source axis.
	Compare Source	Sets the source for comparator.
	Compare Method	Sets the method for comparator.
	Compare Pulse Mode	Sets the pulse mode for comparator.
	Compare Pulse Logic	Sets the active logic for comparator's pulse.
	Compare Pulse Width	Sets the pulse width for comparator.
ERC	Erc Logic	Sets the active logic for ERC signal.
	Erc On Time	Sets the on-time length for ERC active.
	Erc Off Time	Sets the off-time length for ERC active.
	Erc Enable Mode	Enables/Disables ERC Output for source axis.
External Drive	Ext Master Src	Indicates that axis is controlled by which physical axis's external signal.
	Ext Sel Enable	When Ext.drive is enabled, this property enables driving axis selection by digital input channel.
	Ext Pulse Num	The number of output driving pulses when an active edge of input pulse is accept in Hand Wheel mode.
	Ext Preset Num	The number of output driving pulses when an active edge of input pulse is accept in JOG mode.
	Ext Pulse In Mode	Sets the pulse input mode for external drive.
HLMT	HLMT Enable	Enables/Disables the hardware limit signal.
	HLMT Logic	Sets the active logic for hardware limit signal.
	HLMT React	Sets the reacting mode for hardware limit signal.

Home	Home Ex Mode	Sets the stopping modes for HomeEx().
	Home Cross Distance	Sets the home cross distance (Unit: Pulse) for homing.
	Home Ex Switch Mode	Sets the stopping condition for HomeEx().
	ORG Logic	Sets the active logic for ORG signal.
	EZ Logic	Sets the active logic for EZ signal.
	Home Reset Enable	Enables/Disables reset logical counter after homing for source axis.
	ORG React	Sets the reacting mode for ORG signal.
In Position	Inp Enable	Enables/Disables In-Position function for source axis.
	Inp Logic	Sets the active logic for In-Position signal.
Latch	Latch Enable	Enables/Disables latch function for source axis.
	Latch Logic	Sets the active logic for latch signal.
Pulse In	Pulse In Mode	Sets the encoder feedback pulse input mode for source axis.
	Pulse In Logic	Sets the active logic for encoder feedback pulse input signal.
	Pulse In Source	Sets the source for encoder feedback pulse input signal.
	Pulse In Max Frequency	Sets the maximum frequency of encoder pulse input signal.
Pulse Out	Pulse Out Mode	Sets the command pulse output mode for source axis.
SD	SD Enable	Enables/Disables the SD signal for source axis.
	SD Logic	Sets the active logic for SD signal.
	SD React	Sets the reacting mode for SD signal.
	SD Latch	Sets the latch control for SD signal.

Simulate	Simulate Start	Sets the simulate start source for this axis.
SLMT	SLMT Mel Enable	Enables/Disables the minus software limit for source axis.
	SLMT Pel Enable	Enables/Disables the plus software limit for source axis.
	SLMTN React	Sets the reacting mode for minus software limit.
	SLMTP React	Sets the reacting mode for plus software limit.
	SLMTN Value	Sets the value for minus software limit.
	SLMTP Value	Sets the value for plus software limit.
Speed Pattern	Max Velocity	Configures the max velocity for source axis.
	Max Acc	Configures the max acceleration for source axis.
	Max Dec	Configures the max deceleration for source axis.
	Max Jerk	Configures the max jerk for source axis.
	Vel Low	Sets the low velocity (start velocity) for source axis (Unit: PPU/S).
	Vel High	Sets the high velocity (driving velocity) for source axis (Unit:PPU/S).
	Acc	Sets the acceleration for source axis (Unit: PPU/S2).
	Dec	Sets the deceleration for source axis (Unit: PPU/S2)
	Jerk	Sets the type of velocity profile: t-curve or s-curve for source axis.

Vibration	Vibration Enable	Enables/Disables suppress vibration of mechanical system for source axis.
	Vibration Reverse Time	Sets the vibration suppressing timing for source axis.This function is used to suppress vibration of mechanical system by outputting single pulse for negative direction and then single pulse for positive direction right after completion of command movement.
	Vibration Forward Time	Sets the vibration suppressing timing for source axis.

Note1: *In the utility, if no corresponding functions of the selected device, the item will not shown in the left side Tree View. For example, if the selected device is PCI-1245/1265, and this board does not support slow down (SD) and vibration suppression function, then, you will not see the items in the Tree View. At the same time, because single axis dialog has speed parameter setting, the speed pattern item will not be shown.*

Note 2: When "Pulse Out" category is selected, there will be illustration of corresponding mode below the description of "Pulse Out Mode" property.

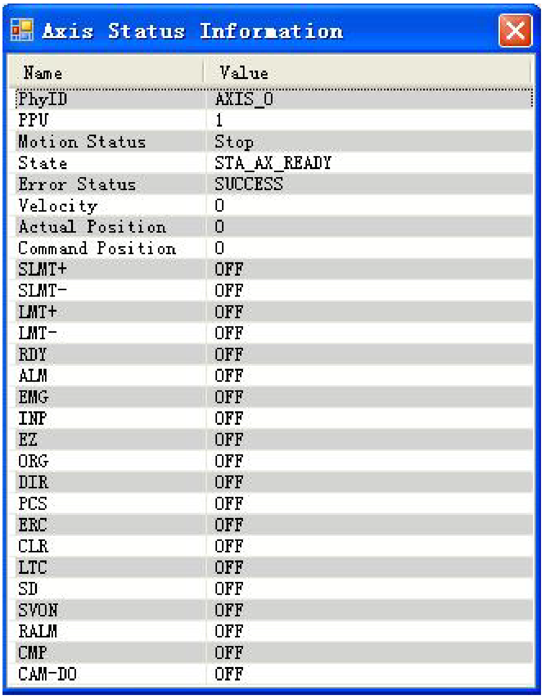
After editing, the property value will become effective (already set in device) after the mouse leaves the edit box.

If you want to duplicate the attributes to other axes, only activate the "Check" on the right side of check box. Then, click [Copy Config].

Click [Close] to close the form.

5.3.4.4 Axis Status

Click the button; you can view the assigned axis information. For example, PhyID, PPU, and basic status (Motion Status, State, Error Status and etc.) and I/O status (Alarm, SLMTP/N and etc.).

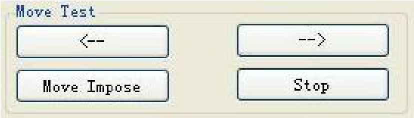


Axis Status Information

Name	Value
PhyID	AXIS_0
PPU	1
Motion Status	Stop
State	STA_AX_READY
Error Status	SUCCESS
Velocity	0
Actual Position	0
Command Position	0
SLMT+	OFF
SLMT-	OFF
LMT+	OFF
LMT-	OFF
RDY	OFF
ALM	OFF
EMG	OFF
INP	OFF
EZ	OFF
ORG	OFF
DIR	OFF
PCS	OFF
ERC	OFF
CLR	OFF
LTC	OFF
SD	OFF
SVON	OFF
RALM	OFF
CMP	OFF
CAM-DO	OFF

5.3.5 Move Test

The operation is as follows:

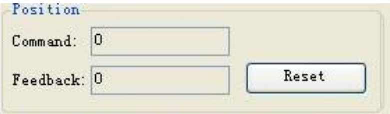


After motion mode is selected, click [<--] or [-->], the axis will do P to P/ Continue/Homing movement in negative or positive direction.

After the movement velocity reaches VelHigh in point to point motion, you can click [Move Impose] to generate a superimposed movement, the distance of the imposed movement is the value of New Pos and the velocity of the imposed movement is the value of New Vel. You can observe specific movement/speed curve through clicking [Speed Chart].

Click [Stop], the motion will be stopped.

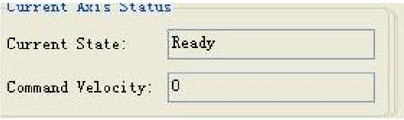
5.3.6 Position



By "Position" status, users can observe the command position and feedback position while in operation.

Click [Reset], you can reset the value to "0".

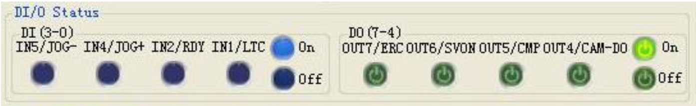
5.3.7 Current Axis Status



You can check the current status and command speed. For details, refer to the description about State in Acm_AxGetState function which is listed in Common API of Programming guide.

5.3.8 DI/O Status

Display the current status of 4 DI and 4 DO of the selected axis. You can also operate the DO to be ON/OFF.



5.3.8.1 DI

As the above figure, DI(3-0) status, from right to left is DI0 to DI3 respectively. Wherein, indicates the DI is in effect (ON) and its value is 1; indicates the DI is not in effect (OFF) and its value is 0.

5.3.8.2 DO

As the above figure, DO(7-4) status, from right to left is DO4 to DO7 respectively. Wherein, indicates the DO is in effect (ON) and its value is 1; indicates the DO is not in effect (OFF) and its value is 0.

5.3.9 Last Error Status

You can check the latest error code and error message. If there is no any error, the error code is "0", error message is "SUCCESS".

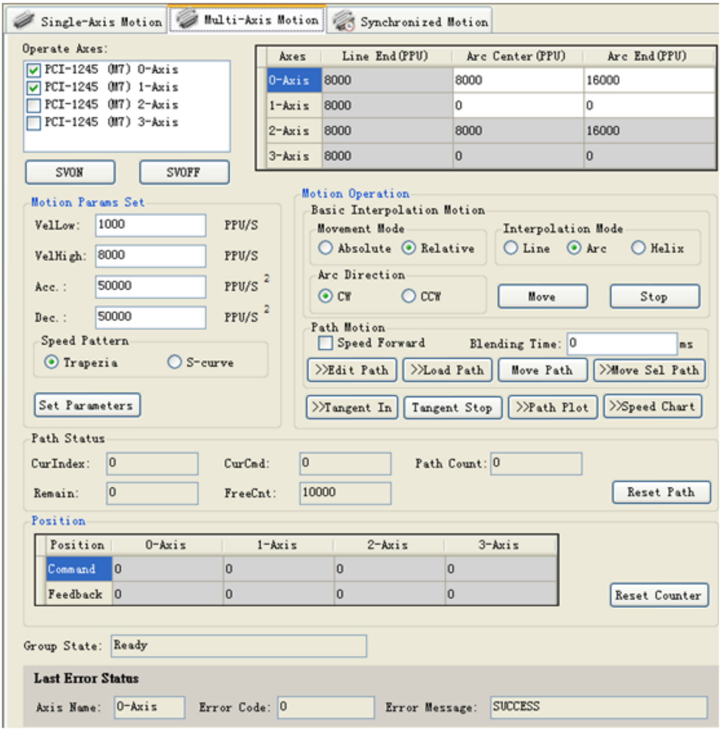
5.3.10 I/O Status

You can visually know the I/O status from the LED bar. Wherein, indicates the device does not support the function or does not have the corresponding I/O; indicates the device support the function, but I/O is not triggered (OFF); indicates the corresponding I/O is triggered (ON).

For details, refer to the description about Status in Acm_AxGetMotionIO function which is listed in Common API of Programming guide.

If no functional or no corresponding item, the text will be displayed as grey. If the board supporting the function, but not enable, the test is also displayed as grey. If the board supports this function and enable this function, then, the test will be display as normal.

5.4 Multi-Axes Motion



5.4.1 Operate Axes

The checkedListBox in the form will list all axes of the selected device, check the Checkbox of corresponding axis, you can add the axis into the Group. When the number of axis added to the Group is less than 2, Group's State will be "Disable". When the number of axis added to the Group is greater than or equal to 2, Group's State will be "Ready", then after you configure appropriate parameters, you can do appropriate interpolation operation.

5.4.2 Motion Params Set

The parameter set includes Group VelLow, Group VelHigh, Group Acc, Group Dec and Speed Pattern.

5.4.3 Motion Ends

Configure motion's center / end as follows.

Axes	Line End(PPU)	Arc Center(PPU)	Arc End(PPU)
0-Axis	8000	8000	16000
1-Axis	8000	0	0
2-Axis	8000	8000	16000
3-Axis	8000	0	0

The dialog will automatically enable the edit box writable by referring to group axis and interpolation mode. As in the Figure, 1-axis and 2-axis are added to Group and Line interpolation mode is selected, thus the edit boxes writable are "1-axis" and "2-axis" Lines of the "Line End (PPU)" column, whose background color is white. The edit boxes whose background color are gray indicate they are not editable.

5.4.4 Motion Operation

5.4.4.1 SVON

Click [SVON], the servos of axes in Group will be turned on.

5.4.4.2 SVOFF

Click [SVOFF], the servos of axes in Group will be turned off.

5.4.4.3 Basic Interpolation Motion

Basic interpolation motion includes linear interpolation (Line), circular interpolation (Arc) and helical interpolation (Helix) as follows.

5.4.4.3.1 Movement Mode

Absolute: the interpolation motion will directly use the set position parameters.

Relative: the interpolation motion will add initial offset to the position parameters and then use it.

5.4.4.3.2 Arc Direction

CW means clockwise.

CCW means counter clockwise.

5.4.4.3.3 Interpolation Mode

Line: linear interpolation

Arc: circular interpolation

Helix: helical interpolation

5.4.4.3.4 Move

After corresponding configuration, click [Move], Group will do the specified interpolation.

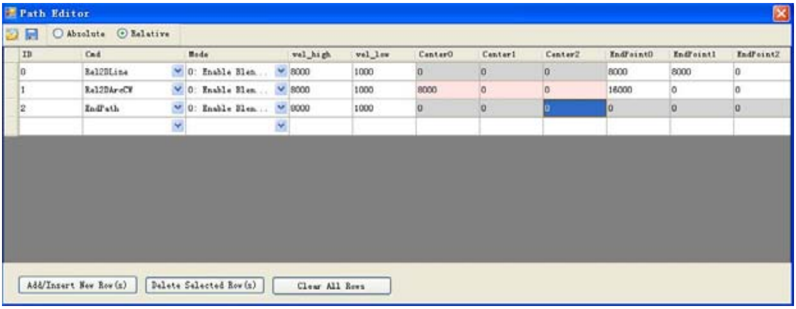
5.4.4.3.5 Stop

While Group is in interpolation motion, click [Stop], the interpolation motion will be stopped.

5.4.4.4 Path Motion

5.4.4.4.1 Edit Path

Click [Edit Path], the following form will be shown up:



Wherein, the top toolbox includes Open File, Save File and movement mode.

1. Open File : Open selected Path file from appropriate file path, which can be a binary file (.bin) or a comma-separated value file (.CSV).
2. Save File : save the edited data to a Path file, which can be a binary file (.bin) or a comma-separated value file (.CSV). CSV files can be opened in Excel, so can be checked / modified conveniently later. But if you want to run the Path through [Load Path], you need to save the data as .bin format, because currently device only supports .bin file to import through [Load Path].
3. Movement mode: Absolute or Relative. If you select “Absolute” , then commands listed in the Cmd column will be the commands related to absolute motion; Similarly, if you choose “Relative”, then commands listed in the Cmd column will be the commands related to relative motion.
4. Path edit items include command (Cmd), Mode (Blending/No Blending), movement velocity (vel_high), initial velocity (vel_low), center (Center), and end (EndPoint). Wherein, there are three axes (Center0/Center1/Center2) in circle interpolation by default, the number of EndPoint is according to the maximum number of axis supported in the selected device interpolation, such as in PCI-1245, the maximum number of axis supported is 4 in Direct interpolation motion, so the end point will be EndPoint0, EndPoint1, EndPoint2, and EndPoint3.
5. Add/Insert New Row(s): after clicking, the following dialog will be shown, you can edit the number of rows to be added/inserted.

   Click [OK], corresponding number of rows will be added after or inserted into the selected row.
6. Delete Selected Row(s): delete the selected rows.
7. Clear All Rows: clear all lines.

5.4.4.4.2 Load Path

Click [Load Path], if group's state is "Ready", you can import selected Path file (. bin format) from the Open File dialog box to the Device.

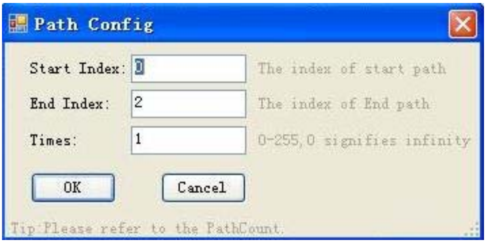
5.4.4.4.3 Move Path

After loading Path, if the edited Path is not wrong, The Path will be run one by one according to the serial number. You can observe movement curve and corresponding state from [Path Status], [Path Plot] and [Speed Chart].

5.4.4.4.4 Move Sel Path

Do continuous interpolation movement with path(s) selected from loaded Path file.

After loading Path, click [Move Sel Path] to activate the dialog:



1. Start Index: select the starting serial number of Path
2. End Index: select the end serial number of Path
3. Times: executable times. The value is 0 to 255. If you set 0, it means an infinite loop until you click "Stop" to terminate the loop.

5.4.4.4.5 Speed Forward

If the value is "Checked", the Group's speed-forward function will be enabled. For detail, refer to the description about CFG_GpSFEnable in Group which is listed in property list of Programming guide.

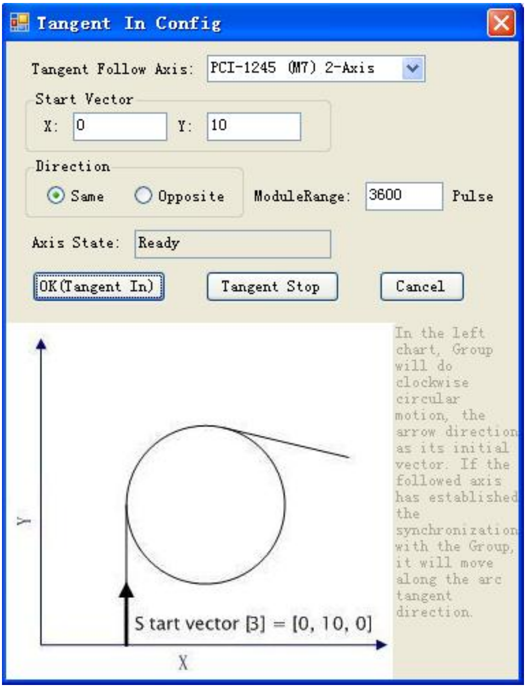
5.4.4.4.6 Blending Time

For detail, refer to the description about CFG_GpBldTime in Group which is listed in property list of Programming guide.

5.4.4.5 Tangent Follow Movement

5.4.4.5.1 Tangent In

Click [Tangent In], the follow dialog will be shown.



The following parameters need to be configured:

1. Tangent Follow Axis: Select tangent follow axis. As the axis can not be axes added in Group, so the axis listed in the combobox does not includes axes added in Group.
2. Start Vector: the start vector of tangent follow motion. In Utility, the default reference plane is X-Y, you only need to set X vector and Y vector. Z axis is not necessary to edit.
3. Direction: The direction of tangent follow axis in motion, which can be the same as or opposite to the direction of Group's motion.
4. ModuleRange: The module range of tangent follow axis, that is, the pulse number of tangent follow axis's one revolution (360 degrees)

There are related diagram and description below the configuration.There

Click [OK (Tangent In)], the tangent follow axis will establish tangent follow synchronization with the Group.

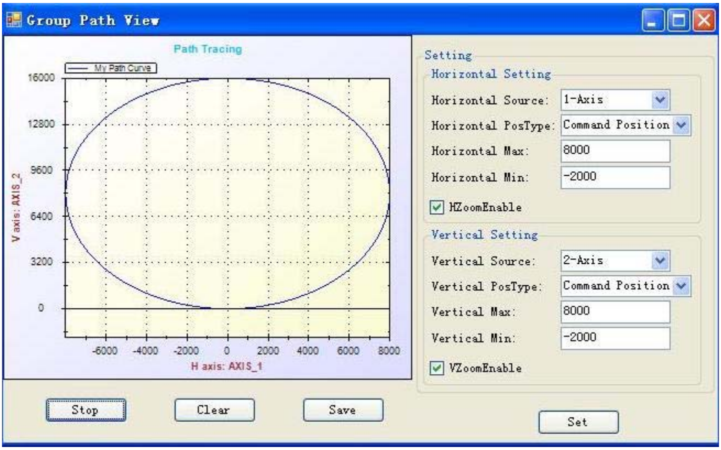
After that, if the Group do interpolation motion, the following axis will move along the tangent direction of the interpolation motion.If the tangent follow axis has established tangent follow synchronization with the Group, click [Tangent In] again, the value of parameters in the form will be the configured value, and you can click [Tangent Stop] to dissolve the synchronization relationship. Click [Cancel], nothing will do but close the form.

5.4.4.5.2 Tangent Stop

Click [Tangent Stop] to dissolve the synchronization relationship between tangent follow axis and the Group.

5.4.4.6 Path Plot

Display the group motion curve. Click [Path Plot], the following form will appear:



5.4.4.6.1 Setting

Set the horizontal and vertical coordinates.

1. Horizontal setting

   a. Horizontal Source: Horizontal data source, 1st axis of group (sorted by the order being added) by default. You can choose any axis in group.

   b. Horizontal PosType: Horizontal position type, you can choose command or feedback position.

   c. Horizontal Max: Horizontal maximum coordinate.

   d. Horizontal Min: Horizontal minimum coordinate.

2. Vertical setting: same way as horizontal setting.

5.4.4.6.2 Set

Click [Set] to activate the effectiveness.

5.4.4.6.3 Start

Click [Start], the graphic box will be ready to draw the curve. If Group is in motion, you can see the trajectory. After clicked, the text on [Start] button will change into "Stop"; click [Stop], drawing the curve will be stopped and the text will back to "Start".

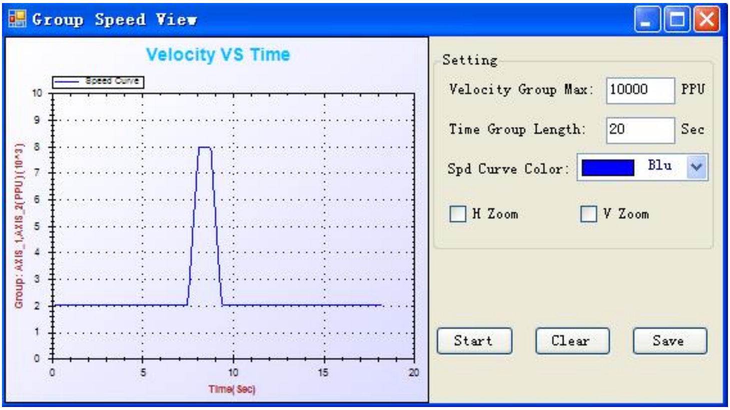
5.4.4.6.4 Clear

Click [Clear], the current curve in graphic box will be cleared.

5.4.4.6.5 Save

Click [Save], the specified path curve will be save as .png, .gif, .jpg, .tif or.bmp format.

5.4.4.7 Speed Chart



The setup and operation are similar to [Speed Chart] in "Single Axis Motion".

5.4.5 Path Status

To display the path status.

CurIndex: The serial number of path currently running.

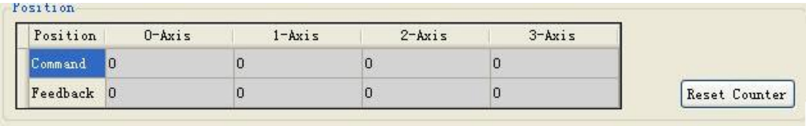
CurCmd: The command code of path currently running.

Remain: Unexecuted path number

FreeCnt: The remain space of Path Buffer

Path Count: The total path number in loaded Path file.

5.4.6 Position



Display the current command and feedback position for all axes of device.

Click [Reset Counter] to reset to 0.

5.4.7 State & Status

Group State: Show the current Group's State. For detail, refer to the description about State in Acm_GpGetState function which is listed in Common API of Programming guide.

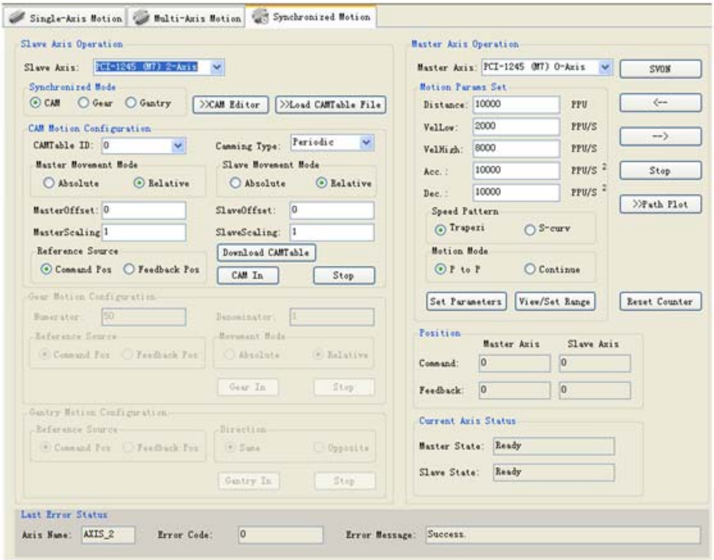
Last Error Status: display the latest error message:

Axis Name: The axis which has error.

Error Code: The error code.

Error Message: The specific error message.

5.5 Synchronized Motion



5.5.1 Slave Axis Operation

5.5.1.1 Slave Axis

Select any one of Device's axes to be slave axis.

Note: *Master axis and slave axis cannot be the same one. The default slave axis is 0-axis of the selected device.*

5.5.1.2 Synchronized Mode

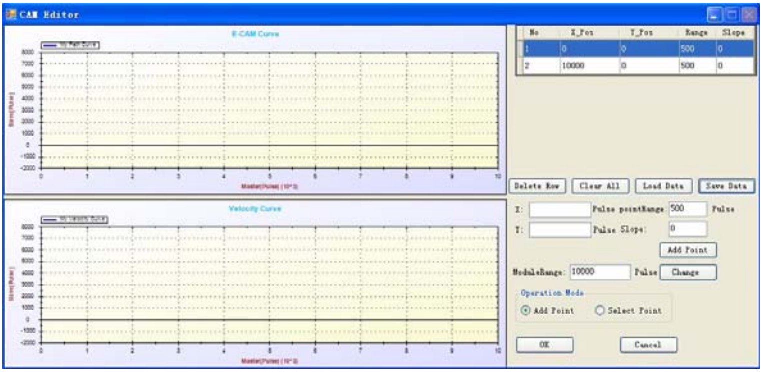
Select the synchronized mode: CAM, Gear and Gantry. You must select synchronized mode first before the configuration and establishment of synchronized motion.

5.5.1.3 CAM Motion

Select CAM as Synchronized Mode, then, you can execute the CAM setup and operation.

5.5.1.4 CAM Editor

Click [CAM Editor], the following dialog will show up:



Left upper corner is E-CAM curve graphic box; left lower corner is Velocity curve graphic box; right upper corner is CAM Table; right lower corner is operation panel.

1. Operation Mode

   a. Add Point: you can directly add CAM points on the E-CAM Curve. Whenever you add one point, CAM Curve will be redrawed. Wherein, CAM Point is espressed with a small red solid circle and CAM Curve is expressed with blue curve. In this operation mode, the shape of the mouse is cross. When the form is opened first time or the CAM curve has not been edited, the operation mode is "Add Point" mode by default.

   b. Select Point: you can select the corresponding CAM Point to drag and drop. At the same time, CAM Curve will be changed accordingly. In this operation mode, the shape of the mouse is arrow. When the form is opened again or the CAM curve has been edited, the default operation mode is "Select Point" mode.

2. CAM Table

   The CAM Table formed by edited CAM Point is shown on the top right. It is noteworthy that, the X_Pos and Y_Pos of the first row of CAM Table, that is the first CAM Point, must be zero, which means the Master axis's starting position is 0, the Slave axis's starting position is 0; The X_Pos and Y_Pos of CAM Table's last line, that is the last CAM Point, must be (ModuleRange, 0), which means the Master axis rotates a circle and the Slave axis backs to the starting position 0.

a. No: CAM Point 's serial number.

b. X_Pos: horizontal position coordinate (Master axis's position)

c. Y_Pos: vertical position coordinate (Slave axis's position)

d. Range: The distance between reference point and CAM Point. For detail, refer to the description about PointRange in Acm_DevDownloadCAMTable function which is listed in Common API of Programming guide.The default value is the edited value of pointRange. You can change the value by editing it. When you add more CAM Points, the pointRange will be also changed. If you want to change pointRange of edited CAM Point, directly modify it in CAM Table.

e. Slope: the slope between two reference points of CAM Point. For detail, refer to the description about PointSlope in Acm_DevDownloadCAMTable function which is listed in Common API of Programming guide. The default value is the value in Slope editing box, which can be modified by editing the value in the edit box, the slope of following added CAM Point will be the modified value in the edit box. If you want to change the Slope of edited CAM Point, directly edit in CAM Table.

Note: The range of Slope value is from -10 to 10. If the value is less than -10, it will be -10 by default. And if the value is larger than 10, it will be 10 by default.

3. CAM Table Operation

a. Delete Row: delete the selected row(s).

b. Clear All: clear all CAM Points (except starting point and final point)

c. Load Data: insert the selected CAM Table file. The file format can be binary (.bin) or .cvs readable by EXCEL.

d. Save Data: save the CAM Table. The file format can be binary (.bin) or .cvs readable by EXCEL.But if you want to import CAMTable through [Load CAMTable File] operation, you need to save the CAM Table as.bin format, because currently the device only support .bin file to import through [Load CAMTable File].

4. Add Point
   To add CAM Point, you can also edit X_Pos, Y_Pos, pointRange and Slope on the lower right and click [Add Point] to add it.
5. Change (ModuleRange)
   The master axis's revoluation pulse (ModuleRange) is set to be 10000 pulses by default. If you want to edit, you can edit in ModuleRange box, and then click [Change] to finish. After modified, the horizontal maximum ordinate of E-CAM Curve and Velocity Curve will be the modified value; if there are edited CAM Points before change the value, the X_Pos and pointRange of the CAM Points will become ModuleRange (after modified)/ Pre_ModuleRange (before modified) fold.
6. OK
   Click [OK] to save the CAM Table. You can use [Download CAMTable] to save the CAM Table into hardware.
7. Cancel
   Click [Cancel] to give up the editing.

5.5.1.4.1 Load CAMTable File

By clicking [Load CAMTable File] to choose binary file, the CAM Table will be saved in the hard drive.

Note: Before you save, you should set up the "CAMTableID" first. The value is 0 or 1. After you execute this step, the CAMTableID cannot be changed before you dissolve the syncrhonozation relation.

5.5.1.4.2 Download CAMTable

If CAM Table has edited in CAM Editor, you can use [Download CAM-Table] to save the CAM Table into hard drive.

Note: Before you save, you should set up the "CAM-TableID" first. The value is 0 or 1. After you execute this step, the CAMTableID cannot be changed before you dissolve the syncrhonozation relation.

5.5.1.4.3 Configuration

Configure cam motion and establish cam synchronization.

Before the establishment of cam synchronization, you need to configure the following parameters:

1. Camming Type:

   a. Non periodic: non-circular pattern. If you select this mode, after the Master axis runs a complete cycle, the Slave axis will no longer follow the Master axis according to CAM curve.

   b.Periodic: circular pattern. If you select this mode, the Slave axis will always follow the Master axis according to CAM curve in cam motion.

2. Master Movement Mode

   a. Absolute: If you select this mode, the current position of the Master axis will be served by CAM curve as the starting point of horizontal coordinate.

b. Relative: If you select this mode, the Master axis will serve the current Command/Actual Position as a starting point to move in relative mode.

3. Slave Movement Mode

   a. Absolute: If you choose this mode, the Slave axis will chase after the Y_Pos value in CAM Table with set speed from the current Command/Actual Position.

   b.Relative: If you choose this mode, the Slave axis will move in relative mode with the current Command/Actual Position according to CAM curve.

4. Master Offset: offset value relative to the Master axis.
5. Slave Offset: offset value relative to Slave axis.
6. Master Scaling: Master axis ratio factor. CAM Curve zoom in/out in the horizontal direction.
7. Slave Scaling: Slave axis ratio factor. CAM Curve zoom in/out in the vertical direction.
8. Reference Source: Master axis's position reference source.

   a. Command Position: reference source is command position.

   b. Feedback Position: reference source is feedback (actual) position.

5.5.1.4.4 CAM In

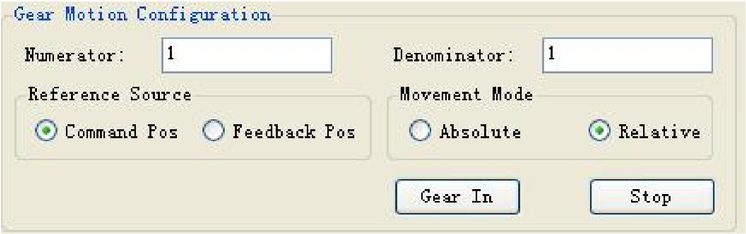
Click [CAM In], the Slave axis will establish CAM synchronization with the Master axis and the Slave's state will change into "Synchronous Driving". Thereafter, if the master axis is in P to P or Continue motion, the slave axis will follow the Master axis to Move with the CAM Curve and the configuration accordingly.

5.5.1.4.5 Stop

Click [Stop] to dissolve the synchronization relation. The slave axis's state will be ready.

5.5.1.5 Gear Motion

Select Gear in Synchronized Mode, you can configure and operate the gear movement.



5.5.1.5.1 Configuration

Before the establishment of gear synchronization, you need to configure the following parameters:

1. Numerator: The numerator of gear ratio
2. Denominator: The denominator of gear ratio
3. Reference Source: The Master axis's position reference source

   a.Command Position: reference source is command position

   b.Feedback Position: reference source is feedback position.
4. Movement Mode:

   a. Absolute: If you select this mode, the Slave axis will chase after the Command/Actual Position of the Master axis with set speed.

   b. Relative: If you select the mode, the Slave axis will keep initial position difference with the Master axis.

5.5.1.5.2 Gear In

Click [Gear In], the Slave axis will establish Gear synchronization with the Master axis and the Slave's state will change into "Synchronous Driving". Thereafter, if master axis is in P to P or Continue motion, the slave axis will follow master axis to move with the configuration accordingly.

5.5.1.5.3 Stop

Click [Stop] to dissolve the synchronization and the slave axis's state will be “Ready”.

5.5.1.6 Gantry Motion

Select Gantry in Synchronized Mode, you can configure and operate the gantry movement.

5.5.1.6.1 Configuration

Before the establishment of gantry synchronization, you need to configure the following parameters:

1. Reference Source: the Master axis's position reference source.

   a. Command Position: reference source is command position.

   b. Feedback Position: reference source is as feedback position.

2. Direction: The Slave axis direction relative to the Master axis

   a. Same: Same as the Master axis.

   b. Opposite: Opposite to the Master axis.

5.5.1.6.2 Gantry In

Click [Gantry In] to the Slave axis will establish gantry synchronization with the Master axis and the Slave's state will change into "Synchronous Driving". Thereafter, if the Master axis is in P to P or Continue motion, the Slave axis will follow the Master axis to move with the configuration accordingly.

5.5.1.6.3 Stop

Click [Stop] to dissolve the gantry synchronization and Slave axis's state will be "Ready".

5.5.2 Master Axis Operation

5.5.2.1 Master Axis

Select an axis as the Master axis from all the axes of the selected device.

Note: *The master and slave cannot be the same axis. The default master axis is 1st axis of selected device.*

5.5.2.2 Motion Params Set

It's the same setting as "Single-axis Motion" -> "Motion Params Set"

5.5.2.3 Operation

5.5.2.3.1 SVON

Click [SVON], the servos of the Master axis and the Slave axis will be turn on, the text on it will change into "SVOFF", click [SVOFF], the servos of axes will be turn off, the text on it will back to "SVON".

5.5.2.3.2 Other Operation

Refer to "Single-axis Motion" -> "Move Test" for other operation. It is noteworthy that after the establishment of synchronization, the Slave axis will follow the Master axis to move accordingly. You can view the movement curve through [Path Plot].

5.5.2.3.3 Position

Show the current command (theoretical) position and the feedback (actual) position of the Master axis and the Slave axis.

Click [Reset Counter] to set the value to 0.

5.5.2.4 State

You can view the current state of the Master axis and the Slave axis through the status bar.For detail, refer to the description about State in Acm_AxGetState function which is listed in Common API of Programming guide.

5.6 Digital Input

Mainly shows the status of device's digital input port.

In PCI-1265, there are 8 DIs.

As the above figure, Bit 7 to 0 from right to left are digital inputs respectively. Wherein, indicates that the DI is in effect (ON) and the value of the bit is 1; indicates that the DI is not in effect (OFF) and the value of the bit is 0. Hex indicates the hexadecimal value of the byte composed by 8 DIs.

5.7 Digital Output

Mainly shows the status of device's digital output port, and the corresponding ON/OFF operation on DO.

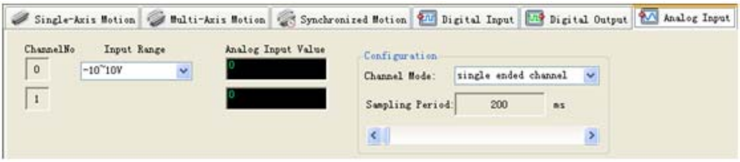
In PCI-1265, there are 8 DOs.

As the above figure, Bit 7 to 0 from right to left are digital outputs. Wherein, indicates that the DO is connected (ON) and the value of the bit is 1; indicates that the DO is not connected (OFF) and the value of the bit is 0. Hex indicates the hexadecimal value of the byte composed by 8 DOs.

5.8 Analog Input

Shows the status of device's analog input channels .

In the PCI-1265, there are two AI channels.



As the above, the parameters are as follows:

Channel No: AI index. PCI-1265 has two analog inputs and channel index is as 0 and 1.

Input Range: analog input range.

1. Analog Input Value: According to the sampling period, the analog input value sampled from the input channel.
2. Configuration
   a. Channel Mode: single ended channel and differential channel are available.
   b. Sampling period: use scrollbar to modify the value and its range is 200-10000ms.

2

CHAPTER 6

Programming Guide

This chapter is to detail the programming API for each function.

Chapter 6 Programming Guide

6.1 Introduction

This chapter supplies the APIs for user, shows the APIs definitions and how to use them.

PCI-1245/1245V/1245E/1265 device driver is based on the Common Motion Architecture. About the detail of Common Motion Architecture, see about Secton 4.3. According to this Architecture, all of functions and properties have been classified three types: **Device Object, Axis Object (Simple Axis)** and **Group Object (Multiple Axis)**. There are several basic concepts which should be known before using the API functions and properties.

- Naming of API and Properties: All of APIs and Properties under the Common Motion Architecture follows the uniform naming regulation. See about section 4.3.3.
- Data type redefinition: For simplifying code, the common data types are redefined.
- Error Code: All of APIs will return code to show success to call or failed for which error.

6.1.1 Data Type Redefinition

The table of redefinition of data types and windows common data types is as follows:

New Type	Windows Data Type	Comments
U8	UCHAR	8 bit unsigned integer
U16	USHORT	16 bit unsigned integer
U32	ULONG	32 bit unsigned integer
U64	ULONGLONG	64 bit unsigned integer
I8	CHAR	8 bit signed integer
I16	SHORT	16 bit signed integer
I32	INT	32 bit signed integer
I64	LONGLONG	64 bit signed integer
F32	FLAOT	32 bit Floating point variable
F64	DOUBLE	64 bit Floating point variable
PU8	UCHAR *	Pointer to 8 bit unsigned integer
PU16	USHORT *	Pointer to 16 bit unsigned integer
PU32	ULONG *	Pointer to 32 bit unsigned integer
PU64	ULONGLONG *	Pointer to 64 bit unsigned integer
PI8	CHAR *	Pointer to 8 bit signed integer
PI16	SHORT *	Pointer to 16 bit signed integer
PI32	INT*	Pointer to 32 bit signed integer
PI64	LONGLONG *	Pointer to 64 bit signed integer
PF32	FLAOT *	Pointer to 32 bit Floating point variable
PF64	DOUBLE *	Pointer to 64 bit Floating point variable

The initial character F/I/U represents the data type, and the digital represents the length of data.

6.1.2 About Error Code

Every API in Common Motion Architecture will get a returned code when it is called. The returned code represents a calling result. About the detail error code, see about Appendix. User can get error message according to the returned error code by Acm_GetErrorMessage. According to error message, user can make modification properly.

6.1.3 About Event

Event is the process of sending and handling message between objects. User can enable/disable event. If the event is enabled, the waiting event will get a notification when the event is triggered in driver if the condition which event needs has been met. And if it is disabled, user will not get the notification even though the event is triggered in driver.

There are seven types of event:

Event Name	Description
EVT_AX_MOTION_DONE	Trigger event when current motion is done.
EVT_AX_COMPARED	Trigger event when compare condition is meeted. (Not support in PCI-1245/ 1245E/1265)
EVT_AX_VH_START	Trigger event when motion velocity reaches High Speed.
EVT_AX_VH_END	Trigger event when motion slows down.
EVT_GPn_MOTION_DONE	Trigger event when group motion is done. n is group_id. (Get from PAR_GpGroupID by Acm_DevGetPropety).
EVT_GPn_VH_START	Trigger event when group motion velocity reaches High Speed. n is group_id.
EVT_GPn_VH_END	Trigger event when group motion slows down. n is group_id.

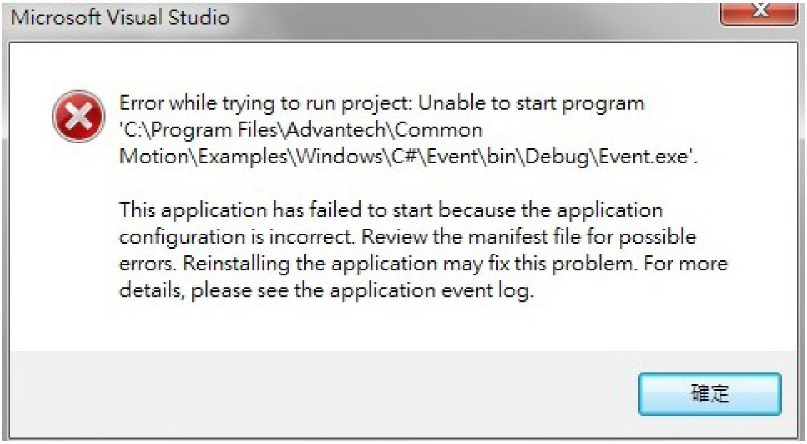
See about Acm_EnableMotionEvent, Acm_CheckMotionEvent.

6.1.4 About Using Common Motion API in Win7

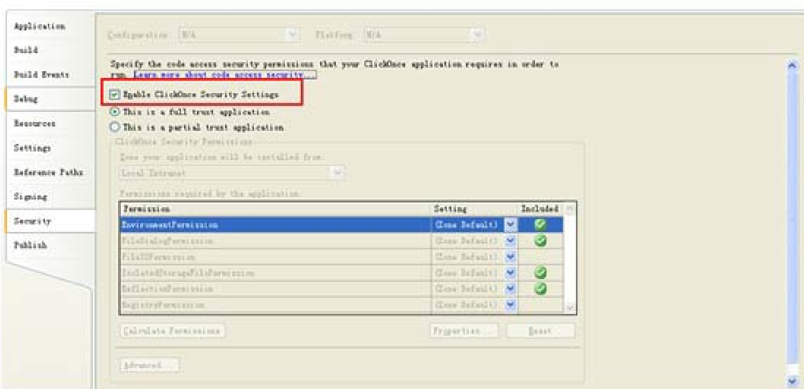
1. Acm_GetAvailableDevs has to read information from the registry in order to get the information of all boards that are installed in the computer. This operation requires Administrator rights. Therefore, if the application has to call this function, please add the corre-

sponding Manifest file and grant administrator rights to the application. (Please refer to "About Granting Administrator Rights to Applications”.)

