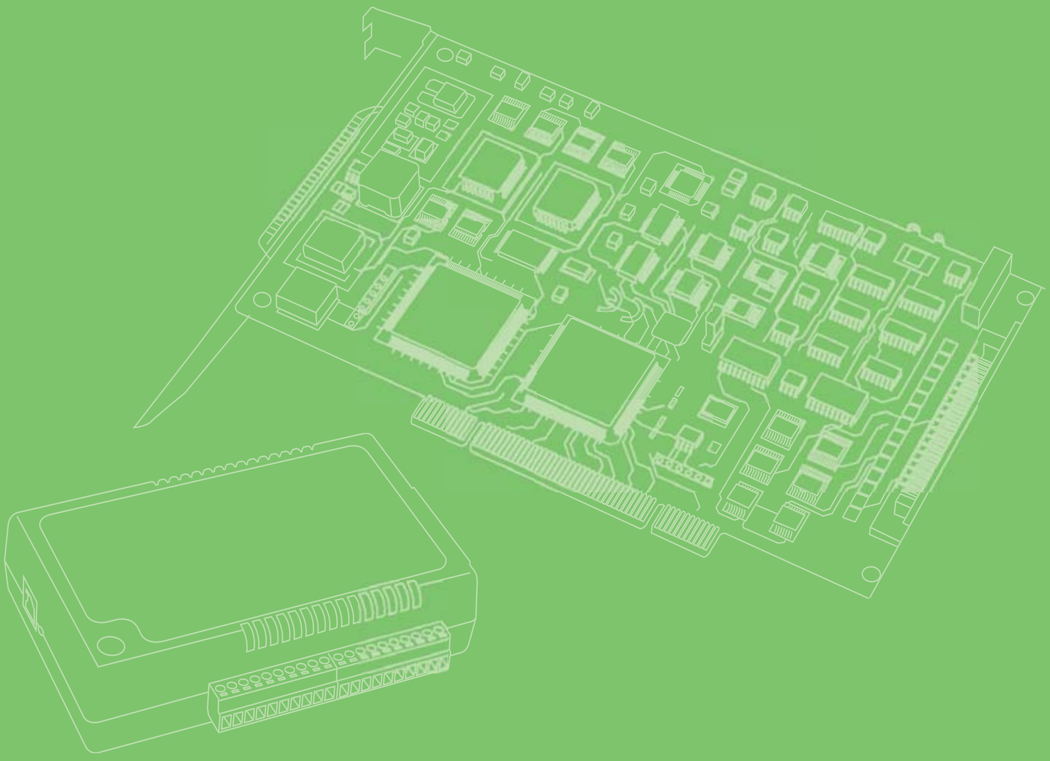
User Manual



PCI-1245L

Basic 4-Axis SoftMotion PCI Controller

ADVANTECH

Enabling an Intelligent Planet

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-1245L, 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.

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.

Part No. 2003124520 Edition 1
Printed in Taiwan July 2013

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-1245L
■ 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.

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.

Warnings, Cautions and Notes

Warning! *Warnings indicate conditions, which if not observed, can cause personal injury!*

Caution! *Cautions are included to help you avoid damaging hardware or losing data. e.g.*

There is a danger of a new battery exploding if it is incorrectly installed. Do not attempt to recharge, force open, or heat the battery. Replace the battery only with the same or equivalent type recommended by the manufacturer. Discard used batteries according to the manufacturer's instructions.

Note! *Notes provide optional additional information.*

Contents

Appendix A Software Function Table 163

Appendix B Specifications 165

Chapter 1

Introduction

This chapter introduces PCI-1245L and lists their special features and detailed specifications.

PCI-1245L are basic 4-axis 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 FPGA with SoftMotion algorithm inside to perform the motion trajectory and precise movement.

SoftMotion features supported by PCI-1245L are Jog move;MPG move;programmable acceleration and deceleration;T&S-curve speed profile; 2-axis in linear interpolation and simultaneously Start/Stop.

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-1245L are featured by the following points:

- Encoder input is 4 MHz for 4xAB mode, 1 MHz for CW/CCW mode
- Pulse output is up to 1 Mpps and can be selected to differential output or single +5V out by jumper setting.
- Hardware emergency input
- Watchdog timer
- Programmable interrupt
- RDY-dedicated input channels & SVON/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-1245L card
- User manual
- Driver and software
- Utility
- ADAM-3956 terminal board (4-axis) and PCL-101100M cable (100-pin to 100-pin SCSI connector) or 2 set ADAM-3955 terminal board (2-axis) and PCL-10251 cable (100-pin to 2 x 50-pin SCSI connector)
- 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 to two 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.
- PCL-101100M - PCL-101100M is a 100-pin to 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.

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-1245L 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-1245L board, only one wiring board is 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-3956 to Panasonic A4 and A5 servo.
- PCL-10153PA5LS - PCL-10153PA5LS is a 50-pin cable connecting ADAM-3956 to Panasonic MINAS A servo.
- PCL-10153YS5 - PCL-10153YS5 is a 50-pin cable connecting ADAM-3956 to Yaskawa Sigma V servo.
- PCL-10153MJ3 - PCL-10153MJ3 is a 50-pin cable connecting ADAM-3956 to Mitsubishi J3 servo.
- PCL-10153DA2 - PCL-10153DA2 is a 50-pin cable connecting ADAM-3956 to Delta A2 servo.

Chapter 2

Installation

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

2.1 Unpacking

After receiving your PCI-1245L package, inspect the contents first. The package should include the following items:

- PCI-1245L card
- CD-ROM (DLL driver & user manual included)

The PCI-1245L 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-1245L 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-1245L 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 instal- lation 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-1245L 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-1245L. 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 (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.

Chapter 3

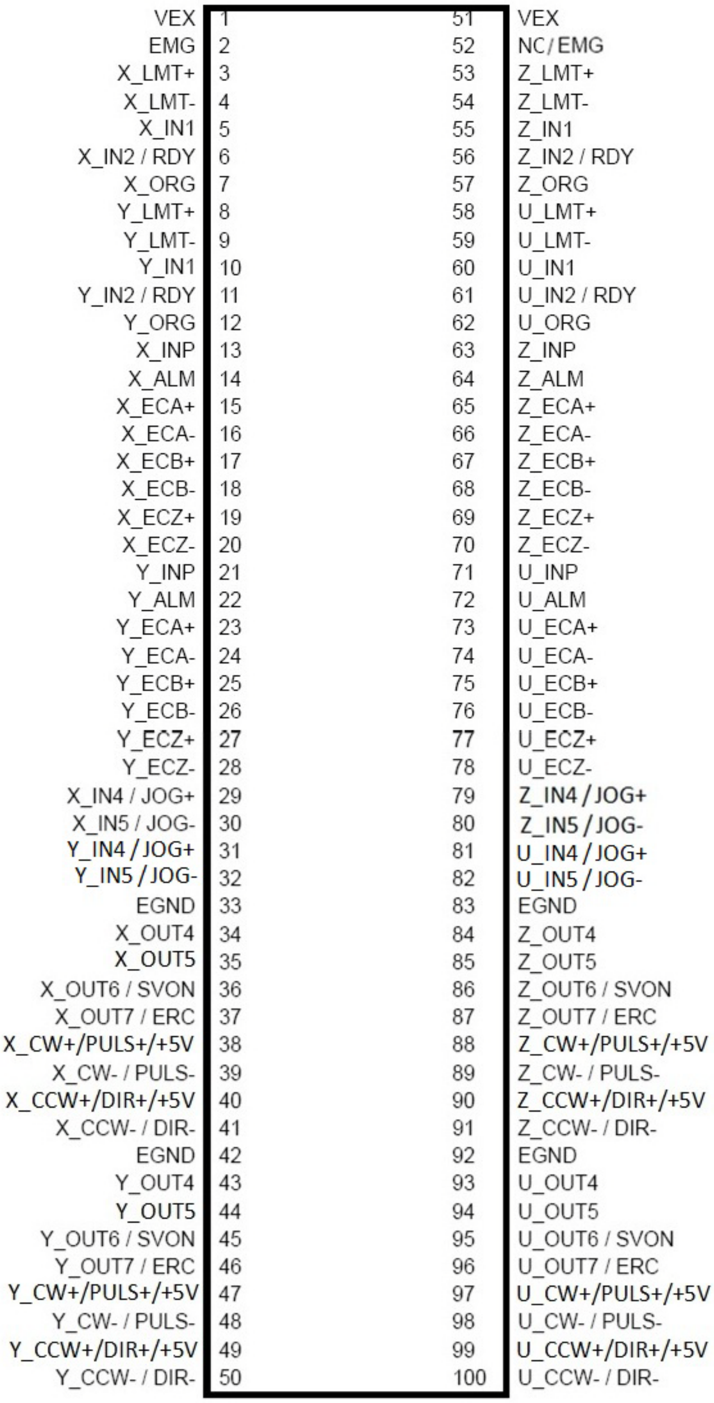
Signal Connections

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

3.1 I/O Connector Pin Assignments

The I/O connector on the PCI-1245L is a 100-pin connector that enables you to connect to two pieces of ADAM-3955 terminal board via the PCL-10251 shielded cable or one piece of ADAM-3956 terminal board via the PCL-101100M shielded cable.

Figure 3.1 shows the pin assignments for the 100-pin I/O connector on the PCI-1245L, and table 3-1 shows its I/O connector signal description.



VEX	1	51	VEX
EMG	2	52	NC / EMG
X_LMT+	3	53	Z_LMT+
X_LMT-	4	54	Z_LMT-
X_IN1	5	55	Z_IN1
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	10	60	U_IN1
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 / JOG+
X_IN5 / JOG-	30	80	Z_IN5 / JOG-
Y_IN4 / JOG+	31	81	U_IN4 / JOG+
Y_IN5 / JOG-	32	82	U_IN5 / JOG-
EGND	33	83	EGND
X_OUT4	34	84	Z_OUT4
X_OUT5	35	85	Z_OUT5
X_OUT6 / SVON	36	86	Z_OUT6 / SVON
X_OUT7 / ERC	37	87	Z_OUT7 / ERC
X_CW+/PULS+/+5V	38	88	Z_CW+/PULS+/+5V
X_CW- / PULS-	39	89	Z_CW- / PULS-
X_CCW+/DIR+/+5V	40	90	Z_CCW+/DIR+/+5V
X_CCW- / DIR-	41	91	Z_CCW- / DIR-
EGND	42	92	EGND
Y_OUT4	43	93	U_OUT4
Y_OUT5	44	94	U_OUT5
Y_OUT6 / SVON	45	95	U_OUT6 / SVON
Y_OUT7 / ERC	46	96	U_OUT7 / ERC
Y_CW+/PULS+/+5V	47	97	U_CW+/PULS+/+5V
Y_CW- / PULS-	48	98	U_CW- / PULS-
Y_CCW+/DIR+/+5V	49	99	U_CCW+/DIR+/+5V
Y_CCW- / DIR-	50	100	U_CCW- / DIR-

Figure 3.1 I/O Connector Pin Assignments for PCI-1245L

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
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
IN	EGND	Input	General-purposed digital input
OUT	EGND	Output	General-purposed digital 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-

Note!

1. *X, Y, Z, U represent for ID of each axis.*
2. *RDY dedicated input channels are designed to be switchable and support general purpose input channel usage.*
3. *SVON and ERC dedicated output channels are designed to be switchable and support general purpose output channel usage.*
4. *IN4 has three switchable functions - general purpose input, JOG+ and MPG+ (Manual Pulser).*
5. *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-1245L. The switch is used to set board ID.

BoardID Switch

PCI-1245L 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.

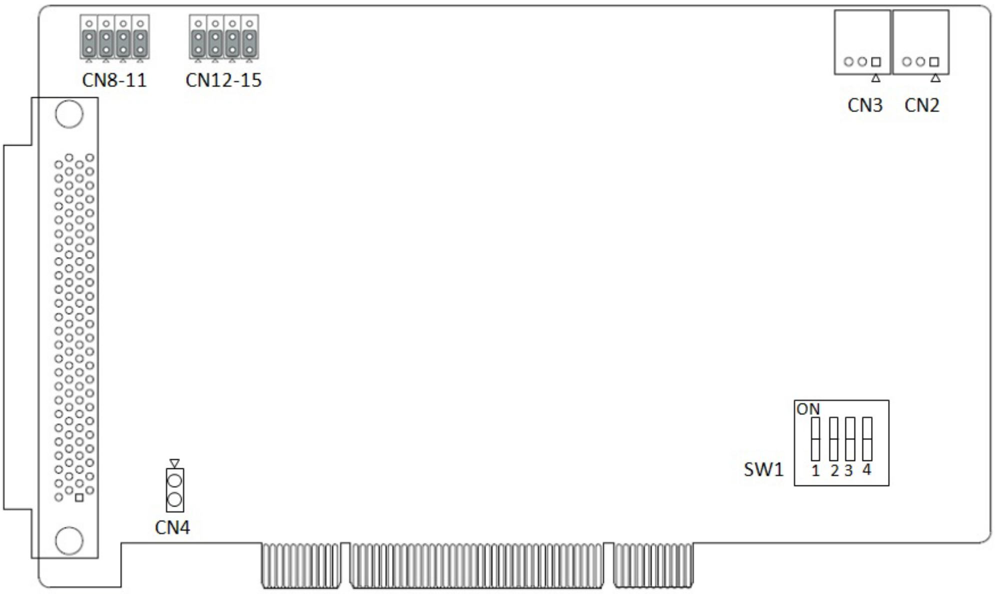


Figure 3.2 Location of Jumpers & DIP Switch

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		

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.

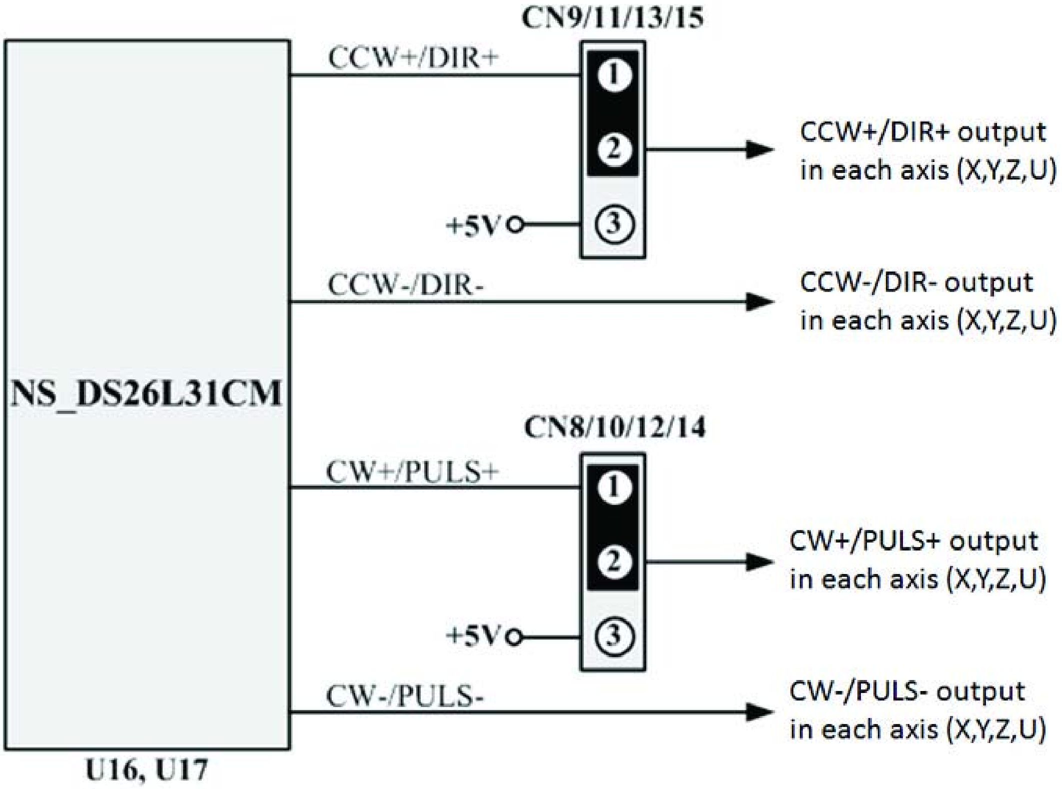

CN9/11/13/15
CCW+/DIR+
1
2
3
+5V
CCW+/DIR+ output in each axis (X,Y,Z,U)
CCW-/DIR-
CCW-/DIR- output in each axis (X,Y,Z,U)
NS_DS26L31CM
CN8/10/12/14
CW+/PULS+
+5V
CW+/PULS+ output in each axis (X,Y,Z,U)
CW-/PULS-
CW-/PULS- output in each axis (X,Y,Z,U)
U16, U17

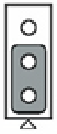
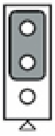
Figure 3.3 Pulse out Signal Diagram

Figure 3-3 shows the default output setting(Pin1 and Pin2 are shorted in CN8-15) is differential mode. If single-end output is needed, user can change the jumper setting. Each axis at the CCW±/DIR± , CW±/PULS± will output +5V as Pin2 and Pin3 are shorted in CN8-15. For Example, the pin outputs of CCW+/DIR+CW+/DIR+ will change to +5V in Z axis as Pin2 and Pin3 are shorted in CN12 and CN13.

Note! *For Stepping motor, CN14 and CN15 must change X axis output mode together; CN10 and CN11 must change Y axis output mode together; CN12 and CN13 must change Z axis output mode together; CN8 and CN9 must change U axis output mode together. PCB indicates the mapping axis for CN8-15: 0 represents X axis, 1 represents Y axis, 2 represents Z axis, 3 represents U axis.*

Note! *You should avoid overloading as the output is in +5V mode. The maximum current provided by these all 4 axes is 120mA.*

Table 3.3: CN8-15Jumper Setting

Jumper	CN8	CN9	CN10	CN11	CN12	CN13	CN14	CN15
	I/O connector pin output							
	Pin 99	Pin 97	Pin 49	Pin 47	Pin 90	Pin 88	Pin 40	Pin 38
	U_CCW+ /DIR+	U_CW+/ PULS+	Y_CCW+/ DIR+	Y_CW+/ PULS+	Z_CCW+/ DIR+	Z_CW+/ PULS+	X_CCW+/ DIR+	X_CW+ /PULS+
	+5V	+5V	+5V	+5V	+5V	+5V	+5V	+5V

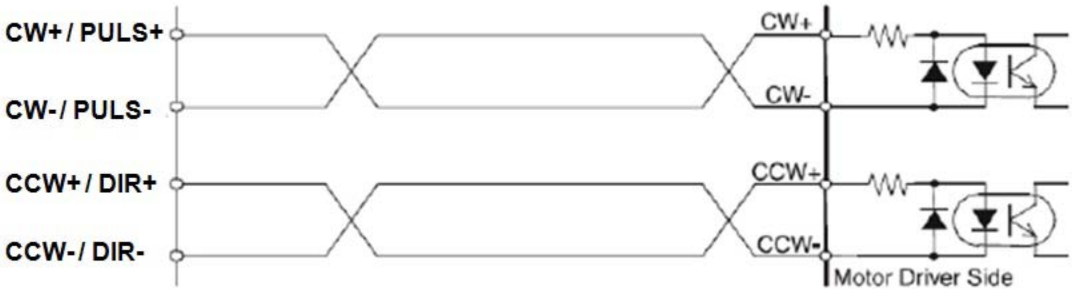


Figure 3.4 Photocoupler Interface

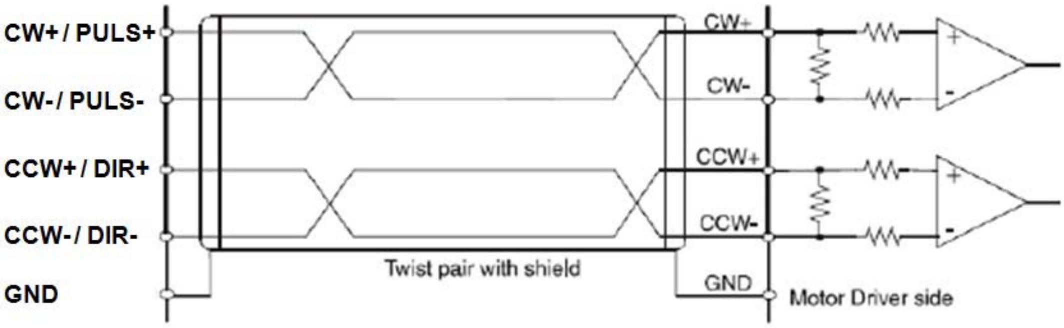


Figure 3.5 Line Drive Interface

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.

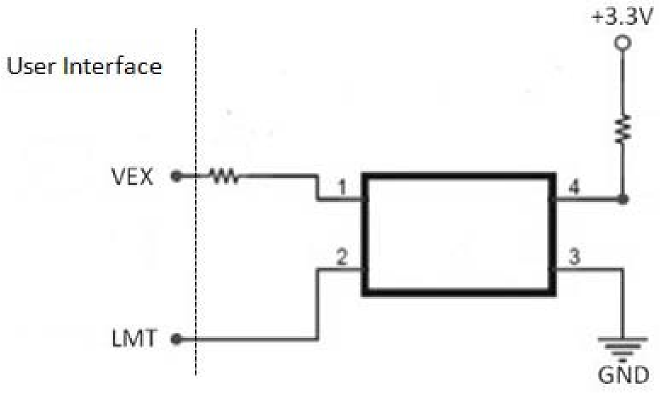


Figure 3.6 Circuit Diagram for Limit Input Signals

3.5 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.

3.6 Home Position [ORG]

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

3.7 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.

