Industrial Automation Headquarters

Delta Electronics, Inc.
Taoyuan Technology Center
No.18, Xinglong Rd., Taoyuan City,
Taoyuan County 33068, Taiwan
TEL: 886-3-362-6301 / FAX: 886-3-371-6301

Asia

Delta Electronics (Jiangsu) Ltd.
Wujiang Plant 3
1688 Jiangxing East Road,
Wujiang Economic Development Zone
Wujiang City, Jiang Su Province,
People's Republic of China (Post code: 215200)
TEL: 86-512-6340-3008 / FAX: 86-769-6340-7290

Delta Greentech (China) Co., Ltd.
238 Min-Xia Road, Pudong District,
ShangHai, P.R.C.
Post code : 201209
TEL: 86-21-58635678 / FAX: 86-21-58630003

Delta Electronics (Japan), Inc.
Tokyo Office
2-1-14 Minato-ku Shibadaimon,
Tokyo 105-0012, Japan
TEL: 81-3-5733-1111 / FAX: 81-3-5733-1211

Delta Electronics (Korea), Inc.
1511, Byucksan Digital Valley 6-cha, Gasan-dong,
Geumcheon-gu, Seoul, Korea, 153-704
TEL: 82-2-515-5303 / FAX: 82-2-515-5302

Delta Electronics Int'l (S) Pte Ltd
4 Kaki Bukit Ave 1, #05-05, Singapore 417939
TEL: 65-6747-5155 / FAX: 65-6744-9228

Delta Electronics (India) Pvt. Ltd.
Plot No 43 Sector 35, HSIIDC
Gurgaon, PIN 122001, Haryana, India
TEL : 91-124-4874900 / FAX : 91-124-4874945

Americas

Delta Products Corporation (USA)
Raleigh Office
P.O. Box 12173,5101 Davis Drive,
Research Triangle Park, NC 27709, U.S.A.
TEL: 1-919-767-3800 / FAX: 1-919-767-8080

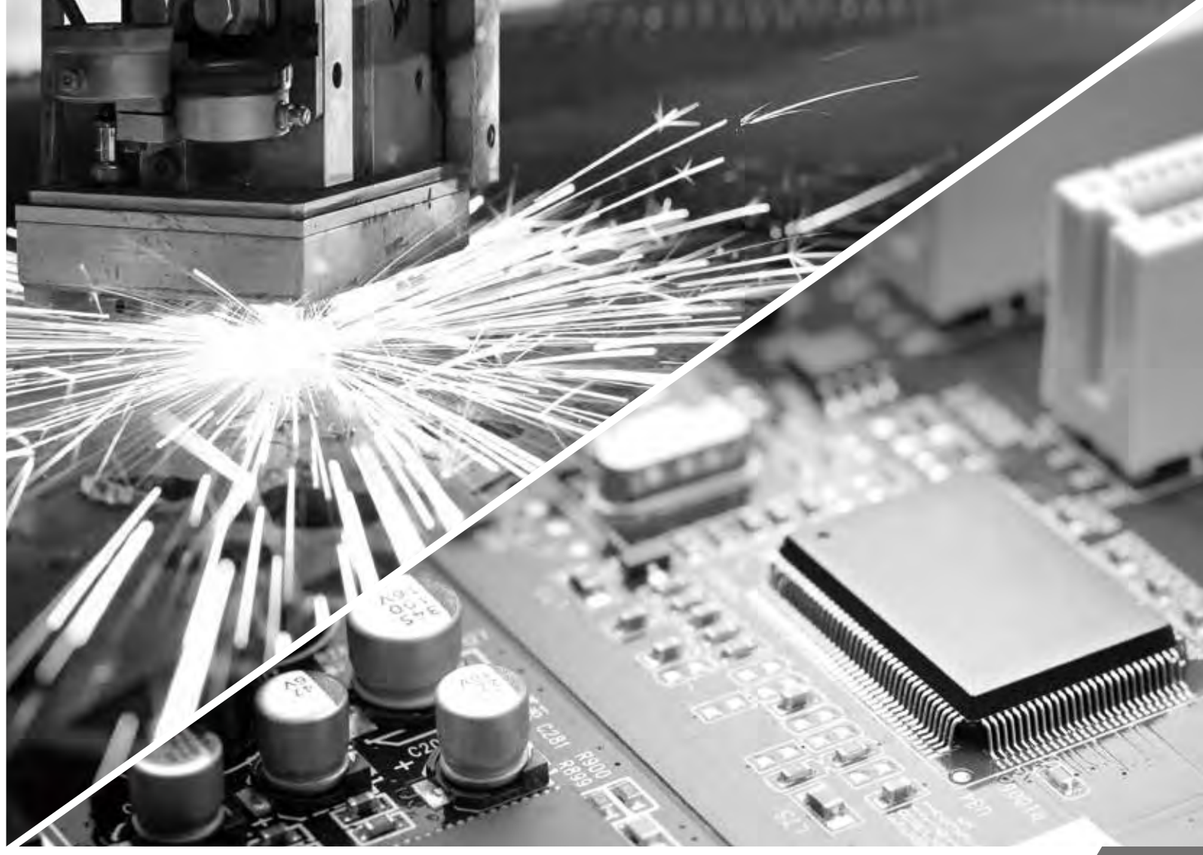
Delta Greentech (Brasil) S.A
Sao Paulo Office
Rua Itapeva, 26 - 3° andar Edificio Itapeva One-Bela Vista
01332-000-São Paulo-SP-Brazil
TEL: +55 11 3568-3855 / FAX: +55 11 3568-3865

Europe

Deltronics (The Netherlands) B.V.
Eindhoven Office
De Witbogt 15, 5652 AG Eindhoven, The Netherlands
TEL: 31-40-2592850 / FAX: 31-40-2592851

DELTA_IA-ASD_DMCNET_UM_EN_20140711

Delta High-speed Motion Control System General Instruction for DMCNET system



Delta High-speed Motion Control System

General Instruction for DMCNET system

www.deltaww.com

Smarter. Greener. Together.

About this Manual

This manual explains the structure, performance, and software function interface of Delta's A01/B01/F01 motion control card system, and seeks to provide users with a quick and convenient guide to applications.

References

1. A01/B01 card manual PCI-DMC-01_UserGuide_Traditional.pdf
2. F01 card manualPCI-DMC-F01_UserGuide_Traditional.pdf
3. Remote module manual ASD-DMC-RM_UserGuide_Traditional.pdf
4. Integrated expansion module manual ASD-DMC-GAGE_UserGuide_Traditional.pdf
5. Software develop manual PCI-DMC_ProgrammingGuide_Traditional.pdf
6. Server manual DELTA_ASDA-A2_M_TC_********.pdf

Technical Support and Service

If you require technical support, service and related information or have any questions during the use of this product, please visit our website (http://www.delta.com.tw/industrialautomation) or contact us. We look forward to providing the best possible support and service for your needs. Our contact details are provided below.

ASIA
DELTA ELECTRONICS, INC.
Taoyuan Plant 1
31-1, XINGBANG ROAD,
GUISHAN INDUSTRIAL ZONE,
TAOYUAN COUNTY 33370, TAIWAN, R.O.C.
TEL: 886-3-362-6301
FAX: 886-3-362-7267

JAPAN
DELTA ELECTRONICS (JAPAN), INC.
Tokyo Office
DELTA SHIBADAIMON BUILDING
2-1-14 SHIBADAIMON, MINATO-KU,
TOKYO, 105-0012, JAPAN
TEL: 81-3-5733-1111
FAX: 81-3-5733-1211

NORTH/SOUTH AMERICA
DELTA PRODUCTS CORPORATION (USA)
Raleigh Office
P.O. BOX 12173
5101 DAVIS DRIVE,
RESEARCH TRIANGLE PARK, NC 27709, U.S.A.
TEL: 1-919-767-3813
FAX: 1-919-767-3969

EUROPE
DELTRONICS (THE NETHERLANDS) B.V.
Eindhoven Office
DE WITBOGT 15, 5652 AG EINDHOVEN,
THE NETHERLANDS
TEL: 31-40-259-2850
FAX: 31-40-259-2851

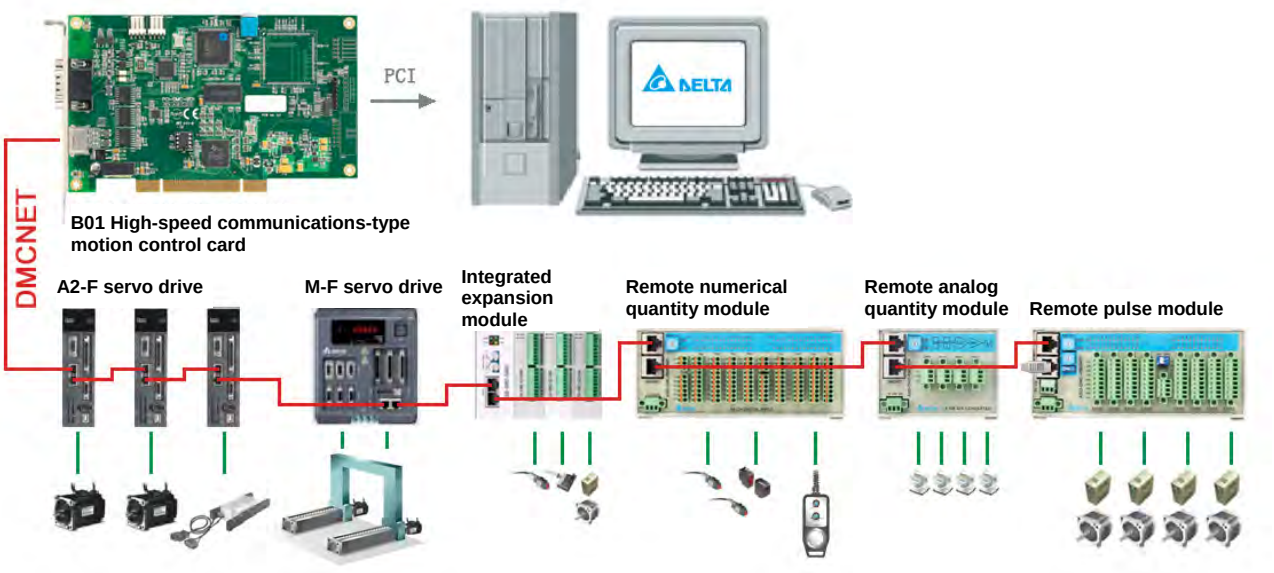
Table of Contents

(This page is intentionally left blank.)

Chapter 1 System Overview

The “Delta motion control card system” is a comprehensive motion control solution based on an industrial control computer recently introduced by Delta Electronics.

With a PCI interface motion control card, this system employs the DMCNET (Delta Motion Control Network) communications bus, and can be connected with Delta's servo motors, digital measurement modules, analog measurement modules, and pulse modules. It also provides a rich set of software development interface functions, and can realize such functions as planar linear interpolation, planar arc interpolation, spatial linear interpolation, spatial helical interpolation, multi-axis interpolation, up to 12 axis interpolation, location mode motion, velocity mode motion, torque mode motion, digital input and output, analog input and output, pulse module, and MPG access, etc. All functions can be combined or switched in a flexible manner, providing users with a high-speed, precise, stable motion control solution.

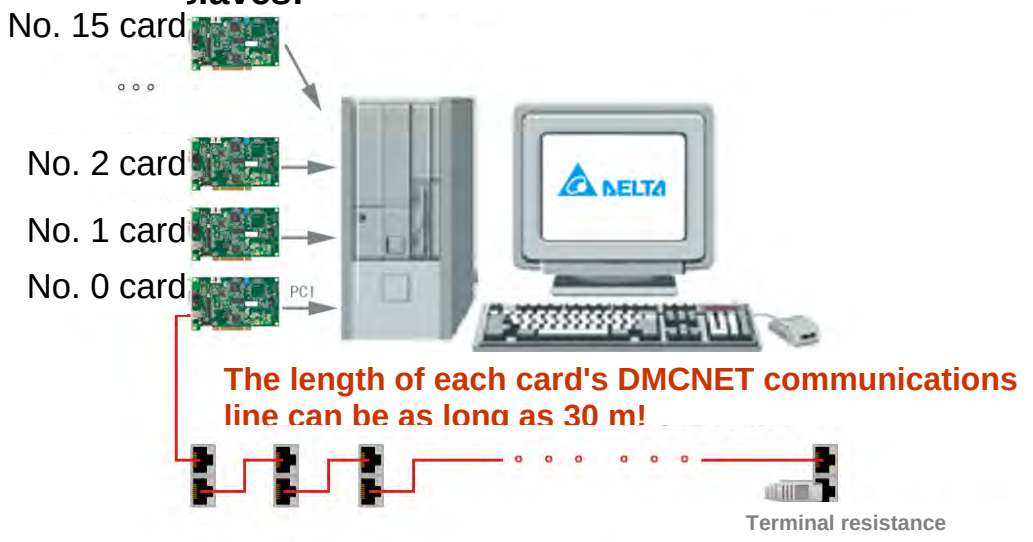


1.1 Features & Functions

- Planar linear interpolation
- Planar arc interpolation
- Spatial linear interpolation
- Spatial helical interpolation
- Multi-set multi-axis interpolation
- Up to 12-axis interpolation
- Position mode motion
- 35 types of homing modes
- Software limits
- Synchronous initiation of motion commands
- JOG
- Speed mode motion
- Torque mode motion
- Digital input / output
- Analog input / output
- Pulse module
- MPG access
- Embedded security
- Motion command buffer area
- Motion interrupt
- Triggered when the external position is reached

1.2 Maximum Configuration

Maximum system configuration: 16 cards × 12 slaves/card = 192 slaves!



Each card can have a maximum of 12 slaves!

1.3 Communication Bus

The DMCNET (Delta Motion Control Network) communication bus is a motion control communication protocol employing the high-speed bus developed by Delta; it employs different CRC code when engaging in two-way communications, has a transmission rate of 10Mbps in each channel, and supports a maximum communication distance of up to 30 m. Physical communication cables consist of CAT-5e STP Ethernet cable (24AWG/4Pairs).

1.4 Product Checklist

Type	Title	Model	Notes
Card	A01 motion control card	PCI-DMC-A01	PCI interface, 12 nodes (maximum of 12 axes)
	B01 motion control card	PCI-DMC-B01	PCI interface, 12 nodes (maximum of 12 axes)
	F01 motion control card	PCI-DMC-F01	PCI interface, 12 nodes (maximum of 6 axes (first 6 nodes))
DMCNET com.	Terminal Resistor	ASD-TR-DM0008	One is included with each card
	DMCNET communication cable 0.3 m	TAP-CB03	
	DMCNET communication cable 0.5 m	TAP-CB05	
	DMCNET communication cable 1.0 m	TAP-CB04	
	DMCNET communication cable	TAP-CB10	

	1.0 m		
	DMCNET communication cable 0.3 m	NC-CAB-DMC003	
	DMCNET communication cable 1.5 m	NC-CAB-DMC015	
	DMCNET communication cable 3.0 m	NC-CAB-DMC030	
	DMCNET communication cable 5.0 m	NC-CAB-DMC050	
	DMCNET communication cable 10.0 m	NC-CAB-DMC100	
Servo drive	A2-F series servo drive	ASD-A2-####-F	100~7500w
	M-F series servo drive	ASD-M-####-F	3-in-1 gantry servo drive, 100-2000w
Servo motor	ECMA series servo motor	ECMA-#########	Select the specific motor type based on the drive type
Remote module	Remote digital input module (32 points)	ASD-DMC-RM32MN	
	Remote digital input module (64 points)	ASD-DMC-RM64MN	Supports MPG
	Remote digital output module (32 points)	ASD-DMC-RM32NT	
	Remote digital output module (64 points)	ASD-DMC-RM64NT	
	Remote digital combination module (16 digital input, 16 digital output)	ASD-DMC-RM32PT	

	Remote analog input module (4 channels)	ASD-DMC-RM04AD	
	Remote analog output module (4 channels)	ASD-DMC-RM04DA	
	Remote pulse module (4 channels)	ASD-DMC-RM04PI	200KHz
Integrated extension module	Integrated node module	ASD-DMC-GA01	
	Extension digital input module (16 points)	ASD-DMC-GE16MN	Must be attached to ASD-DMC-GA01
	Extension digital output module (16 points)	ASD-DMC-GE16NT	Must be attached to ASD-DMC-GA01
	Extension pulse module (1 channel)	ASD-DMC-GE01PH	Must be attached to ASD-DMC-GA01; 4MHz

1.5 Software Development

■ Supported Operating System Versions

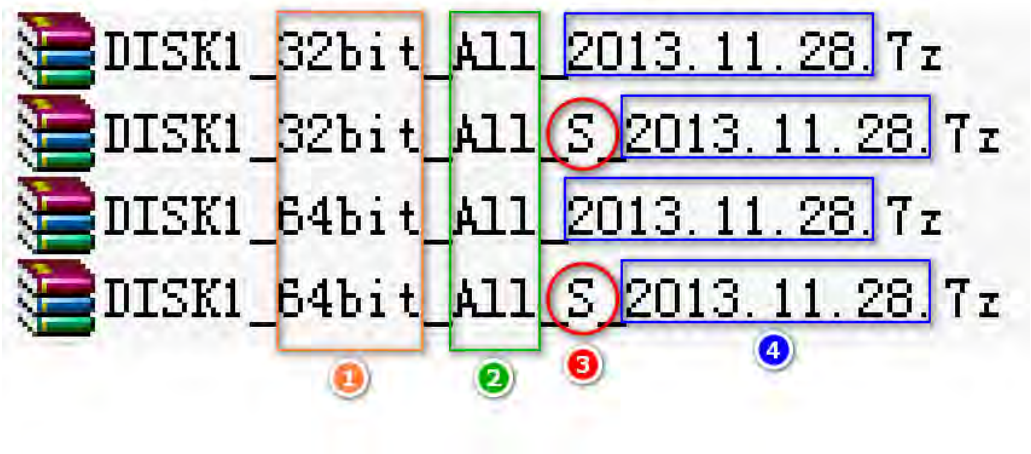
Supported operating systems: Microsoft Windows XP, Windows Vista, Windows 7, 32-bit and 64-bit operating systems.

■ Supports High-level Language Software Development Environments

Provides a standard DLL program library, and supports the following high-level language software development environments: Visual C++, Visual Basic, Borland C++ Builder, Delphi, Visual Basic.Net, Visual C#, etc.

■ Introduction to Software Development Packet

The software development packet typically consists of compression files as shown below.



❶ 32 bit: applicable to Windows 32 bit systems; 64 bit: applicable to Windows 64 bit systems.

❷ All: applicable to A01, B01, and F01 cards.

❸ With “S”: Includes only driver program; Without “S”: Includes driver program, manual, source code, EzDMC software.

❹ 2013.12.28: Issuance date of December 28, 2013.

DISK1_32bit_All_2013.11.28/ DISK1_64bit_All_2013.11.28 path after decompression:

EzDMC software program files\Delta Industrial Automation\PCI-DMC\app

Manual program files\Delta Industrial Automation\PCI-DMC\manual

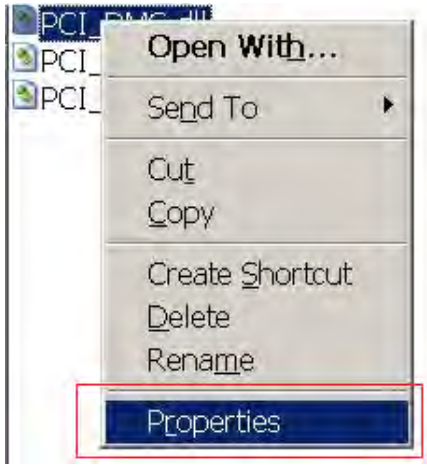
Example source codeprogram files\Delta Industrial Automation\PCI-DMC\samples

DLL path program files\Delta Industrial Automation\PCI-DMC\dll

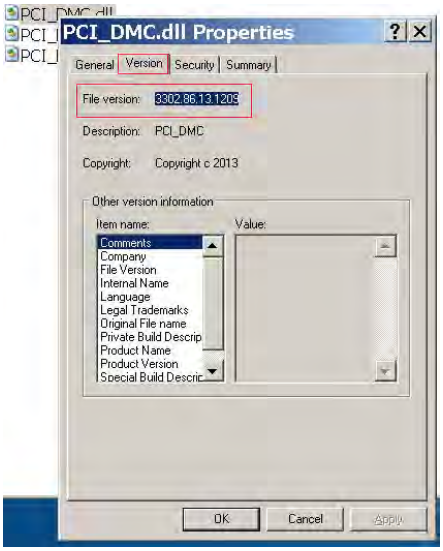
Check DLL File Version

Method 1 (DLL File Attribute)

1. Find the program files \Delta Industrial Automation\PCI-DMC\dll from the software installation path or software decompression path (the one that users choose)
2. Find the three documents: PCI_DMC.dll, PCI_DMC_01.dll, and PCI_DMC_F01.dll.
3. Press the right mouse button while highlighting a document (PCI_DMC.dll is used as an example here), and select [Attributes] from the menu that appears.

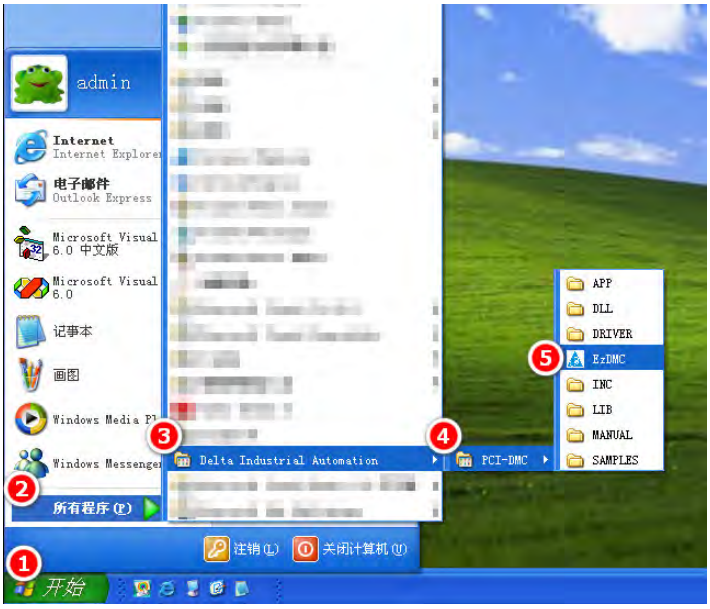


Select the [Version] tab bar from the [Attributes] dialog box for PCI_DMC.dll, and you will see the file version marked by a red bracket below. The file version of PCI_DMC.dll is: 3302.86.13.1127.

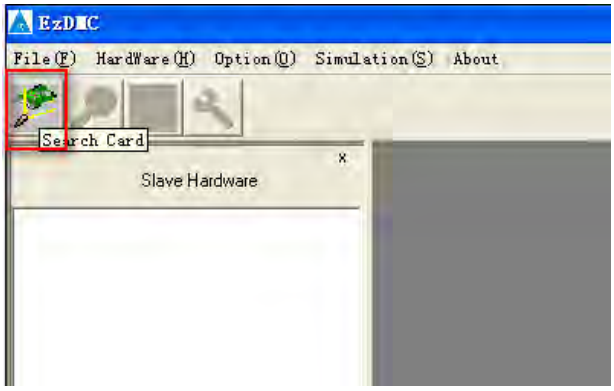


- **Method 2 (EzDMC Software)**

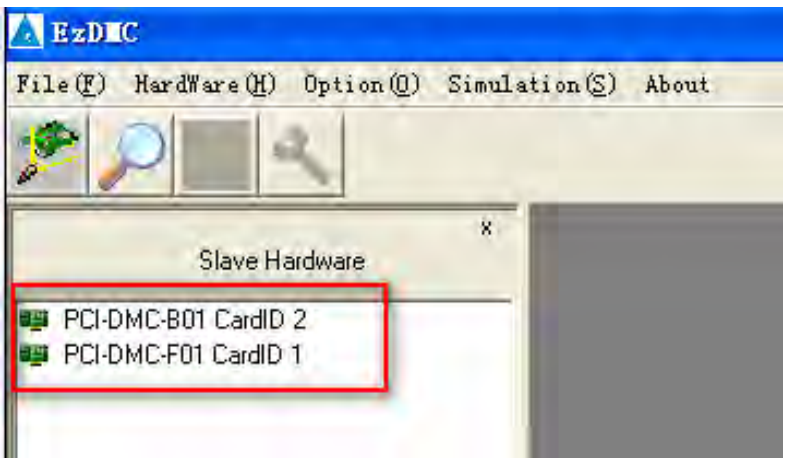
1. Click on the Start menu, select All Programs, select Delta Industrial Automation, select PCI-DMC, and select EzDMC to launch the EzDMC software.



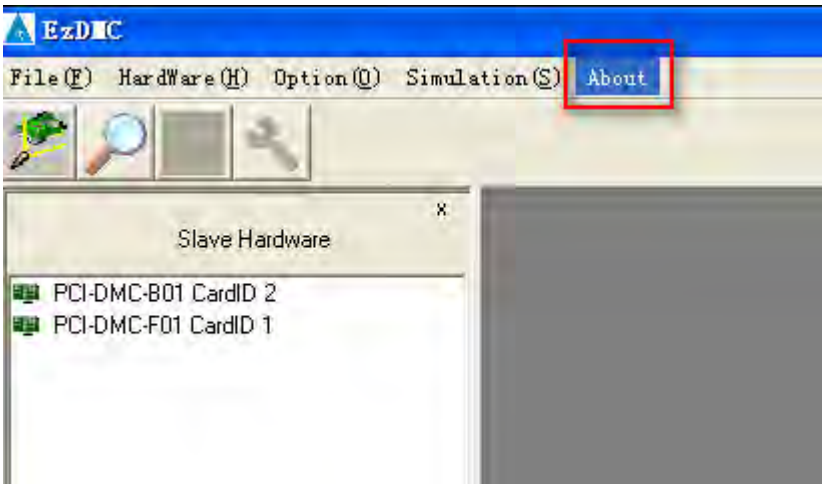
2. Click on [Search Card] in the toolbar to search for cards.



