PCI-8134

LabVIEWâ VIs

(For Windows 95/NT)

User’ s Manual

Manual Rev.1.00a: March 6, 2000

The information in this document is subject to change without prior notice in order to improve reliability, design and VI and does not represent a commitment on the part of the manufacturer.

In no event will the manufacturer be liable for direct, indirect, special, incidental, or consequential damages arising out of the use or inability to use the product or documentation, even if advised of the possibility of such damages.

Trademarks

PCI-8134 is a registered trademark of ADLINK Technology Inc.
IBM PC is a registered trademark of International Business Machines Corporation. Intel is a registered trademark of Intel Corporation. Other product names mentioned herein are used for identification purposes only and may be trademarks and/or registered trademarks of their respective companies.

Getting service from ADLINK

- **Customer Satisfaction is always the most important thing for ADLINK Tech Inc. If you need any help or service, please contact us and get it.**

ADLINK Technology Inc.			
Web Site	http://www.adlink.com.tw		
Sales & Service	service@adlink.com.tw		
Technical Support	NuDAQ	nudaq@adlink.com.tw	
	NuDAM	nudam@adlink.com.tw	
	NuIPC	nuipc@adlink.com.tw	
	NuPRO	nupro@adlink.com.tw	
	Software	sw@adlink.com.tw	
	AMB	amb@adlink.com.tw	
TEL	+886-2-82265877	FAX	+886-2-82265717
Address	9F, No. 166, Jian Yi Road, Chungho City, Taipei, 235 Taiwan, R.O.C.		

- **Please inform or FAX us of your detailed information for a prompt, satisfactory and constant service.**

Detailed Company Information			
Company/Organization			
Contact Person			
E-mail Address			
Address			
Country			
TEL		FAX	
Web Site			

Questions	
Product Model	
Environment to Use	□OS : ____________ □Computer Brand : ____________ □M/B : □CPU : □Chipset : □Bios : □Video Card : □Network Interface Card : □Other :
Challenge Description	
Suggestions for ADLINK	

CONTENTS

How to Use This Manual................................. i

CHAPTER 1 Introduction to PCI-8134 LabVIEW VIs........................... 1

1.1 PCI-8134 LabVIEW VIs/95 Installation....................................... 1

1.2 PCI-8134 LabVIEW VIs/NT Installation...................................... 4

1.3 Using PCI-8134 LabVIEW Vis in LabVIEW................................. 6

1.4 How to get help 8

1.5 PCI-8134 LabVIEW VIs/NT Uninstallation.................................. 9

CHAPTER 2 PCI-8134 VI Function Library 11

2.1 List of VIs 11

2.2 Common Argument.. 14

2.3 Initialization.. 16

2.4 Pulse Input / Output Configuration.. 19

2.5 Continuously Motion Move 21

2.6 Trapezoidal Motion Mode... 25

2.7 S-Curve Profile Motion.. 32

2.8 Multiple Axes Point to Point Motion... 38

2.9 Linear and Circular Interpolated Motion..................................... 41

2.10 Interpolation Parameters Configuring... 46

2.11 Home Return 50

2.12 Manual Pulser Motion..52
2.13 Motion Status..54
2.14 Servo Drive Interface 55
2.15 I/O Control and Monitoring...58
2.16 Position Control..60

CHAPTER 3 PCI-8134 VI Samples............63

3.1 Sample introduction..63
3.2 The Detail of These VIs 66

How to Use This Manual

This manual is designed to help you to use the PCI-8134 Motion Control card in NI LabVIEW® software package. The manual describes how to install and use the software library to meet your requirements and help you to program your own software applications. This manual is organized as follows:

- Chapter 1, "Introduction to PCI-8134 LabVIEW VIs" describes how to install the software.
- Chapter 2, "PCI-8134 VIs Function Descriptions" describe all VI functions supported by PCI-8134 LabVIEW VIs.
- Chapter 3, "PCI-8134 VI Samples" describes how to run the PCI-8134 VI samples.

1

Introduction to PCI-8134 LabVIEW VIs

1.1 PCI-8134 LabVIEW VIs/95 Installation

1.1.1 System Requirements

PCI-8134 LabVIEW VIs/95 requires the following minimum configuration:

- An IBM PC/AT or compatible or a CompactPCI system, running Windows 95/98
- A hard disk with enough disk space to install PCI-8134 LabVIEW VIs/95
- A 1.44 MB 3.5 inch floppy disk drive, or a CD-ROM drive.
- Application development system: National Instruments LabVIEW for Windows 95.

1.1.2 Installation

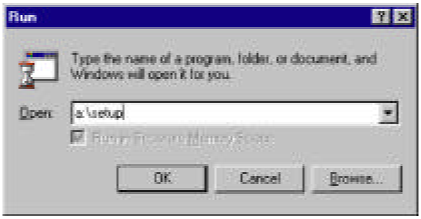
The Setup program provided by PCI-8134 LabVIEW VIs/95 performs all tasks necessary for installing the software.

With ADLINK’s “PCI-8134 LabVIEW VIs/95” diskettes:

Step 1. Place the “PCI-8134 LabVIEW VIs/95 Disk” in the 3.5-inch floppy drive A:

Step 2. If Windows 95 is loaded, choose Run from the taskbar.

Step 3. Type A:\SETUP in the Run dialog box.



Type the name of a program, folder, or document, and Windows will open it for you.

Open: a:\setup

☑ Run in Separate Memory Space

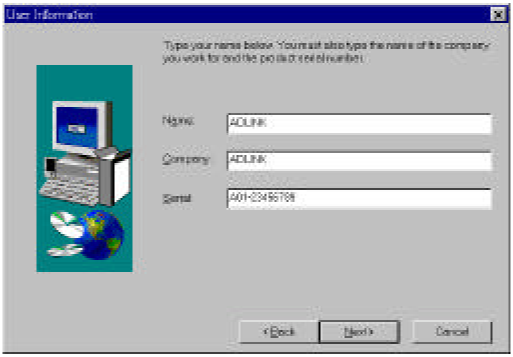
With “ADLINK All-In-One Compact Disc”:

Step 1. Place “ADLINK All-In-One Compact Disc” in the CD-ROM drive.

Step 2. If autorun setup program is not invoked, execute x: \setup.exe (x indicates the CD-ROM drive).

Step 3. Select NuDAQ PCI→PCI8134→Drivers→Windows 95 LabVIEW Driver to install the software.

Setup first displays a Welcome dialog box. Please click “Next” button to go to the next step.



Type your name below. You must also type the name of the company you work for and the product serial number.

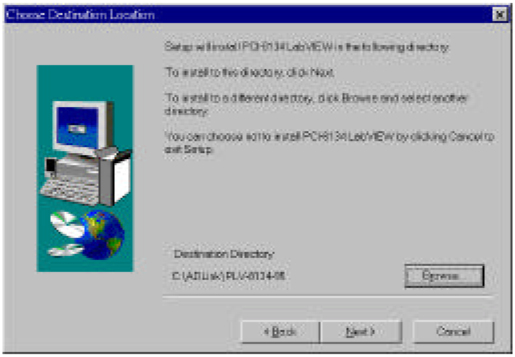
Name: ADLINK

Company: ADLINK

Serial: A01-23456789

Then Setup will display a “User Information” dialog box. Please fill items in the dialog box (including the serial number data). Then click “Next” button to go on installation. You have to enter the valid serial number to get a valid license (use upper case). Otherwise you can only use PCIS-ISG driver in 120 minutes demo mode.

Setup then prompts the following dialog box for you to specify the destination directory for PCI-8134 LabVIEW VIs/95. The default path is C:\ADLINK\PLV-8134-95. If you want to install the software in another directory, please enter the directory you would like to install PCI-8134 LabVIEW VIs/95.


Choose Destination Location

Setup will install PCI-8134 LabVIEW in the following directory.

To install to this directory, click Next.

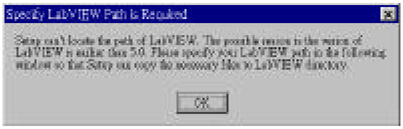
To install to a different directory, click Browse and select another directory.

You can choose not to install PCI-8134 LabVIEW by clicking Cancel to exit Setup.

Destination Directory

C:\ADLink\PLV-8134-95

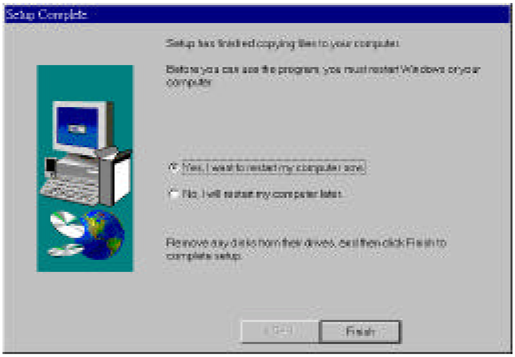
Setup will install LabVIEW function palette for PCI-8134 in your LabVIEW. If the version of your LabVIEW is earlier than 5.0, the message will be shown as below:


Specify LabVIEW Path is Required

Setup can't locate the path of LabVIEW. The possible reason is the version of LabVIEW is earlier than 5.0. Please specify your LabVIEW path in the following window so that Setup can copy the necessary files to LabVIEW directory.

And please specify your LabVIEW path.

Setup must restart your computer, please press "Finish" button to complete the installation.


Setup Complete

Setup has finished copying files to your computer.

Before you can use the program, you must restart Windows or your computer.

● Yes, I want to restart my computer now.

○ No, I will restart my computer later.

Remove any disks from their drives, and then click Finish to complete setup.

1.2 PCI-8134 LabVIEW VIs/NT Installation

1.2.1 System Requirements

PCI-8134 LabVIEW VIs/NT requires the following minimum configuration:

- An IBM PC/AT or compatible or a CompactPCI system, running Windows NT version 4.0 or later
- A hard disk with enough disk space to install PCI-8134 LabVIEW VIs/NT
- A 1.44 MB 3.5 inch floppy disk drive, or a CD-ROM drive.
- Application development system: National Instrument LabVIEW for Windows NT.

1.2.2 Installation

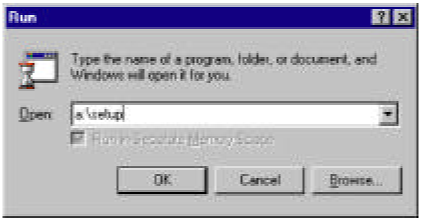
The Setup program provided by PCI-8134 LabVIEW VIs/NT performs all tasks necessary for installing the software.

With ADLINK' s "PCI-8134 LabVIEW VIs/NT" diskettes:

Step 1. Place the "PCI-8134 LabVIEW VIs/NT Disk" in the 3.5-inch floppy drive A:

Step 2. If Windows NT is loaded, choose Run from the taskbar.

Step 3. Type A:\SETUP in the Run dialog box.