3.8 Servo Error & Alarm [ALM]

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

3.9 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-1245L is quadrature input (4xAB phase). The following diagram shows the interface circuit for one channel:

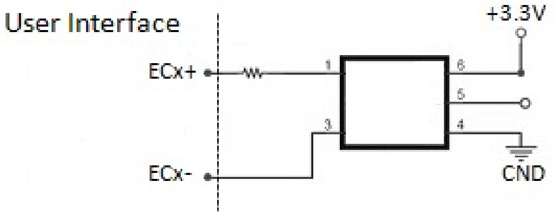


Figure 3.7 Circuit Diagram of Encoder Feedback

In the circuit diagram above, PCI-1245L 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 4 MHz.

3.10 Emergency Stop Input (EMG)

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

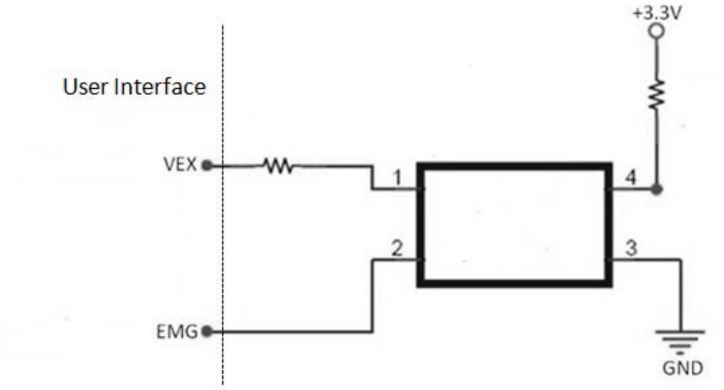


Figure 3.8 Circuit Diagram of Emergency Stop Input Signal

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.

3.11 External Power Input (VEX)

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

Note! *Please don't direct connect VEX to inductive load.*

3.12 Activate Servo ON [SVON]

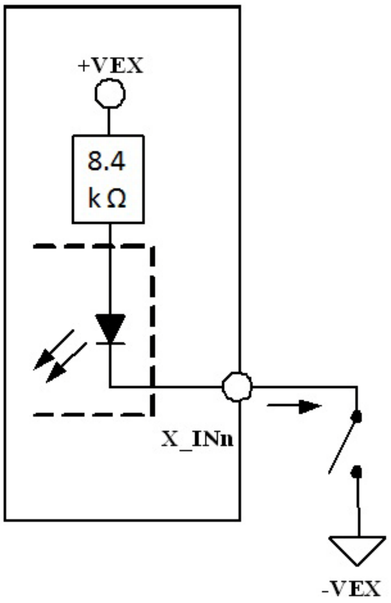
This SVON is to generate a digital output to activate the servo drive to be ready for move status.

3.13 Servo Error Counter Clear [ERC]

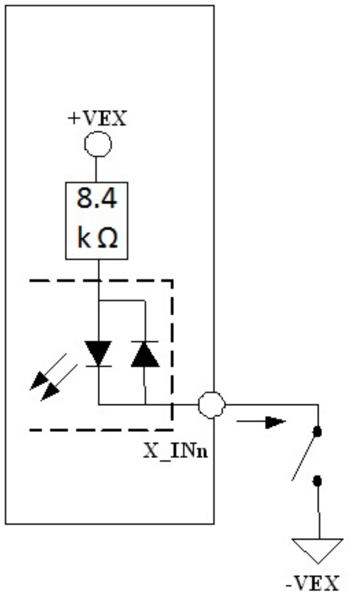
The deviation counter clear is generated by servo drive and the board can receive it as a general purpose input. The counter will be cleared by some instances: homing, emergency stop case, servo alarm and over travelling limit activated.

3.14 Digital Input and Digital Output

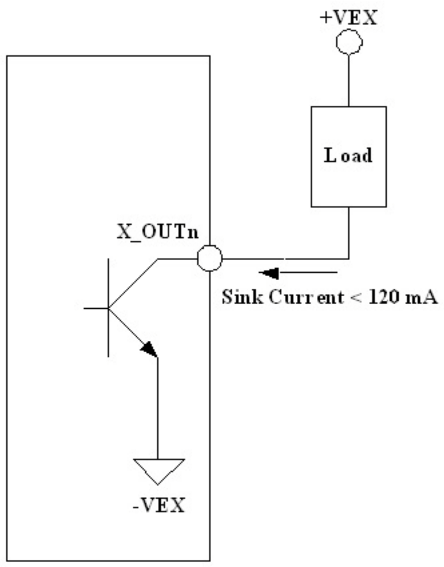
The recommended external wiring connection of DI and DO for PCI-1245L are provided below:



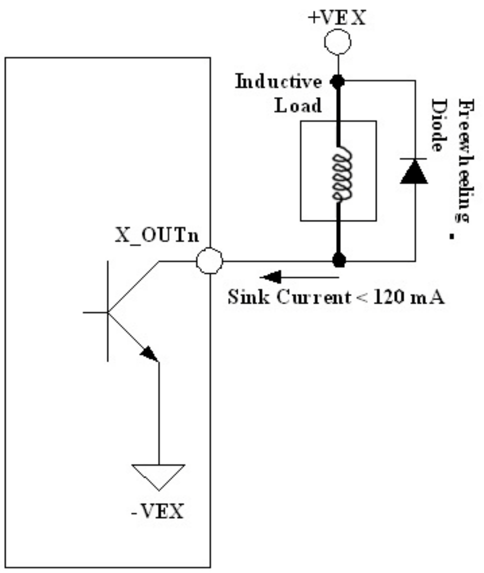
a. High speed DI



b. DI



c. DO with general load



d. DO with inductive load

3.15 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_IN3 has three functions: general purpose digital input, JOG+ and MPG+. X_IN4 also has three functions: general purpose digital input, JOG- and MPG-. Same as Y, Z, U-axis.

3.16 Simultaneous Start and Stop within Multiple Cards

Simultaneous start and stop within multiple cards is supported by connecting the CN2 and CN3 on each card one by one. For the function call of simultaneous start and stop, refer to chapter 6.

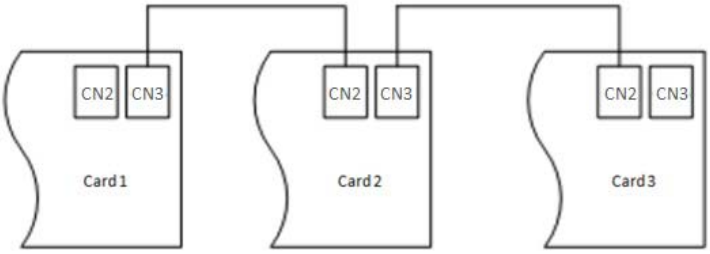


Figure 3.9 Connection of Multiple Cards

Chapter 4

Common Motion API

This chapter introduces common motion API architecture & concept.

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.

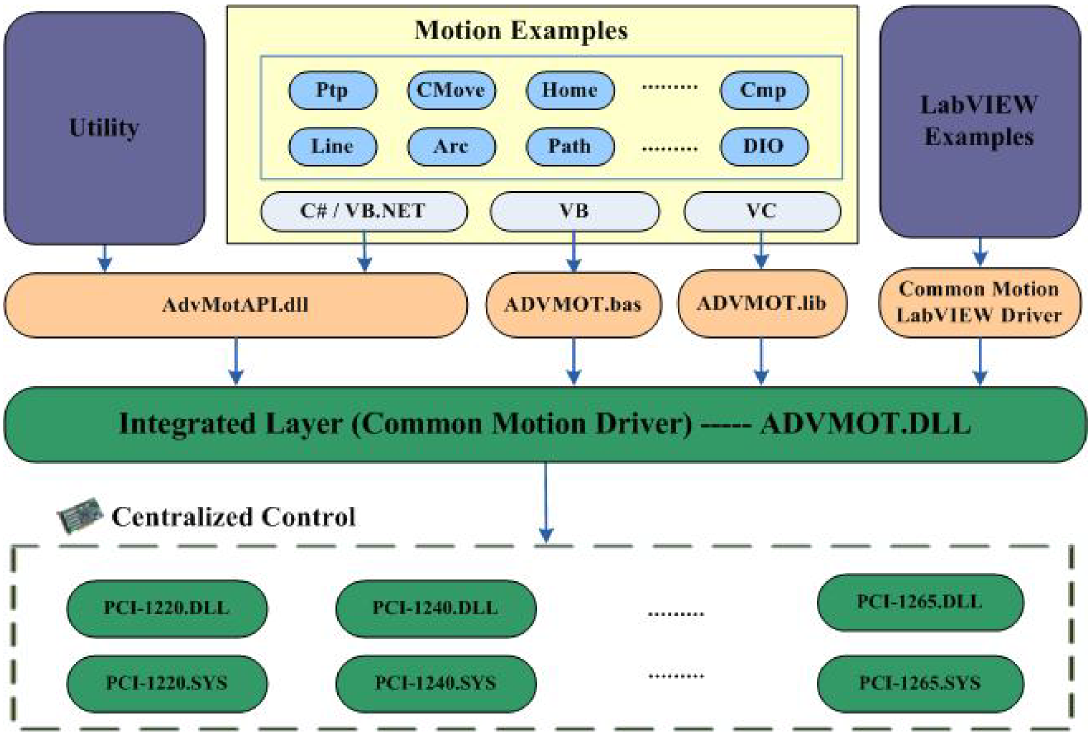
There are three layers in this architecture: Device Driver Layer, Integrated Layer and Application Layer. Users do not need to know how to operate the specific driver of a specify device, but only to know the Common Motion Driver. Even though the device which supports this architecture has changed, the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

- 4th byte
  Master/device type ID (refer to master device type ID table))
- 3rd & 2nd H byte:
  Master/device board ID (or base address)
- 2nd L byte:
  Master ring number, used by remote device, use 0 as default value for local device
- 1st 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-1245L 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	

Table 4.1: Abbreviations and Their Meanings

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
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.

Chapter 5

Utility

This chapter is to describe the comprehensive & graphical utility

5.1 Introduction

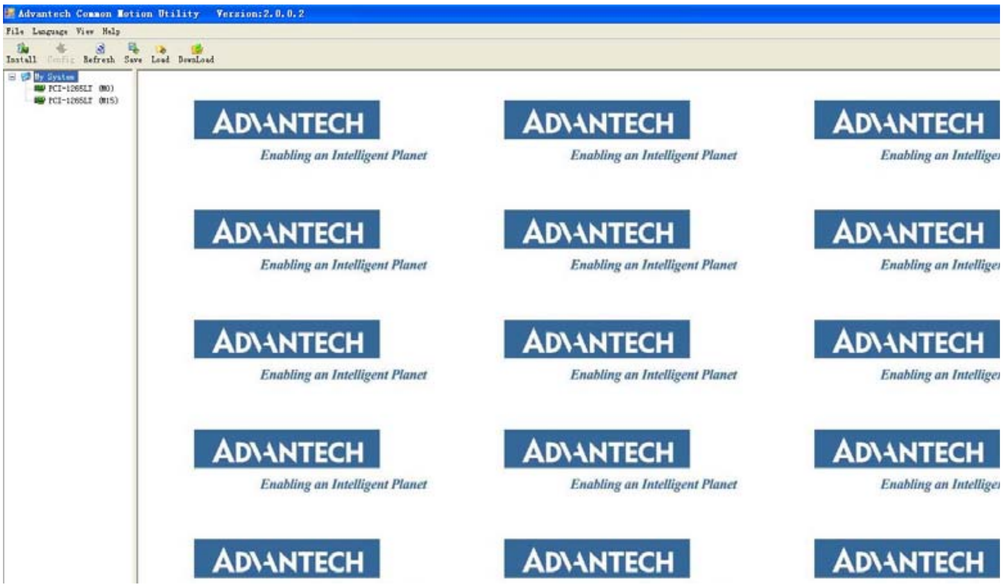
The utility is developed with .Net control library according to Common Motion API architecture. The .Net control library includes control - Device, Axis, Group and component - AxisSetupView, AxisScopeView, AxisDiagView, GroupPathView and GroupSpeedView. The new utility is consistent and compatible with old AdvMotionUtility. The new utility supports PCI-1220U, PCI-1240U, PCI-1245L, PCI-1245/1245V/1245E/1265/1285/1285E series products.

5.1.1 Contents

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

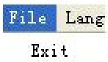
1. Main Form: includes Main Menu, Toolbar and Device Tree.
2. Single-axis Motion: focuses on the I/O and attribute configuration, and status and movement operations (P to P/ Continue/ Homing) of single axis.
3. Multi-axis Motion: focuses on multi-axis (Group) interpolation operation, including the basic Line Interpolation.
4. Synchronized Motion: focuses on synchronized motion operations.
5. Digital Input: displays device's input status.
6. Digital Output: displays device's output status.

5.2 Main Form



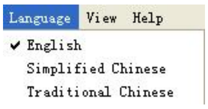
5.2.1 Main Form

5.2.1.1 File



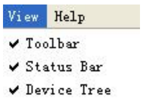
Click [Exit] to terminate this process.

5.2.1.2 Language



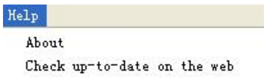
Through this menu, language in Utility can be switched. This utility supports three languages: English, simplified Chinese and traditional Chinese. After you select a language, the corresponding menu item will be checked. When you close the Utility, the language you selected will be saved to register.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

5.2.1.3 View



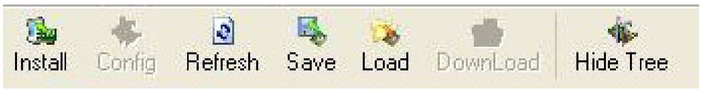
This menu allow users to display/hide the toolbar, status bar and device tree. If Toolbar/Status Bar/Device Tree is visible, the corresponding menu item will be checked.

5.2.1.4 Help



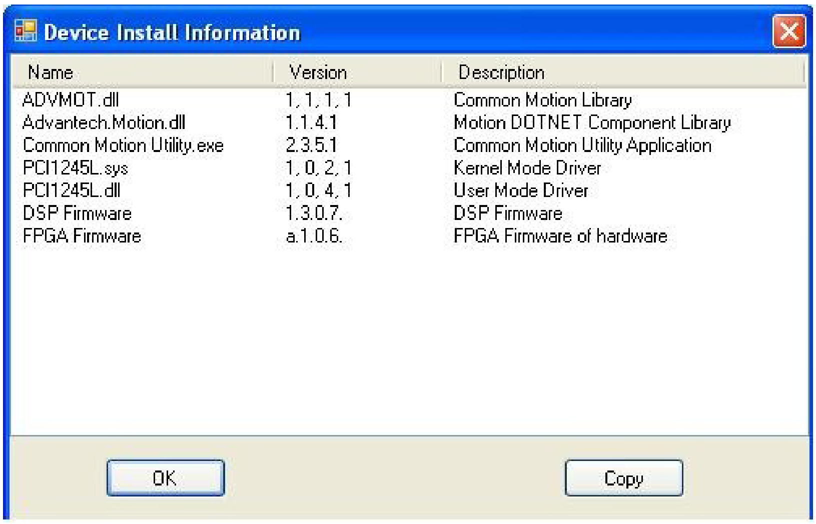
The [About] menu item supports the copyright notice of the driver and utility for device. Click [Check up-to-date on the web], you can link to company's website to check whether the firmware, driver and utility are the latest ones by comparing version information of Install interface.

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.



Name	Version	Description
ADVMOT.dll	1, 1, 1, 1	Common Motion Library
Advantech.Motion.dll	1.1.4.1	Motion DOTNET Component Library
Common Motion Utility.exe	2.3.5.1	Common Motion Utility Application
PCI1245L.sys	1, 0, 2, 1	Kernel Mode Driver
PCI1245L.dll	1, 0, 4, 1	User Mode Driver
DSP Firmware	1.3.0.7.	DSP Firmware
FPGA Firmware	a.1.0.6.	FPGA Firmware of hardware

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! *PCI-1245L is a FPGA-based motion control card without DSP.*

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

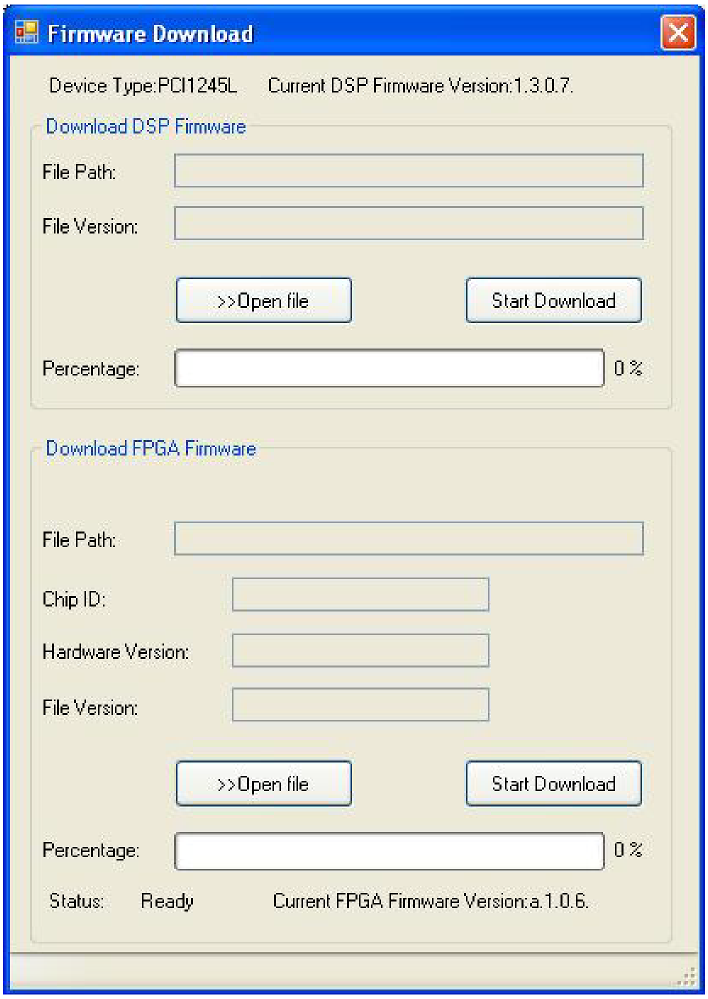
This button can save all properties of the axes of the selected device.

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-1245L motion controller. After clicking device, you can see the interface as follows.



Firmware Download

Device Type:PCI1245L Current DSP Firmware Version:1.3.0.7.

Status: Ready Current FPGA Firmware Version:a.1.0.6.

The tool is to upgrade the FPGA firmware. There is no DSP firmware download function in PCI-1245L. 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.*

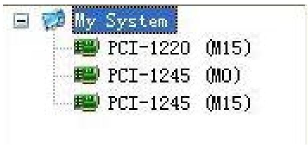
5.2.2.6 Hide Tree

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".

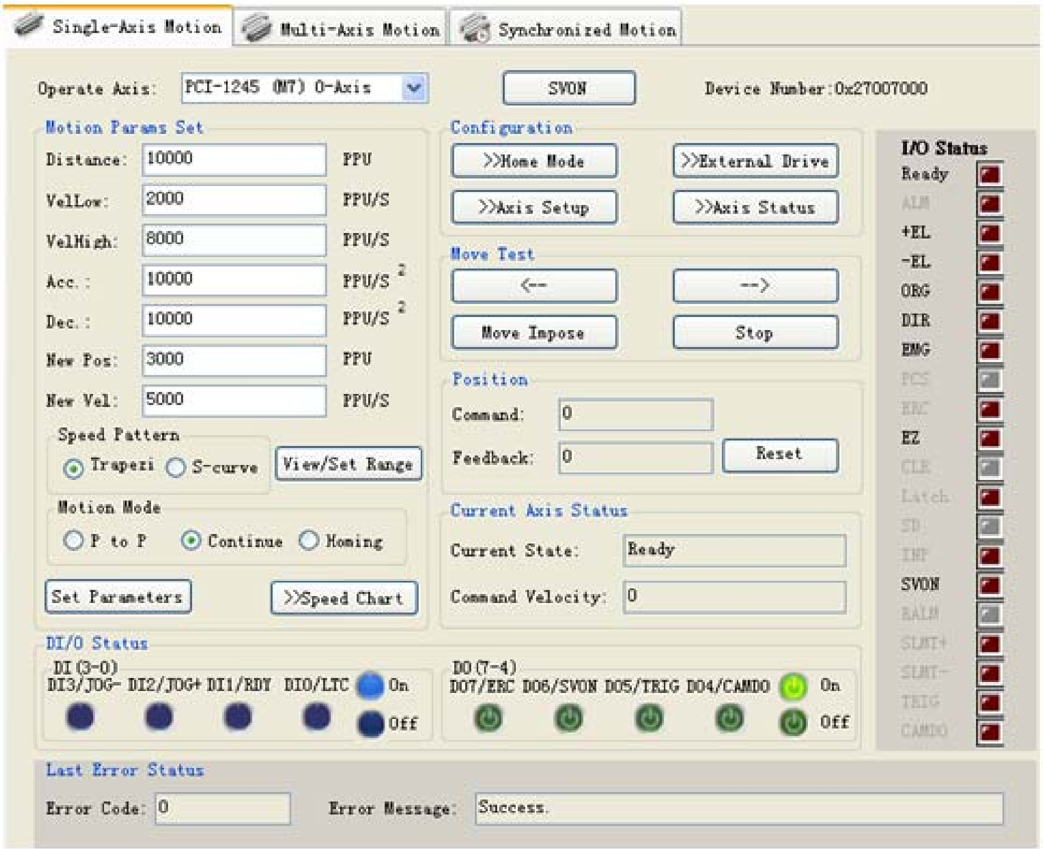
5.2.3 Device Tree



My System
- PCI-1220 (M15)
- PCI-1245 (M0)
- PCI-1245 (M15)

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

5.3 Single-Axis Motion



Operate Axis: PCI-1245 (M7) 0-Axis

Device Number:0x27007000

Distance:	10000	PPU
VelLow:	2000	PPU/S
VelHigh:	8000	PPU/S
Acc.:	10000	PPU/S ^{2}
Dec.:	10000	PPU/S ^{2}
New Pos:	3000	PPU
New Vel:	5000	PPU/S

Speed Pattern: ● Trapezi ○ S-curve

Motion Mode: ○ P to P ● Continue ○ Homing

Command: 0

Feedback: 0

Current State: Ready

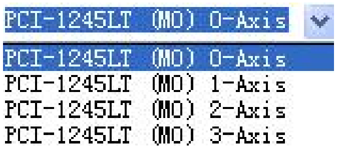
Command Velocity: 0

Error Code: 0

Error Message: Success.