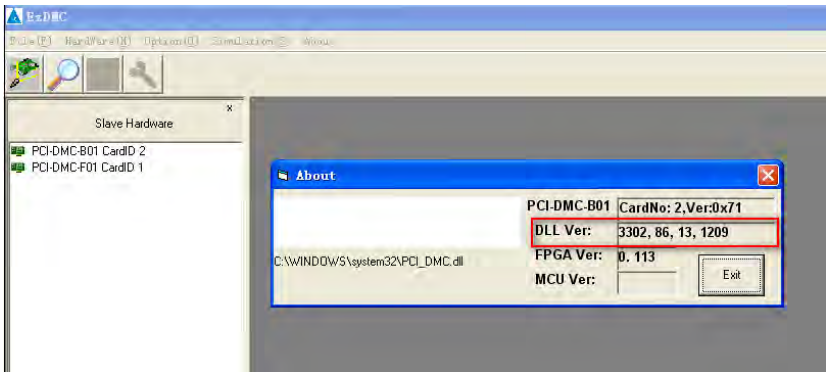
3. A list similar to that shown below will appear after the cards have been found.



4. Click on About in the menu.



5. You can now see the file version, which is marked by a red bracket below, in the {About} dialog box. The file version of PCI_DMC.dll is: 3302, 86, 13, 1209.



■ Example of a New Project

● Visual C++6.0 Environment

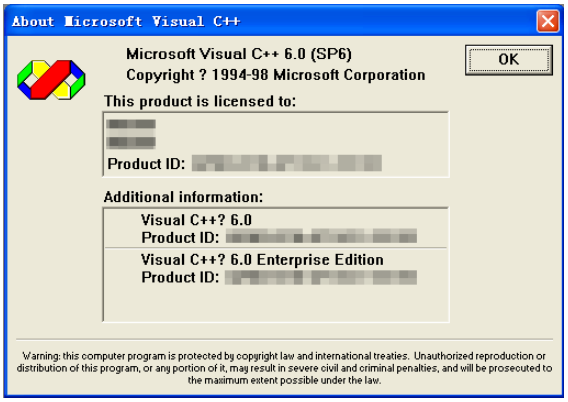
Operating steps for a new axis card project (dialog-based) in the VC++ development environment

Summary:

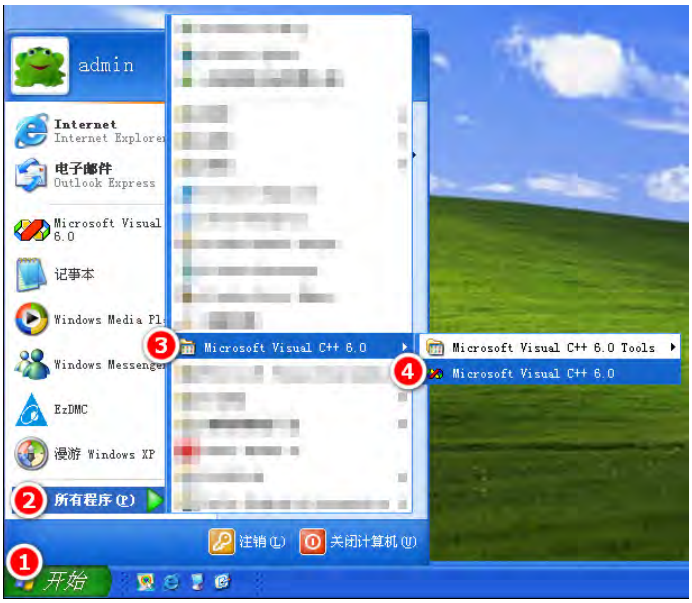
1. A new dialog-based style project.
2. Copy PCI_DMC.H, PCI_DMC_Err.h, TYPE_DEF.H to the root directory.
3. Copy PCI_DMC.lib to the root directory.
4. Add “PCI_DMC.lib” to {Project Settings} {Link} [Object/library modules:] in the VC environment.
5. Add a reference to PCI_DMC.H and PCI_DMC_Err.h in the .cpp source code document.
6. Add Button1 button and write code, creating the program.

Detailed steps:

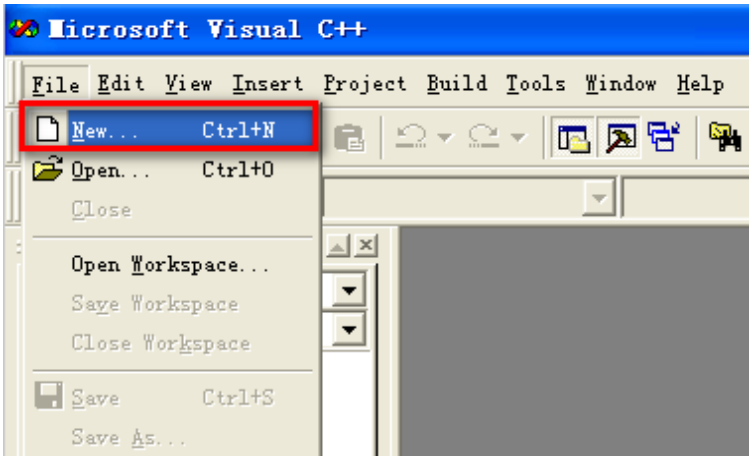
VC++ version: Visual C++ 6.0 (SP6), Enterprise Edition.



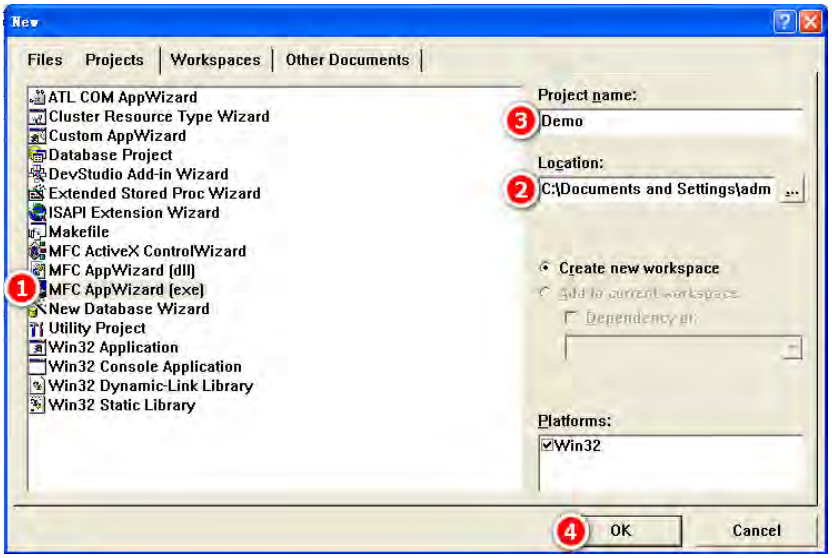
Step 1: Click on the Start menu, select All Programs, select Microsoft Visual C++ 6.0, and select Microsoft Visual C++ 6.0 to launch the VC++ software.



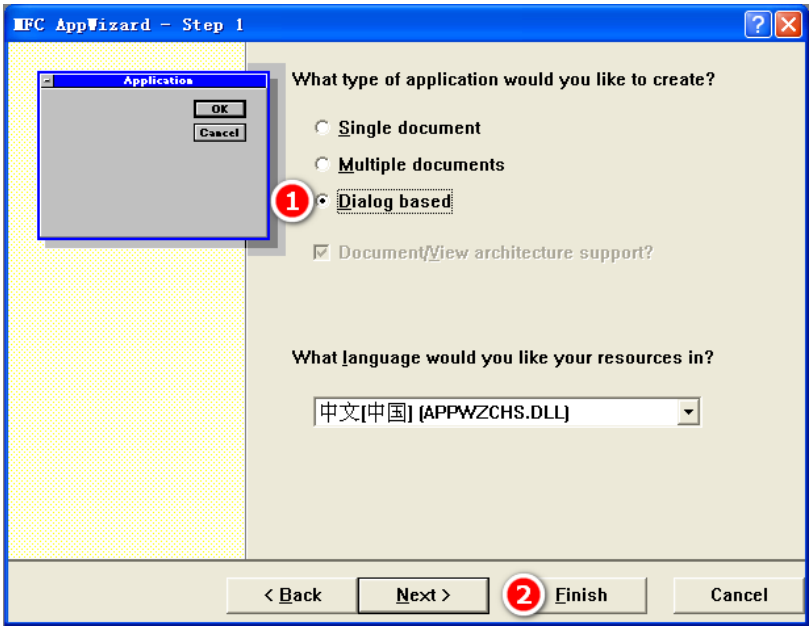
Step 2: Click on the "New…" item in the File menu.



Step 3: Select the item [MFC AppWizard (exe)] from the checklist on the left side of the {Projects} tab menu in the {New} dialog box, and select the storage path (C: \Documents and Settings\admin\desktop in this example) in [Location:]. Input the project name in [Project name:] (Demo in this example), and click on [OK].

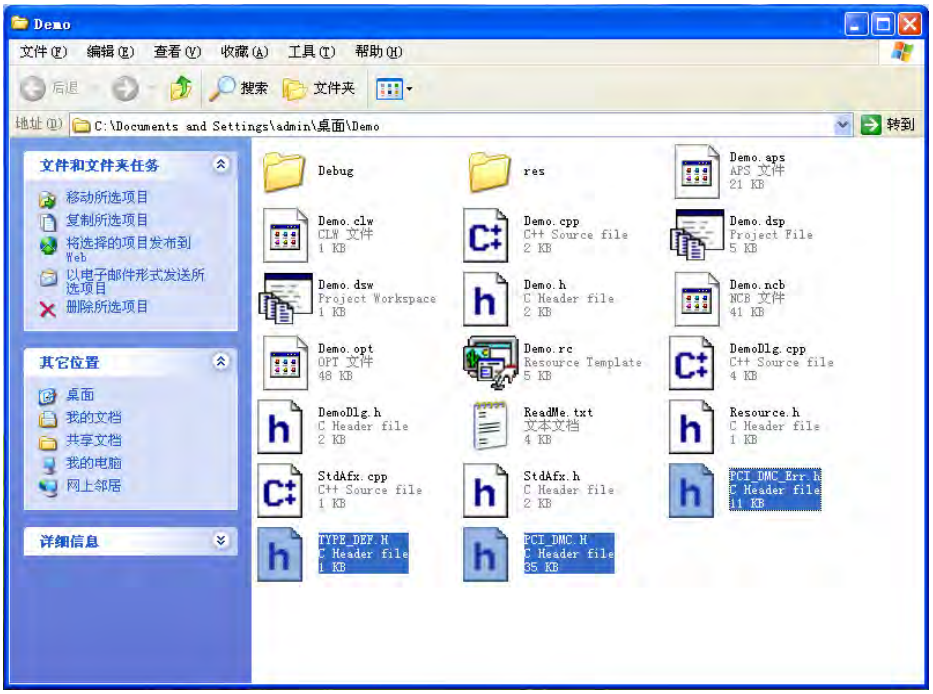


Step 4: Select the [Dialog-based] in {MFC AppWizard – Step 1} dialog box, and click on [Finish].

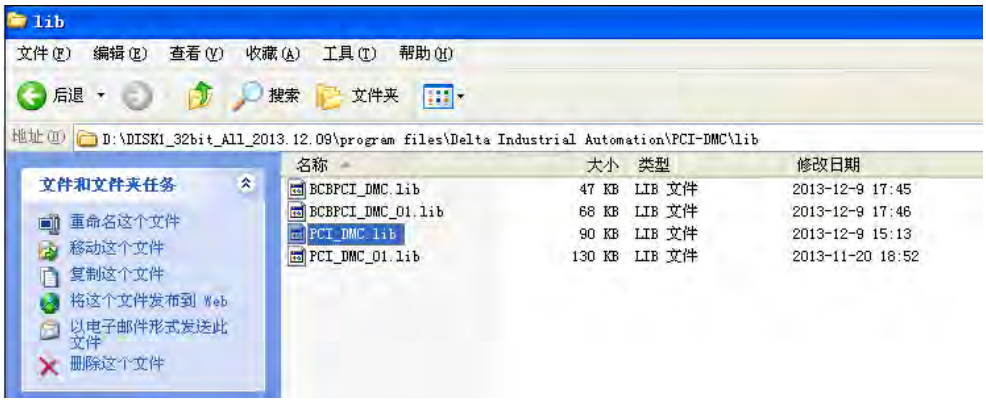


Step 5: Copy the three documents: PCI_DMC.H, PCI_DMC_Err.h, TYPE_DEF.H on the path program files\Delta Industrial Automation\PCI-DMC\inc\VC.

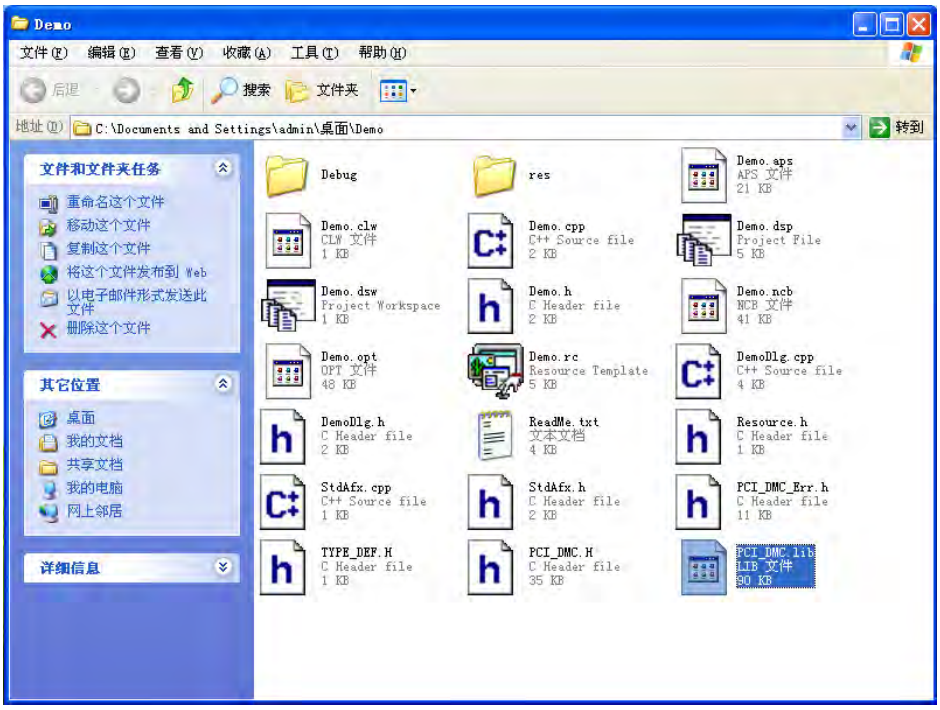
Step 6: Paste to the root directory (location of .dsw files).



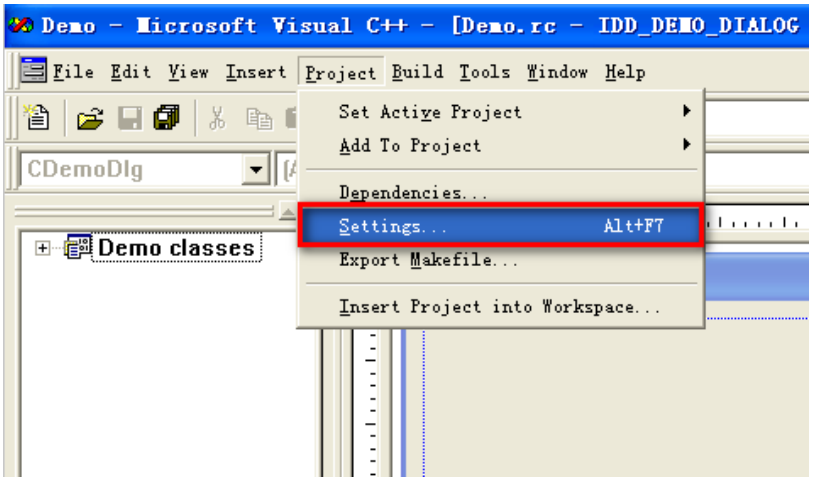
Step 7: Copy the PCI_DMC.lib document from program files\Delta Industrial Automation\PCI-DMC\lib.



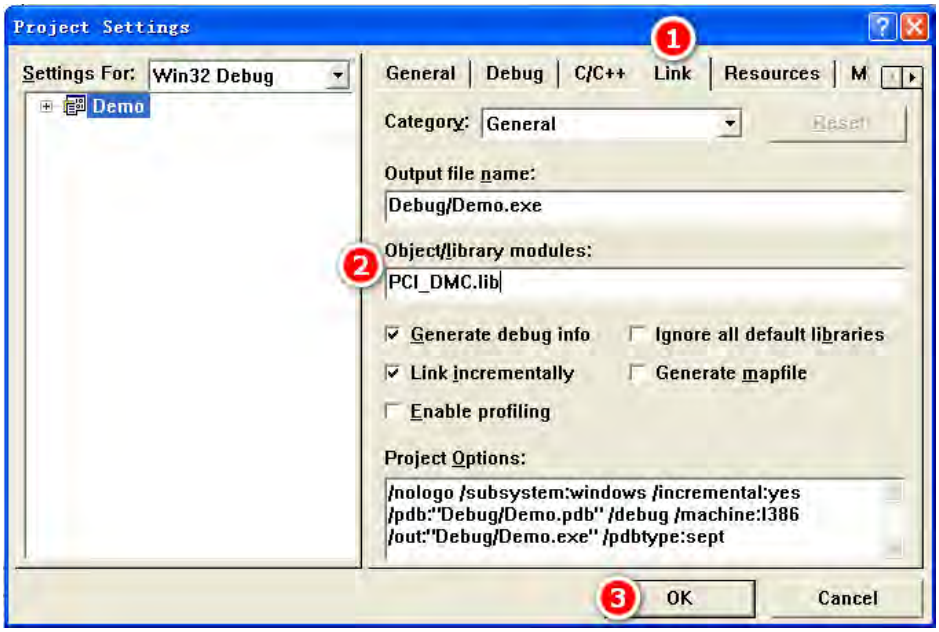
Step 8: Paste to the root directory (location of .dsw files).



Step 9: Click on Settings in Project menu.



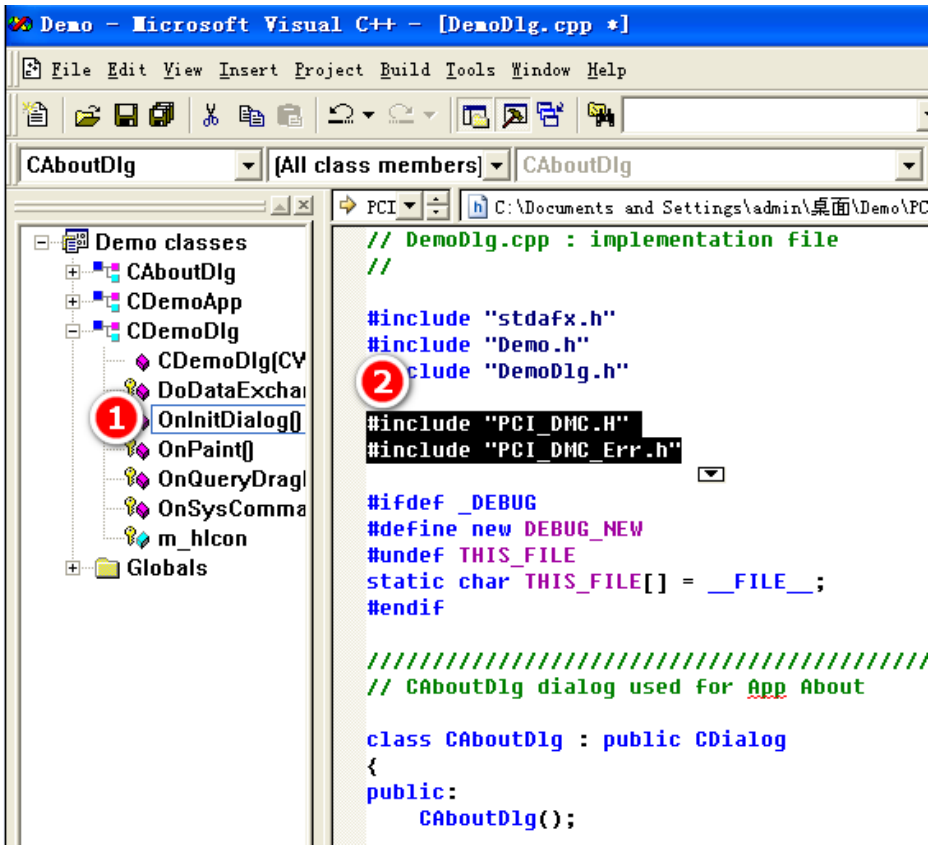
Step 10: Select the {Link} tab menu in {Project Settings} dialog box that has appeared, input “PCI_DMC.lib" to [Object/library modules:], and click on [OK].



Step 11: Double click on the [OnInitDialog ()] node. When [DemoDlg.cpp] is opened, the following two included items will be added to the top of the source code file.

#include "PCI_DMC.H"

#include "PCI_DMC_Err.h"



Step 12: In [Controls] toolbox, drag [Button] to the [Demo edit area].

Step 13: Double click on [Button1], and click on [OK] in the {Add Member Function} dialog box.

Step 14: Input the following code in the void CDemoDlg:: OnButton1 () function:

I16 rt;

I16 CardCounts=0;

//initialize system resources and access the total number of axis cards installed on the computer.

rt = _DMC_01_open(&CardCounts);
if (rt!=0)

{

//If opening _DMC_01_ fails, a message box will appear.

MessageBox("_DMC_01_open Error!");

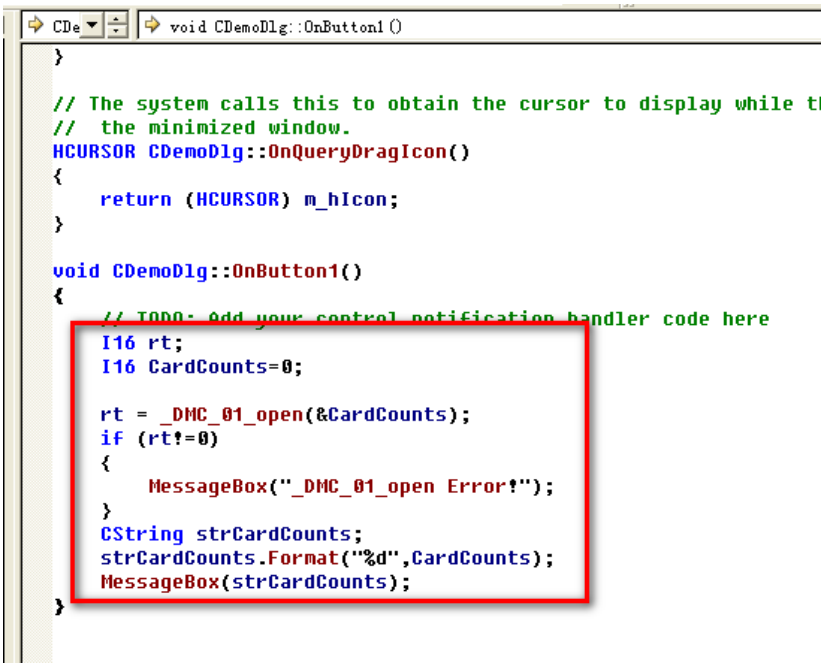
}

CString strCardCounts;

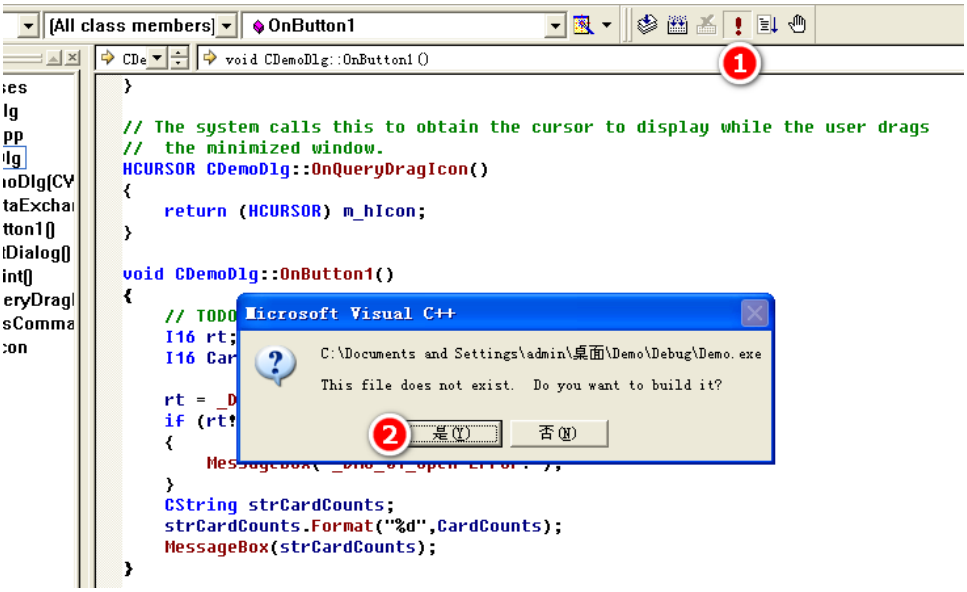
strCardCounts.Format("%d",CardCounts);

MessageBox(strCardCounts);//The message box that appears will display “Total number of axis cards.”

ber of axis cards.”



Step 15: Click on [Exclamation point], and click on [Yes] in {Microsoft Visual C++} dialog box that appears.



Step 16: In the [Demo] software that has been created, click on [Button1] to call up the {Demo} dialog box. The area marked with a red bracket will display the total number of axis cards installed on the computer (2 axis cards have been installed in this example).