2. IF you open C#/VB.net examples with VS2008 or VS2010 and the following error messages appear:



Uncheck the “Enable ClickOnce Security Setting” option in Security column of Project properties. Recompile and the application will run successfully.



6.1.5 About Elevating Application Privileges

1. To develop applications with Microsoft Visual Studio 2005(VS2005), you can copy the Manifest file "app.manifest" from the Properties folder of C#/VB.net examples to the Projects folder of the project. Click "Project"->"Add Existing Item" to add it to the project.

2. To develop applications with Microsoft Visual C++ 6.0, you can copy the Manifest file "App.manifest" from VC examples to the path of the project. Import this fle to the source. Source type: 24; Source ID: 1.

3. To develop applications with Microsoft Visual Studio 2008/2010, Method 1: Copy app.manifest from examples to the project (as in VS2005);
   Method 2: Directly change settings of project privilege management: Click "Project Properties"->"Configuration Properties"-->"Linker"-->"Manifest File"-->"UAC Execution Level"-->"requireAdministrator".
   Method 3: Check the "Enable ClickOnce Security Setting" option in "Security" column of "Project perperties", and the Manifest file will be automatically generated under "Properties". Open the Manifest file and change the content marked by the red box in the following image to “<requestedExecutionLevel level="requireAdministrator" uiAccess="false" />”. Uncheck the "Enable ClickOnce Security Setting" option in Security column of "Project properties".

```
 <requestedPrivileges xmlns="urn:schemas-microsoft-com:asm.v3">
   <!-- UAC Manifest Options
       If you want to change the Windows User Account Control level replace the
       requestedExecutionLevel node with one of the following.

   <requestedExecutionLevel  level="asInvoker" uiAccess="false" />
   <requestedExecutionLevel  level="requireAdministrator" uiAccess="false" />
   <requestedExecutionLevel  level="highestAvailable" uiAccess="false" />

       Specifying requestedExecutionLevel node will disable file and registry virtualization.
       If you want to utilize File and Registry Virtualization for backward
       compatibility then delete the requestedExecutionLevel node.
   -->
   <requestedExecutionLevel level="asInvoker" uiAccess="false" />
 </requestedPrivileges>
```

6.2 Getting Started

6.2.1 PCI-1245/1245V/1245E/1265 Software architecture

The PCI-1245/1245V/1245E/1265 software architecture based on Common Motion Architecture is as follows:

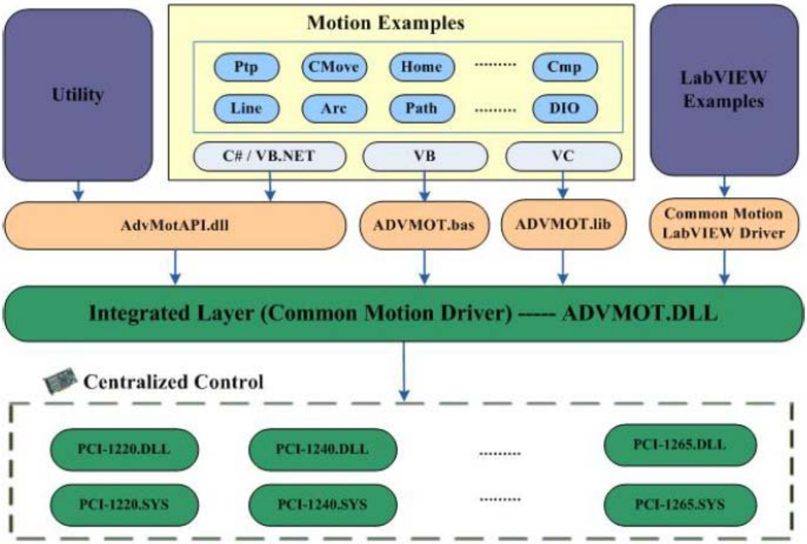


Figure 6.1: PCI-1265 Software Architecture

All of API used to implement device functions can be acquired from **ADVMOT.DLL** which is a common interface for user. The AdvMotAPI.dll, ADVMOT.bas and ADVMOT.lib are created upon ADVMOT.dll for user developing application easily. AdvMotAPI.dll is used for C# application and VB.net application which includes Utility, C# examples and VB.net example. ADVMOT.bas is used to develop VB application. ADVMOT.lib is used to develop VC application.

6.2.2 Flow Charts

6.2.2.1 Basic Flow

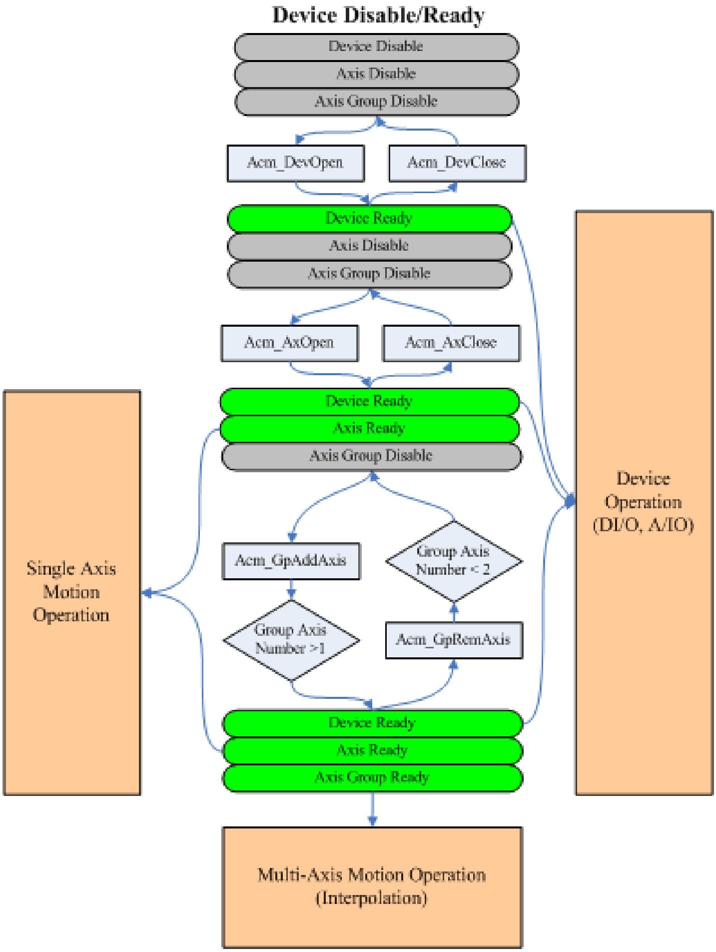


Figure 6.2: Basic Operation Flow Chart

6.2.2.2 Single Axis Flow

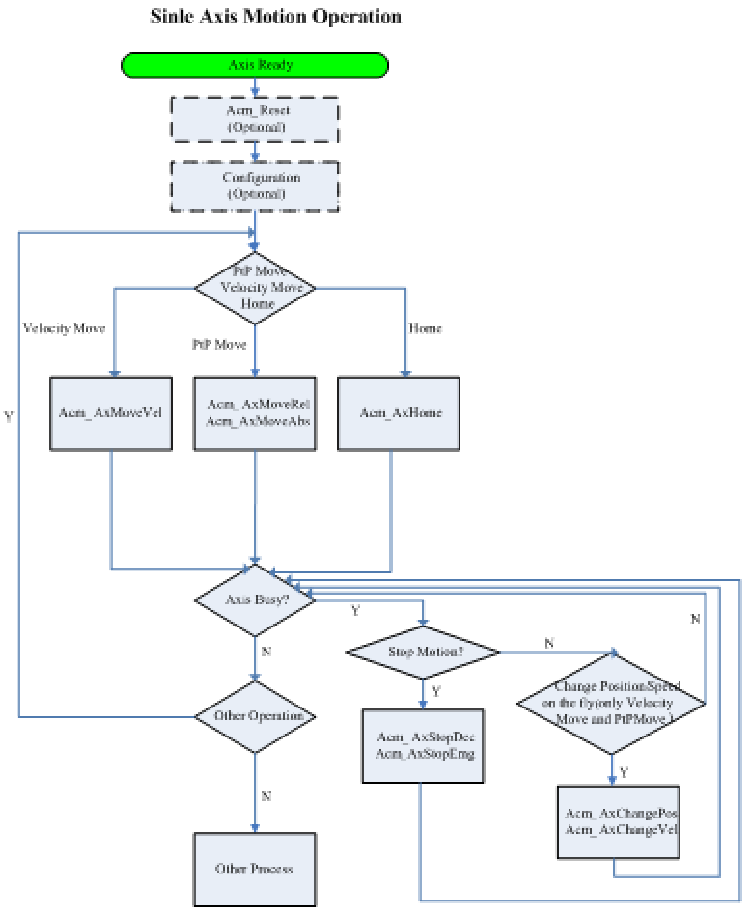


Figure 6.3: Single Axis Operation Flow Chart

6.2.2.3 Multiple Axis Flow Chart

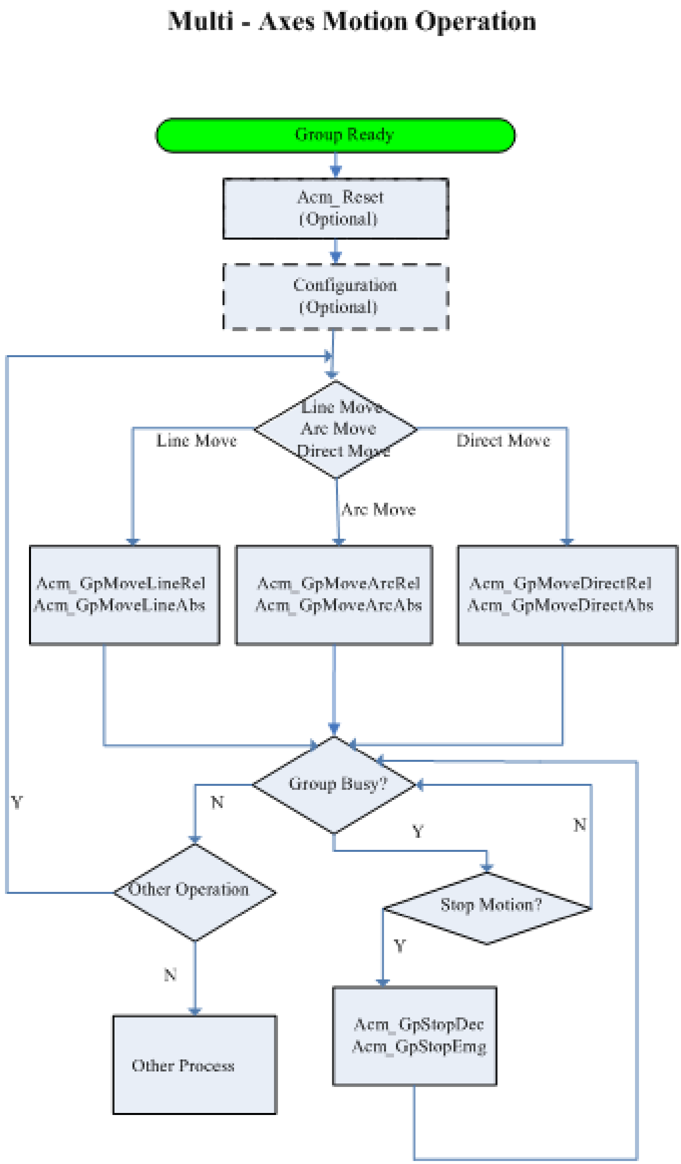


Figure 6.4: Multiple Axis Operation Flow Chart

6.2.2.4 E-cam Flow Chart

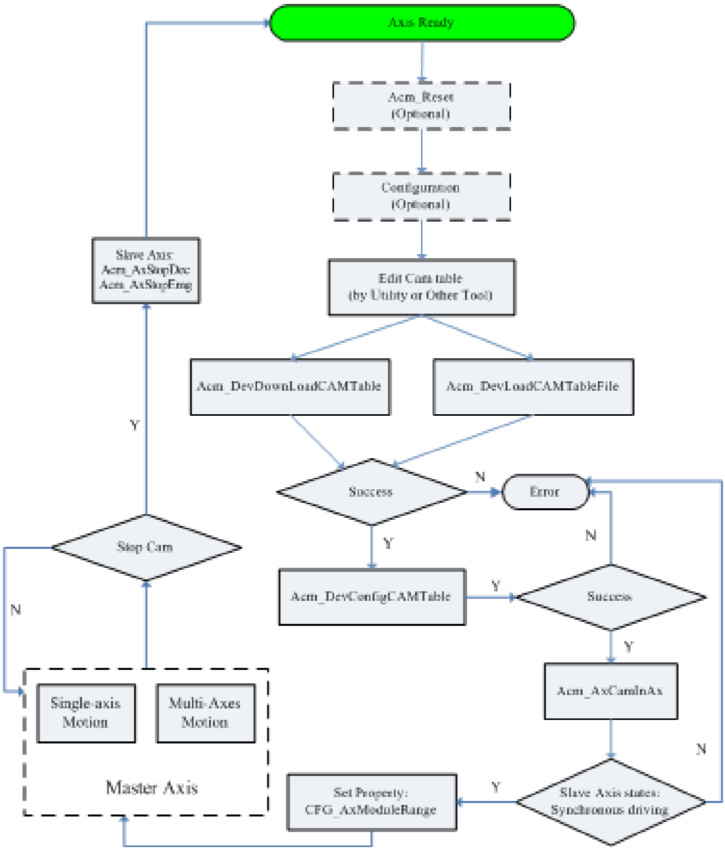


Figure 6.5: Cam Operation Flow Chart

6.2.2.5 E-Gear/Gantry Flow Chart

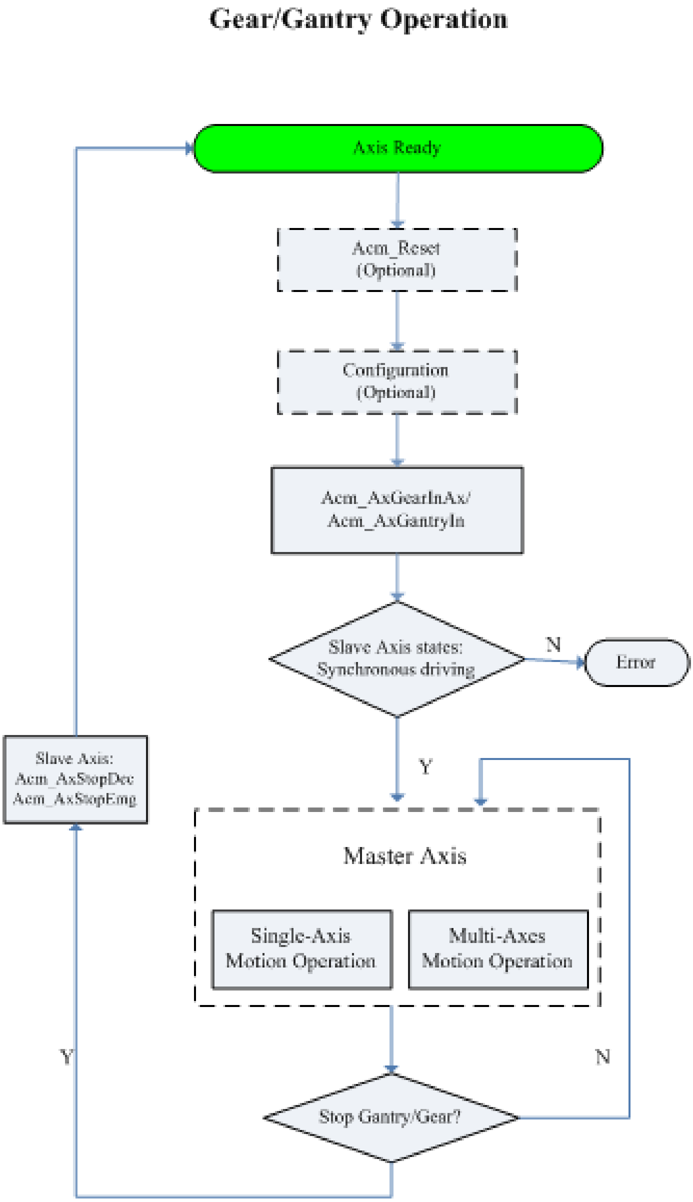


Figure 6.6: Gear/Gantry Operation Flow Chart

6.2.2.6 Tangential Following Flow Chart

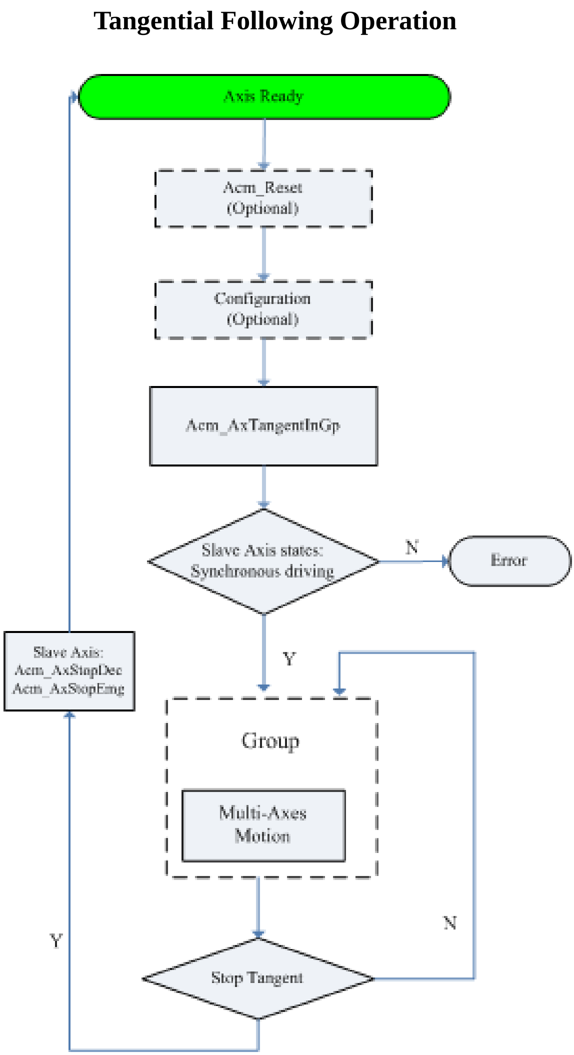


Figure 6.7: Tangential Following Operation Flow Chart

6.2.3 Example Support List

Example	VC	C#	VB	VB .NET	Description
ARC	√	√		√	Demonstrates how to control an interpolation group's arc motion.
Change_P	√	√		√	Demonstrates how to change the 1 axis motion position on the fly.
Change_V	√	√		√	Demonstrates how to change the 1 axis motion velocity on the fly.
Cmove	√	√		√	Demonstrates how to use the ACM API to control one axis continuous motion.
Compare	√	√		√	Demonstrates how to use the compare function.
DIO	√	√		√	Demonstrates axis digital input/output function.
Event	√	√		√	Demonstrates how to check event from driver.
Home	√	√		√	Demonstrates how to use the home function.
Line	√	√		√	Demonstrates how to control an interpolation group's line motion.
MPG_JOG	√	√		√	Demonstrates how to start external drive operation on the specified device and axis.
Path	√	√		√	Demonstrates how to control an interpolation group's path (continuous interpolation) motion.
PTP	√	√	√	√	Demonstrates how to control one axis point to point motion
SetCardRelation					Demonstrates how to control relations between multi PCI-1220 devices.
SimulateOpe	√	√			Demonstrates how to control simultaneous movement between multi-axis.
Direct	√	√			Demonstrates how to control an interpolation group's direct motion.
MoveImpose	√	√			Demonstrates how to use Move Impose function.
Latch	√	√			Demonstrates how to use latch function.
Helix	√	√			Demonstrates how to control an interpolation group's helix motion.
E-CAM	√	√			Demonstrates how to use electronic cam (E-CAM) function.
E-Gear	√	√			Demonstrates how to use electronic gear (E-Gear) function.
Tangent	√	√			Demonstrates how to use tangent follow function.
Gantry	√	√			Demonstrates how to use gantry function.
Device DIO	√	√			Demonstrates device digital input/output function.
Device AI	√	√			Demonstrates device analog input function.

6.2.4 PCI-1245/1245V/1245E/1265 Support API List

Type		Method/Event	PCI-1265	PCI-1245	PCI-1245V	PCI-1245E	Description
Device	Method	Acm_DevOpen	√	√	√	√	Open device.
		Acm_DevClose	√	√	√	√	Close device.
		Acm_DevLoadConfig	√	√	√	√	Load configuration file
		Acm_GetProperty	√	√	√	√	Get property.
		Acm_SetProperty	√	√	√	√	Set property.
		Acm_GetLastError	√	√	√	√	Get last error.
		Acm_CheckMotionEvent	√	√	√	√	Check if EVT_AX_MOTION_DONE happened.
		Acm_EnableMotionEvent	√	√	√	√	Enable/disable event.
		Acm_DevDownloadCAMTable	√	√	X	X	Load data in CamTable.
		Acm_DevConfigCAMTable	√	√	X	X	Configure Cam.
		Acm_DevLoadCAMTableFile	√	√	X	X	Load CamTable file.
		Acm_DevMDaqConfig	√	√	√	√	Set MDaq related configurations.
		Acm_DevMDaqGetConfig	√	√	√	√	Get MDaq related configurations.
		Acm_DevMDaqStart	√	√	√	√	Start MDaq function.
		Acm_DevMDaqStop	√	√	√	√	Stop MDaq function.
		Acm_DevMDaqReset	√	√	√	√	Get MDaq related data.
		Acm_DevMDaqGetStatus	√	√	√	√	Get current MDaq status.
		Acm_DevMDaqGetData	√	√	√	√	Get recorded MDaq data.

Device	Event	EVT_AX_MOTION_DONE	√	√	√	√	Event happens when axis motion is done.
		EVT_AX_COMPARED	X	X	X	X	Event happens when compare is matched.
		EVT_AX_LATCH	√	√	X	X	Event happens when Latch occurs.
		EVT_AX_ERROR	√	√	X	X	Event happens when an error occurs.
		EVT_AX_VH_START	√	√	√	√	Event happens when motion velocity reaches High Speed.
		EVT_AX_VH_END	√	√	√	√	Trigger happens when motion slows down.
		EVT_GPn_MOTION_DONE	√	√	√	√	Event happens when group motion is done.
		EVT_GPn_VH_START	√	√	√	√	Event happens when group motion velocity reaches High Speed.
		EVT_GPn_VH_END	√	√	√	√	Trigger happens when group motion slows down.
DAQ	Digital Input/output	Acm_DaqDiGetByte	√	X	X	X	DAQ function
		Acm_DaqDiGetBit	√	X	X	X	
		Acm_DaqDoSetByte	√	X	X	X	
		Acm_DaqDoSetBit	√	X	X	X	
		Acm_DaqDoGetByte	√	X	X	X	
		Acm_DaqDoGetBit	√	X	X	X	
	Analog Input/Output	Acm_DaqAiGetRawData	√	X	X	X	
		Acm_DaqAiGetVoltData	√	X	X	X	

Axis	SYSTEM	Acm_AxOpen	√	√	√	√	Open axis.
		Acm_AxClose	√	√	√	√	Close axis.
		Acm_AxResetError	√	√	√	√	Reset error when axis is error-stop.
	Motion I/O	Acm_AxSetSvOn	√	√	√	√	Open Servo Driver.
		Acm_AxGetMotionIO	√	√	√	√	Get status of motion-IO.
	Motion Status	Acm_AxGetMotionStatus	√	√	√	√	Get status of current motion.
		Acm_AxGetState	√	√	√	√	Get states of axis.
	Stop	Acm_AxStopDec	√	√	√	√	Decelerated stop.
		Acm_AxStopEmg	√	√	√	√	Emergency stop.
		Acm_AxStopDecEx	√	√	√	√	Command the axis to stop and specify the deceleration.
	Velocity Motion	Acm_AxMoveVel	√	√	√	√	Command continuous motion.
		Acm_AxChangeVel	√	√	√	√	Command velocity changing on current motion.
		Acm_AxChangeVelByRate	√	√	√	√	Change the velocity of current motion according to the given rate.
		Acm_AxChangeVelEx	√	√	√	√	Change the velocity, acceleration and deceleration simultaneously in motion status.
		Acm_AxChangeVelExByRate	√	√	√	√	Change the velocity, acceleration and deceleration simultaneously in motion status.
		Acm_AxGetCmdVelocity	√	√	√	√	Get current command velocity.
	Point-to-Point Motion	Acm_AxMoveRel	√	√	√	√	Command relative point-to-point motion.
		Acm_AxMoveAbs	√	√	√	√	Command absolute point-to-point motion.
		Acm_AxChangePos	√	√	√	√	Change end position on point-to-point motion.
		Acm_AxMoveImpose	√	√	X	X	Impose new motion on current motion.
	Simultaneous Motion	Acm_AxSimStartSuspendAbs	√	√	X	X	Suspend absolute simultaneous motion.
		Acm_AxSimStartSuspendRel	√	√	X	X	Suspend relative simultaneous motion.
		Acm_AxSimStartSuspendVel	√	√	X	X	Suspend continuous motion.
		Acm_AxSimStart	√	√	X	X	Start suspending simultaneous motion.
		Acm_AxSimStop	√	√	X	X	Stop suspending simultaneous motion.
	Home	Acm_AxHome	√	√	√	√	Command home.

Axis	Position/ Counter	Acm_AxSetCmdPosition	√	√	√	√	Set command position.
		Acm_AxGetCmdPosition	√	√	√	√	Get command position.
		Acm_AxSetActualPosition	√	√	√	√	Set actual position.
		Acm_AxGetActualPosition	√	√	√	√	Get actual position.
	Compare	Acm_ AxSetCmpData	√	√	X	X	Set comparison data.
		Acm_AxSetCmpTable	√	√	X	X	Set comparison data table.
		Acm_AxSetCmpAuto	√	√	X	X	Set line comparison datas.
		Acm_AxGetCmpData	√	√	X	X	Get current compare data.
	Latch	Acm_AxGetLatchData	√	√	X	X	Get latched data.
		Acm_AxTriggerLatch	√	√	X	X	Trigger latch data.
		Acm_AxResetLatch	√	√	X	X	Reset latch information.
		Acm_AxGetLatchFlag	√	√	X	X	Get latch flag.
	Aux/Gen Output	Acm_AxDoSetBit	√	√	√	√	Set bit value in DO.
		Acm_AxDoGetBit	√	√	√	√	Get bit value in DO.
		Acm_AxDiGetBit	√	√	√	√	Get bit value in DI.
	Ext-Drive	Acm_AxSetExtDrive	√	√	√	√	Set external driver.
	Application	Acm_AxCamInAx	√	√	X	X	Command e-cam.
		Acm_AxGearInAx	√	√	√	√	Command e-gear.
		Acm_AxGantryInAx	√	√	X	X	Command gantry.
		Acm_AxPhaseAx	√	√	√	√	Enable the phase lead or phase lag motion of the salve axis duringthe process of electronic cam or electronic gear.
		Acm_AxTangentInGp	√	√	X	X	Command tangent motion follow group.

Group	SYSTEM	Acm_GpAddAxis	√	√	√	√	Add axis into group.
		Acm_GpRemAxis	√	√	√	√	Remove axis from group.
		Acm_GpClose	√	√	√	√	Close group.
		Acm_GpResetError	√	√	√	√	Reset error when group is error-stopped.
	Motion Status	Acm_GpGetState	√	√	√	√	Get current states of group.
	Velocity	Acm_GpChangeVel	√	√	X	X	Command group to change the velocity while group is in line-interpolation motion.
		Acm_GpChangeVelByRate	√	√	X	X	Change the velocity of the current group motion according to the specified ratio.
		Acm_GpGetCmdVel	√	√	√	√	Get current velocity of the group.
	Motion Stop	Acm_GpStopDec	√	√	√	√	Decelerated stop.
		Acm_GpStopEmg	√	√	√	√	Emergency stop.

Group	Interpolation Motion	Acm_GpMoveLinearRel	√	√	√	√	Command relative linear interpolation.
		Acm_GpMoveLinearAbs	√	√	√	√	Command absolute linear interpolation.
		Acm_GpMoveCircularRel	√	√	X	√	Command relative arc interpolation.
		Acm_GpMoveCircularAbs	√	√	X	√	Command absolute arc interpolation.
		Acm_GpMoveCircularRel_3P	√	√	X	√	Command relative 3-point arc interpolation.
		Acm_GpMoveCircularAbs_3P	√	√	X	√	Command absolute 3-point arc inter-polation.
		Acm_GpMoveCircularRel_Angle	√	√	√	X	Complete circular inter-polation through relative center coordinates,and angle & direction of rotaion.
		Acm_GpMoveCircularAbs_Angle	√	√	√	X	Complete circular inter-polation through abso-lute center coordinates,and angle & direction of rotaion.
		Acm_GpMoveDirectAbs	√	√	√	√	Command absolute direct linear inter-polation.
		Acm_GpMoveDirectRel	√	√	√	√	Command relative direct linear inter-polation.
		Acm_GpMoveHelixRel	√	√	X	X	Command relative helix interpolation.
		Acm_GpMoveHelixAbs	√	√	X	X	Command absolute helix interpolation.
		Acm_GpMoveHelixRel_3P	√	√	X	X	Command relative 3-point helix interpo-lation.
		Acm_GpMoveHelixAbs_3P	√	√	X	X	Command absolute 3-point helix interpo-lation.
	Path	Acm_GpAddPath	√	√	√	√	Add one path into sys-tem buffer.
		Acm_GpResetPath	√	√	√	√	Reset path system buf-fer.
		Acm_GpLoadPath	√	√	√	√	Load a path file.
		Acm_GpUnloadPath	√	√	√	√	Unload path.
		Acm_GpMovePath	√	√	√	√	Move path in system buffer.
		Acm_GpGetPathStatus	√	√	√	√	Get current path sta-tus.
		Acm_GpMoveSelPath	√	√	√	√	Move assigned range paths.
		Acm_GpGetPathIndexStatus	√	√	√	√	Get status of assigned index path.

6.2.5 Property Support List

Type		Property	PCI-1265	PCI-1245	PCI-1245V	PCI-1245E
Device	Feature	FT_DevIpoTypeMap	√	√	√	√
		FT_DevAxesCount	√	√	√	√
		FT_DevFunctionMap	√	√	√	√
		FT_DevOverflowCntr	√	√	√	√
		FT_DevMDAQTypeMap	√	√	√	√
		FT_DevMDAQTrigMap	√	√	√	√
		FT_DevMDAQMaxChan	√	√	√	√
		FT_DevMDAQMaxBufCount	√	√	√	√
	Configure	CFG_DevBoardID	√	√	√	√
		CFG_DevBaseAddress	√	√	√	√
		CFG_DevInterrupt	√	√	√	√
		CFG_DevBusNumber	√	√	√	√
		CFG_DevSlotNumber	√	√	√	√
		CFG_DevDriverVersion	√	√	√	√
		CFG_DevDllVersion	√	√	√	√
		CFG_DevFirmVersion	√	√	√	√
		CFG_DevCPLDVersion	√	√	√	√
DAQ	Feature	FT_DaqDiMaxChan	√	√	√	√
		FT_DaqDoMaxChan	√	√	√	√
		FT_DaqAiRangeMap	√	√	√	√
		FT_DaqAiMaxSingleChan	√	√	√	√
		FT_DaqAiMaxDiffChan	√	√	√	√
		FT_DaqAiResolution	√	√	√	√
	Configure	CFG_DaqAiChanType	√	X	X	X
		CFG_DaqAiRanges	√	X	X	X
Axis	System	FT_AxFunctionMap	√	√	√	√
		CFG_AxPPU	√	√	√	√
		CFG_AxPhyID	√	√	√	√

Axis	Speed Pattern	FT_AxMaxVel	√	√	√	√
		FT_AxMaxAcc	√	√	√	√
		FT_AxMaxDec	√	√	√	√
		FT_AxMaxJerk	√	√	√	√
		CFG_AxMaxVel	√	√	√	√
		CFG_AxMaxAcc	√	√	√	√
		CFG_AxMaxDec	√	√	√	√
		CFG_AxMaxJerk	√	√	√	√
		PAR_AxVelLow	√	√	√	√
		PAR_AxVelHigh	√	√	√	√
		PAR_AxAcc	√	√	√	√
		PAR_AxDec	√	√	√	√
		PAR_AxJerk	√	√	√	√
	Pulse IN	FT_AxPulseInMap	√	√	√	√
		FT_AxPulseInModeMap	√	√	√	√
		CFG_AxPulseInMode	√	√	√	√
		CFG_AxPulseInLogic	√	√	√	√
		CFG_AxPulseInMaxFreq	√	√	√	√
	Pulse OUT	FT_AxPulseOutMap	√	√	√	√
		FT_AxPulseOutModeMap	√	√	√	√
		CFG_AxPulseOutMode	√	√	√	√
	Alarm	FT_AxAlmMap	√	√	√	√
		CFG_AxAlmLogic	√	√	√	√
		CFG_AxAlmEn	√	√	√	√
		CFG_AxAlmReact	√	√	√	√
	In Position	FT_AxInpMap	√	√	√	√
		CFG_AxInpEnable	√	√	√	√
		CFG_AxInpLogic	√	√	√	√
	ERC	FT_AxErcMap	√	√	√	√
		FT_AxErcEnableModeMap	√	√	√	√
		CFG_AxErcLogic	√	√	√	√
		CFG_AxErcEnableMode	√	√	√	√
	SD	FT_AxSdMap	√	√	√	√

Axis	Hardware Limit	FT_AxElMap	√	√	√	√
		CFG_AxElReact	√	√	√	√
		CFG_AxElLogic	√	√	√	√
		CFG_AxElEnable	√	√	√	√
	Software Limit	FT_AxSwMelMap	√	√	√	√
		FT_AxSwPelMap	√	√	√	√
		CFG_AxSwMelEnable	√	√	√	√
		CFG_AxSwPelEnable	√	√	√	√
		CFG_AxSwMelReact	√	√	√	√
		CFG_AxSwPelReact	√	√	√	√
		CFG_AxSwMelValue	√	√	√	√
		CFG_AxSwPelValue	√	√	√	√
	Home	FT_AxHomeMap	√	√	√	√
		CFG_AxOrgLogic	√	√	√	√
		CFG_AxEzLogic	√	√	√	√
		CFG_AxHomeResetEnable	√	√	√	√
		PAR_AxHomeCrossDistance	√	√	√	√
		PAR_AxHomeExSwitchMode	√	√	√	√
	BackLash	FT_AxBacklashMap	√	√	√	√
		CFG_AxBacklashEnable	√	√	√	√
		CFG_AxBacklashPulses	√	√	√	√
		CFG_AxBacklashVel	√	√	√	√
	Compare	FT_AxCompareMap	√	√	√	√
		CFG_AxCmpSrc	√	√	X	X
		CFG_AxCmpMethod	√	√	X	X
		CFG_AxCmpPulseMode	√	√	X	X
		CFG_AxCmpPulseLogic	√	√	X	X
		CFG_AxCmpPulseWidth	√	√	X	X
		CFG_AxCmpEnable	√	√	X	X
	Latch	FT_AxLatchMap	√	√	√	√
		CFG_AxLatchLogic	√	√	X	X
		CFG_AxLatchEnable	√	√	X	X

Axis	Aux/Gen DIO	FT_AxGenDOMap	√	√	√	√
		FT_AxGenDIMap	√	√	√	√
		CFG_AxGenDoEnable	√	√	√	√
	Ext-Drive	FT_AxExtDriveMap	√	√	√	√
		FT_AxExtMasterSrcMap	√	√	√	√
		CFG_AxExtMasterSrc	√	√	√	√
		CFG_AxExtSelEnable	√	√	√	√
		CFG_AxExtPulseNum	√	√	√	√
		CFG_AxExtPulseInMode	√	√	√	√
		CFG_AxExtPresetNum	√	√	√	√
	CAM DO	FT_AxCamDOMap	√	√	√	√
		CFG_AxCamDOEnable	√	√	X	X
		CFG_AxCamDOLoLimit	√	√	X	X
		CFG_AxCamDOHiLimit	√	√	X	X
		CFG_AxCamDOCmpSrc	√	√	X	X
		CFG_AxCamDOLogic	√	√	X	X
	Module	CFG_AxModuleRange	√	√	X	X
	Simultaneity	FT_AxSimStartSourceMap	√	√	√	√
		CFG_AxSimStartSource	√	√	√	X
	DI Stop	FT_AxIN1Map	√	√	√	√
		FT_AxIN4Map	√	√	√	√
		FT_AxIN5Map	√	√	√	√
		CFG_AxIN1StopEnable	√	√	√	√
		CFG_AxIN1StopLogic	√	√	√	√
		CFG_AxIN2StopEnable	√	√	√	√
		CFG_AxIN2StopReact	√	√	√	√
		CFG_AxIN2StopLogic	√	√	√	√
		CFG_AxIN4StopEnable	√	√	√	√
		CFG_AxIN4StopReact	√	√	√	√
		CFG_AxIN4StopLogic	√	√	√	√
		CFG_AxIN5StopEnable	√	√	√	√
		CFG_AxIN5StopReact	√	√	√	√
		CFG_AxIN6StopLogic	√	√	√	√

Group	System	PAR_GpGroupID	√	√	√	√
		CFG_GpAxesInGroup	√	√	√	√
	Application	CFG_GpSFEnable	√	√	X	X
		CFG_GpBldTime	√	√	X	X
		PAR_GpRefPlane	√	√	X	X
	Speed Pattern	PAR_GpVelLow	√	√	√	√
		PAR_GpVelHigh	√	√	√	√
		PAR_GpAcc	√	√	√	√
		PAR_GpDec	√	√	√	√
		PAR_GpJerk	√	√	√	√

6.2.6 Creating a New Application

For creating a new application under PCI-1245/1245V/1245E/1265, user ought to install Common Motion Examples, there are many examples developed in different language in folder Advantech\Motion Common\Examples, user can follow these examples to develop a new application.

After installing CommonMotion examples, user can find two folders Include and Public in folder \Advantech\Motion Common, the files in Public folder are supplied for user to create applications in different languages, the relationship between files and developing language is as figure 6.1.

6.2.6.1 Creating a New VC Console Application

For creating a new console application, the procedure is as follow:

1. Click **File/New** from the main menu to create your application project and source code as you would for any other Visual C++ program.

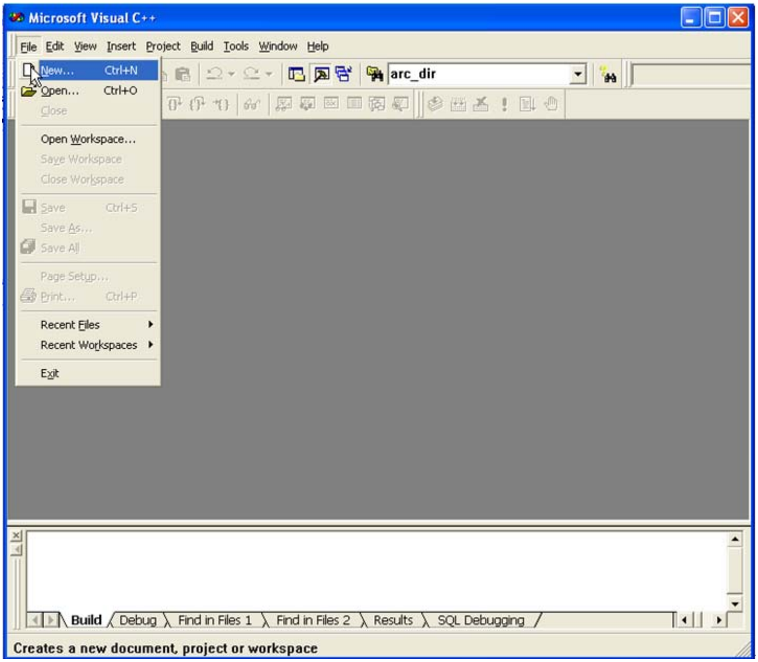


Figure 6.8: Open File to Creating a New VC Application

2. Define the type of new project as "**Win32 Console Application**", define the platform to be "**Win32**" and assign a project file directory.

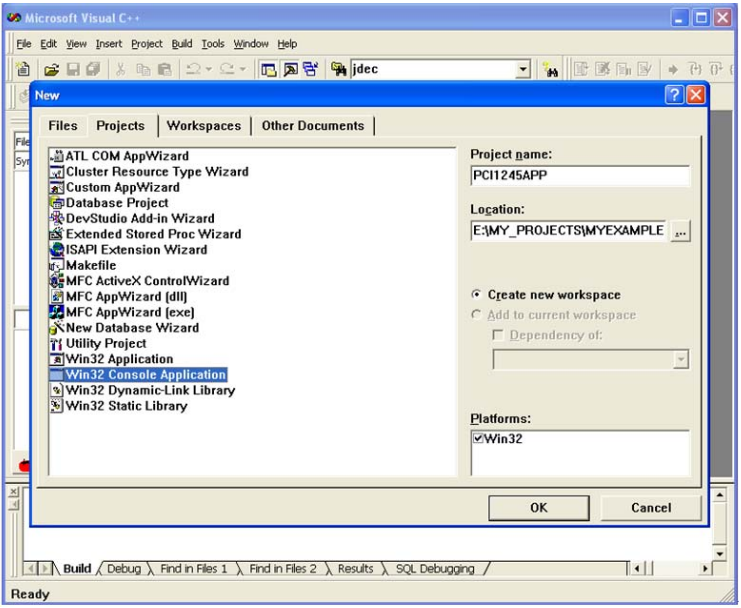


Figure 6.9: Creating a New VC Console Application

Click "OK", you can chose one kind of console application to create. Then a new console application has been created.

3. Config the new project. User should add the path of head files and necessary Lib file, and config the project in Project Setting. Use can open "Project Setting" in Menu - Poject - Settings ... or right click the new Project and chose "Setting" to open. The configuration is as follows.
a. In Common Motion Architecture, the Calling Convention should be "_stdcall", so user need to config the Calling convention as follow:

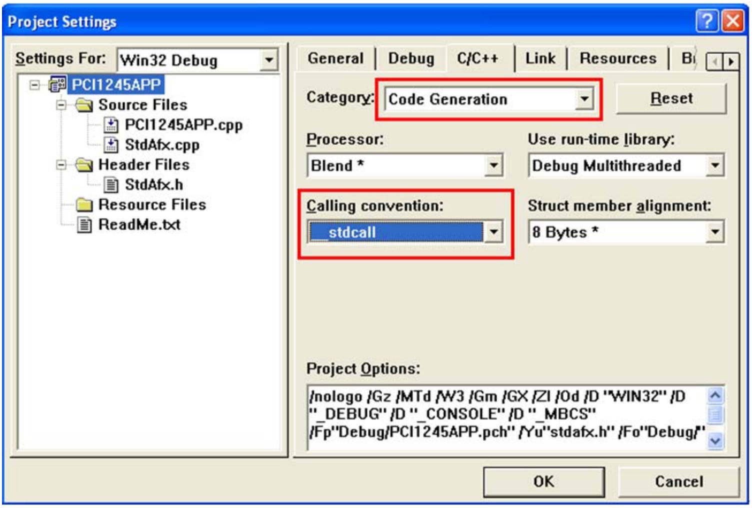


Figure 6.10: Setting Calling Convention

b. Set the head files path, the paths as follows contains all of head files which may be used by user. Plese pay attention the paths which must be corrective. For example, the content of folder which contains this project is as follow.

