EPCIO Series
Device Driver Library Integrated Testing Environment User Manual for WINDOWS

Version: V.2.00

Date: 2008.07

http://www.epcio.com.tw

Table of Contents

Chapter 1 Overview

This manual is the EPCIO testing program user manual. By reading this manual, one will know how to use the EPCIO testing program and set the parameters. The testing program can be used to test control cards which are designed and developed with EPCIO ASIC and which use the ISA-Bus/PCI-Bus interface, including the motion and input/output modules of EPCIO-6000/6005 and EPCIO-4000/4005 (PCI-Bus).

The testing program can be executed in the Win98/WinNT/Win2000/WinXP operating systems. To run the testing program, please load the installation CD-ROM and execute Driver Test Tool.exe (for PCI-Bus) in the directory of the CD-ROM. Then you will see the user interface of the testing program.

Chapter 2 How to Use the User Interface

2.1 Overview

The screen image of the user interface is shown in Figure 2-1.1 and described as follows:

(1) Main function selection area
(2) Setting selection area
(3) Status display area
(4) Version
(5) System clock
(6) Message field
(7) Control card selection area

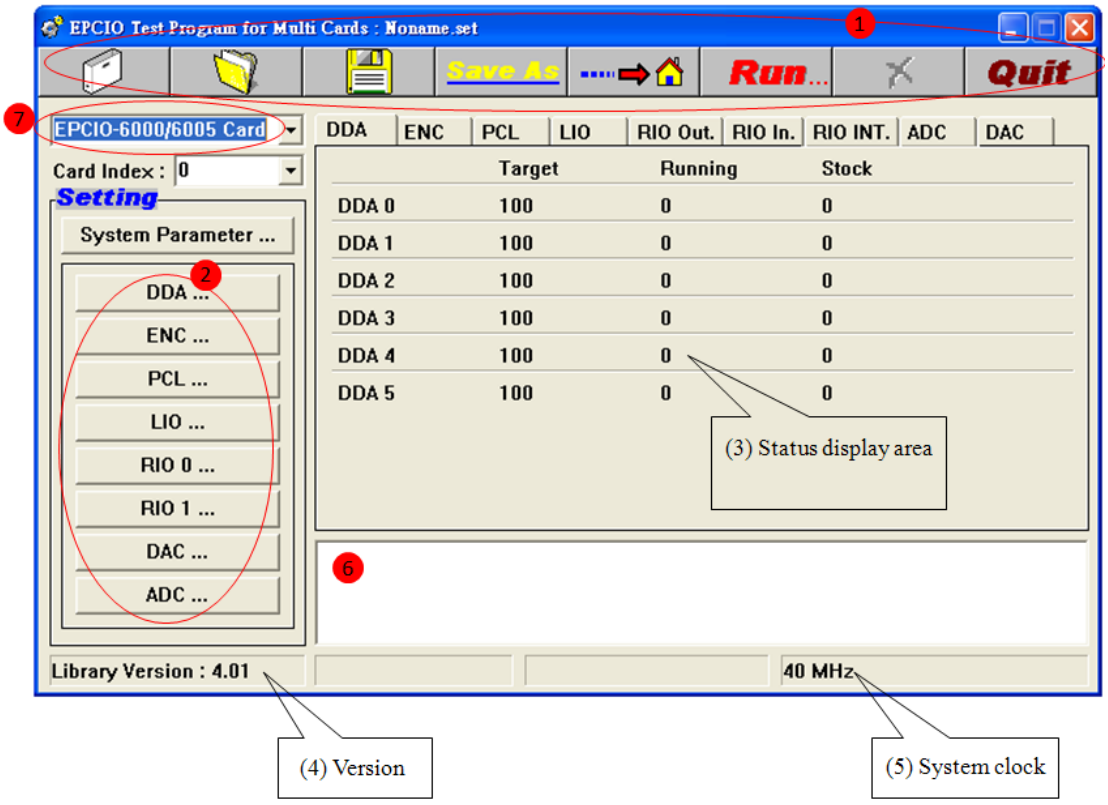

Figure 2-1.1

2.2 Description

(1) Main function selection area

This area includes such main function options as New (), Open (), Save (), Save as (), Go Home (), Run (), Stop (), and Exit (). For a detailed description of parameter setting, please refer to Chapter 4.

(2) Setting selection area

This area includes such function options as System Parameter, DDA, ENC, PCL, LIO, RIO 0, RIO 1, DAC, and ADC. For a detailed description of parameter settings, please refer to Chapter 4.

(3) Status display area

(a) DDA display area (as shown in Figure 2-2.1)

Target: total number of commands

$$Target = Times \times Repeat$$

Running: number of commands actually executed

Stock: number of commands in FIFO

See also Section 4.3 DDA Main Function Options.

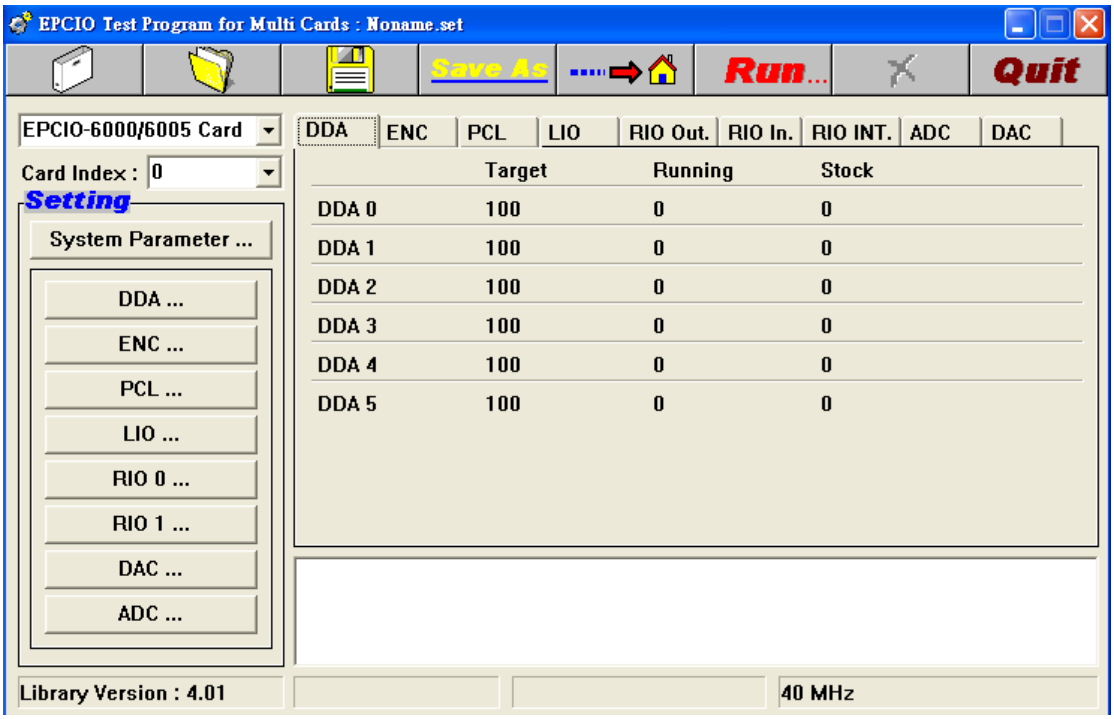

Figure 2-1.1

(b) ENCODER display area (as shown in Figure 2-2.2)

Counter: value read back from encoder counter

Latch: number of times of latch interrupt

Comp.: number of times of comparator interrupt

See also Section 4.4 Encoder Main Function Options.

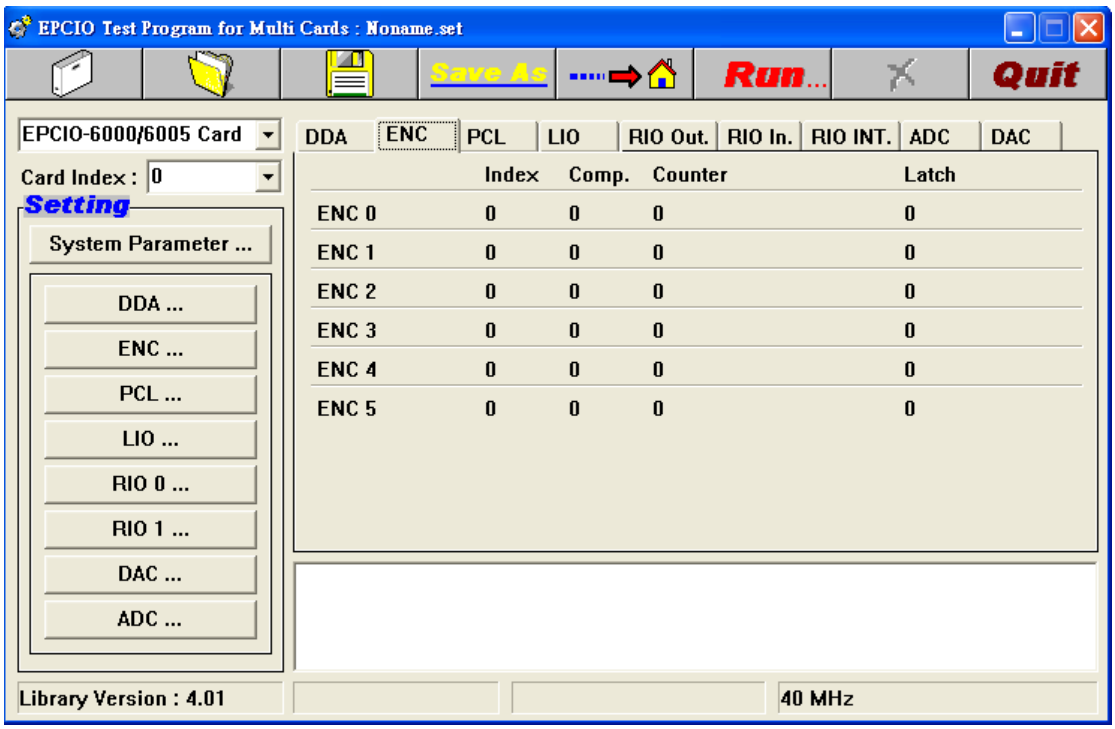

Figure 2-2.2

(c) PCL display area (as shown in Figure 2-2.3)

Error: error counter's value

Interrupt count: number of times of overflow interrupt

See also Section 4.5 PCL Main Function Options.