- **Visual Basic 6.0 Environment**

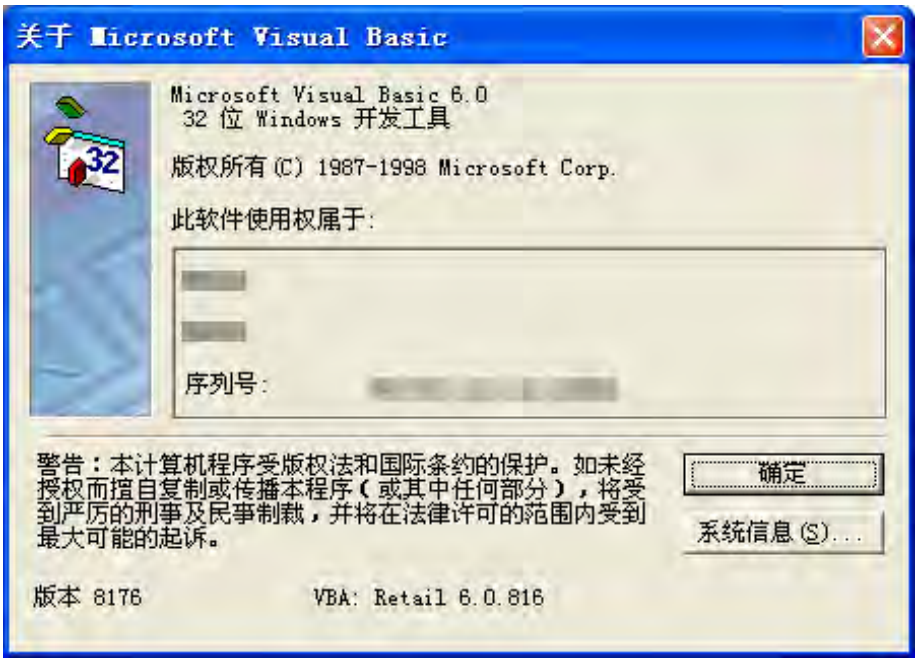
  Operating steps for creating a new environment of motion control card in Visual Basic 6.0:

  Summary:

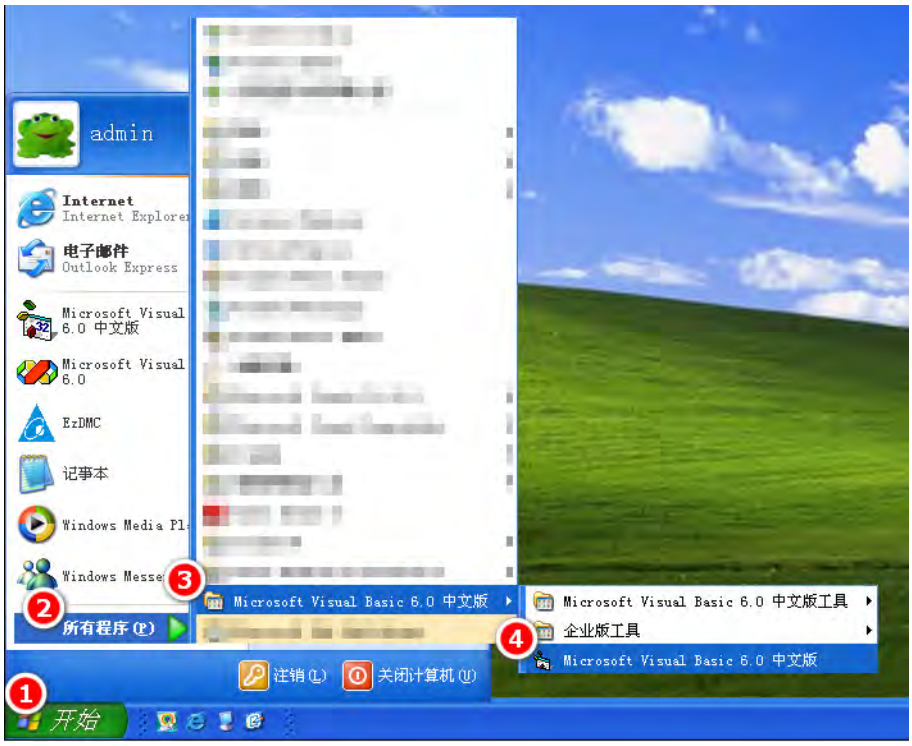
  1. Create a new project.
  2. Copy PCI_DMC.bas, PCI_DMC_Err.bas to the root directory.
  3. Use Project (P) and Add (A)… to incorporate PCI_DMC.bas and PCI_DMC_Err.bas.
  4. Add the button Command1 and write code to create the program.

  Detailed steps:

  VB version: Visual Basic 6.0



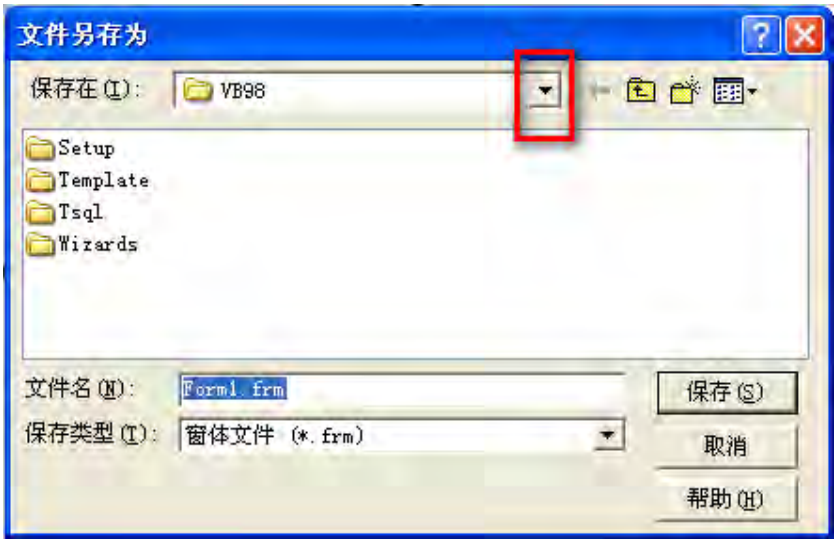
Step 1: Click on the Start menu, select All Programs, select Microsoft Visual Basic 6.0, and select Microsoft Visual Basic 6.0 to launch the VB software.



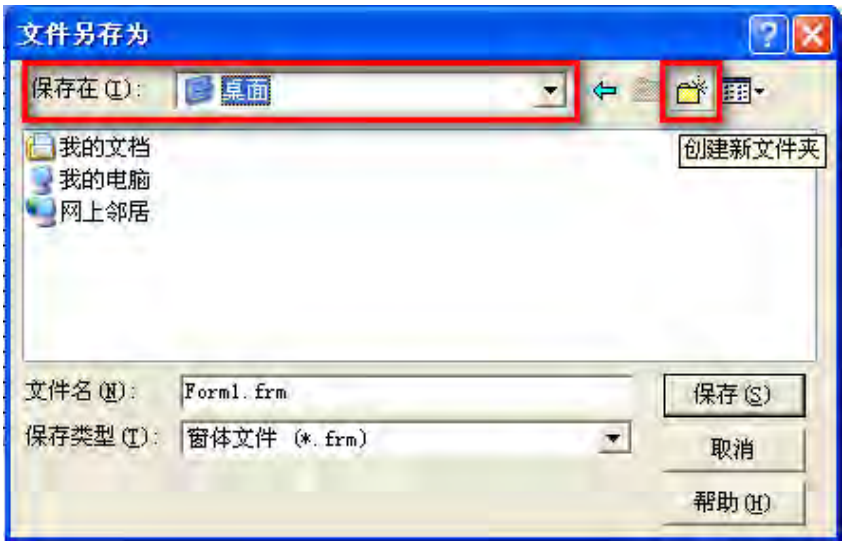
Step 2: While still in the “standard EXE” state in the {New} tab menu of the {New Projec t} dialog box, click on [Open].

Step 3: In the main window that appears, click on Save project (V) in the File (F) menu.

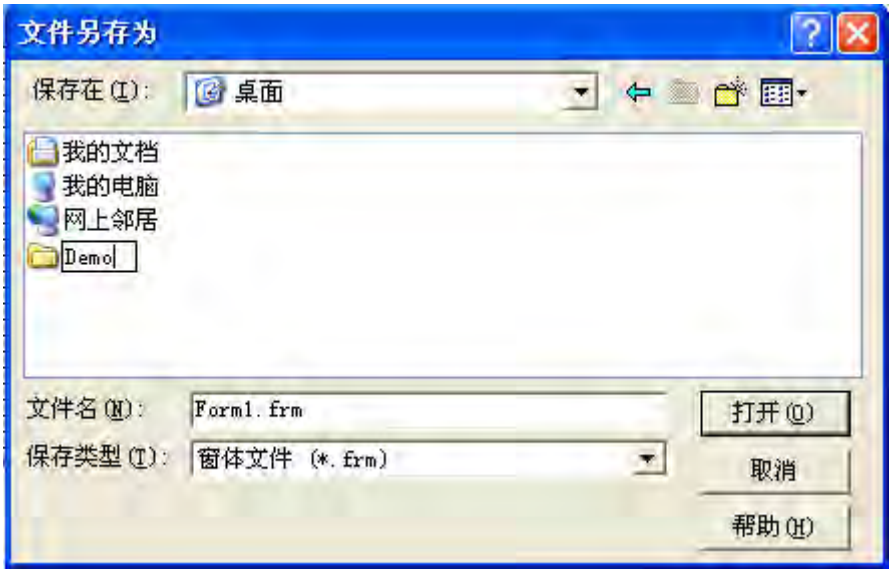
Step 4: In the {S ave as} dialog box, click on the downward-pointing arrowhead after “Save in (I): ” and browse to “Desktop."



Click on “Create new folder”.

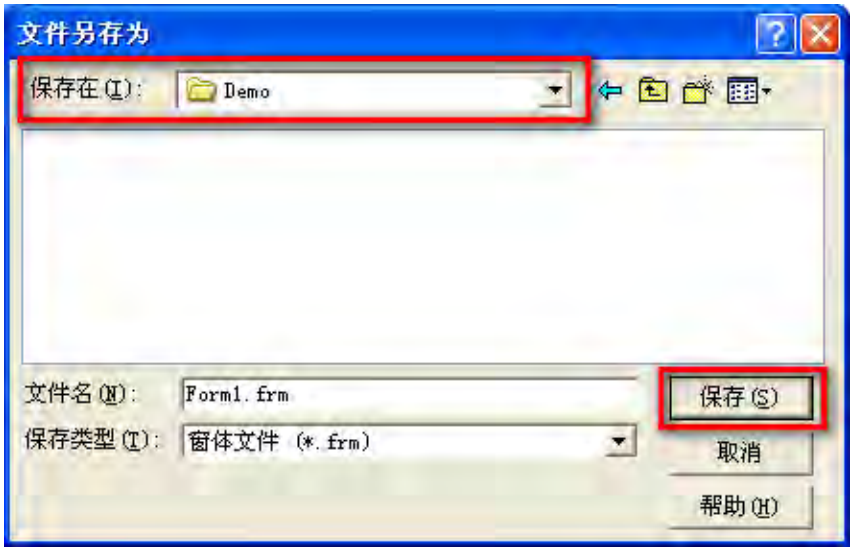


Change the name of the folder to “Demo."

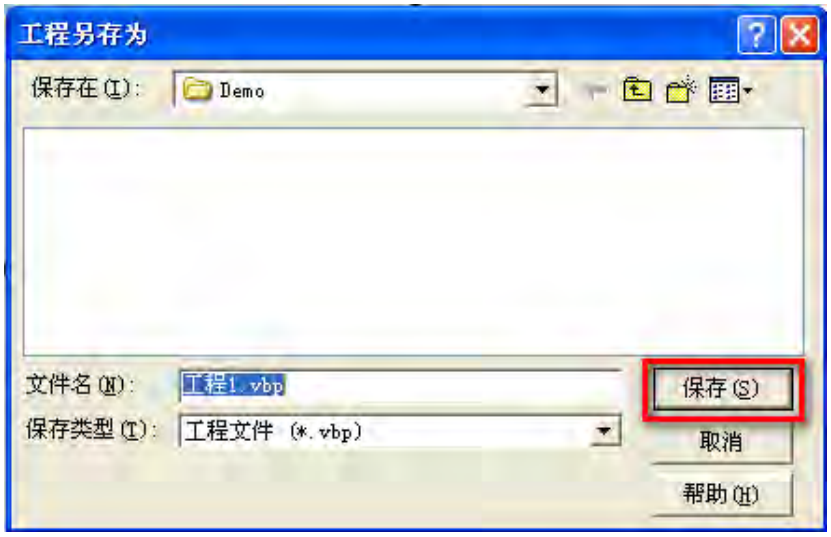


Double click “Demo” folder icon.

Click on [Save] to save “Form1.frm."



Click on [Save] to save “Project 1.vbp.”

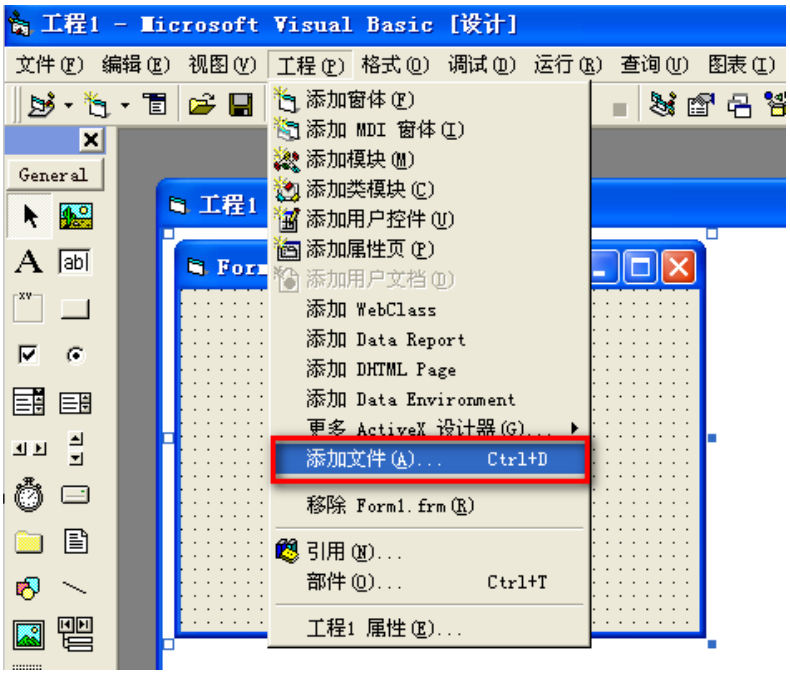


Confirm that “Form1.frm” and “Project 1.vbp” have been saved in the location “C: \Documents and Settings\admin\desktop\Demo.”

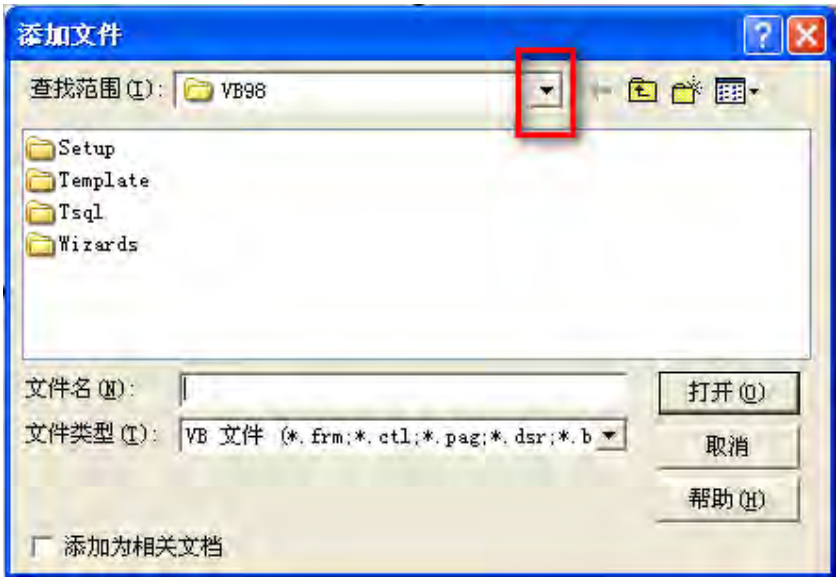
Step 5: Copy the two documents PCI_DMC.bas and PCI_DMC_Err.bas from program files\Delta Industrial Automation\PCI-DMC\inc\VB.

Step 6: P Paste to the root directory.

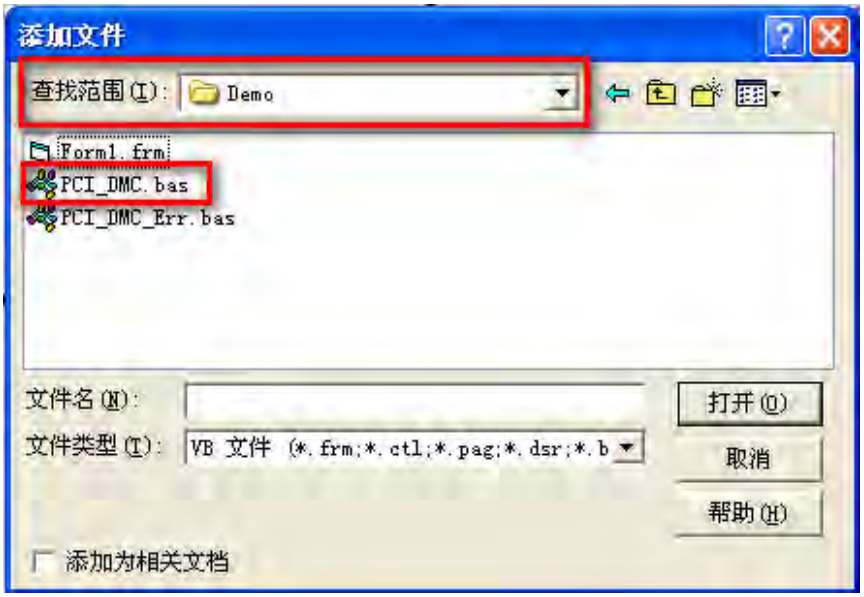
Step 7: Click on Add file (A) in Project (P) menu.



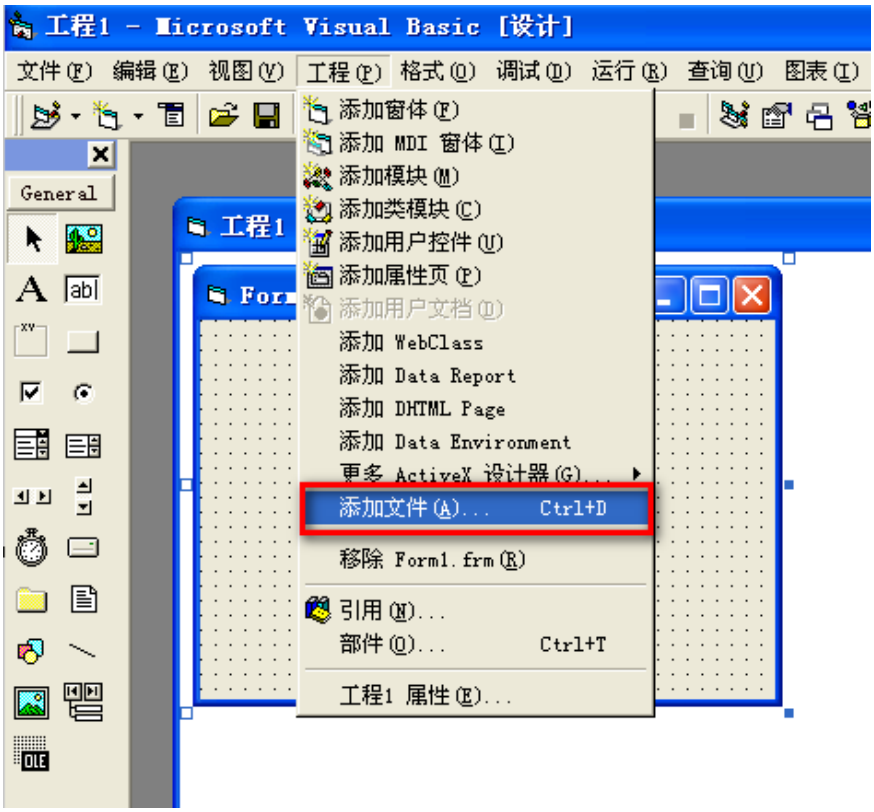
Step 8: In the {Add file} dialog box that appears, click on the downward-pointing arrowhead after “Find range (I): ” and browse to the “Demo” folder.



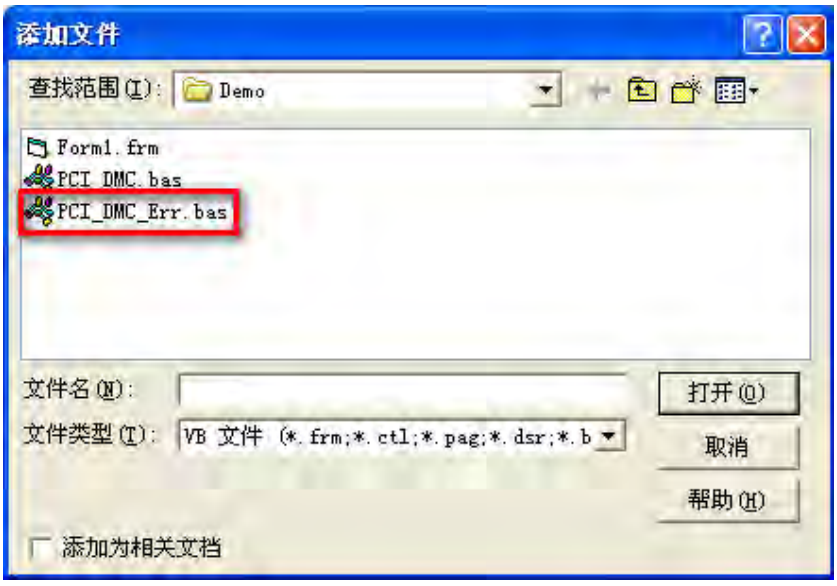
Double click on “PCI_DMC.bas" to add it to the project.



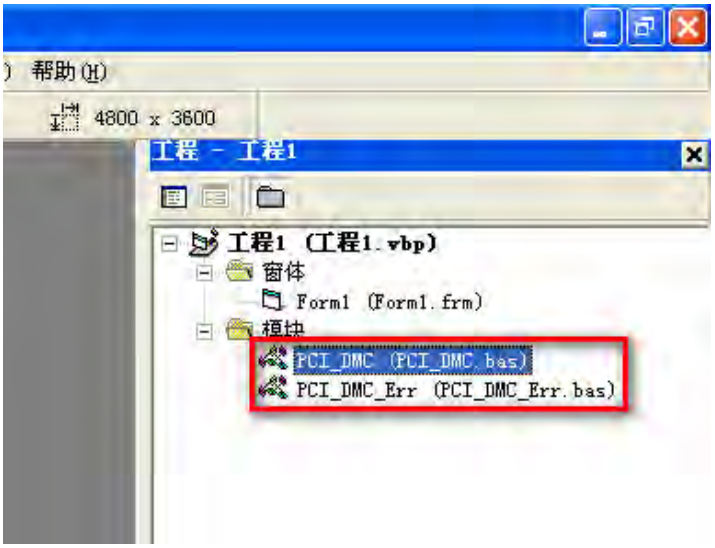
Step 9: Click again on the item Add file (A) in Project (P) menu.



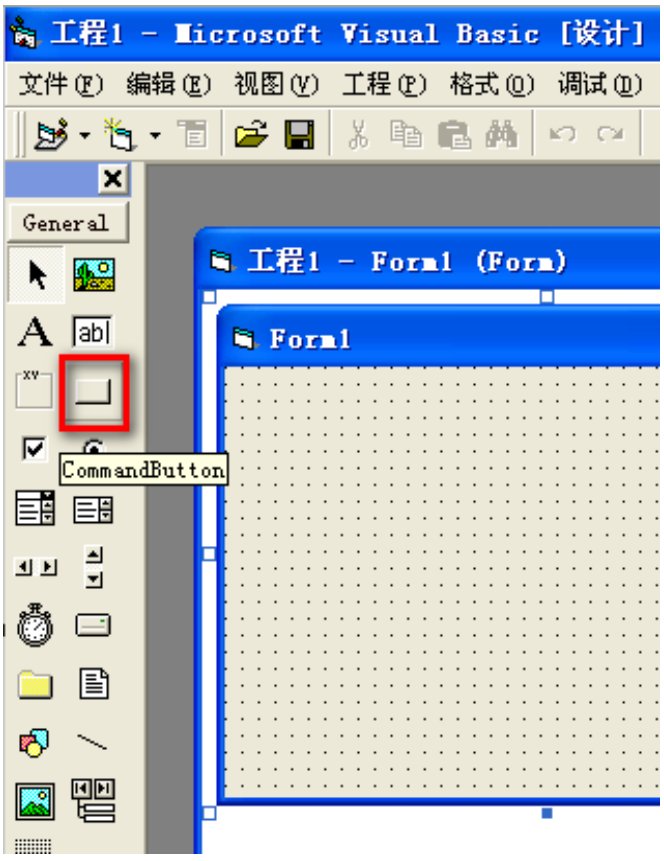
Step 10: In the {Add file} dialog box that appears, double click on the "PCI_DMC_Err.bas" to add it to the project.



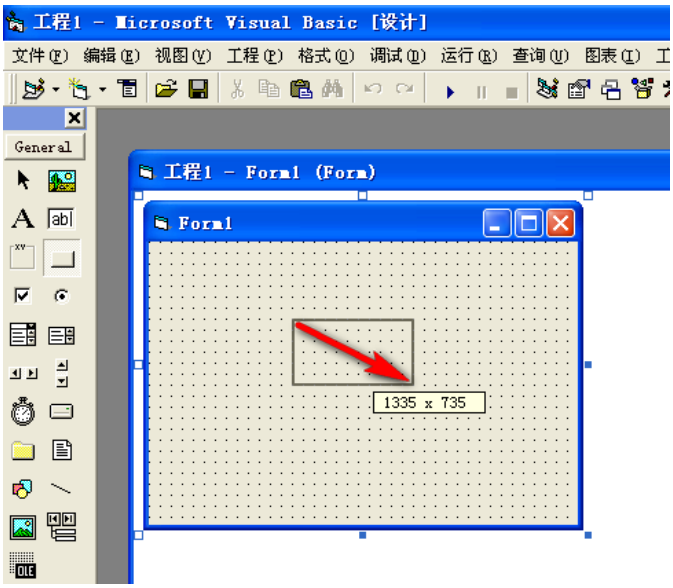
Step 11: Confirm that the two documents “PCI_DMC.bas" and "PCI_DMC_Err.bas” have been added to the [Module] node of the {Project resources manager}.