Type the name of a program, folder, or document, and Windows will open it for you.

Open: a:\setup

☑ Run in Separate Memory Space

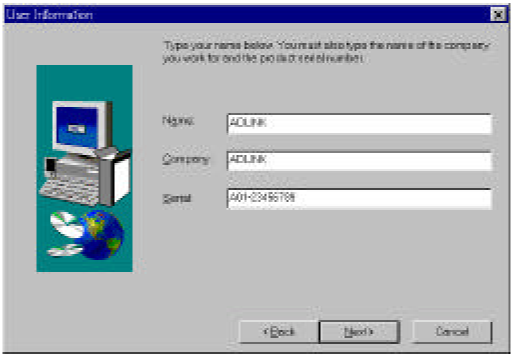
With "ADLINK All-In-One Compact Disc":

Step 1. Place "ADLINK All-In-One Compact Disc" in the CD-ROM drive.

Step 2. If autorun setup program is not invoked, execute x: \setup.exe (x indicates the CD-ROM drive).

Step 3. Select NuDAQ PCI→PCI8134→Drivers→Windows NT LabVIEW Driver to install the software.

Setup first displays a Welcome dialog box. Please click “Next” button to go to the next step.



User Information

Type your name below. You must also type the name of the company you work for and the product serial number.

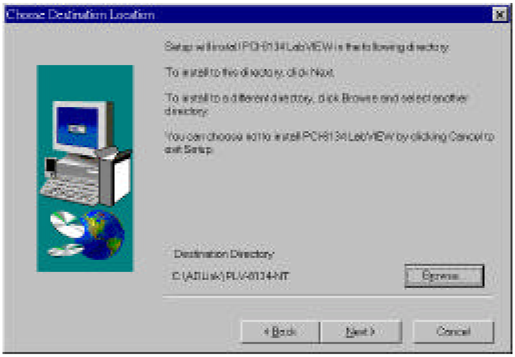
Name: ADLINK

Company: ADLINK

Serial: A01-23456789

Then Setup will display a “User Information” dialog box. Please fill items in the dialog box (including the serial number data). Then click “Next” button to go on installation. You have to enter the valid serial number to get a valid license (use upper case). Otherwise you can only use PCIS-ISG driver in 120 minutes demo mode.

Setup then prompts the following dialog box for you to specify the destination directory for PCI-8134 LabVIEW VIs/NT. The default path is C:\ADLINK\PLV-8134-NT. If you want to install the software in another directory, please enter the directory you would like to install PCI-8134 LabVIEW VIs/NT.



Choose Destination Location

Setup will install PCI-8134 LabVIEW in the following directory.

To install to this directory, click Next.

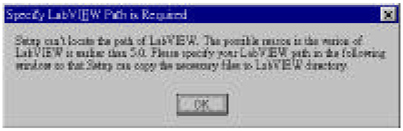
To install to a different directory, click Browse and select another directory.

You can choose not to install PCI-8134 LabVIEW by clicking Cancel to exit Setup.

Destination Directory

C:\ADLink\PLV-8134-NT

Setup will install LabVIEW function palette for PCI-8134 in your LabVIEW. If the version of your LabVIEW is earlier than 5.0, the message will be shown as below:

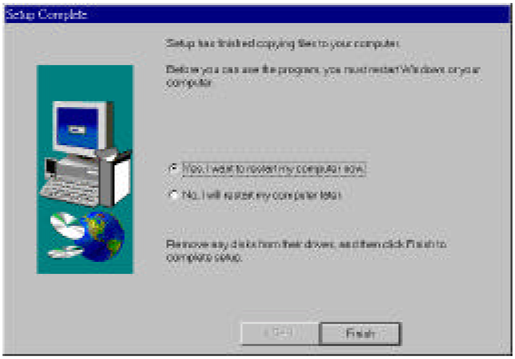


Specify LabVIEW Path is Required

Setup can't locate the path of LabVIEW. The possible reason is the version of LabVIEW is earlier than 5.0. Please specify your LabVIEW path in the following window so that Setup can copy the necessary files to LabVIEW directory.

And please specify your LabVIEW path.

Setup must restart your computer, please press “Finish” button to complete the installation.



Setup Complete

Setup has finished copying files to your computer.

Before you can use the program, you must restart Windows or your computer.

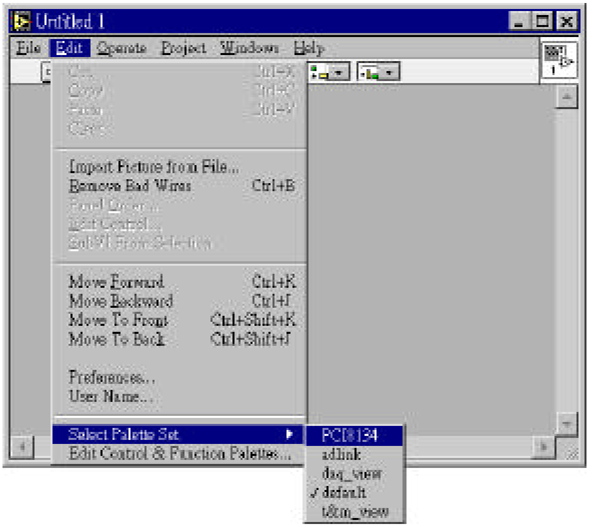
● Yes, I want to restart my computer now.

○ No, I will restart my computer later.

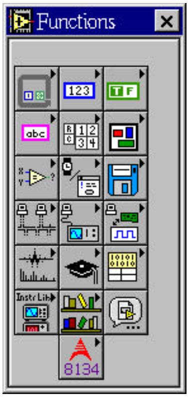
Remove any disks from their drives, and then click Finish to complete setup.

1.3 Using PCI-8134 LabVIEW Vis in LabVIEW

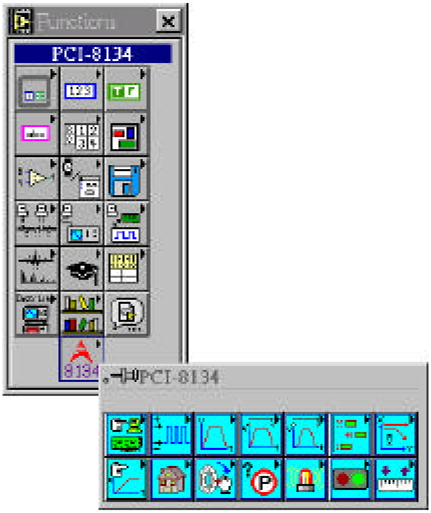
To use PCI-8134 LabVIEW VIs, you may switch the palette to the “PCI8134” view. Select “Select Palette Set” command in the “Edit” menu from panel or block diagram in LabVIEW. Then select the “PCI8134” view from the menu setup ring.



The “Functions” palette then becomes to the following:



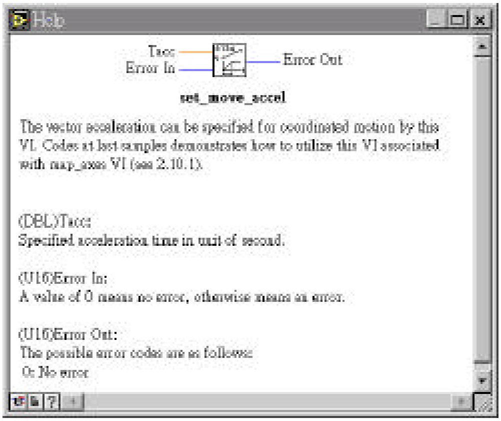
You can find PCI-8134 VIs in “8134” icon.



1.4 How to get help

You can find the detailed description of each VI by the following ways:

1. Select “Show Help” command of “Help” menu in LabVIEW. When you put the mouse cursor on PCI-8134 VI, LabVIEW will show the description of the VI.



set_move_accel

The vector acceleration can be specified for coordinated motion by this VI. Codes at last samples demonstrates how to utilize this VI associated with map_axes VI (see 2.10.1).

(DBL)Tacc:
Specified acceleration time in unit of second.

(U16)Error In:
A value of 0 means no error, otherwise means an error.

(U16)Error Out:
The possible error codes are as follows:
0: No error

2. PDF manual files in <Install Dir>\Manual directory.

1.5 PCI-8134 LabVIEW VIs/NT Uninstallation

PCI-8134 LabVIEW VIs/NT has the capability of automatic uninstall.

To uninstall PCI-8134 LabVIEW VIs/NT, open the “Control Panel”, double-click “Add/Remove Programs”, select “PCI-8134 LabVIEW VIs/NT” to uninstall it.

After uninstall, all files in the directory of PCI-8134 LabVIEW VIs will be removed, but the palette of the PCI-8134 still exist in your LabVIEW directory. If you don’t use them again, you can find “User.lib“ and “Menu\813495” (or “Menu\8134NT”) folder and remove them.

2

PCI-8134 VI Function Library

In this character, the PCI-8134 LabVIEW VIs for Windows NT are described.

2.1 List of VIs

All of the PCI-8134 VIs are divided to 14 groups in LabVIEW Function Palette. They are listed as below:

Initialization

8134_Initial	Software initialization
8134_Close	Software Close
8134_Set_Config	Configure PCI-8134 according to Motion Creator
8134_Get_IRQ_Channel	Get the PCI-8134 card' s IRQ number
8134_Get_Base_Addr	Get the PCI-8134 card' s base address

Pulse Input/Output Configuration

set_pls_outmode	Set pulse command output mode
set_pls_iptmode	Set encoder input mode
set_cnt_src	Set counter input source

Continuously Motion Mode

v_move	Accelerate an axis to a constant velocity with

	trapezoidal profile
sv_move	Accelerate an axis to a constant velocity with S-curve profile
v_change	Change speed on the fly
v_stop	Decelerate to stop

Trapezoidal Motion Mode

a_move	Perform an absolute trapezoidal profile move
start_a_move	Begin an absolute trapezidal profile move
r_move	Perform a relative trapezoidal profile move
start_r_move	Begin a relative trapezoidal profile move
t_move	Perform a relative non-symmetrical trapezoidal profile move
start_t_move	Begin a relative non-symmetrical trapezidal profile move
start_ta_move	Begin an absolute non-symmetrical trapezidal profile move
ta_move	Perform an absolute non-symmetrical trapezoidal profile move
wait_for_done	Wait for an axis to finish

S-Curve Profile Motion

s_move	Perform an absolute S-curve profile move
start_s_move	Begin an absolute S-curve profile move
rs_move	Perform a relative S-curve profile move
start_rs_move	Begin a relative S-curve profile move
tas_move	Perform an absolute non-symmetrical S-curve profile move
start_tas_move	Begin an absolute non-symmetrical S-curve profile move

Multiple Axes Point to Point Motion

start_move_all	Begin a multi-axis trapezodial profile move
move_all	Perform a multi-axis trapezodial profile move
wait_for_all	Wait for all axes to finish