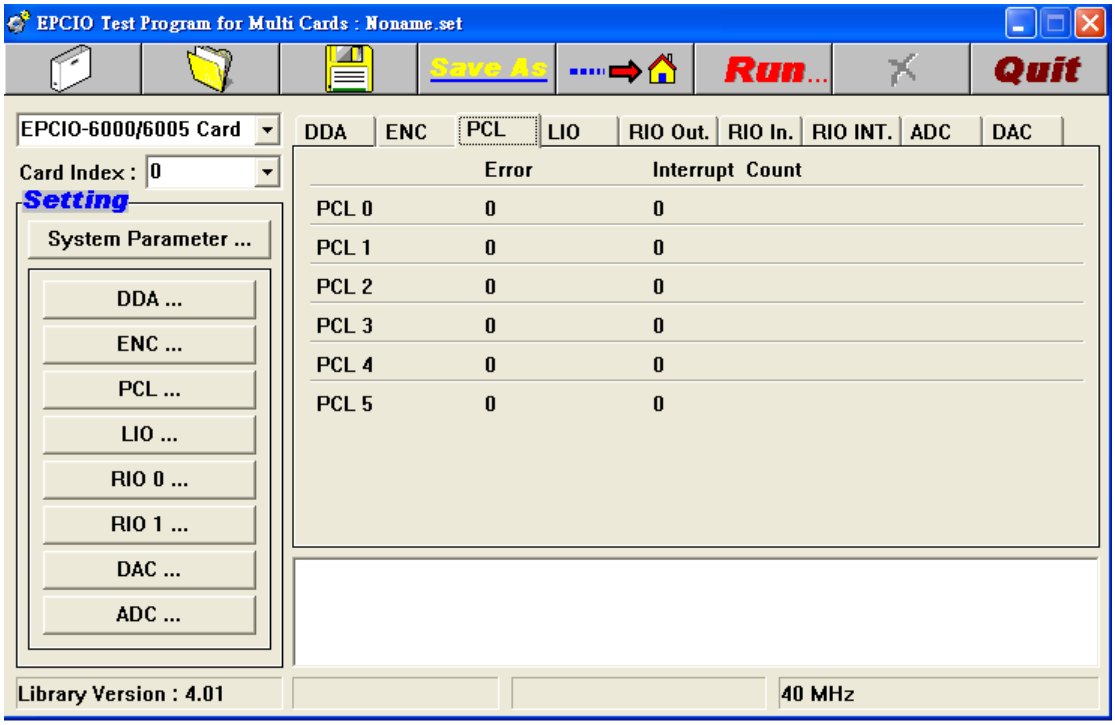

Figure 2-2.3

(d) LIO display area (as shown in Figure 2-2.4)

①Interrupt Counter

LDI 0 ~ 7: interrupt counters of the first 8 LIO connections.

DFI 0 ~ 6: interrupt counters of the first 7 double-function connections.

See also Section 4.6 LIO Main Function Options.

②Refresh Target

Setting for refreshing watchdog timer.

See also Section 4.6 LIO Main Function Options.

③Refresh Count

Number of times of refresh.

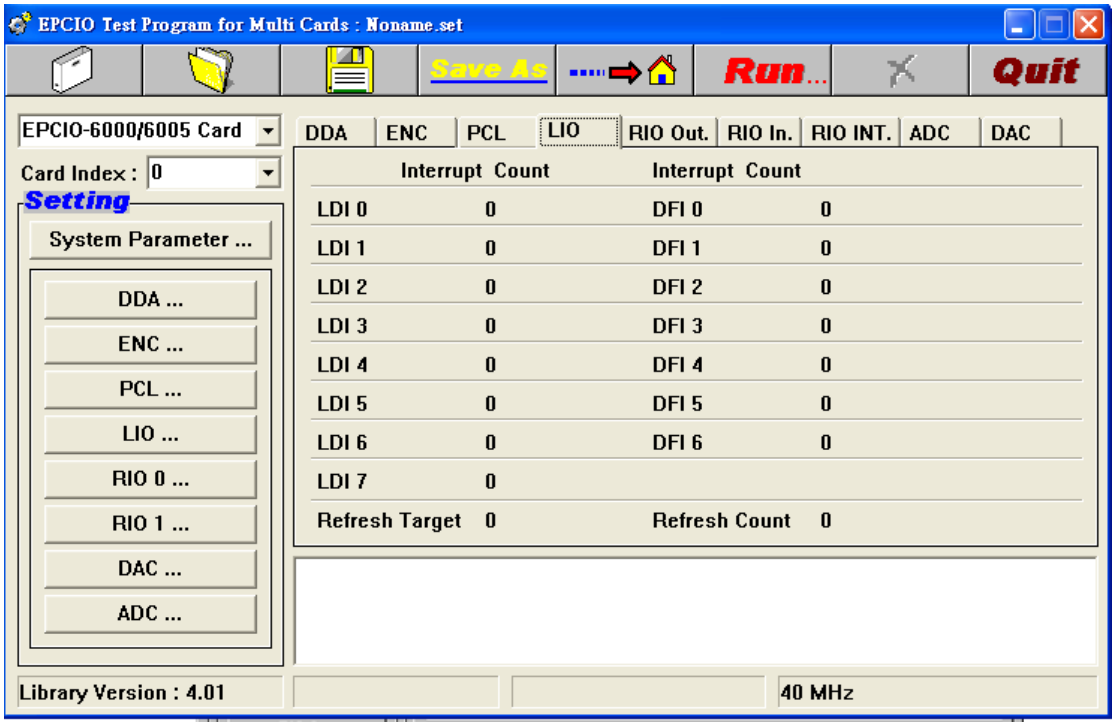

Figure 2-2.4

(e) RIO Out. display area (as shown in Figure 2-2.5)

This display area displays a total of 2 sets, and each set includes 3 slaves (RIO0S0, RIO0S1, RIO0S2, RIO1S0, RIO1S1, RIO1S2). Each slave is configured for 64-bit output.

Port 0 ~ Port 3 are each expressed with 16 bits.

See also Section 4.7 RIO 0 Main Function Options and Section 4.8 RIO 1 Main Function Options.

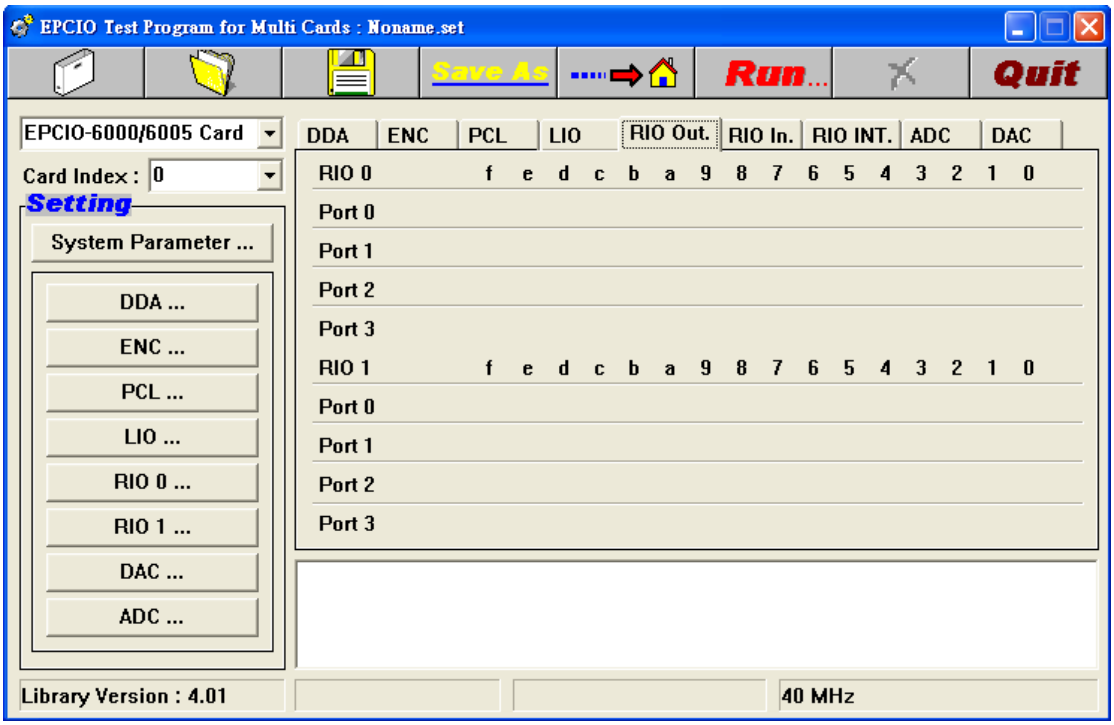

Figure 2-2.5

(f) RIO In. display area (as shown in Figure 2-2.6)

This display area displays a total of 2 sets, and each set includes 3 slaves (RIO0S0, RIO0S1, RIO0S2, RIO1S0, RIO1S1, RIO1S2). Each slave is configured for 64-bit output.

Port 0 ~ Port 3 are each expressed with 16 bits.

See also Section 4.7 RIO 0 Main Function Options and Section 4.8 RIO 1 Main Function Options.