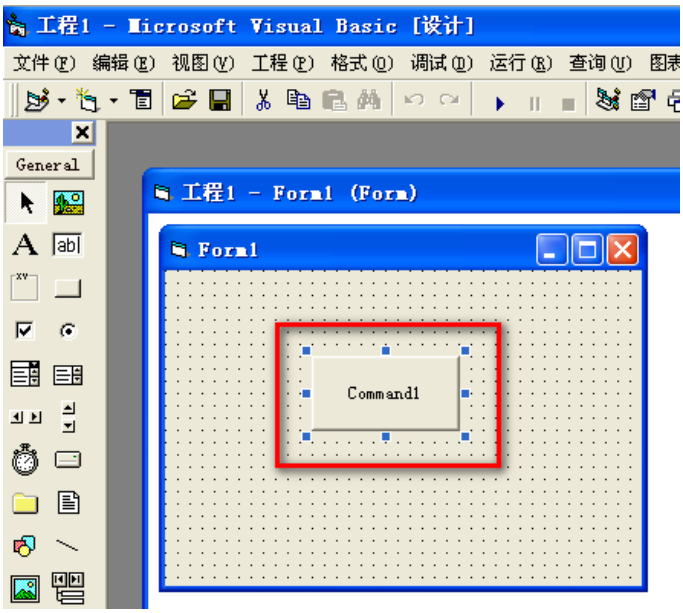
Step 12: Click on [CommandButton] in the toolbox.

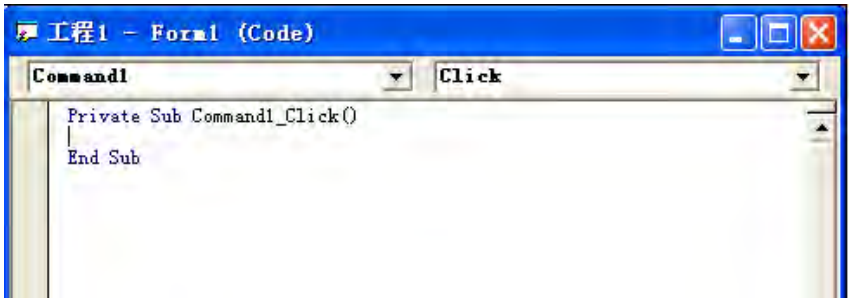


Use the right mouse button to drop and drag in Form1, creating the “Command1” button.



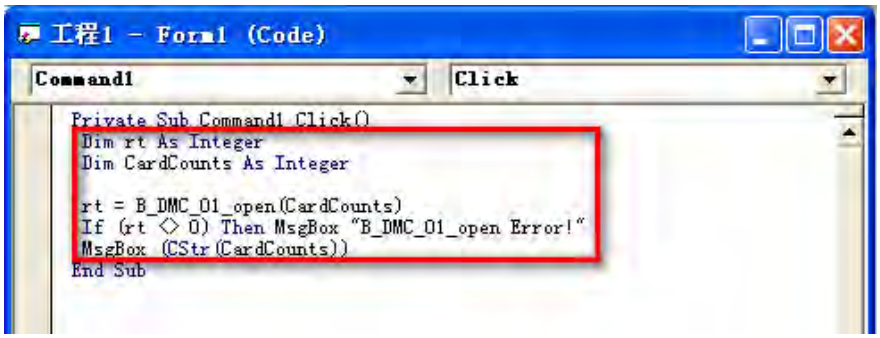
Step 13: Double click on [Command1] to open the code editing Windows.



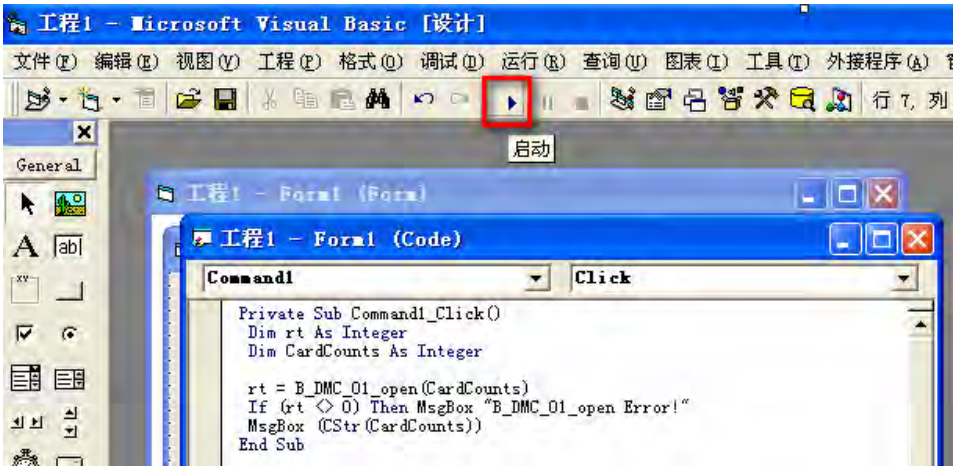


Step14: Input the following code in Private Sub Command1_Click ().

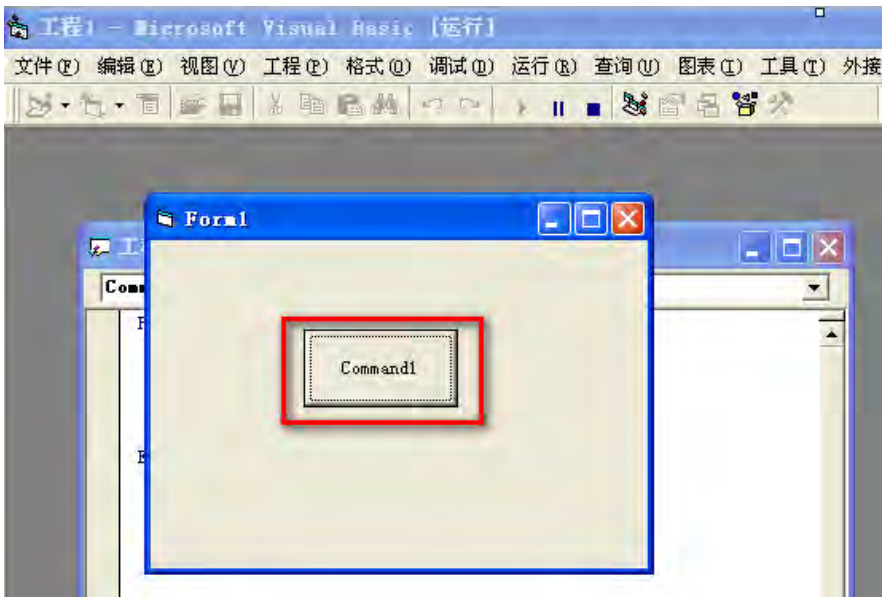
```
Dim rt As Integer
Dim CardCounts As Integer

rt = B_DMC_01_open(CardCounts)
If (rt <> 0) Then MsgBox "B_DMC_01_open Error!"
MsgBox (CStr(CardCounts))
```

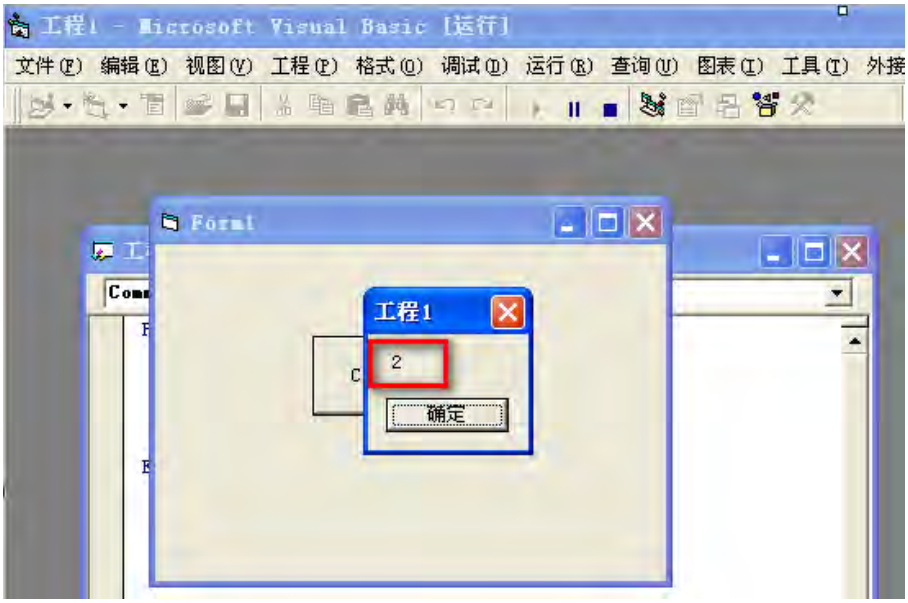


Step 15: Click on the [Right arrow] to create the program.



Step 16: Click on [Command1] in the newly-created {Form1} program to bring up the {Project 1} dialog box. The total number of axis cards installed on the computer will be displayed in the area marked with a red bracket (two axis cards have been installed in this example).

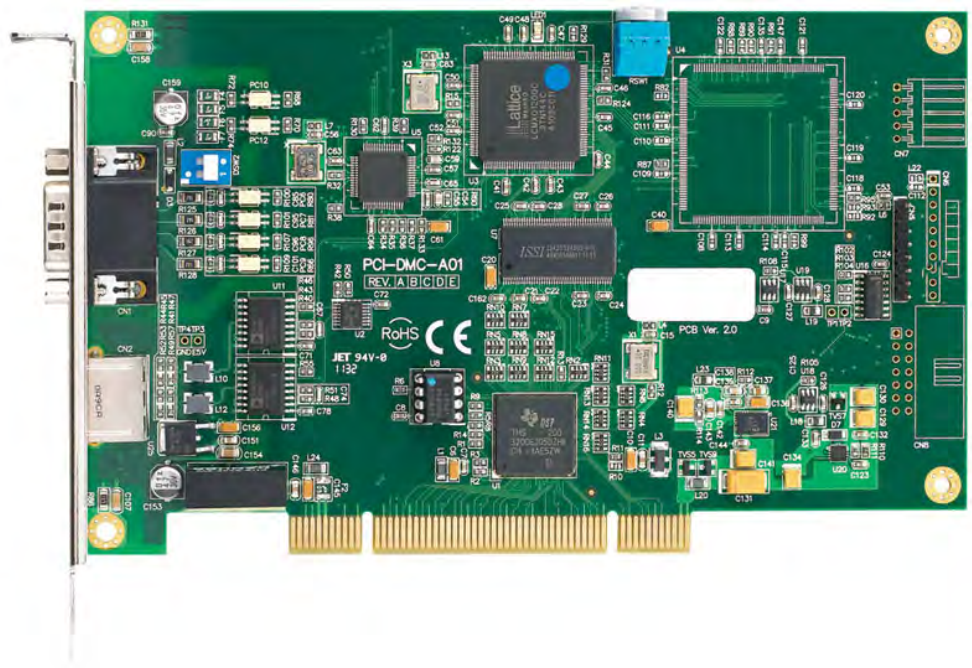




Chapter 2 Card

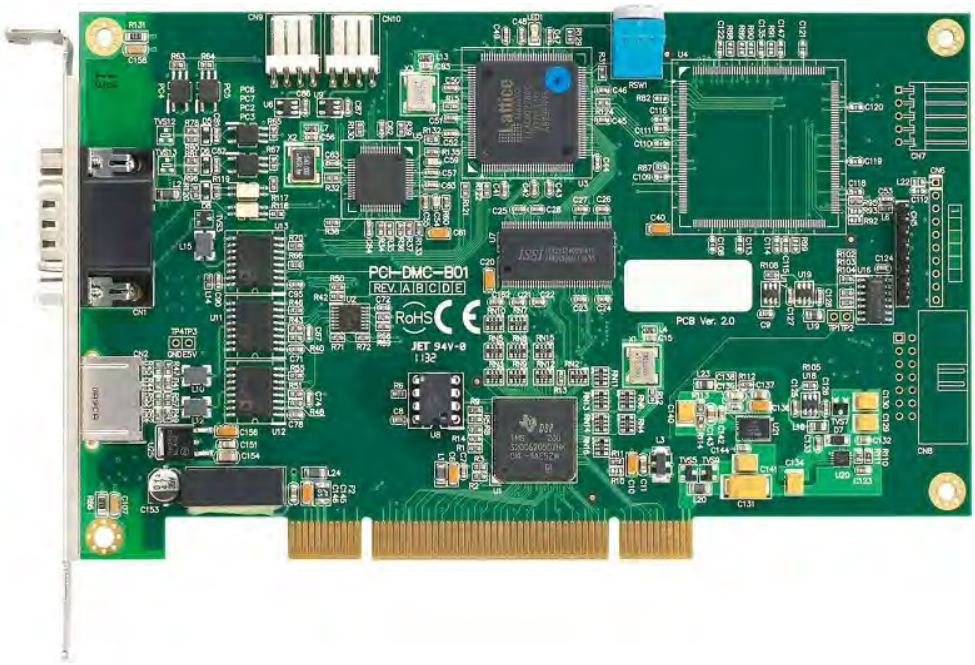
2.1 Introduction to A01/B01/F01 Cards and Their Differences

- **A01**



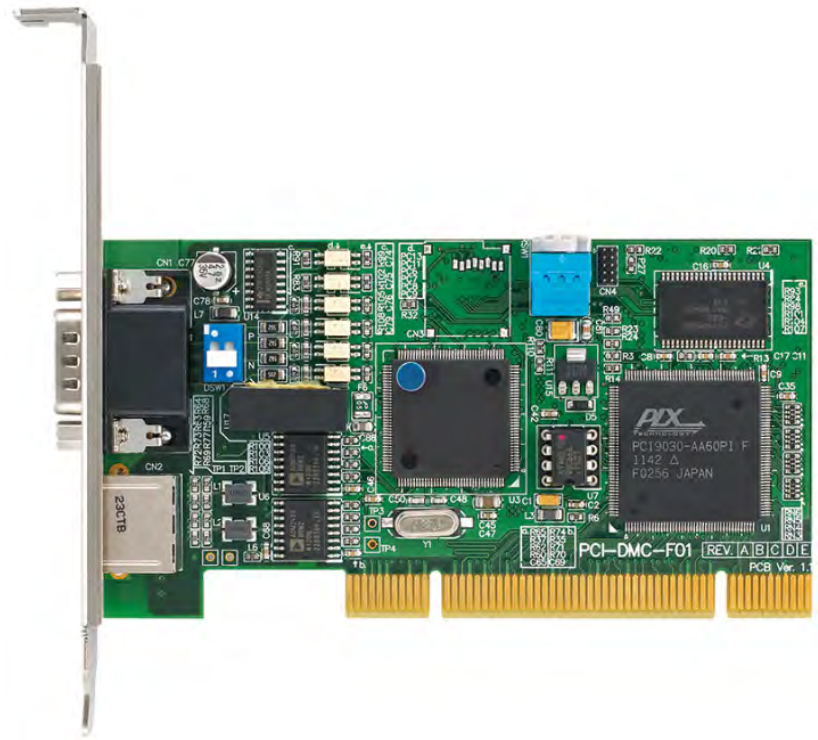
The A01 motion control card can be connected to 12 slaves, and each slave can have one servo motion axis (i.e., each card can operate 12 servo axes) or one 4-channel pulse output module can operate 4 stepper axes (i.e., each card can operate 48 stepper axes).

■ B01



The B01 motion control card can be connected to 12 slaves, among and each slave can have one servo motion axis (i.e., each card can operate 12 servo axes) or one 4-channel pulse output module can operate 4 stepper axes (i.e., each card can operate 48 stepper axes). The B01 card provides a position compare and triggering function.

■ F01



The F01 motion control card can be connected to 12 slaves. If there is an axis of motion (servo slave, pulse module slave), the node number must be in the range of 1 ~ 6.

■ A01/B01/F01 Comparative Table

Item	A01	B01	F01
Among the 12 slaves, the node number which can connect to the servo	Any node	Any node	First 6 nodes
DB15 terminal, number of DI points	8	1	8
DB15 terminal, number of DO points	4	1	4
Position compare and trigger		Yes	
Interval between two motion commands	Yes	Yes	
Set group	Yes	Yes	
Interrupt model	Yes	Yes	
Speed Continue	Yes	Yes	
Mechanical compensation	Yes	Yes	

2.2 A01/B01 Card No. Setting and Slave No. Setting Rules

Card number setting: Turn the dial until the arrow points to the needed card number.

Card number setting rules:
Permissible range of card numbers: 0 ~ 9, A (10), B (11), C (12), D (13), E (14), F (15);
A card number cannot be repeated on the same industrial control computer.

Node number setting rules for slaves connected with card:
Permissible node number range: 1~12;
There must be a node 1;
A node number cannot be repeated on the same card.

2.3 F01 Card No. Setting and Slave No. Setting Rules

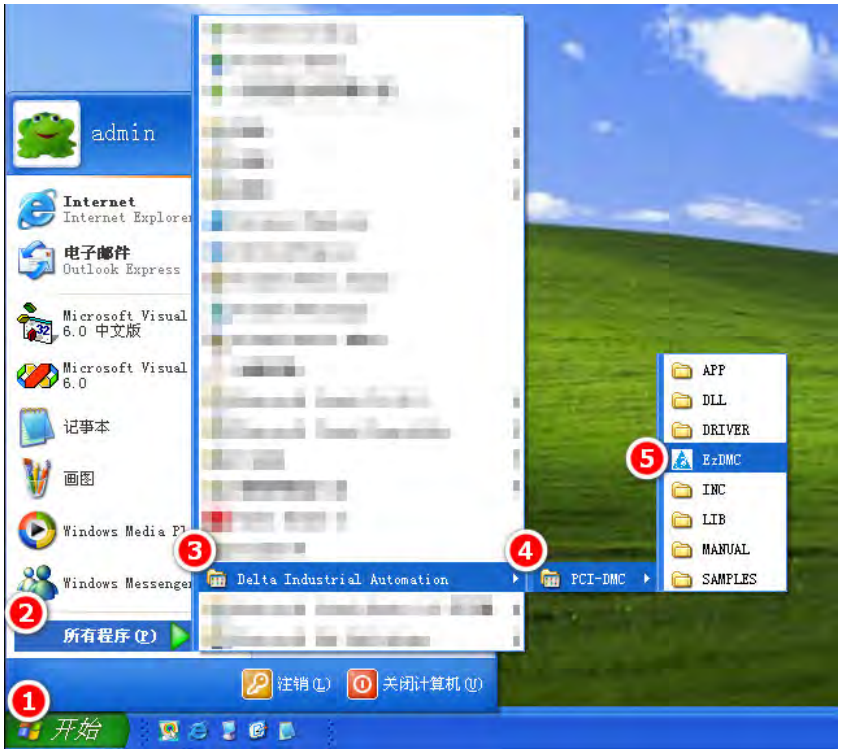
Card number setting: Turn the dial until the arrow points to the needed card number.

Card number setting rules:
Permissible range of card numbers: 0 ~ 9, A (10), B (11), C (12), D (13), E (14), F (15);
A card number cannot be repeated on the same industrial control computer.

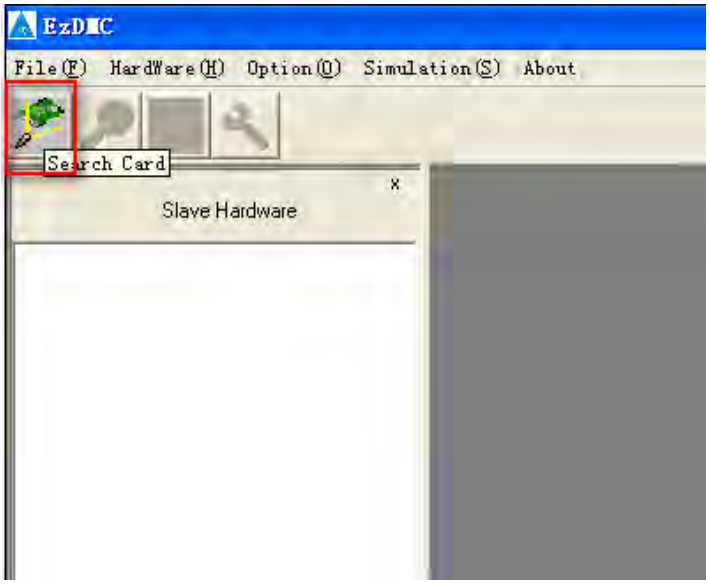
Node number setting rules for slaves connected with card:
Permissible node number range: 1 ~ 12, and the permissible node number to connect to the servo: 1 ~ 6;
There must be a node 1;
A node number cannot be repeated on the same card.

2.4 View Card Version (EzDMC Software)

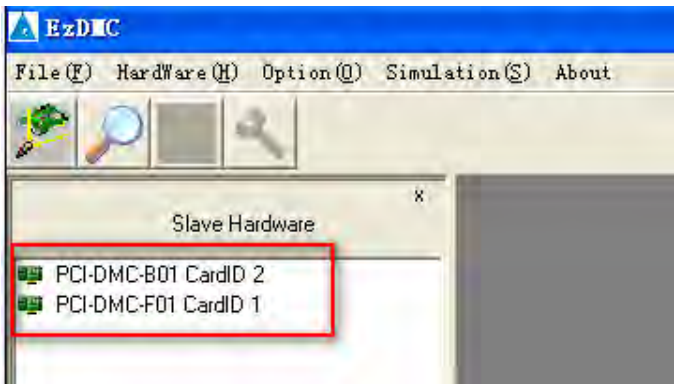
Step 1: Click on the Start menu, select All Programs, select Delta Industrial Automation, select PCI-DMC, and select EzDMC to launch the EzDMC software.



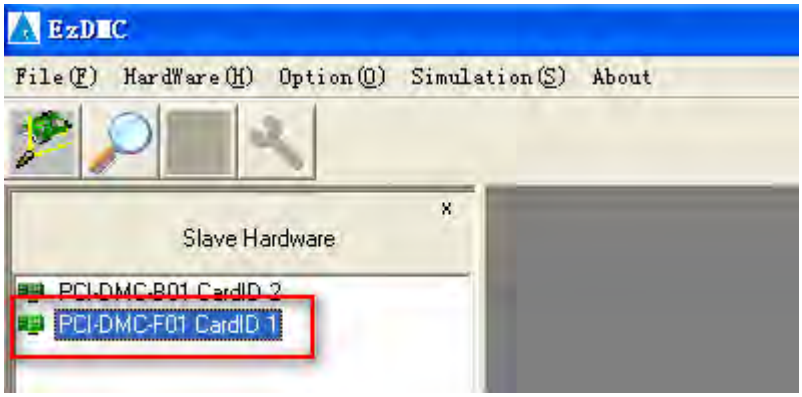
Step 2: Click on [Search Card] in the toolbar to search for cards.



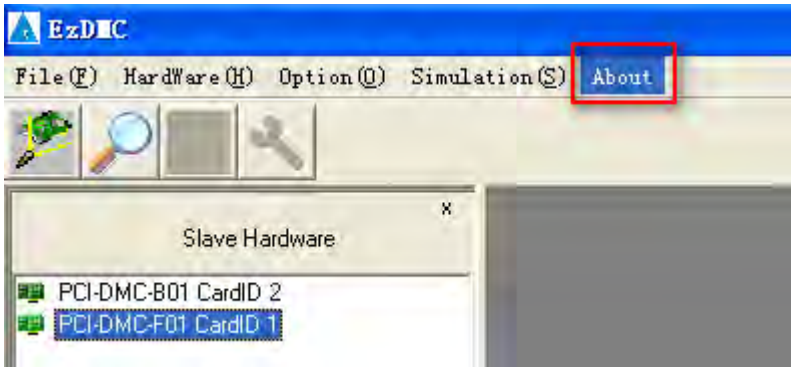
Step 3: A list similar to that shown below will appear after the cards have been found.



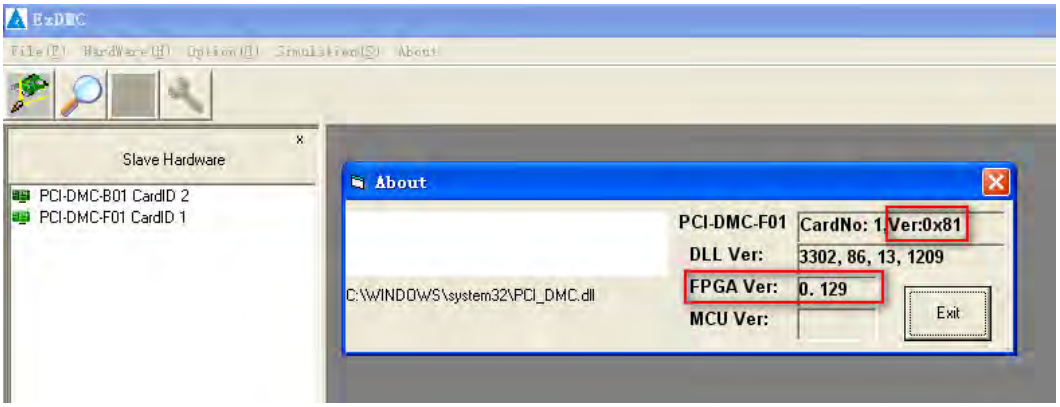
Step 4: Click on the name of the card you wish to view (this example uses PCI-DMC-F01 CardID 1), as shown below.



Step 5: Click on About in the menu.



Step 6: In the {About} dialog box, you will see the "card version" and "card FPGA version" information indicated below marked with a red bracket. The card version of PCI-DMC-F01 CardID 1 in this example is 0x81, and the card FPGA version is 0.129.



(This page is intentionally left blank.)

Chapter 3 Servo

3.1 Servo ASD-A2####-F Series

3.1.1 Node Number Setting

Parameter P3-00 of the servo drive: station number.

Permissible setting range of P3-00: When connected to card A01/B01: 1 ~ 9, A (10), b (11), C (12); when connected to card F01: 1 ~ 6.

Parameter setting steps:

Step 1: Enter parameter mode: Press [MODE] repeatedly, until the LED segment display shows "P# - **" (#, *indicates a number or character).

Step 2.1: Specify parameter group: Press [SHIFT] repeatedly, until the "#" position shows the desired value.

Step 2.2: Specify parameter sequence number: Press Up or Down repeatedly, until "**" increases or decreases to the desired value.

Step 3: Enter parameter "view/edit" state: Press [SET], and a position in the LED segment display will start flashing.

Step 4: Change parameter value: Press [SHIFT] to switch to the flashing position. Press the Up or Down to increase or decrease the parameter value.

Step 5: Save and exit: Press [SET], and the LED segment display will automatically switch to "P# - **" word. (Exit without saving: Press [MODE], and the LED segment display will automatically switch back to "P# - **" word.)