Figure 6.11: Folder Content of This Example

So the path setting is as follow.

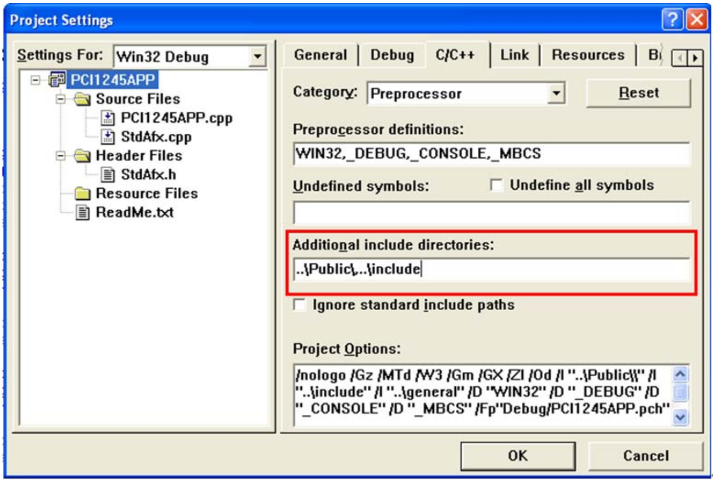


Figure 6.12: Add Head Files Path

c. Set the necessary Lib file.
The Lib file “ADVMOT.lib” which is corresponding to “ADV-MOT.dll” in folder systemroot\ system32\ is supplied for user to develop application easily. This Lib file is in “Public” folder after installing example package.
User should pay attention the path of the head files.

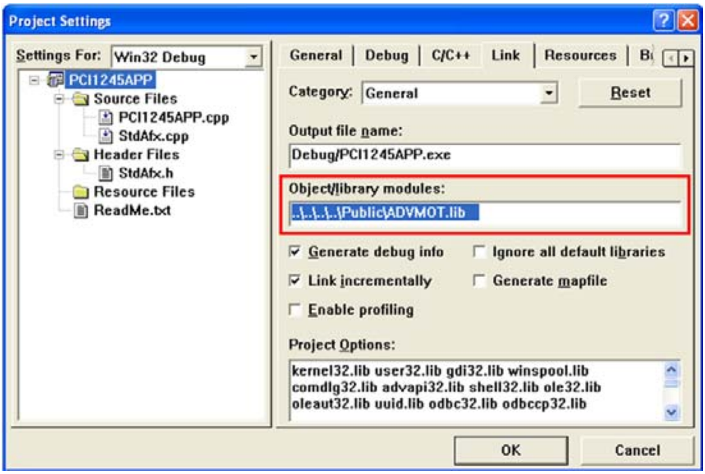


Figure 6.13: Setting Lib File Path

When finish the project setting, user can build this project if build successfully.

4. Write the code.

```
#include "stdafx.h"
#include <wtypes.h>
#include <stdio.h>
#include "AdvMotApi.h"

#define  MAX_CNT 100

int main(int argc, char* argv[])
{
ULONG errcde;
HAND devHandle;
HAND axHandle[MAX_CNT];
ULONG devNum , devCnt,buffLen, axisCntPerDev;
USHORT i;
DEVLIST devList[MAX_CNT];
//Step1. Get available devices by calling API
"Acm_GetAvailableDevs"
errcde = Acm_GetAvailableDevs(devList, MAX_CNT, &devCnt);
if (errcde!=0)
{
        printf("Can not find available device! \n");
        getchar();
        return 0;
}
    printf("Get available devices successfully! \n");
//Step2. Open device.
    devNum = devList[0].dwDeviceNum;
errcde = Acm_DevOpen(devNum, &devHandle);
```

```
if (errcde!=0)
{
      printf("Open device is failed! \n");
      getchar();
      return 0;
}
     printf("Open device successfully! \n");
//Step3. After open device successfully, user can get necessary
property.
     buffLen=sizeof(axisCntPerDev);
errcde = Acm_GetProperty (devHandle,FT_DevAxesCount, axis-
CntPerDev, &buffLen );
if (errcde!=SUCCESS)
{
         Acm_DevClose(&devHandle);
         printf("Get property is failed! \n");
         getchar();
         return 0;
 }
    printf("Get property successfully! \n");
//Step2. Open the axes.
    for (i=0; i<axisCntPerDev; i++)
    {
        errcde = Acm_AxOpen(devHandle, i, &axHandle[i]);
        if (errcde!=0)
        {
                 printf("Open axis_0 is failed! \n");
                 getchar();
            return 0;
         }
     }
printf("Open axes successfully! \n");
```

```
//Stp3. Move relative Axis 0 Point to Point motion.
errcde = Acm_AxMoveRel(axHandle[0], 10000);
if (errcde!=0)
{
        printf("move axis_0 is failed! \n");
        getchar();
        return 0;
}
    printf("Command axis 0 to move point to point successfully!
\n");
// Step 4. At last, Close axis and device before application exit.
     for (i=0; i<axisCntPerDev; i++)
     {
          errcde = Acm_AxClose(&axHandle[i]);
          if (errcde!=0)
          {
             printf("Open axis_0 is failed! \n");
             getchar();
             return 0;
    }
     }
 Acm_DevClose(&devHandle);
    getchar();
return 0;
}
```

5. The execution result.

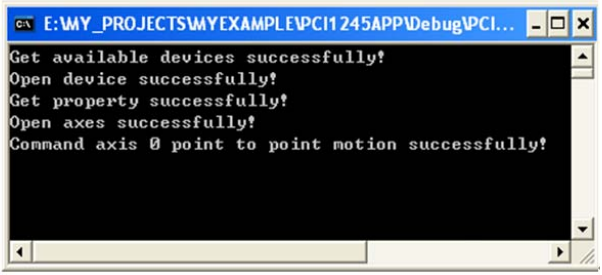


Figure 6.14: Result of VC Sonsole Example

6.2.6.2 Creating a New Visual Basic Application

For creating a new console application, the procedure is as follow:

1. Open the Visual Basic 6.0 development program, it will be loaded as follow:

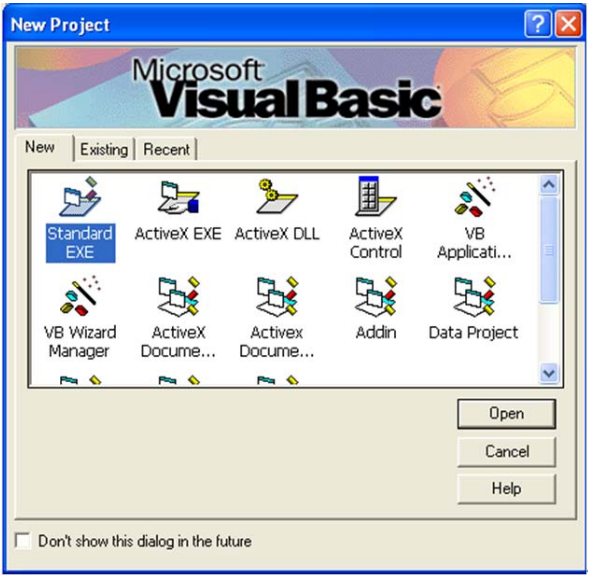


Figure 6.15: Load VB Development Environment

2. Select the **Standard EXE** icon and press the "**Open**" button. A new project is created.

3. Adding the module into project. Click on the **Project Explorer** in the **View** menu. Add ADVMOT.bas (In the Advantech\Motion Common\Public folder after installing examples package) module and general.bas (In the folder \Advantech\Motion Common\Examples after installing examples package) by clicking on **Add Module** in the **Project** menu.

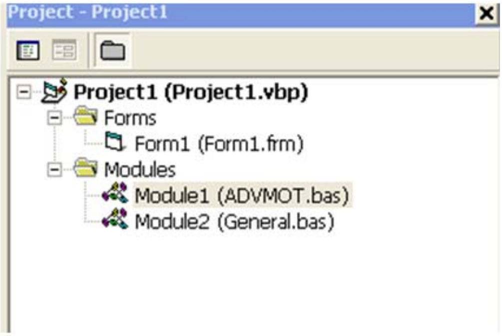


Figure 6.16: Add Module Files into Project

4. Design the form.

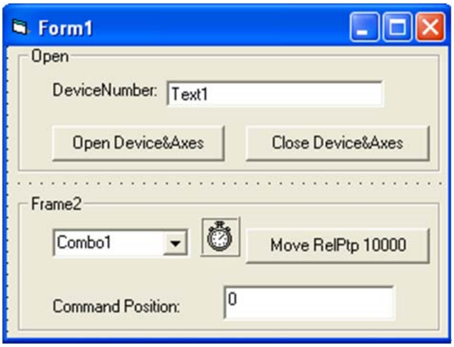


Figure 6.17: Design the Form

5. Write the code.
The variables definitions are as follow.

```
Option Explicit
Dim m_DevHand As Long
Dim m_dwDevNum As Long
Dim AxisPerDev As Long
Dim m_AxisHand() As Long
Dim m_CurAxis As Long
Dim m_avaDevs() As DEVLIST
```

When form is loaded, find the available devices by API "Acm_GetAvailableDevs". The code is as follow:

```
Private Sub Form_Load()
   Dim Result As Long
   Dim i, DeviceNumber As Long
   Dim strTemp As String
   ReDim m_avaDevs(16)
ReDim m_AxisHand(32)
//Get available devices by Acm_GetAvailableDevs
   Result = Acm_GetAvailableDevs(m_avaDevs(0),
MAX_DEVICES, DeviceNumber)
   If Result <> SUCCESS Then
     MsgBox "no available device in system", vbOKOnly, "error"
     Exit Sub
   End If
   If DeviceNumber <> 0 Then
      m_dwDevNum = m_avaDevs(0).dwDeviceNum
      tx_DevNum.Text = "0x" + Hex(m_dwDevNum)
      Timer1.Interval = 200
   Else
        MsgBox "no available device in system", vbOKOnly, "error"
   End IfEnd Sub
```

Click “Open Device&Axes”, the device and axes in the device will be opened. The timer is enabled. The combox will contain all of axes. The code is as follow:

```
Private Sub btn_OpenDev_Click()
    Dim Result As Long, i As Long, slaveDevs() As Long
    Dim strTemp As String
    Dim buffLen As Long
    Dim AxisNumber As Long
   //Open device.
    Result = Acm_DevOpen(m_dwDevNum, m_DevHand)
    If Result <> SUCCESS Then
        MsgBox "Open Device Failed", vbOKOnly, "PTP"
        Exit Sub
    End If

buffLen = 64
// Get Axis count by getting property.
    Result = Acm_GetProperty(m_DevHand, FT_DevAxesCount,
AxisPerDev, buffLen)
    If Result <> SUCCESS Then
        Acm_DevClose (m_DevHand)
        MsgBox "get axis number error", vbOKOnly, "PTP"
        Exit Sub
    End If
   // Open all of axes
    For AxisNumber = 0 To AxisPerDev - 1 Step 1
        Result = Acm_AxOpen(m_DevHand, AxisNumber,
m_AxisHand(AxisNumber))
        If Result <> SUCCESS Then
            MsgBox "Open Axis Failed", vbOKOnly, "PTP"
            Exit Sub
        End If
```

```
        Acm_AxSetCmdPosition m_AxisHand(AxisNumber), 0
        If Result <> SUCCESS Then
            MsgBox "Set command position failed", vbOKOnly, "PTP"
            Exit Sub
        End If
        strTemp = AxisNumber & "-Axis"
        cm_Axis.AddItem strTemp
    Next
    cm_Axis.ListIndex = 0
    m_CurAxis = 0
    Timer1.Enabled = True
End Sub
```

Click the combox to select axis, the code is as follow:

```
Private Sub cm_Axis_Click()
    m_CurAxis = cm_Axis.ListIndex
End Sub
```

The timer is used to get the command position of selected axis. The code is as follow:

```
Private Sub Timer1_Timer()
    Dim CurPos() As Double
    Dim strTemp As String
ReDim CurPos(32)
// Get command position of selected axis
    Acm_AxGetCmdPosition m_AxisHand(m_CurAxis), Cur-
Pos(m_CurAxis)
    strTemp = CurPos(m_CurAxis)
    tx_CmdPos.Text = strTemp
End Sub
```

Click “Close Device&Axes”, the device and axes in the device will be Closed.The timer is disabled. The code is as follow:

```
Private Sub btn_Close_Click()
    Dim AxisNum As Long
    For AxisNum = 0 To AxisPerDev - 1 Step 1
        Acm_AxClose m_AxisHand(AxisNum)
    Next
    Acm_DevClose m_DevHand
    cm_Axis.Clear
    Timer1.Enabled = False
End Sub
```

6. The result is as follow:

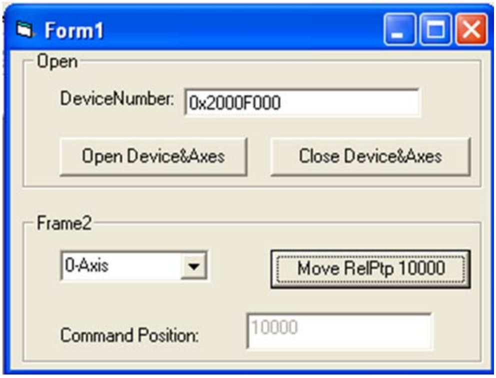


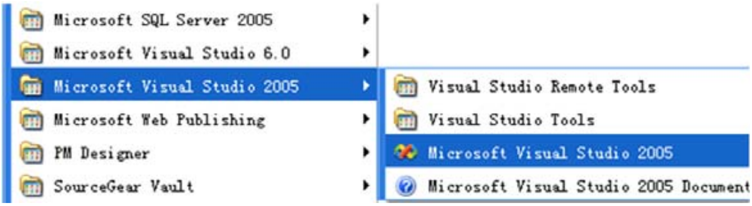
Figure 6.18: The Execution Result

6.2.6.3 Creating a New C# Application

To use PCI-1245/1245V/1245E/1265 Series DSP-Based SoftMotion PCI Controller, ADVMOT.dll and relevant driver files are needed. Be sure to install the driver before development.

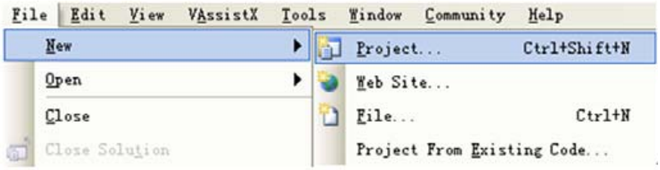
Create a C# project as follows:

1. **Create a new project**
   Select [Microsoft Visual Studio 2005] from the Microsoft Visual Studio 2005 in Start Menu, as follows:

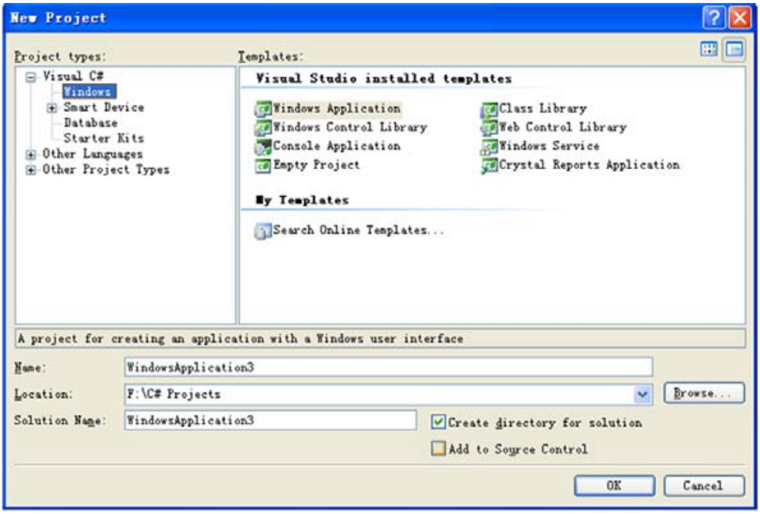


   The development environment of Microsoft Visual Studio 2005 is as follows:

To create a new project, Select [File] ---> [New] ---> [Project] of Main menu, as follows:



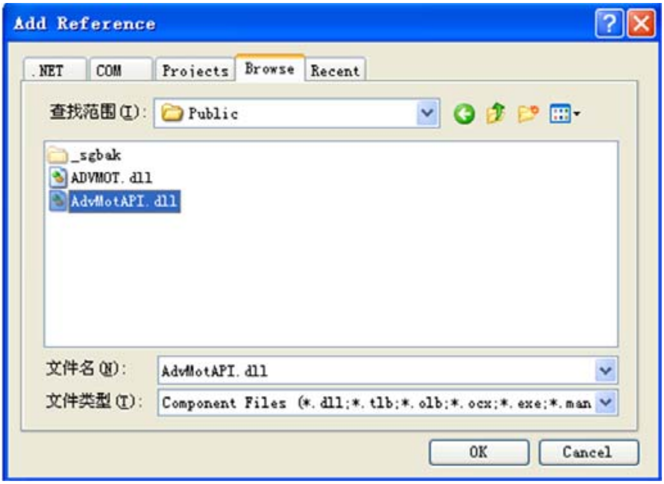
In the new form, the default language is “Visual C#”, select [Windows Application] template, Configure the Name, Location and Solution Name (Same as Name by default), and then click [OK].



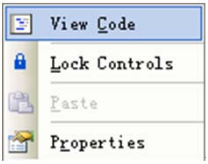
2. **Add relevant .dll**

a. Click [References] on the top right corner of development environment, as follows:

b. Click [Browse] of the [Add Reference] dialog box, Select “AdvMotAPI.dll” in the “Public” file folder from search path, then click [OK], as follows:



c. Right click on the Edit interface; select [View Code] to enter the program source code compilation interface, as follows:



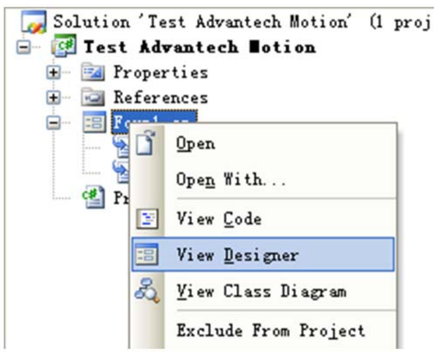
d. Add “using Advantech.Motion” under original referred namespaces, as follows:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using Advantech.Motion;
```

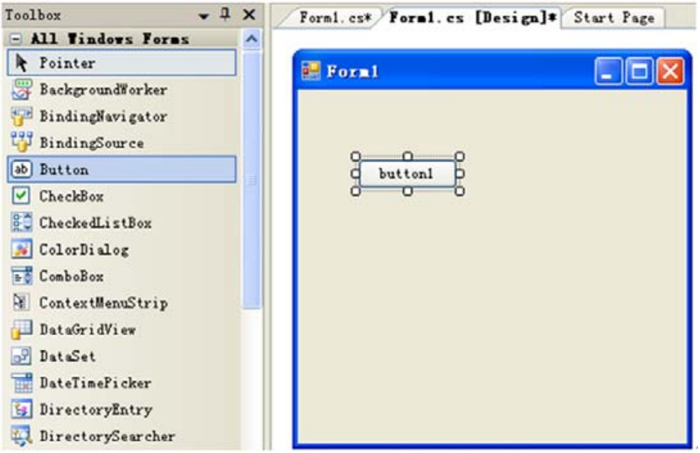
3. Coding

a. UI design

Double click [Form1.cs] or right click to select [View Designer] on [Form1.cs], then the UI edit interface will appear, as follows:



You can drag any Control/Component you need from the left Toolbox to edit user interface, as follows:



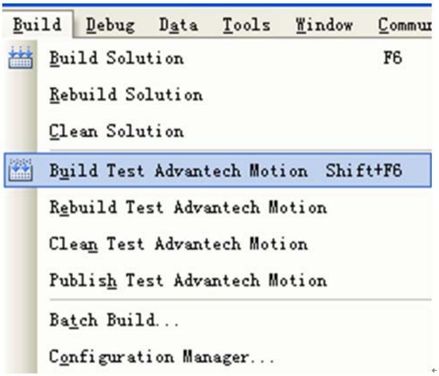
For detail, refer to Microsoft Visual C # user manual.

b. Coding

Right click on Form1.cs to select [View Code], then you enter the coding interface, you can code in relevant method/event of control/Component. For detail, refer to C# Examples of PCI-1245/1245V/1245E/1265 Series DSP-Based SoftMotion PCI Controller.

4. **Test program**

After the programming or if you want to compile the program, you can click [Build] ---> [Build Solution]\[Build Test Advantech Motion] in the menu bar, as follows:



You can directly click  in the toolbar, the program will run if there is no error.

If you want to debug the program, you can set breakpoint at corresponding line of code by clicking or pressing [F9], as follows:

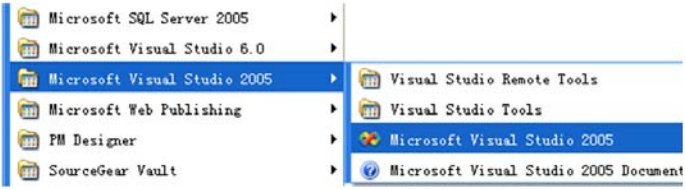
Click [Debug] ---> [Start Debugging] to debug, when run to the breakpoint, you can press [F11] or [F10] to step into/over, as follows:

6.2.6.4 Creating a New VB.net Application

To use PCI-1245/1245V/1245E/1265 Series DSP-Based SoftMotion PCI Controller, ADVMOT.dll and relevant driver files are needed. Be sure to install the driver before development.

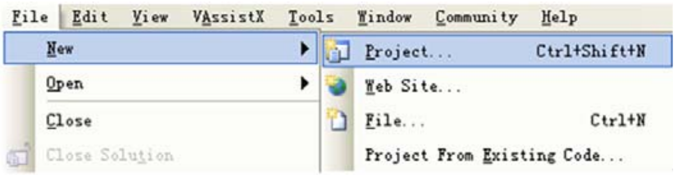
Create a Visual Basic project as follows:

1. **Create a new project**
   Select [Microsoft Visual Studio 2005] from the Microsoft Visual Studio 2005 in Start Menu, as follows:

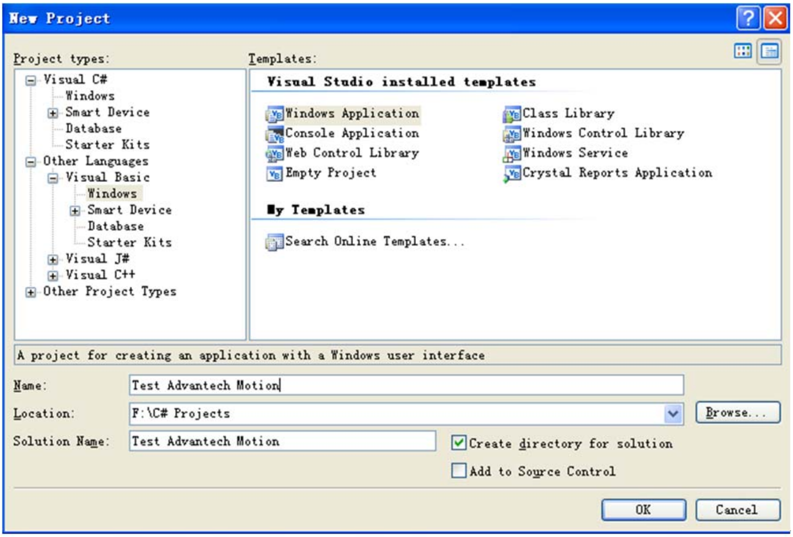


The development environment of Microsoft Visual Studio 2005 is as follows:

To create a new project, Select [File]--->[New]--->[Project] of Main menu, as follows:

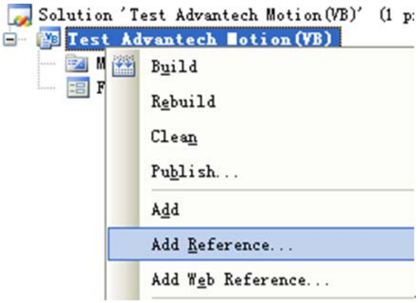


In the new form, Select [Other Languages]--->[Visual Basic], select [Windows Application] template, Configure the Name, Location and Solution Name(Same as Name by default), then click[OK].

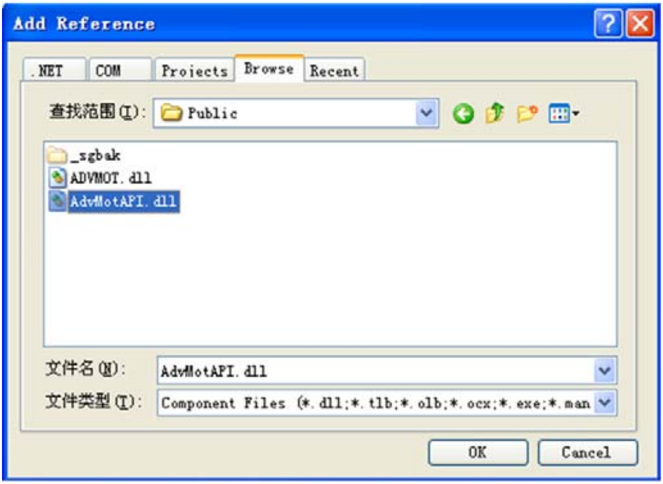


2. Add relevant .dll

a. Click [References] on the top right corner of development environment, as follows:

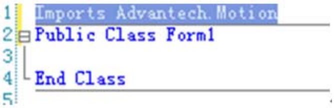


b. Click [Browse] of the [Add Reference] dialog box, Select "AdvMotAPI.dll" in the "Public" file folder from search path, then click [OK], as follows:



c. Right click on the Edit interface; select [View Code] to enter the program source code compilation interface, as follows:

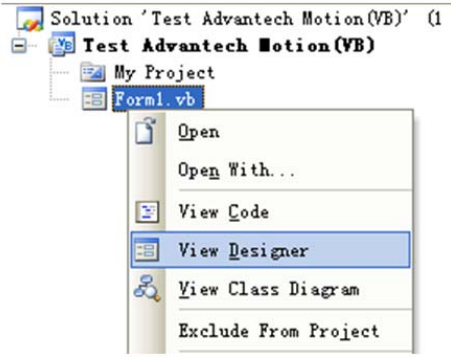
d. Add “Imports Advantech.Motion” under original referred namespaces, as follows:



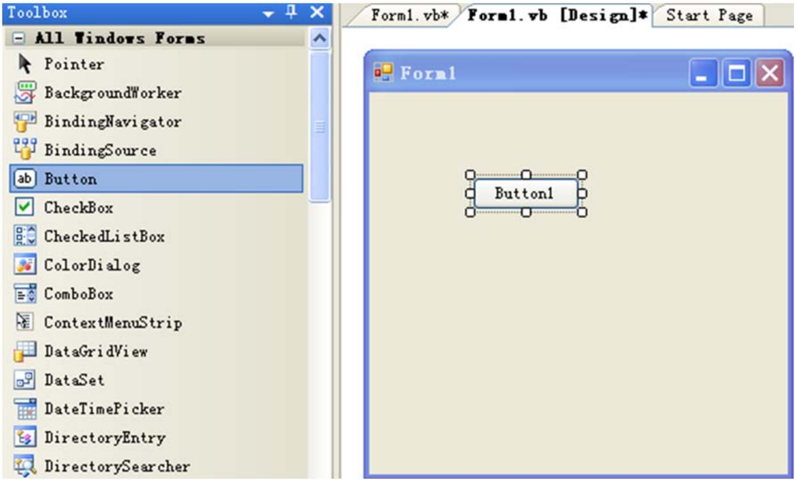
3. Coding

a. UI design

Double click [Form1.vb] or right click to select [View Designer] on [Form1.vb], then the UI edit interface will appear, as follows:



You can drag any Control/Component you need from the left Toolbox to edit user interface, as follows:



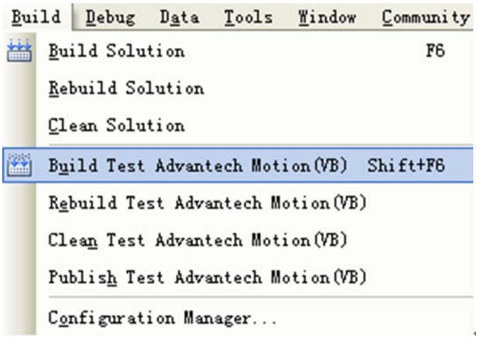
For details, refer to the Microsoft Visual Basic user manual.

b. Coding

Right click on Form1.vb to select [View Code], then you enter the coding interface, you can code in relevant method/event of control/Component. For detail, refer to VB.NET Examples of PCI-1245/1245V/1245E/1265 Series DSP-Based SoftMotion PCI Controller.

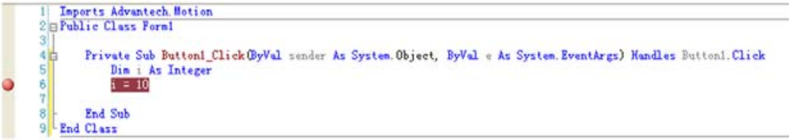
4. **Test program**

After the programming or if you want to compile the program, you can click [Build] ---> [Build Solution]\[Build Test Advantech Motion(VB)] in the menu bar, as follows:



You can directly click  in the toolbar, the program will run if there is no error.

If you want to debug the program, you can set breakpoint at corresponding line of code by clicking or pressing [F9], as follows:



Click [Debug] ---> [Start Debugging] to debug£¨when run to the breakpoint, you can press [F11] or [F10] to step into/over, as follows:

6.3 Function List

6.3.1 Common API

6.3.1.1 Acm_GetAvailableDevs

Format:

U32 Acm_GetAvailableDevs (DEVLIST *DeviceList, U32 MaxEntries, PU32 OutEntries)

Purpose:

Get the list of available device numbers and names of devices, of which driver has been loaded successfully.

Parameters:

Name	Type	In or Out	Description
DeviceList	DEVLIST*	OUT	Pointer to returned available device info list.
MaxEntries	U32	IN	The max devices count to get.
OutEntries	PU32	OUT	The count of available device.

Return Value:

Error Code.

Comments:

The structure of DEVLIST is:

```
typedef struct tagPT_DEVLIST
{
        DWORD       DeviceNum;
        CHAR        DeviceName[50];
        SHORT       NumOfSubDevices;
} DEVLIST, *LPDEVLIST;
```

DeviceNum:

Device Number needed for Acm_DevOpen.

DeviceName:

Device name. For example, PCI-1265/PCI-1245.

NumOfSubDevices:

Just for AMONET device. It is zero in PCI-1245 and PCI-1265.

6.3.1.2 Acm_GetErrorMessage

Format:

BOOL Acm_GetErrorMessage (U32 ErrorCode, LPTSTR lpszError, U32 nMaxError)

Purpose:

Get the error message according to error code returned from API.

Parameters:

Name	Type	In or Out	Description
ErrorCode	U32	IN	The returned error code of API.
lpszError	LPTSTR	OUT	The pointer to the string of error message.
nMaxError	U32	IN	The max length of string to receive error message.

Return Value:

Nonzero if the function is successful; otherwise 0 if no error message text is available.

Comments:

Acm_GetErrorMessage will not copy more than nMaxError -1 characters to the buffer and it will always add a trailing null to end the string. If the buffer is too small, the error message may be truncated.

6.3.2 Device Object

6.3.2.1 Acm_DevOpen

Format:

U32 Acm_DevOpen (U32 DeviceNumber, PHAND DeviceHandle)

Purpose:

Open a specified device to get device handle.

Parameters:

Name	Type	In or Out	Description
DeviceNumber	U32	IN	Device Number
DeviceHandle	PHAND	OUT	Return a point to the device handle

Return Value:

Error Code.

Comments:

This function should be called firstly before any operation of the device.

6.3.2.2 Acm_DevClose

Format:

U32 Acm_DevClose (PHAND DeviceHandle)

Purpose:

Close a device.

Parameters:

Name	Type	In or Out	Description
DeviceHandle	PHAND	IN	A pointer to the device handle

Return Value:

Error Code.

Comments:

Last of all, the device must be closed through this function.

6.3.2.3 Acm_DevLoadConfig

Format:

U32 Acm_DevLoadConfig (HAND DeviceHandle, PI8 ConfigPath)

Purpose:

Set all configurations for the device according to the loaded file.

Parameters:

Name	Type	In or Out	Description
DeviceHandle	HAND	IN	Device handle from Acm_DevOpen.
ConfigPath	PI8	IN	Pointer to a string that saves configuration file's path.

Return Value:

Error Code

Comments:

Configuration file can be binary or text file. If the file extension is.bin, driver reads the file in binary format. Otherwise, driver reads the file in .INI (text format).

User should debug device and set necessary configuration by Utility, then save these configuration information into file. This configuration file can be loaded in user's application by calling **Acm_DevLoadConfig.**

If user wants to save configuration information in .bin file format, the saved data structure (MOT_DEV_CONFIG) of configuration information should be as follow:

```
typedef struct _MOT_AX_CONFIG
{
        ULONG PlsPerUnit;
        DOUBLE MaxVel;
        DOUBLE MaxAcc;
        DOUBLE MaxDec;
        DOUBLE MaxJerk;
        DOUBLE VelHigh;
        DOUBLE VelLow;
        DOUBLE Dec;
        DOUBLE Acc;
        ULONG PlsInMde;
```

```
ULONG PlsInLgc;
ULONG PlsInMaxFreq;
ULONG PlsOutMde;
ULONG AlmEnable;
ULONG AlmLogic;
ULONG AlmReact;
ULONG InpEnable;
ULONG InpLogic;
ULONG ErcLogic;
ULONG ErcEnMde;
ULONG ElEnable;
ULONG ElLogic;
ULONG ElReact;
ULONG SwMelEnable;
ULONG SwPelEnable;
ULONG SwMelReact;
ULONG SwPelReact;
ULONG SwMelValue;
ULONG SwPelValue;
ULONG OrgLogic;
ULONG EzLogic;
ULONG HomeModeEx;
ULONG HomeExSwitchMode;
DOUBLE HomeCrossDis;
ULONG HomeResetEnable;
ULONG BacklashEnable;
ULONG BacklashPulses;
ULONG BacklashVel;
ULONG CmpSrc;
ULONG CmpMethod;
ULONG CmpPulseLogic;
ULONG CmpPulseWidth;
ULONG CmpEnable;
ULONG CmpPulseMode;
ULONG LatchLogic;
ULONG LatchEnable;
```

```
        ULONG GenDoEnable;
        ULONG ExtMasterSrc;
        ULONG ExtSelEnable;
        ULONG ExtPulseNum;
        ULONG ExtPulseInMode;
        ULONG ExtPresetNum;
        ULONG CamDoEnable;
        ULONG CamDOLoLimit;
        ULONG CamDOHiLimit;
        ULONG CamDoCmpSrc;
        ULONG CamDoLogic;
        ULONG ModuleRange;
        ULONG SimStartSource;
} MOT_AX_CONFIG, *PMOT_AX_CONFIG;

typedef struct _MOT_DAQ_CONFIG
{
        ULONG AiChanType;
        ULONG AiRanges;
} MOT_DAQ_CONFIG, *PMOT_DAQ_CONFIG;