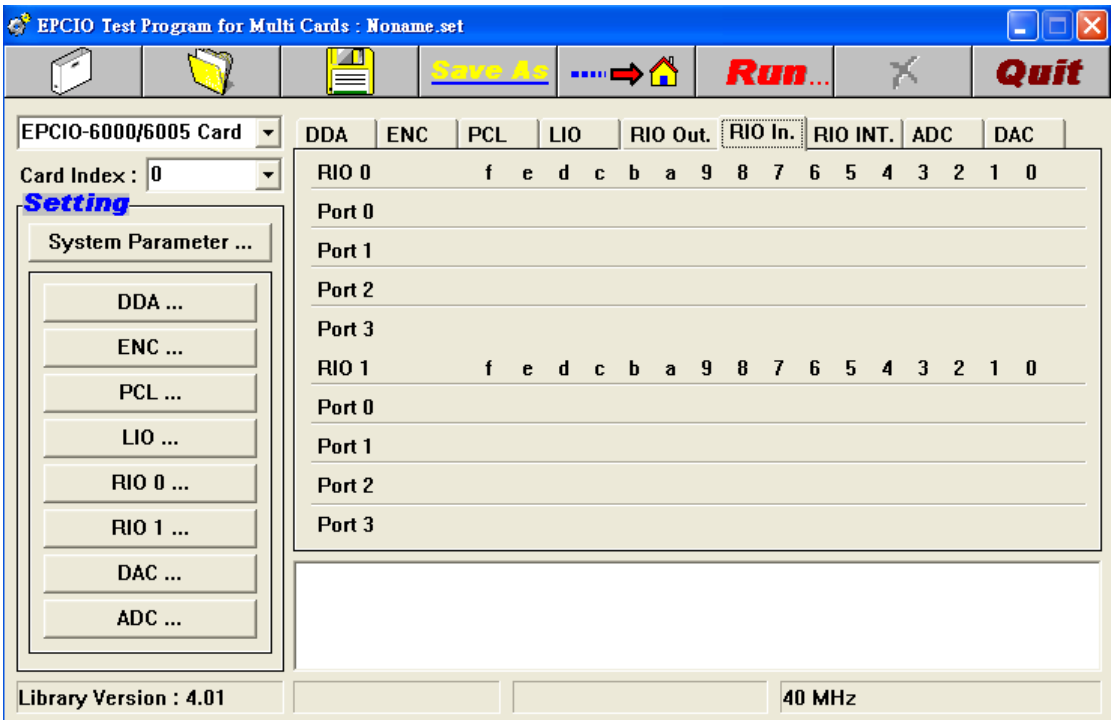

Figure 2-2.6

(g) RIO INT. display area (as shown in Figure 2-2.7)

I0 ~ I3: number of times of interrupt generated by each of the first 4 inputs of each slave.

Fail: number of times of transmission error interrupt of each set.

Status: transmission status. 1 stands for normal transmission, 0 stands for abnormal transmission, and Stop stands for transmission stopped.

See also Section 4.7 RIO 0 Main Function Options and Section 4.8 RIO 1 Main Function Options.

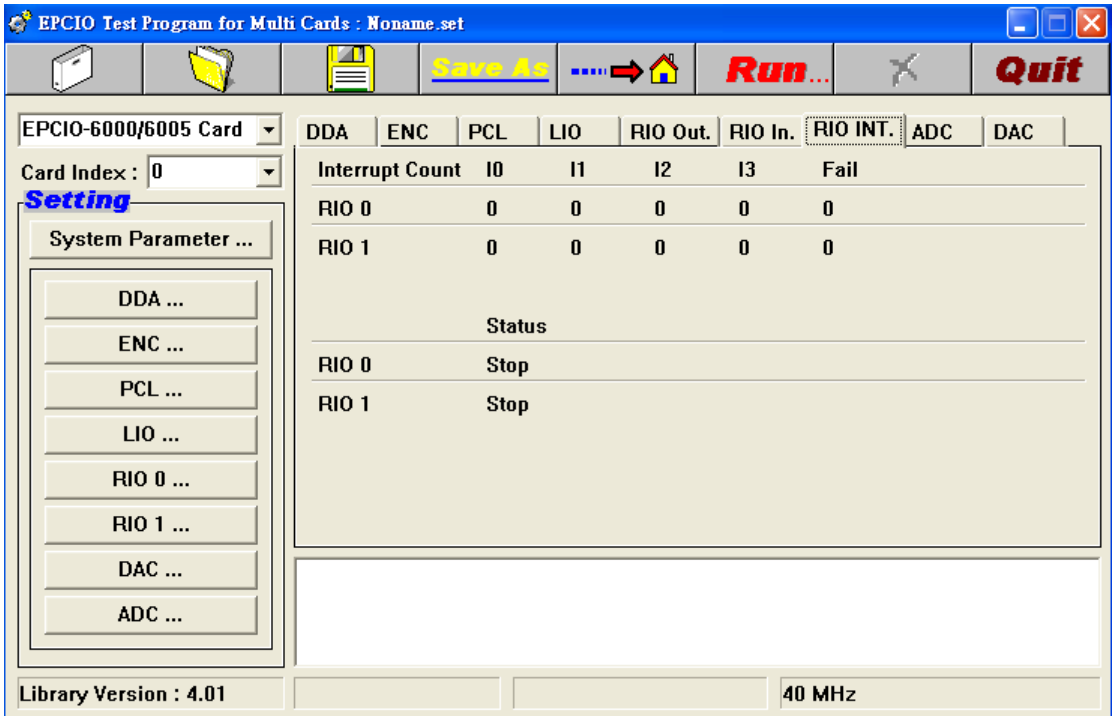

Figure 2-2.7

(h) ADC display area (as shown in Figure 2-2.8)

Input: input value of each ADC channel

Comparator: number of times of comparator interrupt of each channel

Converter One: number of times of interrupt of Convert Channel.

Converter Tag: number of times of interrupt of Tag Channel.

See also Section 4.10 ADC Main Function Options.

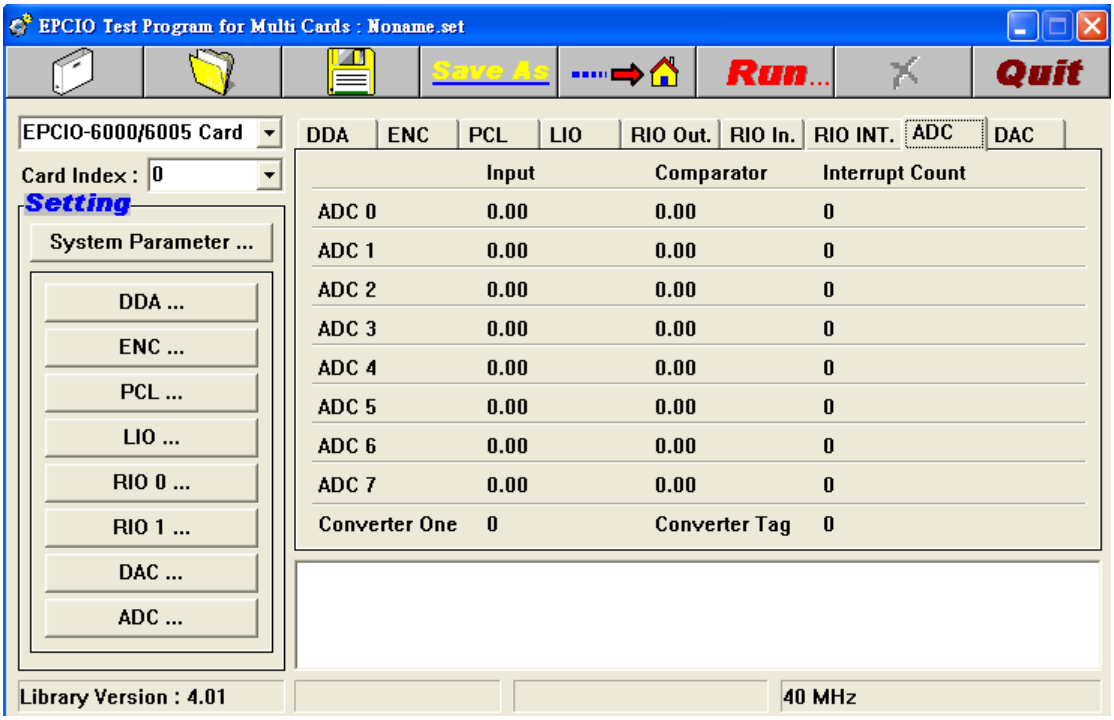

Figure 2-2.8

(i) DAC output value display area (as shown in Figure 2-2.9)
Software command output value of each DAC channel.
See also Section 4.9 DAC Main Function Options.