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:



PCI-1245LT (M0) 0-Axis

- PCI-1245LT (M0) 0-Axis
- PCI-1245LT (M0) 1-Axis
- PCI-1245LT (M0) 2-Axis
- PCI-1245LT (M0) 3-Axis

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

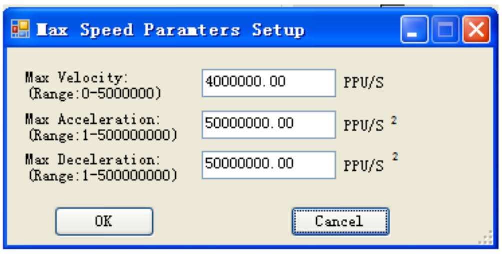
It's mainly about the settings of distance(Distance) in point to point movement, initial velocity (VelLow), movement velocity (VelHigh), acceleration (Acc.) and deceleration (Dec.) in single-axis motion, movement distance (New Pos.) and velocity (New Vel) in superimposed movement (Move Impose).

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.



Max Speed Paramters Setup

Max Velocity: (Range:0-5000000) 4000000.00 PPU/S

Max Acceleration: (Range:1-500000000) 50000000.00 PPU/S 2

Max Deceleration: (Range:1-500000000) 50000000.00 PPU/S 2

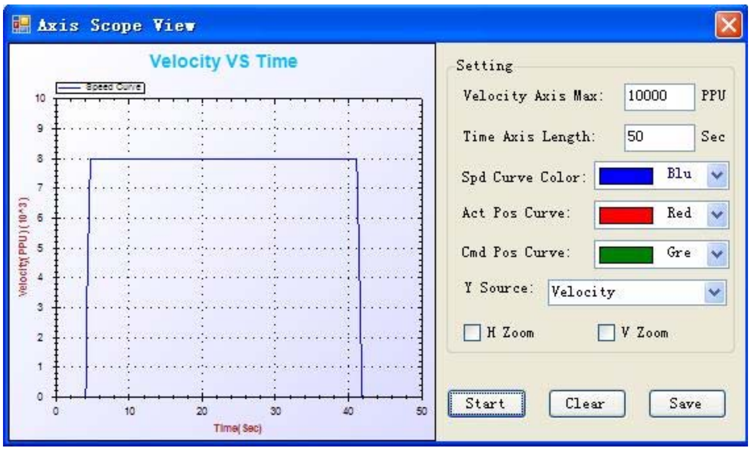
Note! *VelHigh in Single-axis Motion can not be greater than the Max Velocity; Acc. can not be greater than the Max Acceleration and Dec. can not be greater than the Max Deceleration.*

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.



Setting

Velocity Axis Max: 10000 PPU

Time Axis Length: 50 Sec

Spd Curve Color: Blu

Act Pos Curve: Red

Cmd Pos Curve: Gre

Y Source: Velocity

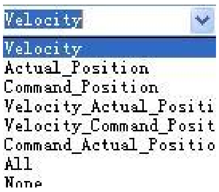
□ H Zoom □ V Zoom

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

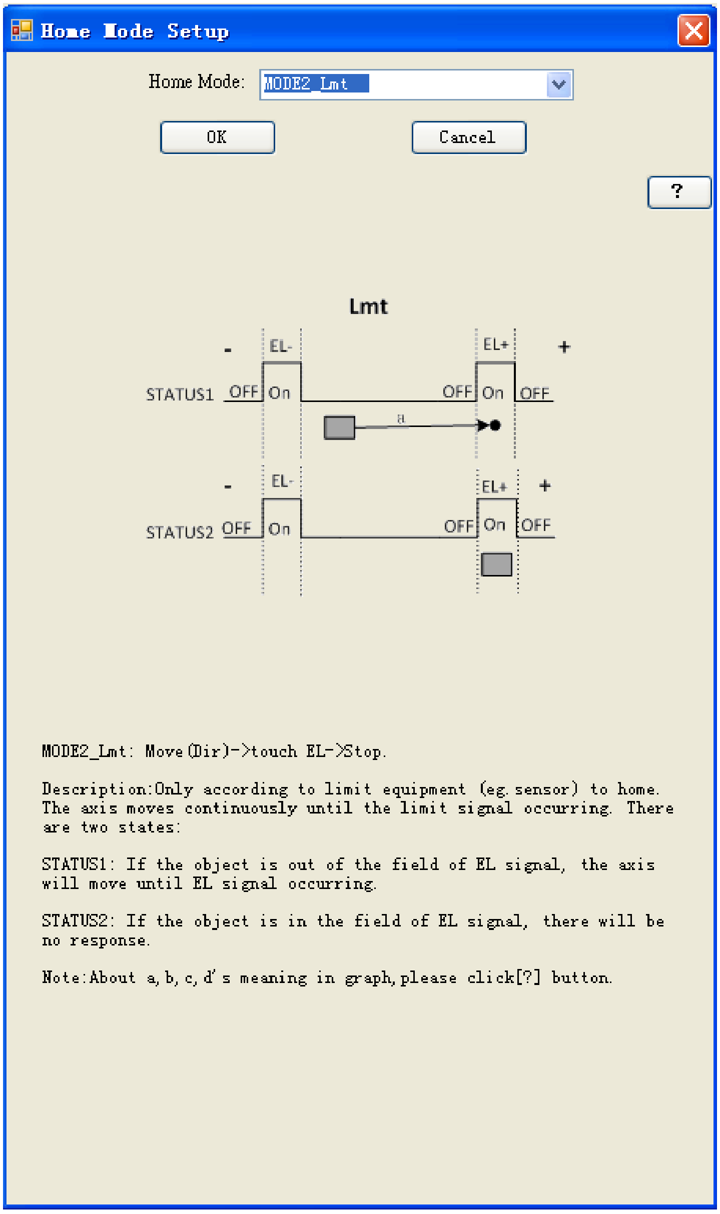
It includes Home Mode configuration, External Drive mode, the property configuration and I/O status of the axis.

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:



Home Mode: MODE2_Lmt

MODE2_Lmt: Move(Dir)->touch EL->Stop.

Description:Only according to limit equipment (eg.sensor) to home. The axis moves continuously until the limit signal occurring. There are two states:

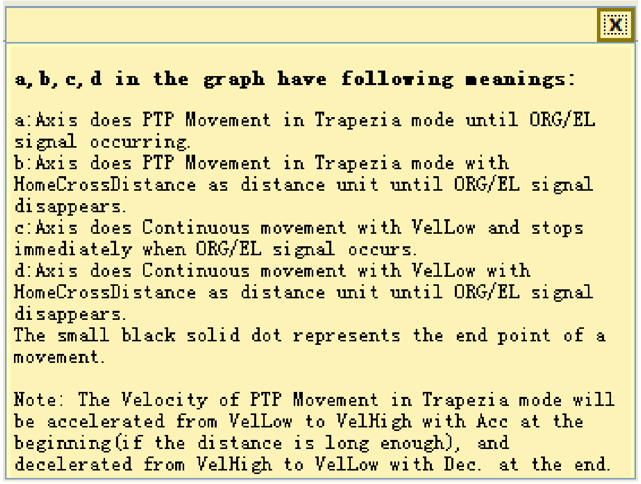
STATUS1: If the object is out of the field of EL signal, the axis will move until EL signal occurring.

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

Note:About a,b,c,d's meaning in graph,please click[?] button.

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:



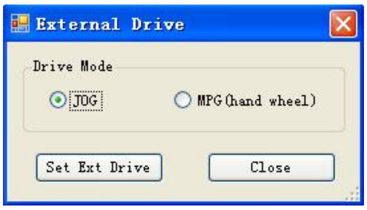
a,b,c,d in the graph have following meanings:

a:Axis does PTP Movement in Trapezia mode until ORG/EL signal occurring.
b:Axis does PTP Movement in Trapezia mode with HomeCrossDistance as distance unit until ORG/EL signal disappears.
c:Axis does Continuous movement with VelLow and stops immediately when ORG/EL signal occurs.
d:Axis does Continuous movement with VelLow with HomeCrossDistance as distance unit until ORG/EL signal disappears.
The small black solid dot represents the end point of a movement.

Note: The Velocity of PTP Movement in Trapezia mode will be accelerated from VelLow to VelHigh with Acc at the beginning(if the distance is long enough), and decelerated from VelHigh to VelLow with Dec. at the end.

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.



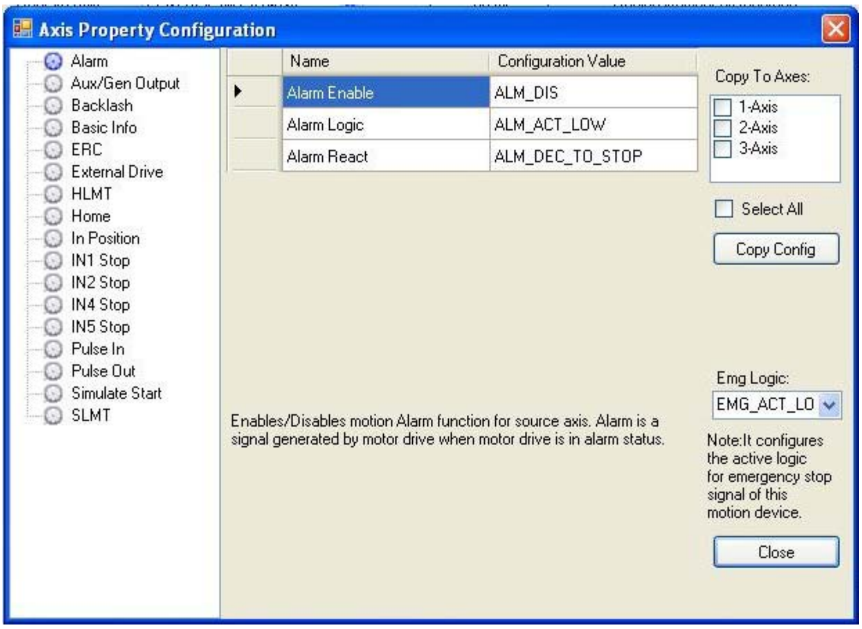
Drive Mode
● JOG ○ MPG(hand wheel)

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-1245L, 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:



Axis Property Configuration

	Name	Configuration Value
▶	Alarm Enable	ALM_DIS
	Alarm Logic	ALM_ACT_LOW
	Alarm React	ALM_DEC_TO_STOP

Enables/Disables motion Alarm function for source axis. Alarm is a signal generated by motor drive when motor drive is in alarm status.

Emg Logic: EMG_ACT_LO

Note:It configures the active logic for emergency stop signal of this motion device.

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.

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.
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.
Simulate Start	Simulate Start Source	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.

Note! *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-1245L, 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! *When "Pulse Out" category is selected, there will be illustration of corresponding mode below the description of "Pulse Out Mode" property.*

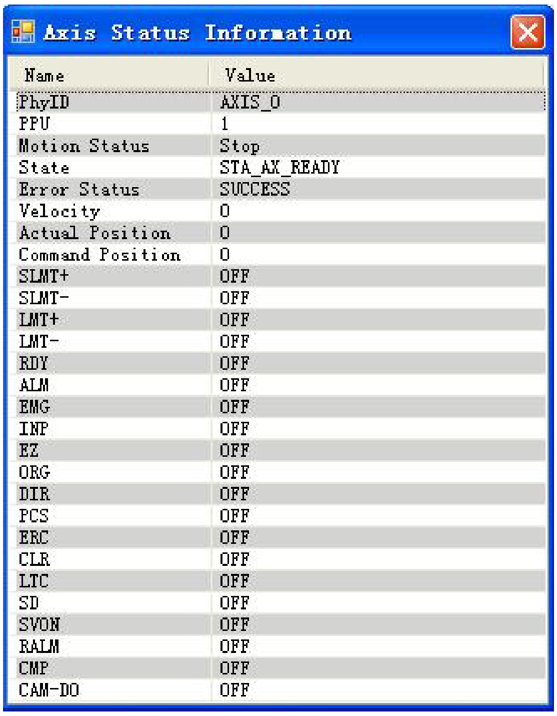
After editing, the property value will become effective (already set in device) after the mouse leaves the edit box.

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

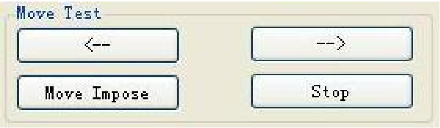
Click the button; you can view the assigned axis information. For example, PhyID, PPU, and basic status (Motion Status, State, Error Status and etc.) and I/O status (Alarm, SLMTP/N and etc.).


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:

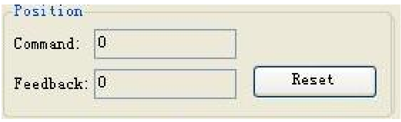


After motion mode is selected, click [<--] or [-->], the axis will do P to P/Continue/ Homing movement in negative or positive direction.

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



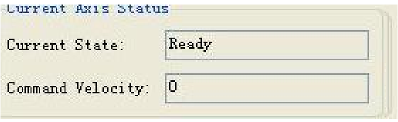
Command: 0

Feedback: 0

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



Current Axis Status

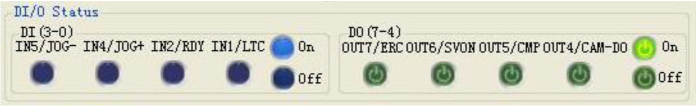
Current State: Ready

Command Velocity: 0

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.



DI/O Status

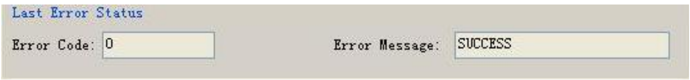
5.3.8.1 DI

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



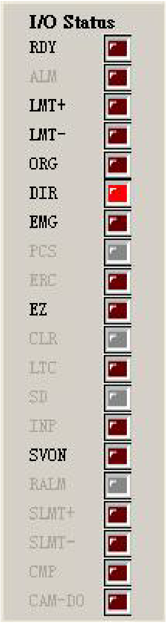
Last Error Status

Error Code: 0

Error Message: SUCCESS

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

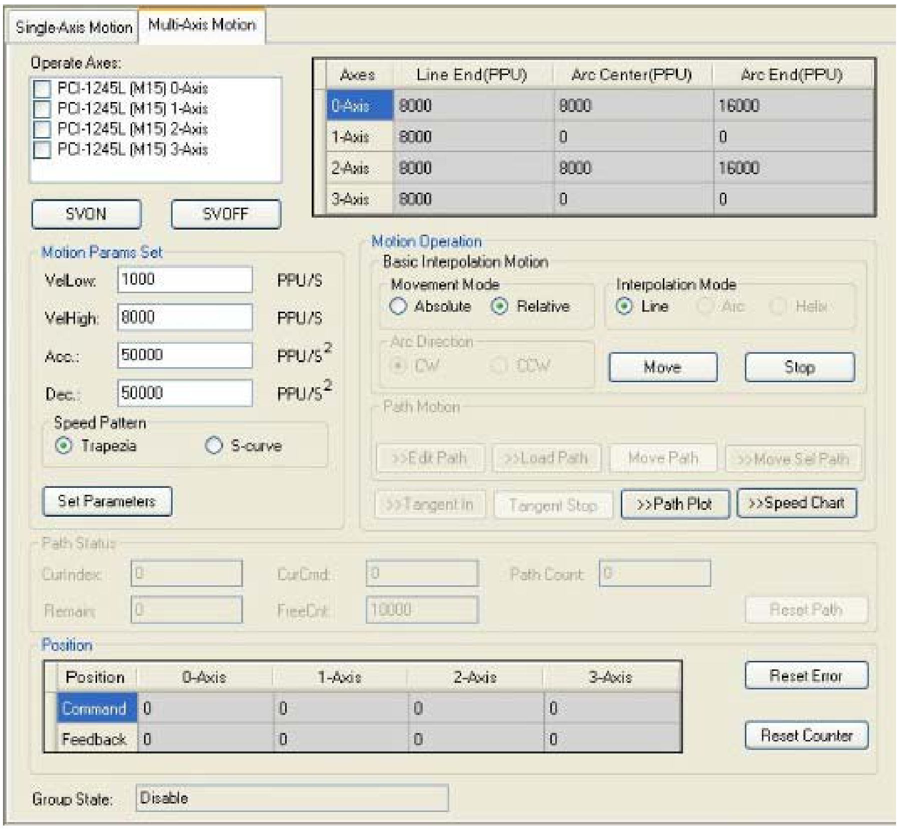


You can visually know the I/O status from the LED bar. Wherein, ▨ indicates the device does not support the function or does not have the corresponding I/O; ■ indicates the device support the function, but I/O is not triggered (OFF); 🟥 indicates the corresponding I/O is triggered (ON).

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



Operate Axes:

- □ PCI-1245L (M15) 0-Axis
- □ PCI-1245L (M15) 1-Axis
- □ PCI-1245L (M15) 2-Axis
- □ PCI-1245L (M15) 3-Axis

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

Motion Params Set

VelLow:	1000	PPU/S
VelHigh:	8000	PPU/S
Acc.:	50000	PPU/S^2
Dec.:	50000	PPU/S^2

Speed Pattern: ● Trapezia ○ S-curve

Motion Operation

Basic Interpolation Motion

Movement Mode: ○ Absolute ● Relative

Interpolation Mode: ● Line ○ Arc ○ Helix

Arc Direction: ● CW ○ CCW

Path Motion

Path Status

CurIndex:	0	CurCmd:	0	Path Count:	0
Remain:	0	FreeCnt:	10000		

Position

Position	0-Axis	1-Axis	2-Axis	3-Axis
Command	0	0	0	0
Feedback	0	0	0	0

Group State: Disable

5.4.1 Operate Axes

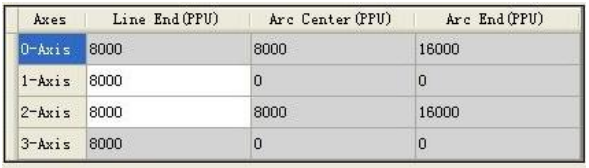
The checkedListBox in the form will list all axes of the selected device, check the Checkbox of corresponding axis, you can add the axis into the Group. When the number of axis added to the Group is less than 2, Group's State will be "Disable". When the number of axis added to the Group is greater than 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

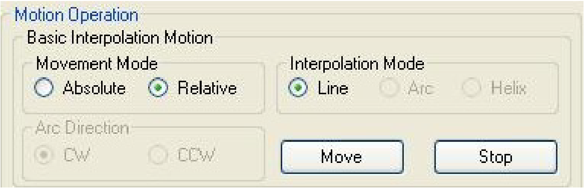
Click [SVON], the servos of axes in Group will be turned on.

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).



Motion Operation

Basic Interpolation Motion

Movement Mode: ○ Absolute ● Relative

Interpolation Mode: ● Line ○ Arc ○ Helix

Arc Direction: ● CW ○ CCW

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 Interpolation Mode

Line: linear interpolation

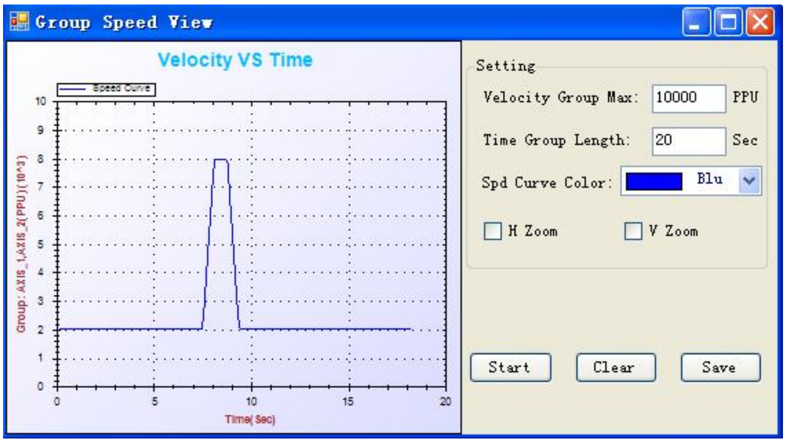
5.4.4.3.3 Move

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

5.4.4.3.4 Stop

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

5.4.4.4 Speed Chart



Velocity Group Max: 10000 PPU

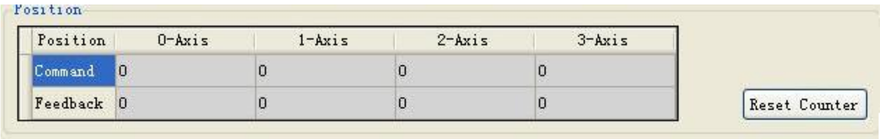
Time Group Length: 20 Sec

Spd Curve Color: Blu

□ H Zoom □ V Zoom

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

5.4.5 Position



Position	0-Axis	1-Axis	2-Axis	3-Axis
Command	0	0	0	0
Feedback	0	0	0	0

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

Click [Reset Counter] to reset to 0.

5.4.6 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.

Chapter 6

Programming Guide

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

6.1 Introduction

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

PCI-1245L 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 (Single 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	FLOAT	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	FLOAT *	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.