typedef struct _MOT_DEV_CONFIG
{
        MOT_DAQ_CONFIG DaqConfig;
        MOT_Ax_CONFIG Axis_Cfg[Axis_Num];
} MOT_DEV_CONFIG, *PMOT_DEV_CONFIG;
```

Axis_Num is 4 for PCI-1245/1245V/1245E, is 6 for PCI-1265.

6.3.2.4 Acm_GetProperty

Format:

U32 Acm_GetProperty(HAND Handle, U32 ProperyID, PVOID Buffer, PU32 BufferLength)

Purpose:

Get the property (feature property, configuration property or parameter property) value through assigned PropertyID.

Parameter:

Name	Type	In or Out	Description
Handle	HAND	IN	Object handle. This handle may be device handle from Acm_DevOpen, or axis handle from Acm_AxOpen, or group handle from Acm_GpAddAxis
ProperyID	U32	IN	Property ID to query.
Buffer	PVOID	OUT	Data buffer for property.
BufferLength	PU32	IN/OUT	IN, buffer size for the property; OUT, returned data required length.

Return Value:

Error Code.

Comments:

User should pay attention on the data type and **BufferLength** of **Buffer** to get the value of property according to PropertyID. If the **Buffer** is too small, the return value will be error code "**InvalidInputParam**". In this case, driver will return the actual size of the property in **BufferLength**.

About the detail information of PerpertyID, see about Property List.

6.3.2.5 Acm_SetProperty

Format:

U32 Acm_SetProperty (HAND Handle, U32 ProperyID, PVOID Buffer, U32 BufferLength).

Purpose:

Set the property (configuration property or parameter property) value through assigned PropertyID.

Parameters:

Name	Type	In or Out	Description
Handle	HAND	IN	Object handle. This handle may be device handle from Acm_DevOpen, or axis handle from Acm_AxOpen, or group handle from Acm_GpAddAxis
ProperyID	U32	IN	Property ID to set.
Buffer	PVOID	OUT	Data buffer for property.
BufferLength	U32	IN	Buffer size for the property.

Return Value:

Error Code.

Comments:

For some properties, driver may package the value with some adjustment for precision consideration. So some properties' output value may be different from the input value. Eg. PAR_AxJerk.

Not all of properties in Property List can be set new property value; only the writable properties can be reset property value.

User should pay attention on data type and data length property needed. If the value of **BufferLength** is smaller than actual data size, error code "InvalidInputParamter" will be returned.

About the detail information of PropertyID, see about Property List.

6.3.2.6 Acm_GetLastError

Format:

U32 Acm_GetLastError (HAND ObjectHandle)

Purpose:

Get device or axis or group's last error code.

Parameters:

Name	Type	In or Out	Description
DeviceHandle	HAND	IN	Object handle. This handle may be device handle from Acm_DevOpen, or axis handle from Acm_AxOpen, or group handle from *Acm_GpAddAxis*

Return Value:

Error Code.

Comments:

To get detail information of error code by Acm_GetErrorMessage.

6.3.2.7 Acm_CheckMotionEvent

Format:

U32 Acm_CheckMotionEvent (HAND DeviceHandle, PU32 AxEvtStatusArray, PU32 GpEvtStatusArray, U32 AxArrayElements, U32 GpArrayElements, U32 Millisecond)

Purpose:

Check axis and groups enabled motion event status.

Parameters:

Name	Type	In or Out	Description
DeviceHandle	HAND	IN	Device handle from Acm_DevOpen.

<table>
<tr><td>AxEvtStatusArray</td><td>PU32</td><td>IN</td><td>Array[n]: Returned interrupt event status of each axis. n is the axis count of motion device.
Each array element is 32 bits data type, each bit represents different event types:
<table>
<tr><th>Bit</th><th>31...6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>Description</td><td>Reserved</td><td>EVT_AX_VH_END</td><td>EVT_AX_VH_START</td><td>EVT_AX_ERROR</td><td>EVT_AX_LATCHED (PCI-1245E not support)</td><td>EVT_AX_COMPARED (PCI-1245E not support)</td><td>EVT_AX_MOTION_DONE</td></tr>
</table>
Bit n = 1: Axis Motion Done event occurred;
Bit n = 0: Event not occurred or disabled.</td></tr>
<tr><td>GpEvtStatusArray</td><td>PU32</td><td>IN/OUT</td><td>Array[n]: Returned Interrupt event status for each group. n is just 1.
GpEvtStatus is 32 bits data type array and currently the values of n can only be 1.
<table>
<tr><th>Bit</th><th>Data</th><th>31...n</th><th>1</th><th>0</th></tr>
<tr><td rowspan="3">Description</td><td>GpEnableEvtArray[0]</td><td>EVT_GPn_MOTION_DONE</td><td>EVT_GP1_MOTION_DONE</td><td>EVT_GP0_MOTION_DONE</td></tr>
<tr><td>GpEnableEvtArray[1]</td><td>EVT_GPn_VH_START</td><td>EVT_GP1_VH_START</td><td>EVT_GP0_VH_START</td></tr>
<tr><td>GpEnableEvtArray[2]</td><td>EVT_GPn_VH_END</td><td>EVT_GP1_VH_END</td><td>EVT_GP0_VH_END</td></tr>
</table>
Bit n = 1: Group Motion Done event occurred; Bit n = 0: Event not occurred or disabled.
Note:
EVT_GPn_MOTION_DONE/ EVT_GPn_VH_START/ EVT_GPn_VH_END, n is GroupID. It can be get form PAR_GpGroupID property.</td></tr>
<tr><td>AxArrayElements</td><td>U32</td><td>IN</td><td>Number of AxEvtStatusArray elements.</td></tr>
<tr><td>GpArrayElements</td><td>U32</td><td>IN</td><td>Number of GpEvtStatusArray elements. It should be 1.</td></tr>
<tr><td>Millisecond</td><td>U32</td><td>IN</td><td>Specify the time out value in millisecond for each checking</td></tr>
</table>

Return Value:

Error Code.

Comments:

If you want to get event status of axis or groups, you should enable these events by calling Acm_EnableMotionEvent.

User should create a new thread to check event status.

6.3.2.8 Acm_EnableMotionEvent

Format:

U32 Acm_EnableMotionEvent (HAND DeviceHandle, PU32 AxEnableEvtArray, PU32 GpEnableEvtArray, U32 AxArrayElements, U32 GpArrayElements)

Purpose:

Enable motion event.

Parameters:

<table>
<tr><th>Name</th><th>Type</th><th>In or Out</th><th>Description</th></tr>
<tr><td>DeviceHandle</td><td>HAND</td><td>IN</td><td>Device handle from Acm_DevOpen.</td></tr>
<tr><td>AxEnableEvtArray</td><td>PU32</td><td>IN</td><td>Array[n], enable interrupt event for each axis, n is the axis count of motion device.
Array is of 32 bits data type, each bit represents different Event types:\
<table>
<tr><td>Bit</td><td>31...6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Description</td><td>Reserved</td><td>EVT_AX_VH_END</td><td>EVT_AX_VH_START</td><td>EVT_AX_ERROR</td><td>EVT_AX_LATCHED (PCI-1245E not support)</td><td>EVT_AX_COMPARED (PCI-1245E not support)</td><td>EVT_AX_MOTION_DONE</td></tr>
</table>
Bit n = 1: Enable event;
Bit n = 0: Disable event.</td></tr>
</table>

<table>
<tr><td>GpEnableEvtArray</td><td>PU32</td><td>IN</td><td>Array[n], enable interrupt event for each group. GpEnableEvtArray is 32 bits data type array and currently the value of n can only be 1.
<table>
<tr><th>Bit</th><th>Data</th><th>31...n</th><th>1</th><th>0</th></tr>
<tr><td rowspan="3">Description</td><td>GpEnableEvtArray[0]</td><td>EVT_GPn_MOTION_DONE</td><td>EVT_GP1_MOTION_DONE</td><td>EVT_GP0_MOTION_DONE</td></tr>
<tr><td>GpEnableEvtArray[1]</td><td>EVT_GPn_VH_START</td><td>EVT_GP1_VH_START</td><td>EVT_GP0_VH_START</td></tr>
<tr><td>GpEnableEvtArray[2]</td><td>EVT_GPn_VH_END</td><td>EVT_GP1_VH_END</td><td>EVT_GP0_VH_END</td></tr>
</table>
Bit n = 1: Enable event;
Bit n = 0: Disable event.
Note: For EVT_GPn_MOTION_DONE, n is GroupID. It can be got form PAR_GpGroupID property.</td></tr>
<tr><td>AxArrayElements</td><td>U32</td><td>IN</td><td>number of AxEvtStatusArray elements</td></tr>
<tr><td>GpArrayElements</td><td>U32</td><td>IN</td><td>number of GpEvtStatusArray elements</td></tr>
</table>

Return Value:

Error Code.

Comments:

After enable some events of axis or groups, the event status can be get from Acm_CheckMotionEvent.

6.3.2.9 Acm_DevDownloadCAMTable

Format:

U32 Acm_DevDownloadCAMTable (HAND DeviceHandle, U32 CamTableID, PF64 pMasterArray, PF64 pSlaveArray, PF64 pPointRangeArray, PF64 pPointSlopeArray, U32 ArrayElements)

Purpose:

This function downloads a CAM table profile which describes the ratio relationship of leading and following axis.

Parameters:

Name	Type	In/Out	Description
DeviceHandle	HAND	IN	Device handle. This handle is device handle from Acm_DevOpen.
CamTableID	U32	IN	Identifier of CAM table. PCI-1245 and PCI-1265 reserves 2 sets of cam table. So the ID can be 0, 1.
pMasterArray	PF64	IN	Master position array of CAM table points.
pSlaveArray	PF64	IN	Slave position array of CAM table points.
pPointRangeArray	PF64	IN	Point range of CAM table point.
pPointSlopeArray	PF64	IN	Point slope of CAM table point.
ArrayElements	U32	IN	Element count in the pMasterArray/ pSlaveArray/ pPointRangeArray/ pPointSlopeArray array. The Max. Element count is 128.

Return Value:

Error Code.

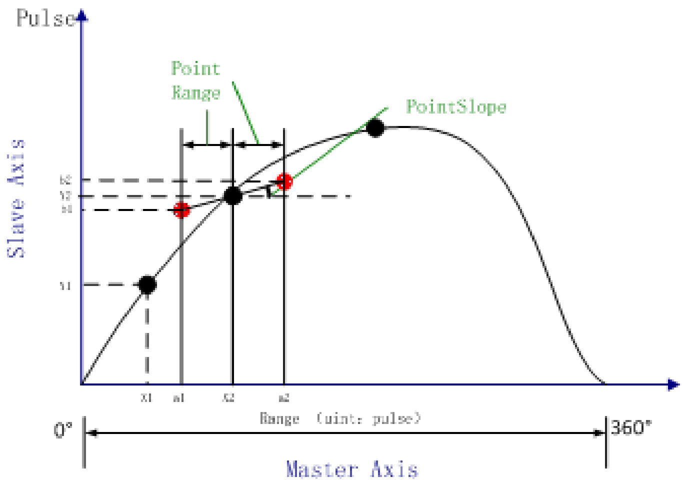
Comments:

Camming is characterized by dynamic ratio between the leading and following axis, and by the phase shift. The transmission ratio is described by a **CamTable**.

Camming is done with one table (two dimensional - describing master and slave positions together). The table should be strictly monotonic rising or falling, going both reverse and forward with the master.

PCI-1245E does not support this API.

The meaning of the parameters is as follow:



As top-figure, **Range** is the needed pulse number when master axis rotates 360°. The black points in figure are composed of master values (eg. X1, X2) in **MasterArray** and corresponding slave values in **SlaveArray**, and the red points created by black points and assigned **PointRange** and **PointSlope** are called reference points. Cam curve is fitted by points in **CamTable** composed of MasterArray and SlaveArray. The horizontal axis is the pulse number when master axis moves at some angles. The vertical axis is the pulse number of slave axis when master moves at some angels.

Range must be set into master axis by property CFG_AxModuleRange.

About cam operation, see about E-Cam Flow Chart in Chapter 6.2.2.

6.3.2.10 Acm_DevConfigCAMTable

Format:

U32 Acm_DevConfigCAMTable (HAND DeviceHandle, U32 CamTableID, U32 Periodic, U32 MasterAbsolute, U32 SlaveAbsolute)

Purpose:

This function sets the relevant parameters of the cam table.

Parameters:

Name	Type	In/Out	Description
DeviceHandle	HAND	IN	Device handle. This handle is device handle from Acm_DevOpen.
CamTableID	U32	IN	Identifier of Cam table. It is assigned by Acm_DevDownloadCAMTable. Device reserves 2 cam tables. So the ID can be 0, 1.
Periodic	U32	IN	CAM curve is executed periodic or non-periodic. 0: non-periodic, 1: periodic
MasterAbsolute	U32	IN	Interpret cam curve relative (0) or absolute (1) to the master axis. 0: relative, 1:absolute
SlaveAbsolute	U32	IN	Interpret cam curve relative (0) or absolute (1) to the slave axis. 0: relative, 1:absolute

Return Value:

Error Code.

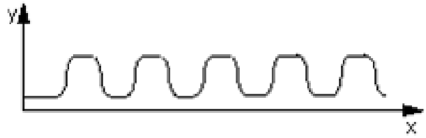
Comments:

Camming is done with one table (two dimensional - describing master and slave positions together).

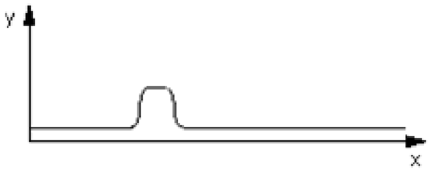
There are two types of Camming:

Periodic

Periodic: Once a curve is executed the camming immediately starts again at the beginning of the curve. As follow:



Non-periodic: After a curve is executed the execution stops.



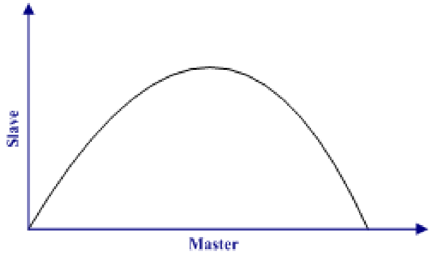
- MasterAbsolute and SlaveAbsolute

  Absolute: When absolute camming is set, the control values or the slave values based on the CamTable are interpreted as being absolute.The system compensates any offset developing between the leading and following axis during synchronization. When synchronism is reached, a defined phase relationship is established between the control value and the slave value.

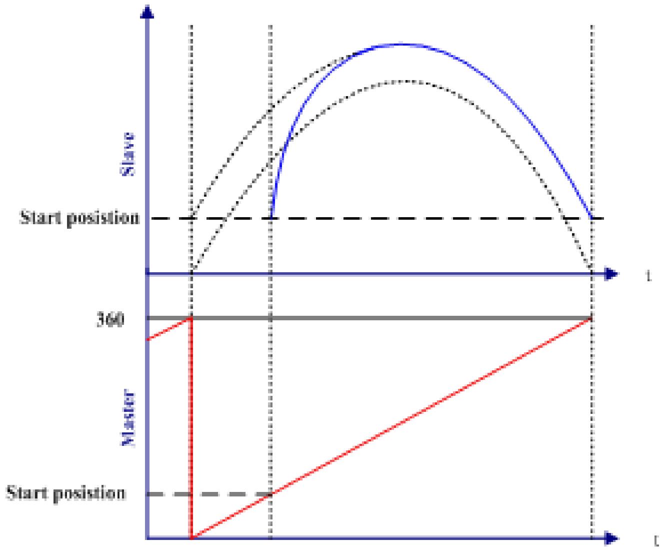
  Relative: When relative camming is set, this means that any offsets between the control value and the slave value are not compensated for during synchronization.

  For example, sectional drawing from Utility as follows:

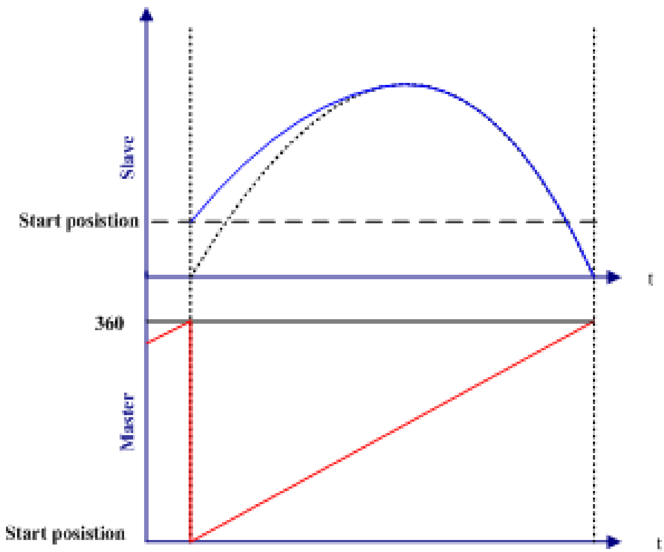
  The intial cam curve is as follow:



- Master axis: Absolute. Slave axis: Relative.



- Master axis: Relative. Slave axis: Absolute.



PCI-1245E/1245V does not support this API.

6.3.2.11 Acm_DevLoadCAMTableFile

Format:

U32 Acm_DevLoadCAMTableFile (HAND DeviceHandle, PI8 FilePath, U32 CamTableID, PU32 Range, PU32 PointsCount)

Purpose:

Load Cam Table file edited and saved by Utility into device.

Parameters:

Name	Type	In or Out	Description
DeviceHandle	HAND	IN	Device handle from Acm_DevOpen.
FilePath	PI8	IN	Pointer to a string that saves Cam-Table file's path.
CamTableID	U32	IN	CamTableID: 0 or 1.
Range	PU32	OUT	The pulse number which master needed in one period.
PointsCount	PU32	OUT	The points number in CamTable.

Return Value:

Error Code.

Comments:

The CamTable file is saved in .bin format in utility. When it is loaded successfully by this API, the API Acm_DevDownLoadCACMTable need not be called.

About E-cam operation, see about E-Cam Flow Chart in Chapter 6.2.2.

PCI-1245E/1245V does not support this API.

6.3.2.12 Acm_DevMDaqConfig

Format:

U32 Acm_DevMDaqConfig (HAND DeviceHandle,
U16 ChannelID, U32 Period, U32 AxisNo, U32 Method,
U32 ChanType, U32 Count)

Purpose:

Set MotionDAQ related configurations.

Parameters:

Name	Type	In or Out	Description
DeviceHandle	HAND	IN	Device handle form Acm_DevOpen.
ChannelID	U16	IN	Specify a Channel ID. The range is 0 ~ 3.
Period	U32	IN	Set an interval time between each data. The range is 0 ~ 255 and unit is ms.
AxisNo	U32	IN	Specify an axis. The range is : 0 ~ 3 (PCI1245/45E), 0 ~ 5 (PCI1265).

Method	U32	IN	Methods to trigger MDaq: 0: Disable; 1: Software trigger; 2: DI trigger; 3: Axis 0 starts to trigger; 4: Axis 1 starts to trigger; 5: Axis 2 starts to trigger; 6: Axis 3 starts to trigger; 7: Axis 4 starts to trigger;(PCI1245(E) does not support) 8: Axis 5 starts to trigger;(PCI1245(E) does not support) 9: Axis 6 starts to trigger;(PCI1265/ PCI1245(E) does not support) 10: Axis 7 starts to trigger;(PCI1265/ PCI1245(E) does not support) 11: Axis 8 starts to trigger;(PCI1265/ PCI1245(E) does not support) 12: Axis 9 starts to trigger;(PCI1265/ PCI1245(E) does not support) 13: Axis 10 starts to trigger;(PCI1265/ PCI1245(E) does not support) 14: Axis 11 starts to trigger;(PCI1265/ PCI1245(E) does not support)
ChanType	U32	IN	Get Channel Type: 0: Command Position; 1: Actual Position; 2: Lag Position (Difference value between Command Position and Actual Position); 3: Command Velocity (PCI1265/ PCI1245(E) does not support)
Count	U32	IN	Specify a data count. The range is 0 ~ 2000.

Return Value:

Error Code.

Comments:

6.3.2.13 Acm_DevMDaqGetConfig

Format:

U32 Acm_DevMDaqGetConfig (HAND DeviceHandle,
U16 ChannelID, PU32 Period, PU32 AxisNo, PU32 Method,
PU32 ChanType, PU32 Count)

Purpose:

Get MotionDAQ related values of a specified ChannelID.

Parameters:

Name	Type	In or Out	Description
DeviceHandle	HAND	IN	Device Handle.
ChannelID	U16	IN	Specify to a MDaq Channel to get the configurations. The range is 0 ~ 3.
Period	PU32	OUT	Get interval time between each data. The range is 0 ~ 255 and the unit is ms.
AxisNo	PU32	OUT	Get related informationof an axis. The range is 0 ~ 3 (PCI1245/45E), 0 ~ 5 (PCI1265).
Method	PU32	OUT	Methods to trigger MDaq: 0: Disable; 1: Software trigger; 2: DI trigger; 3: Axis 0 starts to trigger; 4: Axis 1 starts to trigger; 5: Axis 2 starts to trigger; 6: Axis 3 starts to trigger; 7: Axis 4 starts to trigger; 8: Axis 5 starts to trigger; 9: Axis 6 starts to trigger; 10: Axis 7 starts to trigger; 11: Axis 8 starts to trigger; 12: Axis 9 starts to trigger; 13: Axis 10 starts to trigger; 14: Axis 11 starts to trigger.

ChanType	PU32	OUT	Get Channel Type: 0: Command Position; 1: Actual Position; 2: Lag Position (Difference value between Command Position and Actual Position); 3: Command Velocity.
Count	PU32	OUT	Get the max count. The range is 0 ~ 2000.

Return Value:

Error Code.

Comments:

6.3.2.14 Acm_DevMDaqStart

Format:

U32 Acm_DevMDaqStart(HAND DeviceHandle)

Purpose:

Enable MotionDAQ function to record related data.

Parameters:

Name	Type	In or Out	Description
DeviceHandle	HAND	IN	Device Handle.

Comments:

When Method in Acm_DevMDaqConfig is set to MQ_TRIG_SW, command can be sent through this API to trigger MotionDAQ function.

6.3.2.15 Acm_DevMDaqStop

Format:

U32 Acm_DevMDaqStop (HAND DeviceHandle)

Purpose:

Stop recording MotionDAQ related data.

Parameters:

Name	Type	In or Out	Description
DeviceHandle	HAND	IN	Device Handle.

Comments:

6.3.2.16 Acm_DevMDaqReset

Format:

U32 Acm_DevMDaqReset (HAND DeviceHandle,
U16 ChannelID)

Purpose:

Clear MotionDAQ related data of corresponding ChannelID.

Parameters:

Name	Type	In or Out	Description
DeviceHandle	HAND	IN	Device Handle.
ChannelID	U16	IN	Channel ID. The range is 0 ~ 3.

Comments:

6.3.2.17 Acm_DevMDaqGetStatus

Format:

U32 Acm_DevMDaqGetStatus(HAND DeviceHandle,
U16 ChannelID, PU16 CurrentCnt, PU8 Status)

Purpose:

Get MotionDAQ status of corresponding ChannelID.

Parameters:

Name	Type	In or Out	Description
DeviceHandle	HAND	IN	Device Handle.
ChannelID	U16	IN	Channel ID. The range is 0 ~ 3.
CurrentCnt	PU16	OUT	Return current count that has been recorded.
Status	PU8	OUT	Return current status: 0: Ready; 1: Waiting Trigger; 2: Started.

Comments:

6.3.2.18 Acm_DevMDaqGetData

Format:

U32 Acm_DevMDaqGetData(HAND DeviceHandle,
U16 ChannelID, U16 StartIndex, U16 MaxCount,
PI32 DataBuffer)

Purpose:

Get all data in the range specified by MotionDAQ that has been recorded by corresponding channel.

Parameters:

Name	Type	In or Out	Description
DeviceHandle	HAND	IN	Device Handle.
ChannelID	U16	IN	Channel ID. The range is 0 ~ 3.
StartIndex	U16	IN	Set start posiiton to get data.
MaxCount	U16	IN	Set data count.
DataBuffer	PI32	OUT	Return data in the specified range.

Comments:

6.3.3 DAQ

6.3.3.1 Digital Input/ Output

6.3.3.1.1 Acm_DaqDiGetByte

Format:

U32 Acm_DaqDiGetByte (HAND DeviceHandle, U16 DiPort, PU8 ByteData)

Purpose:

Get the data of specified DI port in one byte.

Parameters:

Name	Type	In or Out	Description
DeviceHandle	HAND	IN	Device handle from Acm_DevOpen.
DiPort	U16	IN	It should be 0.
ByteData	PU8	OUT	Pointer of returned byte data.

Return Value:

Error Code.

Comments:

PCI-1245, PCI-1245V does not support DI. In PCI-1265, there are 8 DIs, so the DiPort should be 0.

6.3.3.1.2 Acm_DaqDiGetBit

Format:

U32 Acm_DaqDiGetBit (HAND DeviceHandle, U16 DiChannel, PU8 BitData)

Purpose:

Get the bit data of specified DI channel.

Parameters:

Name	Type	In or Out	Description
DeviceHandle	HAND	IN	Device handle from Acm_DevOpen.
DiChannel	U16	IN	DI channel.
BitData	PU8	OUT	Returned bit data. 0 or 1.

Return Value:

Error Code.

Comments:

PCI-1245, PCI-1245V does not support DI. In PCI-1265, there are 8 DIs, so the DiChannel should be 0 ~ 7.

6.3.3.1.3 Acm_DaqDoSetByte

Format:

U32 Acm_DaqDoSetByte (HAND DeviceHandle, U16 DoPort, U8 ByteData)

Purpose:

Set value to specified DO port.

Parameters:

Name	Type	In or Out	Description
DeviceHandle	HAND	IN	Device handle from Acm_DevOpen.
DoPort	U16	IN	DO port.
ByteData	U8	IN	Value to be set.

Return Value:

Error Code.

Comments:

PCI-1245, PCI-1245V does not support DO. In PCI-1265, there are 8 DOs, so the DoPort should be 0.

6.3.3.1.4 Acm_DaqDoSetBit

Format:

U32 Acm_DaqDoSetBit (HAND DeviceHandle, U16 DoChannel, U8 BitData)

Purpose:

Set the value to specified DO channel.

Parameters:

Name	Type	In or Out	Description
DeviceHandle	HAND	IN	Device handle from Acm_DevOpen.
DoChannel	U16	IN	DO channel.
BitData	U8	IN	Value to be set. 0 or 1.

Return Value:

Error Code.

Comments:

PCI-1245, PCI-1245V does not support DO. In PCI-1265, there are 8 DOs, so the DoChannel should be 0 ~ 7.

6.3.3.1.5 Acm_DaqDoGetByte

Format:

U32 Acm_DaqDoGetByte(HAND DeviceHandle, U16 DoPort,PU8 ByteData)

Purpose:

Get the byte data of specified DO port.

Parameters:

Name	Type	In or Out	Description
DeviceHandle	HAND	IN	Device handle from Acm_DevOpen.
DoPort	U16	IN	DO port.
ByteData	PU8	OUT	Returned value.

Return Value:

Error Code.

Comments:

PCI-1245, PCI-1245V does not support DO. In PCI-1265, there are 8 DOs, so the DoPort should be 0.

6.3.3.1.6 Acm_DaqDoGetBit

Format:

U32 Acm_DaqDoGetBit(HAND DeviceHandle, U16 DoChannel, PU8 BitData)

Purpose:

Get the bit value of specified DO channel.

Parameters:

Name	Type	In or Out	Description
DeviceHandle	HAND	IN	Device handle from Acm_DevOpen.
DoChannel	U16	IN	DO channel 0 ~ 7.
BitData	PU8	OUT	Returned value 0 or 1.

Return Value:

Error Code.

Comments:

PCI-1245, PCI-1245V does not support DO. In PCI-1265, there are 8 DOs, so the DoChannel should be 0 ~ 7.

6.3.3.2 Analog Input/Output

6.3.3.2.1 Acm_DaqAiGetRawData

Format:

U32 Acm_DaqAiGetRawData(HAND DeviceHandle,
U16 AiChannel, PU16 AiData)

Purpose:

Get the binary value of an analog input value.

Parameters:

Name	Type	In or Out	Description
DeviceHandle	HAND	IN	Device handle from Acm_DevOpen.
AiChannel	U16	IN	AI channel: 0 or 1.
AiData	PU16	OUT	Pointer to the returned binary AI value.

Return Value:

Error Code.

Comments:

PCI-1245, PCI-1245V do not support AI functions.
In PCI-1265, there are two AI channels, 0 and 1.
The channel can be set to Single-Ended or Differential through CFG_DaqAiChanType property. If it is set to Single-Ended, users can get AI value through any channel; if it is set to Differential, users can only get AI value through Channel 1.

6.3.3.2.2 Acm_DaqAiGetVoltData

Format:

U32 Acm_DaqAiGetVoltData(HAND DeviceHandle,
U16 AiChannel, PF32 AiData)

Purpose:

Get the actual analog input value.

Parameters:

Name	Type	In or Out	Description
DeviceHandle	HAND	IN	Device handle from Acm_DevOpen.
AiChannel	U16	IN	AI channel: 0 or 1.
AiData	PF32	OUT	Pointer to the returned analog value.

Return Value:

Error Code.

Comments:

PCI-1245, PCI-1245V do not support AI functions.
In PCI-1265, there are two AI channels, 0 and 1.
The channel can be set to Single-Ended or Differential through CFG_DaqAiChanType property. If it is set to Single-Ended, users can get AI value through any channel; if it is set to Differential, users can only get AI value through Channel 1.

6.3.4 Axis

6.3.4.1 System

6.3.4.1.1 Acm_AxOpen

Format:

U32 Acm_AxOpen (HAND DeviceHandle, U16 PhyAxis, PHAND AxisHandle)

Purpose:

Open specified axis and get this axis object's handle.

Parameters:

Name	Type	In or Out	Description
DeviceHandle	HAND	IN	Device handle from Acm_DevOpen.
PhyAxis	U16	IN	Physical Axis Number. (PCI-1265: 0, 1, 2,3,4,5. PCI-1245/1245E: 0,1,2,3)
AxisHandle	PHAND	OUT	Returns a pointer to the axis handle.

Return Value:

Error Code.

Comments:

Before any axis operation, this API should be called firstly. The physical axis number in PCI-1265: 0, 1, 2, 3, 4, 5. The physical axis number in PCI-1245, PCI-1245V, PCI-1245E: 0, 1, 2, 3.

6.3.4.1.2 Acm_AxClose

Format:

U32 Acm_AxClose (PHAND AxisHandle)

Purpose:

Close axis which has been opened.

Parameters:

Name	Type	In or Out	Description
AxisHandle	PHAND	IN	Pointer to the axis handle

Return Value:

Error Code.

Comments:

After calling this API, the axis handle cannot be used again.

6.3.4.1.3 Acm_AxResetError

Format:

U32 Acm_AxResetError (HAND AxisHandle)

Purpose:

Reset the axis' state. If the axis is in ErrorStop state, the state will be changed to Ready after calling this function.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.

Return Value:

Error Code.

Comments:

6.3.4.2 Motion IO

6.3.4.2.1 Acm_AxSetSvOn

Format:

U32 Acm_AxSetSvOn (HAND AxisHandle, U32 OnOff)

Purpose:

Set servo Driver ON or OFF.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
OnOff	U32	IN	Setting the action of SVON signal. 0: Off; 1: On

Return Value:

Error Code.

Comments:

6.3.4.2.2 Acm_AxGetMotionIO

Format:

U32 Acm_AxGetMotionIO (HAND AxisHandle, PU32 Status)

Purpose:

Get the motion I/O status of the axis.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
Status	PU32	OUT	Bit Description 0:RDY1---- RDY pin input; 1:ALM ---- Alarm Signal input; 2:LMT_P ---- Limit Switch+; 3:LMT_N---- Limit Switch-; 4:ORG---- Origin Switch; 5:DIR ---- DIR output; 6:EMG ---- Emergency signal input; 7:PCS ---- PCS signal input(not support in PCI-1245/1245E/1265); 8: ERC ---- Output deflection counter clear signal to a servomotor driver;(OUT7) 9: EZ ---- Encoder Z signal; 10: CLR ---- ext. input to Clear postion counter ;(not support in PCI-1245/1245V/1245E/1265 11: LTC ---- Latch signal input; 12: SD ---- Slow Down signal input;(not support in PCI-1245/1245V/1245E/1265) 13: INP ---- In-Position signal input; 14: SVON ---- Servo-ON (OUT6); 15: ALRM ----Alarm Reset output status; 16:SLMT_P ---- Software Limit+; 17: SLMT_N ---- Software Limit-; 18: TRIG-----Compare signal(OUT5); 19: CAMDO ---- position window do(OUT4);

Return Value:

Error Code.

Comments:

6.3.4.3 Motion Status

6.3.4.3.1 Acm_AxGetMotionStatus

Format:

U32 Acm_AxGetMotionStatus (HAND AxisHandle, PU32 Status)

Purpose:

Get current motions status of the axis.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
Status	PU32	OUT	Bit Description 0: Stop ---- Stop; 1: Res1 ---- Reserved; 2: WaitERC---- Wait ERC finished; 3: Res2 ---- Reserved; 4: CorrectBksh ---- Correcting Backlash; 5: Res3 ---- Reserved; 6: InFA ---- Feeding in return velocity = FA; 7: InFL ---- Feeding in StrVel speed =FL; 8: InACC ---- Accelerating; 9: InFH ---- Feeding in MaxVel speed = FH; 10: InDEC ---- Decelerating; 11:WaitINP----Wait in position.

Return Value:

Error Code.

Comments:

6.3.4.3.2 Acm_AxGetState

Format:

U32 Acm_AxGetState (HAND AxisHandle, PU16 State)

Purpose:

Get the axis's current state.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
State	PU16	IN	Axis states: Value Description 0: STA_AxDisable ---- Axis is disabled, you can open it to active it. 1: STA_AxReady ---- Axis is ready and waiting for new command. 2: STA_Stopping ---- Axis is stopping. 3: STA_AxErrorStop ---- Axis has stopped because of error. 4: STA_AxHoming ---- Axis is executing home motion. 5: STA_AxPtpMotion ---- Axis is executing PTP motion. 6: STA_AxContiMotion ---- Axis is executing continuous motion. 7: STA_AxSyncMotion ---- Axis is in one group and the group is executing interpolation motion, or axis is slave axis in E-cam/E-gear/Gantry motion. 8: STA_AX_EXT_JOG ---- Axis is controlled by external signal and will execute JOG mode motion once external signal is active. 9: STA_AX_EXT_MPG ---- Axis is controlled by external signal and will execute MPG mode motion once external signal is active.

Return Value:

Error Code.

Comments:

6.3.4.4 Velocity Motion

6.3.4.4.1 Acm_AxMoveVel

Format:

U32 Acm_AxMoveVel (HAND AxisHandle, U16 Direction)

Purpose:

To command axis to make a never ending movement with a specified velocity.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
Direction	U16	IN	Direction: 0: Positive direction; 1: Negative direction.

Return Value:

Error Code.

Comments:

The speed curve is decided by properties: PAR_AxVelLow, PAR_AxVelHigh, PAR_AxAcc, PAR_AxDec, and PAR_AxJerk.

6.3.4.4.2 Acm_AxChangeVel

Format:

U32 Acm_AxChangeVel (HAND AxisHandle, F64 NewVelocity)

Purpose:

To command axis to change the velocity while axis is in velocity motion.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
NewVelocity	F64	IN	New velocity. (unit = PPU/s)

Return Value:

Error Code.

Comments:

The speed curve is decided by properties: PAR_AxVelLow, **NewVelocity**, PAR_AxAcc, PAR_AxDec, and PAR_AxJerk. The range of NewVelocity is: 0 ~ CFG_AxMaxVel.
If this command runs successfully, then NewVelocity will be used in next motion in case the velocity is not specified before the motion.

6.3.4.4.3 Acm_AxGetCmdVelocity

Format:

U32 Acm_AxGetCmdVelocity (HAND AxisHandle, PF64 Velocity)

Purpose:

Get current command velocity of the specified axis.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
Velocity	PF64	OUT	Return the command velocity. (unit = PPU/s)

Return Value:

Error Code.

Comments:

6.3.4.4.4 Acm_AxChangeVelEx

Format:

U32 Acm_AxChangeVelEx (HAND AxisHandle, F64 NewVelocity, F64 NewAcc, F64 NewDec)

Purpose:

Change the velocity, acceleration and deceleration simultaneously in motion status.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
NewVeloc-ity	F64	IN	New velocity. (unit = PPU/s)
NewAcc	F64	IN	New acceleration. (unit = PPU/s^2)

NewDec	F64	IN	New deceleration. (unit = PPU/s^2)

Return Value:

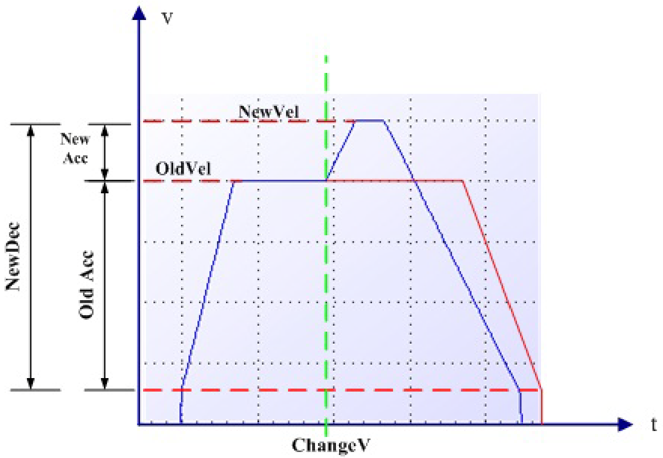
Error Code.

Comments:

NewVelocity should not exceed the maximum specified by CFG_AxMaxVel, **NewAcc** should not exceed the maximum acceleration specified by CFG_AxMaxAcc, and **NewDec** should not exceed the maximum deceleration specified by CFG_AxMaxDec.

If **NewAcc** or **NewDec** is "0", then the previous acceleration or deceleration can be used.

If this command runs successfully, then **NewVelocity** will be used in next motion in case the velocity is not specified before the motion.



6.3.4.4.5 Acm_AxChangeVelExByRate

Format:

U32 Acm_AxChangeVelExByRate (HAND AxisHandle, U32 Rate, F64 NewAcc, F64 NewDec)

Purpose:

Change the velocity, acceleration and deceleration simultaneously in motion status.

Parameters:

Name	Type	In or Out	Description

AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
Velocity	U32	IN	Percentage of velocity change. NewVel = OldVel * Rate *0.01
NewAcc	F64	IN	New acceleration. (unit = PPU/s^2)
NewDec	F64	IN	New deceleration. (unit = PPU/s^2)

Return Value:

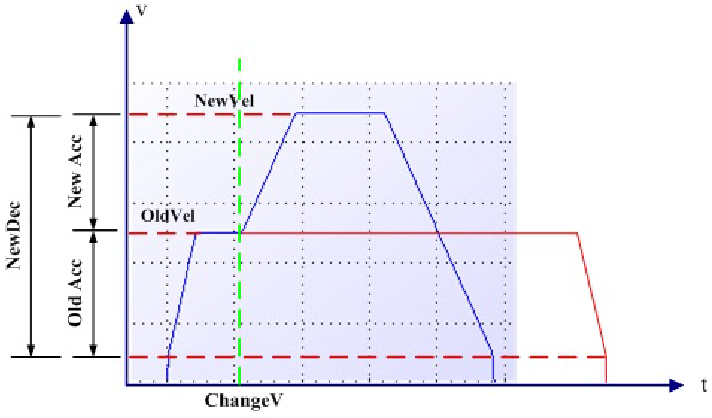
Error Code.

Comments:

NewVel = OldVel***Rate***0.01. The NewVel value caculated by **Rate** should not exceed the maximum specified by CFG_AxMaxVel, **NewAcc** should not exceed the maximum acceleration specified by CFG_AxMaxAcc, and **NewDec** should not exceed the maximum deceleration specified by CFG_AxMaxDec.

If **NewAcc** or **NewDec** is "0", then the previous acceleration or deceleration can be used.

The new velocity, **NewAcc** and **NewDec** is only valid for the current motion.



6.3.4.4.6 Acm_AxChangeVelByRate

Format:

U32 Acm_AxChangeVelByRate (HAND AxisHandle, U32 Rate)

Purpose:

Change the velocity of current motion according to the given rate.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
Rate	U32	IN	Percentage of velocity change.

Return Value:

Error Code.

Comments:

NewVel = OldVel*Rate*0.01. Rate must be more than "0" and lower than the rate of CFG_AxMaxVel to the previous velocity. The new velocity is only valid for the current motion.

6.3.4.5 Point-to-Point Motion

6.3.4.5.1 Acm_AxMoveRel

Format:

U32 Acm_AxMoveRel (HAND AxisHandle, F64 Distance).

Purpose:

Start single axis's relative position motion.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
Distance	F64	IN	Relative distance.(unit = PPU)

Return Value:

Error Code.

Comments:

The speed curve is decided by properties: PAR_AxVelLow, PAR_AxVelHigh, PAR_AxAcc, PAR_AxDec, and PAR_AxJerk.

The range of Distance is: -2147483647 ~ 2147483647, if **PPU** is 1.

6.3.4.5.2 Acm_AxMoveAbs

Format:

U32 Acm_AxMoveAbs (HAND AxisHandle, F64 Position)

Purpose:

Start single axis's absolute position motion.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
Position	F64	IN	Absolute position (unit = PPU)

Return Value:

Error Code.

Comments:

The speed curve is decided by properties: PAR_AxVelLow, PAR_AxVelHigh, PAR_AxAcc, PAR_AxDec, and PAR_AxJerk.

The range of Distance is: -2147483647 ~ 2147483647, if PPU is 1.

6.3.4.5.3 Acm_AxChangePos

Format:

U32 Acm_AxChangePos (HAND AxisHandle, F64 NewDistance)

Purpose:

To command axis to change the end distance while axis is in point to point motion.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
NewDistance	F64	IN	New relative distance. (unit = PPU)

Return Value:

Error Code.

Comments:

This function will change the end position to specified position on current ptp motion.

The range of **Distance** is: -2147483647~2147483647 if **PPU** is 1.

6.3.4.5.4 Acm_AxMoveImpose

Format:

U32 Acm_AxMoveImpose (HAND AxisHandle, F64 Position, F64 NewVel)

Purpose:

Impose a new motion on current motion.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
Position	F64	IN	The relative position of new motion.(unit = PPU)
NewVel	F64	IN	The velocity of this imposed motion. (unit = PPU/s)

Return Value:

Error Code.

Comments:

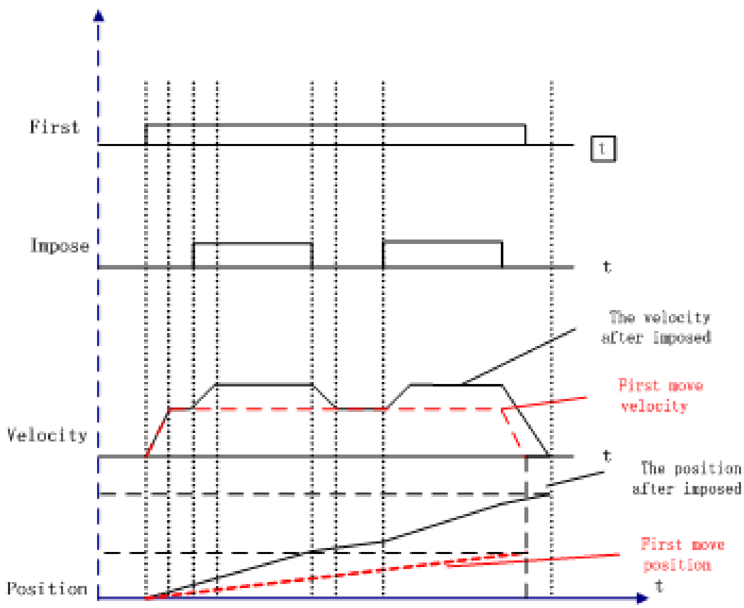
This function just supports trapezoidal profile on this imposed motion.

The end position when motion stops will be the original position added/ subtracted **Position**. And the driver speed will be changed too.