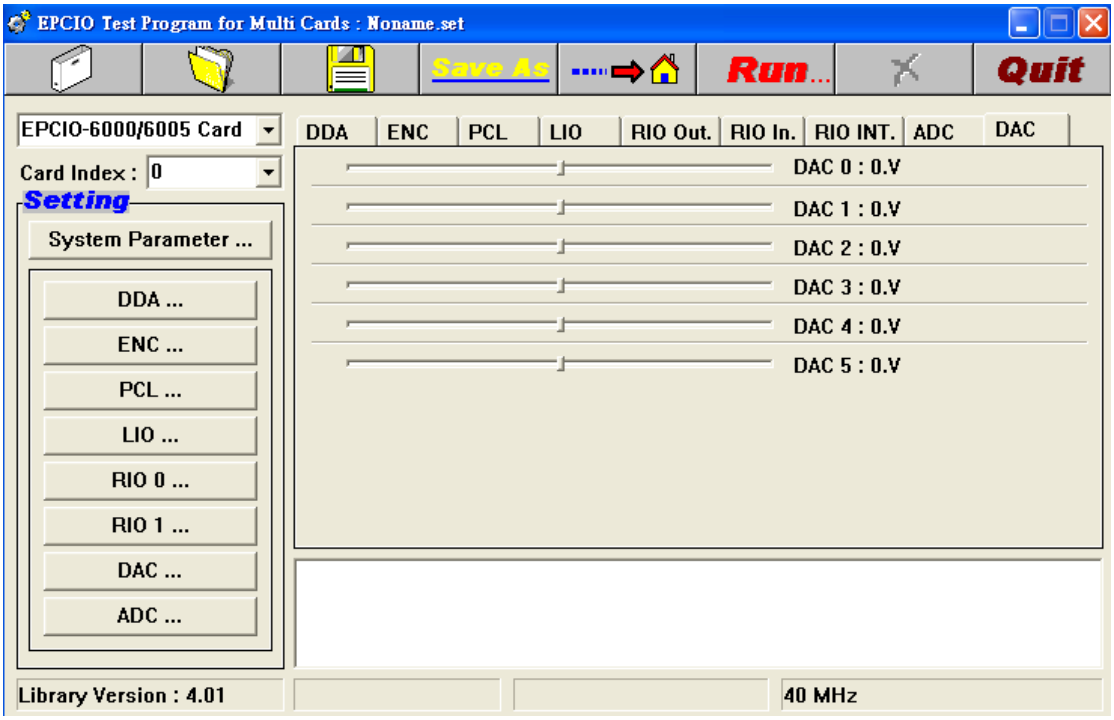

Figure 2-2.9

Chapter 3 EPCIO Rapid Testing Program

3.1 Basic Installation

A. Before installation, turn off the power, including the PC, motors, etc.

B. Make sure the PC PCI bus expansion slot in use is a complete 16-bit expansion slot.

C. Insert the EPCIO Series motion control card into the PCI bus, and fix the card in place.

D. Wire the peripheral circuits (e.g., motors, I/Os, ADCs, etc.) to the EPCIO Series motion control card, and fasten the connector to the card with screws. (For the wiring of the peripheral circuits, please refer to EPCIO-6000/6005, and EPCIO-4000/4005 User Manuals.)

E. Make sure the computer, the driven motors, and the I/O modules are properly grounded at the same reference potential to avoid system damage at start-up.

F. Turn on the computer.

G. Install and activate the testing program that comes with the card. In the Windows mode, execute Driver Test Tool.exe for PCI-Bus in the directory installed.

H. Check if EPCIOTest4PCI.exe can be found in the directory where the testing program is.

3.2 6-Axis Synchronous Pulse Output Control Test

3.2.1 Hardware wiring (refer to Section 3.1 Basic Installation and the following description). Taking EPCIO-6000/6005 for example, the following diagram (Figure 3-2.1) shows how EPCIO-6000/6005 is connected to a pulse-command servo motor/stepper motor system.

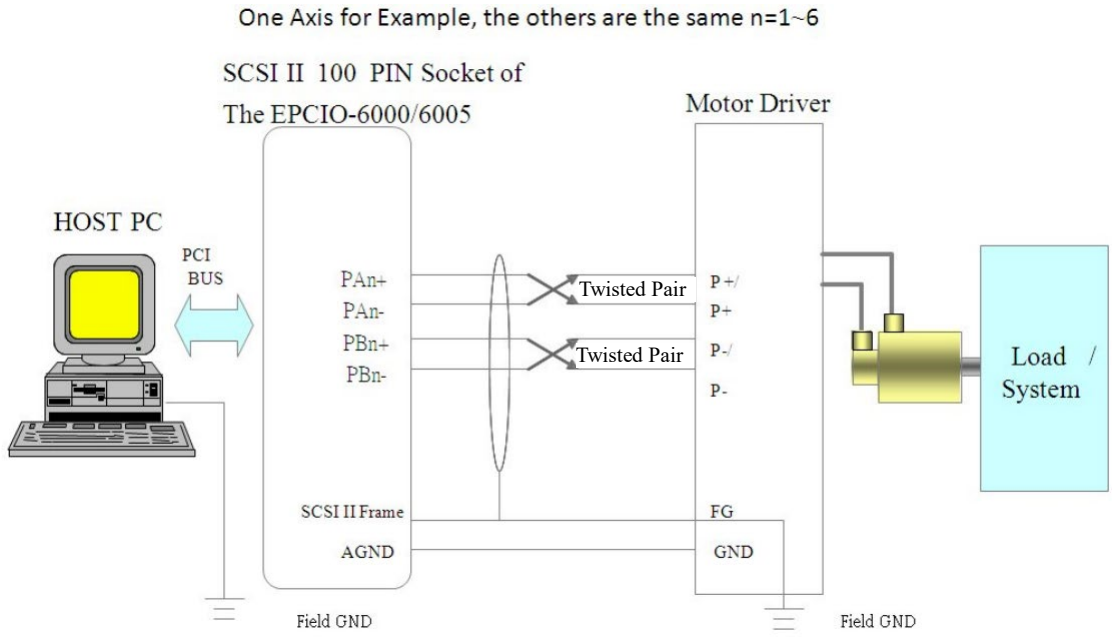

Figure 3-2.1

- PAn+, PAn-, PBn+ and PBn- represent the pulse output signals. They should be connected to P+, P+/, P-, P-/ of the nth MOTOR DRIVER as shown in the diagram. (Please refer to the manual of motor driver.)
- It is recommended that the foregoing four wires are twisted-pair cables to reduce common-mode noise. In addition, as illustrated in the diagram, the cables with a shield twisted-pair isolate signal transmission from external electromagnetic interference.
- Connect one side of each shielded twisted-pair cable to the frame of the SCSI II 100-pin connector of EPCIO-6000/6005 and the other side to the motor driver's ground. Make sure both the PC and the servo driver are grounded. (Note: The frame of the SCSI II 100-pin connector is connected to the PC chassis, and generally the PC chassis is connected to ground.)
- Important---A ground wire must be connected between GND of the servo driver and AGND of EPCIO-6000/6005. (This is very important because failure to do so may result in serious damage.)

3.2.2 Testing

A. Start the computer and run the testing program.

B. Set DDA pulse width

➔Make sure of the minimum pulse widths required by the pulse-command motor drivers, and assign them to the motion control card. (Select a proper value in each “Pulse Width” field in the DDA Setting window, and then click OK.)

C. Set DDA pulse format

➔Make sure of the pulse input formats of the pulse-command motor drivers, and assign them to the motion control card. (Select a proper pulse format in each “Format” field in the DDA Setting window, and then click OK.)

D. Set ENC input control

➔Make sure of the motor encoder feedback pulse formats of the pulse-command motor drivers, and assign them to the motion control card. (In the ENC Input Control Setting window, select a proper pulse format for each axis, and then click OK.) (If the A/B phase format is used, you can also input the multiplication rate.)

E. Make sure the Emergency Stop input of the motion control card is disabled (Refer to EPCIO-4000/4005, and EPCIO-6000/6005 User Manuals).

F. Turn on the motor drivers (SERVO ON).

G. Set pulse commands

➔Open the DDA Setting window, and set the pulse commands in the following manner of Figure 3-2.2:

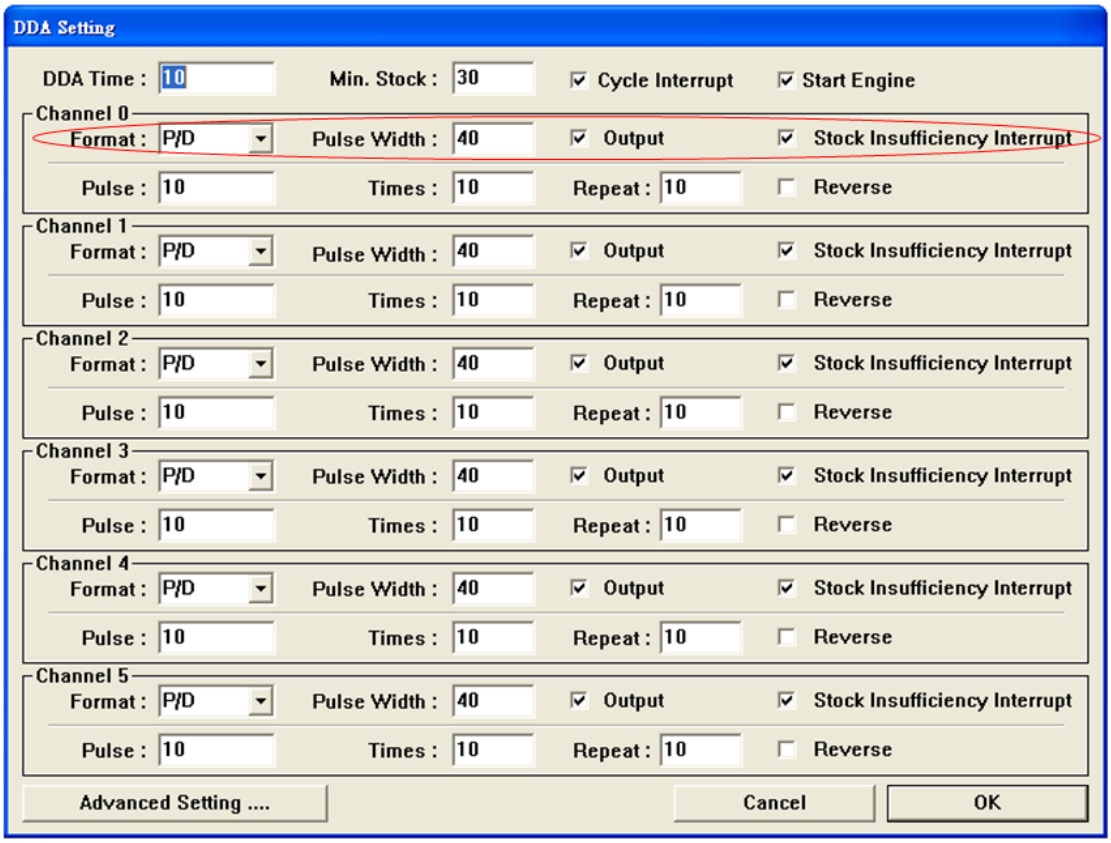

Figure 3-2.2

- While setting the values in the “Pulse” fields, bear in mind that the smaller the Pulse value, the lower the motor speed.
- By setting Pulse=X, Timer=Y, Repeat=Z, and leaving Backward unchecked, it means that the software will send out Z loops, that each loop includes Y DDA commands, and that each DDA command requires X pulses to be sent out. Therefore, when output from the DDA engine is completed, a total of X times Y times Z pulses will have been output.
- While the above process is in operation, the “Target” field in the DDA status display area (as shown in Figure 3-2.3) shows (X×Y) commands, and the “Counter” field shows the current counter value. The counter value will eventually be (X×Y×Z) pulses.
  In Figure 3-2.3, Pulse=10, Times=10, Repeat=10, and Backward is

unchecked, so during operation, the “Target” field will show (10×10 = 100) commands while the “Counter” field shows the current counter value.