Linear / Circular Interpolated Motion

start_move_xy	Begin a 2-axis linear interpolated move for X & Y
start_move_zu	Begin a 2-axis linear interpolated move for Z & U
move_xy	Perform a 2-axis linear interpolated move for X & Y
move_zu	Perform a 2-axis linear interpolated move for Z & U
recover_xy	Recover 2-axis mode to singal axis mode for X & Y
recover_zu	Recover 2-axis mode to singal axis mode for Z & U
arc_xy	Perform a 2-axis circular interpolated move for X & Y
arc_zu	Perform a 2-axis circular interpolated move for Z & U

Interpolation Parameters Configuring

map_axes	Maps coordinated motion axes x, y, z.…
set_move_speed	Set the vector velocity
set_move_accel	Set the vector acceleration time
set_arc_division	Set the interpolation arc segment length
arc_optimization	Enable/Disable optimum acceleration calculations for arce
set_move_ratio	Set the axis resolution ratios

Home Return Mode

set_home_config	Set or get the home/index logic configuration
home_move	Begin a home return action

Manual Pulser Motion

set_manu_iptmode	Set pulser input mode and operation mode
manu_move	Begin a manual pulser movement

Motion Status

motion_done	Returns TRUE if motion done

Servo Drive Interface

set_alm_logic	Set alarm logic and alarm mode
set_inp_logic	Set In-Position logic and enable/disable
set_sd_logic	Set slow down point logic and enable/disable
set_erc_enable	Set the ERC output enable/disable

I/O Control and Monitoring

8134_Set_SVON	Set the state of general purpose output bit
get_io_status	Get all the I/O staus of PCI-8134

Position Control

set/get_position	Set or get current actual position
set/get_command	Set or get current command position

2.2 Common Argument

There are some common arguments in PCI-8134 VI Functions. They are described as below.

2.2.1 cardNo

It means card number. The card number is the first argument for some initialization VIs. A numeric card number for the card initialized. If you use multiple PCI-8134 cards in your machine, it must use card number to differentiate every PCI-8134 card. The range of card number in PCI-8134 is between 0 and 11. The value of this output terminal is used as the value of input terminal card number of other PCI-8134 LabVIEW VIs.

You can get the numbers of existing PCI-8134 cards by calling *8134_Initial* (See 2.3.1), and will put in *existCard*. The maximum number of *cardNo* is equal to *existCardI*-1.

One thing need to be noticed by users is to identify the card number of PCI-8134 when multiple cards are applied. The card number of one PCI-8134 depends on the locations on the PCI slots. They are numbered either from left to right or right to left on the PCI slots. You should test every card to obtain the correct card number. These card numbers will effect the corresponding axis number on the cards.

2.2.2 axis

The axis number is the first argument for most funcions called in the library. So it is important to identify the axis number before writing application programs. The range of axis in PCI-8134 is between 0 and 47. The corresponding axis number on each card will be:

cardNo	0	1	2	3	4	5	6	7	8	9	10	11
Axis1	0	4	8	12	16	20	24	28	32	36	40	44
Axis2	1	5	9	13	17	21	25	29	33	37	41	45
Axis3	2	6	10	14	18	22	26	30	34	38	42	46
Axis4	3	7	11	15	19	23	27	31	35	39	43	47

2.2.3 Error In

The error condition occurred before this VI executes. This code default to no error. If an error has already occurred, this VI does not perform any operation. A value of 0 means no error, otherwise means an error.

The *8134_Initial* VI is often the first sub VI called in your diagram, and you may not need to wire this input.

2.2.4 Error Out

It returns an error code. If the *Error In* indicated an error, the *Error Out* contains the same value. Otherwise, *Error Out* describes the error status of this VI.

The error code of the every VI will be listed in its statement.

2.2.5 str_vel, max_vel

These two arguments mean starting velocity and maximum velocity in unit of pulse per second.

The value of *max_vel* must lager then *str_vel*, otherwise, the VI will cause an unexpected motion but not occur error. If they are all negative, they will all be turn to positive and make motion.

If the value of *max_vel* equal to zero, or *max_vel* and *str_vel* aren' t all positive or negative, the VI will return an error code.

2.2.6 Tacc, Tdec ,Tlacc, Tldec, Tsacc, Tsdec

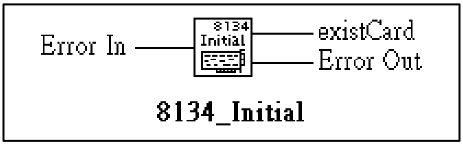
The value of Tacc and Tdec are the acceleration and deceleration time in unit of second, Tlacc and Tldec are the linear acceleration and deceleration time in unit of second, and Tsacc and Tsdec are the S-Curve acceleration and deceleration time in unit of second.

They are must be positive, otherwise, they will be turn to positive and make motion. If they equal to zero, the VI will return an error code.

2.3 Initialization

2.3.1 8134_Initial

This VI is used to initialize PCI-8134 card. Every PCI-8134 card has to be initialized by this VI before using other VIs.



U16 Error In: A value of 0 means no error, otherwise means an error.

The *8134_Initial* VI is often the first sub VI called in your diagram, and you may not need to wire this input.

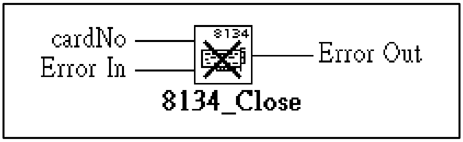
U16 existCard: The numbers of existing PCI-8134 cards.

U16 Error Out: The possible error codes are as follows:

0: No error
1: Board not initialized
8: PIC bios not exist

2.3.2 8134_Close

This VI is used to close PCI-8134 card and release the PCI-8134 related resources, which should be called at the end of an application.



U16 cardNo: A numeric card id for the card initialized. The range of *cardNo* is between 0 and 11.

U16 Error In: A value of 0 means no error, otherwise means an error.

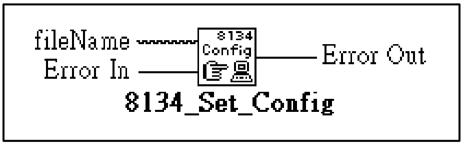
U16 Error Out: The possible error codes are as follows:

0: No error
1: Board not initialized

8: PIC bios not exist

2.3.3 8134_Set_Config

This VI is used to configure PCI-8134 card. All the I/O configurations and some operating modes appeared on “ Axis Configuration Window” of Motion Creator will be set to PCI-8134. Click “ Save Configuration” button on the“ Axis Configuration Window” if you want to use this VI in the application program. Click “ Save Configuration” button will save all the configurations to a file call “ *8134.cfg*” . This file will appear in the “ WINDOWS\SYSTEM\” directory.



abc fileName: The name and locate of the configuration file.

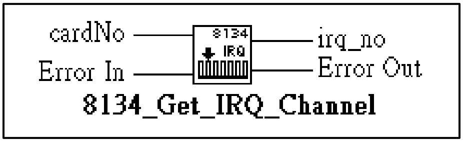
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error
1: Board not initialized
8: PIC bios not exist

2.3.4 8134_Get_IRQ_Channel

This VI is used to get the PCI-8134 card’ s IRQ number.



U16 cardNo: A numeric card id for the card initialized. The range of *cardNo* is between 0 and 11.

U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Irq_no: The PCI-8134 card’ s IRQ number.

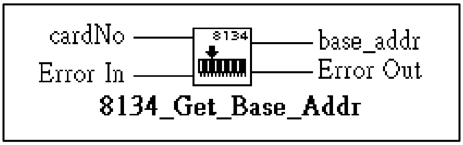
U16 Error Out: The possible error codes are as follows:

0: No error

1: Board not initialized
8: PIC bios not exist

2.3.5 8134_Get_Base_Addr

This VI is used to get the PCI-8134 card’ s base address.



U16 cardNo: A numeric card id for the card initialized. The range of *cardNo* is between 0 and 11.

U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Base_addr: The PCI-8134 card’ s base address.

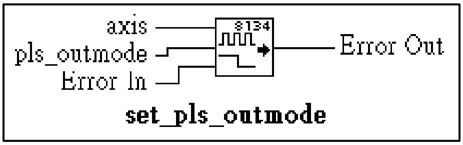
U16 Error Out: The possible error codes are as follows:

0: No error
1: Board not initialized
8: PIC bios not exist

2.4 Pulse Input / Output Configuration

2.4.1 set_pls_outmode

Configure the output modes of command pulse. There are two modes for command pulse output.



I16 axis: Axis number designated to configure pulse Input/Output. The range of *axis* is between 0 and 47.

I16 pls_outmode: Setting of command pulse output mode for OUT and DIR pins.

pls_outmode=0, OUT/DIR type pulse output.
pls_outmode=1, CW/CCW type pulse output.

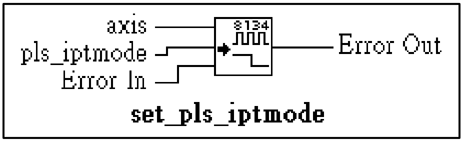
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error

2.4.2 set_pls_iptmode

Configure the input modes of external feedback pulse. There are four types for feedback pulse input. Note that this VI makes sense only when *cnt_src* parameter in *set_cnt_src* VI (see 2.4.3) is enabled.



I16 axis: Axis number designated to configure pulse Input/Output. The range of *axis* is between 0 and 47.

I16 pls_iptmode: Setting of encoder feedback pulse input mode for EA and EB pins.

pls_iptmode=0, 1X AB phase type pulse input.
pls_iptmode=1, 2X AB phase type pulse input.
pls_iptmode=2, 4X AB phase type pulse input.
pls_iptmode=3, CW/CCW type pulse input.

U16 Error In: A value of 0 means no error, otherwise means an error.

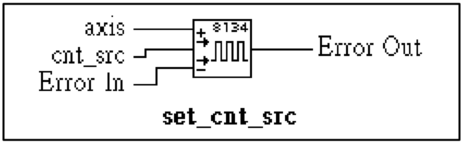
U16 Error Out: The possible error codes are as follows:

0: No error

2.4.3 set_cnt_src

If external encoder feedback is available in the system, set the *cnt_src* parameter in this VI to Enabled state. Then internal 28-bit up/down counter will count according configuration of *set_pls_iptmode* VI (see 2.4.2).

Or the counter will count the command pulse output.



I16 axis: Axis number designated to configure pulse Input/Output. The range of *axis* is between 0 and 47.

I16 cnt_src: Counter source.

cnt_src=0, counter source from command pulse
cnt_src=1, counter source from external input EA, EB

U16 Error In: A value of 0 means no error, otherwise means an error.