The whole speed curve is decided by **NewVel** of this motion, and PAR_AxVelLow, PAR_AxVelHigh, PAR_AxAcc, PAR_AxDec, PAR_AxJerk of original motion.

The range of **Position** + original position is: -2147483647/ **PPU** ~ 2147483647/ **PPU**, and the range of **NewVel** + orginal FH is 0~ CFG_AxMaxVel.

For example, as follow:



Note: Can not impose new motion on imposed motion.

6.3.4.6 Simultaneous Motion

6.3.4.6.1 Acm_AxSimStartSuspendAbs

Format:

U32 Acm_AxSimStartSuspendAbs (HAND AxisHandle, F64 EndPoint)

Purpose:

Set the axis in wait state for simultaneous operation. When started by start trigger, the axis will start point-to-point absolute moving to the assigned end position.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
EndPoint	F64	IN	The absolute position to move.(unit = PPU)

Return Value:

Error Code.

Comments:

If more than one ax is wanted to do simultaneous operation, should call this function for each axis.

The range of **EndPoint** is: -2147483647/ **PPU** ~ 2147483647/ **PPU**.

6.3.4.6.2 Acm_AxSimStartSuspendRel

Format:

U32 Acm_AxSimStartSuspendRel (HAND AxisHandle, F64 Distance)

Purpose:

Set the axis in wait state for simultaneous operation. When started by start trigger, the axis will start point-to-point relative moving to the assigned end position.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
EndPoint	F64	IN	The relative position to move.(unit = PPU)

Return Value:

Error Code.

Comments:

If more than one ax is wanted to do simultaneous operation, should call this function for each axis.

The range of **EndPoint** is: -2147483647/ **PPU** ~ 2147483647/ **PPU**.

6.3.4.6.3 Acm_AxSimStartSuspendVel

Format:

U32 Acm_AxSimStartSuspendVel (HAND AxisHandle, U16 DriDir)

Purpose:

Set the axis in wait state for simultaneous operation. When started by start trigger, the axis will start continuously moving.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
DriDir	U16	IN	Direction: 0: Positive direction; 1: Negative direction.

Return Value:

Error Code.

Comments:

If more than one ax is wanted to do simultaneous operation, should call this function for each axis.

6.3.4.6.4 Acm_AxSimStart

Format:

U32 Acm_AxSimStart (HAND AxisHandle)

Purpose:

Simultaneous start axis and make it output simultaneous start signal to start all axis that are waiting for start trigger.

Parameter:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.

Return Value:

Error Code.

Comments:

If more than one ax is waiting on start trigger, user should set the mode of simultaneous starting /stopping by CFG_AxSimStartSource before this API.

6.3.4.6.5 Acm_AxSimStop

Format:

U32 Acm_AxSimStop (HAND AxisHandle)

Purpose:

Stop the axis and make the axis output a simultaneous stop trigger to stop all axis that are waiting for the stop trigger.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.

Return Value:

Error Code.

Comments:

When doing simultaneous operation, you can do this operation on any axis to stop all axis if the Simultaneous starting mode is on STA. Or else every simultaneous axis needs to call this API to stop simultaneous motion.

About simultaneous starting/stopping mode, see about CFG_AxSimStartSource.

6.3.4.7 Home

6.3.4.7.1 Acm_AxHome

Format:

U32 Acm_AxHome (HAND AxisHandle, U32 HomeMode, U32 Dir)

Purpose:

To command axis to start typical home motion. The 11 types of typical home motion are composed of extended home.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
HomeMode	U32	IN	HomeMode: 0: MODE1_Abs; 1: MODE2_Lmt; 2: MODE3_Ref; 3: MODE4_Abs_Ref; 4: MODE5_Abs_NegRef; 5: MODE6_Lmt_Ref; 6: MODE7_AbsSearch; 7: MODE8_LmtSearch; 8: MODE9_AbsSearch_Ref; 9: MODE10_AbsSearch_NegRef; 10: MODE11_LmtSearch_Ref; 11: MODE12_AbsSearchReFind; 12: MODE13_LmtSearchReFind; 13: MODE14_AbsSearchReFind_Ref; 14: MODE15_AbsSearchReFind_NegRe; 15: MODE16_LmtSearchReFind_Ref.
Dir	U32	IN	0: Positive direction; 1: Negative direction.

Return Value:

Error Code.

Comments:

During home motion of MODE3_Ref~MODE16_LmtSearchReFind_Ref, the initial velocity will be used in some stages. Therefore, the initial velocity decided by PAR_AxVelLow must be larger than zero.

If property CFG_AxHomeResetEnable is set to be true, command position and actual position will be reset to be zero after home motion ends.

Before using this method, the cross distance should be set through PAR_AxHomeCrossDistance. The speed curve is decided by PAR_AxVelLow, PAR_AxVelHigh, PAR_AxAcc, PAR_AxDec, and PAR_AxJerk.

Explanations:

The meanings of a, b, c and d in the below figures are:

a. The velocity will decrease when trapezoid PTPmotion meets ORG/EL signal.

b. Trapezoid PTP motion moves with HomeCrossDistance as distance until the motion finishes. ORG/EL signal is in effective.

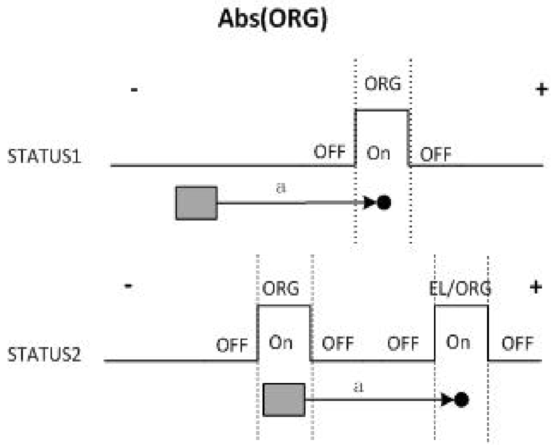
c. Trapezoid PTP take a uniform motion at VelLow. It will stop immediately when it meets ORG/EL signal.

d. Trapezoid PTP motion moves at VelLow with HomeCrossDistance as distance unit until the motion finishes. ORG/EL signal is in effective.

• : This solid black dot means the ending point of a motion.

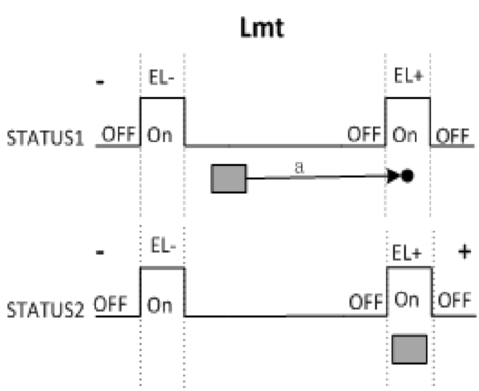
Note: *Features of trapezoid PTP motion: When start, the velocity will increase from VelLow to VelHigh with Acc (If distance is long enough); when end, the velocity will decrease from VelHigh to VelLow with Dec.*

1. MODE1_Abs: Move (Dir) ->touch ORG->Stop.
   Only according to origin equipment (eg.sensor) to home. The object moves continuously until the origin signal occurring.
   For example:
   Dir: Positive.
   Org Logic (CFG_AxOrgLogic): Active High.
   EL (Hard Limit switch) Logic (CFG_AxElLogic): Active High.



- STATUS1: If the object is out of the field of ORG signal, when home command is written, the object will move until ORG signal occurring.
- STATUS2: If the object is in the field of ORG signal or the direction is opposite with ORG switch, the object will move until ORG signal (if there are more than one ORG switch or the axis equipment is occlusive) or EL signal (axis's states is error stop).

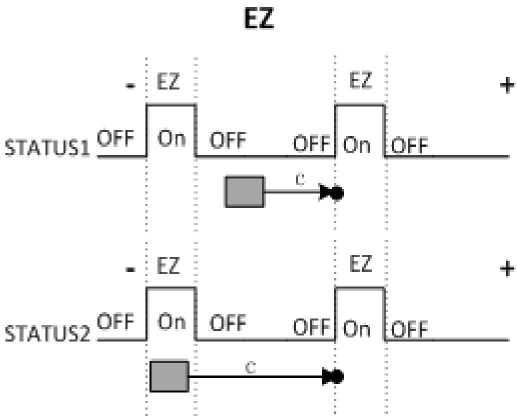
2. MODE2_Lmt: Move(Dir)->touch EL->Stop
Only according to limit equipment (eg.sensor) to home. The object moves continuously until the limit signal occurring.
For Example:
Dir: Positive. Limit Logic (CFG_AxElLogic): Active High.



• STATUS1: If the object is out of the field of EL signal, when home command is written, the object will move until EL signal occurring.

• STATUS2: If the object is in the field of EL signal, there will be no response.

3. MODE3_Ref: Move (Dir) ->touch EZ->Stop.
Only according to EZ to home. The object moves continuously until the EZ signal occurring.
For Example:
Dir: Positive. EZ Logic (CFG_AxEzLogic): Active High.

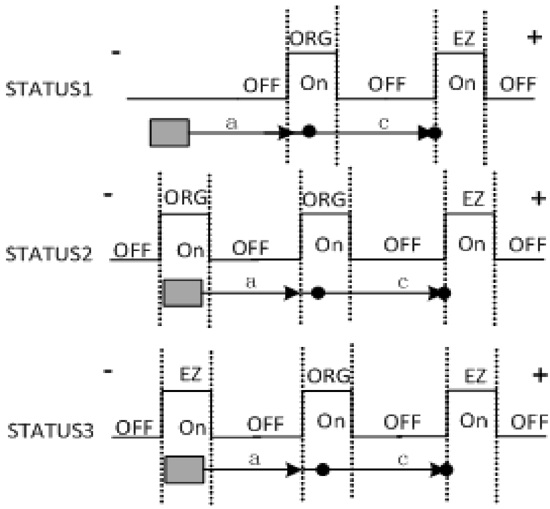


• STATUS1: If the object is out of the field of EZ signal, when home command is written, the object will move until EZ signal occurring.

• STATUS2: If the object is in the field of ORG signal, the object will move until next EZ signal occurring.

4. MODE4_Abs_Ref: ORG+EZ, Move(Dir) ->touch ORG ->Stop ->Move(Dir)->touch EZ ->Stop
This is a composed home mode. Firstly, the object moves until origin signal occurring, and then continues to move in same direction with ORG until EZ signal occurring.
For Example:
Dir: Positive. ORG logic: Active Logic. EZ Logic: Active Logic.

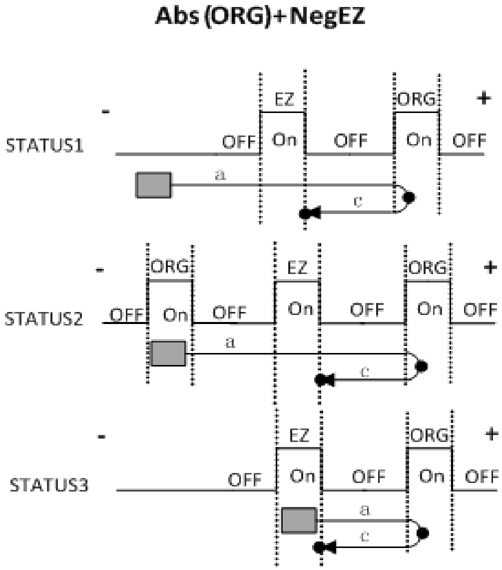
Abs(ORG)+EZ



- STATUS1: If the object is out of the field of EZ signal and ORG signal, when home command is written. Firstly, the object will move until ORG signal occurring, then continue to move until EZ signal occurring.
- STATUS2: If the object is in the field of ORG signal, the home command is written, the object begins to move. Firstly, the ORG signal disappears, and then next ORG signal occurs. At last, motion is stopped when EZ signal occurring.
- STATUS3: If the object is in the field of EZ signal, the home command is written, the object begins to move. Firstly, the EZ signal disappears, and then ORG signal occurs. At last, motion stops when EZ signal occurring.
  Note: Home will stop in case EL signal occurs.

5. MODE5_Abs_NegRef: ORG+EZ, Move (Dir) ->touch ORG ->Stop ->Move (-Dir) ->touch EZ ->Stop.
   This is a composed home mode. The object moves until origin signal occurring firstly, and then continues to move in opposite direction with ORG until EZ signal is occurred.

For Example:
Dir: Positive. ORG logic: Active Logic. EZ Logic: Active Logic.



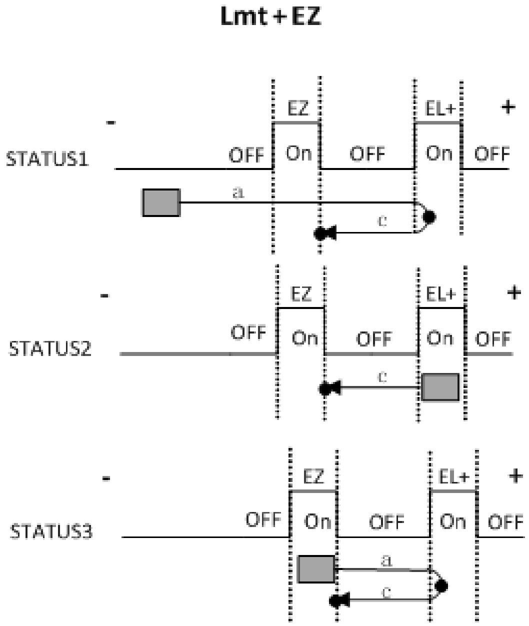
• STATUS1: If the object is out of the field of EZ signal and ORG signal, when home command is written. Firstly, the object will move until ORG signal occurring, then continue to move in opposite direction until EZ signal occurring.

• STATUS2: If the object is in the field of ORG signal, the home command is written, the object begins to move. Firstly, the ORG signal disappears, and then next ORG signal occurs, at the same time reverses motion direction. At last, motion is stopped when EZ signal occurring.

• STATUS3: If the object is in the field of EZ signal, the home command is written, the object begins to move. Firstly, the EZ signal disappears, and then ORG signal occurs, at the same time reverses motion direction. At last, motion stops when EZ signal occurring.
Note: Home will stop in case EL signal occurs.

6. MODE6_Lmt_Ref: EL + NegEZ, Move (Dir) ->touch EL ->Stop -> Move (-Dir) ->touch EZ ->Stop.
The object moves until limit signal occurring firstly, and then con-

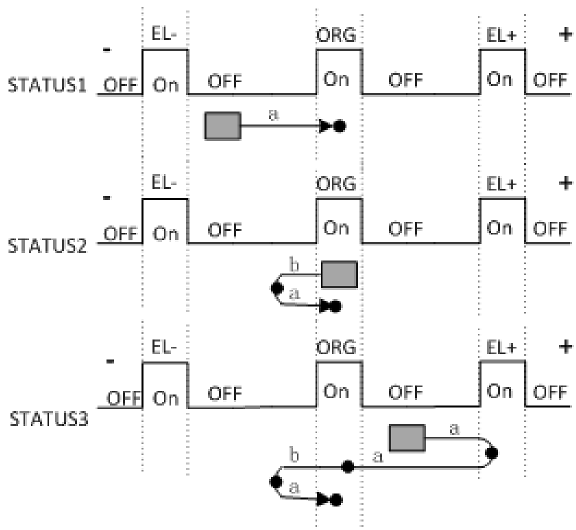
tinues to move in opposite direction until EZ signal is occurred.
For Example:
Dir: Positive. EZ Logic: Active Logic. Limit Logic: Active High.



- STATUS1: If the object is out of the field of EZ signal and EL signal, when home command is written. Firstly, the object will move until EL signal occurring, then continue to move in opposite direction until EZ signal occurring.
- STATUS2: If the object is in the field of EL signal, the object will move in opposite direction until EZ signal occurring.
- STATUS3: If the object is in the field of EZ signal, the home command is written, the object begins to move. Firstly, the EZ signal disappears, and then EL signal occurs, at the same time reverses motion direction. At last, motion stops when EZ signal occurring.

7. MODE7_AbsSearch: Move (Dir) ->Search ORG ->Stop.
This is a mode of searching transformation of ORG signal from no signal to signal occurring.
For Example:
Dir: Positive. ORG logic: Active high. Limit logic: Active High.

AbsSearch



• STATUS1: If there is no ORG signal occurring, the object will stop when ORG signal occurs.

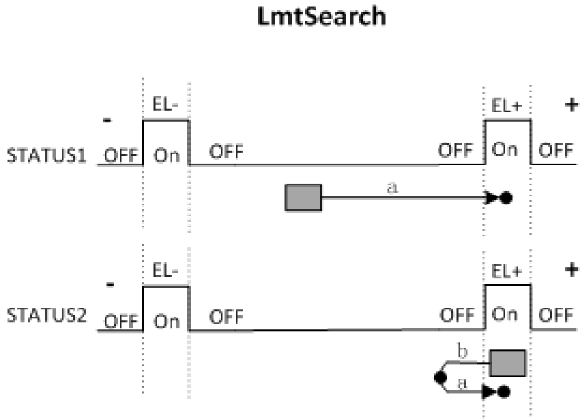
• STATUS2: If the object is in the field of ORG signal. The Object moves in opposite direction until the signal disappears, and then converts direction to move until ORG signal occurring.

• STATUS3: If there is no ORG signal occurring. EL signal happens while moving firstly, the object reverses direction and continues to move, and then the ORG signal from happening to disappearing. Reverses direction again, and moves until ORG signal occurring. Motion stops.

8. MODE8_LmtSearch: Move (Dir) ->Search EL ->Stop.
This is a mode of searching transformation of limit signal from no signal to signal occurring.

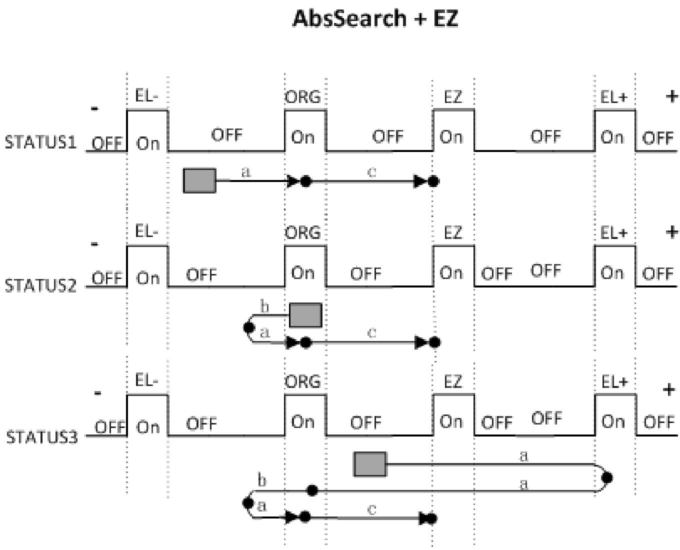
For Example:

Dir: Positive. Limit logic: Active High.



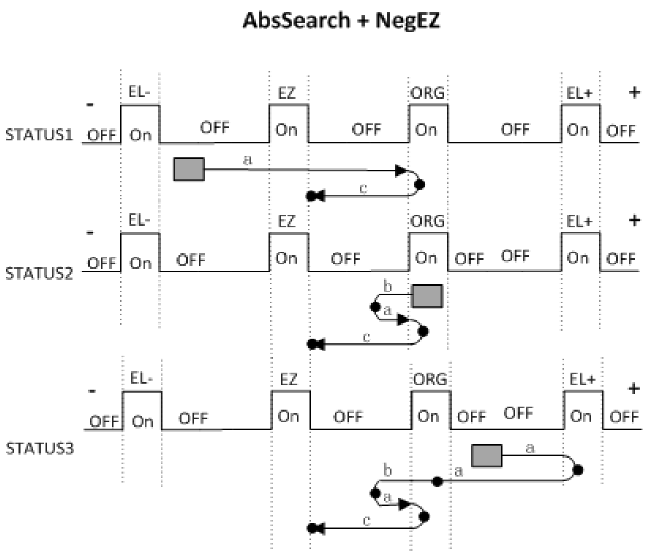
- STATUS1: If the Limit signal is occurred firstly while the object is moving, the home process is end.
- STATUS2: If the object is in the field of limit signal. The Object moves in opposite direction until the signal disappears, and then converts direction to move until limit signal occurring.

9. MODE9_AbsSearch_Ref: Search ORG + EZ, Move (Dir) ->Search ORG ->Stop ->Move (Dir) ->touch EZ ->Stop.
Firstly, object moves in the way of MODE7_AbsSearch, and then moves in same direction until EZ signal occurring.
For example:
Dir: Positive. Limit logic: Active High. ORG logic: Active High.



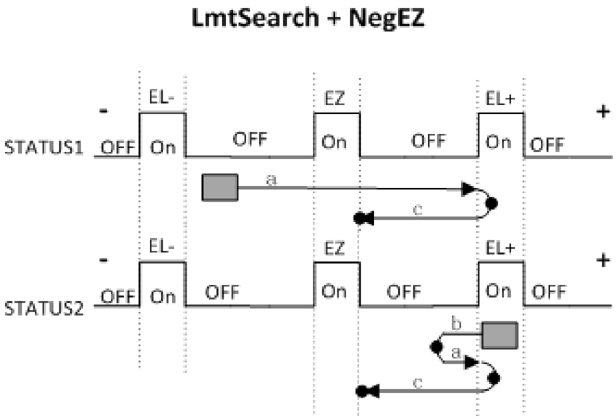
- STATUS1: If the object is out of the field of EZ signal and ORG signal, when home command is written, firstly, the object will move until ORG signal occurring, then continue to move until EZ signal occurring.
- STATUS2: If the object is in the field of ORG signal, the home command is written. Firstly, the object reserves direction and moves, the ORG signal disappears, then reverses direction again and continues to move, the ORG signal occurs again. At last, motion is stopped when EZ signal occurring.
- STATUS3: If there is no ORG signal occurring. EL signal happens before ORG signal, the object reverses direction when EL signal happens and continues to move, and then the ORG signal from happening to disappearing. Reverses direction again, continues to move, the ORG signal will happen and disappear again. At last, motion is stopped when EZ signal occurring.

10. MODE10_AbsSearch_NegRef: Search ORG + NegEZ, Move (Dir) ->Search ORG ->Stop ->Move (-Dir) ->touch EZ ->Stop.
Firstly, object moves in the way of MODE7_AbsSearch, and then moves in opposite direction until EZ signal occurring.
For example:
Dir: Positive. Limit logic: Active High. ORG logic: Active High.



- STATUS1: If the object is out of the field of EZ signal and ORG signal, when home command is written. Firstly, the object will move until ORG signal occurring, then reverse direction and continue to move until EZ signal occurring.
- STATUS2: If the object is in the field of ORG signal, the home command is written, firstly, the object reserves direction and moves, the ORG signal disappears, then reverses direction again and continues to move, the ORG signal occurs again, reverses direction and moves. At last, motion is stopped when EZ signal occurring.
- STATUS3: If there is no ORG signal occurring. EL signal happens before ORG signal, the object reverses direction when EL signal happens and continues to move, and then the ORG signal from happening to disappearing. Reverses direction again, continues to move, the ORG signal will happen again, then reverses direction. At last, motion is stopped when EZ signal occurring.

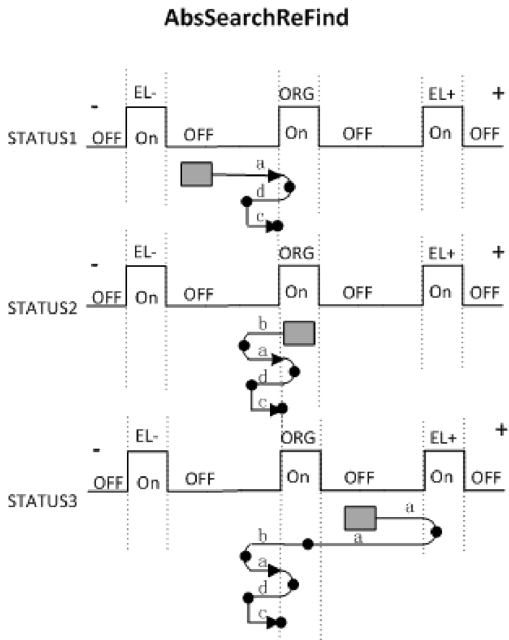
11. MODE11_LmtSearch_Ref: Search EL +NegEZ, Move (Dir) ->Search EL ->Stop->Move (-Dir) ->touch EZ ->Stop.
Firstly, object moves in the way of MODE8_LmtSearch, and then moves in opposite direction until EZ signal occurring.
For example:
Dir: Positive. Limit logic: Active High.



• STATUS1: When object is not in field of limit signal. Firstly, the object will move until EL signal occurring, then reverse direction and continue to move until EZ signal occurring.

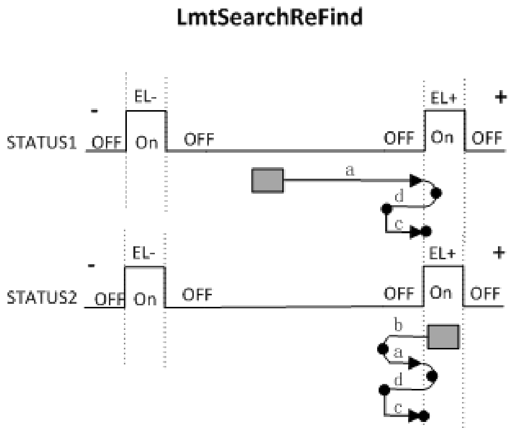
• STATUS2: When object is in the field of limit signal. Firstly, the object reserves direction and moves, the EL signal disappears, then reverses direction again and continues to move, the EL signal occurs again, reverses direction again and moves. At last, motion is stopped when EZ signal occurring.

12. MODE12_ AbsSearchRefind: Search ORG +Refind ORG, Move (Dir) ->Search ORG ->Stop->Move (-Dir) ->Leave ORG(FL) ->Stop-> Move (-Dir)->Refind ORG(FL)->Stop.
Firstly, axis moves in the way of MODE7_AbsSearch, and then moves uniformly in opposite direction at VelLow until ORG signal disappears. Then, axis reverses the direction again and continues to move uniformly at VelLow until ORG singal occurs.
For example:
Dir: Positive.
ORG Logic: Active High.
Limit Logic: Active High.

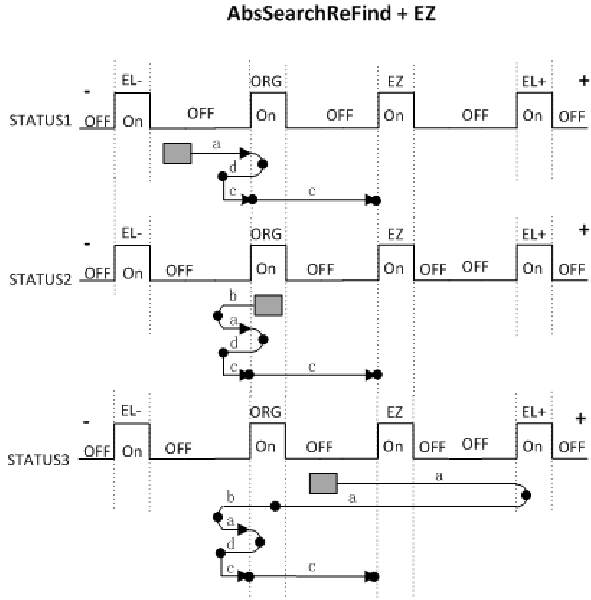


AbsSearch process has three situations. For detailed information, see about descriptions in MODE7_AbsSearch.

13. MODE13_ LmtSearchRefind: Search EL +Refind EL, Move (Dir) ->Search EL ->Stop->Move (-Dir) ->Leave EL(FL) ->Stop-> Move (-Dir)->Refind EL(FL)->Stop.
Firstly, axis moves in the way of MODE8_LmtSearch, and then moves uniformly in opposite direction at VelLow until EL signal disappears. Thent, axis reverses the direction again and continues to move uniformly at VelLow until EL singal occurs.
For example:
Dir: Positive.
Limit Logic: Active High.

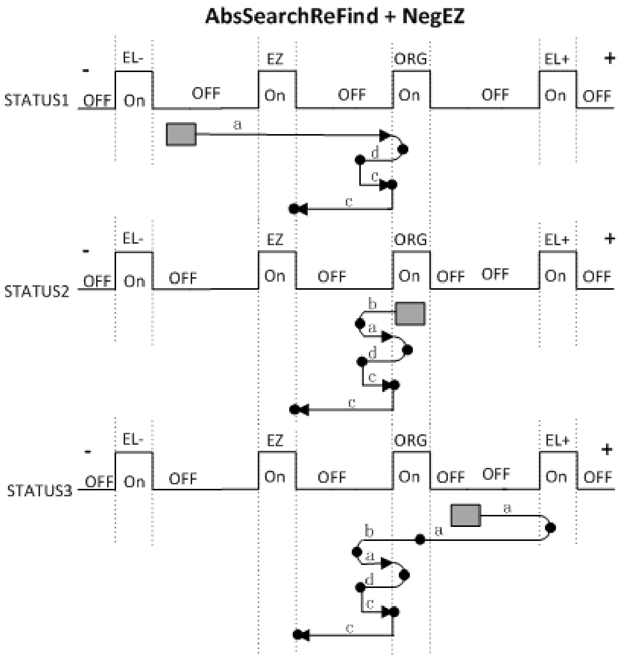


14. MODE14_AbsSearchRefind_Ref: Search ORG +Refind ORG+EZ, Move (Dir) ->Search ORG ->Stop->Move (-Dir) ->Leave ORG(FL) ->Stop-> Move (-Dir)->Refind ORG(FL)->Stop->Move (Dir) ->touch EZ ->Stop.
Firstly, axis moves in the way of MODE7_AbsSearch, and then moves uniformly in opposite direction at VelLow until ORG signal disappears. Then, axis reverses the direction again and continues to move uniformly at VelLow until ORG singal occurs. At last, axis moves in the same direction to Z phase.
For example:
Dir: Positive.
Limit Logic: Active High.
ORG Logic: Active High.



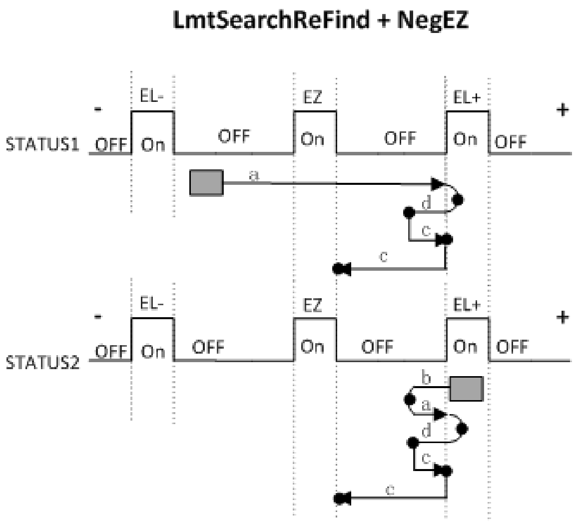
AbsSearch process has three situations. For detailed information, see about descriptions in MODE7_AbsSearch.

15. MODE15_AbsSearchRefind_NegRef: Search ORG +Refind ORG+NegEZ, Move (Dir) ->Search ORG ->Stop->Move (-Dir) ->Leave ORG (FL)->Stop-> Move (-Dir)->Refind ORG(FL)-> Stop-> Move (-Dir) ->touch EZ ->Stop.
Firstly, axis moves in the way of MODE7_AbsSearch, and then moves uniformly in opposite direction at VelLow until ORG signal disappears. Then, axis reverses the direction again and continues to move uniformly at VelLow until ORG singal occurs. At last, axis moves in opposite direction again until EZ signal occurs.
For example:
Dir: Positive.
Limit Logic: Active High.
ORG Logic: Active High.



AbsSearch process has three situations. For detailed information, see about descriptions in MODE7_AbsSearch.

16. MODE16_LmtSearchRefind_Ref: Search EL +Refind EL+EZ, Move (Dir) ->Search EL ->Stop->Move (-Dir) ->Leave EL(FL) ->Stop-> Move (-Dir)->Refind EL(FL)->Stop->Move (-Dir) ->touch EZ ->Stop.
    Firstly, axis moves in the way of MODE8_LmtSearch, and then moves uniformly in opposite direction at VelLow until EL signal disappears. Then, axis reverses the direction again and continues to move uniformly at VelLow until EL singal occurs. At last, axis moves in opposite direction again until EZ signal occurs.
    For example:
    Dir: Positive.
    Limit Logic: Active High.



LmtSearch process has three situations. For detailed information, see about descriptions in MODE8_LmtSearch.

6.3.4.8 Position/Counter Control

6.3.4.8.1 Acm_AxSetCmdPosition

Format:

U32 Acm_AxSetCmdPosition (HAND AxisHandle, F64 Position)

Purpose:

Set command position for the specified axis.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
Position	F64	IN	New command position(uint:PPU)

Return Value:

Error Code.

Comments:

6.3.4.8.2 Acm_AxGetCmdPosition

Format:

U32 Acm_AxGetCmdPosition (HAND AxisHandle, PF64 Position)

Purpose:

Get current command position of the specified axis.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
Position	PF64	OUT	Return the command position.(uint:PPU)

Return Value:

Error Code.

Comments:

6.3.4.8.3 Acm_AxSetActualPosition

Format:

U32 Acm_AxSetActualPosition (HAND AxisHandle, F64 Position)

Purpose:

Set actual position for the specified axis.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
Position	F64	IN	New actual position(uint:PPU)

Return Value:

Error Code.

Comments:

6.3.4.8.4 Acm_AxGetActualPosition

Format:

U32 Acm_AxGetActualPosition (HAND AxisHandle, PF64 Position)

Purpose:

Get current actual position of the specified axis.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
Position	PF64	IN	Return the actual position. (uint:PPU)

Return Value:

Error Code.

Comments:

6.3.4.9 Compare

6.3.4.9.1 Acm_AxSetCmpData

Format:

U32 Acm_AxSetCmpData (HAND AxisHandle, F64 CmpPosition)

Purpose:

Set compare data for the specified axis.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
CmpPosition	F64	IN	The data to be compared. (Unit: PPU)

Return Value:

Error Code.

Comments:

If property CFG_AxCmpMethod is set to **MTD_GREATER_POSITION**, the **CmpPosition** should be greater than current position (command position or actual position). If property CFG_AxCmpMethod is set to **MTD_SMALLER_POSITION**, the **CmpPosition** should be smaller than current position (command position or actual position).

Before setting compare data, you need to set property CFG_AxCmpEnable to **CMP_ENABLE** first. If you want to close compare function, you only need to set property CFG_AxCmpEnable to **CMP_DISABLE** and nothing is necessary to clear compare data. You can set CFG_AxCmpSrc to **SRC_COMMAND_POSITION** or **SRC_ACTUAL_POSITION**.

Once any function of Acm_AxSetCmpData, Acm_AxSetCmpAuto , and Acm_AxSetCmpTable is called, the previous compared data will be cleared.

If property CFG_AxEnableGenDO is enabled, this function will be disabled.

6.3.4.9.2 Acm_AxSetCmpTable

Format:

U32 Acm_AxSetCmpTable (HAND AxisHandle, PF64 TableArray, I32 ArrayCount)

Purpose:

Set compare data list for the specified axis.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
TableArray	PF64	IN	Compare data table. (Unit: PPU)
ArrayCount	I32	IN	Compare data count in the table

Return Value:

Error Code.

Comments:

If property CFG_AxCmpMethod is set to **MTD_GREATER_POSITION**, the first data in table should be greater than current position (command position or actual position) and the compare data should be greater than last compare data in table. After setting the compare table, the first data will be loaded to comparator, and if the current position matches the comparator, pulse will be generated; the next compare data will be loaded to comparator automatically. If property CFG_AxCmpMethod is set to **MTD_SMALLER_POSITION**, the first data in table should be smaller than current position (command position or actual position) and the compare data should be smaller than last compare data in table. The first data will be loaded to comparator, and if the current position matches the comparator, pulse will be generated; the next compare data will be loaded to comparator automatically.
Before setting compare data, set property CFG_AxCmpEnable to **CMP_Enable** first. If you want to close compare function, you only need to set property CFG_AxCmpEnable to **CMP_Disable** and nothing is necessary to clear compare data. You can set **CFG_AxCmpSrc** to SRC_COMMAND_POSITION or SRC_ACTUAL_POSITION.

As long as any of the three functions, Acm_AxSetCmpData, Acm_AxSetCmpAuto , Acm_AxSetCmpTable is called, the previously compared data will be cleared.

At most, there are 100,000 compare data.

This function just supports X axis and Y axis currently.

If CFG_AxGenDoEnable is enabled, the compare function will be disabled, so the compare signal will not be output.

6.3.4.9.3 Acm_AxSetCmpAuto

Format:

U32 Acm_AxSetCmpAuto (HAND AxisHandle, F64 Start, F64 End, F64 Interval)

Purpose:

Set compare data for the specified axis.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
Start	F64	IN	First compare data. (Unit: PPU)
End	F64	IN	Last compare data. (Unit: PPU)
Interval	F64	IN	Compare interval. (Unit: PPU)

Return Value:

Error Code.

Comments:

If property CFG_AxCmpMethod is set to **MTD_GREATER_POSITION**, the **Start** data should be greater than current position (command position or actual position). The first data will be loaded to comparator, and if the current position matches the comparator, pulse will be generated; the next compare data will be loaded to comparator automatically. If property CFG_AxCmpMethod is set to **MTD_SMALLER_POSITION**, the **Start** data should be smaller than current position (command position or actual position). The first data will be loaded to comparator, and if the current position matches the comparator, pulse will be generated; the next compare data will be loaded to comparator automatically.

Before setting compare data, set property CFG_AxCmpEnable to **CMP_Enable** first. If you want to close compare function, you only need to set property CFG_AxCmpEnable to **CMP_Disable** and nothing is necessary to clear compare data. You can set

CFG_AxCmpSrc to SRC_COMMAND_POSITION or SRC_ACTUAL_POSITION.

As long as any of the three functions Acm_AxSetCmpData, Acm_AxSetCmpAuto , Acm_AxSetCmpTable is called, the previously compared data will be cleared.

At most, there are 100,000 compare data.

This function just supports X axis and Y axis currently.

If CFG_AxGenDoEnable is enabled, the compare function will be disabled, so the compare signal will not be output.

6.3.4.9.4 Acm_AxGetCmpData

Format:

U32 Acm_AxGetCmpData (HAND AxisHandle, PF64 CmpPosition)

Purpose:

Get current compare data in the comparator.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
CmpPosition	PF64	OUT	Compare data.(Unit: PPU)

Return Value:

Error Code.

Comments:

6.3.4.10 Latch

6.3.4.10.1 Acm_AxGetLatchData

Format:

U32 Acm_AxGetLatchData (HAND AxisHandle, U32 PositionNo, PF64 Position)

Purpose:

Get the latch data in device after triggering latch.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
PositionNo	U32	IN	0: Command position 1: Actual position.
Position	PF64	OUT	Latch data.(uint:PPU)

Return Value:

Error Code.

Comments:

6.3.4.10.2 Acm_AxTriggerLatch

Format:

U32 Acm_AxTriggerLatch (HAND AxisHandle)

Purpose:

Trigger the hardware to latch position data.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.

Return Value:

Error Code.

Comments:

6.3.4.10.3 Acm_AxResetLatch

Format:

U32 Acm_AxResetLatch (HAND AxisHandle)

Purpose:

Clear the latch data and latch flag in device.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.

Return Value:

Error Code.

Comments:

6.3.4.10.4 Acm_AxGetLatchFlag

Format:

U32 Acm_AxGetLatchFlag (HAND AxisHandle, PU8 LatchFlag)

Purpose:

Get the latch flag in device if data is latched.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.

LatchFlag	PU8	OUT	The flag data. 1: Data is latched. 0: not latched.

Return Value:

Error Code.

Comments:

6.3.4.11 Aux/Gen Output

6.3.4.11.1 Acm_AxDoSetBit

Format:

Acm_AxDoSetBit (HAND AxisHandle, U16 DoChannel, U8 BitData)

Purpose:

Output DO value to channel.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
DoChannel	U16	IN	Digital output channel(4~7)
BitData	U8	IN	DO value: 0 or 1

Return Value:

Error Code.

Comments:

If you want to use this general DO function, you must set property CFG_AxGenDoEnable to **GEN_DO_EN** first. When CFG_AxGenDoEnable is enabled, the function of CamDo, Erc and Compare will be disabled automatically and Acm_AxSetCmpData, Acm_AxSetCmpAuto , Acm_AxSetCmpTable will not be able to generate trigger pulse because these two functions use the same output pins(OUT4 ~ OUT7).

6.3.4.11.2 Acm_AxDoGetBit

Format:

U32 Acm_AxDoGetBit (HAND AxisHandle, U16 DoChannel, PU8 BitData)

Purpose:

Get DO channel status.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
DoChannel	U16	IN	Digital output channel(4~7)
BitData	PU8	OUT	DO value: 0 or 1

Return Value:

Error Code.

Comments:

See about Acm_AxDoSetBit.

6.3.4.11.3 Acm_AxDiGetBit

Format:

U32 Acm_AxDiGetBit (HAND AxisHandle, U16 DiChannel, PU8 BitData)

Purpose:

Get the specified channel's DI value.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
DiChannel	U16	IN	Digital input channel. (0~3)
BitData	PU8	OUT	DI value: 0 or 1

Return Value:

Error Code.

Comments:

6.3.4.12 Ext-Drive

6.3.4.12.1 Acm_AxSetExtDrive

Format:

U32 Acm_AxSetExtDrive (HAND AxisHandle, U16 ExtDrvMode)

Purpose:

Enable or disable external drive mode.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
ExtDrvMode	U16	IN	0: Disabled (stop command) 1: JOG Mode 2: MPG Mode 3: JOG Step mode(reserved)

Return Value:

Error Code.

Comments:

6.3.4.13 Cam/Gear

6.3.4.13.1 Acm_AxCamInAx

Format:

U32 Acm_AxCamInAx (HAND AxisHandle, HAND MasAxis-Handle, F64 MasterOffset, F64 SlaveOffset, F64 MasterScaling, F64 SlaveScaling, U32 CamTableID, U32 RefSrc)

Purpose:

This function starts cam synchronization with a cam table between a slave (following) axis and master (leading) axis.

Camming is done with one table (two dimensional - describing master and slave positions together). The table should be strictly monotonic rising or falling, going both reverse and forward with the master.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen, this handle should be slave (following) axis handle.
MasAxisHandle	HAND	IN	Axis handle from Acm_AxOpen, this handle should be master (leading) axis handle.
MasterOffset	F64	IN	Shifting the cam along the coordinates of the master axis.
SlaveOffset	F64	IN	Shifting the cam along the coordinates of the slave axis.
MasterScaling	F64	IN	Scaling factor for the cam in the coordinates of the master axis. The overall master profile is multiplied by this factor.This should be Larger than zero.
SlaveScaling	F64	IN	Scaling factor for the cam in the coordinates of the slave axis. The overall slave profile is multiplied by this factor. This should be Larger than zero.

CamTableID	U32	IN	Identifier of CAM table. It is assigned by Acm_DevDownloadCAMTable. The PCI-1245 and PCI-1265 reserves 2 cam tables. So the ID can be 0, 1.
RefSrc	U32	IN	Cam table's master position reference to command-position (0) or actual-position (1). 0: Command position. 1: Actual position.

Return Value:

Error Code.

Comments:

If this command is set successfully, salve axis will move according to cam curve fitted by CamTable when master axis is moving continuously or point to point.

Can not set command/acutual position for salve aixs and master axis by Acm_AxSetCmdPosition or Acm_AxSetAcutalPosition.

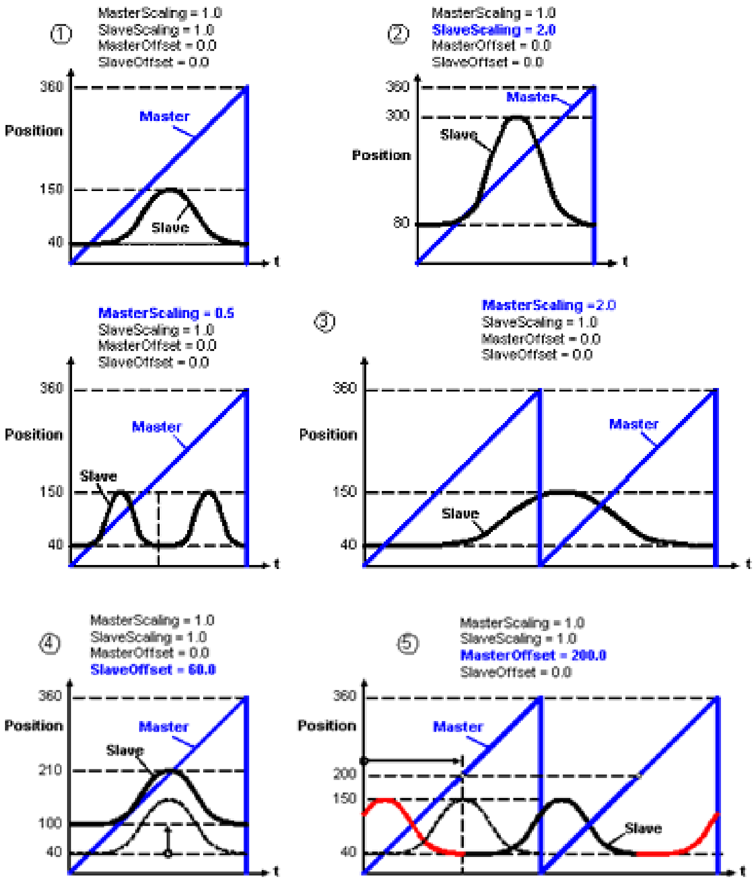
To terminate follow relationship of slave axis, user can call Acm_AxStopDec, Acm_AxStopEmg, and the slave axis will be readby status.

The pulse number of master axis rotated 360 degree should be set by property CFG_AxModuleRange.The edited CamTable needs to set by Acm_DevDownloadCAMTable, and related E-cam configure can be set by Acm_DevConfigCAMTable.

After all of above, slave axis will be synchronous status if calls **Acm_AxCamInAx** successfully. Then slave axis will follow mater axis to move.

The parameter **MasterScaling**, **SlaveScaling**, **MasterOffset**, **SlaveOffset** are used to ajust current Camtable based on edited CamTable.

The figures about scalling and offset are as follow.



About detail E-Cam operation, see **E-Cam Flow Chart** in Chapter 6.2.2.

See Also Acm_DevDownLoadCAMTable, Acm_DevCofigCAMTable, Acm_DevLoadCAMTableFile.

6.3.4.13.2 Acm_AxGearInAx

Format:

U32 Acm_AxGearInAx (HAND AxisHandle, HAND MasAxis-Handle, I32 Numerator, I32 Denominator, U32 RefSrc, U32 Absolute)

Purpose:

This function starts gear synchronization with a ratio between a slave (following) axis and master (leading) axis.

Parameters:

Name	Type	In orOut	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen, this handle should be slave (following) axis handle.
MasAxis-Handle	HAND	IN	Axis handle from Acm_AxOpen, this handle should be master (leading) axis handle.
Numerator	I32	IN	Gear ratio numerator.
Denominator	I32	IN	Gear ratio denominator
RefSrc	U32	IN	Slave axis engages to master axis's command-position (0) or actual-position (1).
Absolute	U32	IN	The synchronization is relative to start position (random position values upon reaching synchronization) or absolute. 0: relative, 1:absolute

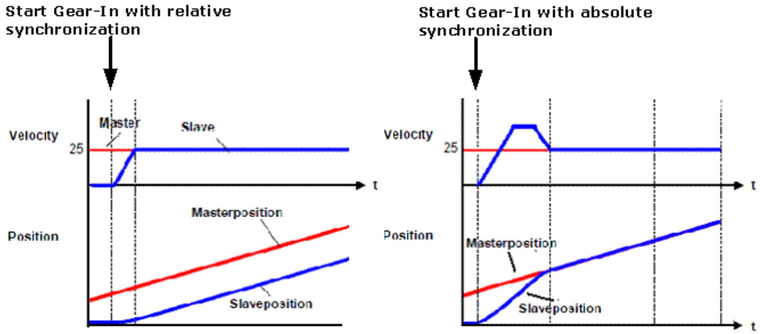
Return Value:

Error Code.

Comments:

Slave axis will follow mater axis motion if this functionis called successfully. The master axis and slave axis can not be reset command/actual position by Acm_AxSetCmdPosition / Acm_AxSetActualPosition in gear motion. The relationship of master and slave can be terminated by calling the Acm_AxStopDec and Acm_AxStopEmg, the axis will return SteadBy status.

- **Gear Ratio:** Numerator/Denominator. If the value is positive, the slave will move at the same direction with master axis, or else it will move at the opponent direction with mater axis.
- **Absolute:**
  - Absolute=1: Absolute relationship. Slave axis will compensate the offset with master axis. Absolute=0: Relative relationship. Slave axis will not compensate any offset with master axis.



About the E-gear operation, see about E-Gear/gantry flow chart in Chapter 6.2.2.

6.3.4.13.3 Acm_AxPhaseAx

Format:

U32 Acm_AxPhaseAx(HAND AxisHandle, F64 Acc, F64 Dec, F64 PhaseSpeed, F64 PhaseDist)

Purpose:

Enable the phase lead or phase lag motion of the salve axis during the process of electronic cam or electronic gear.