The counter value will eventually be (10 × 10 × 10 = 1000) pulses.

- With Pulse=X, Timer=Y, Repeat=Z, and Backward checked, the software will send out Z loops, in which the first loop includes Y DDA commands each requiring X pulses (forward rotation) to be sent out, the second loop includes Y DDA commands each requiring -X pulses (with - representing backward rotation) to be sent out, the third loop includes Y DDA commands each requiring X pulses (forward rotation) to be sent out, the fourth loop includes Y DDA commands each requiring -X pulses (with - representing backward rotation) to be sent out, and so on. Thus, forward rotation alternates with backward rotation until the last loop is output. When output from the DDA engine is completed, the net number of output pulses will be X times Y times Z (if Z is an odd number) or 0 (if Z is an even number).
- While the above process is in operation, the “Target” field shows (X × Y) commands, and the “Counter” field shows the current counter value. The counter value will eventually be (X × Y × Z) pulses or 0 pulse.

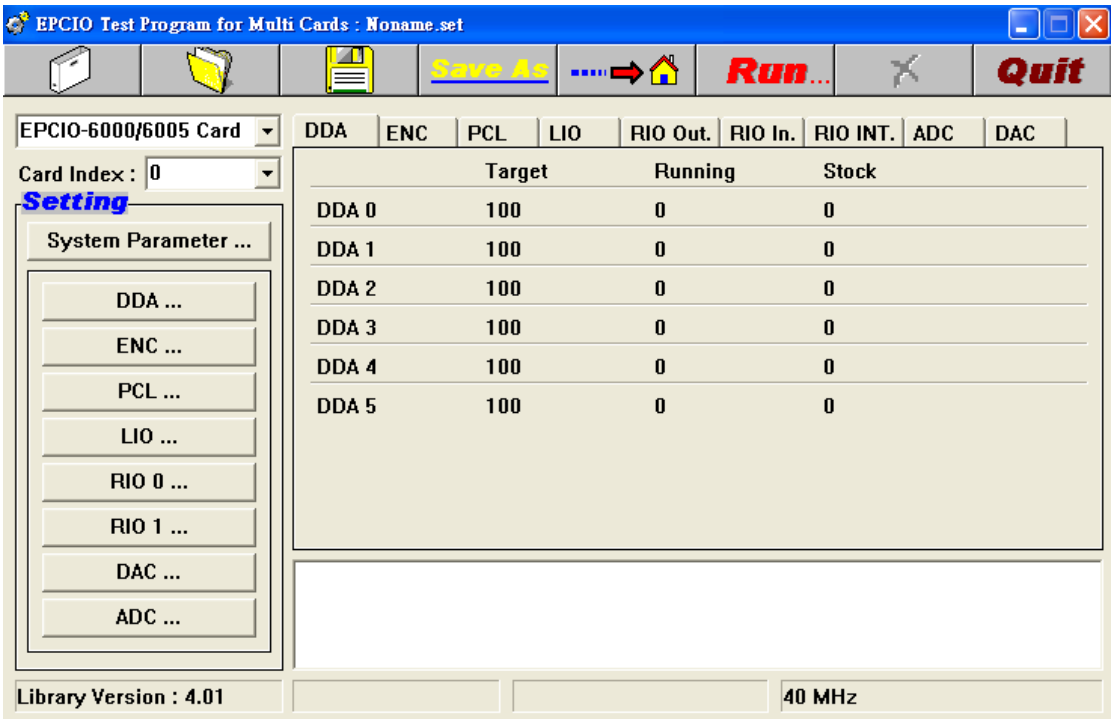

Figure 3-2.3

H. Click Run.

I. Click Stop (*X*) after the test is completed.

3.3 Rapid Test on Handwheel Input

3.3.1 Hardware wiring (refer to Section 3.1 Basic Installation and the following description). Taking EPCIO-6000/6005 for example, the following two diagrams show how EPCIO-6000/6005 is connected to a line driver output handwheel (Figure 3-3.1) and a voltage output handwheel (Figure 3-3.2).

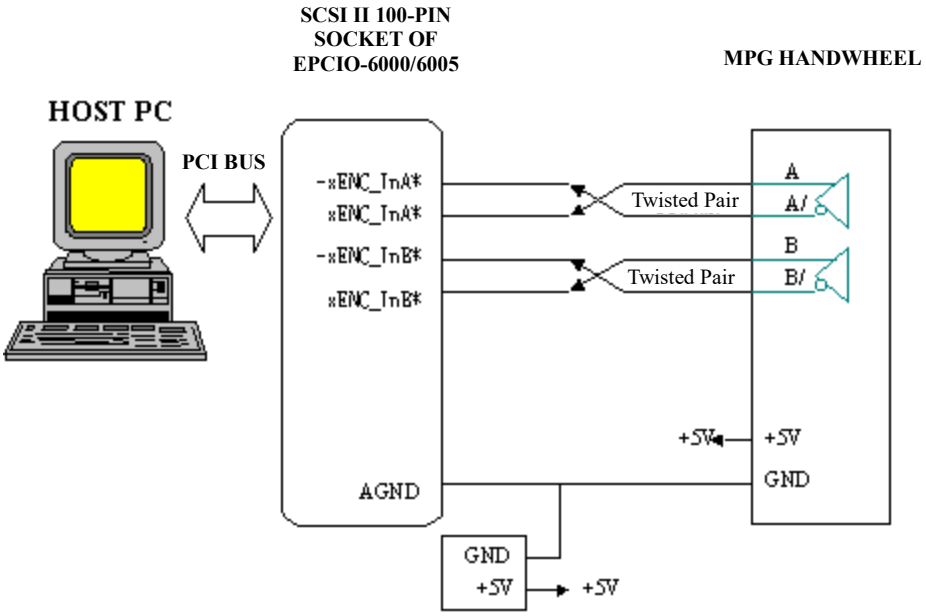

Figure 3-3.1

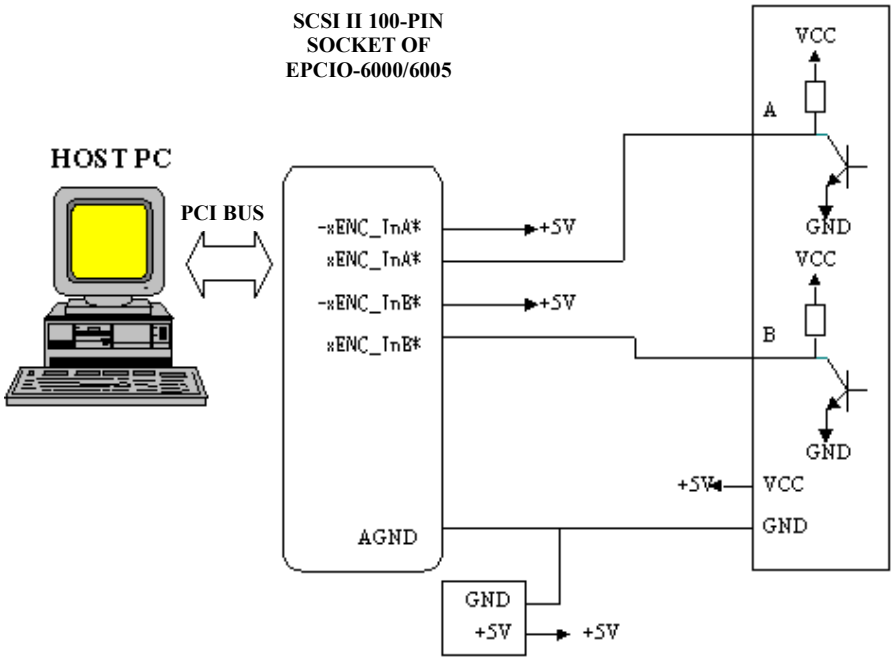

Figure 3-3.2

3.3.2 Testing

A. Start the computer and run the testing program.

B. Set ENC input control (as shown in Figure 3-3.3)

➔In the ENC Input Control Setting window, set the pulse format of the handwheel-input axis to A/B phase, and input the multiplication rate.

C. Click Run.

D. Turn the handwheel, and the handwheel input value will be shown in the "Counter" field. For example, turning the handwheel clockwise by one increment is equal to increasing/decreasing by 4 increments (with the multiplication rate being 4), 2 increments (with the multiplication rate being 2), 1 increment (with the multiplication rate being 1), or 0 increment (with the multiplication rate being 0).

E. Click Stop (X) after the test is completed.

F. Adjust the multiplication rate in the "Multiple" list, and run the test again by observing changes in the "Counter" field while adjusting the handwheel.

➔In the ENC Input Control Setting window, set the multiplication rate in the "Multiple" list to 1, 2, 4, or 0 (input inhibited).