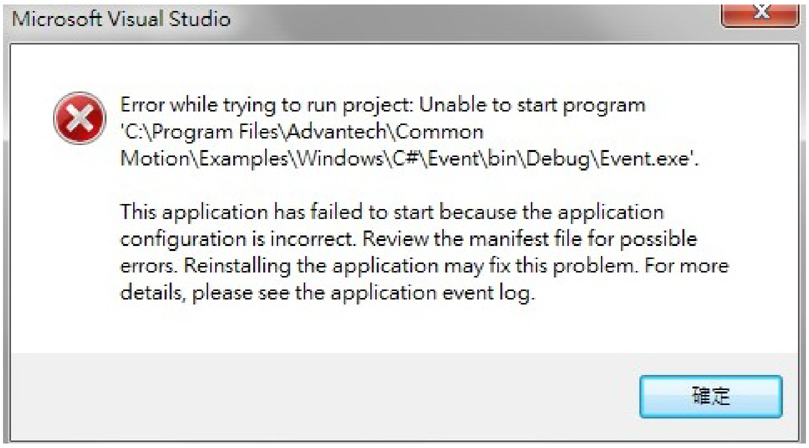
There are seven types of event:

Event Name	Description
EVT_AX_MOTION_DONE	Trigger event when current motion is done.
EVT_AX_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 corresponding 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:

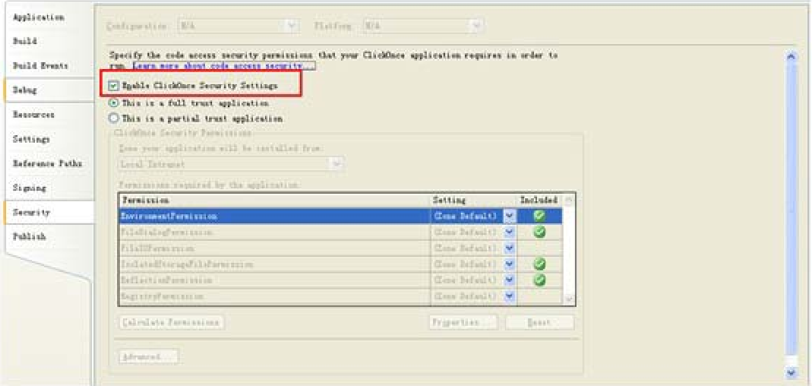


Microsoft Visual Studio

Error while trying to run project: Unable to start program 'C:\Program Files\Advantech\Common Motion\Examples\Windows\C#\Event\bin\Debug\Event.exe'.

This application has failed to start because the application configuration is incorrect. Review the manifest file for possible errors. Reinstalling the application may fix this problem. For more details, please see the application event log.

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



Configuration: N/A Platform: N/A

Specify the code access security permissions that your ClickOnce application requires in order to run. Learn more about code access security...

☑ Enable ClickOnce Security Settings

● This is a full trust application

○ This is a partial trust application

ClickOnce Security Permissions

Zone your application will be installed from:

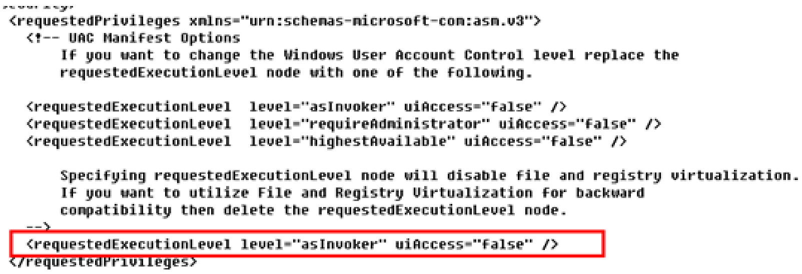
Local Intranet

Permissions required by the application:

Permission	Setting	Included
EnvironmentPermission	(Zone Default)	✓
FileDialogPermission	(Zone Default)	✓
FileIOPermission	(Zone Default)	
IsolatedStorageFilePermission	(Zone Default)	✓
ReflectionPermission	(Zone Default)	✓
RegistryPermission	(Zone Default)	

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-1245L Software architecture

The PCI-1245L software architecture based on Common Motion Architecture is as follows:

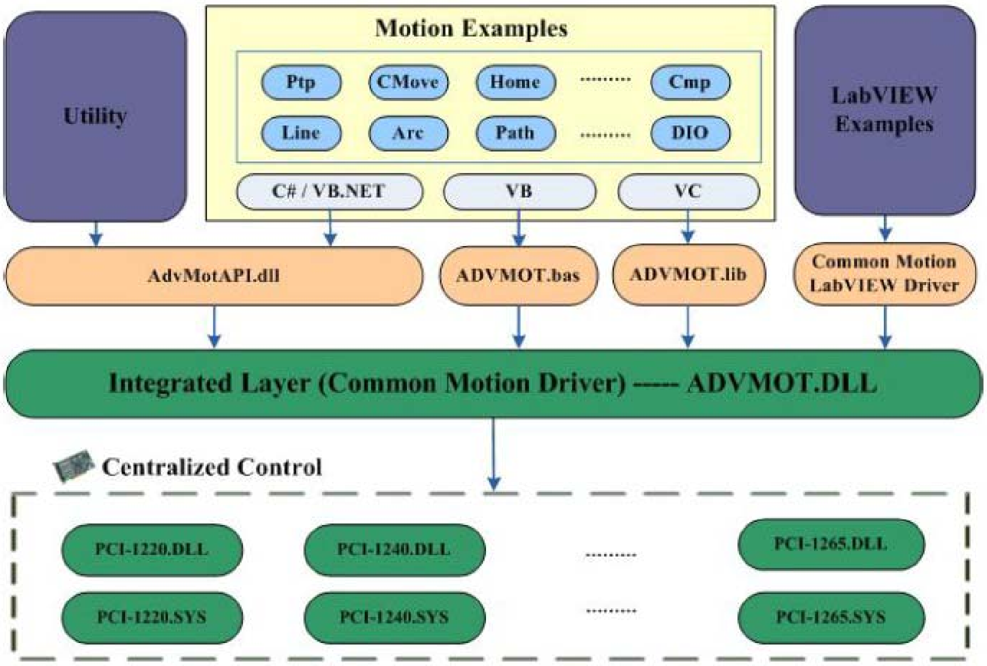


Figure 6.1 PCI-1245L Software Architecture

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

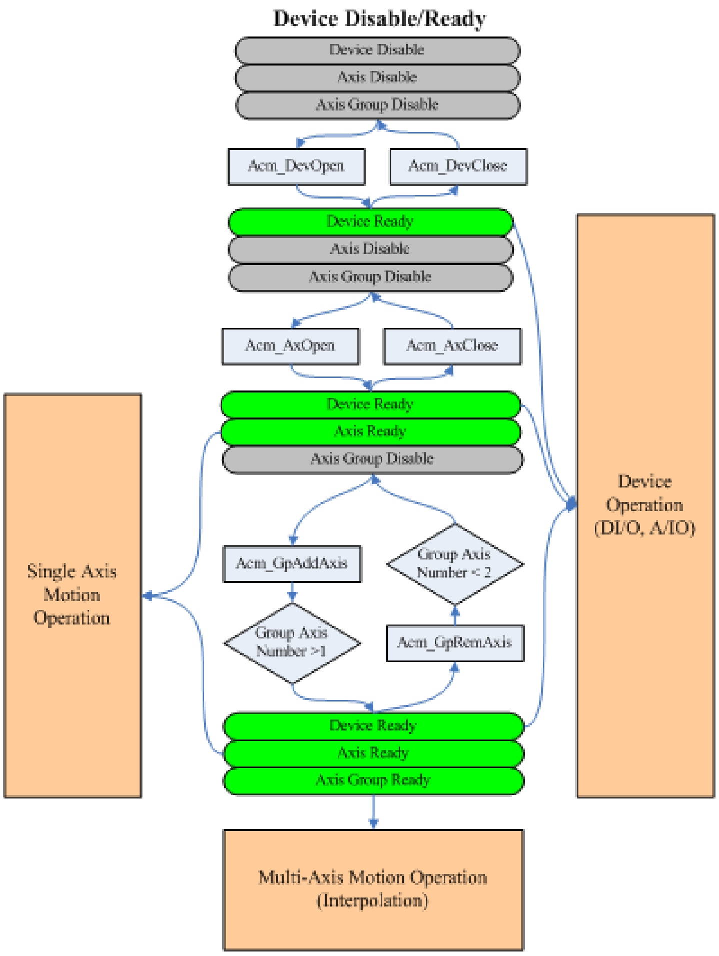


Figure 6.2 Basic Operation Flow Chart

6.2.2.2 Single Axis Flow

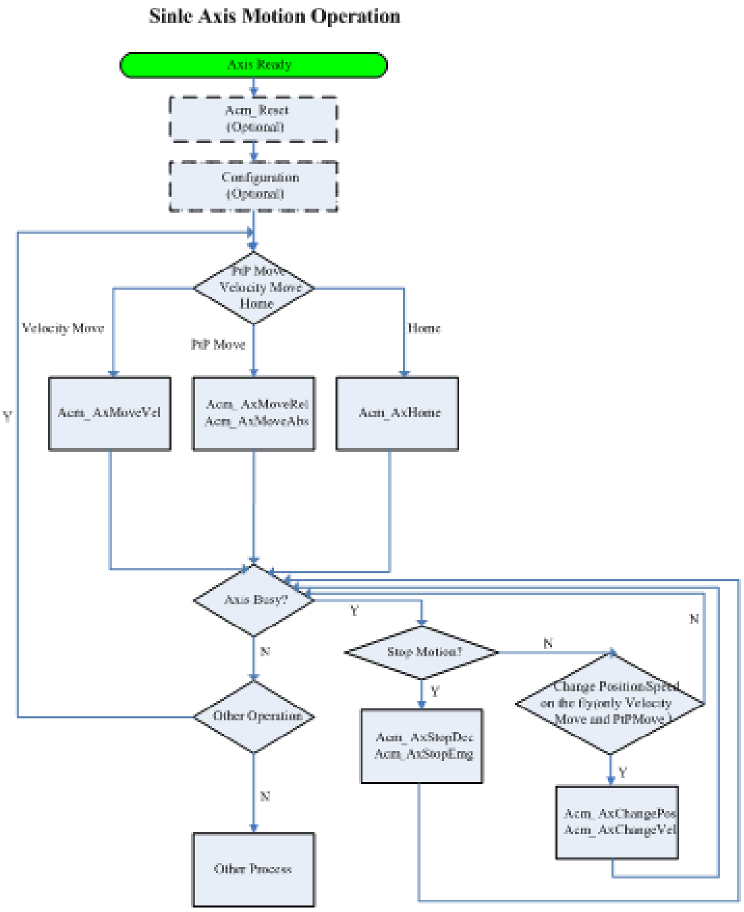


Figure 6.3 Single Axis Operation Flow Chart

6.2.2.3 Multiple Axis Flow Chart

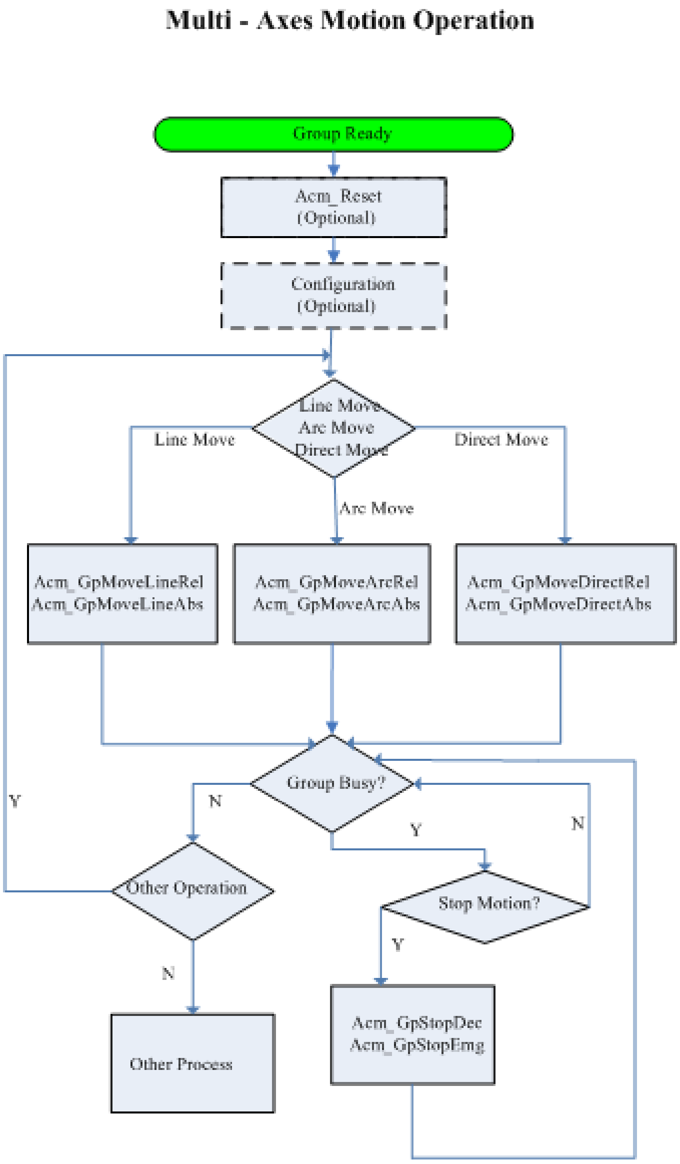


Figure 6.4 Multiple Axis Operation Flow Chart

6.2.3 Example Support List

Example	VC	C#	VB	VB .NET	Description
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.
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 linemotion.
MPG_JOG	√	√		√	Demonstrates how to start external drive operation on the specified device and axis.
PTP	√	√	√	√	Demonstrates how to control one axis point to point motion
SimulateOpe	√	√		√	Demonstrates how to control simultaneous movement between multi-axis.
Direct	√	√			Demonstrates how to control an interpolation group's direct motion.
DataProtect	√	√	√	√	read/write private data

6.2.4 PCI-1245L Support API List

Type		Method/Event	PCI-1245L	Description
Device 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 event happened.
		Acm_EnableMotionEvent	√	Enable/disable event.
	Event	EVT_AX_MOTION_DONE	√	Event happens when axis motion is done.
		EVT_AX_ERROR	√	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.
	Private data read & write	Acm_DevReadEEPROM_Ex	√	Read EEPROM private data.
		Acm_DevWriteEEPROM_Ex	√	Write EEPROM private data.

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.
	Simultaneous Motion	Acm_AxSimStartSuspendAbs	√	Suspend absolute simultaneous motion.
		Acm_AxSimStartSuspendRel	√	Suspend relative simultaneous motion.
		Acm_AxSimStartSuspendVel	√	Suspend continuous motion.
		Acm_AxSimStart	√	Start simultaneous motion.
		Acm_AxSimStop	√	Stop simultaneous motion.
	Home	Acm_AxHome	√	Command home.
	Position/ Counter	Acm_AxSetCmdPosition	√	Set command position.
		Acm_AxGetCmdPosition	√	Get command position.
		Acm_AxSetActualPosition	√	Set actual position.
		Acm_AxGetActualPosition	√	Get actual position.
	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.

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_GpGetCmdVel	√	Get current velocity of the group.
	Motion Stop	Acm_GpStopDec	√	Decelerated stop.
		Acm_GpStopEmg	√	Emergency stop.
	Interpolation Motion	Acm_GpMoveLinearRel	√	Command relative linear interpolation.
		Acm_GpMoveLinearAbs	√	Command absolute linear interpolation.
		Acm_GpMoveDirectAbs	√	Command absolute direct linear interpolation.
		Acm_GpMoveDirectRel	√	Command relative direct linear interpolation.

6.2.5 Property Support List

Type		Property	PCI-1245L
Device	Feature	FT_DevIpoTypeMap	√
		FT_DevAxesCount	√
		FT_DevFunctionMap	√
		FT_DevOverflowCntr	√
	Configure	CFG_DevBoardID	√
		CFG_DevBaseAddress	√
		CFG_DevInterrupt	√
		CFG_DevBusNumber	√
		CFG_DevSlotNumber	√
		CFG_DevDriverVersion	√
		CFG_DevDllVersion	√
		CFG_DevFwVersion	√
		CFG_DevFPGA_1Version	√
		CFG_DevEmgLogic	√

Axis	System	FT_AxFunctionMap	√
		CFG_AxPPU	√
		CFG_AxPhyID	√
	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_AxAlmEnable	√
		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_AxOrgReact	√
		CFG_AxEzLogic	√
		CFG_AxHomeResetEnable	√
		PAR_AxHomeCrossDistance	√
		PAR_AxHomeExSwitchMode	√
	BackLash	FT_AxBacklashMap	√
		CFG_AxBacklashEnable	√
		CFG_AxBacklashPulses	√
		CFG_AxBacklashVel	√

Axis	Aux/Gen DIO	FT_AxGenDOMap	√
		FT_AxGenDIMap	√
		CFG_AxGenDoEnable	√
	Ext-Drive	FT_AxExtDriveMap	√
		FT_AxExtMasterSrcMap	√
		CFG_AxExtMasterSrc	√
		CFG_AxExtPulseNum	√
		CFG_AxExtPulseInMode	√
		CFG_AxExtPresetNum	√
	Simultaneity	FT_AxSimStartSourceMap	√
		CFG_AxSimStartSource	√
	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_AxIN5StopLogic	√
Group	System	PAR_GpGroupID	√
		CFG_GpAxesInGroup	√
	Speed Pattern	PAR_GpVelLow	√
		PAR_GpVelHigh	√
		PAR_GpAcc	√
		PAR_GpDec	√
		PAR_GpJerk	√

6.2.6 Creating a New Application

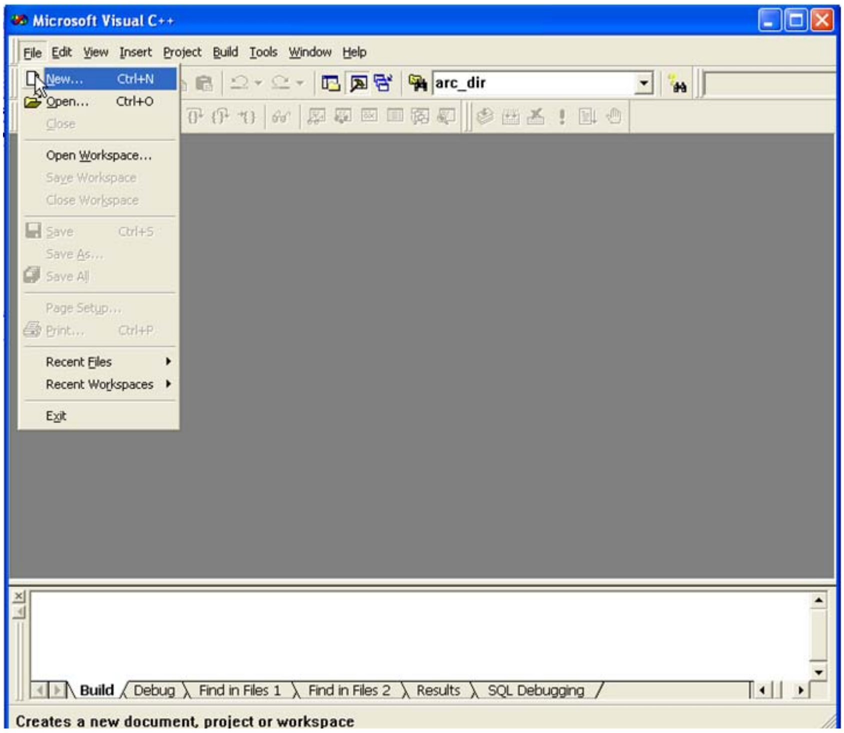
For creating a new application under PCI-1245L, user ought to install Common Motion Examples, there are many examples developed in different language in folder Advantech\Motion Common\Examples, user can follow these examples to develop a new application.

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:

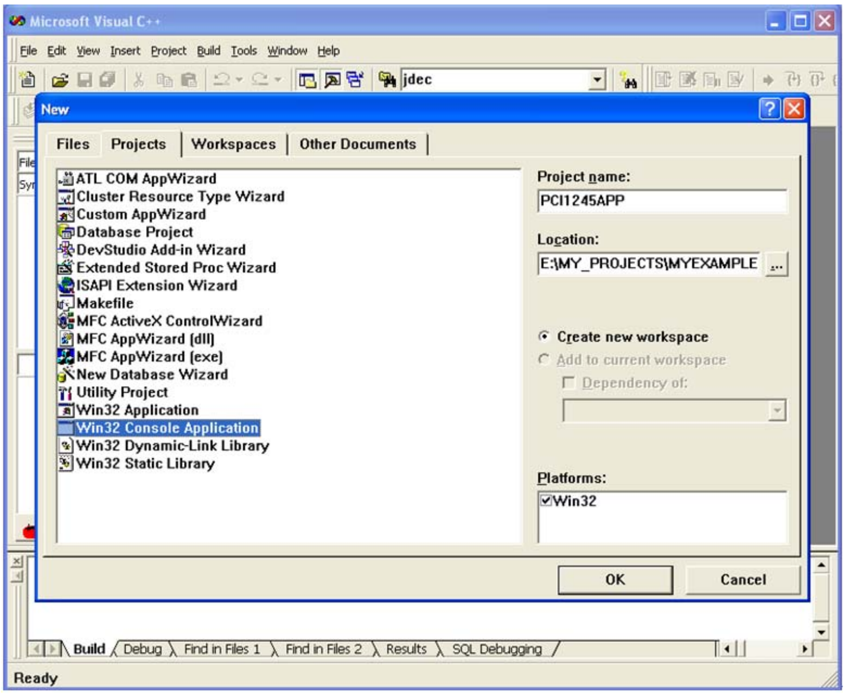
1. Click **File/New** from the main menu to create your application project and source code as you would for any other Visual C++ program.