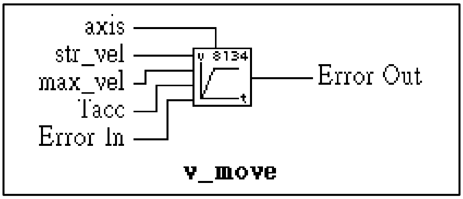
U16 Error Out: The possible error codes are as follows:

0: No error

2.5 Continuously Motion Move

2.5.1 v_move

This VI is used to accelerate an axis to the specified constant velocity. The axis will continue to travel at a constant velocity until the velocity is changed or the axis is commanded to stop. The direction is determined by the sign of velocity parameter.



I16 axis: Axis number designated to move. The range of *axis* is between 0 and 47.

DBL str_vel: Starting velocity in unit of pulse per second.

DBL max_vel: Maximum velocity in unit of pulse per second.

DBL Tacc: Specified acceleration time in unit of second.

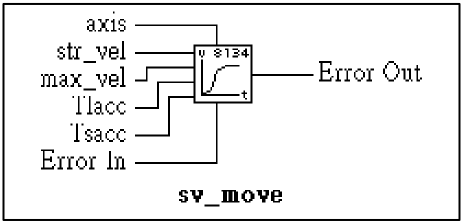
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error
12: Velocity setting error

2.5.2 sv_move

This VI is similar to *v_move* VI (see 2.5.1) but accelerating with S-curve.



I16 axis: Axis number designated to move. The range of *axis* is between 0 and 47.

DBL str_vel: Starting velocity in unit of pulse per second.

DBL max_vel: Maximum velocity in unit of pulse per second.

DBL Tlacc: Specified linear acceleration time in unit of second.

DBL Tsacc: Specified S-curve acceleration time in unit of second.

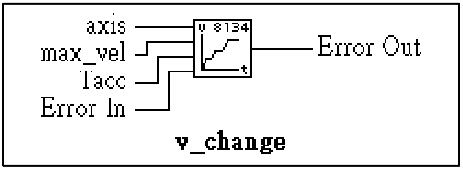
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error
12: Velocity setting error

2.5.3 v_change

You can change the velocity profile of command pulse ouput during operation by this VI. This VI changes the maximum velocity setting during operation. However, if you operate under “Preset Mode” (like *start_a_move* VI (see 2.6.1),…), you are not allowed to change the acceleration parameter during operation because the deceleration point is pre-determined. But changing the acceleration parameter when operating under “Constant Velocity Mode” is valid.



I16 axis: Axis number designated to move. The range of *axis* is between 0 and 47.

DBL max_vel: Maximum velocity in unit of pulse per second.

DBL Tacc: Specified acceleration time in unit of second.

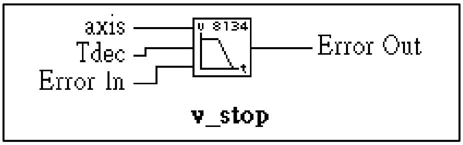
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error
12: Velocity setting error

2.5.4 v_stop

This VI is used to decelerate an axis to stop. This VI is also useful when *preset move* (both trapezoidal and S-curve motion), *manu_move* VI (see 2.12.2) or *home_move* VI (see 2.11.2) is performed.



I16 axis: Axis number designated to stop. The range of *axis* is between 0 and 47.

DBL Tdec: Specified deceleration time in unit of second.

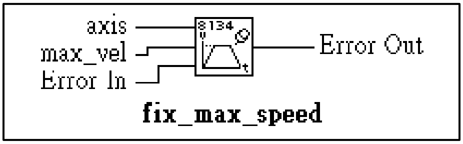
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error

2.5.5 fix_max_speed

This VI is used to preassign the range of the possible velocity change by *v_change* VI (see 2.5.3).



I16 axis: Axis number designated to move. The range of *axis* is between 0 and 47.

DBL max_vel: Maximum velocity in unit of pulse per second.

U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

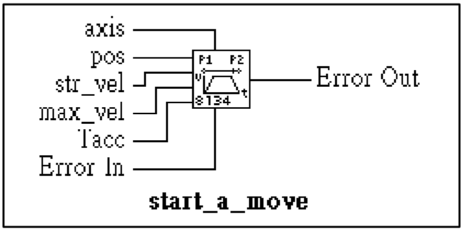
0: No error

12: Velocity setting error

2.6 Trapezoidal Motion Mode

2.6.1 start_a_move

This VI causes the axis to accelerate from a starting velocity, slew at constant velocity, and decelerate to stop at the specified absolute position, immediately returning control to the program. The acceleration rate is equal to the deceleration rate.



I16 axis: Axis number designated to move. The range of *axis* is between 0 and 47.

DBL pos: Specified absolute position to move.

DBL str_vel: Starting velocity in unit of pulse per second.

DBL max_vel: Maximum velocity in unit of pulse per second.

DBL Tacc: Specified acceleration time in unit of second.

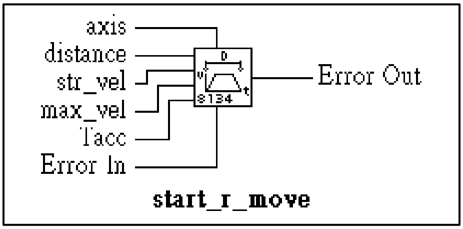
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error
12: Velocity setting error
18: Move error

2.6.2 start_r_move

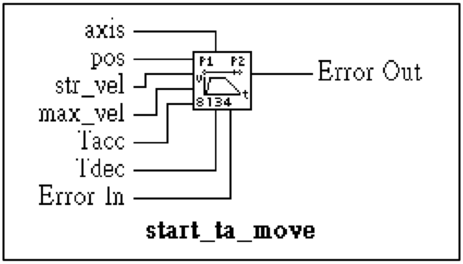
This VI causes the axis to accelerate from a starting velocity, slew at constant velocity, and decelerate to stop at the relative distance, immediately returning control to the program. The acceleration rate is equal to the deceleration rate.



I16	axis:	Axis number designated to move. The range of *axis* is between 0 and 47.
DBL	distance:	Specified relative distance to move.
DBL	str_vel:	Starting velocity in unit of pulse per second.
DBL	max_vel:	Maximum velocity in unit of pulse per second.
DBL	Tacc:	Specified acceleration time in unit of second.
U16	Error In:	A value of 0 means no error, otherwise means an error.
U16	Error Out:	The possible error codes are as follows: **0:** No error **12:** Velocity setting error **18:** Move error

2.6.3 start_ta_move

This VI causes the axis to accelerate from a starting velocity, slew at constant velocity, and decelerate to stop at the specified absolute position, immediately returning control to the program.



I16	axis:	Axis number designated to move. The range of *axis* is between 0 and 47.

DBL pos: Specified absolute position to move.

DBL str_vel: Starting velocity in unit of pulse per second.

DBL max_vel: Maximum velocity in unit of pulse per second.

DBL Tacc: Specified acceleration time in unit of second.

DBL Tdec: Specified deceleration time in unit of second.

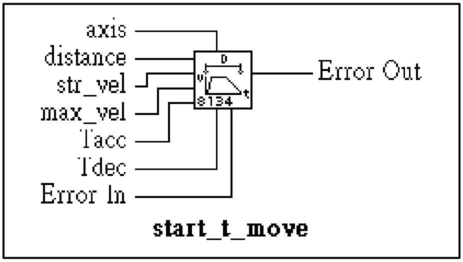
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error
12: Velocity setting error
18: Move error

2.6.4 start_t_move

This VI causes the axis to accelerate from a starting velocity, slew at constant velocity, and decelerate to stop at the relative distance, immediately returning control to the program. The acceleration rate is equal to the deceleration rate.



I16 axis: Axis number designated to move. The range of *axis* is between 0 and 47.

DBL distance: Specified relative distance to move.

DBL str_vel: Starting velocity in unit of pulse per second.

DBL max_vel: Maximum velocity in unit of pulse per second.

DBL Tacc: Specified acceleration time in unit of second.

DBL Tdec: Specified deceleration time in unit of second.

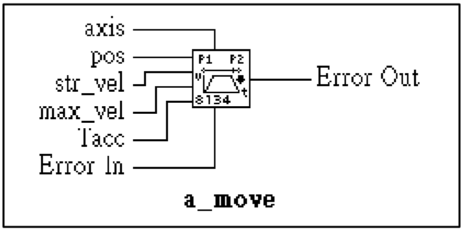
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error
12: Velocity setting error
18: Move error

2.6.5 a_move

This VI is as same as *start_a_move* VI (see 2.6.1) but starts an absolute coordinate move and waits for completion.



I16 axis: Axis number designated to move. The range of *axis* is between 0 and 47.

DBL pos: Specified absolute position to move.

DBL str_vel: Starting velocity in unit of pulse per second.

DBL max_vel: Maximum velocity in unit of pulse per second.

DBL Tacc: Specified acceleration time in unit of second.

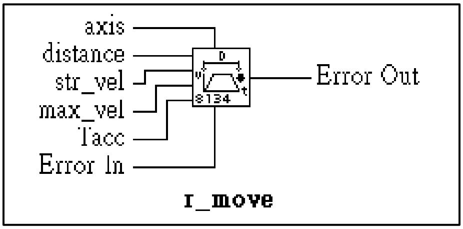
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error
12: Velocity setting error
18: Move error

2.6.6 r_move

This VI is as same as *start_r_move* VI (see 2.6.2) but starts an absolute coordinate move and waits for completion.



I16 axis: Axis number designated to move. The range of *axis* is between 0 and 47.

DBL distance: Specified relative distance to move.

DBL str_vel: Starting velocity in unit of pulse per second.

DBL max_vel: Maximum velocity in unit of pulse per second.

DBL Tacc: Specified acceleration time in unit of second.

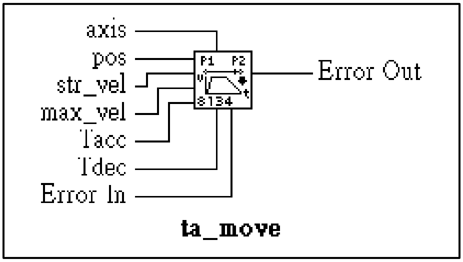
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error
12: Velocity setting error
18: Move error

2.6.7 ta_move

This VI is as same as *start_ta_move* VI (see 2.6.3) but starts an absolute coordinate move and waits for completion.

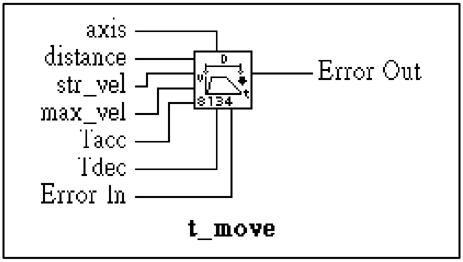


I16 axis: Axis number designated to move. The range of *axis* is between 0 and 47.