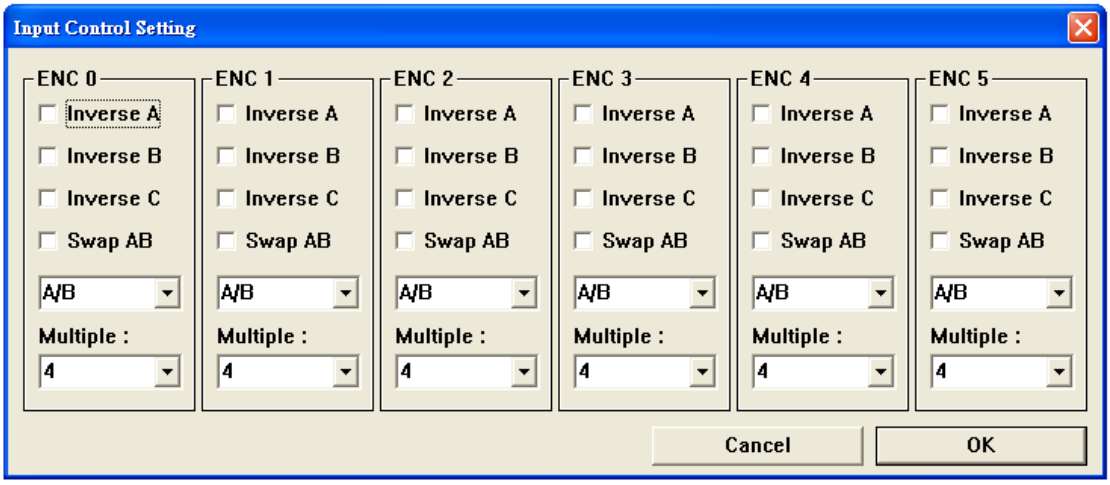

Figure 3-3.3

3.4 Rapid Test on 6-Axis Synchronous Closed Loop Control

3.4.1 Hardware wiring (refer to Section 3.1 Basic Installation and the following description). Taking EPCIO-6000 for example, the following diagram (Figure 3-4.1) shows how EPCIO-6000 is connected to a velocity-command servo motor system.

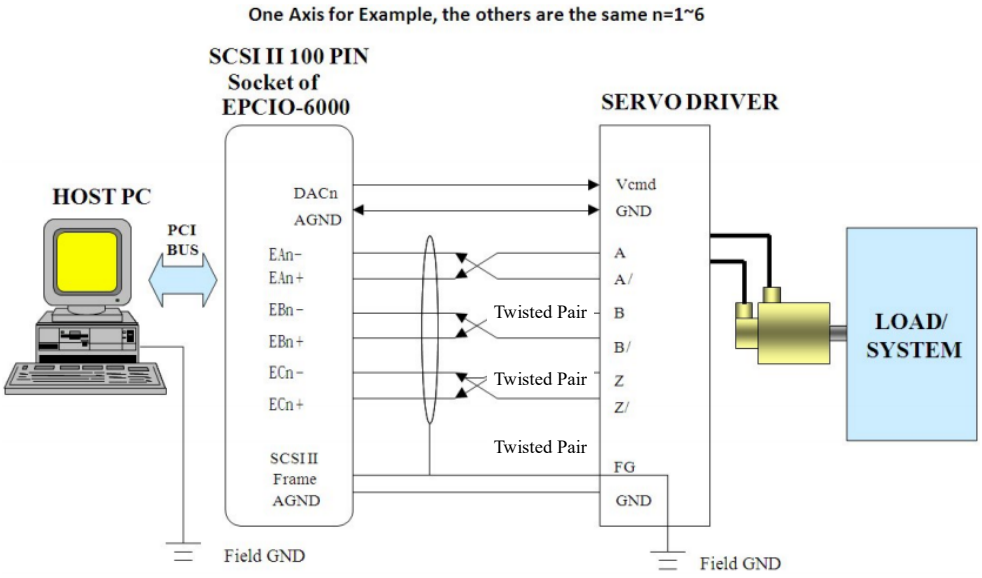

Figure 3-4.1

- DACn is the velocity command output of the nth closed loop control module. It must be connected to the V_{cmd} (Velocity Command) input of the nth servo driver. The signal ground (AGND) of DACn must be connected with the V_{cmd} ground (GND).
- The motor encoder signals (A/B/Z signals) of the servo driver must be connected back to EPCIO-6000 in differential form (as shown in Figure 3-4.1). It is recommended that the three sets of signal lines A and A/, B and B/, and Z and Z/ are cables with a shielded twisted-pair to reduce common-mode noise. In addition, as shown in Figure 3-4.1, the cables with a shielded twisted-pair isolate signal transmission from external electromagnetic interference.
- Connect one side of each shielded twisted-pair cable to the frame of the SCSI II 100-pin connector of EPCIO-601 and the other side to the ground of the servo driver. Make sure both the PC and the servo driver are grounded. (Note: The frame of the SCSI II 100-pin connector is connected to the PC chassis, and generally the PC chassis is connected to ground.)
- Important---A ground wire must be connected between the servo driver ground and AGND of EPCIO-6000. (This is very important because failure to do so may result in serious damage.)

3.4.2 Testing

A. Start the computer and run the testing program.

B. Make sure the motor drivers' command input format is velocity command (-10V ~ 10V).

C. Set ENC input control

➔Make sure of the motor encoder feedback formats of the pulse-command motor drivers, and assign them to the motion control card. (In the ENC Input Control Setting window, select a proper pulse format for each axis, and then click OK.) (If the A/B phase format is used, you can also input the multiplication rate.)

D. Make sure the Emergency Stop input of the motion control card is disabled (Refer to EPCIO-4000/4005, and EPCIO-6000/6005 User Manuals).

E. Turn on the motor drivers (SERVO ON).

F. Set pulse commands (as shown in Figure 3-4.2)

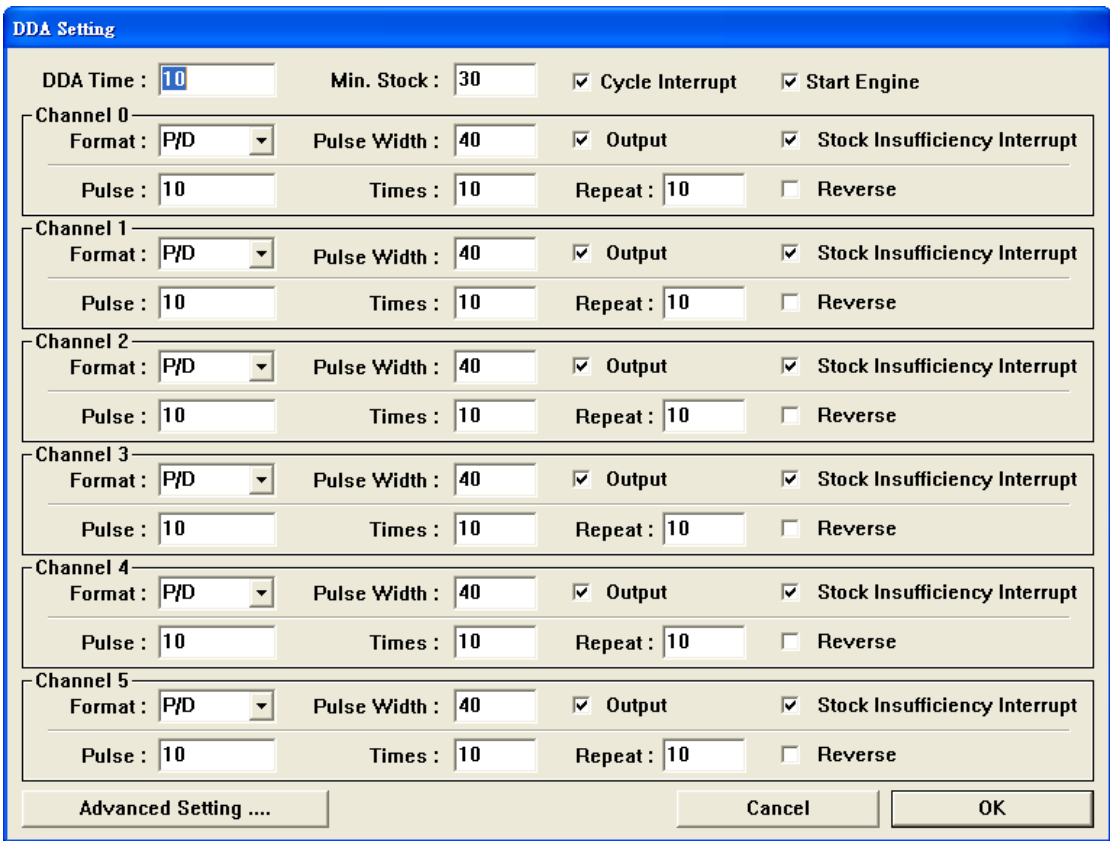

Figure 3-4.2

➔Open the DDA Setting window, and set the pulse commands in the following manner:

- While setting the values in the “Pulse” fields, bear in mind that the smaller the Pulse value, the lower the motor speed.
- By setting Pulse=X, Times=Y, Repeat=Z, and Back=NO, it means that the software will send out Z loops, that each loop includes Y DDA commands, and that each DDA command requires X pulses to be sent out. Therefore, when output from the DDA engine is completed, a total of X times Y times Z pulses will have been output to the closed-loop control module. The motor will eventually be driven to move, under closed loop control, by (X×Y×Z) pulses.
- While the above process is in operation, the “Target” field in the DDA status display area (as shown in Figure 3-4.3) shows (X×Y) commands, and the “Counter” field shows the current counter value. The counter value will eventually be (X×Y×Z) pulses.
- With Pulse=X, Times=Y, Repeat=Z, and Back=YES, the software will

send out Z loops, in which the first loop includes Y DDA commands each requiring X pulses (forward rotation) to be sent out, the second loop includes Y DDA commands each requiring -X pulses (with - representing backward rotation) to be sent out, the third loop includes Y DDA commands each requiring X pulses (forward rotation) to be sent out, the fourth loop includes Y DDA commands each requiring -X pulses (with - representing backward rotation) to be sent out, and so on. Thus, forward rotation alternates with backward rotation until the last loop is output. When output from the DDA engine to the closed loop control module is completed, the net number of output pulses (which corresponds to the net displacement of the motor) will be X times Y times Z (if Z is an odd number) or 0 (if Z is an even number).