Creates a new document, project or workspace

Figure 6.5 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.


Microsoft Visual C++
File Edit View Insert Project Build Tools Window Help
jdec
New
Files Projects Workspaces Other Documents
ATL COM AppWizard
Cluster Resource Type Wizard
Custom AppWizard
Database Project
DevStudio Add-in Wizard
Extended Stored Proc Wizard
ISAPI Extension Wizard
Makefile
MFC ActiveX ControlWizard
MFC AppWizard (dll)
MFC AppWizard (exe)
New Database Wizard
Utility Project
Win32 Application
Win32 Console Application
Win32 Dynamic-Link Library
Win32 Static Library
OK
Cancel
Build Debug Find in Files 1 Find in Files 2 Results SQL Debugging
Ready

Project name: PCI1245APP

Location: E:\MY_PROJECTS\MYEXAMPLE

● Create new workspace

○ Add to current workspace

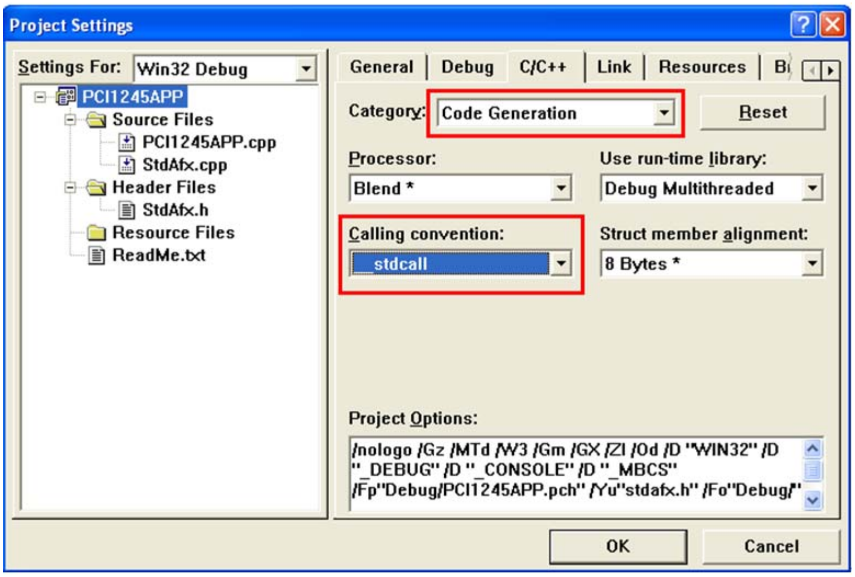
□ Dependency of:

Platforms: ☑ Win32

Figure 6.6 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:



Settings For: Win32 Debug

Category: Code Generation

Processor: Blend *

Use run-time library: Debug Multithreaded

Calling convention: __stdcall

Struct member alignment: 8 Bytes *

Project Options:

```
/nologo /Gz /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D
"_DEBUG" /D "_CONSOLE" /D "_MBCS"
/Fp"Debug/PCI1245APP.pch" /Yu"stdafx.h" /Fo"Debug/"
```

Figure 6.7 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.

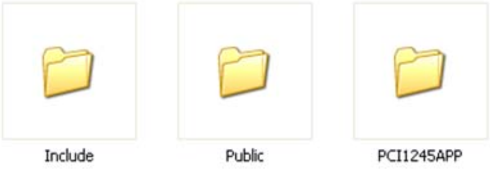
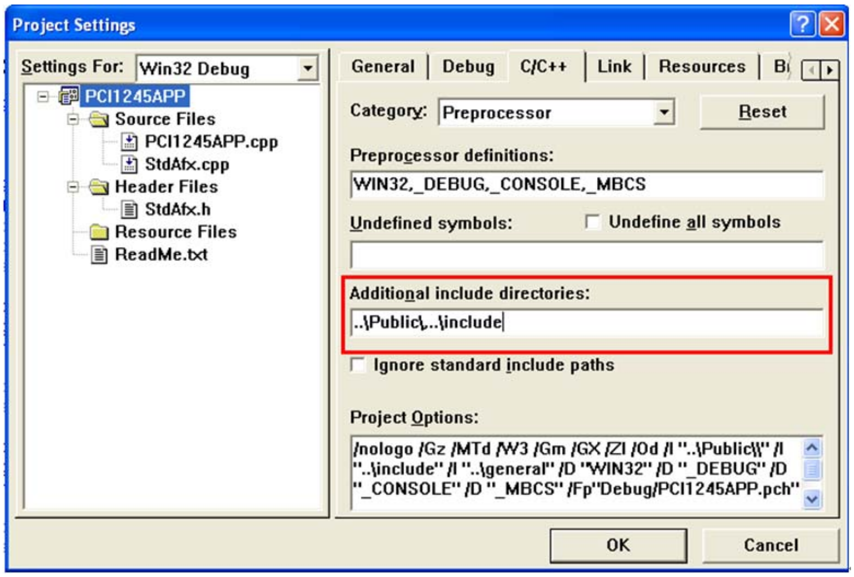


Figure 6.8 Folder Content of This Example

So the path setting is as follow.



Project Settings

Settings For: Win32 Debug

- PCI1245APP
  - Source Files
    - PCI1245APP.cpp
    - StdAfx.cpp
  - Header Files
    - StdAfx.h
  - Resource Files
  - ReadMe.txt

Category: Preprocessor

Preprocessor definitions:

WIN32,_DEBUG,_CONSOLE,_MBCS

Undefined symbols: ☐ Undefine all symbols

Additional include directories:

..\Public\,..\include

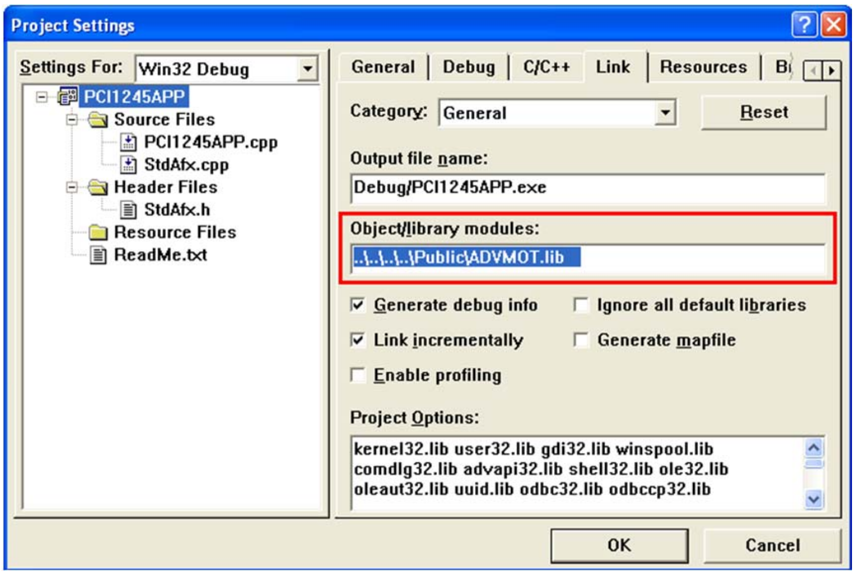
☐ Ignore standard include paths

Project Options:

```
/nologo /Gz /MTd /W3 /Gm /GX /ZI /Od /I "..\Public\\" /I
"..\include" /I "..\general" /D "WIN32" /D "_DEBUG" /D
"_CONSOLE" /D "_MBCS" /Fp"Debug/PCI1245APP.pch"
```

Figure 6.9 Add Head Files Path

c. Set the necessary Lib file.
The Lib file “ADVMOT.lib” which is corresponding to “ADVMOT.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.



Project Settings

Settings For: Win32 Debug

- PCI1245APP
  - Source Files
    - PCI1245APP.cpp
    - StdAfx.cpp
  - Header Files
    - StdAfx.h
  - Resource Files
  - ReadMe.txt

Category: General

Output file name:

Debug/PCI1245APP.exe

Object/library modules:

..\..\..\..\Public\ADVMOT.lib

☑ Generate debug info ☐ Ignore all default libraries

☑ Link incrementally ☐ Generate mapfile

☐ Enable profiling

Project Options:

```
kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib
oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
```

Figure 6.10 Setting Lib File Path

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

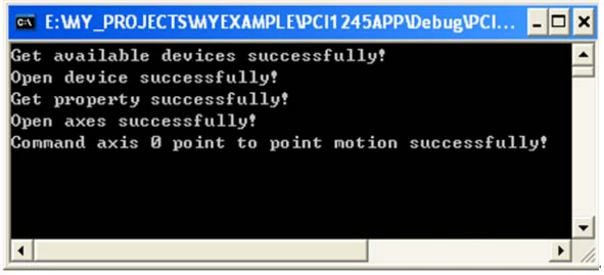
4. Write the code.

```
#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, axisCntPerDev, &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.



```
Get available devices successfully!
Open device successfully!
Get property successfully!
Open axes successfully!
Command axis 0 point to point motion successfully!
```

Figure 6.11 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:

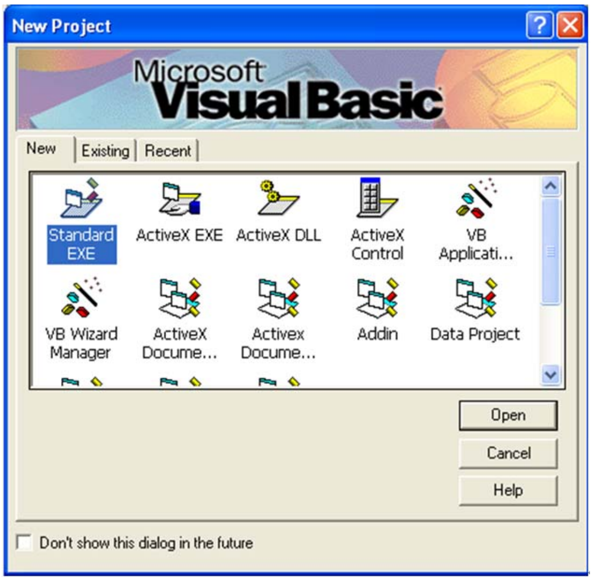


Figure 6.12 Load VB Development Environment

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.

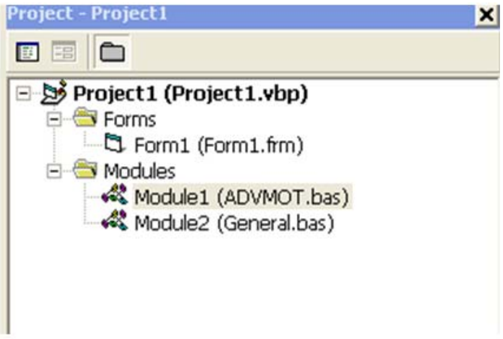


Figure 6.13 Add Module Files into Project

4. Design the form.

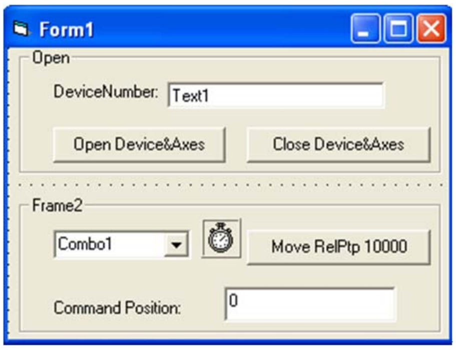


Figure 6.14 Design the Form

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
```

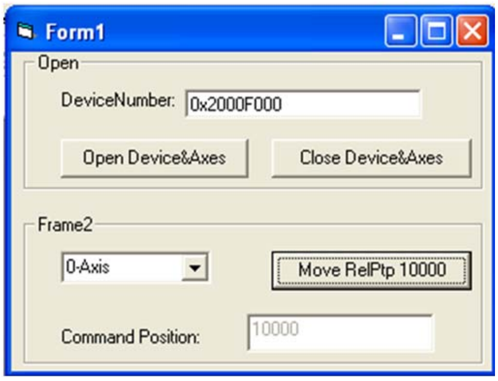
The timer is used to get the command position of selected axis. The code is as follow:

```
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), CurPos(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:



DeviceNumber: 0x2000F000

0-Axis

Command Position: 10000

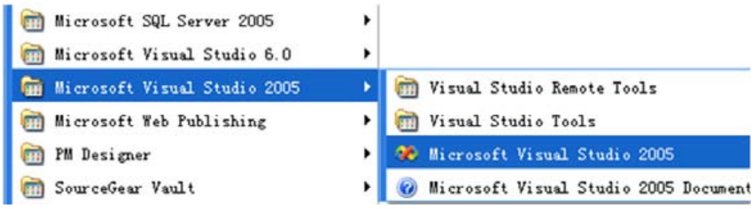
Figure 6.15 The Execution Result

6.2.6.3 Creating a New C# Application

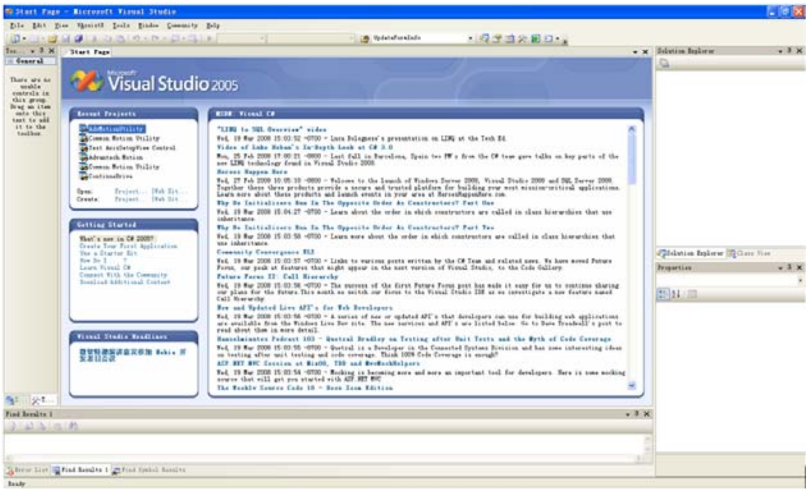
To use PCI-1245L SoftMotion PCI Controller, ADVMOT.dll and relevant driver files are needed. Be sure to install the driver before development.

Create a C# project as follows:

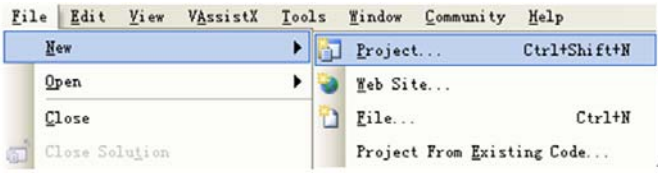
1. **Create a new project**
   Select [Microsoft Visual Studio 2005] from the Microsoft Visual Studio 2005 in Start Menu, as follows:


Microsoft SQL Server 2005
Microsoft Visual Studio 6.0
Microsoft Visual Studio 2005
Microsoft Web Publishing
PM Designer
SourceGear Vault
Visual Studio Remote Tools
Visual Studio Tools
Microsoft Visual Studio 2005
Microsoft Visual Studio 2005 Document

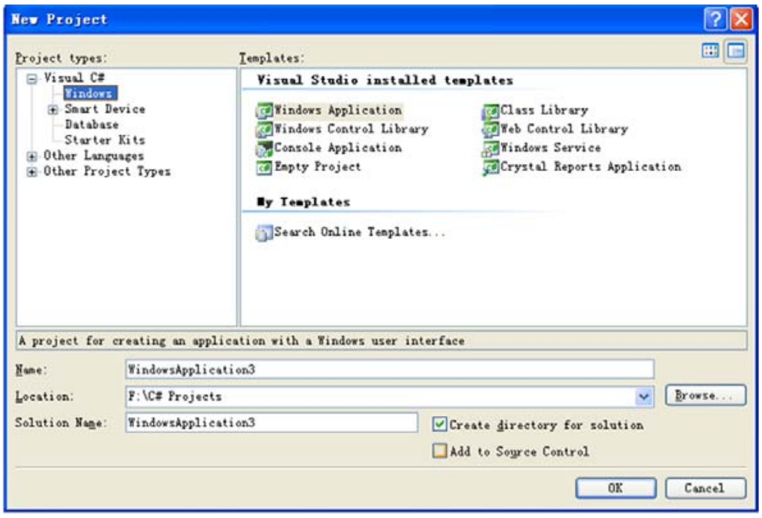
The development environment of Microsoft Visual Studio 2005 is as follows:



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].



A project for creating an application with a Windows user interface

Name: WindowsApplication3

Location: F:\C# Projects

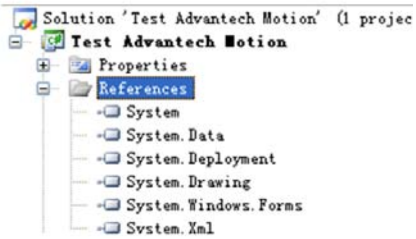
Solution Name: WindowsApplication3

☑ Create directory for solution

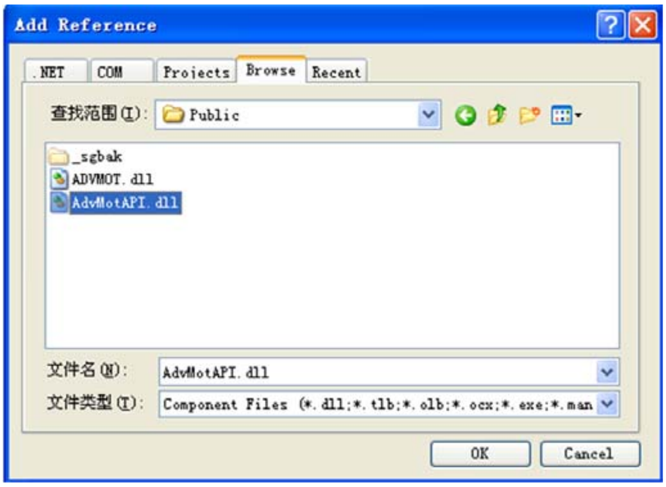
□ Add to Source Control

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:

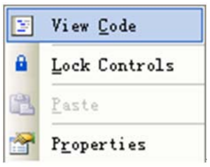


查找范围(I): Public

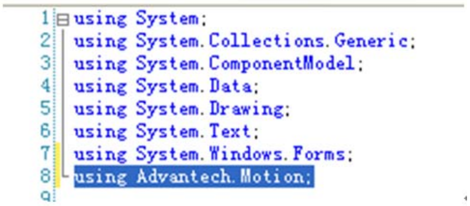
文件名(N): AdvMotAPI.dll

文件类型(T): Component Files (*.dll;*.tlb;*.olb;*.ocx;*.exe;*.man

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:

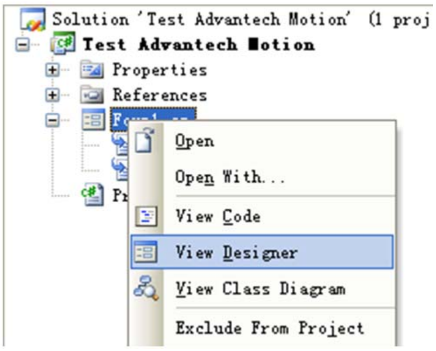


```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using Advantech.Motion;
```

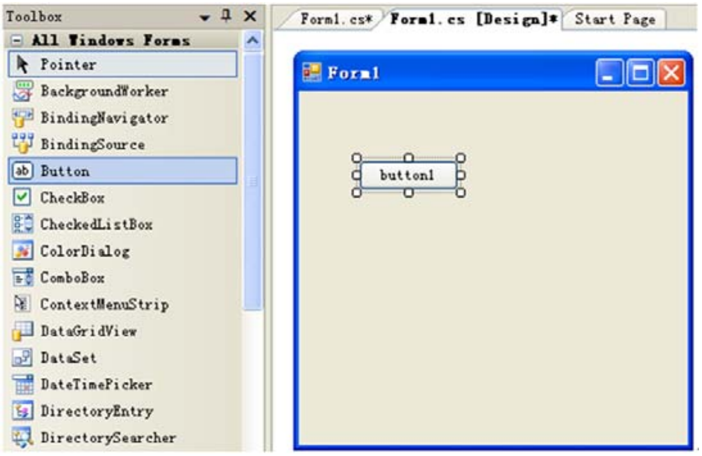
3. Coding

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:



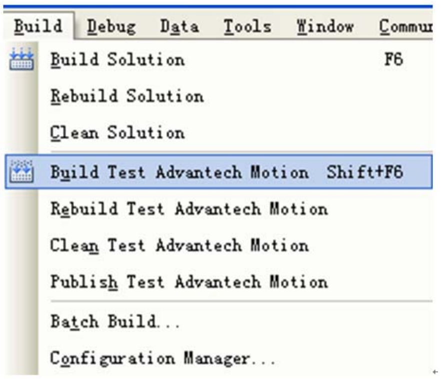
For detail, refer to Microsoft Visual C # user manual.

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-1245L SoftMotion PCI Controller.

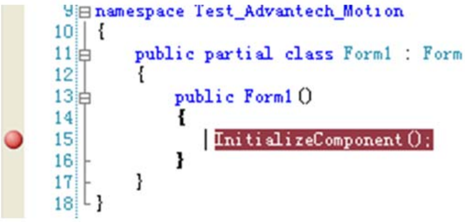
4. **Test program**

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:


Build
Debug
Data
Tools
Window
Commu
Build Solution F6
Rebuild Solution
Clean Solution
Build Test Advantech Motion Shift+F6
Rebuild Test Advantech Motion
Clean Test Advantech Motion
Publish Test Advantech Motion
Batch Build...
Configuration Manager...

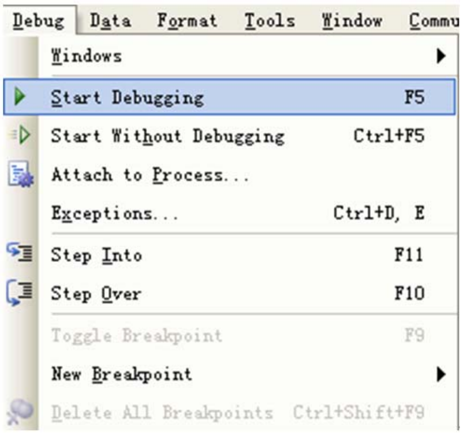
You can directly click ▶ in the toolbar, the program will run if there is no error.

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



```
namespace Test_Advantech_Motion
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

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



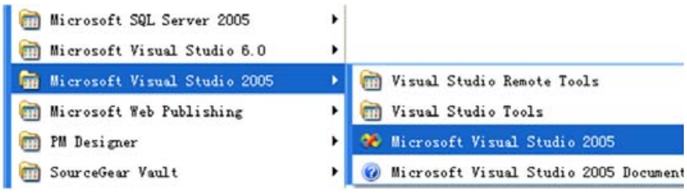
6.2.6.4 Creating a New VB.net Application

To use PCI-1245L SoftMotion PCI Controller, ADVMOT.dll and relevant driver files are needed. Be sure to install the driver before development.