Servo drive parameter viewing step:

Step 1: Enter parameter mode: Press [MODE] repeatedly, until the LED segment display shows "P# - **" (#, *indicates a number or character).

Step 2.1: Specify parameter group: Press [SHIFT] repeatedly, until the "#" position shows the desired value.

Step 2.2: Specify parameter sequence number: Press the [Up arrow] or [Down arrow] repeatedly, until "**" increases or decreases to the desired value.

Step 3: Enter parameter "view/edit" state: Press [SET], and a position in the LED segment display will start flashing.

Step 4: Exit: Press [MODE], and the LED segment display will automatically switch back to "P# - **" word

Restore servo drive parameter defaults

Step 1: Set parameters P2-08 to 10.

Step 2: Restart the device.

Clear AL013 (servo parameters are default settings, CN1 interface is not linked with a limit switch)

Note: This is a normal phenomenon.

Step 1: Set parameters P2-17 to 121.

Step 2: Restart the device.

Clear AL015 (servo parameters are default settings, CN1 interface is not linked with a limit switch)

Note: This is a normal phenomenon.

Step 1: Set parameters P2-16 to 123.

Step 2: Restart the device.

Clear AL014 (servo parameters are default settings, CN1 interface is not linked with a limit switch)

Note: This is a normal phenomenon.

Step 1: Set parameters P2-15 to 122.

Step 2: Restart the device.

3.1.2 Read Servo Parameter

//Read servo parameter data.

//Note: It is recommended that this function not be placed in a code area requiring fast loop implementation.

```
I16 PASCAL _DMC_01_read_servo_parameter (U16 CardNo,//card number,
                                        value range: 0-15.
                                        U16 NodeID,//node number, value range: 1-12.
                                        U16 SlotID,//SlotID, this is assigned a value of 0.
                                            U16 group,//parameter group code.
                                            U16 idx,//parameter sequence number.
                                            U32*data//returned parameter value.
                                            )
```

Note: For example, to read parameters P3-00, assign the group a value of 3, and assign idx of value of 0.

3.1.3 Write Servo Parameter

//Write servo parameter data.

//Note: It is recommended that this function not be placed in a code area requiring fast loop implementation.

```
I16 PASCAL _DMC_01_write_servo_parameter (U16 CardNo,//card number,
                                         value range: 0-15.
                                         U16 NodeID,//node number, value range: 1-12.
                                         U16 SlotID,//SlotID, this is assigned a value of 0.
                                             U16 group,//parameter group code.
                                             U16 idx,//parameter sequence number.
                                             U32 data//entered new parameter value.
                                                 )
```

3.1.4 Parameters Reset during the System Initialization Process:

In "motion control card system initialize process" or "re-power on the servo drive", the servo drive parameters will be reset:

Parameters	Explanation of parameter
P1-32	Motor stop mode
P1-34	Acceleration constant of S-curve
P1-38	Zero speed range setting
P1-44	Electronic gear (denominator) (N1)
P1-45	Electronic gear (denominator) (M)
P1-47	Speed reached (DO.SP_OK) range
P1-49	Time range after Speed reached
P2-35	Condition of excessive position control deviation warning

Corresponding method:

{Method 1} After completing initialization of the axis card, use the _DMC_01_write_servo_parameter (U16 CardNo, U16 NodeID, U16 SlotID, U16 group,U16 idx,U32 data) function to write the expected value as a servo driver parameter to be used.

{Method 2} Change the Z bit of servo drive parameters P3-12 (defined as follows) to 1, restart the servo drive, and change the parameter to be used.

P3-12	QSTPO	CANopen Support Setting	Communication address:0318H 0319H
	Default value	0x0000	
	Control method	CANopen	
	Unit:	0x0000 ~ 0x0111	
	Range:	As below	
	Data size:	16bit	
	Format:	HEX	

Settings: CANopen synchronization setting is divided into X, Y, Z, U (hexadecimal):

Digit	U	Z	Y	X
Function	-	CANopen value will be loaded in	If the motor will enter Quick Stop mode when in auto protection	Whether OD-6040 supports Quick Stop
Range	-	0 ~ 1	0 ~ 1	0 ~ 1

Defined as follows:

X=0: Servo ON only requires OD-6040 Bit3 (Enable Operation) to be ON

X=1: Servo ON requires OD-6040 Bit0, Bit1, Bit3 to be ON (complies with CANopen DS402 standard), and supports entry to Quick Stop mode via OD-6040 Bit2 (Quick Stop)

Y=0: Will not enter Quick Stop mode when automatic protection causes motor decelerates to stop.

Y=1: Will enter Quick Stop mode when automatic protection causes motor decelerates to stop; requires fault reset in order to continue other commands.

Z=0: After re-power on or reset the communication, CANopen default value will be loaded in.

Z=1: Will not change parameter values after re-power on or reset the communication.

U: Undefined

3.1.5 Enable / Disable

//Enable (excite) or disable motor.

I16 PASCAL _DMC_01_ipo_set_svon (U16 CardNo,//card number, value range: 0-15.

U16 NodeID,//node number, value range: 1-12.

U16 SlotID,//SlotID, this is assigned a value of 0.

U16 ON_OFF//ON_OFF value, 0: disable; 1: enable.

3.1.6 Return to Origin

■ Parameter Configuration

//parameter configuration, set homing mode, offset, low speed, high speed, acceleration time.

//Note: Software limits must be disabled before homing.

I16 PASCAL _DMC_01_set_home_config (U16 CardNo,//card number, value range: 0~15.

U16 NodeID,//node number, value range: 1~12.

U16 SlotID,//SlotID, this is assigned a value of 0.

U16 Mode,//mode, value range 1~35.

I32 offset,//deflection.

//low speed, units: rpm, value range: 1~500.

U16 low speed

//high speed, units: rpm, value range:

U16 high speed, 1~2000

F64 acc//acceleration time.

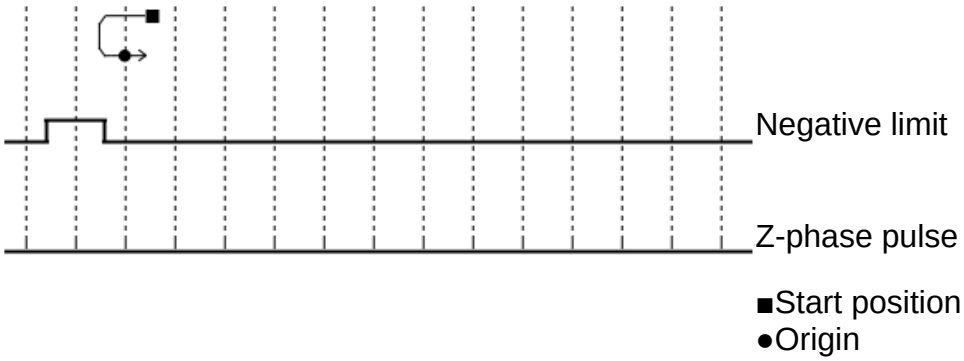
)

■ Homing Mode

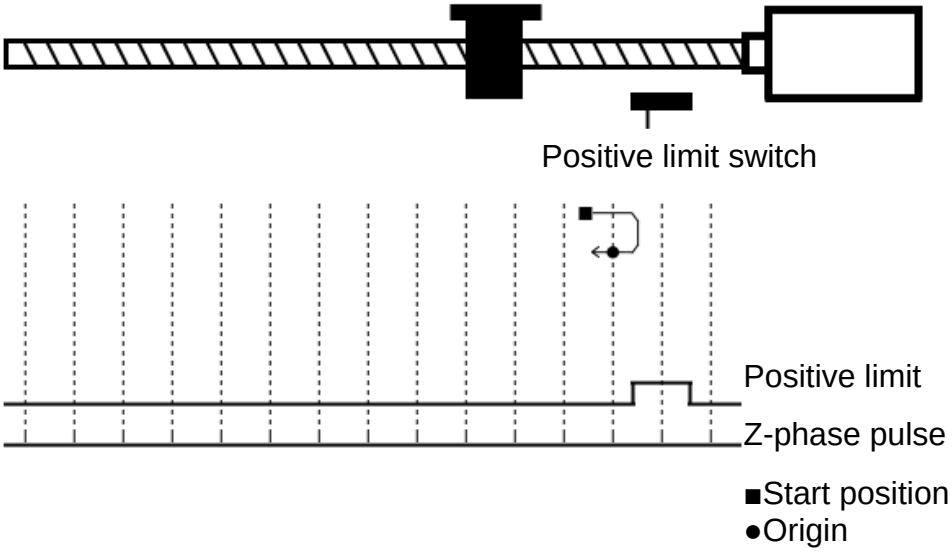
Negative limit switch　Origin switch　Positive limit switch