Parameters:

Name	Type	In orOut	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
Acc	F64	IN	Acceleration of phase lead/lag motion. (Unit: PPU/s^2)
Dec	F64	IN	Deceleration of phase lead/lag motion. (Unit: PPU/s^2)

PhaseSpeed	F64	IN	Speed of phase lead/lag motion. (Unit: PPU/s)
PhaseDist	F64	IN	Distance of phase lead/lag motion. (Unit: PPU)

Return Value:

Error Code.

Comments:

The API enables the slave axis to pull ahead or lag behind the previous motion during the process of electronic cam or electronic gear. If PhaseDist>0, it enables the phase lead motion; if PhaseDist<0, it enables the phase lag motion.

If the remaining Pulse is not enough, then the slave axis could not reach the specified phase. An error will be retrieved if this command is re-sent before the phase lead/lag motion ends.

Becuase of the floating number caculation error, the maximum acceleration/deceleratoin specified through CFG_AxMaxAcc/CFG_AxMaxDec by the slave axis must exceed Acc/Dec by 100,000 at least.

6.3.4.14 Gantry/Tangent

6.3.4.14.1 Acm_AxTangentInGp

Format:

U32 Acm_AxTangentInGp (HAND AxisHandle, HAND MasGroupHandle, PI16 StartVectorArray, U8 Working_plane, I16 Direction)

Purpose:

Command axis to move at same direction with tangent of group path.

Parameters:

Name	Type	In or Out	Description
AxHandle	HAND	IN	Axis handle from Acm_AxOpen.
MasGroupHandle	HAND	IN	Group handle from Acm_GpAddAxis.
StartVectorArray	PI16	IN	Must be 3 dimension.
Working_plane	U8	IN	0:XY ; 1:YZ ; 2:XZ
Direction	I16	IN	Same:0; Opposite:1

Return Value:

Error Code.

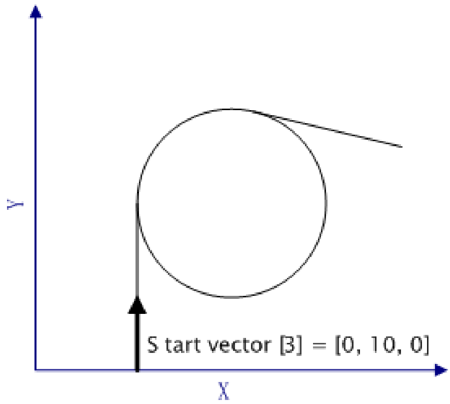
Comments:

The axis will be synchronous state and will follow group motion at the direction of tangent of group's path if this function is called successfully. The axis and group can not be reset command/actual position. Using Acm_AxStopDec/Acm_AxStopEmg will terminate synchronous state of the axis, and it will return SteadBy state.

The axis can not be one axis in group.

StartVectorArray is the initial vector which is starting direction of axis.

For example, StartVectorArray[3] = {0, 10, 0}, working plane = 0: XY plane.



Note: The start vector should be as close as possible with group's motion starting direction, or else there may be the error of motion accelerator greater than max accelerator of device. And if the angle between two conjoint paths is too large, the error will happen too, so user should pay attention to angle between paths. The formula of calculating max angle deviation is as follow:

For example:
Setting:
Module Range (CFG_AxModuleRange): 3600 pulse.
Max Acceleration (CFG_AxMaxAcc): 10^7.
Max angle of slope transformation:

10^7 X 10^-6 X 360° /3600 = 1°.

6.3.4.14.2 Acm_AxGantryIn

Format:

U32 Acm_AxGantryIn (HAND AxisHandle, HAND MasAxisHandle, I16 RefMasterSrc, I16 Direction)

Purpose:

Command two axes to move e-gantry motion.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
MasAxisHandle	HAND	IN	Axis handle from Acm_GpAddAxis.
RefMasterSrc	I16	IN	The reference source: 0: Command position 1: Actual position (reserved)
Direction	I16	IN	The direction with master axis. 0:same; 1:opposite

Return Value:

Error Code.

Comments:

The slave axis will move synchronously with mater axis. The relationship of master and slave can be terminated by calling the Acm_AxStopDec /Acm_AxStopEmg; the axis will return SteadBy status.

There are some restrictions about gantry:

1. Can not set any command except Acm_AxStopDec / Acm_AxStopEmg to slave axis;
2. Slave axis can not be add in any group;
3. If the axis is already one axis in group, it can not be slave axis of ganry.

If the command/actual position of master axis is reset, command/actual position of slave axis is reset same value too.

About the gantry operation, see about E-Gear/gantry flow chart in Chapter 6.2.2.

6.3.4.15 Stop

6.3.4.15.1 Acm_AxStopDec

Format:

U32 Acm_AxStopDec (HAND AxisHandle)

Purpose:

Command the axis to decelerate and stop.

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.

Return Value:

Error Code.

Comments:

If the axis is in synchronous drive mode, for example, the slave axis of E-cam/E-gear/Tangent motion, then the API can be used to terminate the synchronization.

6.3.4.15.2 Acm_AxStopEmg

Format:

U32 Acm_AxStopEmg (HAND AxisHandle)

Purpose:

Command the axis to stop (without decelerating).

Parameters:

Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.

Return Value:

Error Code.

Comments:

If the axis is in synchronous drive mode, for example, the slave axis of E-cam/E-gear/Tangent motion, then the API can be used to terminate the synchronization.

6.3.4.15.3 Acm_AxStopDecEx

Format:

U32 Acm_AxStopDecEx (HAND AxisHandle, F64 NewDec)

Purpose:

Command the axis to stop and specify the deceleration.

Parameters:

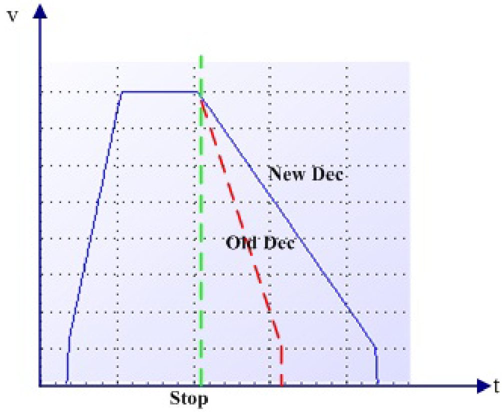
Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
NewDec	F64	IN	Deceleration for decelerating. (Unit: PPU/s^2)

Return Value:

Error Code.

Comments:

If the decelerating command is sent and the remaing pulse is not enough for supporting the specified NewDec, then pulse break will occur.



6.3.5 Group

6.3.5.1 SYSTEM

6.3.5.1.1 Acm_GpAddAxis

Format:

U32 Acm_GpAddAxis (PHAND GpHandle, HAND AxHandle)

Purpose:

Add an axis to the specified group.

Parameters:

Name	Type	In or Out	Description
GpHandle	PHAND	IN/OUT	Point to group handle (NULL or not).
AxHandle	HAND	IN	Axis handle from Acm_AxOpen.

Return Value:

Error Code.

Comments:

If **GpHandle** points to NULL, driver will create a new group handle and add the axis to this new group. If **GpHandle** points to a valid group handle, driver will just add the axis to the group.

At most, there are 2 groups in PCI-1245/1245E and 3 groups in PCI-1265.

The same axis can not be added in different groups.

The master axis in group is the minimal **PhysicalID** one.

The parameters of group are initialized when the first axis is added. Such as, CFG_GpPPU, PAR_GpVelLow, PAR_GpVelHigh, PAR_GpAcc, PAR_GPDec and PAR_GpJerk.

6.3.5.1.2 Acm_GpRemAxis

Format:

U32 Acm_GpRemAxis (HAND GpHandle, HAND AxHandle)

Purpose:

Remove an axis from the specified group.

Parameters:

Name	Type	In or Out	Description
GpHandle	HAND	IN	Group handle from Acm_GpAddaxis.
AxHandle	HAND	IN	Axis handle from Acm_AxOpen.

Return Value:

Error Code.

Comments:

After **Acm_GpRemAxis** is called and no axis is in group, the **GpHandle** can still be used. You can use this group handle to add other axes. But if you have called Acm_GpClose to close this group handle, the group handle can't be used again.

6.3.5.1.3 Acm_GpClose

Format:

U32 Acm_GpClose (PHAND pGroupHandle)

Purpose:

Remove all axis in the group and close the group handle.

Parameters:

Name	Type	In or Out	Description
GpHandle	PHAND	IN	Point to group handle to be closed

Return Value:

Error Code.

Comments:

If the group number is greater than maximal group number of device, new group can not be created. At the time, you must close one existing group if you want to create new group.

6.3.5.1.4 Acm_GpResetError

Format:

U32 Acm_GpResetError (HAND GroupHandle)

Purpose:

Reset group states.

Parameters:

Name	Type	In or Out	Description
GpHandle	HAND	IN	Group handle from Acm_GpAddAxis.

Return Value:

Error Code.

Comments:

If the group is in STA_GP_ERROR_STOP state, the state will be changed to STA_GP_READY after calling this function.

6.3.5.2 Motion Status

6.3.5.2.1 Acm_GpGetState

Format:

U32 Acm_GpGetState (HAND GroupHandle, PU16 pState)

Purpose:

Get the group's current state.

Parameters:

Name	Type	In or Out	Description
GpHandle	HAND	IN	Group handle from Acm_GpAddAxis.
pState	PU16	OUT	Group states: 0:STA_GP_DISABLE 1:STA_GP_READY 2:STA_GP_STOPPING 3:STA_GP_ERROR_STOP 4:STA_GP_MOTION 5:STA_GP_AX_MOTION(not support) 6:STA_GP_MOTION_PATH

Return Value:

Error Code.

Comments:

If an axis of group is implementing command of single-axis motion, the group's state will be unchanged.

6.3.5.2.2 Acm_GpGetCmdVel

Format:

U32 Acm_GpGetCmdVel(HAND GroupHandle, PF64 CmdVel)

Purpose:

Get the current velocity of the group.

Parameters:

Name	Type	In or Out	Description
GroupHandle	HAND	IN	Group handle from Acm_GpAddAxis.
CmdVel	PF64	OUT	Return the current velocity of the group. Unit: PPU/s. (PPU is of the axis with the lowest ID.)

Return Value:

Error Code.

Comments:

Get the current velocity during interpolation or continuous interpolation of the group through API.

6.3.5.3 MotionStop

6.3.5.3.1 Acm_GpStopDec

Format:

U32 Acm_GpStopDec (HAND GroupHandle)

Purpose:

Command axis in this group to decelerate to stop.

Parameters:

Name	Type	In or Out	Description
GpHandle	HAND	IN	Group handle from Acm_GpAddAxis.

Return Value:

Error Code.

Comments:

6.3.5.3.2 Acm_GpStopEmg

Format:

U32 Acm_GpStopEmg(HAND GroupHandle)

Purpose:

Command axis in this group to stop immediately without deceleration.

Parameters:

Name	Type	In or Out	Description
GpHandle	HAND	IN	Group handle from Acm_GpAddAxis.

Return Value:

Error Code.

Comments:

6.3.5.4 Interpolation Motion

6.3.5.4.1 Acm_GpMoveLinearRel

Format:

U32 Acm_GpMoveLinearRel(HAND GroupHandle, PF64 DistanceArray, PU32 pArrayElements)

Purpose:

Command group to execute relative line interpolation.

Parameters:

Name	Type	In or Out	Description
GpHandle	HAND	IN	Group handle from Acm_GpAddAxis.
DistanceArray	PF64	IN	Distance array of axis in group, each value of array elements represent the axis relative position.
pArrayElements	PU32	IN/OUT	Element count in the array(This count must equal to the axis count in this group, or else it will be returned axis count in group)

Return Value:

Error Code.

Comments:

The sequence of data in **DistanceArray** must follow the order of X axis, Y axis, Z axis, U axis. For example, if one group has two axes: Y axis and U axis. The first data in **DistanceArray** means Y axis' relative distance and the second data means U axis' relative distance. The unit of distance in **DistanceArray** is PPU of each axis in group.

The difference between line interpolation and direct interpolation: line interpolation’s speed is divided into even line speed for every axis, axis moves at this even line speed. Mostly, line interpolation is applied to the axis assembled as right angle. But direct interpolation's line speed is set into master axis (The smallest Physical ID axis) and other axes in group start/stop as the same time as master axis. Mostly, direct interpolation is applied to the axis assembled as oblique-angle.

At most, it just supports 3 axes linear interpolation in PCI-1245/ PCI-1265 and 2 axes linear interpolation in PCI-1245V/PCI-1245E.

6.3.5.4.2 Acm_GpMoveLinearAbs

Format:

U32 Acm_GpMoveLinearAbs (HAND GroupHandle, PF64 PositionArray, PU32 pArrayElements)

Purpose:

Command group to execute absolute line interpolation.

Parameters:

Name	Type	In or Out	Description
GpHandle	HAND	IN	Group handle from Acm_GpAddAxis.
PositionArray	PF64	IN	Position array of axis in group, each value of array elements represent the axis absolute position.
pArrayElements	PU32	IN/OUT	Element count in the array(This count must equal to the axis count in this group, or else it will be returned axis count in group)

Return Value:

Error Code.

Comments:

The sequence of data in **PositionArray** must follow the order of X axis, Y axis, Z axis, U axis. For example, if one group has two axes: Y axis and U axis. The first data in **PositionArray** means Y axis' absolute position and the second data means U axis' absolute position. The unit of distance in **PositionArray** is PPU of each axis in group.

The difference between line interpolation and direct interpolation: line interpolation's speed is divided into even line speed for every axis, axis moves at this even line speed. Mostly, line interpolation is applied to the axis assembled as right angle. But direct interpolation's line speed is set into master axis (The smallest Physical ID axis) and other axes in group start/stop as the same time as master axis. Mostly, direct interpolation is applied to the axis assembled as oblique-angle.

At most, it just supports 3 axes linear interpolation in PCI-1245/PCI-1265 and 2 axes linear interpolation in PCI-1245V/PCI-1245E.

6.3.5.4.3 Acm_GpMoveCircularRel

Format:

U32 Acm_GpMoveCircularRel (HAND GroupHandle, PF64 CenterArray, PF64 EndArray, PU32 pArrayElements, I16 Direction)

Purpose:

Command group to execute relative ARC interpolation.

Parameters:

Name	Type	In or Out	Description
GpHandle	HAND	IN	Group handle from Acm_GpAddAxis.
CenterArray	PF64	IN	Axis relative distance of center point.
EndArray	PF64	IN	Axis relative distance of end point.
pArrayElements	PU32	IN/OUT	Element count in the array (This count must equal to the axis count in this group, or else it will be returned axis count in group).

Direction	I16	IN	Direction: 0: DIR_CW(clockwise) 1: DIR_CCW(counterclockwise)

Return Value:

Error Code.

Comments:

The sequence of data in **CenterArray** and **EndArray** must follow the order of X axis, Y axis, Z axis, U axis and so on. For example, if one group has Y axis and U axis, the first data in **CenterArray** means Y axis' center distance and the second data means U axis' center distance. The unit of distance in **CenterArray** and **EndArray** is PPU of each axis in group.

At most, it just supports 2 axes arc interpolation in PCI-1245/1245E/1265. If the axis count in group is 3, user can chose two axes in group to move arc interpolation by PAR_GpRefPlane; the other axes has no respose. If axis count in group is greater than 3, it will be error.

6.3.5.4.4 Acm_GpMoveCircularAbs

Format:

U32 Acm_GpMoveCircularAbs (HAND GroupHandle, PF64 CenterArray, PF64 EndArray, PU32 pArrayElements, I16 Direction)

Purpose:

Command group to execute absolute ARC interpolation.

Parameters:

Name	Type	In or Out	Description
GpHandle	HAND	IN	Group handle from Acm_GpAddAxis.
CenterArray	PF64	IN	Absolute distance of center point.
EndArray	PF64	IN	Absolute distance of end point.
pArrayElements	PU32	IN/OUT	Element count in the array (This count must equal to the axis count in this group, or else it will be returned axis count in group).
Direction	I16	IN	Direction: 0: DIR_CW(clockwise) 1: DIR_CCW(counterclockwise)

Return Value:

Error Code.

Comments:

The sequence of data in **CenterArray** and **EndArray** must follow the order of X axis, Y axis, Z axis, U axis. For example, if one group has Y axis and U axis, the first data in **CenterArray** means Y axis' center position and the second data means U axis' center position. The unit of distance in **CenterArray** and **EndArray** is PPU of each axis in group.

At most, it just supports 2 axes arc interpolation in PCI-1245/1245V/1265. If the axis count in group is 3, user can chose two axes in group to move arc interpolation by PAR_GpRefPlane; the other axes has no respose. If axis count in group is greater than 3, it will be error.

6.3.5.4.5 Acm_GpMoveCircularRel_3P

Format:

U32 Acm_GpMoveCircularRel_3P (HAND GroupHandle, PF64 RefArray, PF64 EndArray, PU32 pArrayElements, I16 Direction)

Purpose:

Command group to execute relative ARC interpolation by three specified points.

Parameters:

Name	Type	In or Out	Description
GpHandle	HAND	IN	Group handle from Acm_GpAddAxis.
RefArray	PF64	IN	Relative distance of reference point.
EndArray	PF64	IN	Relative distance of end point.
pArrayElements	PU32	IN/OUT	Element count in the array (This count must equal to the axis count in this group, or else it will be returned axis count in group).
Direction	I16	IN	Direction: 0: DIR_CW(clockwise) 1: DIR_CCW (counterclockwise)

Return Value:

Error Code.

Comments:

The sequence of data in **RefArray** and **EndArray** must follow the order of X axis, Y axis, Z axis, U axis and so on. For example, if one group has Y axis and U axis, the first data in **RefArray** means Y axis' reference distance and the second data means U axis' reference distance. The unit of distance in **RefArray** and **EndArray** is PPU of each axis in group.

At most, it just supports 2 axes arc interpolation in PCI-1245/1245E/1265. If the axis count in group is 3, user can chose two axes in group to move arc interpolation by PAR_GpRefPlane; the other axes has no respose. If axis count in group is greater than 3, it will be error.

6.3.5.4.6 Acm_GpMoveCircularAbs_3P

Format:

U32 Acm_GpMoveCircularAbs_3P (HAND GroupHandle, PF64 RefArray, PF64 EndArray, PU32 pArrayElements, I16 Direction)

Purpose:

Command group to execute absolute ARC interpolation by three specified points.

Parameters:

Name	Type	In or Out	Description
GpHandle	HAND	IN	Group handle from Acm_GpAddAxis.
RefArray	PF64	IN	Absolute position of reference point.
EndArray	PF64	IN	Absolute position of end point.
pArrayElements	PU32	IN/OUT	Element count in the array (This count must be equal to the axis count in this group, or else it will be returned axis count in group).
Direction	I16	IN	Direction: 0: DIR_CW (clockwise) 1: DIR_CCW (counterclockwise)

Return Value:

Error Code.

Comments:

The sequence of data in **RefArray** and **EndArray** must follow the order of X axis, Y axis, Z axis, U axis and so on. For example, if one group has Y axis and U axis, the first data in **RefArray** means Y axis' reference position and the second data means U axis' reference position. The unit of distance in **RefArray** and **EndArray** is PPU of each axis in group.

At most, it just supports 2 axes arc interpolation in PCI-1245/1245V/1265. If the axis count in group is 3, user can chose two axes in group to move arc interpolation by PAR_GpRefPlane; the other axes has no respose. If axis count in group is greater than 3, it will be error.

6.3.5.4.7 Acm_GpMoveDirectAbs

Format:

U32 Acm_GpMoveDirectAbs (HAND GroupHandle, PF64 PositionArray, PU32 ArrayElements)

Purpose:

Command group to execute absolute direct line interpolation.

Parameters:

Name	Type	In or Out	Description
GpHandle	HAND	IN	Group handle from Acm_GpAddAxis.
PositionArray	PF64	IN	Distance array of axis in group, each value of array elements represent the axis absolute position.
pArrayElements	PU32	IN/OUT	Element count in the array (This count must equal to the axis count in this group, or else it will be returned axis count in group).

Return Value:

Error Code.

Comments:

The sequence of data in **PositionArray** must follow the order of X axis, Y axis, Z axis, U axis and so on. For example, if one group has two axes: Y axis and U axis. The first data in **PositionArray** means Y axis' absolute position and the second data means U axis' absolute position. The unit of distance in **PositionArray** is PPU of each axis in group.

The difference between line interpolation and direct interpolation: line interpolation's speed is divided into even line speed for every axis, axis moves at this even line speed. Mostly, line interpolation is applied to the axis assembled as right angle. But direct interpolation's line speed is set into master axis (The smallest Physical ID axis) and other axes in group start/stop as the same time as master axis. Mostly, direct interpolation is applied to the axis assembled as oblique-angle.

At most, it just supports 4 axes direct interpolation in PCI-1245, 2 axes in PCI-1245V/PCI-1245E and 6 axes in PCI-1265.

6.3.5.4.8 Acm_GpMoveDirectRel

Format:

U32 Acm_GpMoveDirectRel (HAND GroupHandle, PF64 DistanceArray, PU32 ArrayElements)

Purpose:

Command group to execute relative direct line interpolation.

Parameters:

Name	Type	In or Out	Description
GpHandle	HAND	IN	Group handle from Acm_GpAddAxis.
DistanceArray	PF64	IN	Distance array of axis in group, each value of array elements represent the axis relative position.
ArrayElements	PU32	IN/OUT	Element count in the array(This count must equal to the axis count in this group, or else it will be returned axis count in group)

Return Value:

Error Code.

Comments:

The sequence of data in **DistanceArray** must follow the order of X axis, Y axis, Z axis, U axis and so on. For example, if one group has two axes: Y axis and U axis. The first data in **DistanceArray** means Y axis' relative distance and the second data means U axis' relative distance. The unit of distance in **DistanceArray** is PPU of each axis in group.

The difference between line interpolation and direct interpolation: line interpolation’s speed is divided into even line speed for every axis, axis moves at this even line speed. Mostly, line interpolation is applied to the axis assembled as right angle. But direct interpolation's line speed is set into master axis (The smallest Physical ID axis) and other axes in group start/stop as the same time as master axis. Mostly, direct interpolation is applied to the axis assembled as oblique-angle.

At most, it just supports 4 axes direct interpolation in PCI-1245, 2 axes in PCI-1245V/PCI-1245E and 6 axes in PCI-1265.

6.3.5.4.9 Acm_GpChangeVel

Format:

U32 Acm_GpChangeVel (HAND GroupHandle, F64 NewVelocity)

Purpose:

Command group to change the velocity while group is in line-interpolation motion.

Parameters:

Name	Type	In or Out	Description
GpHandle	HAND	IN	Group handle from Acm_GpAddAxis.
NewVelocity	F64	IN	New velocity. (unit: PPU/s)

Return Value:

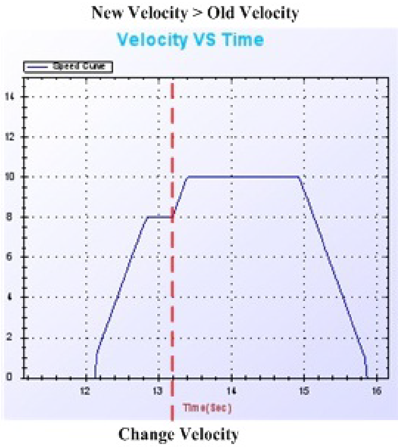
Error Code.

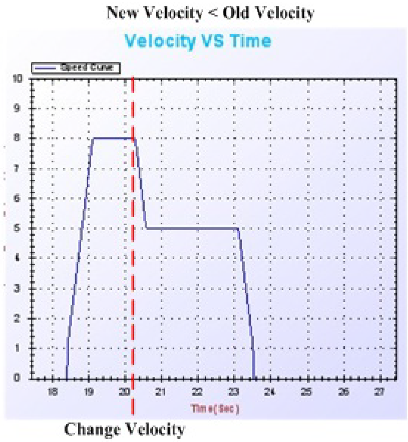
Comments:

This function is not supported by PCI-1245E.

When group is in motion status, the velocity changing command can be sent, in order for the motion card to change the velocity accordingly. When group is in line-interpolation, circular interpolation, helical interpolation or continuous interpolation motion, this command can be sent to change the velocity. If the command runs successfully, NewVelocity will be used in next motion if the velocity is not specified before the motion.

1.In single step interpolation motion, the velocity changing process is shown below:





If the remaing pulse is not enough for getting to the new velocity, the card will automatically calculate the available velocity.

2.In continuous interpolation motion
•BufferMode: Blending Disabled

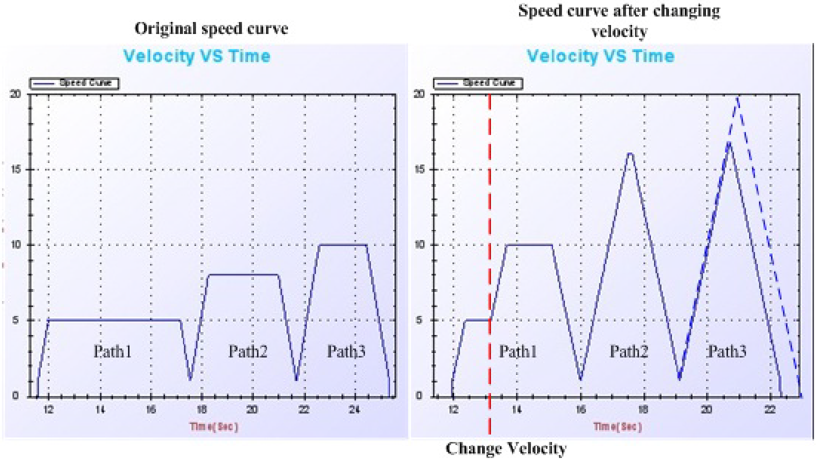
In this mode, each path in Path Buffer has its own process of acceleration/deceleration. If the ChangeV function is run in motion status, then the velocity of path of the current phase will accelerate/decelerate to new velocity, and thus the velocity of each phase will increase/decrease proportionally. If the remaining pulse is not enough when the velocity of each phase is being changed, then the new velocity will be automatically calculated.

For example: path1: VL = 1000, VH = 5000

path 2: VL = 1000, VH = 8000

path 3: VL = 1000, VH = 10000

During Path1, if ChangeV is run and New Velocity = 10000, then velocity of the second path should be 16000, and velocity of the third path should be 20000. However, the real status is as below:



•BlendingMode: Blending Enable, BlendingTime >0

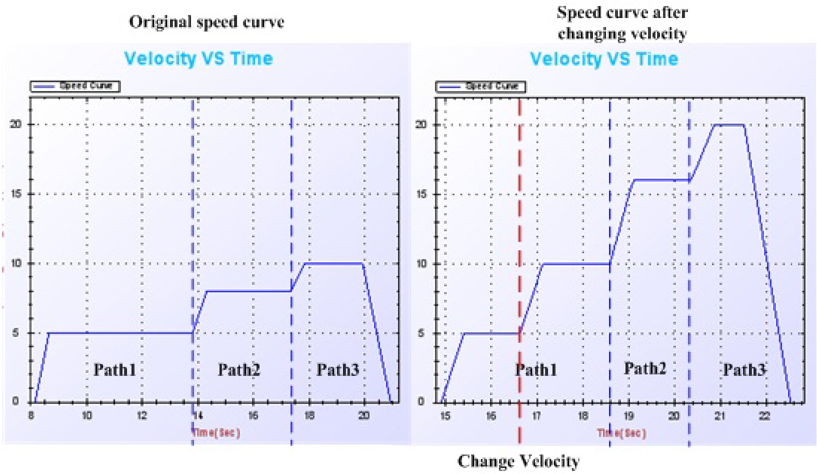
In this mode, each path in Path Buffer doesn't have a complete process of acceleration/deceleration. Its velocity path is decided by BlendingTime and the FL of each phase. For details, please refer to Acm_GpAddPath. If the ChangeV function is run in motion status, then the velocity of path of the current phase will accelerate/decelerate to new velocity, and thus the velocity of each phase will increase/decrease proportionally. If the remaining pulse is not enough when the velocity of each phase is being changed, then the new velocity will be automatically calculated.

For example: path1: VL = 1000, VH = 5000

path 2: VL = 1000, VH = 8000

path 3: VL = 1000, VH = 10000

During Path1, if ChangeV is run and New Velocity = 10000, then velocity of the second path should be 16000, and velocity of the third path should be 20000. However, the real status is as below:



If the velocity changing command is sent in Blending phase, then the ChangeV function will be delayed to next path.

•FlyMode: Blending Enable, BlendingTime=0

The volocity path in this mode is similar to that in Blending mode, while the velocity between two paths will not decelerate to FL. For details, please refer to Acm_GpAddPath. If the ChangeV function is run in motion status, then the velocity of path of the current phase will accelerate/decelerate to new velocity, and thus the velocity of each phase will increase/decrease proportionally. If the remaining pulse is not enough when the velocity of each phase is being changed, then the new velocity will be automatically calculated.

Note: *The ChangeV function is not supported by deceleration phase.*

6.3.5.4.10 Acm_GpMoveHelixAbs

Format:

U32 Acm_GpMoveHelixAbs (HAND GroupHandle, PF64 CenterArray, PF64 EndArray, PU32 pArrayElements, I16 Direction)

Purpose:

Command group to move absolute spiral.

Parameters:

Name	Type	In or Out	Description
GpHandle	HAND	IN	Group handle from Acm_GpAddAxis.
CenterArray	PF64	IN	Absolute distance of center point.
EndArray	PF64	IN	Absolute distance of end point.
pArrayElements	PU32	IN/OUT	Element count in the array (This count must be equal to the axis count in this group, or else it will be returned axis count in group).
Direction	I16	IN	Direction: 0: DIR_CW (clockwise) 1: DIR_CCW (counterclockwise)

Return Value:

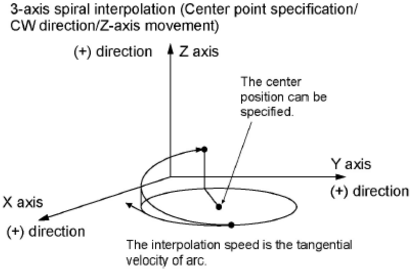
Error Code.

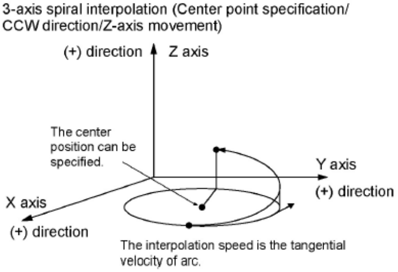
Comments:

This command just supports 3 axes. The elments in **CenterArray** and **EndArray** must be in order with PhysicalID of axis. User can chose two axes in group to move arc interpolation by property PAR_GpRefPlane; the other axes decides the height of helix. The unit of distance in **CenterArray** and **EndArray** is PPU of each axis in group.
For example:

Group(Y, Z, U), **CenterArray(Y, Z, U)**, **EndArray(Y, Z, U)**. If PAR_GpRefPlane =1(YZ_Plane), Z axis and U axis will move arc interpolation, and Y value in EndArray is the height of helix.

The figure of helix is as follow.





6.3.5.4.11 Acm_GpMoveHelixRel

Format:

U32 Acm_GpMoveHelixRel (HAND GroupHandle, PF64 Center-Array, PF64 EndArray, PU32 pArrayElements, I16 Direction)

Purpose:

Command group to move relative spiral.

Parameters:

Name	Type	In or Out	Description
GpHandle	HAND	IN	Group handle from Acm_GpAddAxis.
CenterArray	PF64	IN	Relative distance of center point.
EndArray	PF64	IN	Relative distance of end point.
pArrayElements	PU32	IN/OUT	Element count in the array (This count must equal to the axis count in this group, or else it will be returned axis count in group).
Direction	I16	IN	Direction: 0: DIR_CW (clockwise) 1: DIR_CCW (counterclockwise)

Return Value:

Error Code.

Comments:

See about Acm_GpMoveHelixAbs.

6.3.5.4.12 Acm_GpMoveHelixAbs_3P

Format:

U32 Acm_GpMoveHelixAbs_3P (HAND GroupHandle, PF64 RefArray, PF64 EndArray, PU32 pArrayElements, I16 Direction)

Purpose:

Command group to move absolute spiral by three specified points.

Parameters:

Name	Type	In or Out	Description
GpHandle	HAND	IN	Group handle from Acm_GpAddAxis.
RefArray	PF64	IN	Absolute distance of reference point.
EndArray	PF64	IN	Absolute distance of end point.
pArrayElements	PU32	IN/OUT	Element count in the array (This count must equal to the axis count in this group, or else it will be returned axis count in group).
Direction	I16	IN	Direction: 0: DIR_CW (clockwise) 1: DIR_CCW (counterclockwise)

Return Value:

Error Code.

Comments:

There must be 3 axes in group.

Command helix motion by assigning three points.The orders of valus in parameter **RefArray**, **CenterArray** and **EndArray** must follow the order of axis PhysicalID. The unit of distance in **RefArray** and **EndArray** is PPU of each axis in group.

User can chose two axes in group to move arc interpolation by PAR_GpRefPlane.

For example:

Group(Y, Z, U), **RefArray(Y, Z, U)**, **CenterArray(Y, Z, U)**, **EndArray(Y, Z, U)**, PAR_GpRefPlane =1(YZ_Plane).

Z axis and U axis will move arc interpolation, and Y value in **EndArray** is the height of helix.

6.3.5.4.13 Acm_GpMoveHelixRel_3P

Format:

U32 Acm_GpMoveHelixRel_3P (HAND GroupHandle, PF64 RefArray, PF64 EndArray, PU32 pArrayElements, I16 Direction)

Purpose:

Command group to move relative spiral by three specified points.

Parameters:

Name	Type	In or Out	Description
GpHandle	HAND	IN	Group handle from Acm_GpAddAxis.
RefArray	PF64	IN	Relative distance of reference point.
EndArray	PF64	IN	Relative distance of end point.
pArrayElements	PU32	IN/OUT	Element count in the array (This count must equal to the axis count in this group, or else it will be returned axis count in group).
Direction	I16	IN	Direction: 0: DIR_CW(clockwise) 1: DIR_CCW (counterclockwise)

Return Value:

Error Code.

Comments:

See about Acm_GpMoveHelixAbs_3P.

6.3.5.4.14 Acm_GpMoveCircularRel_Angle

Format:

U32 Acm_GpMoveCircularRel_Angle (HAND GroupHandle, PF64 CenterArray, U16 Degree, PU32 ArrayElements,I16 Direction)

Purpose:

Complete circular interpolation through relative center coordinates, and angle & direction of rotaion.

Parameters:

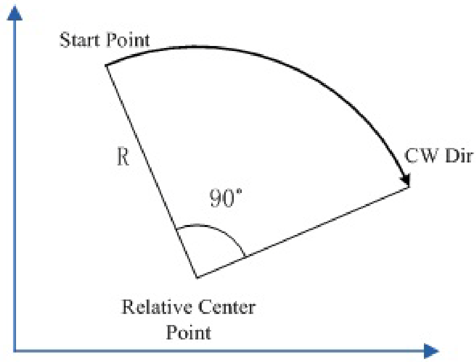
Name	Type	In or Out	Description

GroupHandle	HAND	IN	Group handle from Acm_GpAddAxis.
CenterArray	PF64	IN	Relative distance between center point and start point.
Degree	U16	IN	Angle of rotation. (Range: 0~360)
pArrayElements	PU32	IN	Axis count.
Direction	I16	IN	Direction: 0: CW-Dir 1: CCW_Dir

Return Value:

Error Code.

Comments:



6.3.5.4.15 Acm_GpMoveCircularAbs_Angle

Format:

U32 Acm_GpMoveCircularAbs_Angle (HAND GroupHandle, PF64 CenterArray, U16 Degree, PU32 ArrayElements,I16 Direction)

Purpose:

Complete circular interpolation through absolute center coordinates, and angle & direction of rotaion.

Parameters:

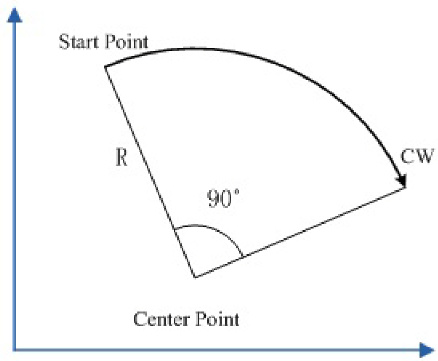
Name	Type	In or Out	Description
GpHandle	HAND	IN	Group handle from Acm_GpAddAxis.

CenterArray	PF64	IN	Center coordinates.
Degree	U16	IN	Angle of rotation. (Range: 0~360)
ArrayElements	PU32	IN	Axis count.
Direction	I16	IN	Direction: 0: CW-Dir 1: CCW_Dir

Return Value:

Error Code.

Comments:



6.3.5.4.16 Acm_GpChangeVelByRate

Format:

U32 Acm_GpChangeVelByRate(HAND GroupHandle, U32 Rate)

Purpose:

Change the velocity of the current group motion according to the specified ratio.

Parameters:

Name	Type	In or Out	Description
GpHandle	HAND	IN	Group handle from Acm_GpAddAxis.

Return Value:

Error Code.

Comments:

New velocity = Former velocity of group * rate *0.01. Rate must be larger than zero, and less than the ratio of the maximum velocity to current group velocity of the axis with the lowest ID. New velocity is valid for the current motion only. For more details about changing velocity in interpolation or continuous interpolation motion, please refer to Acm_GpChangeVel.

6.3.5.5 Path

6.3.5.5.1 Acm_GpAddPath

Format:

U32 Acm_GpAddPath (HAND GroupHandle, U16 MoveCmd, U16 MoveMode, F64 FH, F64 FL, PF64 EndPoint_DataArray, PF64 CenPoint_DataArray, PU32 ArrayElements)

Purpose:

Add an interpolation path to system path buffer.

Parameters:

Name	Type	In or Out	Description
GpHandle	HAND	IN	Group handle from Acm_GpAddAxis.

MoveCmd	U16	IN	Move command: 0:EndPath 1: Abs2DLine; 2: Rel2DLine; 3: Abs2DArcCW; 4: Abs2DArcCCW; 5: Rel2DArcCW; 6: Rel2DArcCCW; 7: Abs3DLine; 8: Rel3DLine; 9: Abs4DLine;(not support) 10: Rel4DLine; (not support) 11: Abs2DDirect; 12: Rel2DDirect; 13: Abs3DDirect; 14: Rel3DDirect; 15: Abs4DDirect; 16: Rel4DDirect; 17: Abs5DDirect; 18: Rel5DDirect; 19: Abs6DDirect; 20: Rel6DDirect; 21: Abs3DArcCW; (not support) 22: Rel3DArcCW; (not support) 23: Abs3DArcCCW; (not support) 24: Rel3DArcCCW;(not support) 25: Abs3DHelixCW 26: Rel3DHelixCW 27: Abs3DHelixCCW 28: Rel3DHelixCCW 29: GPDELAY (uint:ms)
MoveMode	U16	IN	Move mode: 0: No blending 1: Blending
FH	F64	IN	High velocity / delay time for GPDELAY move command. (driving velocity) (Unit:PPU/s of group)
FL	F64	IN	Low velocity (start velocity) (Unit:PPU/s of group)

EndPoint_DataArray	PF64	IN	End points (Unit: PPU of each axis)
CenPoint_DataArray	PF64	IN	Center points (Unit: PPU of each axis)
ArrayElements	PU32	IN/OUT	Number of array element can not be less than axis count in group, or else it will be returned axis count in group.

Return Value:

Error Code.

Comments:

The group handle of every path in system buffer must be the same. So, if there are some unexecuted paths in system buffer and you want to add new path into it by call **Acm_GpAddPath**, the parameter GroupHandle must be the same with the first unexecuted path's group handle. The current status of system path buffer can be got by call Acm_GpGetPathStatus. Path data in buffer will be loaded to hardware execution registers sequentially after calling Acm_GpMovePath.

The absolute commands and relative commands can not be mixed in system path buffer except **EndPath** and **GPDELAY**, or else the error will be returned.

The **ArrayElements** parameter which is element count in **EndPoint_DataArray** parameter and **CenPoint_DataArray** can not be less than axis count in group.

All of paths needed axis count according to motion command is not greater than axis count of group can be loaded in same system buffer. eg: axis count in group is 4, then the paths with **Rel2DLine**, **Rel3DLine**, **4DDirect** and so on can be loaded into device. If the axis count in group is greater than needed axis count in path, the foregoing axis in group will be chosen to move; Especially for **Abs2DArcCW**, **Abs2DArcCCW**, Rel2DArcCW and **Rel2DArcCCW**, user can chose two axes in three foregoing axes in group by Par_GpRefPlane to move. At the last, the **EndPath** command must be add in path buffer.

At most, there are 10000 paths in group.

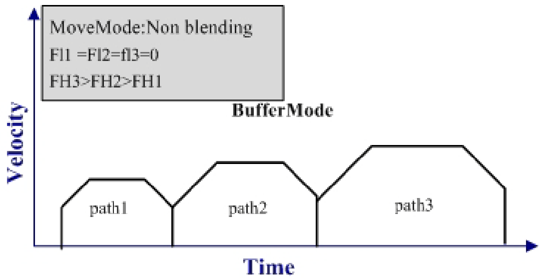
User can enable/disable speed-forward function by CFG_GpSFEnable and set blending time by CFG_GpBldTime before calling Acm_GpAddPath.

GPDELAY: Delay command. The group will delay some time to move next path. User can set time by **FH**. The path with this command cannot be added when speed-forward function is enabled by CFG_GpSFEnable. The unit of delay time is ms.

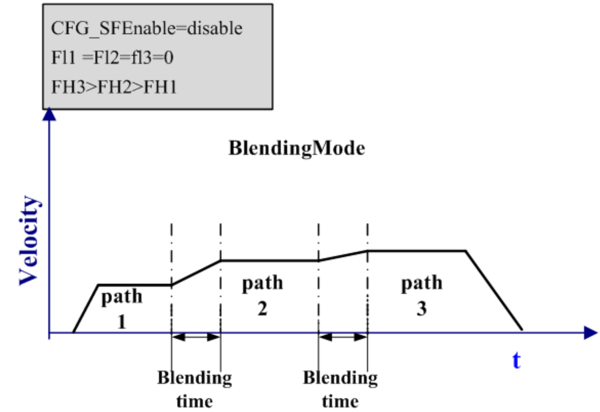
MoveMode: Blending Mode and Non-Blending Mode. CFG_GpSFEnable should be disabled in Non-Blending mode.

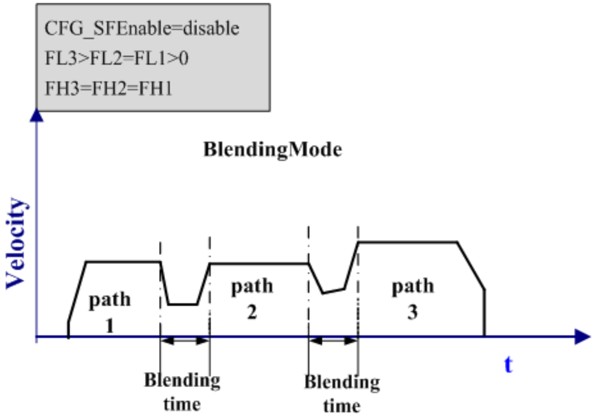
There will be three mode according to the setting of **MoveMode** and CFG_GpBldTime, they are: BufferMode, BlendingMode, FlyMode.

1. **BufferMode**: When the **MoveMode** is Non-Blending.
   At this mode, each path has the whole process of accelerating and decelerating. In this mode, the Speed Foward function can not be supported, so CFG_GpSFEnable should be disabled.

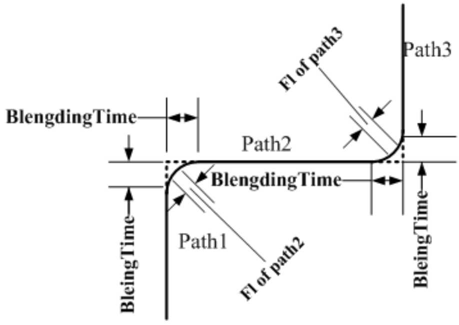


2. **BlendingMode**: When **MoveMode** is Blending and the value of CFG_GpBldTime is greater than zero. These mode doses not support S profile acceleration. When speed forward function is enabled by CFG_SFEnable, the parameter FL and FH is useless because all of speed parameters used in path motion are of group speed setting, all of path have the same driving speed. For example, CFG_SFEnable = Disable, BlendingTime>0. The speed curves are as follows:

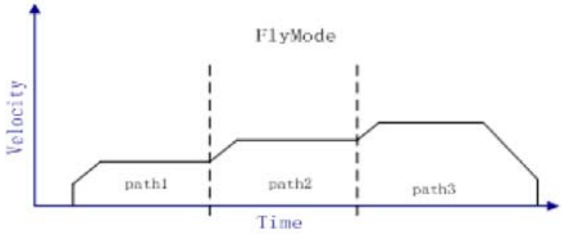




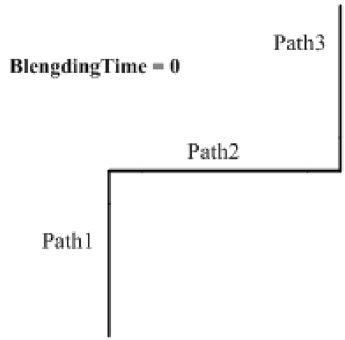
The path is as follow:



3. FlyMode: When MoveMode is Blending and the value of CFG_GpBldTime is 0. When speed forward function is enabled by CFG_SFEnable, the parameter FL and FH is useless because all of speed parameters used in path motion are of group speed setting, all of path have the same driving speed.
The speed curve:



The motion path:



6.3.5.5.2 Acm_GpResetPath

Format:

U32 Acm_GpResetPath (PHAND GroupHandle)

Purpose:

Clear system path buffer. If there is group executing path, the path motion will be stopped.

Parameters:

Name	Type	In or Out	Description
GpHandle	PHAND	IN	Pointer to group handle from Acm_GpAddAxis.

Return Value:

Error Code.

Comments:

6.3.5.5.3 Acm_GpLoadPath

Format:

U32 Acm_GpLoadPath(HAND GroupHandle, PI8 FilePath, PHAND PathHandle, PU32 pTotalCount)

Purpose:

Load path data from path file. It can load up to 600 path data one time.

Parameters:

Name	Type	In or Out	Description
GpHandle	HAND	IN	Group handle from Acm_GpAddAxis.
FilePath	PI8	IN	Point to a file path name of the motion path data to be loaded.
PathHandle	PHAND	OUT	Return the pointer to path handle
pTotalCount	PU32	OUT	Return actual total count of path data in the path file

Return Value:

Error Code.

Comments:

The path data file (binary) is usually generated by Motion Utility's [**Path Editor**]. If you are familiar with Advantech motion product, you can create file by yourself. The **PathHandle** must be unloaded by Acm_GpUnloadPath when the **PathHandle** does not be used any more or application is closing, and the paths contained in **PathHandle** are deleted from driver at the same time.

6.3.5.5.4 Acm_GpUnloadPath

Format:

U32 Acm_GpUnloadPath (HAND GroupHandle, PHAND Path-Handle)

Purpose:

Unload path data.

Parameter:

Name	Type	In or Out	Description
GpHandle	HAND	IN	Group handle from Acm_GpAddAxis.
PathHandle	PHAND	IN	Pointer to path handle from Acm_GpLoadPath.

Return Value:

Error Code.

Comments:

6.3.5.5.5 Acm_GpMovePath

Format:

U32 Acm_GpMovePath (HAND GroupHandle, HAND PathHan-dle)

Purpose:

Start continuous interpolation motion (Path).

Parameters:

Name	Type	In or Out	Description
GpHandle	HAND	IN	Group handle from Acm_GpAddAxis.
PathHandle	PHAND	IN	Pointer to path handle from Acm_GpLoadPath.

Return Value:

Error Code.

Comments:

If the **PathHandle** is returned by Acm_GpLoadPath, the path data will be passed to system path buffer first, then driver start path motion. If the **PathHandle** is NULL, the path data in system path buffer will be executed directly.

6.3.5.5.6 Acm_GpGetPathStatus

Format:

U32 Acm_GpGetPathStatus (HAND GroupHandle, PU32 pCurIndex, PU32 pCurCmdFunc, PU32 pRemainCount, PU32 pFreeSpaceCount)

Purpose:

Get current status of path buffer.

Parameters:

Name	Type	In or Out	Description
GpHandle	HAND	IN	Group handle from Acm_GpAddAxis.
pCurIndex	PU32	OUT	Return current index of path data in path buffer

pCurCmdFunc	PU32	OUT	Return current command function in executing. 0:EndPath 1: Abs2DLine; 2: Rel2DLine; 3: Abs2DArcCW; 4: Abs2DArcCCW; 5: Rel2DArcCW; 6: Rel2DArcCCW; 7: Abs3DLine; 8: Rel3DLine; 9: Abs4DLine;(not support) 10: Rel4DLine; (not support) 11: Abs2DDirect; 12: Rel2DDirect; 13: Abs3DDirect; 14: Rel3DDirect; 15: Abs4DDirect; 16: Rel4DDirect; 17: Abs5DDirect; 18: Rel5DDirect; 19: Abs6DDirect; 20: Rel6DDirect; 21: Abs3DArcCW; (not support) 22: Rel3DArcCW; (not support) 23: Abs3DArcCCW; (not support) 24: Rel3DArcCCW;(not support) 25: Abs3DHelixCW 26: Rel3DHelixCW 27: Abs3DHelixCCW 28: Rel3DHelixCCW 29: GPDELAY (uint:ms)
pRemainCount	PU32	OUT	Return number of unexecuted path data in path.
pFreeSpaceCount	PU32	OUT	Return number of free space in path buffer.

Return Value:

Error Code.

Comments:

You must input the GroupHandle, and then get path status of this group.

6.3.5.5.7 Acm_GpMoveSelPath

Format:

U32 Acm_GpMoveSelPath (HANDGroupHandle, HAND Path-Hanle, U32 StartIndex, U32EndIndex, U8Repeat)

Purpose:

Move path segment in system path buffer from start index and end index.

Parameters:

Name	Type	In or Out	Description
GpHandle	HAND	IN	Group handle from Acm_GpAddAxis.
PathHandle	HAND	IN	Pointer to path handle from Acm_GpLoadPath.
StartIndex	U32	IN	Start Index.(0~9999)
EndIndex	U32	IN	End Index.(0~9999)
Repeat	U8	IN	Repeat count. (0~255)

Return Value:

Error Code.

Comments:

Command to move paths which index is between StartIndex and EndIndex.

If **PathHandle** is null, it will move specified paths in system path buffer; If PathHandle is not null, the paths in PathHandle will be loaded in system path buffer firstly, then move specified paths.

If the value of **Repeat** is zero, the specified paths will be executed continuosly and repeatly until stoping group motion.

If value of **EndIndex** is greater than path count in system path buffer, it will move paths between StartIndex path and last index path in system path buffer.

6.3.5.5.8 Acm_GpGetPathIndexStatus

Format:

Acm_GpGetPathIndexStatus (HAND GroupHandle, U32 Index, PU16 CmdFunc, PU16 MoveMode, PF64 FH, PF64 FL, F64 EndPoint_DataArray, PF64 CenPoint_DataArray, PU32 ArrayElements)

Purpose:

Get the status of specified index path in system path buffer.

Parameters:

Name	Type	In or Out	Description
GpHandle	HAND	IN	Group handle from Acm_GpAddAxis
Index	U32	IN	Index of path.

CmdFunc	PU16	OUT	Return current command function in executing. 0:EndPath 1: Abs2DLine; 2: Rel2DLine; 3: Abs2DArcCW; 4: Abs2DArcCCW; 5: Rel2DArcCW; 6: Rel2DArcCCW; 7: Abs3DLine; 8: Rel3DLine; 9: Abs4DLine;(not support) 10: Rel4DLine; (not support) 11: Abs2DDirect; 12: Rel2DDirect; 13: Abs3DDirect; 14: Rel3DDirect; 15: Abs4DDirect; 16: Rel4DDirect; 17: Abs5DDirect; 18: Rel5DDirect; 19: Abs6DDirect; 20: Rel6DDirect; 21: Abs3DArcCW; (not support) 22: Rel3DArcCW; (not support) 23: Abs3DArcCCW; (not support) 24: Rel3DArcCCW;(not support) 25: Abs3DHelixCW 26: Rel3DHelixCW 27: Abs3DHelixCCW 28: Rel3DHelixCCW 29: GPDELAY (uint:ms)
MoveMode	PU16	OUT	Move mode: 0: No blending 1: Blending
FH	PF64	OUT	Unit: PPU of master Axis(the minimal PhysicalID Axis)
FL	PF64	OUT	Unit: PPU of master Axis(the minimal PhysicalID Axis)

EndPoint_DataArray,	PF64	OUT	Unit: PPU of master Axis(the minimal PhysicalID Axis)
CenPoint_DataArray	PF64	OUT	Unit: PPU of master Axis(the minimal PhysicalID Axis)
ArrayElements	PU32	IN/OUT	Return axis count

Return Value:

Error Code.

Comments:

If you want to know the setting of a path, you can call this API.

6.3.5.6 Pause & Resume

6.3.5.6.1 Acm_GpPauseMotion

Format:

U32 Acm_GpPauseMotion(HAND GroupHandle)

Purpose:

Pause group movement.

Parameters:

Name	Type	In or Out	Description
GroupHandle	HAND	IN	Axis handle from Acm_GpAddAxis

Return Value:

Error Code.

Comments:

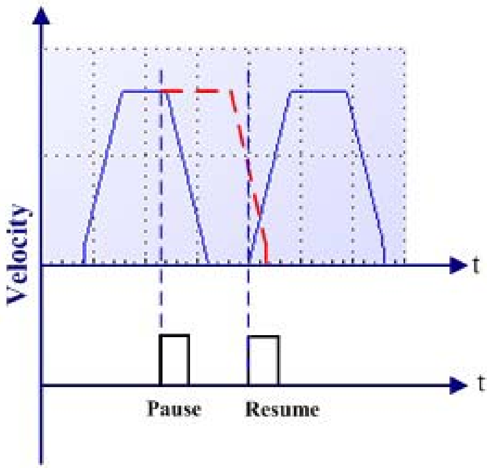
When the Group in the process of movement, its issued pause command, the board after the receipt of the command will decelerate stopped. Its do the restore command, continue not been completed before the pause.

Pause and resume feature supports almost all the action of the Group, including linear interpolation, circular interpolation, helical interpolation and continuous interpolation function.

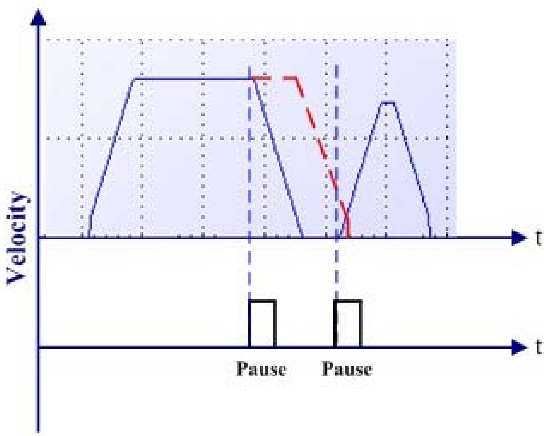
Command group to pause when the group is moving, then group will decelerate to stop after receiving this command. Group will resume the action not completed before pause command. Pause and Resume can support almost all function of Group, including linear interpolation, arc interpolation, helix interpolation and continuous interpolation.

Pause and Resume have some influence on velocity curve and can be classified below:(take T-curve as an example)

1.If command pause in single interpolation movement, the velocity curve is:

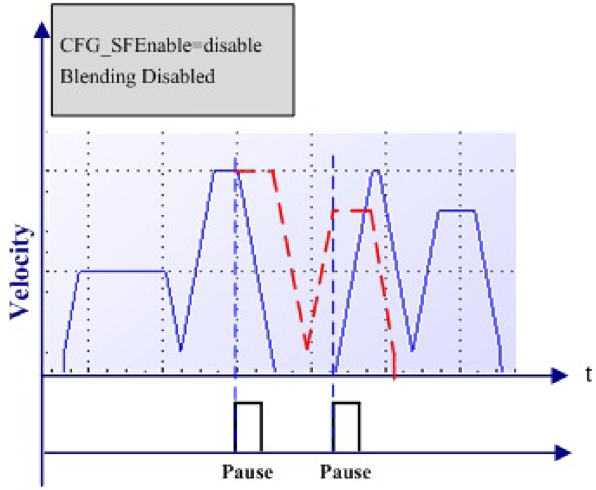


Hardware will calculate if the rest of pulses can support to accelerate to original velocity. If not, the velocity curve is shown as below:



2.There are different situation for continuous interpolation in different mode.

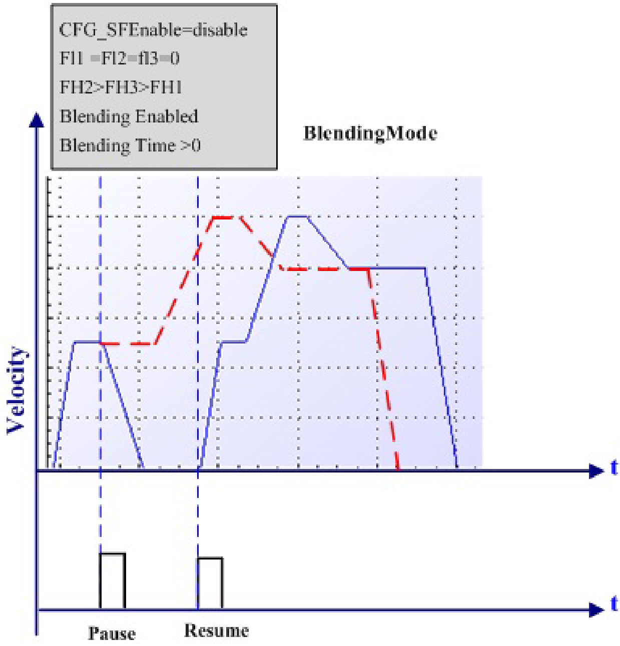
•Buffer Mode: Non-Blending



In BufferMode, each segment has its own accelerated and decelerated period. The current segment will decelerate to stop after receiving Pause command. Hardware will calculate velocity the rest of pulses can support to proceed the rest path.

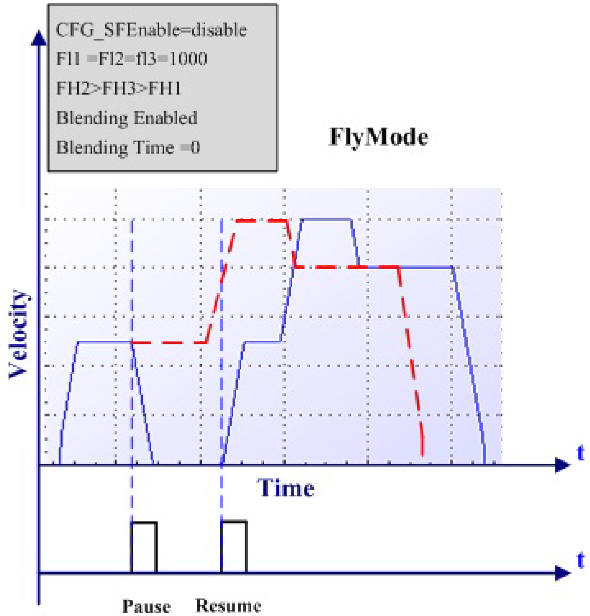
•BlendingMode: Blending Enable?BlendingTime>0?In Blending Mode, there are two situations: the velocity curve is different from FL=0 and FL>0 in Acm_GpAddPath, Please refer to the detail in Acm_GpAddPath.

Pause and resume function will be explained by taking Disable speed forward function (FL=0) as an example below:



In Blending Mode, Pause command will execute after Blending finished if Pause command is issued in the Blending segment.

•FlyMode: Blending Enable?BlendingTime =0



6.3.5.6.2 Acm_GpResumeMotion

Format:

U32 Acm_GpResumeMotion(HAND GroupHandle)

Purpose:

Resume movement after pause

Parameters:

Name	Type	In or Out	Description
GroupHandle	HAND	IN	Axis handle from Acm_GpAddAxis

Return Value:

Error Code.

Comments:

Please refer to the detail in Acm_GpPauseMotion.

6.4 Property List

6.4.1 Device

6.4.1.1 Feature

6.4.1.1.1 FT_DevIpoTypeMap

Data Type:

U32

R/W:

R

PropertyID:

0

Meaning:

Get device supported interpolation types.1: support, 0: Not support

Bits	Description
0	Line interpolation, 2 axes
1	Line interpolation, 3 axes
2	Line interpolation, 4 axes
3	Line interpolation, 5 axes
4	Line interpolation, 6 axes
5~7	Not defined.
8	Arc interpolation, 2 axes
9	Arc interpolation, 3 axes
10	Spiral.
11~15	Not defined.
16	Synchronous electronic gear
17	Synchronous electronic cam
18	Synchronous gantry
19	Synchronous tangent
20~23	Not defined.
24	Select path.
25~31	Not defined.

Comments:

6.4.1.1.2 FT_DevAxisCount

Data Type:

U32

R/W:

R

PropertyID:

1

Meaning:

Get axis number of this device.

Comments:

6.4.1.1.3 FT_DevFunctionMap

Data Type:

U32

R/W:

R

PropertyID:

2

Meaning:

Get device supported functions.1: support, 0: Not support.

Bits	Description
0	Motion
1	DI
2	DO
3	AI (1245/1245V/1245E not Suport)
4	AO (1245/1245V/1245E not Suport)
5	Timer
6	Counter
7	DAQ DI (1245/1245V/1245E do not support)
8	DAQ DO (1245/1245V/1245E do not support)
9	DAQ AI(1245/1245V/1245E not support)
10	DAQ AO (1245/1245V/1245E not support)
11	Emg

Comments:

6.4.1.1.4 FT_DevMDAQTypeMap

Data Type:

U32

R/W:

R

PropertyID:

6

Meaning:

Data types that MotionDAQ supports.

Bits	Description
0	Command Position.
1	Actual Position.
2	Lag Position (difference between Command Position and Actual Position).
3	Command Velocity.

Comments:

6.4.1.1.5 FT_DevMDAQTrigMap

Data Type:

U32

R/W:

R

PropertyID:

7

Meaning:

The methods to trigger MotionDAQ function:

Bits	Description
0	Disable MotionDAQ function.
1	Software command trigger mode (i.e. Start command is issued to trigger).
2	DI trigger.
3	Specify axis motion to start, i.e. trigger MotionDAQ function.

Comments:

6.4.1.1.6 FT_DevMDAQMaxChan

Data Type:

U32

R/W:

R

PropertyID:

8

Meaning:

Record max channel count of MotionDAQ data.

Comments:

In PCI1245/PCI1245V/PCI1245E/1265, the max channel count is 4.

6.4.1.1.7 FT_DevMDAQMaxBufCount

Data Type:

U32

R/W:

R

PropertyID:

9

Meaning:

The max data count of MotionDAQ that each MotionDAQ channal can record.

Comments:

In PCI1245/PCI1245V/PCI1245E/1265, each MotionDAQ channel can record 2000 pieces of data at most.

6.4.1.1.8 FT_DevOverflowCntr

Data Type:

U32

R/W:

R

PropertyID:

3

Meaning:

The maximum data count of position counter.

Comments:

For PCI-1265/PCI-1245/PCI-1245E, the maximum data count is 2147483647.

6.4.1.2 Configuration

6.4.1.2.1 CFG_DevBoardID

Data Type:

U32

R/W:

R

PropertyID:

201

Meaning:

Get Device ID. For PCI-1245/1245V/1245E/1265, this property value will be 0~15.

Comments:

6.4.1.2.2 CFG_DevBaseAddress

Data Type:

U32

R/W:

R

PropertyID:

203

Meaning:

Return IO base address.

Comments:

6.4.1.2.3 CFG_DevInterrupt

Data Type:

U32

R/W:

R

PropertyID:

204

Meaning:

Get Device interrupt number.

Comments:

6.4.1.2.4 CFG_DevBusNumber

Data Type:

U32

R/W:

R

PropertyID:

205

Meaning:

Get device bus number.

Comments:

6.4.1.2.5 CFG_DevSlotNumber

Data Type:

U32

R/W:

R

PropertyID:

206

Meaning:

Get device slot number.

Comments:

6.4.1.2.6 CFG_DevDriverVersion

Data Type:

char*

R/W:

R

PropertyID:

207

Meaning:

Get SYS driver's version. The format is: 1.0.0.1

Comments:

6.4.1.2.7 CFG_DevDllVersion

Data Type:

char*

R/W:

R

PropertyID:

208

Meaning:

Get DLL driver's version. The format is: 1.0.0.1.

Comments:

6.4.1.2.8 CFG_DevFwVersion

Data Type:

char*

R/W:

R

PropertyID:

208

Meaning:

Get the firm version, the format is: 1.0.0.1.

Comments:

6.4.1.2.9 CFG_DevCPLDVersion

Data Type:

char*

R/W:

R

PropertyID:

218

Meaning:

Get the CPLD version of device, the format is: 1.0.0.1.

Comments:

6.4.1.2.10 CFG_DevEmgLogic

Data Type:

U32

R/W:

RW

PropertyID:

220

Meaning:

Set the active logic for emergency stop signal.

Comments:

Bits	Description
0	Low active
1	High active

6.4.2 DAQ

6.4.2.1 Feature

6.4.2.1.1 FT_DaqDiMaxChan

Data Type:

U32

R/W:

R

PropertyID:

50

Meaning:

Get maximum number of DI channels.

Comments:

In PCI-1265, the value is 8.

6.4.2.1.2 FT_DaqDoMaxChan

Data Type:

U32

R/W:

R

PropertyID:

51

Meaning:

Get maximum number of DO channels.

Comments:

In PCI-1265, the value is 8.

6.4.2.1.3 FT_DaqAiRangeMap

Data Type:

U32

R/W:

R

PropertyID:

52

Meaning:

Get the supported AI range. 1: support, 0: not support.

Bits	Description
0	+/- 10V
1	+/- 5V
2	+/- 2.5V
3	+/- 1.25V
4	+/- 0.625V
5~15	Not defined.
16	0~20mA
17~31	Not defined.

Comments:

PCI-1265 only supports +/- 10V.

6.4.2.1.4 FT_DaqAiMaxSingleChan

Data Type:

U32

R/W:

R

PropertyID:

54

Meaning:

Get the Maximum Single-Ended AI channel number of the device.

Comments:

In PCI-1265, the value is 2.

6.4.2.1.5 FT_DaqAiMaxDiffChan

Data Type:

U32

R/W:

R

PropertyID:

55

Meaning:

Get the Maximum Differential AI channel number of the device.

Comments:

In PCI-1265, the value is 1.

6.4.2.1.6 FT_DaqAiResolution

Data Type:

U32

R/W:

R

PropertyID:

56

Meaning:

Get the AI resolution in bit of the device.

Comments:

In PCI-1265, the value is 12.

6.4.2.2 Configuration

6.4.2.2.1 CFG_DaqAiChanType

Data Type:

U32

R/W:

R/W

PropertyID:

251

Meaning:

0: Single ended
1: Differential

Comments:

PCI-1265 has two analog input channels. Users can assign the channel to be single-ended or differential mode. If it is differential mode, the analog input value shall be acquired from channel 1.

6.4.2.2.2 CFG_DaqAiRanges

Data Type:

U32

R/W:

R/W

PropertyID:

252

Meaning:

See FT_DaqAiRangeMap.

Bits	Description
0x0	+/- 10 V
0x1	+/- 5 V
0x2	+/- 2.5 V
0x3	+/- 1.25 V

Comments:

PCI-1265 only supports +/- 10V.

6.4.3 Axis

6.4.3.1 Feature

6.4.3.1.1 System

6.4.3.1.1.1 FT_AxFunctionMap

Data Type:

U32

R/W:

R

PropertyID:

301

Meaning:

Get the axis supported function. 1: support, 0: not support

Bits	Description
0	In position.
1	Alarm
2	Clear the deflection counter in the servo driver.
3	Slow down
4	Hardware limit switch
5	Software limit switch
6	Home sensor
7	Encode Z phase sensor
8	Backlash corrective.
9	Suppress vibration.
10	Home
11	Impose
12	Compare
13	Latch
14	CAMDO
15	Ext-Drive
16	Simultaneous start/stop
17~31	Not defined.

Comments:

6.4.3.1.2 Speed Pattern

6.4.3.1.2.1 FT_AxMaxVel

Data Type:

F64

R/W:

R

PropertyID:

302

Meaning:

Get axis supported max velocity. (Unit: Pulse/s)

Comments:

In PCI-1245/1245V/1245E/1265, the value is 5,000,000.

6.4.3.1.2.2 FT_AxMaxAcc

Data Type:

F64

R/W:

R

PropertyID:

303

Meaning:

Get axis supported max acceleration. (Unit: Pulse/s^2)

Comments:

In PCI-1245/1245V/1245E/1265, the value is 500,000,000.

6.4.3.1.2.3 FT_AxMaxDec

Data Type:

F64

R/W:

R

PropertyID:

304

Meaning:

Get axis supported max deceleration (Unit: Pulse/s^2)

Comments:

In PCI-1245/1245V/1245E/1265, the value is 500,000,000.

6.4.3.1.2.4 FT_AxMaxJerk

Data Type:

F64

R/W:

R

PropertyID:

305

Meaning:

Get axis supported max jerk. (Unit: Pulse/S^3)

Comments:

In PCI-1245/1245V/1245E/1265, the value is 1.

6.4.3.1.3 Pulse In

6.4.3.1.3.1 FT_AxPulseInMap

Data Type:

U32

R/W:

R

PropertyID:

306

Meaning:

Get the pulse input features supported by this motion device.

Bits	Description
0	Mode
1	Logic
2	Source
3~31	Not defined.

Comments:

6.4.3.1.3.2 FT_AxPulseInModeMap

Data Type:

U32

R/W:

R

PropertyID:

307

Meaning:

Get axis supported pulse input mode.

Bits	Description
0	1X A/B
1	2X A/B
2	4X A/B
3	CW/CCW
4~31	Not defined.

Comments:

6.4.3.1.4 Pulse Out

6.4.3.1.4.1 FT_AxPulseOutMap

Data Type:

U32

R/W:

R

PropertyID:

308

Meaning:

Get the pulse output features supported by this motion device.

Bits	Description
0	Mode
1~31	Not defined.

Comments:

In PCI-1245/1245V/1245E/1265, the value is 1.

6.4.3.1.4.2 FT_AxPulseOutModeMap

Data Type:

U32

R/W:

R

PropertyID:

309

Meaning:

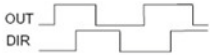
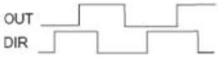
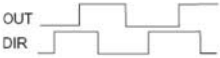
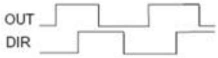
Get pulse output modes supported by this motion device.

Bits	Description
0	OUT/DIR
1	OUT/DIR, OUT negative logic
2	OUT/DIR, DIR negative logic
3	OUT/DIR, OUT&DIR negative logic
4	CW/CCW
5	CW/CCW, CW&CCW negative logic

6	A/B Phase
7	B/A Phase
8	CW/CCW, OUT negative logic.(Not support)
9	CW/CCW, DIR negative logic.(Not support)
10~31	Not defined.

Comments:

In PCI-1245/1245V/1245E/1265 , the value is 63.

Bits	Description	Positive direction		Negative direction	
		OUT output	**DIR output**	**OUT output**	**DIR output**
0	OUT/DIR		High		Low
1	OUT/DIR, OUT negative logic		High		Low
2	OUT/DIR, DIR negative logic		Low		High
3	OUT/DIR, OUT&DIR negative logic		Low		High
4	CW/CCW		High	High	
5	CW/CCW, CW&CCW negative logic		Low	Low	
6	A/B Phase				
7	B/A Phase				

6.4.3.1.5 Alarm

6.4.3.1.5.1 FT_AxAlmMap

Data Type:

U32

R/W:

R

PropertyID:

310

Meaning:

Get the alarm features supported by this motion axis.

Bits	Description
0	Enabled
1	Logic
2	React
3~31	Not defined.

Comments:

6.4.3.1.6 In Position

6.4.3.1.6.1 FT_AxInpMap

Data Type:

U32

R/W:

R

PropertyID:

311

Meaning:

Get the In-Position features supported by this motion axis.

Bits	Description
0	Mode
1	Logic
2~31	Not defined.

Comments:

6.4.3.1.7 ERC

6.4.3.1.7.1 FT_AxErcMap

Data Type:

U32

R/W:

R

PropertyID:

312

Meaning:

Get the ERC features supported by this motion axis.

Bits	Description
0	Enable mode
1	Logic
2	On time(not support)
3	Off time(not support)
4~31	Not defined.

Comments:

6.4.3.1.7.2 FT_AxErcEnableModeMap

Data Type:

U32

R/W:

R

PropertyID:

313

Meaning:

Get axis supported ERC mode.

Bits	Description
0	ERC Output when home finish
1	ERC Output when EMG/ALM/EL active
2	ERC Output when home finish or EMG/ALM/EL active
3~31	Not defined.

Comments:

6.4.3.1.8 SD

6.4.3.1.8.1 FT_AxSdMap

Data Type:

U32

R/W:

R

PropertyID:

316

Meaning:

Get the Slow-Down (SD) features supported by this motion axis.

Bits	Description
0	Enabled
1	Logic
2	React
3~31	Not defined.

Comments:

In this PCI-1245/1245V/1245E/1265, the value is 0.

6.4.3.1.9 Hardware Limit

6.4.3.1.9.1 FT_AxElMap

Data Type:

U32

R/W:

R

PropertyID:

317

Meaning:

Get the hardware end limit (EL) features supported by this motion axis.

Bits	Description
0	Enabled
1	Logic
2	React
3~31	Not defined.

Comments:

6.4.3.1.10 Software Limit

6.4.3.1.10.1 FT_AxSwMelMap

Data Type:

U32

R/W:

R

PropertyID:

318

Meaning:

Get the software minus limit features supported by the motion axis.

Bits	Description
0	Enabled
1	React
2	Value
3~31	Not defined.

Comments:

6.4.3.1.10.2 FT_AxSwPelMap

Data Type:

U32

R/W:

R

PropertyID:

319

Meaning:

Get the software plus limit features supported by the motion axis.

Bits	Description
0	Enabled
1	React
2	Value
3~31	Not defined.

Comments:

6.4.3.1.11 Home

6.4.3.1.11.1 FT_AxHomeMap

Data Type:

U32

R/W:

R

PropertyID:

320

Meaning:

Get the home features supported by this motion axis.

Bits	Description
0	Home mode
1	ORG logic
2	EZ logic
3	Reset Enable
4~31	Not defined.

Comments:

6.4.3.1.11.2 FT_AxHomeModeMap

Data Type:

U32

R/W:

R

PropertyID:

332

Meaning:

The supported Home return modes.

Bits	Description
0	MP_MODE1_Abs
1	MP_MODE2_Lmt
2	MP_MODE3_Ref
3	MP_MODE4_Abs_Ref
4	MP_MODE5_Abs_NegRef
5	MP_MODE6_Lmt_Ref
6	MP_MODE7_AbsSearch
7	MP_MODE8_LmtSearch
8	MP_MODE9_AbsSearch_Ref
9	MP_MODE10_AbsSearch_NegRef
10	MP_MODE11_LmtSearch_Ref
11	MP_MODE12_AbsSearchReFind
12	MP_MODE13_LmtSearchReFind
13	MP_MODE14_AbsSearchReFind_Ref
14	MP_MODE15_AbsSearchReFind_NegRef
15	MP_MODE16_LmtSearchReFind_Ref

Comments:

About detailed information about each mode, see about Acm_AxHome.

6.4.3.1.12 Backlash

6.4.3.1.12.1 FT_AxBackLashMap

Data Type:

U32

R/W:

R

PropertyID:

321

Meaning:

Get the backlash feature supported by this motion axis.

Bits	Description
0	Enabled
1	Value
2~31	Not defined.

Comments:

6.4.3.1.13 Compare

6.4.3.1.13.1 FT_AxCompareMap

Data Type:

U32

R/W:

R

PropertyID:

324

Meaning:

Get axis supported compare features.

Bits	Description
0	Enabled
1	Logic
2	CmpSrc
3	CmpMethod
4	CmpPulseMode
5	CmpPulseWidth
6~31	Not defined.

Comments:

6.4.3.1.14 Latch

6.4.3.1.14.1 FT_AxLatchMap

Data Type:

U32

R/W:

R

PropertyID:

325

Meaning:

Get axis supported latch features.

Bits	Description
0	Enabled
1	Logic
2~31	Not defined.

Comments:

6.4.3.1.15 Cam DO

6.4.3.1.15.1 FT_AxCamDOMap

Data Type:

U32

R/W:

R

PropertyID:

326

Meaning:

Get axis supported CamDO features.

Bits	Description
0	Enabled
1	CamDOLogic
2	CamDOCmpSrc
3	CamDOLoLimit
4	CamDOHiLimit
5	CamDOMode
6	CamDODir
7~31	Not defined.

Comments:

6.4.3.1.16 Ext-Drive

6.4.3.1.16.1 FT_AxExtDriveMap

Data Type:

U32

R/W:

R

PropertyID:

327

Meaning:

Get axis supported external drive features.

Bits	Description
0	ExtMasterSrc
1	ExtSelEnable
2	ExtPulseNum
3	ExtPulseMode
4	ExtPresetNum
5~31	Not defined.

Comments:

6.4.3.1.16.2 FT_AxExtMasterSrcMap

Data Type:

U32

R/W:

R

PropertyID:

328

Meaning:

Get axis supported external drive master source.

Bits	Description
0	axis 0
1	axis 1
2	axis 2
3	axis 3
4~31	Not defined.

Comments:

6.4.3.1.17 Aux/Gen DIO

6.4.3.1.17.1 FT_AxGenDOMap

Data Type:

U32

R/W:

R

PropertyID:

329

Meaning:

Get axis supported general output from OUT4 to OUT7.

Bits	Description
0	OUT4/CAM_DO
1	OUT5/TRIG_Position
2	OUT6/SVON
3	OUT7/ERC
4~31	Not defined.

Comments:

6.4.3.1.17.2 FT_AxGenDIMap

Data Type:

U32

R/W:

R

PropertyID:

330

Meaning:

Get axis supported general input from IN1 to IN5

Bits	Description
0	IN1/LTC
1	IN2/RDY
2	IN4/JOG+
3	IN5/JOG-
4~31	Not defined.

Comments:

6.4.3.1.18 Simultaneity

6.4.3.1.18.1 FT_AxSimStartSourceMap

Data Type:

U32

R/W:

R

PropertyID:

331

Meaning:

The Mode of simultaneous starting that axis supported.

Bits	Description
0	Start Simultaneous Starting on signal from STA Pin on device. (Default)
1~7	Not defined.
8	Start Simultaneous Starting with axis_0's compare signal.
9	Start Simultaneous Starting with axis_1's compare signal
10	Start Simultaneous Starting with axis_2's compare signal
11	Start Simultaneous Starting with axis_3's compare signal
12	Start Simultaneous Starting with axis_4's compare signal
13	Start Simultaneous Starting with axis_5's compare signal
14~15	Not defined.
16	Start Simultaneous Starting when axis_0 is stopped.
17	Start Simultaneous Starting when axis_1 is stopped.
18	Start Simultaneous Starting when axis_2 is stopped.
19	Start Simultaneous Starting when axis_3 is stopped.
20	Start Simultaneous Starting when axis_4 is stopped.
21	Start Simultaneous Starting when axis_5 is stopped.
22~31	Not defined.

Comments:

Get axis supported simultaneous starting mode. See about CFG_AxSimStartSource. In PCI-1265,the default value:4144897. In PCI1245/1245V/1245E, the default value:986881.

6.4.3.1.19 Trigger Stop

6.4.3.1.19.1 FT_AxIN1Map

Data Type:

U32

R/W:

R

PropertyID:

333

Meaning:

IN1 trigger stop function property.

Comments:

Bits	Description
0	Enable/Disable stop function
1	Stop logic: high stop or low stop
2	Stop mode: decelerating/sudden stop

6.4.3.1.19.2 FT_AxIN2Map

Data Type:

U32

R/W:

R

PropertyID:

334

Meaning:

IN2 trigger stop function property.

Comments:

Bits	Description
0	Enable/Disable stop function
1	Stop logic: high stop or low stop
2	Stop mode: decelerating/sudden stop

6.4.3.1.19.3 FT_AxIN4Map

Data Type:

U32

R/W:

R

PropertyID:

336

Meaning:

IN4 trigger stop function property.

Comments:

Bits	Description
0	Enable/Disable stop function
1	Stop logic: high stop or low stop
2	Stop mode: decelerating/sudden stop

6.4.3.1.19.4 FT_AxIN5Map

Data Type:

U32

R/W:

R

PropertyID:

337

Meaning:

IN5 trigger stop function property.

Comments:

Bits	Description
0	Enable/Disable stop function
1	Stop logic: high stop or low stop
2	Stop mode: decelerating/sudden stop

6.4.3.2 Config

6.4.3.2.1 System

6.4.3.2.1.1 CFG_AxPPU

Data Type:

U32

R/W:

RW

PropertyID:

551

Meaning:

Pulse per unit (PPU), a virtual unit.

This property value must be greater than 0.

This property value's change will affect CFG_AxMaxVel, CFG_AxMaxAcc, CFG_AxMaxDec, PAR_AxVelHigh, PAR_AxVelLow, PAR_AxAcc, PAR_AxDec, PAR_GpVelHigh, PAR_GpVelLow, PAR_GpAcc, PAR_GpDec,PAR_HomeCrossDistance.

Comments:

The default value is 1.

6.4.3.2.1.2 CFG_AxPhyID

Data Type:

U32

R/W:

R

PropertyID:

552

Meaning:

Get physical ID of the axis.

Value	Meaning
0	0-axis
1	1-axis
2	2-axis
3	3-axis
4	4-axis
5	5-axis

Comments:

6.4.3.2.2 Speed Pattern

6.4.3.2.2.1 CFG_AxMaxVel

Data Type:

F64

R/W:

RW

PropertyID:

553

Meaning:

Configure the max velocity for the motion axis (Unit: PPU/s).

Comments:

This property's max value= FT_AxMaxVel / CFG_AxPPU and min value = 1/ CFG_AxPPU.

In PCI-1245/1245V/1245E/1265, the default value is 5,000,000.

6.4.3.2.2.2 CFG_AxMaxAcc

Data Type:

F64

R/W:

RW

PropertyID:

554

Meaning:

Configure the max acceleration for the motion axis (Unit: PPU/S^2).

Comments:

This property's max value= FT_AxMaxAcc / CFG_AxPPU and min value = 1/ CFG_AxPPU.

In PCI-1245/1245V/1245E/1265, the default value is 500,000,000.

6.4.3.2.2.3 CFG_AxMaxDec

Data Type:

F64

R/W:

RW

PropertyID:

555

Meaning:

Configure the max deceleration for the motion axis (Unit: PPU/S^2).

Comments:

This property's max value= FT_AxMaxDec / CFG_AxPPU and min value = 1/ CFG_AxPPU.

In PCI-1245/1245V/1245E/1265, the default value is 500,000,000.

6.4.3.2.2.4 CFG_AxMaxJerk

Data Type:

F64

R/W:

R

PropertyID:

556

Meaning:

Get max jerk configuration for the motion axis.

Comments:

In PCI-1245/1245V/1245E/1265, the value is 1.

6.4.3.2.3 Pulse In

6.4.3.2.3.1 CFG_AxPulseInMode

Data Type:

U32

R/W:

RW

PropertyID:

557

Meaning:

Set/get encoder feedback pulse input mode.

Value	Description
0	1XAB
1	2XAB
2	4XAB
3	CCW/CW

Comments:

6.4.3.2.3.2 CFG_AxPulseInLogic

Data Type:

U32

R/W:

RW

PropertyID:

558

Meaning:

Set /get logic of encoder feedback pulse.

Value	Description
0	Not inverse direction
1	Inverse direction

Comments:

6.4.3.2.3.3 CFG_AxPulseInMaxFreq

Data Type:

U32

R/W:

RW

PropertyID:

632

Meaning:

Set /get encode max pulse in frequency.

Value	Description
0	500KHz
1	1MHz
2	2MHz
3	4MHz

Comments:

6.4.3.2.4 Pulse Out

6.4.3.2.4.1 CFG_AxPulseOutMode

Data Type:

U32

R/W:

RW

PropertyID:

560

Meaning:

Set/get command pulse output mode.

Value	Description
1	OUT/DIR
2	OUT/DIR, OUT negative logic
4	OUT/DIR, DIR negative logic
8	OUT/DIR, OUT&DIR negative logic
16	CW/CCW
32	CW/CCW, CW&CCW negative logic
256	CW/CCW, OUT negative logic.
512	CW/CCW, DIR negative logic.

Comments:

In PCI-1245/1245V/1245E/1265, the default value is 16.

See also FT_AxPulseOutMode.

6.4.3.2.5 Alarm

6.4.3.2.5.1 CFG_AxAlmLogic

Data Type:

U32

R/W:

RW

PropertyID:

562

Meaning:

Set/get active logic of alarm signal.

Value	Description
0	Low active
1	High active

Comments:

In PCI-1245/1245V/1245E/1265, the default value is 1.

6.4.3.2.5.2 CFG_AxAlmEnable

Data Type:

U32

R/W:

RW

PropertyID:

561

Meaning:

Enable/disable motion alarm function. Alarm is a signal generated by motor drive when motor drive is in alarm status.

Value	Description
0	Disabled
1	Enabled

Comments:

In PCI-1245/1245V/1245E/1265, the default value is 0.

Please modify "CFG_AxAlmReact" and "CFG_AxAlmLogic" before modifying the value of "CFG_AxAlmEnable".

6.4.3.2.5.3 CFG_AxAlmReact

Data Type:

U32

R/W:

RW

PropertyID:

563

Meaning:

Set/get the stop modes when receiving ALARM signal.

Value	Description
0	Motor immediately stops
1	Motor decelerates then stops

Comments:

In PCI-1245/1245V/1245E/1265, the default value is 1.

6.4.3.2.6 In Position

6.4.3.2.6.1 CFG_AxInpEnable

Data Type:

U32

R/W:

RW

PropertyID:

564

Meaning:

Enable/disable In-Position function.

Value	Description
0	Disabled
1	Enabled

Comments:

In PCI-1245/1245V/1245E/1265, the default value is 0.

6.4.3.2.6.2 CFG_AxInpLogic

Data Type:

U32

R/W:

RW

PropertyID:

565

Meaning:

Set/get the active logic for In-Position signal.

Value	Description
0	Low active
1	High active

Comments:

In PCI-1245/1245V/1245E/1265, the default value is 1.

6.4.3.2.7 ERC

6.4.3.2.7.1 CFG_AxErcLogic

Data Type:

U32

R/W:

RW

PropertyID:

566

Meaning:

Set/get active logic for ERC signal

Value	Description
0	Low active
1	High active