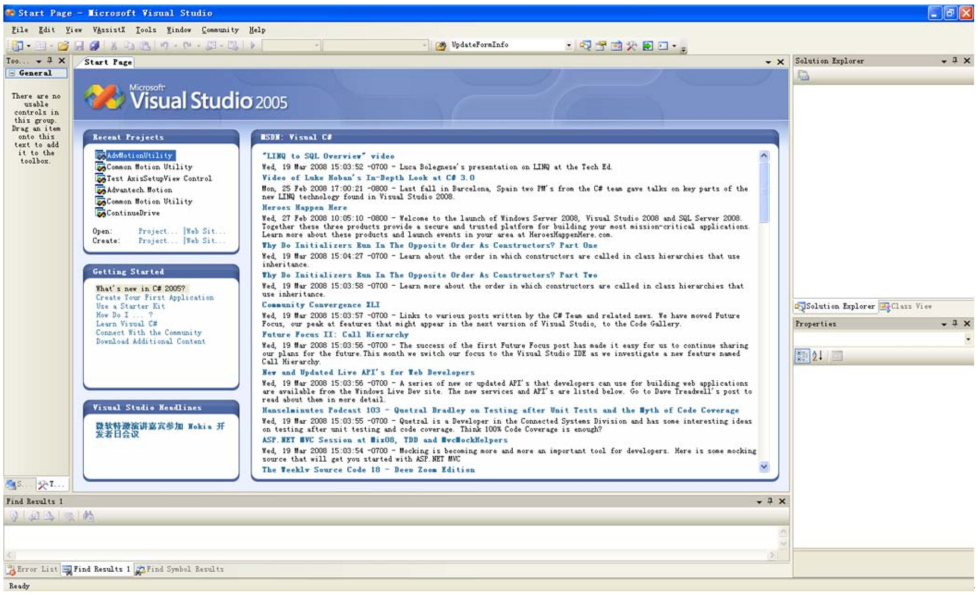
Create a Visual Basic project as follows:

1. **Create a new project**

   Select [Microsoft Visual Studio 2005] from the Microsoft Visual Studio 2005 in Start Menu, as follows:


Microsoft SQL Server 2005
Microsoft Visual Studio 6.0
Microsoft Visual Studio 2005
Microsoft Web Publishing
PM Designer
SourceGear Vault
Visual Studio Remote Tools
Visual Studio Tools
Microsoft Visual Studio 2005
Microsoft Visual Studio 2005 Document

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


Start Page - Microsoft Visual Studio
File Edit View VAssistX Tools Window Community Help
UpdateFormInfo
Too... General
There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.
Start Page
Microsoft Visual Studio 2005
Recent Projects
AdvMotionUtility
Common Motion Utility
Test AxisSetupView Control
Advantech Motion
Common Motion Utility
ContinueDrive
Open: Project... |Web Sit...
Create: Project... |Web Sit...
Getting Started
What's new in C# 2005?
Create Your First Application
Use a Starter Kit
How Do I ... ?
Learn Visual C#
Connect With the Community
Download Additional Content
Visual Studio Headlines
微软特邀演讲嘉宾参加 Nokia 开发者日会议
MSDN: Visual C#
Solution Explorer
Solution Explorer Class View
Properties
Find Results 1
Error List Find Results 1 Find Symbol Results
Ready

"LINQ to SQL Overview" video

Wed, 19 Mar 2008 15:03:52 -0700 - Luca Bolognese's presentation on LINQ at the Tech Ed.

Video of Luke Hoban's In-Depth Look at C# 3.0

Mon, 25 Feb 2008 17:00:21 -0800 - Last fall in Barcelona, Spain two PM's from the C# team gave talks on key parts of the new LINQ technology found in Visual Studio 2008.

Heroes Happen Here

Wed, 27 Feb 2008 10:05:10 -0800 - Welcome to the launch of Windows Server 2008, Visual Studio 2008 and SQL Server 2008. Together these three products provide a secure and trusted platform for building your most mission-critical applications. Learn more about these products and launch events in your area at HeroesHappenHere.com.

Why Do Initializers Run In The Opposite Order As Constructors? Part One

Wed, 19 Mar 2008 15:04:27 -0700 - Learn about the order in which constructors are called in class hierarchies that use inheritance.

Why Do Initializers Run In The Opposite Order As Constructors? Part Two

Wed, 19 Mar 2008 15:03:58 -0700 - Learn more about the order in which constructors are called in class hierarchies that use inheritance.

Community Convergence XLI

Wed, 19 Mar 2008 15:03:57 -0700 - Links to various posts written by the C# Team and related news. We have moved Future Focus, our peak at features that might appear in the next version of Visual Studio, to the Code Gallery.

Future Focus II: Call Hierarchy

Wed, 19 Mar 2008 15:03:56 -0700 - The success of the first Future Focus post has made it easy for us to continue sharing our plans for the future.This month we switch our focus to the Visual Studio IDE as we investigate a new feature named Call Hierarchy.

New and Updated Live API's for Web Developers

Wed, 19 Mar 2008 15:03:56 -0700 - A series of new or updated API's that developers can use for building web applications are available from the Windows Live Dev site. The new services and API's are listed below. Go to Dave Treadwell's post to read about them in more detail.

Hanselminutes Podcast 103 - Quetzal Bradley on Testing after Unit Tests and the Myth of Code Coverage

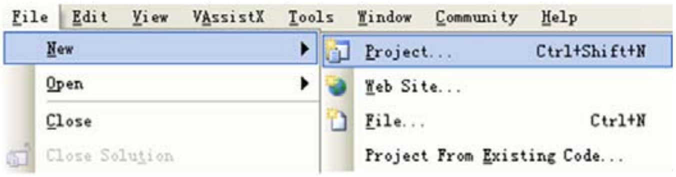
Wed, 19 Mar 2008 15:03:55 -0700 - Quetzal is a Developer in the Connected Systems Division and has some interesting ideas on testing after unit testing and code coverage. Think 100% Code Coverage is enough?

ASP.NET MVC Session at Mix08, TDD and MvcMockHelpers

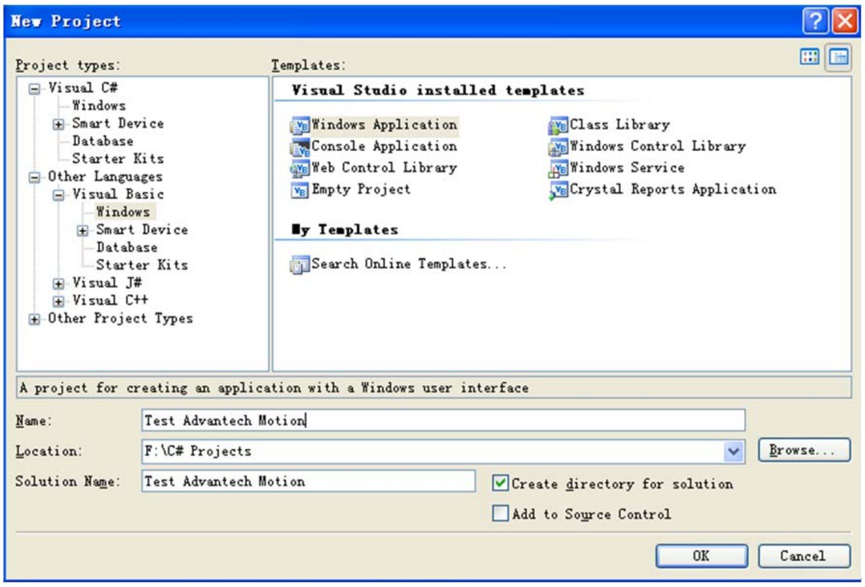
Wed, 19 Mar 2008 15:03:54 -0700 - Mocking is becoming more and more an important tool for developers. Here is some mocking source that will get you started with ASP.NET MVC

The Weekly Source Code 18 - Deep Zoom Edition

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


File Edit View VAssistX Tools Window Community Help
New
Open
Close
Close Solution
Project... Ctrl+Shift+N
Web Site...
File... Ctrl+N
Project From Existing Code...

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].



A project for creating an application with a Windows user interface

Name: Test Advantech Motion

Location: F:\C# Projects

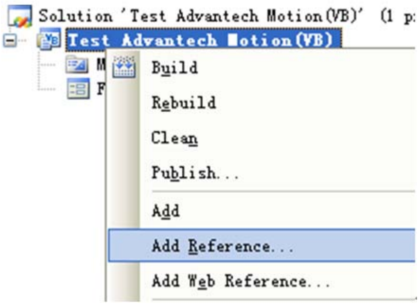
Solution Name: Test Advantech Motion

☑ Create directory for solution

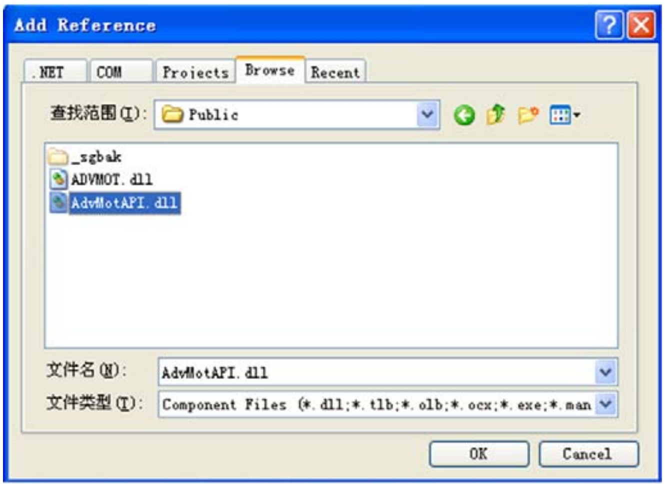
□ Add to Source Control

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:

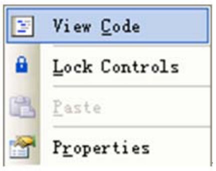


查找范围(I): Public

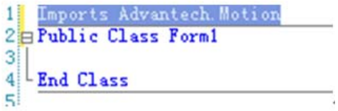
文件名(N): AdvMotAPI.dll

文件类型(T): Component Files (*.dll;*.tlb;*.olb;*.ocx;*.exe;*.man

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:



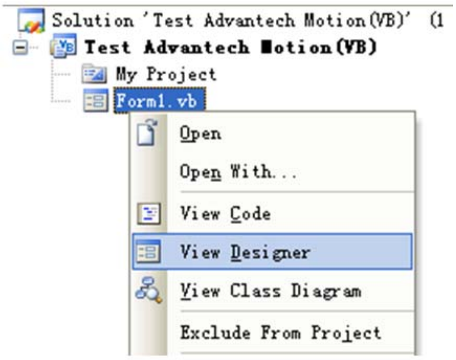
```
Imports Advantech.Motion
Public Class Form1

End Class
```

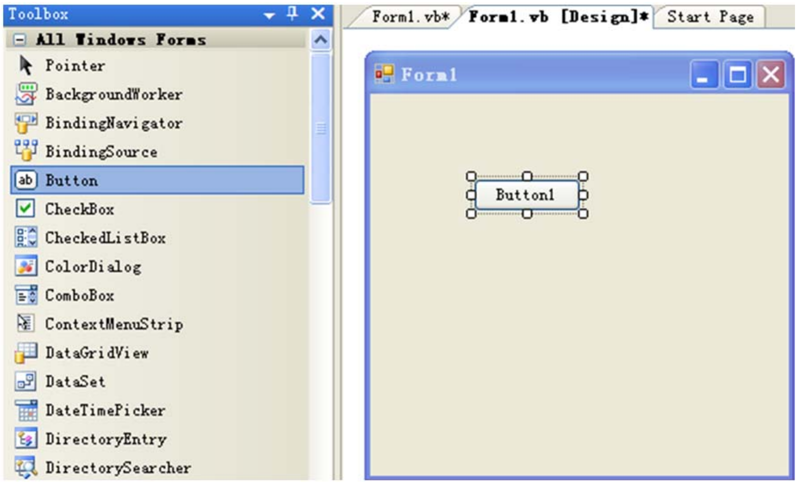
3. Coding

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:



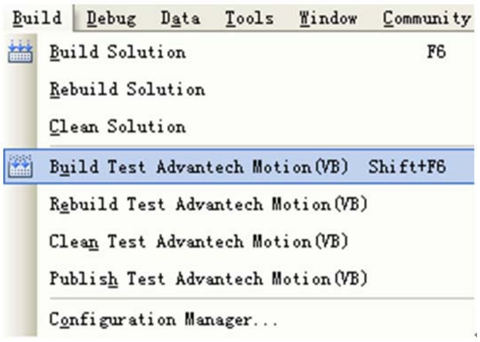
For details, refer to the Microsoft Visual Basic user manual.

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-1245L SoftMotion PCI Controller.

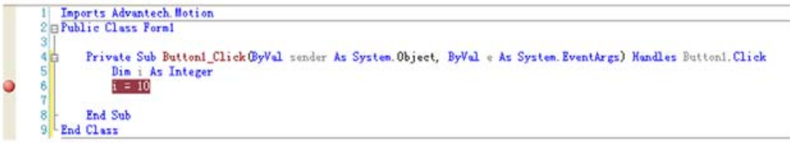
4. Test program

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:


Build
Debug
Data
Tools
Window
Community
Build Solution F6
Rebuild Solution
Clean Solution
Build Test Advantech Motion(VB) Shift+F6
Rebuild Test Advantech Motion(VB)
Clean Test Advantech Motion(VB)
Publish Test Advantech Motion(VB)
Configuration Manager...

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:

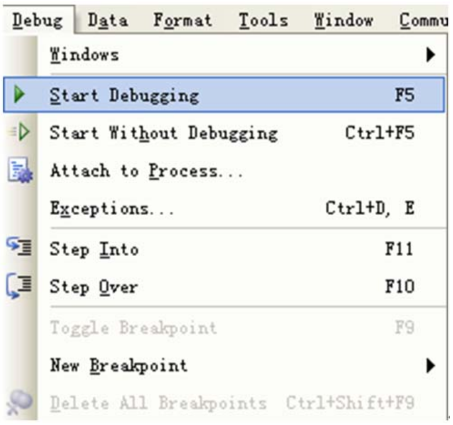


```
Imports Advantech.Motion
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim i As Integer
        i = 10

    End Sub
End Class
```

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-1245L.

NumOfSubDevices:

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

6.3.1.2 Acm_GetErrorMessage

Format:

BOOL Acm_GetErrorMessage (U32 ErrorCode, LPTSTR IpszError, 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.
IpszError	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.

6.3.1.3 Acm_DevWriteEEPROM_Ex

Format:

U32 Acm_DevWriteEEPROM_Ex(HAND DeviceHandle, U16 PrivateID, PU32 PassWordArray, U32 PassArrayCnt, PU32 WriteArray, U32 BufferLength)

Purpose:

Write the protective private data and password to EEPROM. Totally 8 sets are allowed and there are 8 bytes password and 8 bytes data per set.

Parameters:

Name	Type	In or Out	Description
DeviceHandle	HAND	IN	DeviceHandle
PrivateID	U16	IN	0-7
PassWordArray	PU32	IN	Password array pointer
PassArrayCnt	U32	IN	Password length is 2 per set
WriteArray	PU32	IN	Write data array pointer
BufferLength	U32	IN	Data length is 2 per set

Return Value:

Error Code.

Comments:

The default value of password and protective data is 0. It will overwrite original value without checking password while writing data.

6.3.1.4 Acm_DevReadEEPROM_Ex

Format:

Acm_DevReadEEPROM_Ex(HAND DeviceHandle, U16 PrivateID, PU32 PassWordArray, U32 PassArrayCnt, PU32 ReadArray, U32 BufferLength)

Purpose:

Input the right password to read the private data. There are 8 sets and 8 bytes for password and 8 bytes for data per set.

Parameters:

Name	Type	In or Out	Description
DeviceHandle	HAND	IN	DeviceHandle
PrivateID	U16	IN	0-7
PassWordArray	PU32	IN	Password array pointer
PassArrayCnt	U32	IN	Password length is 2 per set
ReadArray	PU32	OUT	Read private data
BufferLength	U32	IN	Private data length is 2 per set

Return Value:

Error Code.

Comments:

Reading will not be allowed by wrong password input over 3 times. The error count will change to 0 after restart.

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 pointer 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 OrgReact;
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-1245L.

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

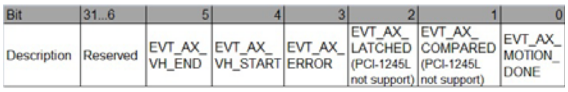
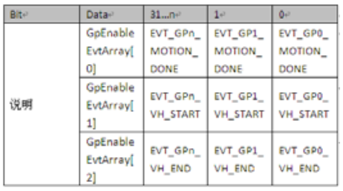
Format:

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:

<table>
<tr><td>Name</td><td>Type</td><td>In or Out</td><td>Description</td></tr>
<tr><td>DeviceHandle</td><td>HAND</td><td>IN</td><td>Device handle from Acm_DevOpen.</td></tr>
<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><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-1245L not support)</td><td>EVT_AX_COMPARED (PCI-1245L 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. <b>n is just 1</b>.
GpEvtStatus is 32 bits data type array and <b>currently the values of n can only be 1</b>.

<table>
<tr><td>Bit</td><td>Data</td><td>31...n</td><td>1</td><td>0</td></tr>
<tr><td>说明</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></td><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></td><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 waiting time 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

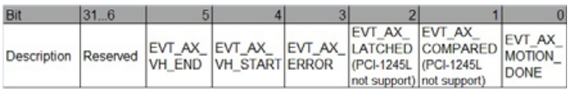
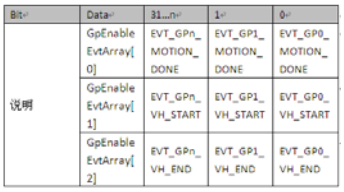
Format:

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

Purpose:

Enable motion event.

Parameters:

<table>
<tr><td>Name</td><td>Type</td><td>In or Out</td><td>Description</td></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-1245L not support)</td><td>EVT_AX_COMPARED (PCI-1245L not support)</td><td>EVT_AX_MOTION_DONE</td></tr>
</table>
Bit n = 1: Enable event;
Bit n = 0: Disable event.</td></tr>
<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 <b>the value of n can only be 1</b>.

<table>
<tr><td>Bit</td><td>Data</td><td>31...n</td><td>1</td><td>0</td></tr>
<tr><td>说明</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></td><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></td><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>
<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:

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

6.3.3 Axis

6.3.3.1 System

6.3.3.1.1Acm_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-1245L: 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-1245L: 0, 1, 2, 3.

6.3.3.1.2Acm_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.

6.3.3.1.3Acm_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:

6.3.3.2 Motion IO

6.3.3.2.1Acm_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:

6.3.3.2.2 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:RDY---- RDY pin input;
			1:ALM ---- Alarm Signal input;
			2:LMT+ ---- Limit Switch+;
			3:LMT- ---- Limit Switch-;
			4:ORG---- Origin Switch;
			5:DIR ---- DIR output;
			6:EMG ---- Emergency signal input;
			7:PCS ---- PCS signal (Not support in PCI-1245L);
			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-1245L);
			11: LTC ---- Latch signal input (Not support in PCI-1245L);
			12: SD ---- Slow Down signal input (not support in PCI-1245L):
			13: INP ---- In-Position signal input;
			14: SVON ---- Servo-ON (OUT6);
			15: ALRM ----Alarm Reset output status;
			16:SLMT+ ---- Software Limit+;
			17: SLMT- ---- Software Limit-;
			18: CMP-----Compare signal (Not support in PCI-1245L);
			19: CAMDO ---- position window do (Not support in PCI-1245L);

Return Value:

Error Code.

Comments:

6.3.3.3 Motion Status

6.3.3.3.1 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:

6.3.3.3.2Acm_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:

6.3.3.4 Velocity Motion

6.3.3.4.1Acm_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.

6.3.3.4.2Acm_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.

6.3.3.4.3Acm_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:

6.3.3.4.4Acm_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.
NewVelocity	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:

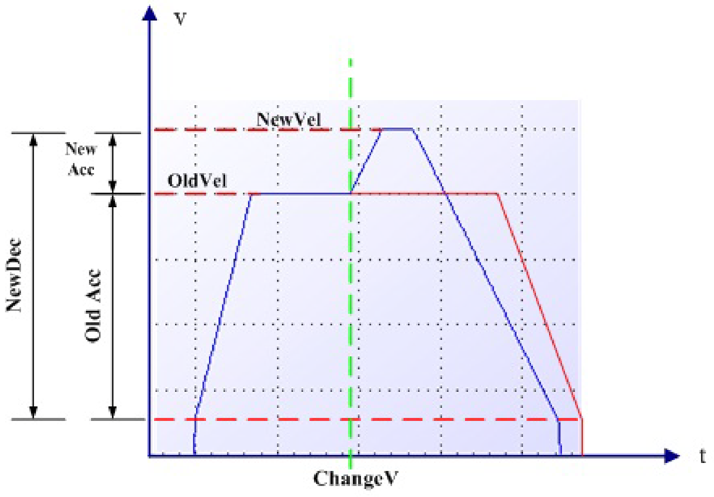
Error Code.