Mode 1

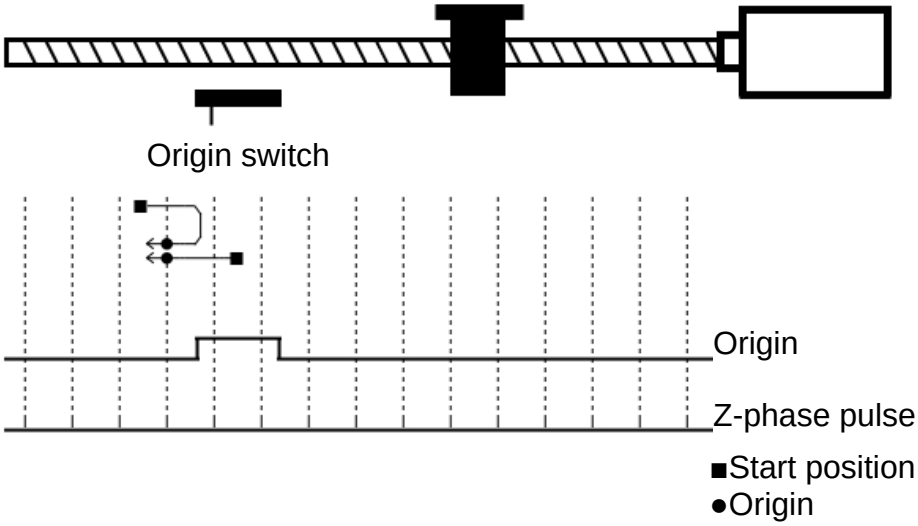
Negative limit switch



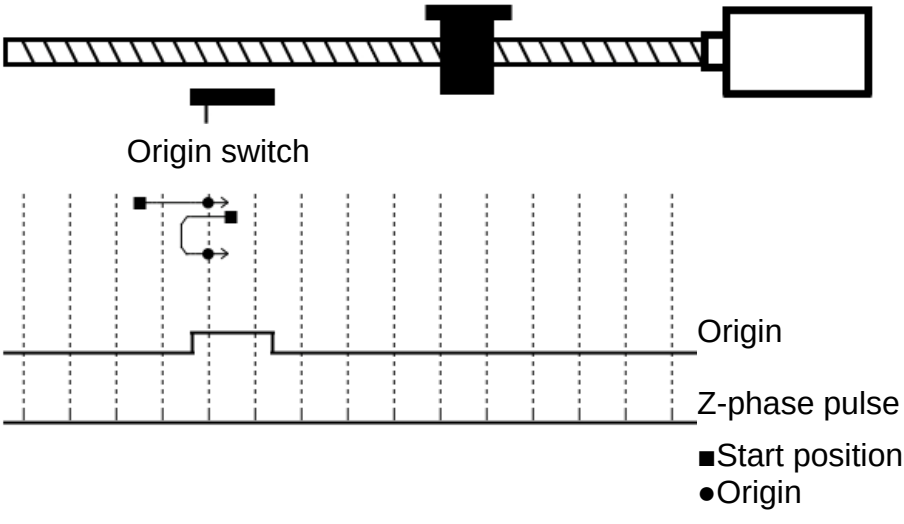
Mode 2



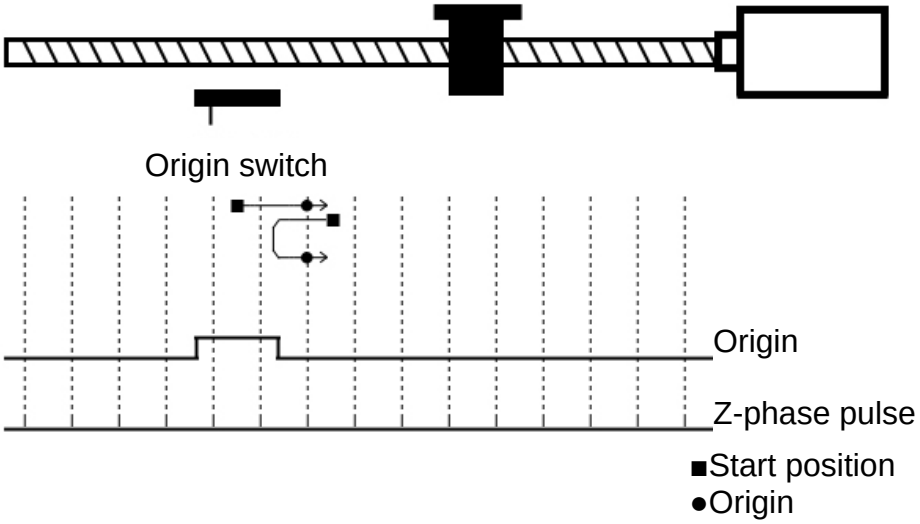
Mode 3



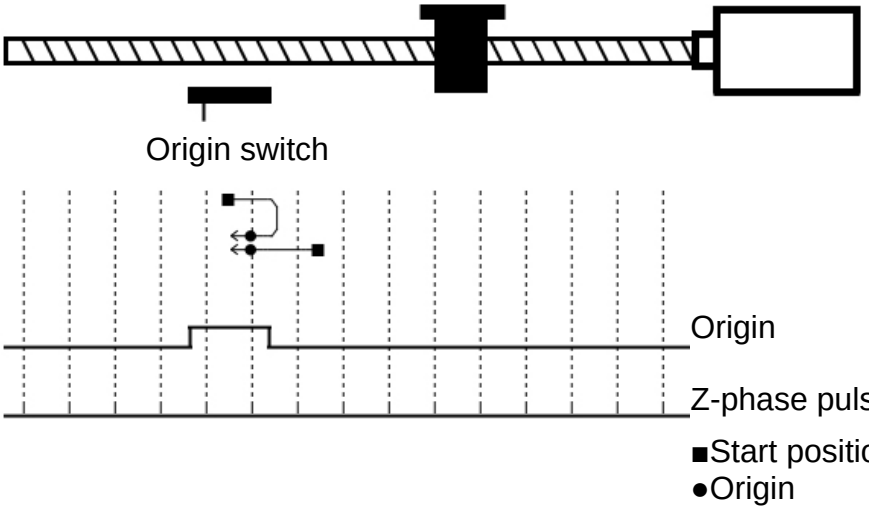
Mode 4



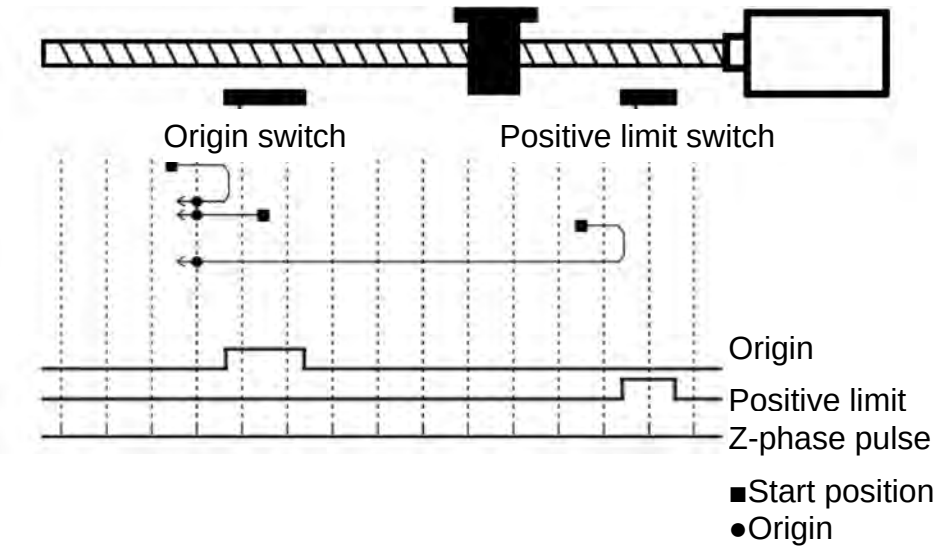
Mode 5



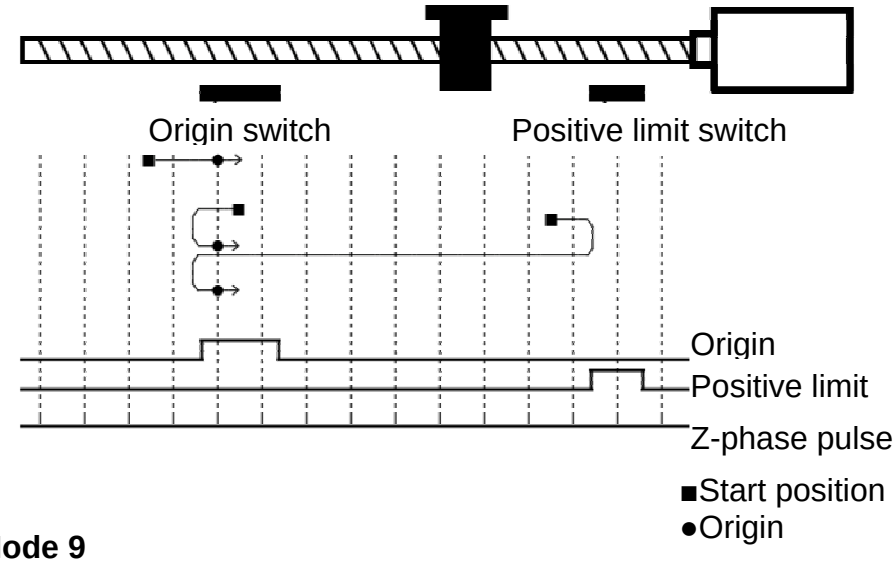
Mode 6



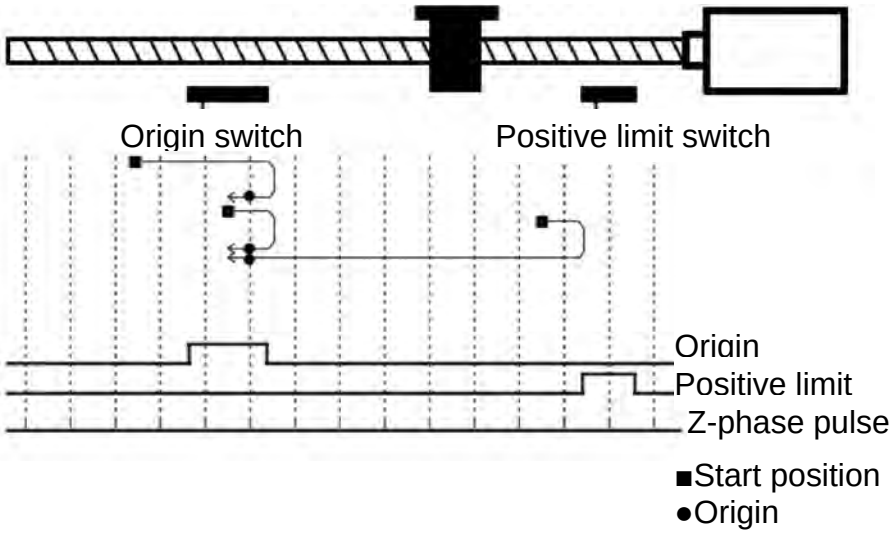
Mode 7



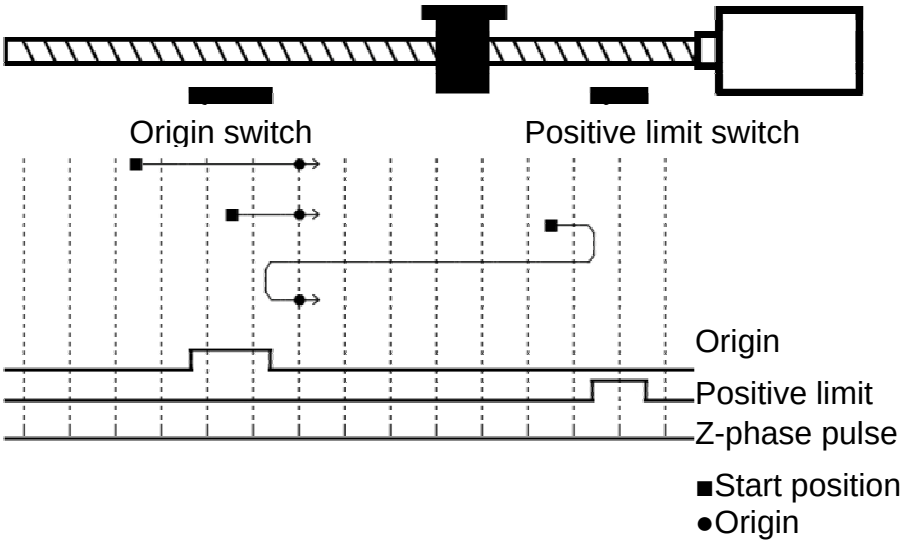
Mode 8



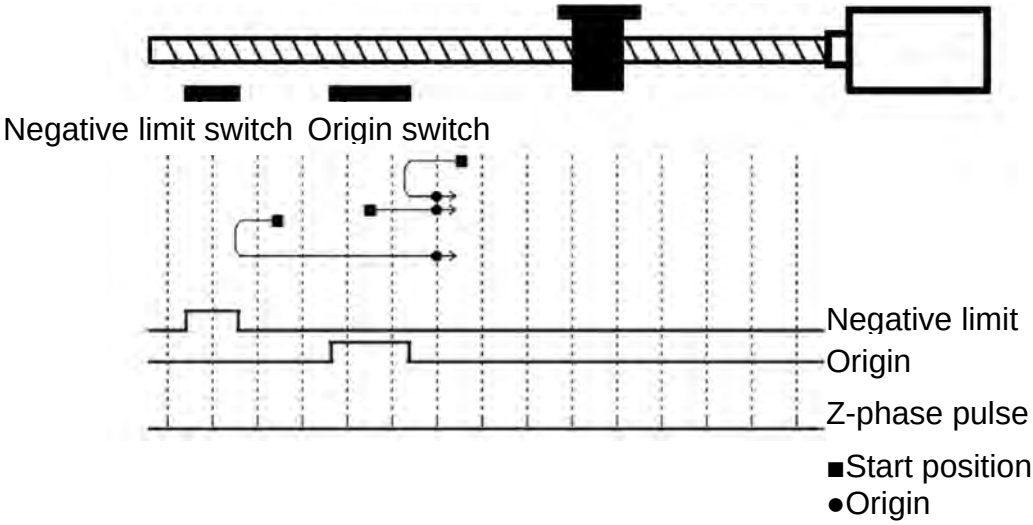
Mode 9



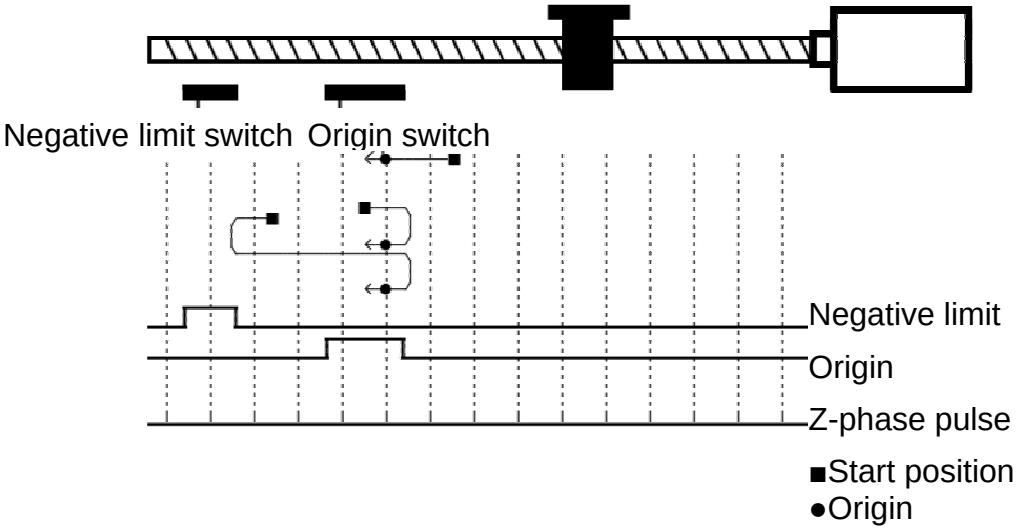
Mode 10



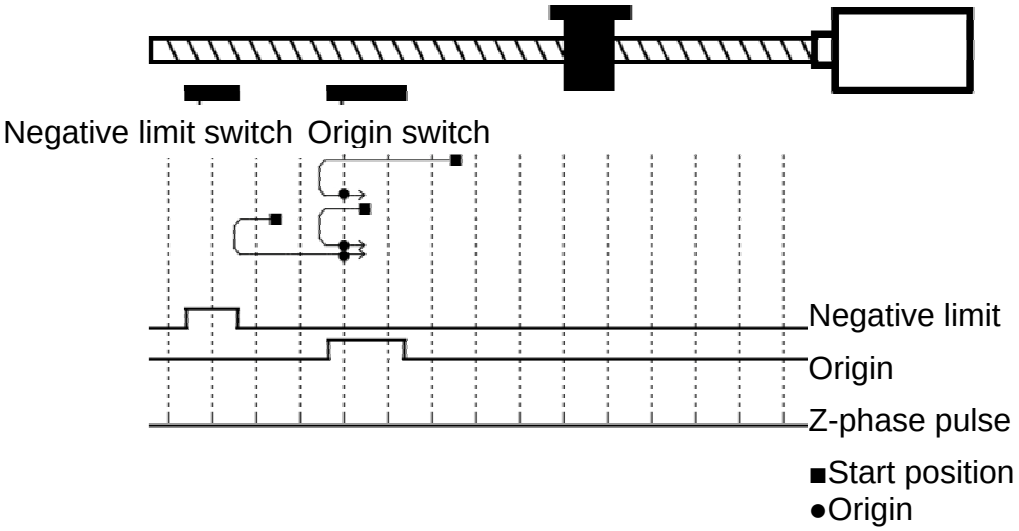
Mode 11



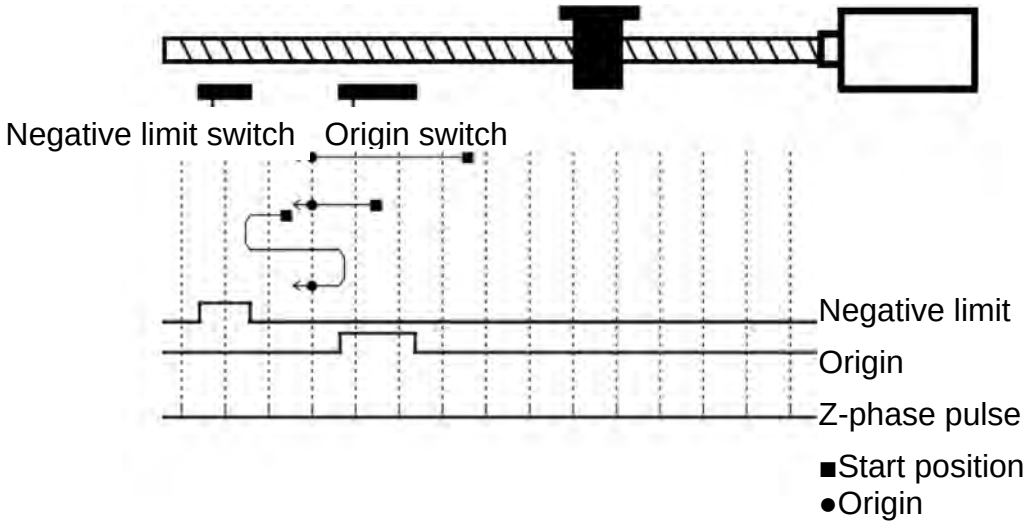
Mode 12



Mode 13



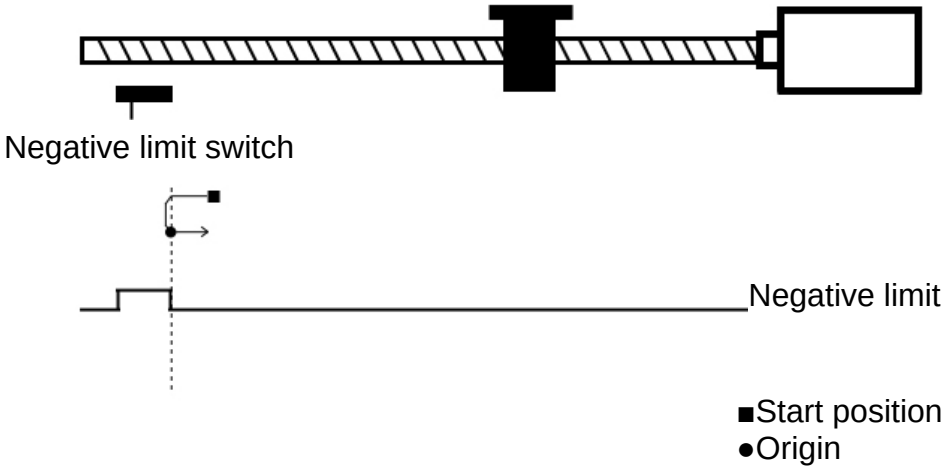
Mode 14



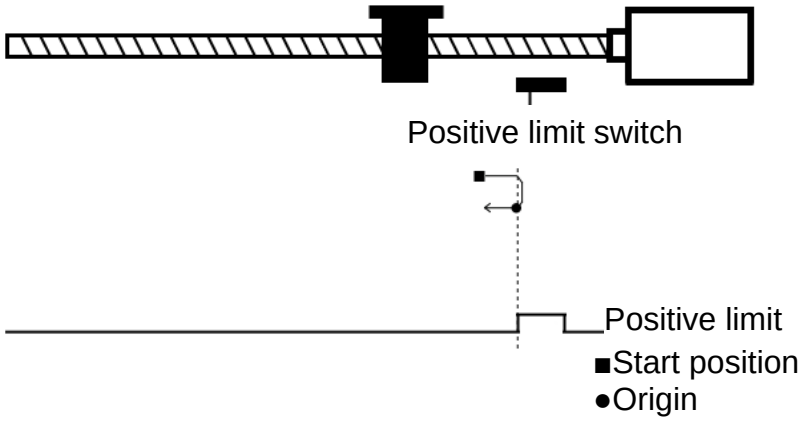
Mode 15 - Reserved

Mode 16 - Reserved

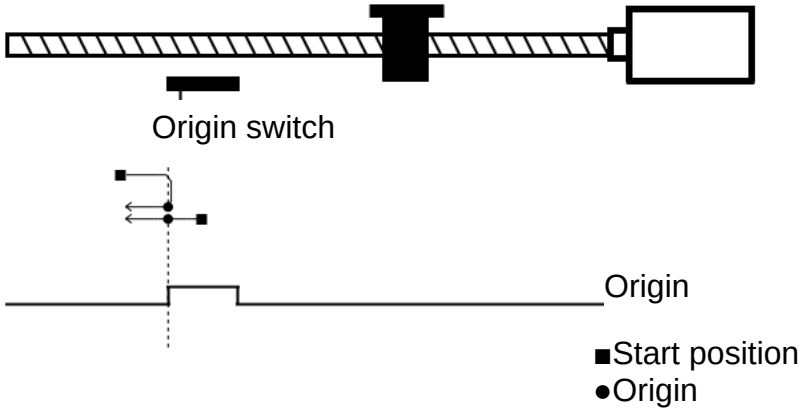
Mode 17



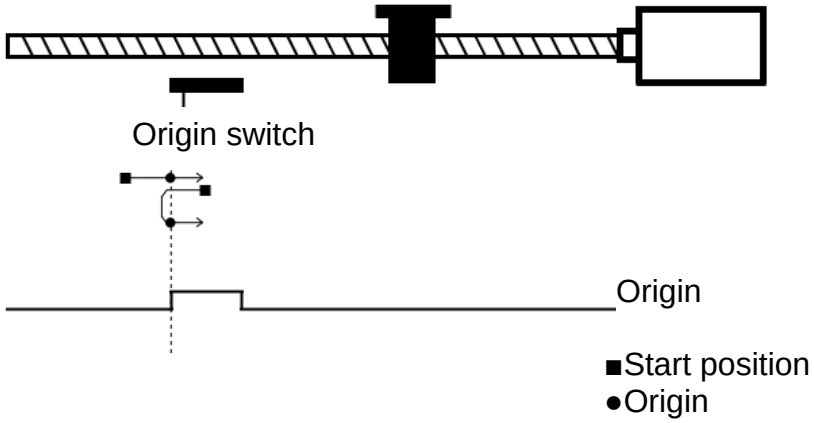
Mode 18



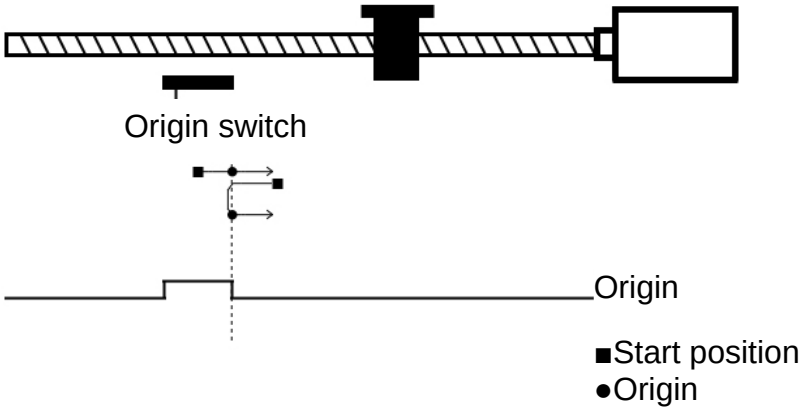
Mode 19



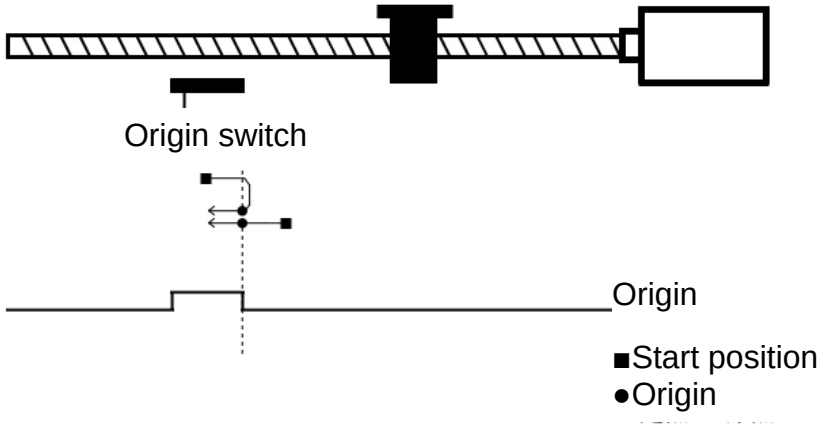
Mode 20



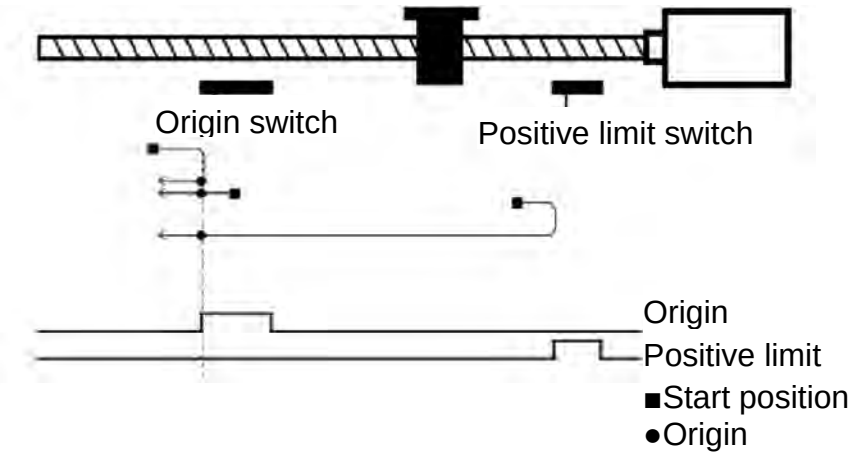
Mode 21



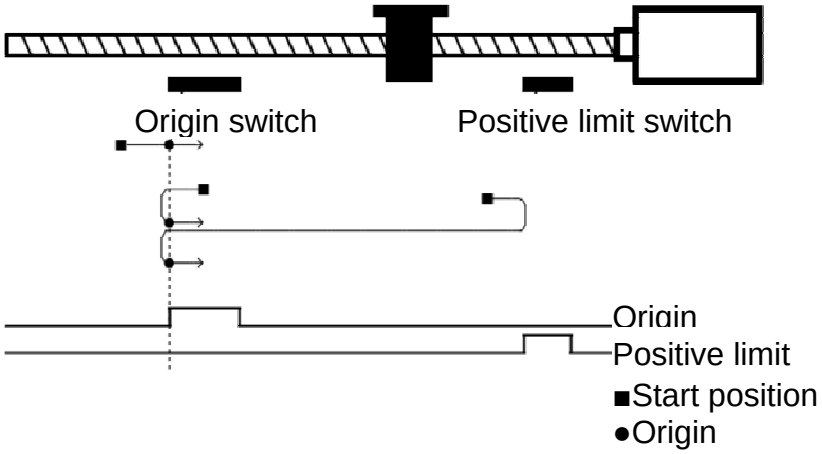
Mode 22



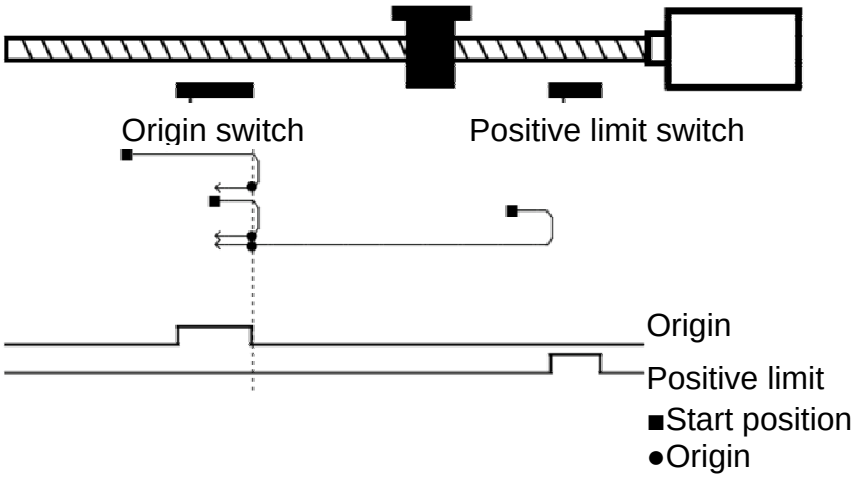
Mode 23



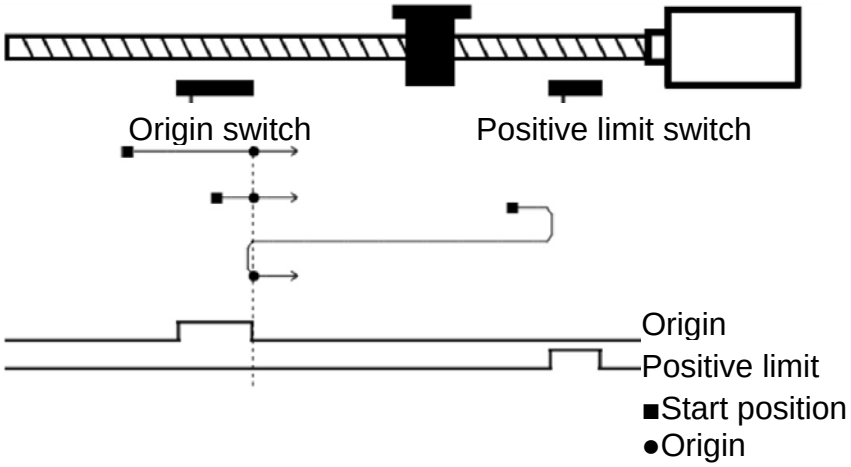
Mode 24



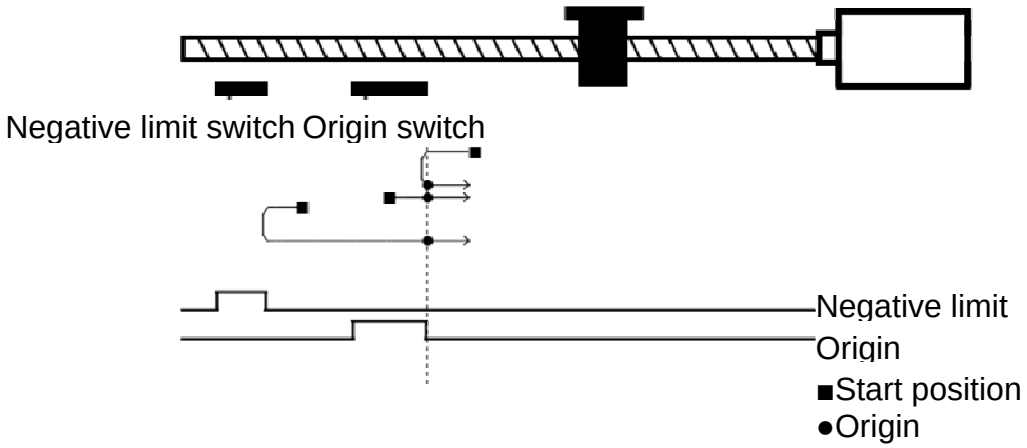
Mode 25



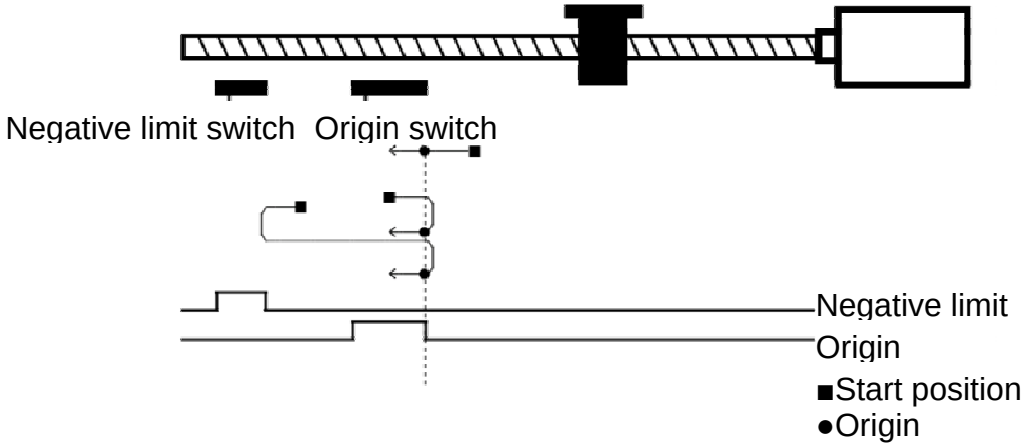
Mode 26



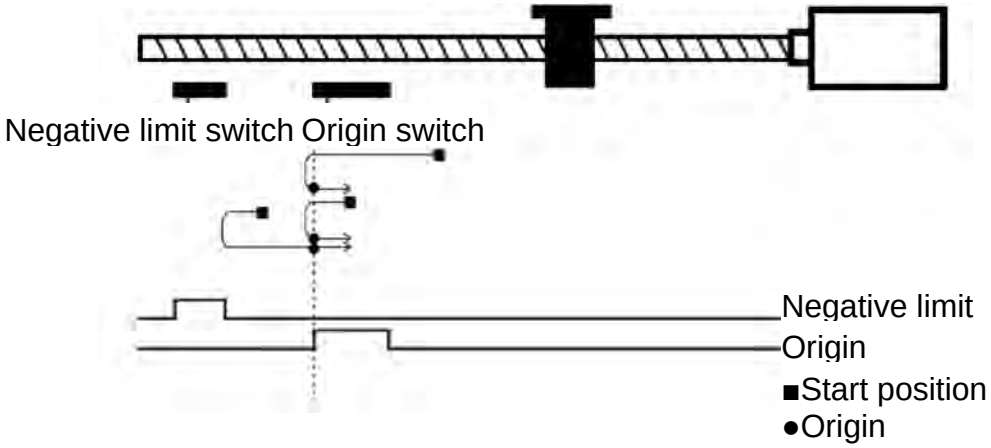
Mode 27



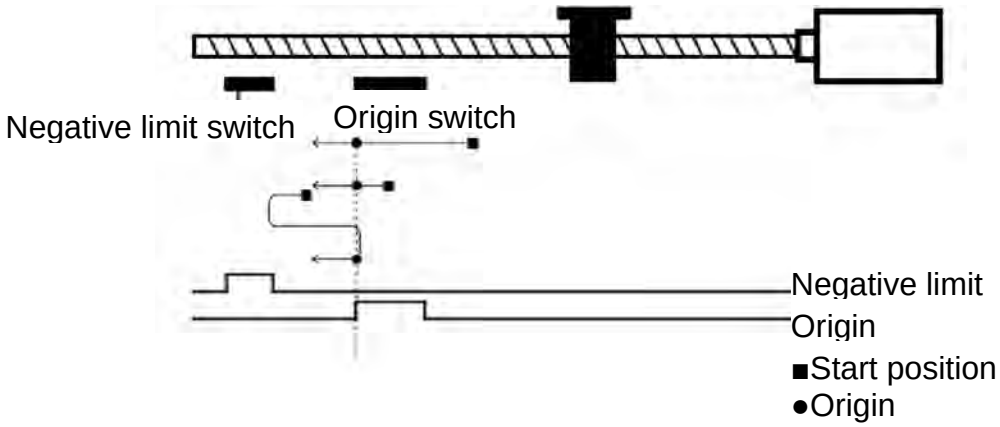
Mode 28



Mode 29



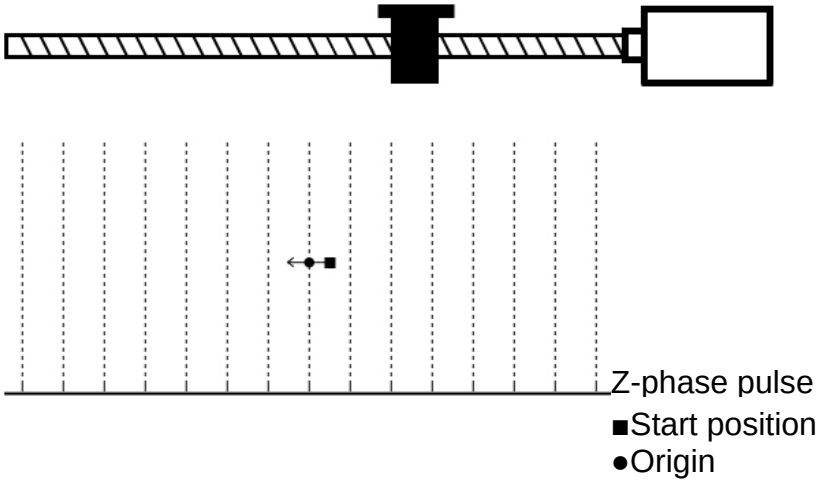
Mode 30



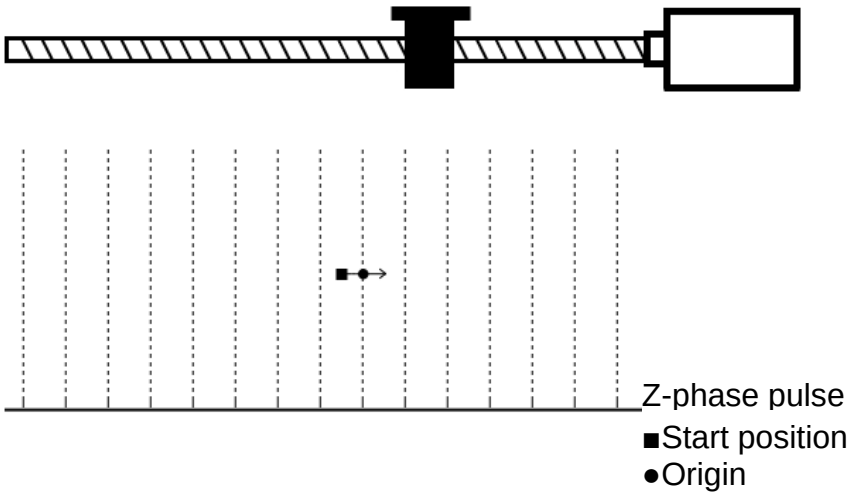
Mode 31 - Reserved

Mode 32 - Reserved

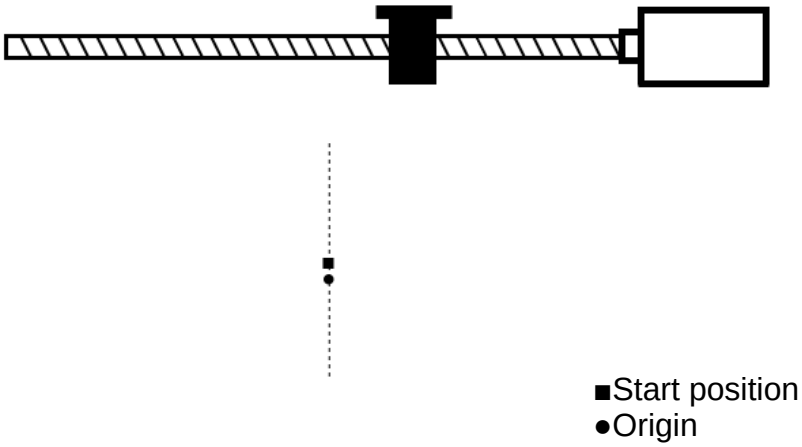
Mode 33



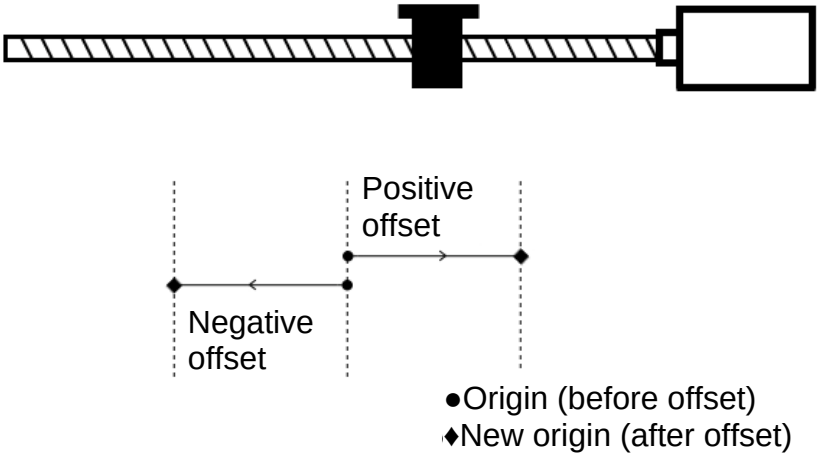
Mode 34



Mode 35



Offset



- Implement Homing

  //Start homing

  I16 PASCAL _DMC_01_set_home_move (U16 CardNo,//card number, value range: 0~15.
  U16 NodeID,//node number, value range: 1~12.
  U16 SlotID//SlotID, this is assigned a value of 0.

- Judge Whether Completed

  Introduction

  Two functions must be invoked when judging whether homing has been completed, and "Homing complete" status must simultaneously satisfy the following two conditions:

  1. Execute _DMC_01_motion_done, which returns MC_done (= 0);
  2. Execute _DMC_01_motion_status, which returns MC_status;

     "Mode Selection" is "Homing mode (bit3 is 0, bit2 is 1, bit1 is 1, and bit0 is 0),"

     "Mode specific" allows "Can implement homing (bit12 is 1),"

     "Target arrival at objective (bit10)" is 1.

  //achieve motion stage.

  I16 PASCAL _DMC_01_motion_done (U16 CardNo,//card number, value range: 0-15.
  U16 NodeID,//node number, value range: 1-12.
  U16 SlotID,//SlotID, this is assigned a value of 0.
  U16*MC_status//transmit back motion stage, 0: stop motion.