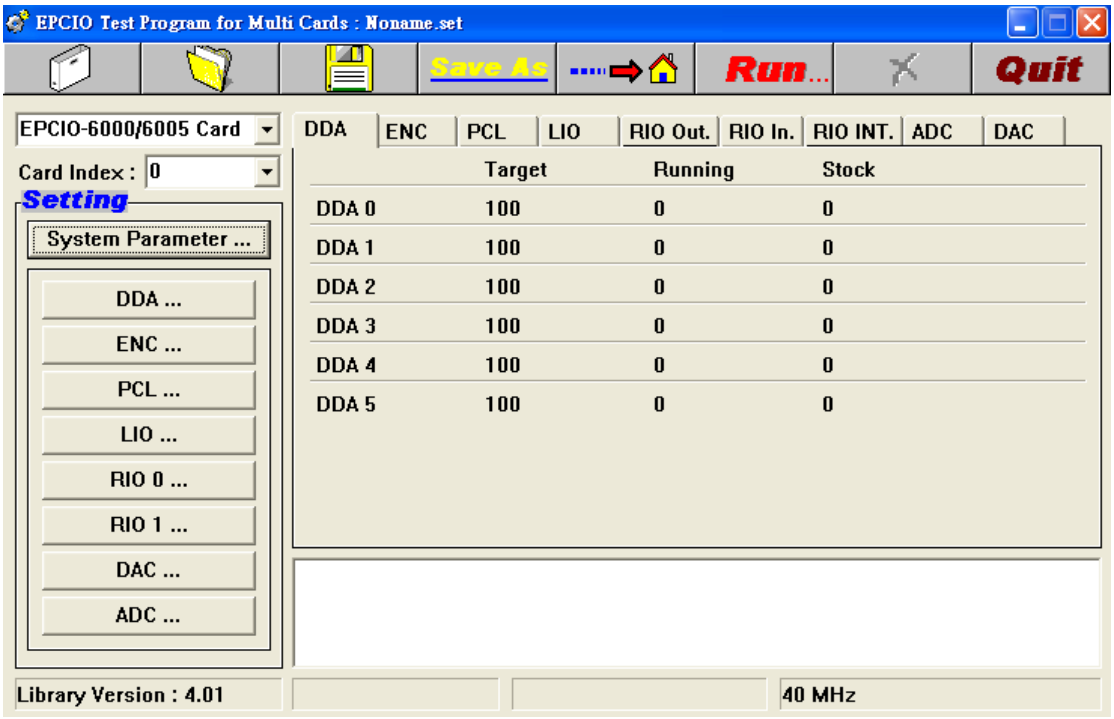

Figure 3-4.3

- While the above process is in operation, the “Target” field shows (X × Y) commands, and the “Counter” field shows the current counter value. The counter value will eventually be (X × Y × Z) pulses or 0 pulse.
- In Figure 3-4.2, Pulse=10, Times=10, Repeat=10, and Back=yes, so during operation, the “Target” field will show (10 × 10 = 100) commands while the “Counter” field shows the current counter value. The counter value will eventually be 0 pulse (for Repeat=10, an even number).

G. Click *Run*.

H. If, after execution of the preset PCL.601 file is completed (the motors have stopped), the final value of a certain axis is not 0, adjust the offset adjustment knob of that axis until the value in the corresponding “Counter” field becomes 0. Thus, the next run will bring the value to 0 ±1.

I. Click Stop (*X*) after the test is completed.

3.5 LIO Input and Output Test

3.5.1 Hardware wiring (refer to Section 3.1 Basic Installation and the following description). Taking EPCIO-6000/6005 for example, the following diagrams show the LIO input connections (Figure 3-5.1) and LIO output connections (Figure 3-5.2) of EPCIO-6000/6005.

◎ The following diagram shows the input of HOME1. The same wiring applies to the other local inputs, too.

◎ Types of input wiring: Source-input type and sink-input type.

◎ When the switch is closed, the photo coupler is activated, and EPCIO-6000/6005 identifies the input at this moment as 0.

◎ When the switch is open, the photo coupler is inactivated, and EPCIO-6000/6005 identifies the input at this moment as 1.

◎ The system must provide +24 V DC power

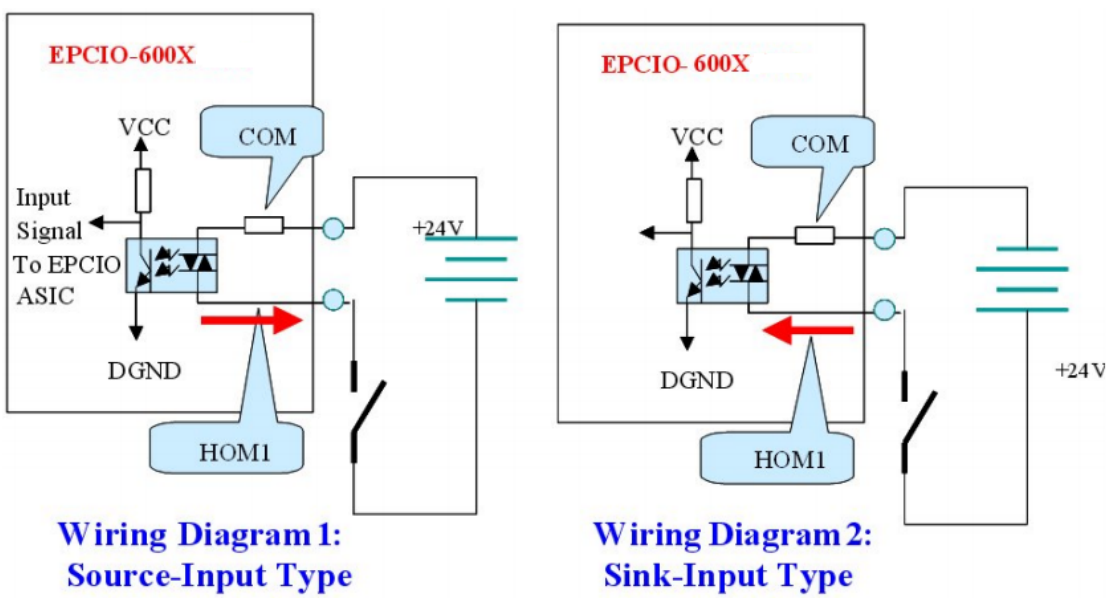

Figure 3-5.1

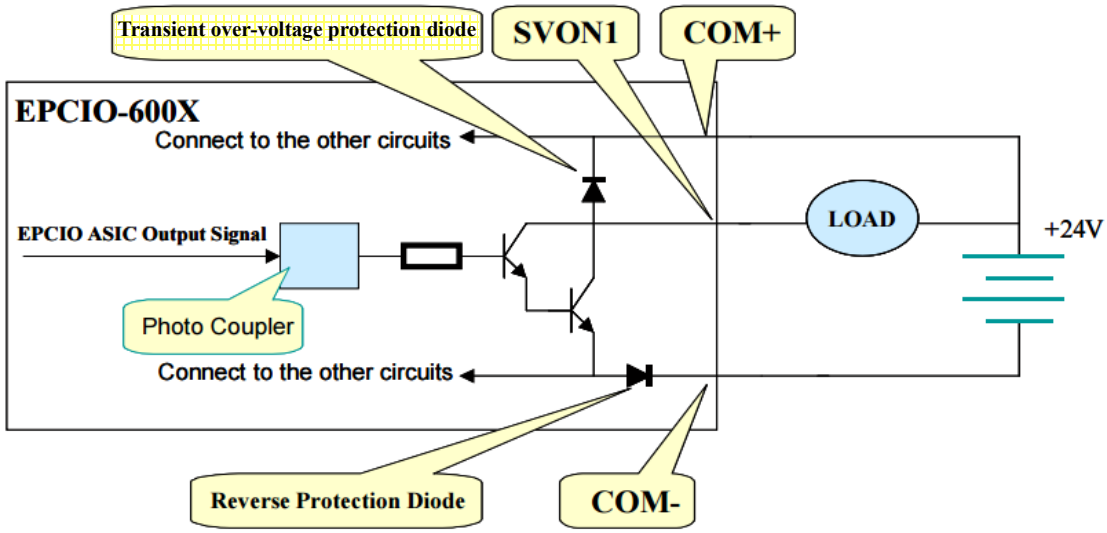

Figure 3.5-2

◎The above diagram shows the output of SVON1. The same wiring applies to the other local outputs, too.

◎When the output signal is 0, the transistor (Darlington stage) is conducted, and the load is power ON.

◎Maximum load of each connection is 60 mA. Overload will cause the transistor to burn out. It is prohibited to directly connect a 24V power supply to the output point in no-load conditions.

◎Since a relay itself as the load has a diode to protect against transient over-voltage, it does not need to connect with an external surge-absorbing diode.

3.5.2 Testing

A. Start the computer and run the testing program.

B. Set output value (as shown in Figure 3-5.3)

➔Make a selection in the LIO Output Value section, in which LDO 15-0 are set as inputs and therefore should not be selected for output value setting. Then fill the Bit 27-16 fields with a duodecimal value, whose least significant binary bit is the value of LDO16.

C. Click *Run*.

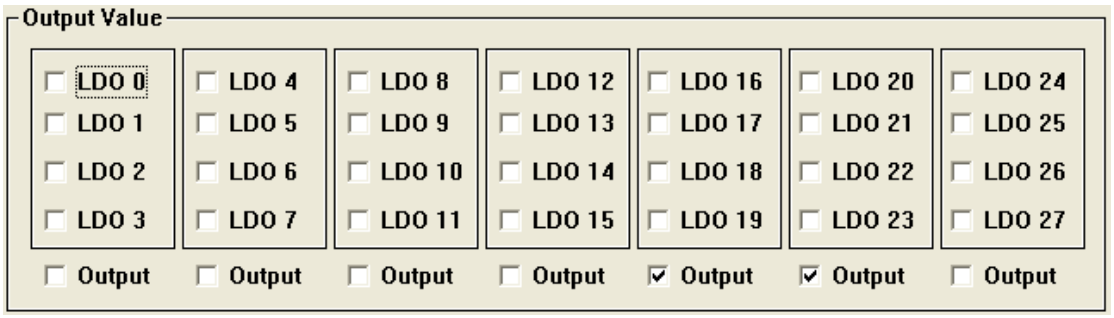

Figure 3.5-3

3.6 RIO Input and Output Test

3.6.1 Hardware wiring (refer to Section 3.1 Basic Installation and the following description). Taking EPCIO-6000/6005 for example, Figure 3-6.1 shows how EPCIO-6000/6005 is connected with the RIO sockets of two EDIO-S00X.