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 decelera tion 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.



6.3.3.4.5Acm_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:

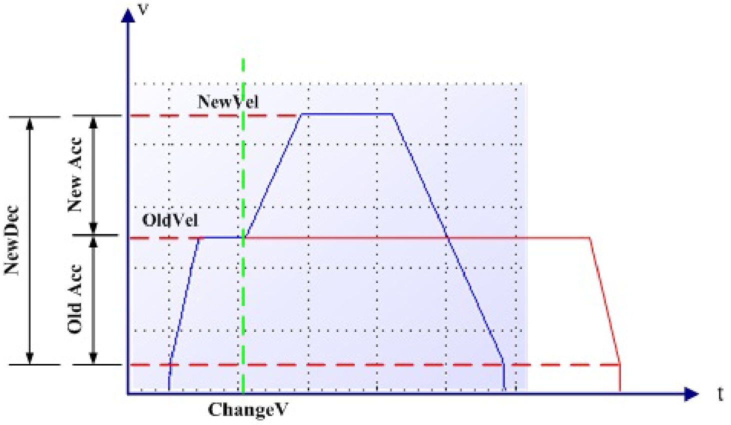
Error Code.

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.



6.3.3.4.6Acm_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.

6.3.3.5 Point-to-Point Motion

6.3.3.5.1 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.

6.3.3.5.2 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.

6.3.3.5.3 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.

6.3.3.6 Simultaneous Motion

6.3.3.6.1Acm_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**.

PCI-1285E does not support this API.

6.3.3.6.2Acm_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**.

PCI-1285E does not support this API.

6.3.3.6.3Acm_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.

6.3.3.6.4Acm_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.

6.3.3.6.5Acm_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.

6.3.3.7 Home

6.3.3.7.1Acm_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.

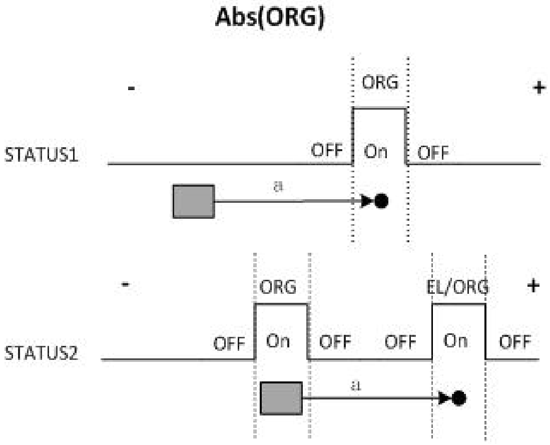
c. Trapezoid PTP take a uniform motion at VelLow. It will stop immediately when it meets ORG/EL signal.

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.

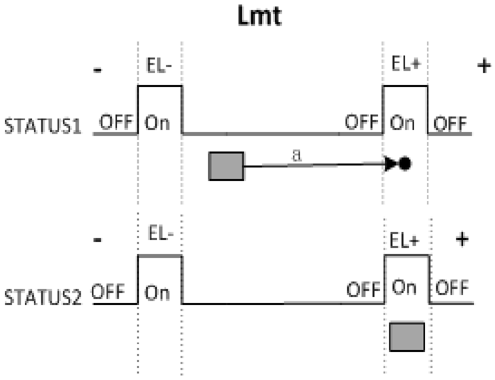
Note! *Features of trapezoid PTP motion: When start, the velocity will increase from VelLow to VelHigh with Acc (If distance is long enough); when end, the velocity will decrease from VelHigh to VelLow with Dec.*

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.



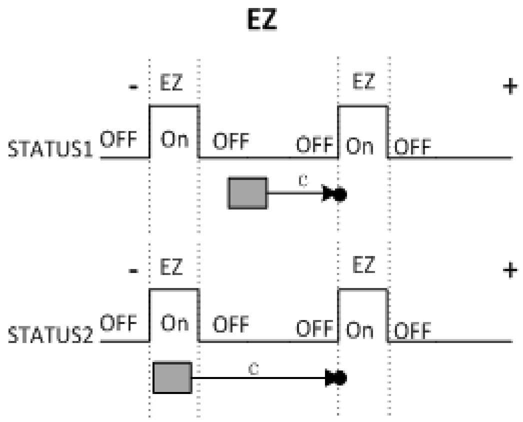
- STATUS1: If the object is out of the field of ORG signal, when home command is written, the object will move until ORG signal occurring.
- STATUS2: If the object is in the field of ORG signal or the direction is opposite with ORG switch, the object will move until ORG signal (if there are more than 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.



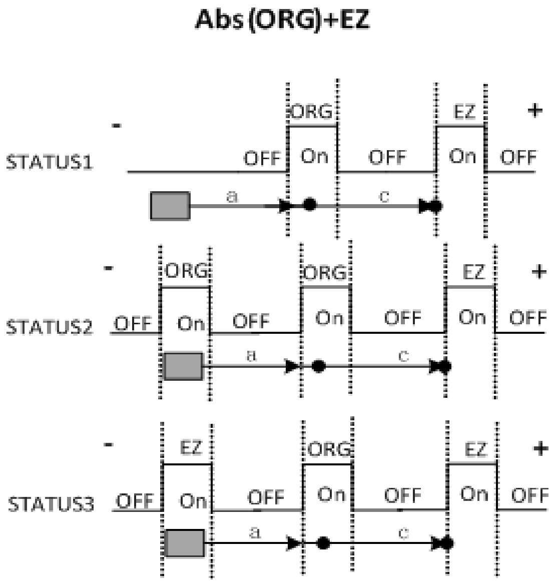
- STATUS1: If the object is out of the field of EL signal, when home command is written, the object will move until EL signal occurring.
- STATUS2: If the object is in the field of EL signal, there will be no response.

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.

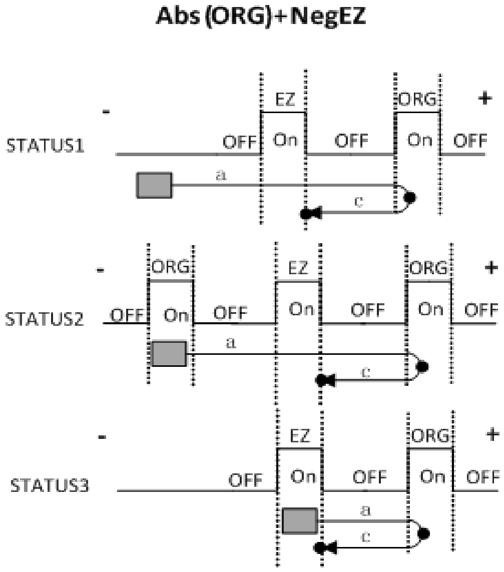
4. MODE4_Abs_Ref: ORG+EZ, Move(Dir) ->touch ORG ->Stop ->Move(Dir)->touch EZ ->Stop
   This is a composed home mode. Firstly, the object moves until origin signal occurring, and then continues to move in same direction with ORG until EZ signal occurring.
   For Example:
   Dir: Positive. ORG logic: Active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



- 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.

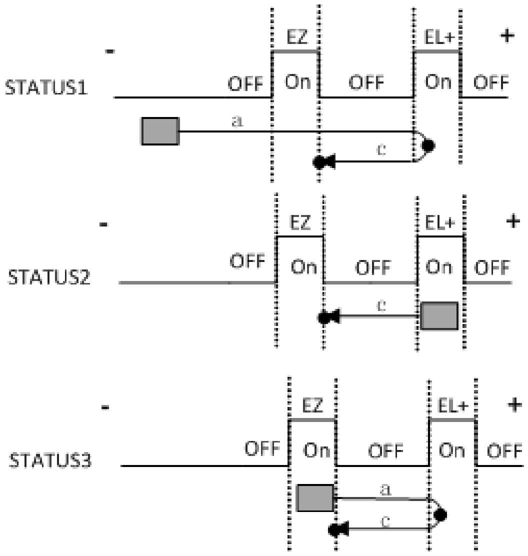
5. MODE5_Abs_NegRef: ORG+EZ, Move (Dir) ->touch ORG ->Stop ->Move (-Dir) ->touch EZ ->Stop.
   This is a composed home mode. The object moves until origin signal occurring firstly, and then continues to move in opposite direction with ORG until EZ signal is occurred.
   For Example:
   Dir: Positive.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

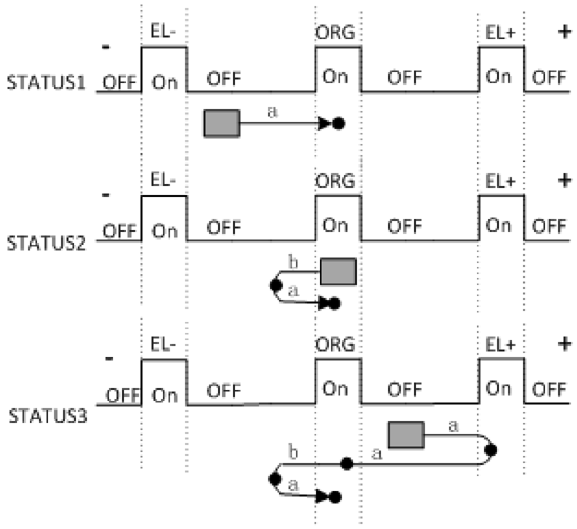
Lmt + EZ



- 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.

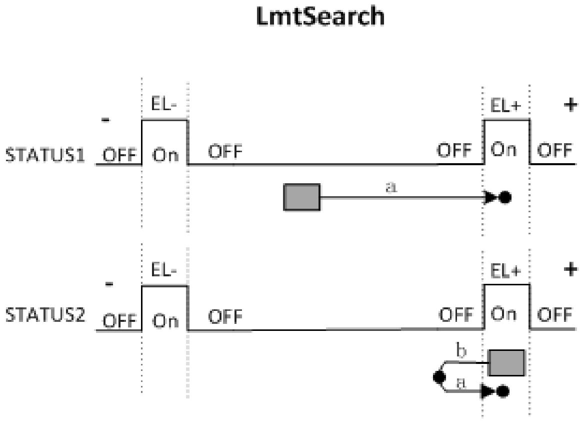
7. MODE7_AbsSearch: Move (Dir) ->Search ORG ->Stop.
   This is a mode of searching transformation of ORG signal from no signal to signal occurring.
   For Example:
   Dir: Positive. ORG logic: Active high. Limit logic: Active High.

AbsSearch



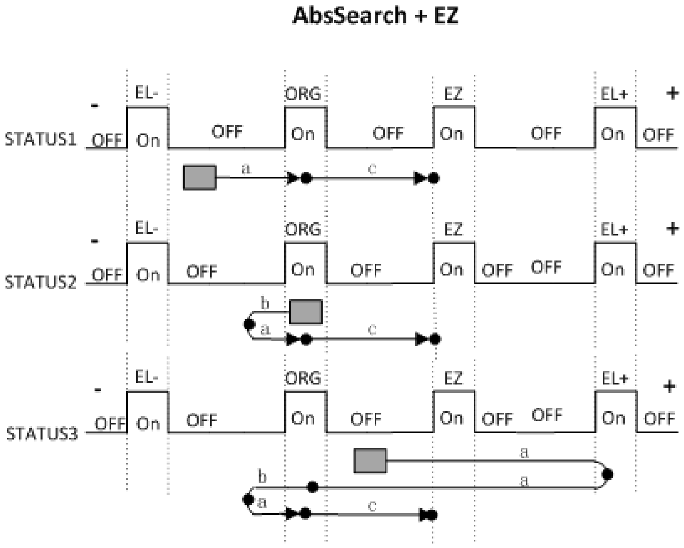
- STATUS1: If there is no ORG signal occurring, the object will stop when ORG signal occurs.
- STATUS2: If the object is in the field of ORG signal. The Object moves in opposite direction until the signal disappears, and then converts direction to move until ORG signal occurring.
- STATUS3: If there 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.



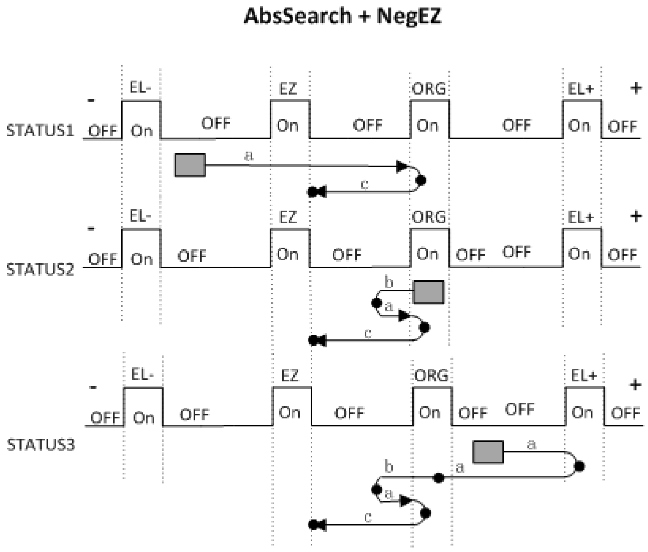
- STATUS1: If the Limit signal is occurred firstly while the object is moving, the home process is end.
- STATUS2: If the object is in the field of limit signal. The Object moves in opposite direction until the signal disappears, and then converts direction to move until limit signal occurring.

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.



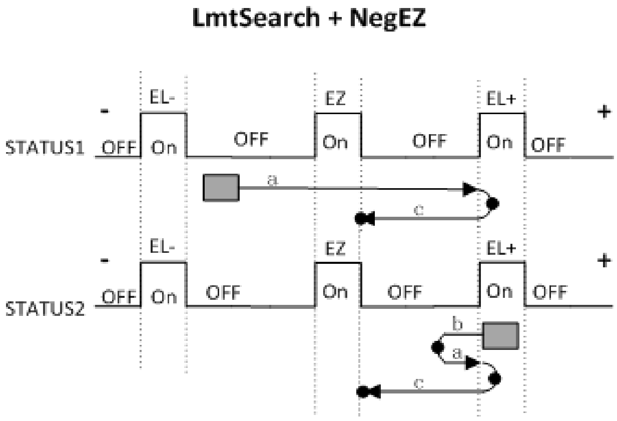
- STATUS1: If the object is out of the field of EZ signal and ORG signal, when home command is written, firstly, the object will move until ORG signal occurring, then continue to move until EZ signal occurring.
- STATUS2: If the object is in the field of ORG signal, the home command is written. Firstly, 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.



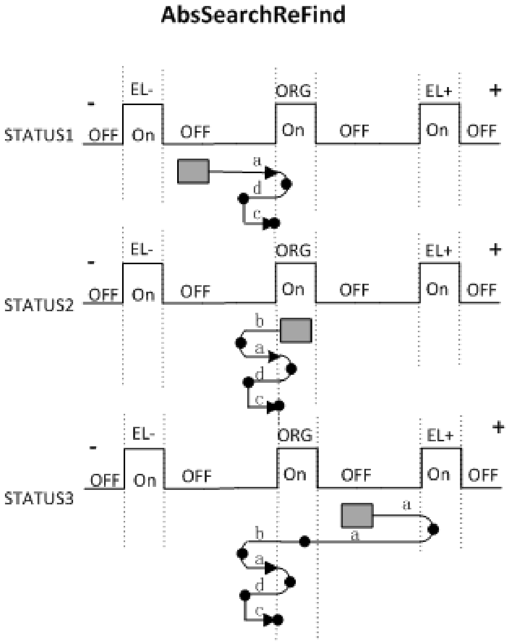
- STATUS1: If the object is out of the field of EZ signal and ORG signal, when home command is written. Firstly, the object will move until ORG signal occurring, then reverse direction and continue to move until EZ signal occurring.
- STATUS2: If the object is in the field of ORG signal, the home command 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.



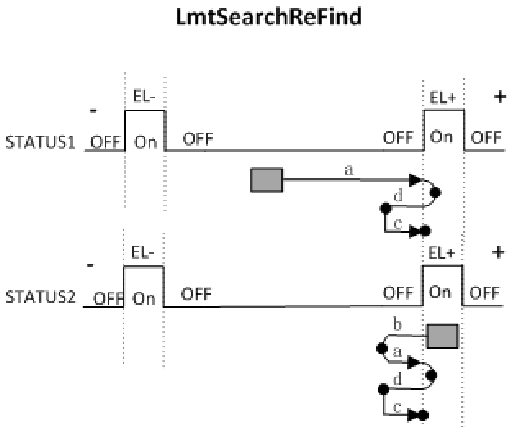
- STATUS1: When object is not in field of limit signal. Firstly, the object will move until EL signal occurring, then reverse direction and continue to move until EZ signal occurring.
- STATUS2: When object is in the field of limit signal. Firstly, the object reserves direction and moves, the EL signal 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.

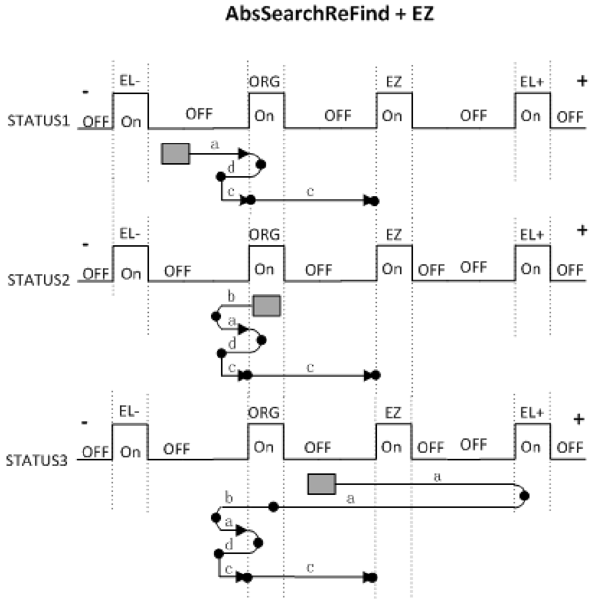


AbsSearch process has three situations. For detailed information, see about descriptions in MODE7_AbsSearch.

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.

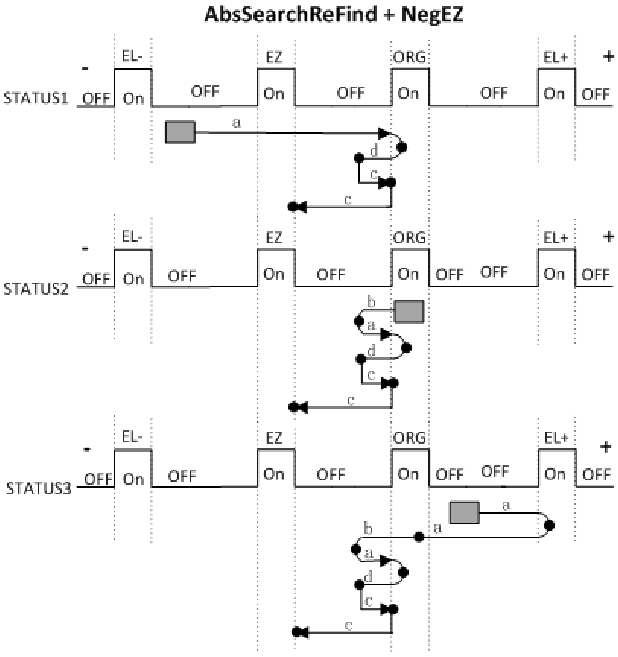


14. MODE14_AbsSearchRefind_Ref: Search ORG +Refind ORG+EZ, Move (Dir) ->Search ORG ->Stop->Move (-Dir) ->Leave ORG(FL) ->Stop-> Move (-Dir)->Refind ORG(FL)->Stop->Move (Dir) ->touch EZ ->Stop.
Firstly, axis moves in the way of MODE7_AbsSearch, and then moves uniformly in opposite direction at VelLow 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.



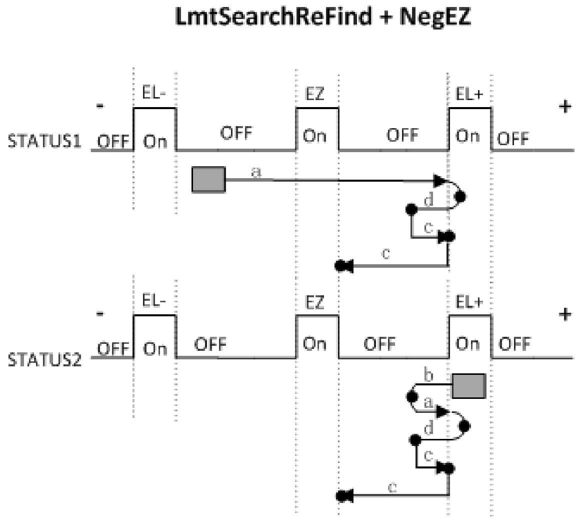
AbsSearch process has three situations. For detailed information, see about descriptions in MODE7_AbsSearch.

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.

6.3.3.8 Position/Counter Control

6.3.3.8.1Acm_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:

6.3.3.8.2Acm_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:

6.3.3.8.3Acm_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:

6.3.3.8.4Acm_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:

6.3.3.9 Aux/Gen Output

6.3.3.9.1Acm_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 Erc will be disabled automatically and these two functions use the same output pins(OUT7).

6.3.3.9.2Acm_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.

6.3.3.9.3Acm_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:

6.3.3.10 Ext-Drive

6.3.3.10.1Acm_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:

6.3.3.11 Stop

6.3.3.11.1Acm_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-gear motion, then the API can be used to terminate the synchronization.