DBL	pos:	Specified absolute position to move.
DBL	str_vel:	Starting velocity in unit of pulse per second.
DBL	max_vel:	Maximum velocity in unit of pulse per second.
DBL	Tacc:	Specified acceleration time in unit of second.
DBL	Tdec:	Specified deceleration time in unit of second.
U16	Error In:	A value of 0 means no error, otherwise means an error.
U16	Error Out:	The possible error codes are as follows: **0:** No error **12:** Velocity setting error **18:** Move error

2.6.8 t_move

This VI is as same as *start_t_move* VI (see 2.6.4) but starts an absolute coordinate move and waits for completion.



I16	axis:	Axis number designated to move. The range of *axis* is between 0 and 47.
DBL	distance:	Specified relative distance to move.
DBL	str_vel:	Starting velocity in unit of pulse per second.
DBL	max_vel:	Maximum velocity in unit of pulse per second.
DBL	Tacc:	Specified acceleration time in unit of second.
DBL	Tdec:	Specified deceleration time in unit of second.

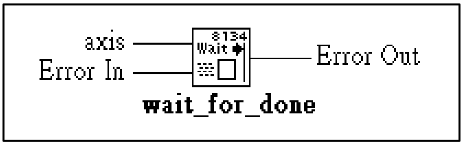
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0:	No error
12:	Velocity setting error
18:	Move error

2.6.9 wait_for_done

This VI waits for the motion to complete.



I16 axis: Axis number designated to move. The range of *axis* is between 0 and 47.

U16 Error In: A value of 0 means no error, otherwise means an error.

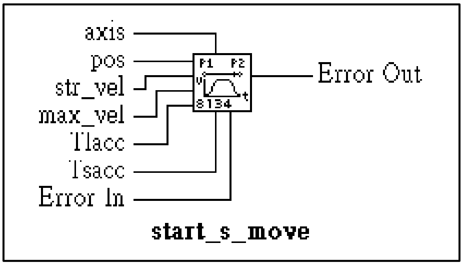
U16 Error Out: The possible error codes are as follows:

0:	No error
18:	Move error

2.7 S-Curve Profile Motion

2.7.1 start_s_move

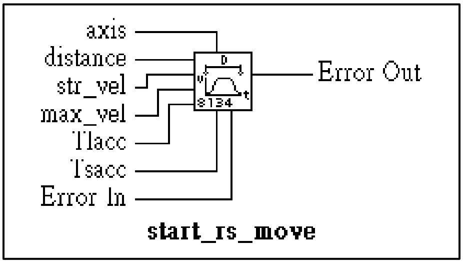
This VI causes the axis to accelerate from a starting velocity, slew at constant velocity, and decelerate to stop at the specified absolute position, immediately returning control to the program. The acceleration rate is equal to the deceleration rate.



I16	axis:	Axis number designated to move. The range of *axis* is between 0 and 47.
DBL	pos:	Specified absolute position to move.
DBL	str_vel:	Starting velocity in unit of pulse per second.
DBL	max_vel:	Maximum velocity in unit of pulse per second.
DBL	Tlacc:	Specified linear acceleration time in unit of second.
DBL	Tsacc:	Specified S-curve acceleration time in unit of second.
U16	Error In:	A value of 0 means no error, otherwise means an error.
U16	Error Out:	The possible error codes are as follows: **0:** No error **12:** Velocity setting error **18:** Move error

2.7.2 start_rs_move

This VI causes the axis to accelerate from a starting velocity, slew at constant velocity, and decelerate to stop at the relative distance, immediately returning control to the program. The acceleration rate is equal to the deceleration rate.



I16 axis: Axis number designated to move. The range of *axis* is between 0 and 47.

DBL distance: Specified relative distance to move.

DBL str_vel: Starting velocity in unit of pulse per second.

DBL max_vel: Maximum velocity in unit of pulse per second.

DBL Tlacc: Specified linear acceleration time in unit of second.

DBL Tsacc: Specified S-curve acceleration time in unit of second.

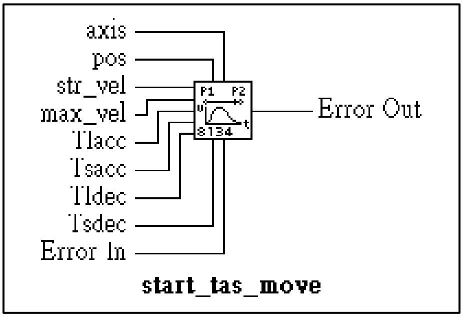
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error
12: Velocity setting error
18: Move error

2.7.3 start_tas_move

This VI causes the axis to accelerate from a starting velocity, slew at constant velocity, and decelerate to stop at the specified absolute position, immediately returning control to the program.



I16 axis: Axis number designated to move. The range of *axis* is between 0 and 47.

DBL pos: Specified absolute position to move.

DBL str_vel: Starting velocity in unit of pulse per second.

DBL max_vel: Maximum velocity in unit of pulse per second.

DBL Tlacc: Specified linear acceleration time in unit of second.

DBL Tsacc: Specified S-curve acceleration time in unit of second.

DBL Tldec: Specified linear deceleration time in unit of second.

DBL Tsdec: Specified S-curve deceleration time in unit of second.

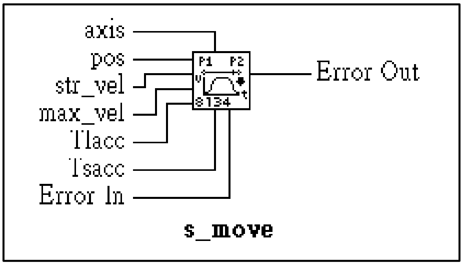
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error
18: Move error

2.7.4 s_move

This VI is as same as *start_s_move* VI (see 2.7.1) but starts an absolute coordinate move and waits for completion.



I16 axis: Axis number designated to move. The range of *axis* is between 0 and 47.

DBL pos: Specified absolute position to move.

DBL str_vel: Starting velocity in unit of pulse per second.

DBL max_vel: Maximum velocity in unit of pulse per second.

DBL Tlacc: Specified linear acceleration time in unit of second.

DBL Tsacc: Specified S-curve acceleration time in unit of second.

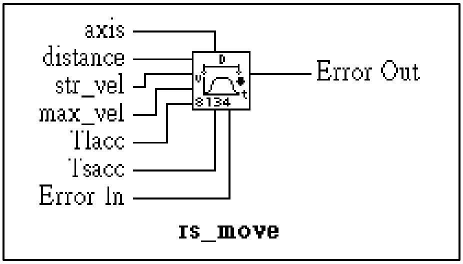
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error
18: Move error

2.7.5 rs_move

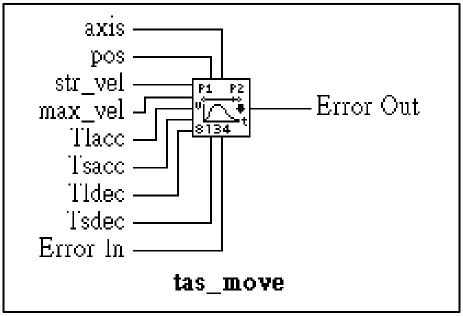
This VI is as same as *start_rs_move* VI (see 2.7.2) but starts an absolute coordinate move and waits for completion.



I16	axis:	Axis number designated to move. The range of *axis* is between 0 and 47.
DBL	distance:	Specified relative distance to move.
DBL	str_vel:	Starting velocity in unit of pulse per second.
DBL	max_vel:	Maximum velocity in unit of pulse per second.
DBL	Tlacc:	Specified linear acceleration time in unit of second.
DBL	Tsacc:	Specified S-curve acceleration time in unit of second.
U16	Error In:	A value of 0 means no error, otherwise means an error.
U16	Error Out:	The possible error codes are as follows: **0:** No error **18:** Move error

2.7.6 tas_move

This VI is as same as *start_tas_move* VI (see 2.7.3) but starts an absolute coordinate move and waits for completion.



I16	axis:	Axis number designated to move. The range of *axis* is between 0 and 47.
DBL	pos:	Specified absolute position to move.
DBL	str_vel:	Starting velocity in unit of pulse per second.
DBL	max_vel:	Maximum velocity in unit of pulse per second.

DBL Tlacc: Specified linear acceleration time in unit of second.

DBL Tsacc: Specified S-curve acceleration time in unit of second.

DBL Tldec: Specified linear deceleration time in unit of second.

DBL Tsdec: Specified S-curve deceleration time in unit of second.

U16 Error In: A value of 0 means no error, otherwise means an error.

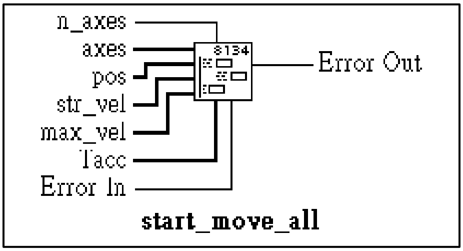
U16 Error Out: The possible error codes are as follows:

0:	No error
18:	Move error

2.8 Multiple Axes Point to Point Motion

2.8.1 start_move_all

This VI causes the specified axes to accelerate from a starting velocity, slew at constant velocity, and decelerate to stop at the specified absolute position, immediately returning control to the program. The move axes are specified by axes and the number of axes are defined by *n_axes*. The acceleration rate of all axes is equal to the deceleration rate. It guarantees that motion begins on all axes at the same sample time.

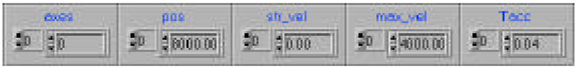


If you want to move axis 0 and axis 4 to position 8000.0 and 120000.0 respectively, and choose velocities and acelerations that are propotional to the ratio of distances, then the axes will arrive at their endpoints at the same time (simultaneous motion). These arguments will be:

axes	positions	str_vel	max_vel	Tacc
0	8000.0	0.0	4000.0	0.04
4	12000.0	0.0	6000.0	0.06

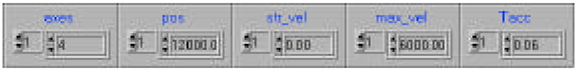
The method for setting these arguments in this VI will be shown as below.

First axes:



axes	pos	str_vel	max_vel	Tacc
0 0	0 8000.00	0 0.00	0 4000.00	0 0.04

Second axes:



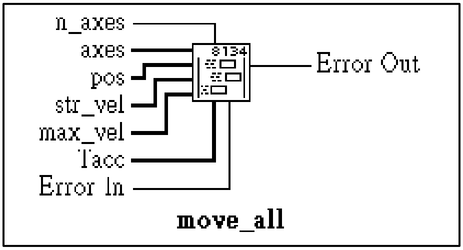
axes	pos	str_vel	max_vel	Tacc
1 4	1 12000.0	1 0.00	1 6000.00	1 0.06

I16 n_axes: Number of axes for simultaneous motion.