Comments:

In PCI-1245/1245V/1245E/1265, the default value is 1.

6.4.3.2.7.2 CFG_AxErcEnableMode

Data Type:

U32

R/W:

RW

PropertyID:

569

Meaning:

Set/get ERC out mode or diable ERC function.

Value	Description

0	Disabled
1	ERC Output when home finish
2	ERC Output when EMG/ALM/EL active(Not support)
3	ERC Output when home finish or EMG/ALM/EL active(Not support)

Comments:

In PCI-1245/1245V/1245E/1265, the default value is 0.

6.4.3.2.8 Hardware Limit

6.4.3.2.8.1 CFG_AxElReact

Data Type:

U32

R/W:

RW

PropertyID:

576

Meaning:

Set/get the reacting mode of EL signal.

Value	Description
0	Motor immediately stops
1	Motor decelerates then stops

Comments:

In PCI-1245/1245V/1245E/1265, the default value is 0.

6.4.3.2.8.2 CFG_AxElLogic

Data Type:

U32

R/W:

RW

PropertyID:

575

Meaning:

Set/get active logic for hardware limit signal.

Value	Description
0	Low active
1	High active

Comments:

In PCI-1245/1245V/1245E/1265, the default value is 0.

6.4.3.2.8.3 CFG_AxElEnable

Data Type:

U32

R/W:

RW

PropertyID:

574

Meaning:

Set/ get hardware limit function enable/disable.

Value	Description
0	Disabled
1	Enabled

Comments:

In PCI-1245/1245V/1245E/1265, the default value is 1.

Please modify "CFG_AxElReact" and "CFG_AxElLogic" before modifying the value of "CFG_AxElEnable".

6.4.3.2.9 Software Limit

6.4.3.2.9.1 CFG_AxSwMelEnable

Data Type:

U32

R/W:

RW

PropertyID:

577

Meaning:

Enable/Disable the minus software limit function.

Value	Description
0	Disabled
1	Enabled

Comments:

6.4.3.2.9.2 CFG_AxSwPelEnable

Data Type:

U32

R/W:

RW

PropertyID:

578

Meaning:

Enable/Disable the plus software limit.

Value	Description
0	Disabled
1	Enabled

Comments:

6.4.3.2.9.3 CFG_AxSwMelReact

Data Type:

U32

R/W:

RW

PropertyID:

579

Meaning:

Set/get the reacting mode of minus software limit.

Value	Description
0	Motor immediately stops
1	Motor decelerates then stops

Comments:

In PCI-1245/1245V/1245E/1265, the default value is 1.

6.4.3.2.9.4 CFG_AxSwPelReact

Data Type:

U32

R/W:

RW

PropertyID:

580

Meaning:

Set/get the reacting mode of plus software limit.

Value	Description
0	Motor immediately stops
1	Motor decelerates then stops

Comments:

In PCI-1245/1245V/1245E/1265, the default value is 1.

6.4.3.2.9.5 CFG_AxSwMelValue

Data Type:

I32

R/W:

RW

PropertyID:

581

Meaning:

Set/get the value of minus software limit. The property value's range is: -2,147,483,647 ~ +2,147,483,647.

Comments:

6.4.3.2.9.6 CFG_AxSwPelValue

Data Type:

I32

R/W:

RW

PropertyID:

582

Meaning:

Set/get the value of plus software limit. The property value's range is: -2,147,483,647 ~ +2,147,483,647.

Comments:

6.4.3.2.10 Home

6.4.3.2.10.1 CFG_AxOrgLogic

Data Type:

U32

R/W:

RW

PropertyID:

589

Meaning:

Set/get the active logic for ORG signal.

Value	Description
0	Low active
1	High active

Comments:

In PCI-1245/1245V/1245E/1265, the default value is 0.

6.4.3.2.10.2 CFG_AxEzLogic

Data Type:

U32

R/W:

RW

PropertyID:

591

Meaning:

Set/get the active logic for EZ signal.

Value	Description
0	Low active
1	High active

Comments:

In PCI-1245/1245V/1245E/1265, the default value is 0.

6.4.3.2.10.3 CFG_AxHomeResetEnable

Data Type:

U32

R/W:

RW

PropertyID:

602

Meaning:

Enable or Disable logical counter reset function after finish Home.

Value	Description
0	Disabled
1	Enabled

Comments:

6.4.3.2.10.4 CFG_AxOrgReact

Data Type:

U32

R/W:

RW

PropertyID:

634

Meaning:

Set the ending reaction mode after finishing Home.

Value	Description
0	Stop immediately.

1	Decelerate and stop.

Comments:

6.4.3.2.11 Backlash

6.4.3.2.11.1 CFG_AxBacklashEnable

Data Type:

U32

R/W:

RW

PropertyID:

593

Meaning:

Enable/Disable corrective backlash.

Value	Description
0	Disabled
1	Enabled

Comments:

In PCI-1245/1245V/1245E/1265, the default value is 0.

6.4.3.2.11.2 CFG_AxBacklashPulses

Data Type:

U32

R/W:

RW

PropertyID:

594

Meaning:

Set/get the compensation pulse numbers. (Uint: pulse)

Comments:

This value should be between 0 and 4095.Whenever direction change occurs, the axis outputs backlash corrective pulses before sending commands.

In PCI-1245/1245V/1245E/1265, the default value is 10.

6.4.3.2.11.3 CFG_AxBacklashVel

Data Type:

U32

R/W:

RW

PropertyID:

630

Meaning:

Set /get the velocity of corrective backlash. (Uint: pulse/s)

Comments:

In PCI-1245/1245V/1245E/1265, the default value is 1000.

6.4.3.2.12 Compare

6.4.3.2.12.1 CFG_AxCmpSrc

Data Type:

U32

R/W:

RW

PropertyID:

603

Meaning:

Get/set compare source.

Value	Description
0	Command Position
1	Actual position

Comments:

In PCI-1245/1245V/1245E/1265, the default value is 0.

6.4.3.2.12.2 CFG_AxCmpMethod

Data Type:

U32

R/W:

RW

PropertyID:

604

Meaning:

Set or get compare method.

Value	Description
0	>= Position Counter
1	<= Position Counter
2	=Counter (Directionless)(Not support)

Comments:

In PCI-1245/1245V/1245E/1265, the default value is 0.

6.4.3.2.12.3 CFG_AxCmpPulseLogic

Data Type:

U32

R/W:

RW

PropertyID:

606

Meaning:

Set /get the active logical of compare signal.

Value	Description
0	Low active
1	High active

Comments:

In PCI-1245/1245V/1245E/1265, the default value is 1.

6.4.3.2.12.4 CFG_AxCmpPulseWidth

Data Type:

U32

R/W:

RW

PropertyID:

607

Meaning:

Get/set the width of compare signal delay.

Value	Description
0	5 ms
1	10 ms
2	20 ms
3	50 ms
4	100 ms
5	200 ms
6	500 ms
7	1000 ms

Comments:

6.4.3.2.12.5 CFG_AxCmpEnable

Data Type:

U32

R/W:

RW

PropertyID:

608

Meaning:

Enable/disable axis compare function.

Value	Description
0	Disabled
1	Enabled

Comments:

6.4.3.2.12.6 CFG_AxCmpPulseMode

Data Type:

U32

R/W:

RW

PropertyID:

605

Meaning:

Set/get axis compare mode.

Value	Description
0	Pulse mode
1	Toggle mode.

Comments:

Toggle mode: inverse DO output when compare signal happens.

6.4.3.2.13 Latch

6.4.3.2.13.1 CFG_AxLatchLogic

Data Type:

U32

R/W:

RW

PropertyID:

601

Meaning:

Set/get the active logic for Latch signal.

Value	Description
0	Low active
1	High active

Comments:

When the Latch is triggered, the command position, actual position and lag distance will be latched.

6.4.3.2.13.2 CFG_AxLatchEnable

Data Type:

U32

R/W:

RW

PropertyID:

631

Meaning:

Enable/disable latch function.

Value	Description
0	Disabled
1	Enabled

Comments:

6.4.3.2.14 Aux/Gen DIO

6.4.3.2.14.1 CFG_AxGenDoEnable

Data Type:

U32

R/W:

RW

PropertyID:

610

Meaning:

Enable/disabe the axis general DO function.

Value	Description
0	Disabled
1	Enabled

Comments:

If property **CFG_AxGenDoEnable** is enabled, the property CFG_AxCmpEnable, CFG_AxCamDoEnable and CFG_AxErcEnableMode is disabled automatically. Functions Acm_AxSetCmpData, Acm_AxSetCmpTable, Acm_AxSetCmpAuto will not be able to output signal.

6.4.3.2.15 Ext-Drive

6.4.3.2.15.1 CFG_AxExtMasterSrc

Data Type:

U32

R/W:

RW

PropertyID:

611

Meaning:

Set/get input pin for external drive.

Value	Description
0	axis 0
1	axis 1 (Not support)
2	axis 2 (Not support)
3	axis 3 (Not support)

Comments:

In PCI-1265/PCI-1245/PCI-1245V/PCI-1245E, only support 0.

6.4.3.2.15.2 CFG_AxExtSelEnable

Data Type:

U32

R/W:

RW

PropertyID:

612

Meaning:

When Ext. drive is enable. This property enables driving axis selection by digital input channel.

Value	Description
0	Disabled
1	Enabled(not support)

Comments:

In PCI-1265/PCI-1245/PCI-1245V/PCI-1245E, only support 0.

6.4.3.2.15.3 CFG_AxExtPulseNum

Data Type:

U32

R/W:

RW

PropertyID:

613

Meaning:

Set command pulse number when axis' external drive mode is MPG and the A/B or B/A phase signal is triggered.

Comments:

In this PCI-1245/1245V/1245E/1265, the default value is 1. This value must be larger than zero.

6.4.3.2.15.4 CFG_AxExtPulseInMode

Data Type:

U32

R/W:

RW

PropertyID:

617

Meaning:

Set/get external drive pulse input mode.

Value	Description
0	1XAB
1	2XAB
2	4XAB
3	CCW/CW

Comments:

6.4.3.2.15.5 CFG_AxExtPresetNum

Data Type:

U32

R/W:

RW

PropertyID:

618

Meaning:

Set/get pulse number of external drive when an active edge of input pulse is accept in JOG mode.

Comments:

In PCI-1245/1245E/1265, the default value is 1.This value must lager than zero.

6.4.3.2.16 Cam Do

6.4.3.2.16.1 CFG_AxCamDOEnable

Data Type:

U32

R/W:

RW

PropertyID:

622

Meaning:

Set/get cam DO enable/diable.

Value	Description
0	Disabled
1	Enabled

Comments:

CamDO and OUT4 use the same pin, if the CFG_AxGenDoEnable is enabled, OUT4 isn't able to output CamDO signal.

6.4.3.2.16.2 CFG_AxCamDOLoLimit

Data Type:

U32

R/W:

RW

PropertyID:

623

Meaning:

Set/get the low limit for CAMDO signal.

Comments:

If CamDo is enabled, when command/actual position is between the low limit value and high limit value, the CamDo signal will triggered.

In PCI-1245/1245E/1265, the default value is 0.

Range: -2147483647~2147483647.

6.4.3.2.16.3 CFG_AxCamDOHiLimit

Data Type:

U32

R/W:

RW

PropertyID:

624

Meaning:

Set/get the high limit for CamDo signal.

Comments:

If CamDo is enabled, when command/actual position is between the low limit value and high limit value, the CamDo signal will triggered.

In PCI-1245/1245V/1245E/1265, the default value is 20000.

Range: -2147483647~2147483647

6.4.3.2.16.4 CFG_AxCamDOCmpSrc

Data Type:

U32

R/W:

RW

PropertyID:

627

Meaning:

Set/get the compare source.

Value	Description
0	Command position.
1	Feedback position

Comments:

In PCI-1245/1245V/1245E/1265, the default value is 0.

6.4.3.2.16.5 CFG_AxCamDOLogic

Data Type:

U32

R/W:

RW

PropertyID:

628

Meaning:

Set/get the active logic of CamDO.

Value	Description
0	Low active
1	High active

Comments:

In PCI-1245/1245V/1245E/1265, the default value is 1.

6.4.3.2.17 Module

6.4.3.2.17.1 CFG_AxModuleRange

Data Type:

U32

R/W:

RW

PropertyID:

629

Meaning:

Set/get pulse number when axis moves 360 degree.

Comments:

This value must be multiple of 4.

It is used in tangent motion and E-cam motion. See about Acm_AxTangentInGp, Acm_DevDownLoadCAMTable and Acm_AxCamInAx.

In PCI-1245/1245V/1245E/1265, the default value is 0.

Range: 0 ~ 8,000,000

6.4.3.2.18 Simultaneity

6.4.3.2.18.1 CFG_AxSimStartSource

Data Type:

U32

R/W:

RW

PropertyID:

633

Meaning:

Set/get simultaneous starting mode for current axis.

Value	Description
0	Disabled
1	Start Simultaneous Starting on signal from STA Pin on device. (Default)
256	Start Simultaneous Starting with axis_0's compare signal.
512	Start Simultaneous Starting with axis_1's compare signal
1024	Start Simultaneous Starting with axis_2's compare signal
2048	Start Simultaneous Starting with axis_3's compare signal
4096	Start Simultaneous Starting with axis_4's compare signal
8192	Start Simultaneous Starting with axis_5's compare signal
65536	Start Simultaneous Starting when axis_0 is stopped.
131072	Start Simultaneous Starting when axis_1 is stopped.
262144	Start Simultaneous Starting when axis_2 is stopped.
524288	Start Simultaneous Starting when axis_3 is stopped.
1048576	Start Simultaneous Starting when axis_4 is stopped.
2097152	Start Simultaneous Starting when axis_5 is stopped.

Comments:

The axis will be waiting status if call Acm_AxSimStartSuspendAbs, Acm_AxSimStartSuspendRel, or Acm_AxSimStartSuspendVel successfully. The axis start motion after calling Acm_AxSimStart and stop motion afer calling Acm_AxSimStop.

The simultaneous starting mode should be set by this property. If the value is 1, the waiting axis will start depending on STA signal. It just needs only one axis of waiting axis to call Acm_AxSimStart or Acm_AxSimStop.

If the value is 256~8192, the simultaneous starting signal comes from compare signal. Every axis needs to assign compare signal source, but cannot assign compare signal of itself to start its simultaneous motion. And ervery simultaneous axis needs to call Acm_AxSimStop to stop motion.

If the value is 65536~2097152, the wait axis will be started simultaneous motion when specified axis's motion is stoped. Every axis needs to specify an axis, but can not be itself. And ervery simultaneous axis needs to call Acm_AxSimStop to stop motion.

If the value is 0. The simultaneous motion is disabled.

You can get axis supported simultaneous mode from FT_AxSimStartSourceMap.

6.4.3.2.19 Trigger Stop

6.4.3.2.19.1 CFG_AxDI1StopEnable

Data Type:

U32

R/W:

R&W

PropertyID:

635

Meaning:

Enable/disable INI trigger stop function.

Comments:

Value	Description
0	Enabled

1	Disabled

6.4.3.2.19.2 CFG_AxDI1StopReact

Data Type:

U32

R/W:

R&W

PropertyID:

636

Meaning:

Set/get INI trigger stop mode.

Comments:

Value	Description
0	Sudden stop
1	Decelerating

6.4.3.2.19.3 CFG_AxDI1StopLogic

Data Type:

U32

R/W:

R&W

PropertyID:

637

Meaning:

Set/get the active logic of IN1 trigger stop function.

Comments:

Value	Description
0	Active low
1	Active high

6.4.3.2.19.4 CFG_AxDI2StopEnable

Data Type:

U32

R/W:

R&W

PropertyID:

638

Meaning:

Enable/disable IN2 trigger stop function.

Comments:

Value	Description
0	Enabled
1	Disabled

6.4.3.2.19.5 CFG_AxDI2StopReact

Data Type:

U32

R/W:

R&W

PropertyID:

639

Meaning:

Set/get IN2 trigger stop mode.

Comments:

Value	Description
0	Sudden stop
1	Decelerating

6.4.3.2.19.6 CFG_AxDI2StopLogic

Data Type:

U32

R/W:

R&W

PropertyID:

640

Meaning:

Set/get the active logic of IN2 trigger stop function.

Comments:

Value	Description
0	Active low
1	Active high

6.4.3.2.19.7 CFG_AxDI4StopEnable

Data Type:

U32

R/W:

R&W

PropertyID:

641

Meaning:

Enable/disable IN4 trigger stop function.

Comments:

Value	Description
0	Enabled
1	Disabled

6.4.3.2.19.8 CFG_AxDI4StopReact

Data Type:

U32

R/W:

R&W

PropertyID:

642

Meaning:

Set/get IN4 trigger stop mode.

Comments:

Value	Description
0	Sudden stop
1	Decelerating

6.4.3.2.19.9 CFG_AxDI4StopLogic

Data Type:

U32

R/W:

R&W

PropertyID:

643

Meaning:

Set/get the active logic of IN4 trigger stop function.

Comments:

Value	Description
0	Active low
1	Active high

6.4.3.2.19.10 CFG_AxDI5StopEnable

Data Type:

U32

R/W:

R&W

PropertyID:

644

Meaning:

Enable/disable IN5 trigger stop function.

Comments:

Value	Description
0	Enabled
1	Disabled

6.4.3.2.19.11 CFG_AxDI5StopReact

Data Type:

U32

R/W:

R&W

PropertyID:

645

Meaning:

Set/get IN2 trigger stop mode.

Comments:

Value	Description
0	Sudden stop
1	Decelerating

6.4.3.2.19.12 CFG_AxDI5StopLogic

Data Type:

U32

R/W:

R&W

PropertyID:

646

Meaning:

Set/get the active logic of IN5 trigger stop function.

Comments:

Value	Description
0	Active low
1	Active high

6.4.3.3 Parameter

6.4.3.3.1 Speed Pattern

6.4.3.3.1.1 PAR_AxVelLow

Data Type:

F64

R/W:

RW

PropertyID:

401

Meaning:

Set/get low velocity (start velocity) of this axis (Unit: PPU/S).

Comments:

This property value must be smaller than or equal to PAR_AxVelHigh. The default value is 2000 PPU.

6.4.3.3.1.2 PAR_AxVelHigh

Data Type:

F64

R/W:

RW

PropertyID:

402

Meaning:

Set/get high velocity (driving velocity) of this axis (Unit: PPU/s).

Comments:

This property value must be smaller than CFG_AxMaxVel and greater than PAR_AxVelLow.The default value is 8000.

6.4.3.3.1.3 PAR_AxAcc

Data Type:

F64

R/W:

RW

PropertyID:

403

Meaning:

Set/get acceleration of this axis (Unit: PPU/s2).

Comments:

This property value must be smaller than or equal to CFG_AxMaxAcc. The default value is 10000.

6.4.3.3.1.4 PAR_AxDec

Data Type:

F64

R/W:

RW

PropertyID:

404

Meaning:

Set/get deceleration of this axis (Unit: PPU/s^2).

Comments:

This property value must be smaller than or equal to CFG_AxMaxDec. The default value is 10000.

6.4.3.3.1.5 PAR_AxJerk

Data Type:

F64

R/W:

RW

PropertyID:

405

Meaning:

Set/get the type of velocity profile: t-curve or s-curve.

Value	Description
0	T-curve(Default)
1	S-curve

Comments:

The actual jerk is calculated by driver.

If PAR_AxJerk is set to be 1, the PAR_AxAcc not means acceleration but max acceleration and PAR_AxDec not means deceleration but max deceleration.

6.4.3.3.2 Home

6.4.3.3.2.1 PAR_AxHomeCrossDistance

Data Type:

F64

R/W:

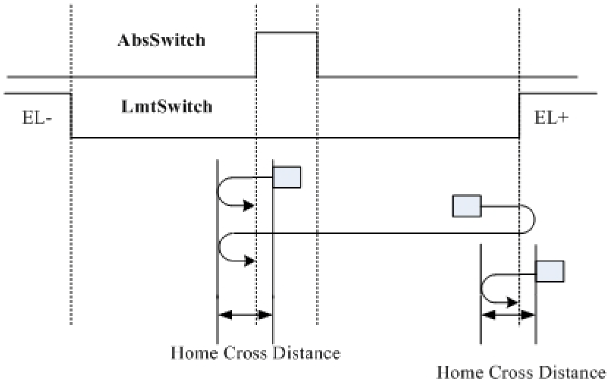
RW

PropertyID:

408

Meaning:

Set the home cross distance (Unit: PPU). This property must be greater than 0. The default value is 10000.



6.4.3.3.2.2 PAR_AxHomeExSwitchMode

Data Type:

U32

R/W:

RW

PropertyID:

407

Meaning:

Setting the stopping condition of Acm_AxHomeEx.

Value	Define	Description
0	LevelOn	When sensor is ON(Active)
1	LevelOff	When sensor is OFF(Non-active)

2	Rising Edge	When OFF to ON transition in sensor
3	Falling Edge	When ON to OFF transition in sensor

6.4.4 Group

6.4.4.1 Config

6.4.4.1.1 System

6.4.4.1.1.1 CFG_GpAxisInGroup

Data Type:

U32

R/W:

R

PropertyID:

806

Meaning:

Get information about which axis is (are) in this group.

Bits	Description
0	0 axis
1	1 axis
2	2 axes
3	3 axes
4	4 axes·(PCI-1265)
5	5 axes·(PCI-1265)

Comments:

6.4.4.1.2 Path

6.4.4.1.2.1 CFG_GpBldTime

Data Type:

U32

R/W:

RW

PropertyID:

808

Meaning:

Set/get blengding time when add a path into system path buffer.

Comments:

It should be 0~65535 and multiple of 2.Unit: ms.

See about Acm_GpAddPath. (PCI-1245E/PCI-1245V not support)

6.4.4.1.2.2 CFG_GpSFEnable

Data Type:

U32

R/W:

RW

PropertyID:

809

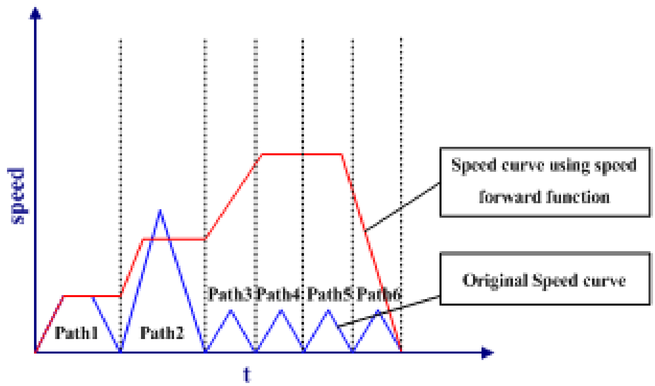
Meaning:

Enable/Disable speed forward function.

Value	Description
0	Disabled
1	Enabled

Comments:

It can not support S profile speed curve. In this mode, the speed parameter of Acm_AddPath is useless, just speed setting of group is used. PCI-1245E/PCI-1245V does not support this mode.



6.4.4.2 Parameter

6.4.4.2.1 Speed Pattern

6.4.4.2.1.1 PAR_GpVelLow

Data Type:

F64

R/W:

RW

PropertyID:

701

Meaning:

Set low velocity (start velocity) of this group (Unit: PPU/s). This property value must be smaller than or equal to Par_GpVelHigh . The default value is the first added axis' PAR_AxVelLow.

6.4.4.2.1.2 PAR_GpVelHigh

Data Type:

F64

R/W:

RW

PropertyID:

702

Meaning:

Set high velocity (driving velocity) of this group (Unit: PPU/s). This property value must be smaller than first added axis' CFG_AxMaxVel and greater than Par_GpVelLow. The default value is the first added axis' PAR_AxVelHigh.

6.4.4.2.1.3 PAR_GpAcc

Data Type:

F64

R/W:

RW

PropertyID:

703

Meaning:

Set acceleration of this group (Unit: PPU/s2). This property value must be smaller than or equal to first added axis' CFG_AxMaxAcc. The default value is the first added axis' PAR_AxAcc.

6.4.4.2.1.4 PAR_GpDec

Data Type:

F64

R/W:

RW

PropertyID:

704

Meaning:

Set deceleration of this group (Unit: PPU/s2). This property value must be smaller than or equal to first added axis' CFG_AxMaxDec. The default value is the first added axis' PAR_AxDec.

6.4.4.2.1.5 PAR_GpJerk

Data Type:

F64

R/W:

RW

PropertyID:

705

Meaning:

Set the type of velocity profile: t-curve or s-curve.

Value	Description
0	T-curve(Default)
1	S-curve

Comments:

If PAR_GpJerk is set to 1, the PAR_GpAcc doesn't mean acceleration but max acceleration and PAR_GpDec doesn't means deceleration but max deceleration. The default value is the first added axis jerk.

6.4.4.2.2 System

6.4.4.2.2.1 PAR_GpGroupID

Data Type:

U32

R/W:

R

PropertyID:

706

Meaning:

Get the GroupID through GroupHandle.

Comments:

In PCI-1265, there are only three GroupID to use. They are 0, 1 and 2. You cannot handle more than three groups at the same time, you must close one group to create new group if there are already three groups.

In PCI-1245/1245V/1245E, there are only two GroupID to use. They are 0, 1. You cannot handle more than two groups at the same time, you must close one group to create new group if there are already two groups.

6.4.4.2.3 Path

6.4.4.2.3.1 PAR_GpRefPlane

DataType:

U32

R/W:

RW

PropertyID:

709

Meaning:

Set/get reference plane for helix motion and arc interpolation.

Value	Description
0	XY PLANE
1	YZ PLANE
2	XZ PLANE

Comments:

See about Acm_GpMoveHelixAbs, Acm_GpMoveHelixRel, Acm_GpMoveHelixAbs_3p, and Acm_GpMoveHelixRel_3p.

6.5 Error Code

Error Code	Error
0x00000000	SUCCESS
0x80000000	InvalidDevNumber
0x80000001	DevRegDataLost
0x80000002	LoadDllFailed
0x80000003	GetProcAddrFailed
0x80000004	MemAllocateFailed
0x80000005	InvalidHandle
0x80000006	CreateFileFailed
0x80000007	OpenEventFailed
0x80000008	EventTimeOut
0x80000009	InvalidInputParam
0x8000000a	PropertyIDNotSupport
0x8000000b	PropertyIDReadOnly
0x8000000c	ConnectWinIrqFailed
0x8000000d	InvalidAxCfgVel
0x8000000e	InvalidAxCfgAcc
0x8000000f	InvalidAxCfgDec
0x80000010	InvalidAxCfgJerk
0x80000011	InvalidAxParVelLow
0x80000012	InvalidAxParVelHigh
0x80000013	InvalidAxParAcc
0x80000014	InvalidAxParDec
0x80000015	InvalidAxParJerk
0x80000016	InvalidAxPulseInMode
0x80000017	InvalidAxPulseOutMode
0x80000018	InvalidAxAlarmEn
0x80000019	InvalidAxAlarmLogic
0x8000001a	InvalidAxInPEn
0x8000001b	InvalidAxInPLogic
0x8000001c	InvalidAxHLmtEn
0x8000001d	InvalidAxHLmtLogic

0x8000001e	InvalidAxHLmtReact
0x8000001f	InvalidAxSLmtPEn
0x80000020	InvalidAxSLmtPReact
0x80000021	InvalidAxSLmtPValue
0x80000022	InvalidAxSLmtMEn
0x80000023	InvalidAxSLmtMReact
0x80000024	InvalidAxSLmtMValue
0x80000025	InvalidAxOrgLogic
0x80000026	InvalidAxOrgEnable
0x80000027	InvalidAxEzLogic
0x80000028	InvalidAxEzEnable
0x80000029	InvalidAxEzCount
0x8000002a	InvalidAxState
0x8000002b	InvalidAxInEnable
0x8000002c	InvalidAxSvOnOff
0x8000002d	InvalidAxDistance
0x8000002e	InvalidAxPosition
0x8000002f	InvalidAxHomeModeKw
0x80000030	InvalidAxCntInGp
0x80000031	AxInGpNotFound
0x80000032	AxisInOtherGp
0x80000033	AxCannotIntoGp
0x80000034	GpInDevNotFound
0x80000035	InvalidGpCfgVel
0x80000036	InvalidGpCfgAcc
0x80000037	InvalidGpCfgDec
0x80000038	InvalidGpCfgJerk
0x80000039	InvalidGpParVelLow
0x8000003a	InvalidGpParVelHigh
0x8000003b	InvalidGpParAcc
0x8000003c	InvalidGpParDec
0x8000003d	InvalidGpParJerk
0x8000003e	JerkNotSupport

0x8000003f	ThreeAxNotSupport
0x80000040	DevIpoNotFinished
0x80000041	InvalidGpState
0x80000042	OpenFileFailed
0x80000043	InvalidPathCnt
0x80000044	InvalidPathHandle
0x80000045	InvalidPath
0x80000046	IoctlError
0x80000047	AmnetRingUsed
0x80000048	DeviceNotOpened
0x80000049	InvalidRing
0x8000004a	InvalidSlaveIP
0x8000004b	InvalidParameter
0x8000004c	InvalidGpCenterPosition
0x8000004d	InvalidGpEndPosition
0x8000004e	InvalidAddress
0x8000004f	DeviceDisconnect
0x80000050	DataOutBufExceeded
0x80000051	SlaveDeviceNotMatch
0x80000052	SlaveDeviceError
0x80000053	SlaveDeviceUnknow
0x80000054	FunctionNotSupport
0x80000055	InvalidPhysicalAxis
0x80000056	InvalidVelocity
0x80000057	InvalidAxPulseInLogic
0x80000058	InvalidAxPulseInSource
0x80000059	InvalidAxErcLogic
0x8000005a	InvalidAxErcOnTime
0x8000005b	InvalidAxErcOffTime
0x8000005c	InvalidAxErcEnableMode
0x8000005d	InvalidAxSdEnable
0x8000005e	InvalidAxSdLogic
0x8000005f	InvalidAxSdReact

0x80000060	InvalidAxSdLatch
0x80000061	InvalidAxHomeResetEnable
0x80000062	InvalidAxBacklashEnable
0x80000063	InvalidAxBacklashPulses
0x80000064	InvalidAxVibrationEnable
0x80000065	InvalidAxVibrationRevTime
0x80000066	InvalidAxVibrationFwdTime
0x80000067	InvalidAxAlarmReact
0x80000068	InvalidAxLatchLogic
0x80000069	InvalidFwMemoryMode
0x8000006a	InvalidConfigFile
0x8000006b	InvalidAxEnEvtArraySize
0x8000006c	InvalidAxEnEvtArray
0x8000006d	InvalidGpEnEvtArraySize
0x8000006e	InvalidGpEnEvtArray
0x8000006f	InvalidIntervalData
0x80000070	InvalidEndPosition
0x80000071	InvalidAxisSelect
0x80000072	InvalidTableSize
0x80000073	InvalidGpHandle
0x80000074	InvalidCmpSource
0x80000075	InvalidCmpMethod
0x80000076	InvalidCmpPulseMode
0x80000077	InvalidCmpPulseLogic
0x80000078	InvalidCmpPulseWidth
0x80000079	InvalidPathFunctionID
0x8000007a	SysBufAllocateFailed
0x80000096	SlaveIOUpdateError
0x80000097	NoSlaveDevFound
0x80000098	MasterDevNotOpen
0x80000099	MasterRingNotOpen
0x800000c8	InvalidDIPort
0x800000c9	InvalidDOPort

0x800000ca	InvalidDOValue
0x800000cb	CreateEventFailed
0x800000cc	CreateThreadFailed
0x800000cd	InvalidHomeModeEx
0x800000ce	InvalidDirMode
0x800000cf	AxHomeMotionFailed
0x800000d0	ReadFileFailed
0x800000d1	PathBufIsFull
0x800000d2	PathBufIsEmpty
0x800000d3	GetAuthorityFailed
0x800000d4	GpIDAllocatedFailed
0x800000d5	FirmWareDown
0x800000d6	InvalidGpRadius
0x800000d7	InvalidAxCmd
0x800000d8	InvalidaxExtDrv
0x800000d9	InvalidGpMovCmd
0x800000da	SpeedCurveNotSupported
0x800000db	InvalidCounterNo
0x800000dc	InvalidPathMoveMode
0x800000dd	PathSelStartCantRunInSpeedForwareMode
0x800000de	InvalidCamTableID
0x800000df	InvalidCamPointRange
0x800000d0	CamTableIsEmpty
0x800000e1	InvalidPlaneVector
0x800000e2	MasAxIDSameSlvAxID
0x800000e3	InvalidGpRefPlane
0x800000e4	InvalidAxModuleRange
0x800000e5	DownloadFileFailed
0x800000e6	InvalidFileLength
0x800000e7	InvalidCmpCnt
0x80002000	HLmtPExceeded
0x80002001	HLmtNExceeded
0x80002002	SLmtPExceeded

0x80002003	SLmtNExceeded
0x80002004	AlarmHappened
0x80002005	EmgHappened
0x80002006	TimeLmtExceeded
0x80002007	DistLmtExceeded
0x80002008	InvalidPositionOverride
0x80002009	OperationErrorHappened
0x8000200a	SimultaneousStopHappened
0x8000200b	OverflowInPAPB
0x8000200c	OverflowInIPO
0x8000200d	STPHappened
0x8000200e	SDHappened
0x8000200f	AxsiNoCmpDataLeft
0x80004001	DevEvtTimeOut
0x80004002	DevNoEvt
0x10000001	Warning_AxWasInGp
0x10000002	Warning_GpInconsistRate
0x10000003	Warning_GpInconsistPPU
0x80005001	ERR_SYS_TIME_OUT
0x80005002	Dsp_PropertyIDNotSupport
0x80005003	Dsp_PropertyIDReadOnly
0x80005004	Dsp_InvalidParameter
0x80005005	Dsp_DataOutBufExceeded
0x80005006	Dsp_FunctionNotSupport
0x80005007	Dsp_InvalidConfigFile
0x80005008	Dsp_InvalidIntervalData
0x80005009	Dsp_InvalidTableSize
0x8000500a	Dsp_InvalidTableID
0x8000500b	Dsp_DataIndexExceedBufSize
0x8000500c	Dsp_InvalidCompareInterval
0x8000500d	Dsp_InvalidCompareRange
0x8000500e	Dsp_PropertyIDWriteOnly
0x8000500f	Dsp_NcError

0x80005010	Dsp_CamTableIsInUse
0x80005011	Dsp_EraseBlockFailed
0x80005012	Dsp_ProgramFlashFailed
0x80005101	Dsp_InvalidAxCfgVel
0x80005102	Dsp_InvalidAxCfgAcc
0x80005103	Dsp_InvalidAxCfgDec
0x80005104	Dsp_InvalidAxCfgJerk
0x80005105	Dsp_InvalidAxParVelLow
0x80005106	Dsp_InvalidAxParVelHigh
0x80005107	Dsp_InvalidAxParAcc
0x80005108	Dsp_InvalidAxParDec
0x80005109	Dsp_InvalidAxParJerk
0x8000510a	Dsp_InvalidAxPptValue
0x8000510b	Dsp_InvalidAxState
0x8000510c	Dsp_InvalidAxSvOnOff
0x8000510d	Dsp_InvalidAxDistance
0x8000510e	Dsp_InvalidAxPosition
0x8000510f	Dsp_InvalidAxHomeMode
0x80005110	Dsp_InvalidPhysicalAxis
0x80005111	Dsp_HLmtPExceeded
0x80005112	Dsp_HLmtNExceeded
0x80005113	Dsp_SLmtPExceeded
0x80005114	Dsp_SLmtNExceeded
0x80005115	Dsp_AlarmHappened
0x80005116	Dsp_EmgHappened
0x80005117	Dsp_CmdValidOnlyInConstSec
0x80005118	Dsp_InvalidAxCmd
0x80005119	Dsp_InvalidAxHomeDirMode
0x80005120	Dsp_NotEnoughPulseForChgV
0x8000511a	Dsp_AxisMustBeModuloAxis
0x8000511b	Dsp_AxIdCantSameAsMasId
0x8000511c	Dsp_CantResetPosiOfMasAxis
0x8000511d	Dsp_InvalidAxExtDrvOperation

0x8000511e	Dsp_AxAccExceededMaxAcc
0x8000511f	Dsp_AxVelExceededMaxVel
0x80005201	Dsp_InvalidAxCntInGp
0x80005202	Dsp_AxInGpNotFound
0x80005203	Dsp_AxisInOtherGp
0x80005204	Dsp_AxCannotIntoGp
0x80005205	Dsp_GpInDevNotFound
0x80005206	Dsp_InvalidGpCfgVel
0x80005207	Dsp_InvalidGpCfgAcc
0x80005208	Dsp_InvalidGpCfgDec
0x80005209	Dsp_InvalidGpCfgJerk
0x8000520a	Dsp_InvalidGpParVelLow
0x8000520b	Dsp_InvalidGpParVelHigh
0x8000520c	Dsp_InvalidGpParAcc
0x8000520d	Dsp_InvalidGpParDec
0x8000520e	Dsp_InvalidGpParJerk
0x8000520f	Dsp_JerkNotSupport
0x80005210	Dsp_ThreeAxNotSupport
0x80005211	Dsp_DevIpoNotFinished
0x80005212	Dsp_InvalidGpState
0x80005213	Dsp_OpenFileFailed
0x80005214	Dsp_InvalidPathCnt
0x80005215	Dsp_InvalidPathHandle
0x80005216	Dsp_InvalidPath
0x80005217	Dsp_GpSlavePositionOverMaster
0x80005219	Dsp_GpPathBufferOverflow
0x8000521a	Dsp_InvalidPathFunctionID
0x8000521b	Dsp_SysBufAllocateFailed
0x8000521c	Dsp_InvalidGpCenterPosition
0x8000521d	Dsp_InvalidGpEndPosition
0x8000521e	Dsp_InvalidGpCmd
0x8000521f	Dsp_AxHasBeenInInGp
0x80005220	Dsp_InvalidPathRange

2

APPENDIX

A

Software Function Comparison Table

Appendix A Software Function Comparison Table

A.1 Software Function Comparison Table

	Item	Description	PCI-1245E	PCI-1245V	PCI-1245	PCI-1265

Motion Functions	Single-axis motion	Jog move	√	√	√	√
		MPG move	√	√	√	√
		T&S-curve speed profile	√	√	√	√
		Prog. acc. and dec.	√	√	√	√
		Point to point motion	√	√	√	√
		Position / Speed Override	√	√	√	√
		Velocity motion	√	√	√	√
		Backlash compensation	√	√	√	√
		Superimposed move			√	√
		Stop	√	√	√	√
	Multi-axes (Group) motion	up to 3 groups	2 groups	2 groups	2 groups	3 groups
		Line: up to 6 axes	2 axes	2 axes	4 axes	6 axes
		2-axes Circular		√	√	√
		Speed Override			√	√
		Helical			√	√
		Pause&Resume	√	√	√	√
	Home	16 modes	√	√	√	√
	Path table motion	3 Tables, size: 10K points	√	√	√	√
		Start / End motion list	√	√	√	√
		line trajectory: up to 6 axes	2 axes Line/ Direct	2 axes Line/ Direct	2/3 axes Line, 2-4 axes Direct	2/3 axes Line, 2-6 axes Direct
		Add arc trajectory: 2 axes		√	√	√
		Add Dwell	√	√	√	√
		Start/Sop/Repeat	√	√	√	√
		Auto Blending			√	√

Applica-tion Function	Gantry				√	√
	Velocity look ahead	Velocity look ahead (Refer to function call, acm_gpmovepath)			√	√
	Tangential Following				√	√
	E-Gear		√	√	√	√
	E-CAM				√	√
	Error check	Error status, Watchdog	√	√	√	√
	CAM DO	Position window output			√	√
	Posi. latch				√	√
	Simultane-ously Start/ Stop	Simultaneously Start/Stop			√	√
Interrupt	Axes	Axes stop	√	√	√	√
		Axes compare			√	√
		Axes error	√	√	√	√
		Axes lock			√	√
		Axes VH start	√	√	√	√
		Axes VH stop	√	√	√	√
	Group	Group stop	√	√	√	√
		Group VH start	√	√	√	√
		Group VH stop	√	√	√	√
Trigger Function	Single Compare	Up to 6 channels			√	√
	Table Compare	Up to 2 channels			√	√
	Linear Compare	(Table size: 100 K points)			√	√

2

APPENDIX B

Specifications

Appendix B Specifications

B.1 Axes

<table>
<tr><td>Axes</td><td colspan="2">PCI-1245/1245V/1245E: 4 axes / PCI-1265: 6 axes</td></tr>
<tr><td rowspan="2">2/3-Axis Linear Interpolation</td><td>Range</td><td>For each axis:
-2,147,483,648 ~ +2,147,483,648</td></tr>
<tr><td>Speed</td><td>1 PPS ~ 5 MPPS</td></tr>
<tr><td rowspan="2">2-Axis Circular Interpolation</td><td>Range</td><td>-2,147,483,648 ~ +2,147,483,648</td></tr>
<tr><td>Speed</td><td>1 PPS ~ 5 MPPS</td></tr>
</table>

B.2 Digital Input/Output

<table>
<tr><td rowspan="9">Input Signals</td><td>Over Traveling Limit Switch Input*</td><td colspan="2">LMT+ and LMT-</td></tr>
<tr><td>Signal for Servo Motor Drives*</td><td colspan="2">RDY (Servo ready); INP (In-Position Complete); ALM (Servo Alarm); ERC (Error Counter Clear)</td></tr>
<tr><td>Emergency Stop</td><td colspan="2">EMG - one emergency stop input</td></tr>
<tr><td>Max. Input Frequency</td><td colspan="2">4 kHz</td></tr>
<tr><td rowspan="3">Input Voltage</td><td>Low</td><td>3 V_{DC} max.</td></tr>
<tr><td rowspan="2">High</td><td>10 V_{DC} min.</td></tr>
<tr><td>30 V_{DC} max.</td></tr>
<tr><td>Input Resistance</td><td colspan="2">3200 Ω</td></tr>
<tr><td>Protection</td><td colspan="2">2,500 V_{DC} photo coupler isolation and RC filtering</td></tr>
<tr><td rowspan="4">General Purpose Output Signals</td><td>Output Signal</td><td colspan="2">DO</td></tr>
<tr><td>Output Voltage</td><td colspan="2">Open Collector 5 ~ 40 V_{DC}</td></tr>
<tr><td>Sink Current</td><td colspan="2">100 mA max./channel</td></tr>
<tr><td>Protection</td><td colspan="2">2,500 V_{DC} photo coupler isolation</td></tr>
</table>

B.3 Analog Input

Channel Count	2	
Resolution	12-bit	
Sampling Frequency	>100 K	
Zero Offset	±30 ppm/° C	
Gain Offset	±30 ppm/° C	
DC Precision	Precision (±10 V)	±0.1% of FSR
	DNLE	±5LSB of FSR
	INLE	±5LSB of FSR
Absolute Value of Max. Input Voltage	±15 V	
Input Impedance	>2 MΩ	

B.4 Input Pulse for Encoder Interface

Input Signal*	ECA, ECB and ECZ	
Encoder Pulse Input Type	Quadrature (A/B phase) or Up/Down x1, x2, x4 (A/B phase only)	
Counts per Encoder Cycle	x1, x2, x4 (A/B phase only)	
Max. Input Frequency	2.5 MHz	
Input Voltage	Low	$V_+ - V_- < 1$ V
	High	$V_+ - V_- > 2.5$ V
Protection	2,500 V_{DC} Isolation Protection	

B.5 General

I/O Connector Type	100-pin SCSI-II female	
Dimensions	175 x 100 mm (6.9” x 3.9”)	
Power Consumption	Typical	+5 V @ 850 mA
	Max.	+5 V @ 1 A
External Power Voltage	DC +12 ~ 24 V	
Temperature	Operating	0 ~ 60° C (32 ~ 140° F) (refer to IEC 60068-2-1,2)
	Storage	-20 ~ 85° C (-4 ~ 185° F)
Relative Humidity	5~95% RH non-condensing (refer to IEC 60068-2-3)	
Certifications	CE,FCC certified	