  //achieve general motion status.

  I16 PASCAL _DMC_01_motion_status (U16 CardNo,//card number, value range: 0-15.
  U16 NodeID,//node number, value range: 1-12.
  U16 SlotID,//SlotID, this is assigned a value of 0.
  U32*MC_status//transmit back general motion status.

Notes:

Transmit back general motion status MC_status

Bit	7	6	5	4	3	2	1	0
Label	TG	DR	WR	DI3	Mode 3	Mode 2	Mode 1	Mode 0
Description	Triggering mode	Data error message	Alarm message	DI3(SLD) Status map	Mode Selection			

Bit	15	14	13	12	11	10	9	8
Label	NEL	PEL	MDS1	MDS0	N/A	Target	DriverErr	PWRON
Description	Negative limit	Positive limit	Mode specific		Motion direction 1: Positive 0: negative	Arrival at objective	Drive Error	Motor start excitation

Transmit back "Mode selection" in general motion status MC_status

Bit	3	2	1	0	Explanation of mode
Label	Mode 3	Mode 2	Mode 1	Mode 0	
Value	0	0	0	1	Profile Position mode
	0	0	1	1	Profile Velocity mode
	0	1	0	0	Profile Torque mode
	0	1	1	0	Homing mode
	1	1	1	1	DMCNET mode

Transmits back "Mode specific" in general motion status MC_status

	Bit	13	12	Explanation of mode
	Label	MDS 1	MDS 0	
Mode Selection: Homing mode	Value	1		Homing error
			1	Can implement homing
Mode Selection: DMCNET mode		1		Undefined
			1	Mode enabled

■ Terminate Homing

//terminates homing motion.

I16 PASCAL _DMC_01_escape_home_move (U16 CardNo,//card number, value range: 0-15.

U16 NodeID,//node number, value range: 1~12.

U16 SlotID//SlotID, this is assigned a value of 0.

3.1.7 Absolute Type of Motor

Applications steps of absolute type of motor:

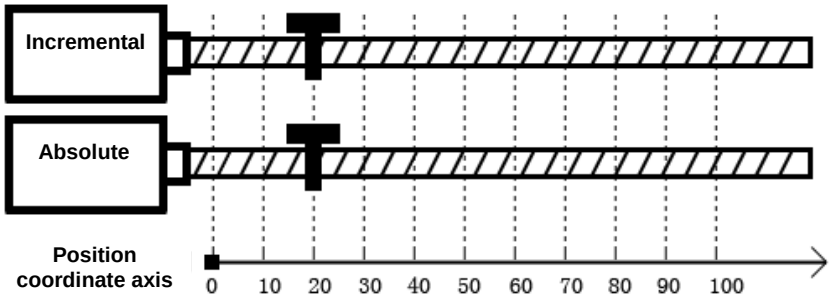
1. Set parameters P2-69 = 1, which is in absolute type of motor mode, and restart device.
2. Set parameters P2-08 = 271 and P2-71 = 1 to initialize absolute coordinates, and the AL060 warning will be cleared.
3. Invoke axis card commands (function commands from incremental type and absolute type of motor are the same) to perform motion control.

■ Obtain Current Position of Absolute Type of Motor

Invoke _DMC_01_get_position command, which will obtain the current position of the absolute motor.

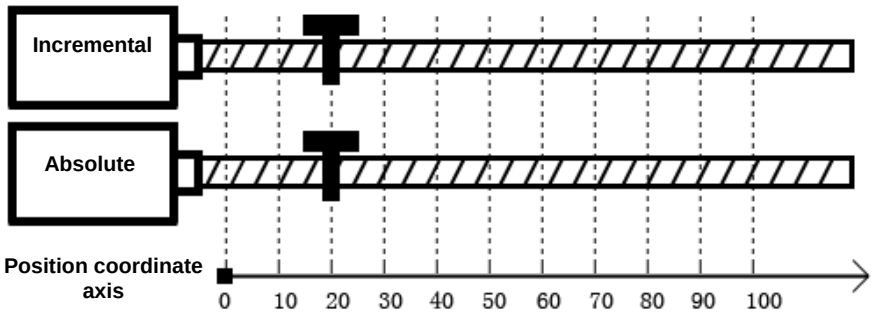
Example 1:

1. When the working platforms of the "incremental motor" and "absolute motor" both have positions of 20,



the value returned by incremental motor _DMC_01_get_position will be 20, and the value returned by absolute motor _DMC_01_get_position will be 20.

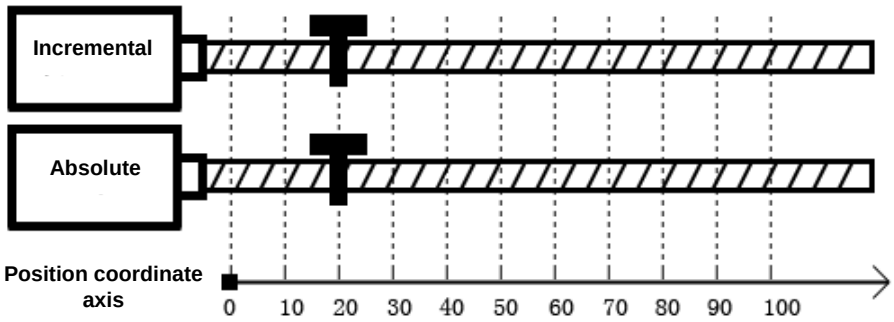
2. When the working platforms of the "incremental motor" and "absolute value motor" are unmoving at a position of 20, the servo driver will be re-powered on,



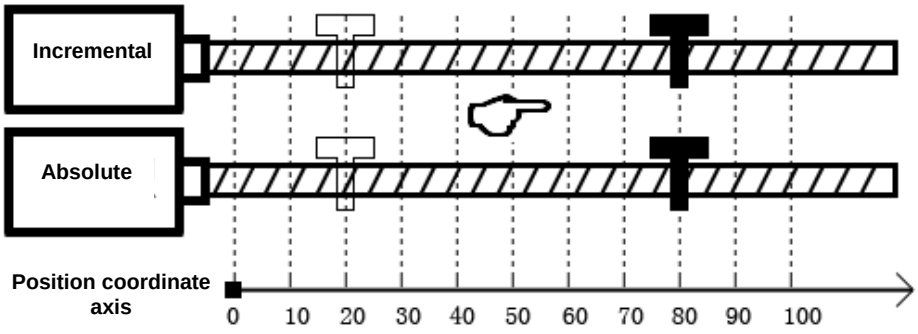
and the value returned by incremental motor _DMC_01_get_position will be 0, and the value returned by absolute motor _DMC_01_get_position will be 20.

Example 2:

1. When the working platforms of the "incremental motor" and "absolute motor" both have positions of 20, the value returned by incremental motor _DMC_01_get_position will be 20, and the value returned by absolute motor _DMC_01_get_position will be 20.



2. With power off the servo driver, move the working platforms of the "incremental motor" and "absolute motor" to 80 by hand.

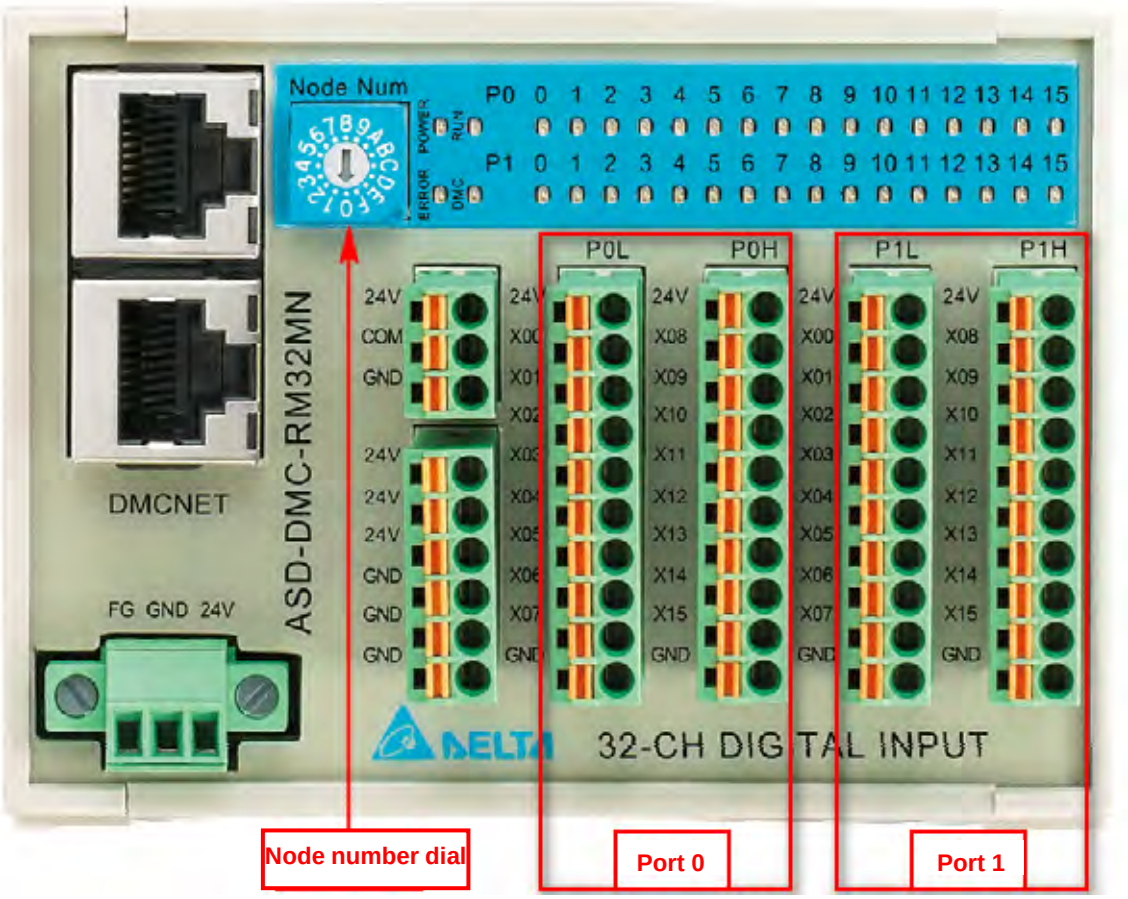


3. When power is applied to the servo drives, and the value returned by incremental motor _DMC_01_get_position will be 0, and the value returned by absolute motor _DMC_01_get_position will be 80, which indicates that the change in the position of the absolute motor when power was off has been recorded.

Chapter 4 Remote Digital Input Module

4.1 Digital Input ASD-DMC-RM32MN (32 points)

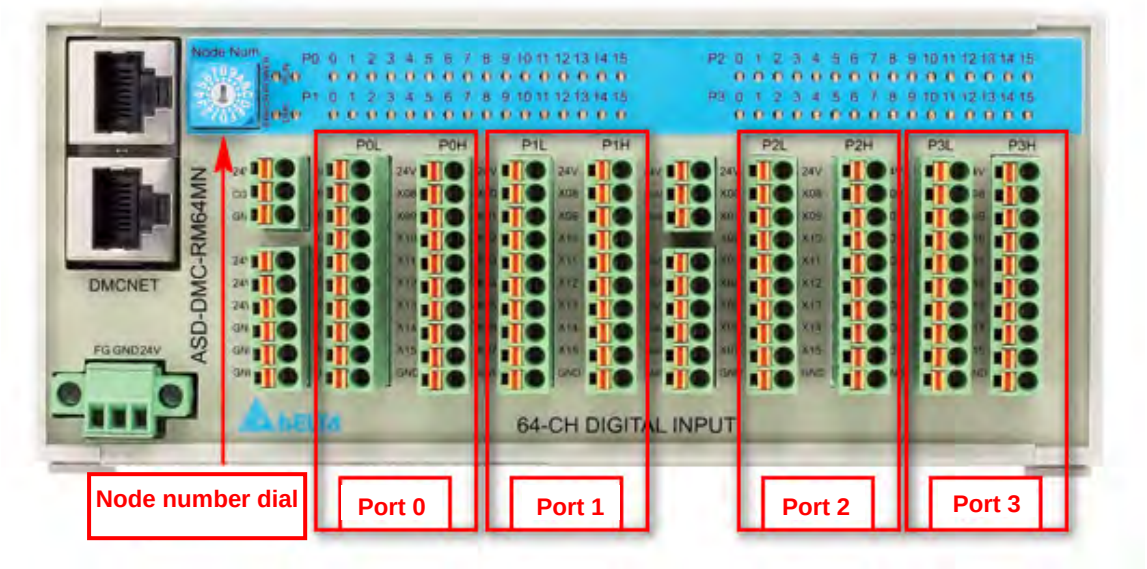
- **Explanation of Node Number Settings and Port Numbers**



Effective range of node number dial: 1~9, A(10), B(11), C(12).

4.2 Digital Input ASD-DMC-RM64MN (64 points)

■ Explanation of Node Number Settings and Port Numbers

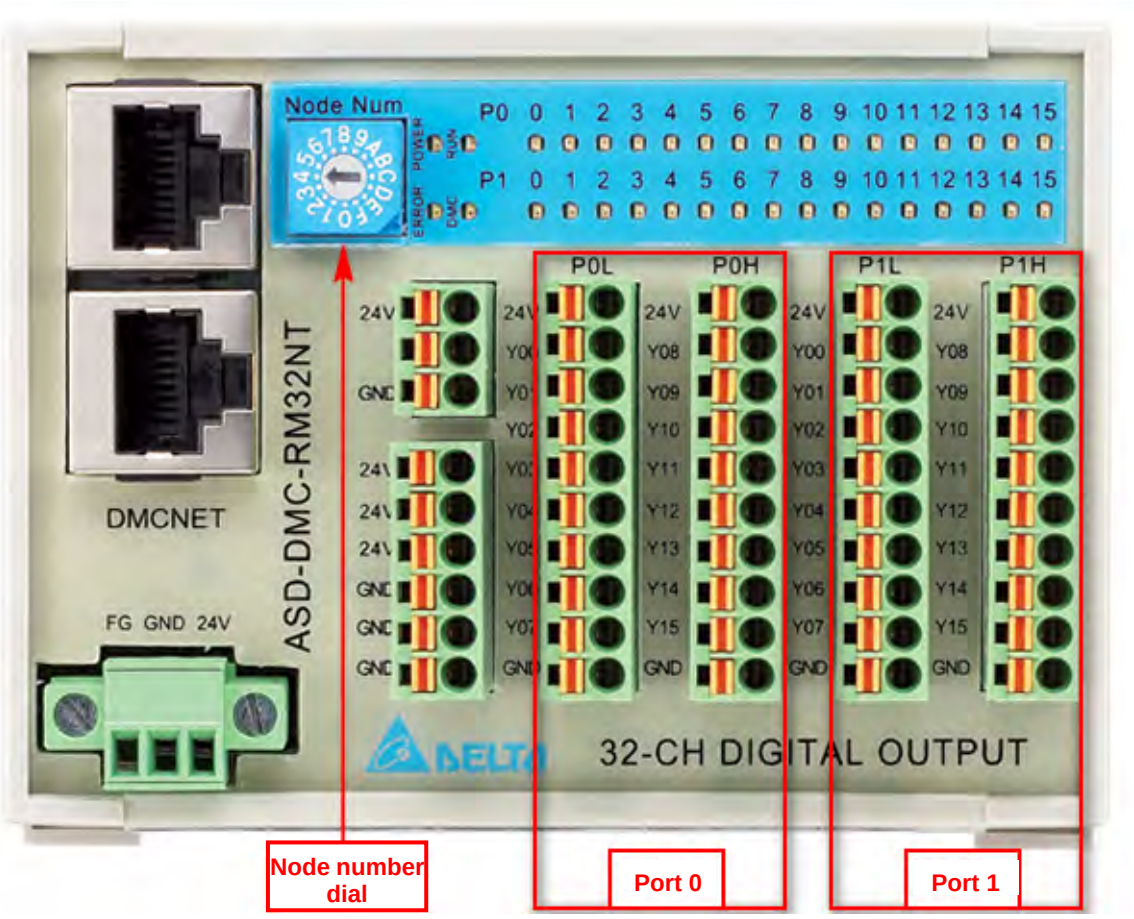


Effective range of node number dial: 1~9, A(10), B(11), C(12).

Chapter 5 Remote Digital Output Module

5.1 Digital Output ASD-DMC-RM32NT (32 points)

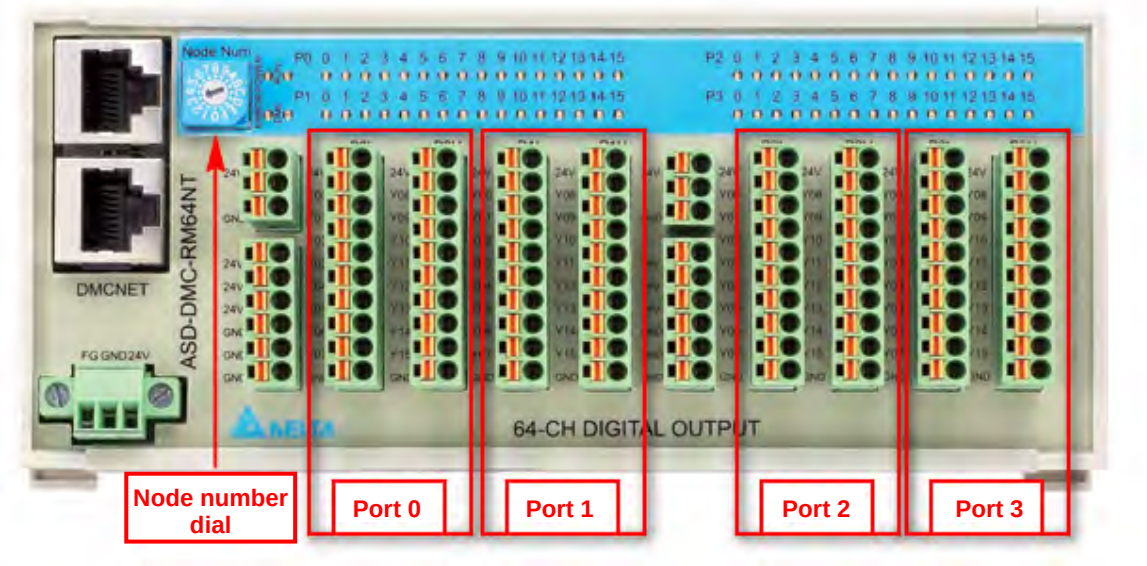
■ **Explanation of Node Number Settings and Port Numbers**



Effective range of node number dial: 1~9, A(10), B(11), C(12).

5.2 Digital Output ASD-DMC-RM64NT (64 points)

■ Explanation of Node Number Settings and Port Numbers

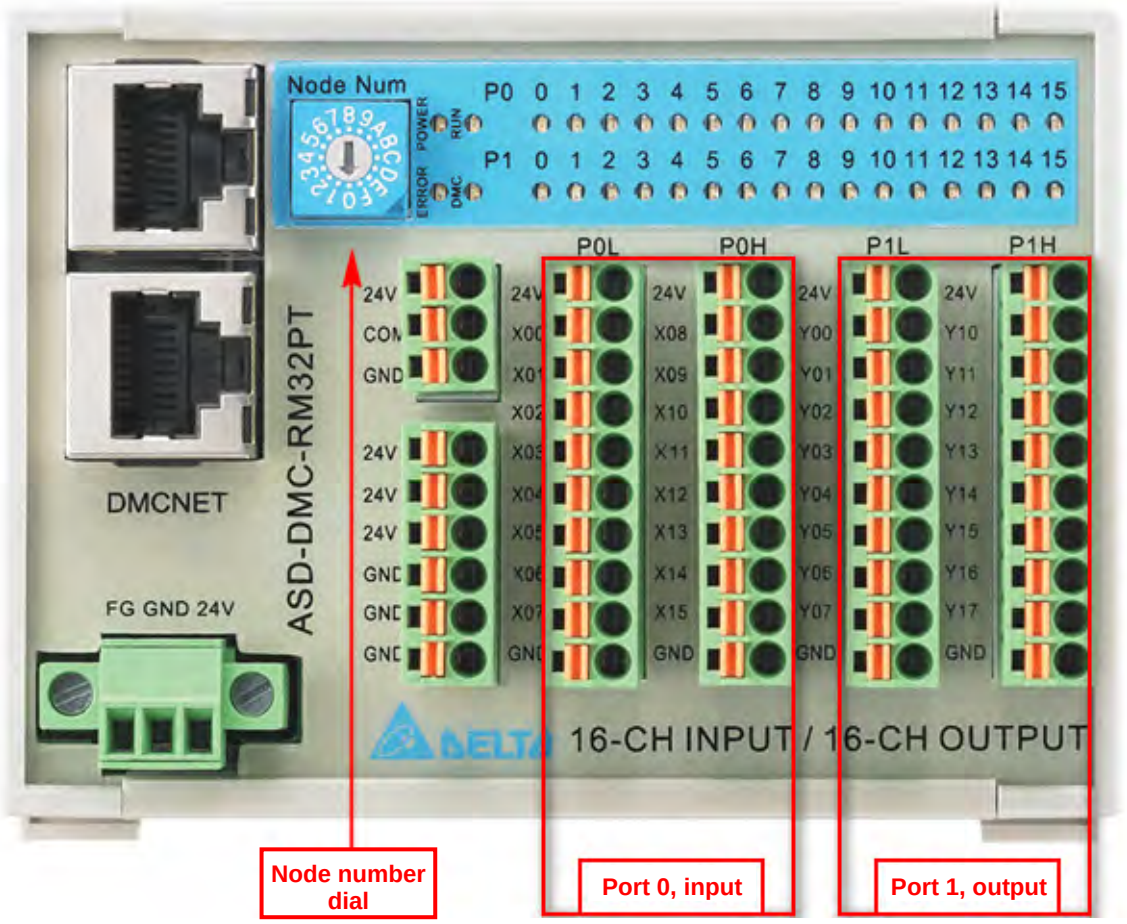


Effective range of node number dial: 1~9, A(10), B(11), C(12).

Chapter 6 Remote Digital Input / Output Combination Module

6.1 Combination Module ASD-DMC-RM32PT (16-point Digital Input / Digital Output)

- **Explanation of Node Number Settings and Port Numbers**



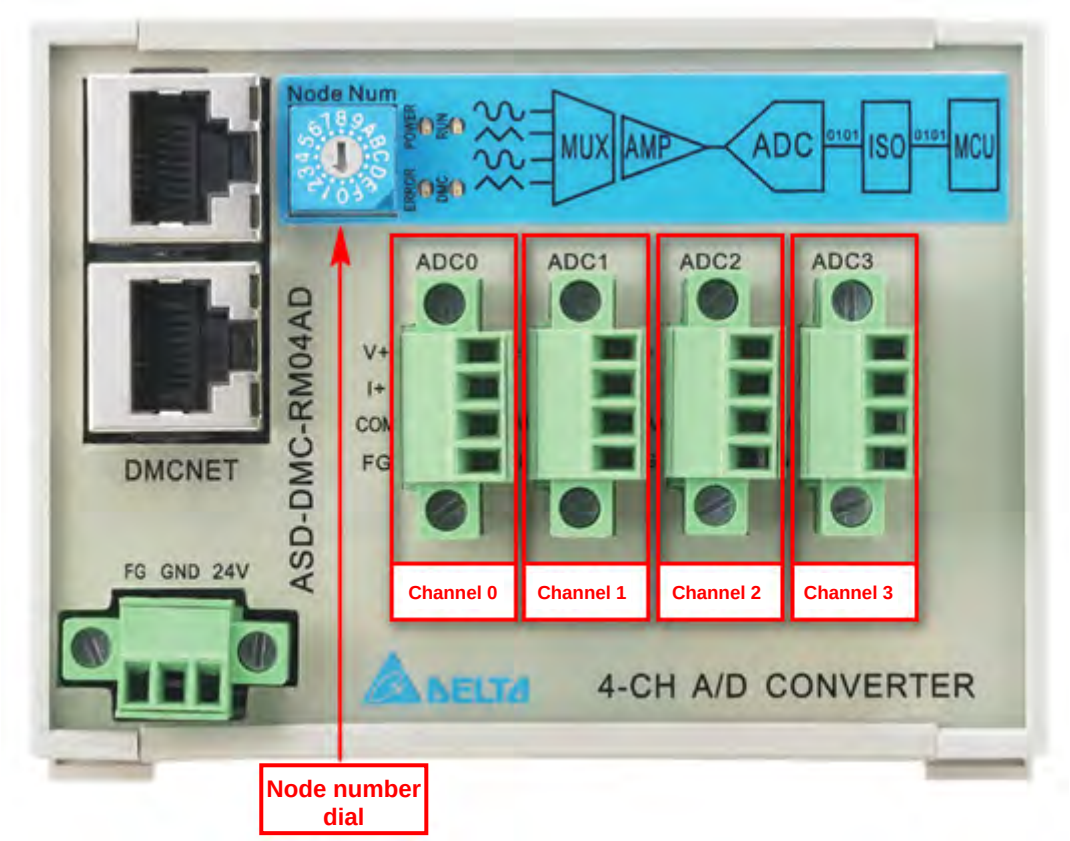
Effective range of node number dial: 1~9, A(10), B(11), C(12).

(This page is intentionally left blank.)

Chapter 7 Remote Analog Input Module

7.1 Analog Input Module ASD-DMC-RM04AD (4 channels)

- **Explanation of Node Number Settings and Port Numbers**



Node number setting: Turn the arrowhead on the “Node number dial” until the desired number of nodes is reached.

Effective range of node number dial: 1~9, A(10), B(11), C(12).