I16	axes:	Specified axes number array designated to move.
DBL	pos:	Specified position array in unit of pulse.
DBL	str_vel:	Starting velocity array in unit of pulse per second.
DBL	max_vel:	Maximum velocity array in unit of pulse per second.
DBL	Tacc:	Acceleration time array in unit of second.
U16	Error In:	A value of 0 means no error, otherwise means an error.
U16	Error Out:	The possible error codes are as follows: **0:** No error **18:** Move error

2.8.2 move_all

This VI is as same as *start_move_all* VI (see 2.8.1) but starts the motion and waits for completion. It guarantees that motion begins on all axes at the same sample time.



I16	n_axes:	Number of axes for simultaneous motion.
I16	axes:	Specified axes number array designated to move.
DBL	pos:	Specified position array in unit of pulse.
DBL	str_vel:	Starting velocity array in unit of pulse per second.
DBL	max_vel:	Maximum velocity array in unit of pulse per second.
DBL	Tacc:	Acceleration time array in unit of second.
U16	Error In:	A value of 0 means no error, otherwise means an error.

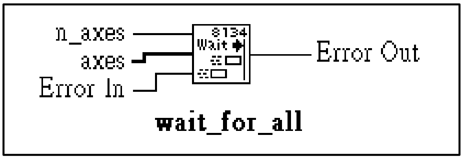
U16 Error Out: The possible error codes are as follows:

0: No error
18: Move error

2.8.3 wait_for_all

This VI waits for the motion to complete for all of the specified axes.

The method to set arguments in this VI is as same as *start_move_all* VI (see 2.8.1).



I16 n_axes: Number of axes for simultaneous motion.

I16 axes: Specified axes number array designated to move.

U16 Error In: A value of 0 means no error, otherwise means an error.

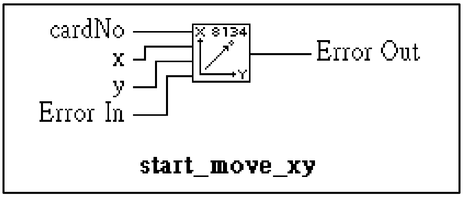
U16 Error Out: The possible error codes are as follows:

0: No error
18: Move error

2.9 Linear and Circular Interpolated Motion

2.9.1 start_move_xy

This VI cause a linear interpolation motion between X and Y axes. The moving speed should be set before performing these VIs.



U16 cardNo: A numeric card id for the card initialized. The range of *cardNo* is between 0 and 11.

DBL x: Absolute target X axis position of linear interpolation motion.

DBL y: Absolute target Y axis position of linear interpolation motion.

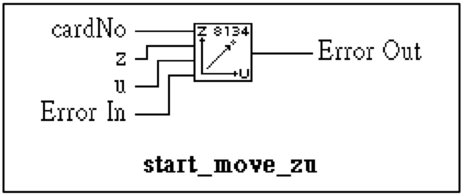
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error

2.9.2 start_move_zu

This VI causes a linear interpolation motion between Z and U axes. The moving speed should be set before performing these VIs.



U16 cardNo: A numeric card id for the card initialized. The range of *cardNo* is between 0 and 11.

DBL z: Absolute target Z axis position of linear interpolation motion.

DBL u: Absolute target U axis position of linear interpolation motion.

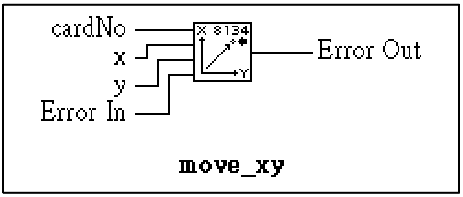
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error

2.9.3 move_xy

This VI is as same as *start_move_xy* VI (see 2.9.1) but waits for completion. The moving speed should be set before performing thes e VIs.



U16 cardNo: A numeric card id for the card initialized. The range of *cardNo* is between 0 and 11.

DBL x: Absolute target X axis position of linear interpolation motion.

DBL y: Absolute target Y axis position of linear interpolation motion.

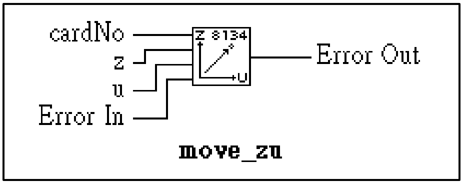
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error

2.9.4 move_zu

This VI cause a linear interpolation motion between Z and U axes and wait for completion. The moving speed should be set before performing these VIs.



U16 cardNo: A numeric card id for the card initialized. The range of *cardNo* is between 0 and 11.

DBL z: Absolute target Z axis position of linear interpolation motion.

DBL u: Absolute target U axis position of linear interpolation motion.

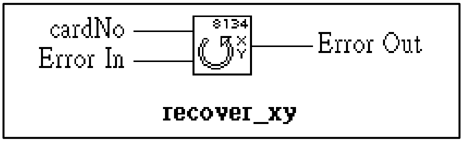
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error

2.9.5 recover_xy

When you use start_move_xy (see 2.9.1) or move_xy (see 2.9.3) VI, PCI-8134 will switch its operation mode to interpolation. You must use this VI to switch to single axis operation mode if you want to use single axis motion after interpolation.



U16 cardNo: A numeric card id for the card initialized. The range of *cardNo* is between 0 and 11.

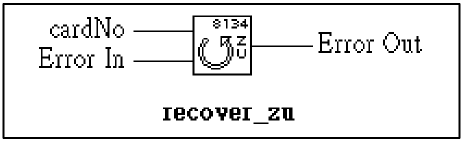
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error

2.9.6 recover_zu

When you use start_move_zu (see 2.9.2) or move_zu (see **錯誤! 找不到參照來源。**) VI, PCI-8134 will switch its operation mode to interpolation. You must use this VI to switch to single axis operation



mode if you want to use single axis motion after interpolation.

U16 cardNo: A numeric card id for the card initialized. The range of *cardNo* is between 0 and 11.

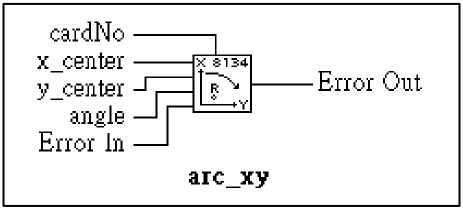
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error

2.9.7 arc_xy

This VI cause the axes to move along a circular arc and wait for completion. The arc starts from origin and continues through the specified angle. A positive value for angle produces clockwise arcs and a negative value produces counter-clockwise arcs. The center of the arc is specified by the parameters x_center and y_center. *set_arc_division* VI (see 2.10.4) specifies the maximum angle (in degrees) between successive points along the arc. The default angle is 5 degrees. The moving speed should be set before performing this VI.



U16 cardNo: A numeric card id for the card initialized. The range of *cardNo* is between 0 and 11.

DBL x_center: X axis center position of an arc.

DBL y_center: Y axis center position of an arc.

DBL angle Specified angle for an arc.

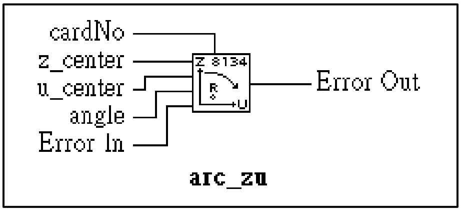
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error

2.9.8 arc_zu

This VI cause the axes to move along a circular arc and wait for completion. The arc starts from origin and continues through the specified angle. A positive value for angle produces clockwise arcs and a negative value produces counter-clockwise arcs. The center of the arc is specified by the parameters z_center and u_center. *set_arc_division* VI (see 2.10.4) specifies the maximum angle (in degrees) between successive points along the arc. The default angle is 5 degrees. The moving speed should be set before performing this VI.

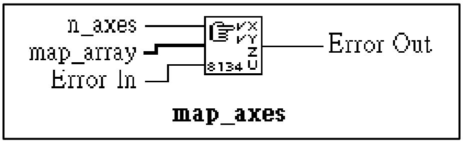


U16	cardNo:	A numeric card id for the card initialized. The range of *cardNo* is between 0 and 11.
DBL	z_center:	Z axis center position of an arc.
DBL	u_center:	U axis center position of an arc.
DBL	angle	Specified angle for an arc.
U16	Error In:	A value of 0 means no error, otherwise means an error.
U16	Error Out:	The possible error codes are as follows: **0:** No error

2.10 Interpolation Parameters Configuring

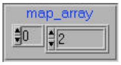
2.10.1 map_axes

This VI initializes a group of axes for coordinated motion. It must be called before any coordinated motion VI is used. For PCI-8134, coordinated motion is made only between two axes.



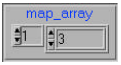
For example, if the Z and U coordinates correspond to axes 2 and 3, the following setting would be used to define the coordinate system:

First axes:



map_array
0 2

Second axes:



map_array
1 3

This condition only has two combinations, such as X and Y, Z and U. The axes number of this array may be 0 and 1, 2 and 3, and so on.

I16 n_axes: Number of axes for coordinated motion.

I16 map_array: Specified axes number array designated to move. It may be 0 and 1, 2 and 3, and so on.

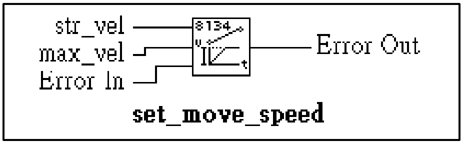
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error

2.10.2 set_move_speed

The vector velocity acceleration can be specified for coordinated motion by this VI. Codes at last samples demonstrates how to utilize this VI associated with *map_axes* VI (see 2.10.1).



DBL str_vel: Starting velocity in unit of pulse per second.

DBL max_vel: Maximum velocity in unit of pulse per second.

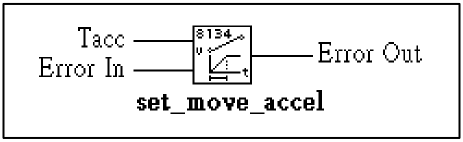
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error

2.10.3 set_move_accel

The vector acceleration can be specified for coordinated motion by this VI. Codes at last samples demonstrates how to utilize this VI associated with *map_axes* VI (see 2.10.1).



DBL Tacc: Specified acceleration time in unit of second.

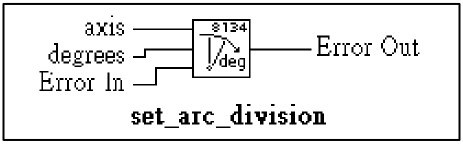
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error

2.10.4 set_arc_division

This VI specifies the maximum angle (in degrees) between successive points along the arc. The default is 5 degrees.



I16 axis: Axis number designated to configure.

DBL degrees: Maximum angle between successive points along the arc.