6.3.3.11.2Acm_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 syn chronization.

6.3.3.11.3Acm_AxStopDecEx

Format:

U32 Acm_AxStopDecEx (HAND AxisHandle, F64 NewDec)

Purpose:

Command the axis to stop and specify the deceleration.

Parameters:

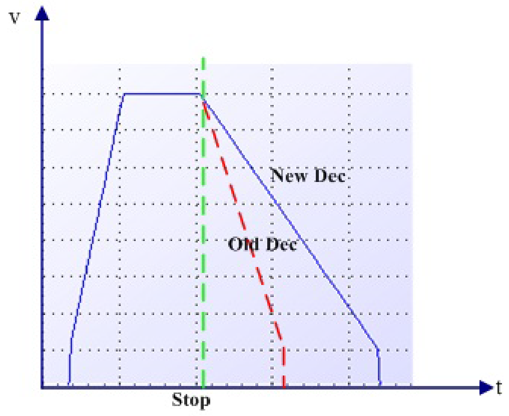
Name	Type	In or Out	Description
AxisHandle	HAND	IN	Axis handle from Acm_AxOpen.
NewDec	F64	IN	Deceleration for decelerating. (Unit: PPU/s^2)

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.



6.3.4 Group

6.3.4.1 SYSTEM

6.3.4.1.1Acm_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 is 1 group in PCI-1245L.

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.

6.3.4.1.2Acm_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.

6.3.4.1.3Acm_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.

6.3.4.1.4Acm_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.

6.3.4.2 Motion Status

6.3.4.2.1Acm_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.

6.3.4.2.2Acm_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.

6.3.4.3 MotionStop

6.3.4.3.1Acm_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:

6.3.4.3.2Acm_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:

6.3.4.4 Interpolation Motion

6.3.4.4.1Acm_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 axis with the longest distance) 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 2 axes linear interpolation in PCI-1245L.

6.3.4.4.2Acm_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 axis with the longest distance) 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 2 axes linear interpolation in PCI-1245L.

6.3.4.4.3Acm_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 axis with the longest distance) 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 2 axes direct interpolation in PCI-1245L.

6.3.4.4.4Acm_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 axis with the longest distance) 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 2 axes direct interpolation in PCI-1245L.

6.4 Property List

6.4.1 Device

6.4.1.1 Feature

6.4.1.1.1FT_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:

6.4.1.1.2FT_DevAxisCount

Data Type:

U32

R/W:

R

PropertyID:

1

Meaning:

Get axis number of this device.

Comments:

6.4.1.1.3FT_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 (PCI-1245L does not support)
2	DO (PCI-1245L does not support)
3	AI (PCI-1245L does not support)
4	AO (PCI-1245L does not support)
5	Timer
6	Counter
7	DAQ DI (PCI-1245L does not support)
8	DAQ DO (PCI-1245L does not support)
9	DAQ AI(PCI-1245L does not support)
10	DAQ AO (PCI-1245L does not support)
11	Emg
12~31	No definition

Comments:

6.4.1.1.4FT_DevOverflowCntr

Data Type:

U32

R/W:

R

PropertyID:

3

Meaning:

The maximum data count of position counter.

Comments:

For PCI-1245L, the maximum data count is 2147483647.

6.4.1.2 Configuration

6.4.1.2.1CFG_DevBoardID

Data Type:

U32

R/W:

R

PropertyID:

201

Meaning:

Get Device ID. For PCI-1245L, this property value will be 0~15.

Comments:

6.4.1.2.2CFG_DevBaseAddress

Data Type:

U32

R/W:

R

PropertyID:

203

Meaning:

Return IO base address.

Comments:

6.4.1.2.3CFG_DevInterrupt

Data Type:

U32

R/W:

R

PropertyID:

204

Meaning:

Get Device interrupt number.

Comments:

6.4.1.2.4CFG_DevBusNumber

Data Type:

U32

R/W:

R

PropertyID:

205

Meaning:

Get device bus number.

Comments:

6.4.1.2.5CFG_DevSlotNumber

Data Type:

U32

R/W:

R

PropertyID:

206

Meaning:

Get device slot number.

Comments:

6.4.1.2.6CFG_DevDriverVersion

Data Type:

char*

R/W:

R

PropertyID:

207

Meaning:

Get SYS driver's version. The format is: 1.0.0.1

Comments:

6.4.1.2.7CFG_DevDllVersion

Data Type:

char*

R/W:

R

PropertyID:

208

Meaning:

Get DLL driver's version. The format is: 1.0.0.1.

Comments:

6.4.1.2.8CFG_DevFwVersion

Data Type:

char*

R/W:

R

PropertyID:

208

Meaning:

Get the firm version, the format is: 1.0.0.1.

Comments:

6.4.1.2.9CFG_DevCPLDVersion

Data Type:

char*

R/W:

R

PropertyID:

218

Meaning:

Get the FPGA version of device, the format is: 1.0.0.1.

Comments:

6.4.1.2.10CFG_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 Axis

6.4.2.1 Feature

6.4.2.1.1System

6.4.2.1.1.1 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:

6.4.2.1.2Speed Pattern

6.4.2.1.2.1 FT_AxMaxVel

Data Type:

F64

R/W:

R

PropertyID:

302

Meaning:

Get axis supported max velocity. (Unit: Pulse/s)

Comments:

In PCI-1245L, the value is 1,000,000.

6.4.2.1.2.2 FT_AxMaxAcc

Data Type:

F64

R/W:

R

PropertyID:

303

Meaning:

Get axis supported max acceleration. (Unit: Pulse/s^2)

Comments:

In PCI-1245L, the value is 100,000,000.

6.4.2.1.2.3 FT_AxMaxDec

Data Type:

F64

R/W:

R

PropertyID:

304

Meaning:

Get axis supported max deceleration (Unit: Pulse/s^2)

Comments:

In PCI-1245L, the value is 100,000,000.

6.4.2.1.2.4 FT_AxMaxJerk

Data Type:

F64

R/W:

R

PropertyID:

305

Meaning:

Get axis supported max jerk. (Unit: Pulse/S^3)

Comments:

In PCI-1245L, the value is 1.

6.4.2.1.3Pulse In

6.4.2.1.3.1 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:

6.4.2.1.3.2 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:

6.4.2.1.4Pulse Out

6.4.2.1.4.1 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-1245L, the value is 1.

6.4.2.1.4.2 FT_AxPulseOutModeMap

Data Type:

U32

R/W:

R

PropertyID:

309

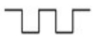
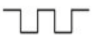
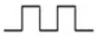
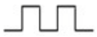
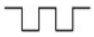
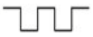
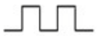
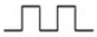
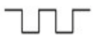
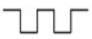
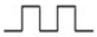
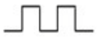
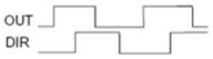
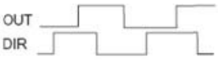
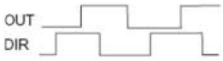
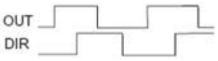
Meaning:

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-1245L, 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				

6.4.2.1.5 Alarm

6.4.2.1.5.1 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:

6.4.2.1.6In Position

6.4.2.1.6.1 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:

6.4.2.1.7ERC

6.4.2.1.7.1 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:

6.4.2.1.7.2 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:

6.4.2.1.8SD

6.4.2.1.8.1 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-1245L, the value is 0.

6.4.2.1.9Hardware Limit

6.4.2.1.9.1 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:

6.4.2.1.10Software Limit

6.4.2.1.10.1 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:

6.4.2.1.10.2 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:

6.4.2.1.11Home

6.4.2.1.11.1 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:

6.4.2.1.11.2 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.

6.4.2.1.12Backlash

6.4.2.1.12.1 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:

6.4.2.1.13Ext-Drive

6.4.2.1.13.1 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:

6.4.2.1.13.2 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:

6.4.2.1.14Aux/Gen DIO

6.4.2.1.14.1 FT_AxGenDOMap

Data Type:

U32

R/W:

R

PropertyID:

329

Meaning:

Get axis supported general output from OUT4 to OUT7.

Bits	Description
0	OUT4
1	OUT5
2	OUT6/SVON
3	OUT7/ERC
4~31	Not defined.

Comments:

6.4.2.1.14.2 FT_AxGenDIMap

Data Type:

U32

R/W:

R

PropertyID:

330

Meaning:

Get axis supported general input from IN1 to IN4

Bits	Description
0	IN1
1	IN2/RDY
2	IN3/JOG+
3	IN4/JOG-
4~31	Not defined.

Comments:

6.4.2.1.15Simultaneity

6.4.2.1.15.1 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
12~15	Start Simultaneous Starting with axis_7's compare signal
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.
20~31	Not defined.

Comments:

Get axis supported simultaneous starting mode. See about CFG_AxSimStartSource. In PCI-1245L, the default value:986881. In PCI-1285E, the default value:0.

6.4.2.1.16Trigger Stop

6.4.2.1.16.1 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

6.4.2.1.16.2 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

6.4.2.1.16.3 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

6.4.2.1.16.4 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

6.4.2.2 Config

6.4.2.2.1System

6.4.2.2.1.1 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.

6.4.2.2.1.2 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

Comments:

6.4.2.2.2Speed Pattern

6.4.2.2.2.1 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-1245L, the default value is 1,000,000.

6.4.2.2.2.2 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-1245L, the default value is 50,000,000.

6.4.2.2.2.3 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-1245L, the default value is 50,000,000.

6.4.2.2.2.4 CFG_AxMaxJerk

Data Type:

F64

R/W:

R

PropertyID:

556

Meaning:

Get max jerk configuration for the motion axis.

Comments:

In PCI-1245L, the value is 1.

6.4.2.2.3Pulse In

6.4.2.2.3.1 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:

6.4.2.2.3.2 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:

6.4.2.2.3.3 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:

6.4.2.2.4Pulse Out

6.4.2.2.4.1 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-1245L, the default value is 16.

See also FT_AxPulseOutMode.

6.4.2.2.5Alarm

6.4.2.2.5.1 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-1245L, the default value is 1.

6.4.2.2.5.2 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-1245L, the default value is 0.

Please modify "CFG_AxAlmReact" and "CFG_AxAlmLogic" before modifying the value of "CFG_AxAlmEnable".

6.4.2.2.5.3 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-1245L, the default value is 1.

6.4.2.2.6In Position

6.4.2.2.6.1 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-1245L, the default value is 0.

6.4.2.2.6.2 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-1245L, the default value is 1.

6.4.2.2.7ERC

6.4.2.2.7.1 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-1245L, the default value is 1.

6.4.2.2.7.2 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-1245L, the default value is 1.

6.4.2.2.8Hardware Limit

6.4.2.2.8.1 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-1245L, the default value is 0.

6.4.2.2.8.2 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-1245L, the default value is 0.

6.4.2.2.8.3 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-1245L, the default value is 1.

Please modify "CFG_AxElReact" and "CFG_AxElLogic" before modifying the value of "CFG_AxElEnable".

6.4.2.2.9Software Limit

6.4.2.2.9.1 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:

6.4.2.2.9.2 CFG_AxSwPelEnable

Data Type:

U32

R/W:

RW

PropertyID:

578

Meaning:

Enable/Disable the plus software limit.

Value	Description
0	Disabled
1	Enabled

Comments:

6.4.2.2.9.3 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-1245L, the default value is 1.

6.4.2.2.9.4 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-1245L, the default value is 1.

6.4.2.2.9.5 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:

6.4.2.2.9.6 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:

6.4.2.2.10Home

6.4.2.2.10.1 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-1245L, the default value is 0.

6.4.2.2.10.2 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-1245L, the default value is 0.

6.4.2.2.10.3 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:

6.4.2.2.10.4 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:

6.4.2.2.11Backlash

6.4.2.2.11.1 CFG_AxBacklashEnable

Data Type:

U32

R/W:

RW

PropertyID:

593

Meaning:

Enable/Disable corrective backlash.

Value	Description
0	Disabled
1	Enabled

Comments:

In PCI-1245L, the default value is 0.

6.4.2.2.11.2 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-1245L, the default value is 10.

6.4.2.2.11.3 CFG_AxBacklashVel

Data Type:

U32

R/W:

RW

PropertyID:

630

Meaning:

Set /get the velocity of corrective backlash. (Uint: pulse/s)

Comments:

In PCI-1245L, the default value is 1000.

6.4.2.2.12 Aux/Gen DIO

6.4.2.2.12.1 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, CFG_AxErcEnableMode is disabled automatically.

6.4.2.2.13Ext-Drive

6.4.2.2.13.1 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-1245L, only support 0.

6.4.2.2.13.2 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-1245L, only support 0.

6.4.2.2.13.3 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-1245L, the default value is 0.

6.4.2.2.13.4 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:

6.4.2.2.13.5 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-1245L, the default value is 1.This value must lager than zero.

6.4.2.2.14Simultaneity

6.4.2.2.14.1 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
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.

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 256~2048, 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 65536~524288, 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.

6.4.2.2.15 Trigger Stop

6.4.2.2.15.1 CFG_AxIN1StopEnable

Data Type:

U32

R/W:

R&W

PropertyID:

635

Meaning:

Enable/disable INI trigger stop function.

Comments:

Value	Description
0	Enabled
1	Disabled

6.4.2.2.15.2 CFG_AxIN1StopReact

Data Type:

U32

R/W:

R&W

PropertyID:

636

Meaning:

Set/get INI trigger stop mode.

Comments:

Value	Description
0	Sudden stop
1	Decelerating

6.4.2.2.15.3 CFG_AxIN1StopLogic

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

6.4.2.2.15.4 CFG_AxIN2StopEnable

Data Type:

U32

R/W:

R&W

PropertyID:

638

Meaning:

Enable/disable IN2 trigger stop function.

Comments:

Value	Description
0	Enabled
1	Disabled

6.4.2.2.15.5 CFG_AxIN2StopReact

Data Type:

U32

R/W:

R&W

PropertyID:

639

Meaning:

Set/get IN2 trigger stop mode.

Comments:

Value	Description
0	Sudden stop
1	Decelerating

6.4.2.2.15.6 CFG_AxIN2StopLogic

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

6.4.2.2.15.7 CFG_AxIN4StopEnable

Data Type:

U32

R/W:

R&W

PropertyID:

641

Meaning:

Enable/disable IN4 trigger stop function.

Comments:

Value	Description
0	Enabled
1	Disabled

6.4.2.2.15.8 CFG_AxIN4StopReact

Data Type:

U32

R/W:

R&W

PropertyID:

642

Meaning:

Set/get IN4 trigger stop mode.

Comments:

Value	Description
0	Sudden stop
1	Decelerating

6.4.2.2.15.9 CFG_AxIN4StopLogic

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

6.4.2.2.15.10 CFG_AxIN5StopEnable

Data Type:

U32

R/W:

R&W

PropertyID:

644

Meaning:

Enable/disable IN5 trigger stop function.

Comments:

Value	Description
0	Enabled
1	Disabled

6.4.2.2.15.11 CFG_AxIN5StopReact

Data Type:

U32

R/W:

R&W

PropertyID:

645

Meaning:

Set/get IN2 trigger stop mode.

Comments:

Value	Description
0	Sudden stop
1	Decelerating

6.4.2.2.15.12 CFG_AxIN5StopLogic

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

6.4.2.3 Parameter

6.4.2.3.1Speed Pattern

6.4.2.3.1.1 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.

6.4.2.3.1.2 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.

6.4.2.3.1.3 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.

6.4.2.3.1.4 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.

6.4.2.3.1.5 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.

6.4.2.3.2Home

6.4.2.3.2.1 PAR_AxHomeCrossDistance

Data Type:

F64

R/W:

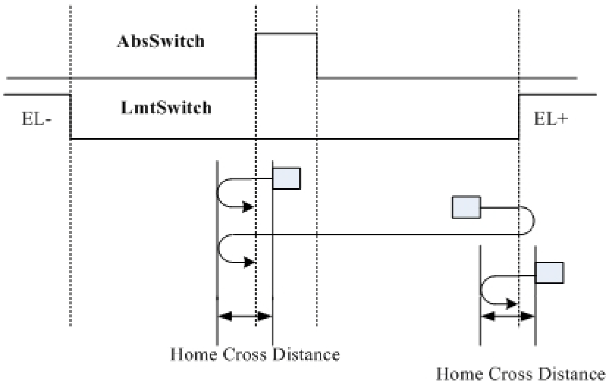
RW

PropertyID:

408

Meaning:

Set the home cross distance (Unit: PPU). This property must be greater than 0. The default value is 10000.



6.4.2.3.2.2 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

6.4.3 Group

6.4.3.1 Config

6.4.3.1.1System

6.4.3.1.1.1 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

Comments:

6.4.3.2 Parameter

6.4.3.2.1Speed Pattern

6.4.3.2.1.1 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.

6.4.3.2.1.2 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.

6.4.3.2.1.3 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.

6.4.3.2.1.4 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.

6.4.3.2.1.5 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.

6.4.3.2.2System

6.4.3.2.2.1 PAR_GpGroupID

Data Type:

U32

R/W:

R

PropertyID:

706

Meaning:

Get the GroupID through GroupHandle.

Comments:

In PCI-1245L, there are only one GroupID to use.

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
0x8000007b	SpeedFordFunNotSpported
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
0x800000e0	CamTableIsEmpty
0x800000e1	InvalidPlaneVector
0x800000e2	MasAxIDSameSlvAxID
0x800000e3	InvalidGpRefPlane
0x800000e4	InvalidAxModuleRange
0x800000e5	DownloadFileFailed
0x800000e6	InvalidFileLength
0x800000e7	InvalidCmpCnt
0x800000e8	JerkExceededMaxValue
0x800000e9	AbsMotionNotSupport
0x800000ea	InvalidAiRange
0x800000eb	AIScaleFailed
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
0x80005014	Dsp_ReadPrivateOverMaxTimes
0x80005015	Dsp_InvalidPrivateID
0x80005017	Dsp_LastOperationNotOver
0x80005018	Dsp_WritePrivateTimeout
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
0x8000511a	Dsp_AxisMustBeModuloAxis
0x8000511b	Dsp_AxIdCantSameAsMasId
0x8000511c	Dsp_CantResetPosiOfMasAxis
0x8000511d	Dsp_InvalidAxExtDrvOperation
0x8000511e	Dsp_AxAccExceededMaxAcc
0x8000511f	Dsp_AxVelExceededMaxVel
0x80005120	Dsp_NotEnoughPulseForChgV
0x80005121	Dsp_NewVelMustGreaterThanVelLow
0x80005122	Dsp_InvalidAxGearMode
0x80005123	Dsp_InvalidGearRatio
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

Appendix A

Software Function Table

A.1 Software Function Table

	Item	Description	PCI-1245L
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	√
		Stop	√
	Multi-axes (Group) motion	Groups	1 group
		Line: axes	2 axes
	Home	16 modes	√
	Simultaneously Start/Stop	Simultaneously Start/Stop	√
Interrupt	Axes	Axes stop	√
		Axes error	√
		Axes VH start	√
		Axes VH stop	√
	Group	Group stop	√
		Group VH start	√
		Group VH start	√

Appendix B

Specifications

B.1 Axis

Item	Description
Number of axis	4
Type of control output	Pulse

B.2 Digital Input

Item		Description
Channels		LMT+,LMT-, ORG, INP, ALM, EMG, RDY
Type		One terminal, opto-isolated
Input voltage	L(max)	4Vdc
	H(min)	10Vdc
	H(max)	30Vdc
Max. input delay time		150us
Protection		2,500V Isolation
Input resistance		8.4kΩ

B.3 High Speed Digital Input

Item		Description
Channels		JOG+, JOG-
Type		One terminal, opto-isolated
Input voltage	L(max)	3Vdc
	H(min)	10Vdc
	H(max)	30Vdc
Max. input delay time		2us
Protection		2,500V Isolation
Input resistance		8.4kΩ

B.4 Digital Output

Item		Description
Channels		SVON, ERC
Type		One terminal, opto-isolated, sink type
Operation Voltage	Low	5Vdc
	High	30Vdc
Max. sink current		120mA per channel
Max. output delay time		60us
Protection		2,500V Isolation

B.5 Digital Output

Item		Description
Channels		OUT4, OUT5
Type		One terminal, opto-isolated, sink type
Operation Voltage	Low	5Vdc
	High	30Vdc
Max. sink current		120mA per channel
Max. output delay time		20us
Protection		2,500V Isolation

B.6 Pulse Input

Item		Description
Channels		ECA+,ECA-,ECB+,ECB-,ECZ+,ECZ
Type		Two terminal, opto-isolated
Max frequency		1MHz x1, x2, x4 (A/B phase only)
Input voltage	L(max)	1Vdc
	H(min)	3.5Vdc
	H(max)	10Vdc
Protection		2,500V Isolation

B.7 Pulse Output

Item		Description
Max frequency		1Mpps
Type		Two terminal, opto-isolated
Output voltage	Min	2Vdc/35mA
	Max	3.9Vdc/0mA
Output current		2VDC/35mA; 2.5VDC/30mA; 3VDC/15mA; 3.4VDC/1mA; 3.9VDC/0mA
Output signal mode		Differential line driving output
Protection		2,500V Isolation

B.8 General

Item		Description
Connector		SCSI D-SUB-100P
Dimensions		175mm x 100mm
Certifications		CE, FCC Class A
Power consumption	Typical	+5V / 0.6A
	Max	+5V / 1A
Temperature	Operating	0-60°C (refer to IEC 60068-2-1,2)
	Storage	-20~85°C
Relative Humidity		5~95% RH non-condensing (refer to IEC 60068-2-3)
External Power Voltage		DC +12 ~ 24 V

ADVANTECH

Enabling an Intelligent Planet

www.advantech.com