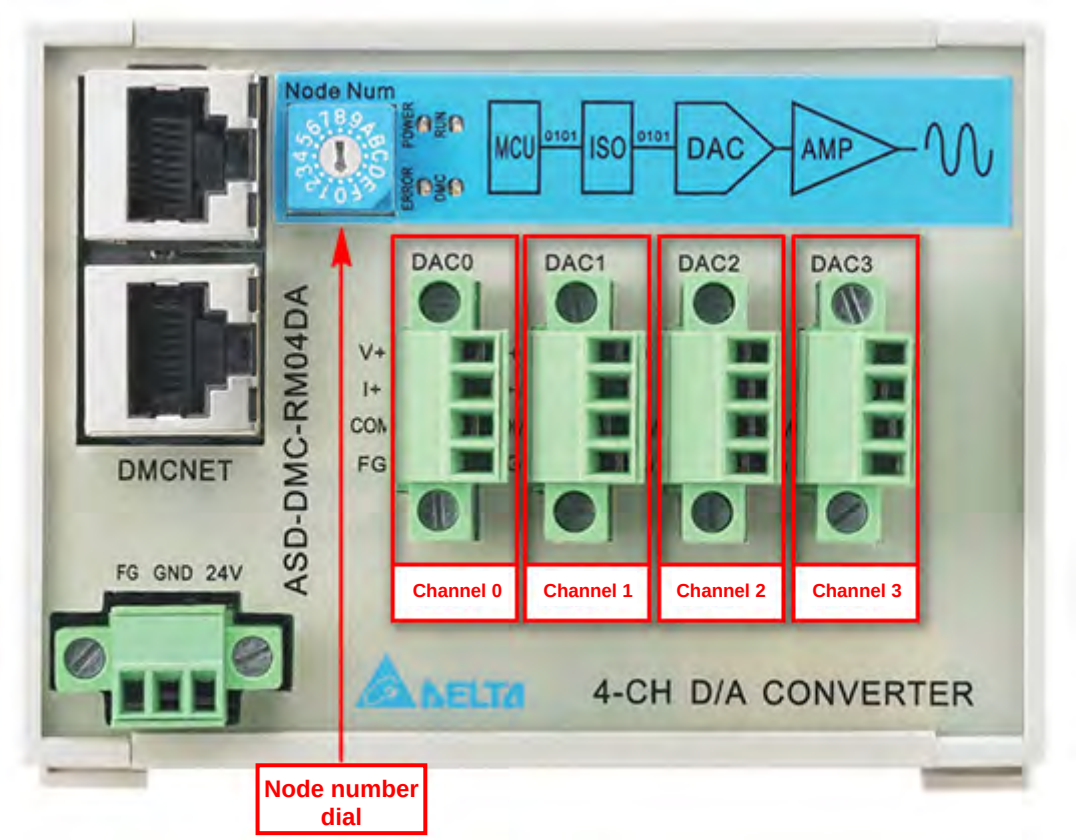
Channel number: 0 ~ 3.

(This page is intentionally left blank.)

Chapter 8 Remote Analog Output Module

8.1 Analog Output Module ASD-DMC-RM04DA (4 channels)

- **Explanation of Node Number Settings and Port Numbers**



Node number setting: Turn the arrowhead on the “Node number dial” until the desired number of nodes is reached.

Effective range of node number dial: 1~9, A(10), B(11), C(12).

Channel number: 0~3.

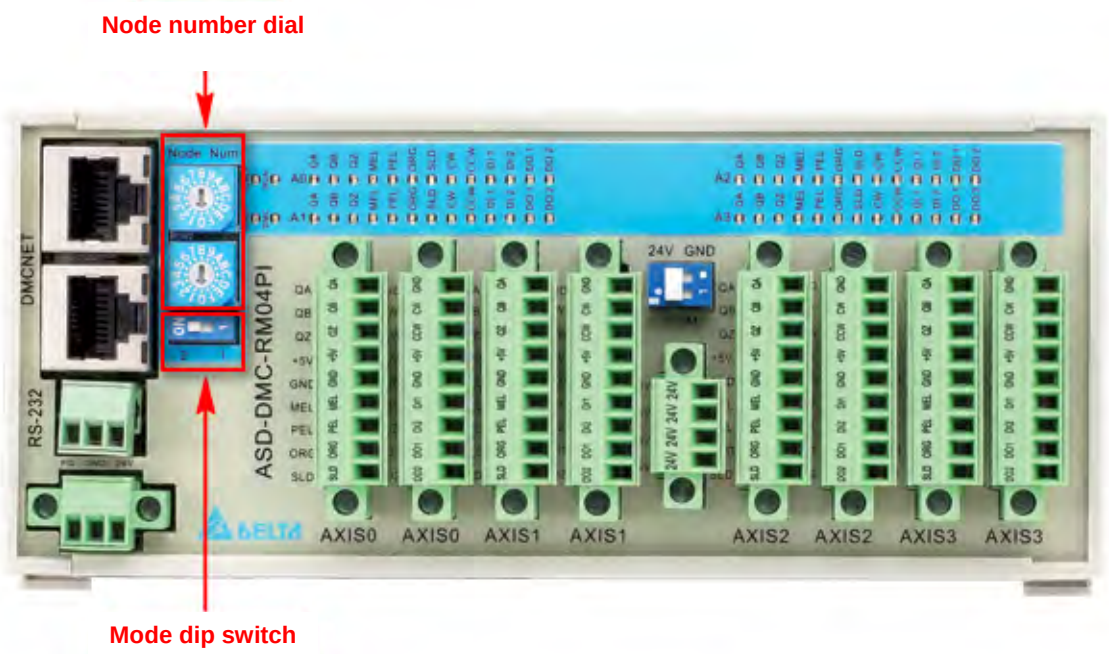
(This page is intentionally left blank.)

Chapter 9 Remote Pulse Module

9.1 Pulse Module ASD-DMC-RM04PI (4 channels)

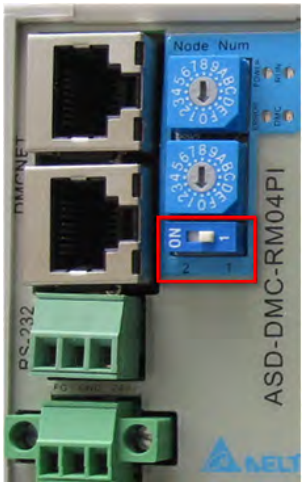
9.1.1 Explanation of Node Number Settings and Port Numbers

The ASD-DMC-RM04PI module (shown below) can be connected with four stepping motors operating axes 0~3, and has the two operating modes "mode 1" and "mode 2".

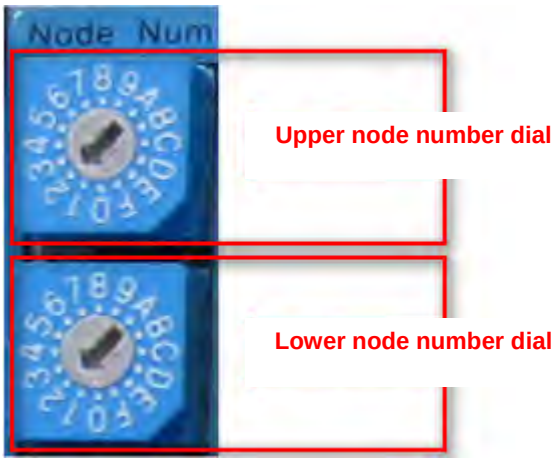


- Mode 1

  1. Mode setting: Move dip switch to position “1”.



2. Node number setting: Each RM04PI module occupies one node number, and the upper and lower node number dials point to the same position (effective range of node number dial: When connected to card A01/B01: 1~9, A (10), B (11), C (12); when connected to card F01: 1~6.) . The following illustration shows 2 nodes.

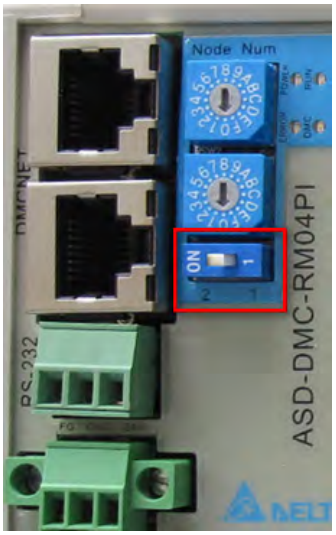


The module only occupies one node number, and command functions are distinguished as AXIS0, AXIS1, AXIS2, AXIS3, and SlotID by means of the SlotID parameter, and the inputs 0, 1, 2, and 3 correspond to AXIS0, AXIS1, AXIS2, and AXIS3, respectively.

For example, assuming the 04PI module node number is 2, and the corresponding card number is 1, if it is wished to connect the stepping motor with AXIS2 to initiate the movement of single-axis, the following function can be invoked: _DMC_01_rm_04pi_md1_start_move (U16 CardNo, U16 NodeID,U16 SlotID, I32 Dist, I32 StrVel, I32 MaxVel, F64 Tacc, F64 Tdec,U16 m_curve,U16 m_r_a), where parameters include CardNo=1, NodeID=2, SlotID=2, and appropriate values are entered for other parameters.

■ Mode 2

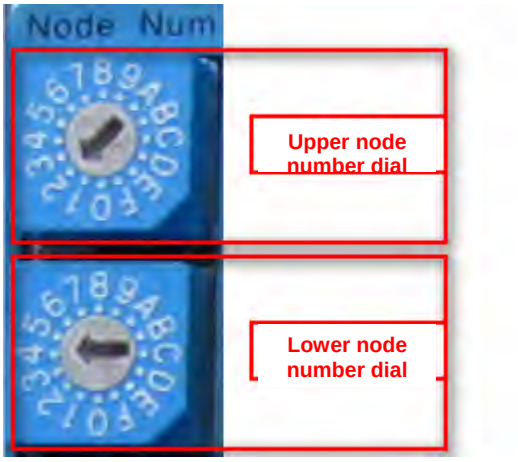
1. Mode setting: Move dip switch to position "2 (ON)".



2. Node number setting: Each RM04PI module occupies node numbers 1~4, which correspond to AXIS0, AXIS1, AXIS2, and AXIS3. The upper node number dial indicates the starting node number, and the lower node number dial indicates the ending node number. The following rules apply:

Axis 0	Axis 1	Axis 2	Axis 3	Number of nodes occupied by module	Range of values on upper dial	Value on lower dial
Valid	Invalid	Invalid	Invalid	1	1--12	Value on upper dial
Valid	Valid	Invalid	Invalid	2	1--11	Value on upper dial +1
Valid	Valid	Valid	Invalid	3	1--10	Value on upper dial +2
Valid	Valid	Valid	Valid	4	1--9	Value on upper dial +3
Note: Effective ranges of upper and lower node number dials: When connected to card A01/B01: 1~9, A (10), B (11), C (12); When connected to card F01: 1~6.						

For example, if AXIS 0, AXIS 1, and AXIS 2 are effective, but AXIS 3 is invalid, as shown in the third row of the table above, the module occupies three node numbers, and the upper node number dial has a range of 1-10. If the upper node number dial points to 2, the lower node number dial should point to 4 (2 + 2), as shown in the illustration below. This indicates that AXIS 0 corresponds to a node number of 2, AXIS 1 corresponds to a node number of 3, AXIS 2 corresponds to a node number of 4, and AXIS 3 is invalid.



The module occupies node numbers 1 ~ 4, the command functions distinguish the individual axes via the NodeID parameter, and the SlotID parameter inputs 0.

9.1.2 Distinguishing Mode 1 and 2

	Mode 1	Mode 2
Mode setting	Dip switch is at position 1	Dip switch is at position 2 (ON)
Node number setting	Only occupies one node number	Occupies node numbers 1 ~ 4
	The upper and lower dials point to the same number	The upper dial points to the starting node, and the lower dial points to the ending node number
Motion command	Commands used only in pulse module mode 1	Corresponding to commands of servo axis
Interpolation motion	Only interpolation of internal axes within module	Can perform interpolation of internal axes within module, and can perform interpolation of axes within module and axes outside module
_DMC_01_get_devicetype Obtains DeviceType value	0x1C100191	0x14100191
Advantages		Module internal and external interpolation

9.1.3 Mode 1 Commands

■ Obtains DI/DO Status (_DMC_01_get_monitor)

1.Invoke _DMC_01_set_monitor (U16 CardNo, U16 NodeID, U16 SlotID, U16 monitorw);

U16 CardNo//card number

U16 NodeID//node number

U16 SlotID//SlotID, corresponds to each channel, inputs 0, 1, 2, 3.

U16 monitorw//inputs 0x27, or 0x28.

{2} invoke _DMC_01_get_monitor (U16 CardNo, U16 NodeID, U16 SlotID, U32 * value);

U16 CardNo//card number

U16 NodeID//node number

U16 SlotID//SlotID, corresponds to each channel, inputs 0, 1, 2, 3.

The following is the relationship between the binary bits and the DI/DO of the value obtained.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	Bit
	MEL	PEL	ORG	SLD	QZ	DI2	DI1			DO2	DO1					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Bit

9.1.4 Mode 2 Commands

■ Obtains DI/DO Status (_DMC_01_get_monitor)

Read DI status

1. Invoke _DMC_01_set_monitor (U16 CardNo, U16 NodeID, U16 SlotID, U16 monitorw);

U16 CardNo//card number

U16 NodeID//node number

U16 SlotID//SlotID, inputs 0.

U16 monitorw//inputs 0x27.

2. invoke _DMC_01_get_monitor (U16 CardNo, U16 NodeID, U16 SlotID, U32 *value);

U16 CardNo//card number

U16 NodeID//node number

U16 SlotID//SlotID, inputs 0.

The following is the relationship between the binary bits and the DI of the value obtained.

7	6	5	4	3	2	1	0	Bit
	MEL	PEL	ORG	SLD	QZ	DI2	DI1	

Read DO status

1. Invoke _DMC_01_set_monitor (U16 CardNo, U16 NodeID, U16 SlotID, U16 monitorw);

U16 CardNo//card number

U16 NodeID//node number

U16 SlotID//SlotID, inputs 0.

U16 monitorw//inputs 0x28.

2. invoke _DMC_01_get_monitor (U16 CardNo, U16 NodeID, U16 SlotID, U32 *value);

U16 CardNo//card number

U16 NodeID//node number

U16 SlotID//SlotID, inputs 0.

The following is the relationship between the binary bits and the DO of the value obtained.

7	6	5	4	3	2	1	0	Bit
						DO2	DO1	

■ Obtain DI status (_DMC_01_get_servo_DI)

invoke _DMC_01_get_servo_DI (U16 CardNo, U16 NodeID, U16 SlotID, U16 *servo_DI);

U16 CardNo//card number

U16 NodeID//node number

U16 SlotID//SlotID, inputs 0.

The following is the relationship between the binary bits and the DI of the servo_DI value obtained.

7	6	5	4	3	2	1	0	Bit
	MEL	PEL	ORG	SLD	QZ	DI2	DI1	

■ Obtain DO status (_DMC_01_get_servo_DO)

Invoke _DMC_01_get_servo_DO (U16 CardNo, U16 NodeID, U16 SlotID, U16 *servo_DO);

U16 CardNo//card number

U16 NodeID//node number

U16 SlotID//SlotID, inputs 0.

The following is the relationship between the binary bits and the DO of the servo_DO value obtained.

7	6	5	4	3	2	1	0	Bit
						DO2	DO1	

(This page is intentionally left blank.)

Chapter 10 Integrated Extension Module

10.1 Integrated Node Module ASD-DMC-GA01

The GA01 module is an accessing point. It connects each GE modules to the DMCNET network.

10.1.1 GA01/GE Pairing and GA01 Node Number Settings

1. One GA01 can connect to a maximum of 8 GE function modules, of which there may be a maximum of four GE01PH modules.
2. One GE01PH module occupies one node number.
3. A maximum of 4 GE16MN modules and a maximum of 4 GE16NT modules may jointly occupy one node number.

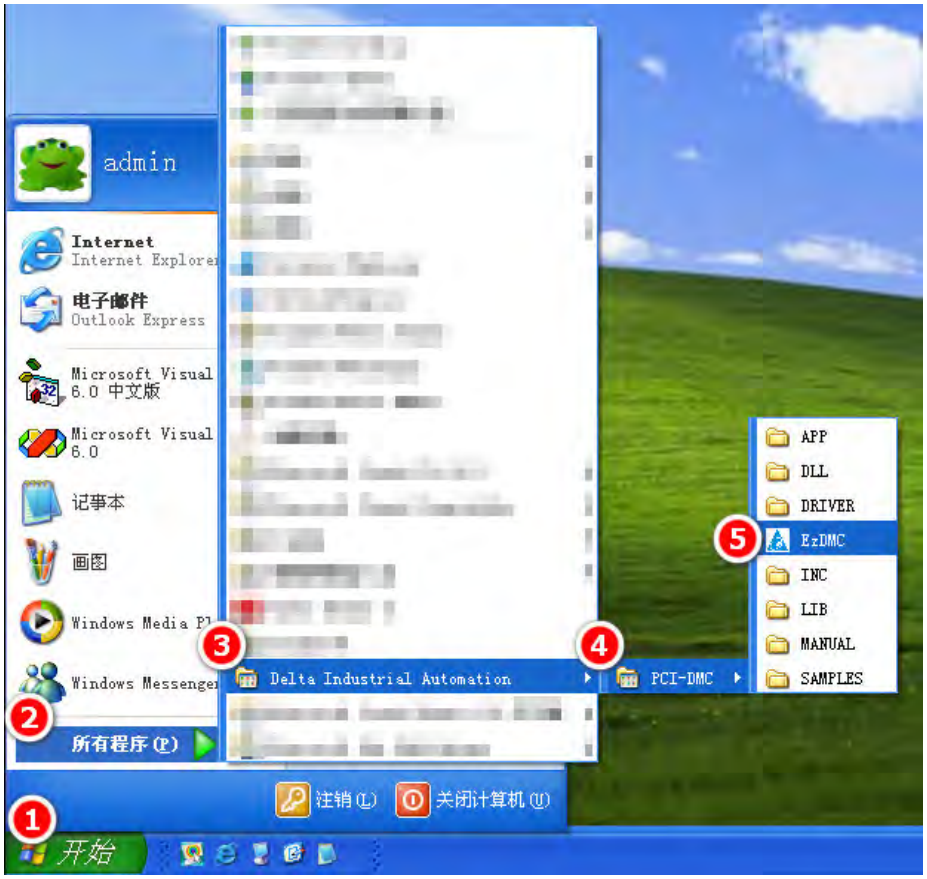
ADDR1 on the GA01 module's upper node number dial designates the starting node, and ADDR2 on the lower node number dial designates the ending node number.

Effective ranges of upper and lower node number dials: 1~9, A (10), B (11), C (12).

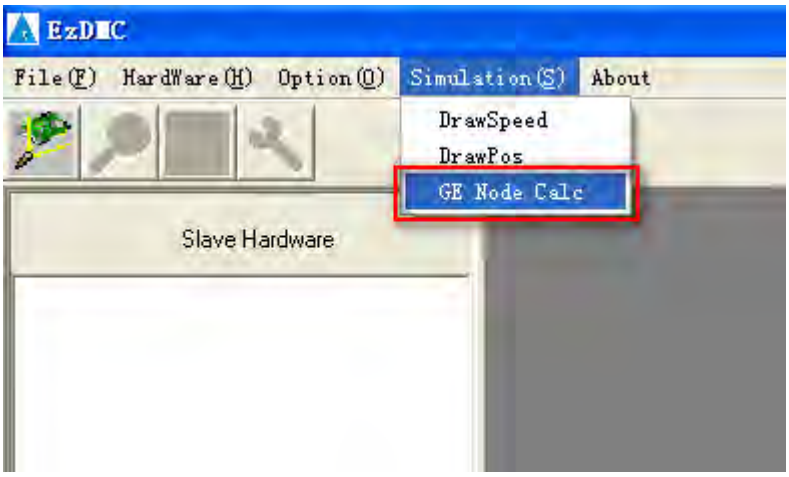
Note: When connected to card F01: If there is 01PH, the 01PH node number must in the range of 1 ~ 6.

10.1.2 GA01 Node Number Setting Helper

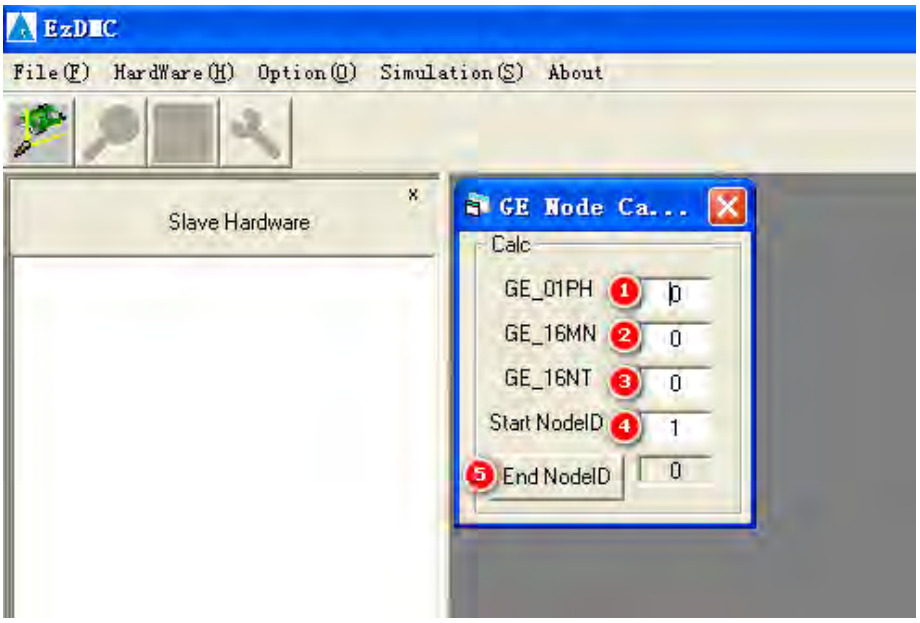
Step 1: Click on the Start menu, select All Programs, select Delta Industrial Automation, select PCI-DMC, and select EzDMC to launch EzDMC software.



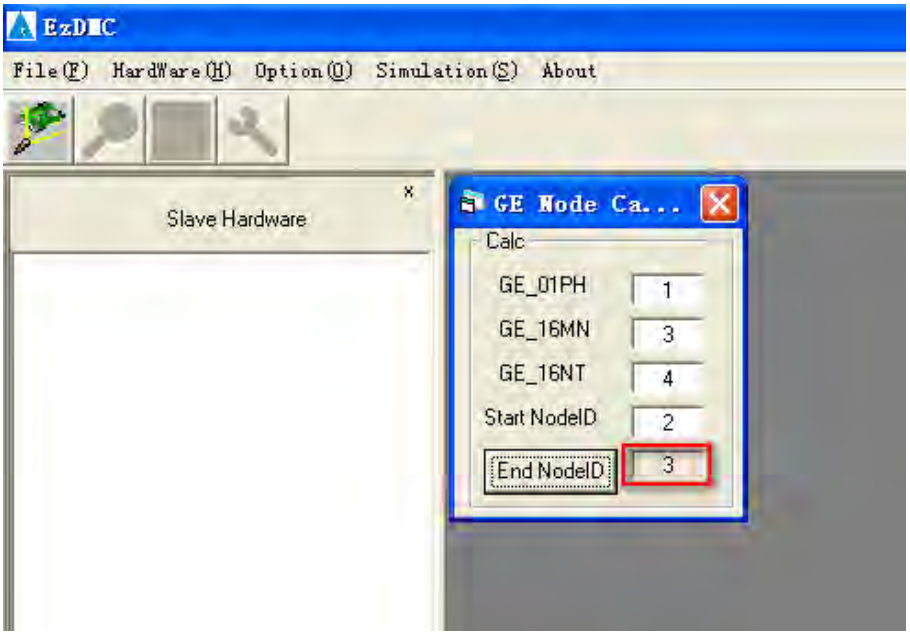
Step 2: Click on GE Node Calc in the Simulation menu.



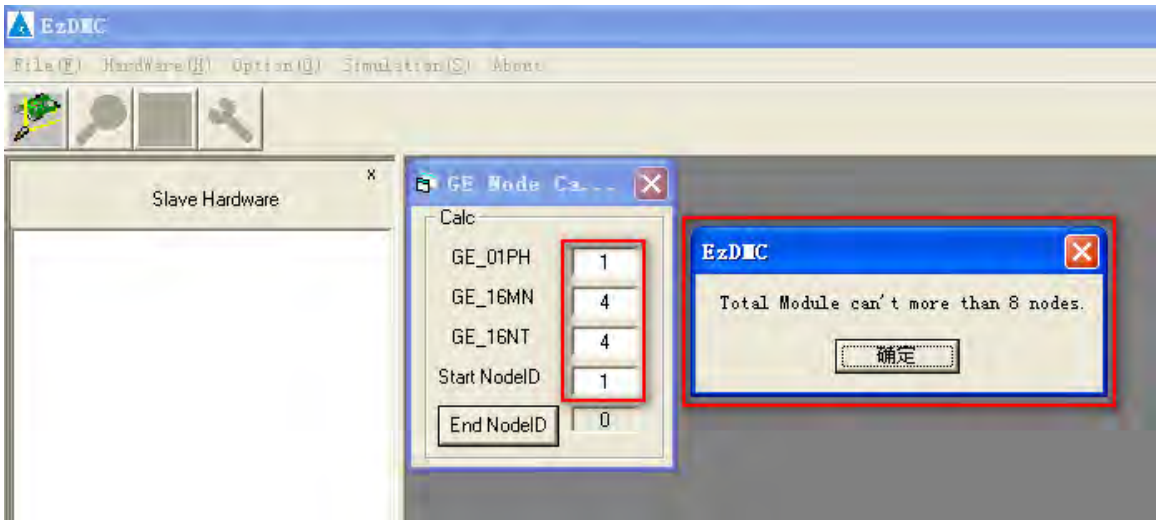
Step 3: In sequence, enter the number of GE01PH modules, GE16MN modules, and G16NT modules, and the upper node number dial starting node for the GA01 module, then click on [End NodeID].



Step 4: Value in the red bracket in the illustration below is the required setting of the lower node number dial. In this example, there is one GE01PH module, three GE16MN modules, and four G16NT modules, the upper node number dial starting node for the GA01 module is 2, and the calculated lower node number dial value is 3. In this situation, the GA/GE module occupies node numbers 2 and 3.



In the following example, there is one GE01PH module, four GE16MN modules, and four G16NT modules, the upper node number dial starting node for the GA01 module is 1. This does not comply with the rule that one GA01 module cannot be connected with more than 8 GE functional modules.



In the following example, there are no GE01PH modules, five GE16MN modules, and no G16NT modules, and the upper node number dial starting node for the GA01 module is 1. In accordance with the rule that a maximum of 4 GE16MN modules and a maximum of 4 GE16NT modules may jointly occupy one node number, the extra GE16MN module must occupy one node number, and the calculated lower node number dial value is 2. In this situation, the GA/GE module occupies node numbers 1 and 2.