U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

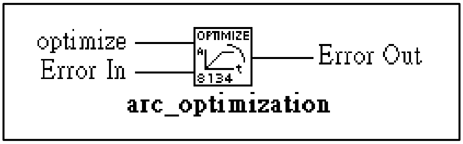
0: No error

2.10.5 arc_optimization

This VI enables (*optimize* = TRUE) or disable (*optimize* = FALSE) the automatic calculation of the optimum acceleration for an arc. The default s tate for arc optimization is enabled. When *arc_optimization* VI is enabled, circular intepolation is greatly improved by choosing the best acceleration for the motion. The optimum acceleration is given by the following formula:

$$A_{opt} = V^2/d;$$

where A_{opt} is the best acceleration, *V* is the *set_move_speed* VI (see 2.10.2) velocity, *d* is the segment length. If the acceleration is higher than A_{opt}, the linear portions may be noticeable. If the acceleration is lower than A_{opt}, the motion will be slowed during the arc and it will lose its roundness. Both *arc_xy* and *arc_zu* automatically change the acceleration to A_{opt} during the circular interpolated move.



I16 optimize: Set optimization enable or disable.

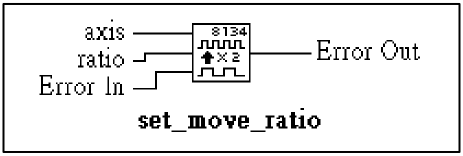
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error

2.10.6 set_move_ratio

This VI configures scale factors for the specified axis. Usually, the axes only need scale factors if their mechanical resolutions are different. For example, if the resolution of feedback sensors is two times resolution of command pulse, then *ratio* = 2.



I16 axis: Axis number designated to configure.

DBL ratio: Ratio of (feedback resolution) / (command resolution).

U16 Error In: A value of 0 means no error, otherwise means an error.

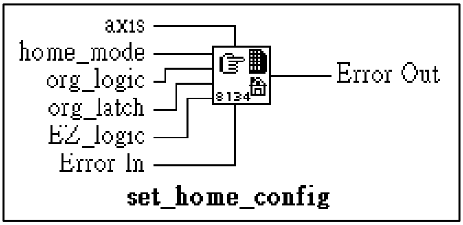
U16 Error Out: The possible error codes are as follows:

0: No error

2.11 Home Return

2.11.1 set_home_config

Configure the logic of origin switch and index signal needed for home_move VI. If you need to stop the axis after EZ signal is active (*home_mode* = 1 or 2), you should keep placing ORG signal in the ON status until the axis stop. If the pulse width of ORG signal is too short to keep it at ON status till EZ goes ON, you should select the *org_latch* as enable. The latched condition is cancelled by the next start or by disabling the *org_latch*. Three home return modes are available.



I16	axis:	Axis number designated to configure and perform home returning.
I16	home_mode:	Stopping modes for home return. home_mode=0, ORG active only. home_mode=1, ORG active and then EZ active to stop, high speed all the way. home_mode=2, ORG active and then EZ active to stop, high speed till ORG active then low speed till EZ active.

I16	org_logic:	Action logic configuration for ORG signal
		org_logic=0, Active low
		org_logic=1, Active high.

I16	org_latch:	Latch state control for ORG signal
		org_latch=0, Don’ t latch input
		org_latch=1, Latch input.

I16	EZ_logic:	Action logic configuration for EZ signal
		EZ_logic=0, Active low.
		EZ_logic=1, Active high.

U16	Error In:	A value of 0 means no error, otherwise means an error.

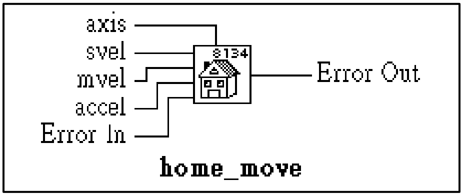
error.

U16 Error Out: The possible error codes are as follows:

0: No error

2.11.2 home_move

This VI will cause the axis to perform a home return move according to the setting of *set_home_config* VI (see 2.11.1). The direction of moving is determined by the sign of velocity parameter (*str_vel*, *max_vel*). Since the stopping condition of this VI is determined by *home_mode* setting, user should take care to select the initial moving direction. Or user should take care to handle the condition when limit switch is touched or other conditions that is possible causing the axis to stop. Executing *v_stop* VI (see 2.5.4) during *home_move* VI can also cause the axis to stop.



I16 axis: Axis number designated to configure and perform home returning.

DBL svel: Starting velocity in unit of pulse per second.

DBL mvel: Maximum velocity in unit of pulse per second.

DBL accel: Acceleration time in unit of second.

U16 Error In: A value of 0 means no error, otherwise means an error.

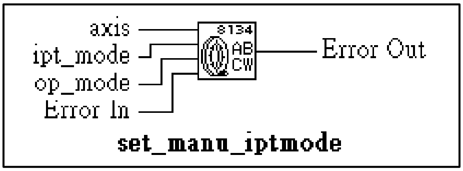
U16 Error Out: The possible error codes are as follows:

0: No error

2.12 Manual Pulser Motion

2.12.1 set_manu_iptmode

Four types of pulse input modes can be available for pulser or hand wheel. User can also move two axes simultaneously with one pulser by selecting the operation mode to *common mode*. Or move the axes independently by selecting the operation mode to *independent mode*.



I16 axis: Axis number designated to start manual move.

I16 ipt_mode: Setting of manual pulser input mode from PA and PB pins.

ipt_mode=0, 1X AB phase type pulse input.
ipt_mode=1, 2X AB phase type pulse input.
ipt_mode=2, 4X AB phase type pulse input.
ipt_mode=3, CW/CCW type pulse input.

I16 op_mode: Common or independent mode selection.

op_mode=0, Independent for each axis.
op_mode=1, PAX, PBX common for PAY, PBY, or PAZ, PBZ common for PAU, PBU.

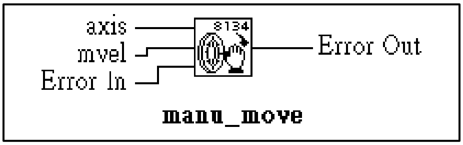
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error

2.12.2 manu_move

Begin to move the axis according to manual pulser input as this command is written. The maximum moving velocity is limited by *mvel* parameter. Not until the *v_stop* VI (see 2.5.4) command is written won' t system end the manual move mode.



I16 axis: Axis number designated to start manual move.

DBL mvel: Limitation for maximum velocity.

U16 Error In: A value of 0 means no error, otherwise means an error.

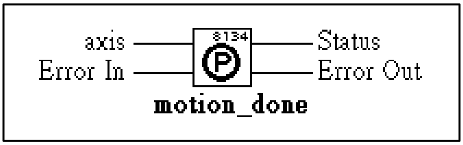
U16 Error Out: The possible error codes are as follows:

0: No error

2.13 Motion Status

2.13.1 motion_done

Return the motion status of PCI-8134. position.



I16 axis: Axis number of motion status.

U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Status: Definition value is as following:

0: The axis is busying.
1: A movement is finished.
2: The axis stops at positive limit switch.
3: The axis stops at negative limit switch.
4: The axis stops at origin switch.
5: The axis stops because the ALARM signal is active.

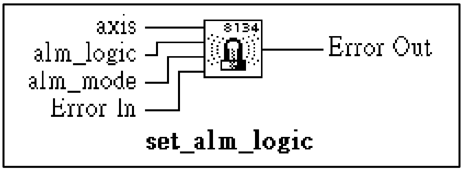
U16 Error Out: The possible error codes are as follows:

0: No error

2.14 Servo Drive Interface

2.14.1 set_alm_logic

Set the active logic of *ALARM* signal input from servo driver. Two reacting modes are available when *ALARM* signal is active.



I16 axis: Axis number designated to configure.

I16 alm_logic: Setting of active logic for ALARM signal.

alm_logic=0, Active LOW.
alm_logic=1, Active HIGH.

I16 alm_mode: Reacting modes when receiving ALARM signal.

alm_mode=0, Motor immediately stops.
alm_mode=1, Motor decelerates then stops.

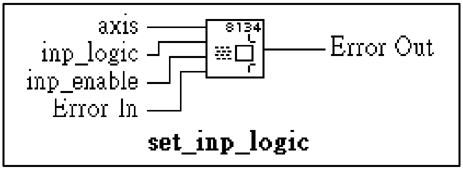
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error

2.14.2 set_inp_logic

Set the active logic of *In-Position* signal input from servo driver. Users can select whether they want to enable this VI. Default state is disabled.



I16 axis: Axis number designated to configure.

I16 inp_logic: Setting of active logic for INP signal.

inp_logic=0, Active LOW.
inp_logic=1, Active HIGH.

I16 inp_enable: INP enable/disable.

inp_enable=0, Disabled.
inp_enable=1, Enabled.

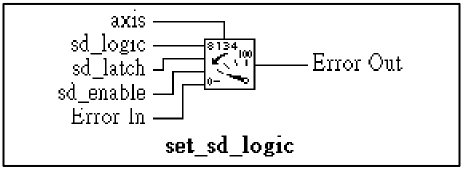
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error

2.14.3 set_sd_logic

Set the active logic and latch control of *SD* signal input from mechanical system. Users can select whether they want to enable this VI. Default state is disabled.



I16 axis: Axis number designated to configure.

I16 sd_logic: Setting of active logic for SD signal.

sd_logic=0, Active LOW.
sd_logic=1, Active HIGH.

I16 sd_latch: Setting of latch control for SD signal.

sd_logic=0, Do not latch.
sd_logic=1, Latch.

I16 sd_enable: Slow down point enable/disable.

sd_enable=0, Disabled
sd_enable=1, Enabled

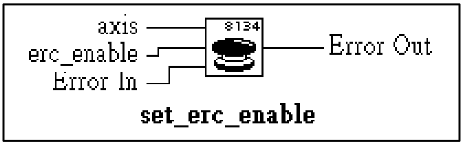
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error

2.14.4 set_erc_enable

You can set *ERC* pin output enable/disable by this VI. Default state is enabled.



I16 axis: Axis number designated to configure.

I16 erc_enable: ERC pin output enable/disable.

erc_enable=0, Disabled.
erc_enable=1, Enabled

U16 Error In: A value of 0 means no error, otherwise means an error.

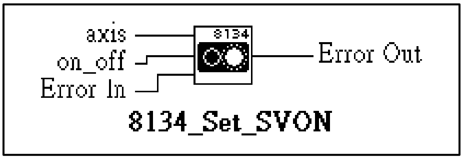
U16 Error Out: The possible error codes are as follows:

0: No error

2.15 I/O Control and Monitoring

2.15.1 8134_Set_SVON

Set the High/Low output state of general purpose output pin *SVON*.



I16 axis: Axis number for I/O control and monitoring.

I16 on_off: Setting for SVON pin digital output.

on_off=0, SVON is LOW.
on_off=1, SVON is HIGH.

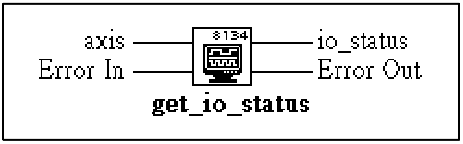
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error

2.15.2 get_io_status

Get all the I/O status for each axis.



The definition for each bit is as following:

Bit	Name	Description
0	+EL	Positive Limit Switch
1	-EL	Negative Limit Switch
2	+SD	Positive Slow Down Point
3	-SD	Negative Slow Down Point
4	ORG	Origin Switch
5	EZ	Index signal
6	ALM	Alarm Signal
7	SVON	SVON of PCL5023 pin output
8	RDY	RDY pin input
9	INT	Interrupt status
10	ERC	ERC pin output
11	INP	In-Position signal input

I16 axis: Axis number for I/O control and monitoring.

I16 io_status: I/O status word. Where “1’ is ON and “0” is OFF. ON/OFF state is read based on the corresponding set logic.

U16 Error In: A value of 0 means no error, otherwise means an error.

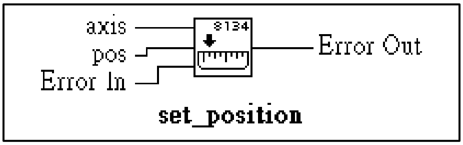
U16 Error Out: The possible error codes are as follows:

0: No error

2.16 Position Control

2.16.1 set_position

Changes the current actual position to the specified position.



I16 axis: Axis number designated to set and get position.

DBL pos: Actual position.

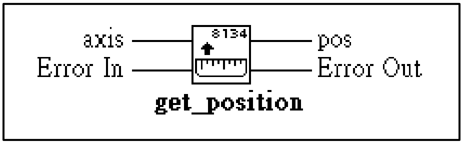
U16 Error In: A value of 0 means no error, otherwise means an error.

U16 Error Out: The possible error codes are as follows:

0: No error

2.16.2 get_position

Reads the current actual position. Note that when feedback signals is not available in the system, thus external encoder feedback is *Disabled* in *set_cnt_src* VI (see 2.4.3), the value gotten from this VI is command position.



I16 axis: Axis number designated to set and get position.

U16 Error In: A value of 0 means no error, otherwise means an error.

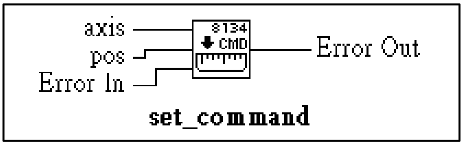
DBL pos: Actual position.

U16 Error Out: The possible error codes are as follows:

0: No error

2.16.3 set_ command

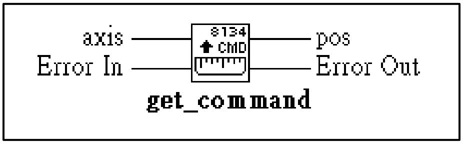
Changes the command position to the specified command position.



I16	axis:	Axis number designated to set and get position.
DBL	pos:	Command position.
U16	Error In:	A value of 0 means no error, otherwise means an error.
U16	Error Out:	The possible error codes are as follows: **0:** No error

2.16.4 get_command

Reads the current command position.



I16	axis:	Axis number designated to set and get position.
U16	Error In:	A value of 0 means no error, otherwise means an error.
DBL	pos:	Command position.
U16	Error Out:	The possible error codes are as follows: **0:** No error

3

PCI-8134 VI Samples

3.1 Sample introduction

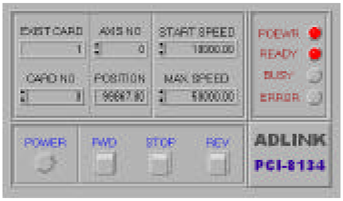
PCI-8134 LabVIEW VIs/NT provides four sample VIs for PCI-8134. You can see how to use PCI-8134 LabVIEW VIs to control this card.

These sample VIs are already installed in “8134 Samples” folder in the PCI-8134 LabVIEW VIs/NT directory, you can copy them to the directory that you want.

3.1.1 8134Ex1.vi

This sample displays how to control the absolute motion.

Click ⇨ bottom to run this VI, the control panel will be shown as below:



EXIST CARD: 1
AXIS NO: 0
START SPEED: 10000.00
CARD NO: 0
POSITION: 99987.80
MAX SPEED: 50000.00

When you press POWER to turn on, it will initialize all of the PCI-8134 cards, and makes all of the axis return to origin. When these steps complete, the READY light will turn on.

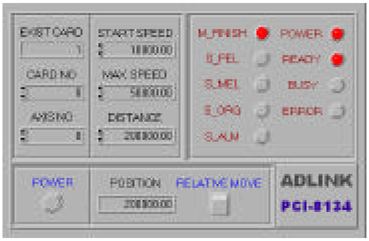
EXIST CARD shows how many cards that are installed correctly in your machine. Assign value for CARD NO, AXIS NO, and set value for START SPEED, MAXIMUN SPEED items. You can press FWD or REV to make it move forward or reverse. You can press STOP to stop motion. The position of the axis will be shown when it moves.

You can press POWER to turn off. If you want to close this VI, you must to turn off it before you close.

3.1.2 8134Ex2.vi

This sample displays how to control the relative motion.

Click ⇨ bottom to run this VI, the control panel will be shown as below:



EXIST CARD 1

START SPEED 10000.00

CARD NO 0

MAX SPEED 50000.00

AXIS NO 0

DISTANCE 200000.00

POSITION 200000.00

When you press POWER to turn on, it will initialize all of the PCI-8134 cards, and makes all of the axis return to origin. When these steps complete, the READY light will turn on.

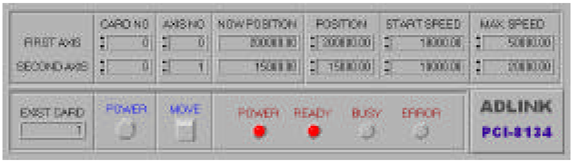
EXIST CARD shows how many cards that are installed correctly in your machine. Assign value for CARD NO, AXIS NO, and set value for START SPEED, MAXIMUN SPEED, and DISTANCE items. You can press RELATIVE MOVE to make it move. If the values of DISTANCE are positive, it will move forward, or it will move reverse. The position of the axis will be shown when it moves.

You can press POWER to turn off. If you want to close this VI, you must to turn off it before you close.

3.1.3 8134Ex3.vi

This sample displays how to control the multiple axes motion.

Click ⇨ bottom to run this VI, the control panel will be shown as below:



	CARD NO	AXIS NO	NOW POSITION	POSITION	START SPEED	MAX SPEED
FIRST AXIS	0	0	200000.00	200000.00	10000.00	50000.00
SECOND AXIS	0	1	150000.00	150000.00	10000.00	20000.00

EXIST CARD 1

When you press POWER to turn on, it will initialize all of the PCI-8134 cards, and makes all of the axis return to origin. When these steps complete, the READY light will turn on.

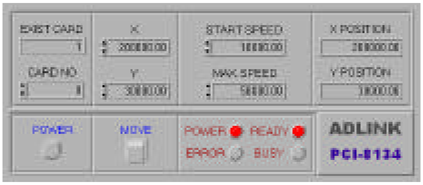
EXIST CARD shows how many cards that are installed correctly in your machine. Assign value for CARD NO, AXIS NO, and set value for POSITION, START SPEED, and MAXIMUN SPEED items of two axes. You can press MOVE to make them move. The position of the axis will be shown when it moves. You can find that they will start motion at the same time, and arrive at their destination by their selves.

You can press POWER to turn off. If you want to close this VI, you must to turn off it before you close.

3.1.4 8134Ex4.vi

This sample displays how to control the interpolated motion.

Click ⇨ bottom to run this VI, the control panel will be shown as below:



EXIST CARD 1
CARD NO 0
X 200000.00
Y 30000.00
START SPEED 10000.00
MAX SPEED 50000.00
X POSITION 200000.00
Y POSITION 30000.00

When you press POWER to turn on, it will initialize all of the PCI-8134 cards, and makes all of the axis return to origin. When these steps complete, the READY light will turn on.

EXIST CARD shows how many cards that are installed correctly in your machine. Assign value for CARD NO, and set value for X, Y position, START SPEED, and MAXIMUN SPEED items. You can press MOVE to make it move. The position of the axis will be shown when it moves. You can find that they will start motion at the same time, and arrive at their destination at the same time too.

You can press POWER to turn off. If you want to close this VI, you must to turn off it before you close.

3.2 The Detail of These VIs

3.2.1 Initial.vi

This VI divide

1. Call *8134_Initial* VI to initial PCI-8134 card, and it will return the amount of the card.
2. If the amount of the card equal to zero, it means that there is not any PCI-8134 card in your machine, or your card may be damaged, and it will return an error code.
3. Calculate the amount of the axis, use *For Loop* to set the status of every axis. This VI uses nine Vis to set, they are *set_cnt_src, set_pls_iptmode, set_pls_outmode, set_home_config, set_inp_logic, set_alm_logic, set_sd_logic, set_move_ratio, set_manu_iptmode*.

These samples of the PCI-8134 LabVIEW Vis are designed for all user, and they can run normally without any device. If you want to use *Initial.vi* to your own VI, the argument of above Vis in *Initial.vi* will be modified to suit your condition.

3.2.2 Waiting Time

In these sample Vis, if they want to wait some times, they don’ t use *Wait* function, and they use Tick Count function and *While Loop* to wait. Use *Wait* function will let your VI stop for some times and can not do any thing. So in these Vis use Tick Count function and *While Loop* to wait, and read POWER status, if POWER be turned off, the VI will be stopped immediately.

3.2.3 Home Return

In these sample Vis use *v_stop* VI to stop home return motion because these Vis are ran without any device and home return motion can’ t stop by origin switch in device. If you run your VI in an actual device, you can’ t need *v_stop* VI to stop home return motion.

In these sample Vis also wait ten seconds to wait home return motion of 48 axes complete. You can use *motion_done* VI to read the status of all axes of your device.

3.2.4 The Value of Start Velocity and Maximum Velocity

Although it will cause an error only when the maximum velocity is equal to zero or the start velocity and maximum velocity aren' t all positive or negative, you should determine that the maximum velocity must lager than zero, and the start velocity must not less then zero.

3.2.5 Close.vi

When you call *8134_Initial* VI to initial PCI-8134 card, you must call *8134_Close* VI to close it, otherwise the resource will be occupied and your operating system will be in error status.

LabVIEW can not close PCI-8134 automatically when you close the VI direct. So you should provide this capability in your VI and let user can close PCI-8134 before close VI. In previous two sample VIs, the PCI-8134 card can be initialized when you press POWER to turn on, and be closed when you press POWER to turn off, then user can close PCI-8134 normally before close VI.