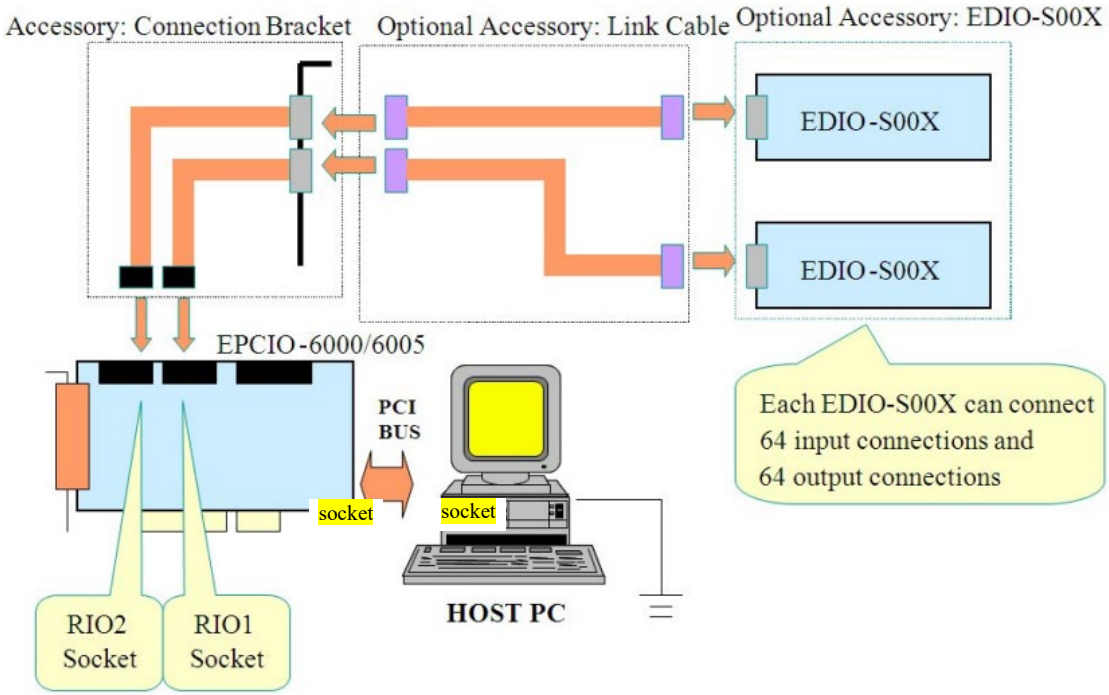

Figure 3-6.1

3.6.2 Testing

A. Start the computer and run the testing program.

B. Click *Run*.

C. Observe the “RIO In.” fields (Figure 3-6.2) and the “RIO Out.” fields (Figure 3-6.3). The values in the “RIO Out.” fields will be output to and displayed on the corresponding EDIO-S00X modules in a sequential and cyclic manner (i.e., each port in the “RIO Out.” display area will have a √ mark displayed as on the corresponding module), and each input from the EDIO-S00X modules will be displayed by a √ mark in the corresponding “RIO In.” field (for the corresponding relationships, please refer to EPCIO-**6000/6005** User Manuals).

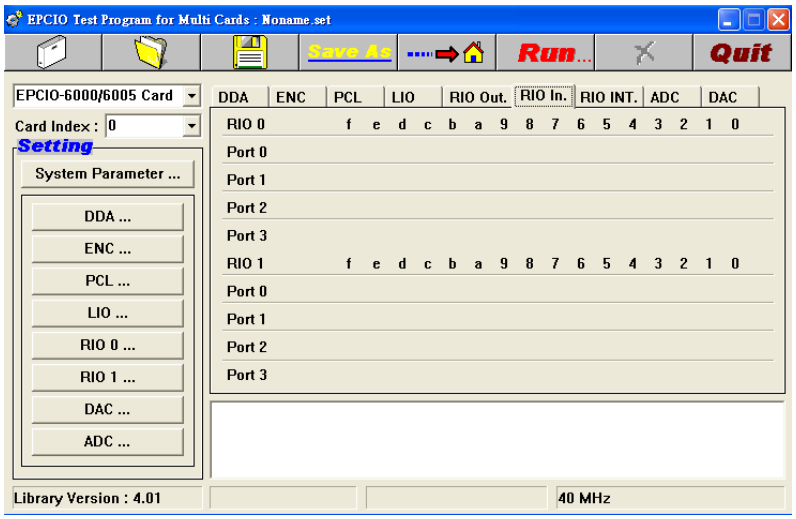

Figure 3-6.2

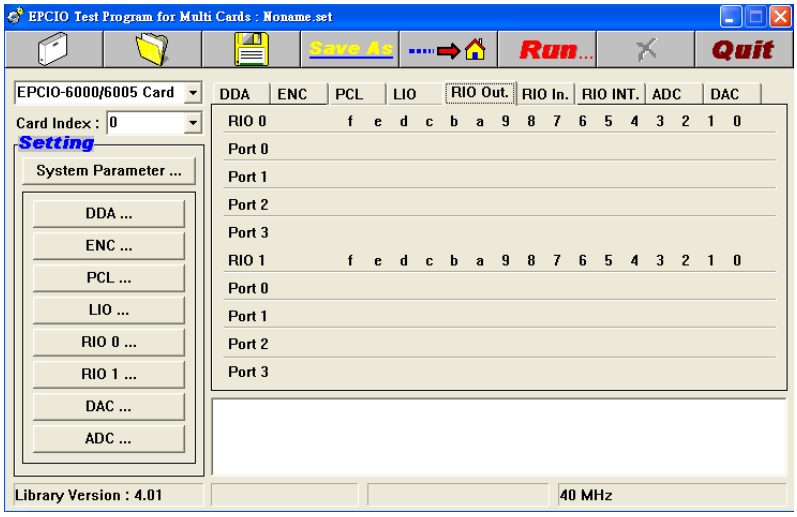

Figure 3-6.3

3.7 ADC Input Test

3.7.1 Hardware wiring (refer to Section 3.1 Basic Installation and the following description). Taking EPCIO-6000 for example, Figure 3-7.1 shows the ADC input connections of EPCIO-6000.

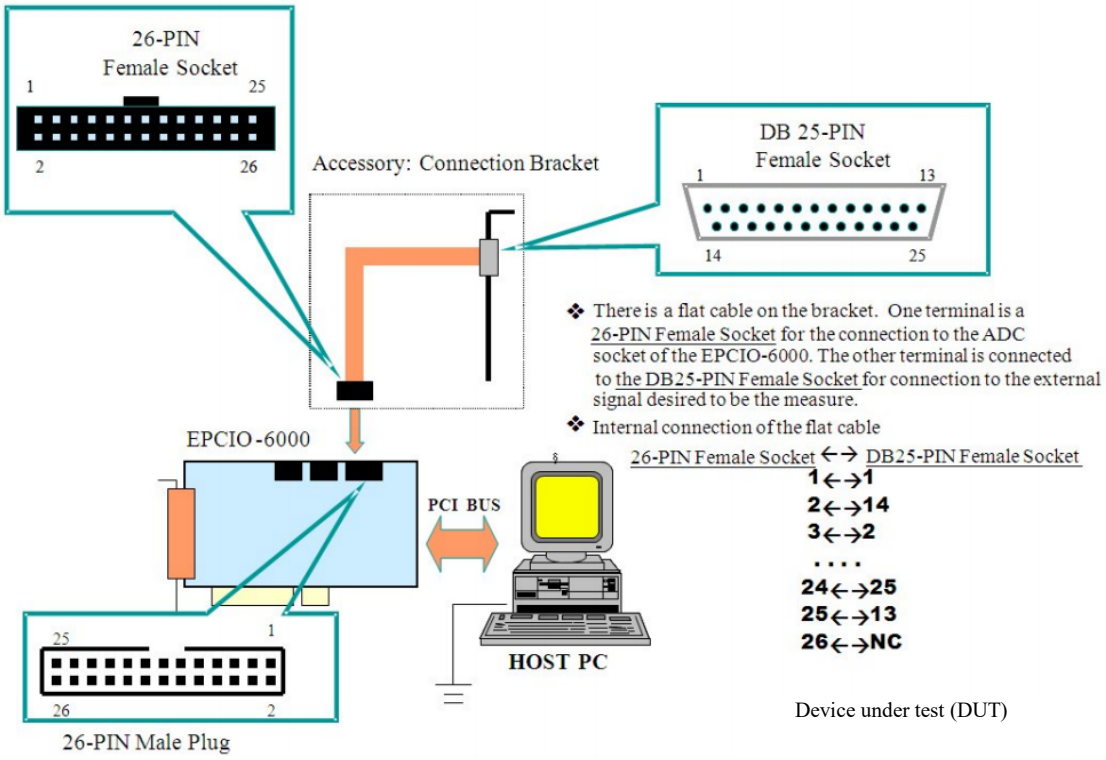

Figure 3-7.1

3.7.2 Testing

A. Start the computer and run the testing program.

B. If the range of input voltage is -10V ~ 10V, please set the EPCIO motion control card to the bipolar mode. If the range of input voltage is 0V ~ 20V, please set the EPCIO motion control card to the unipolar mode

C. Set ADC

Select the bi/unipolar mode in accordance with hardware (circuit board) setting.

➔Set bi/unipolar in the ADC Setting window (as shown in Figure 3-7.2).

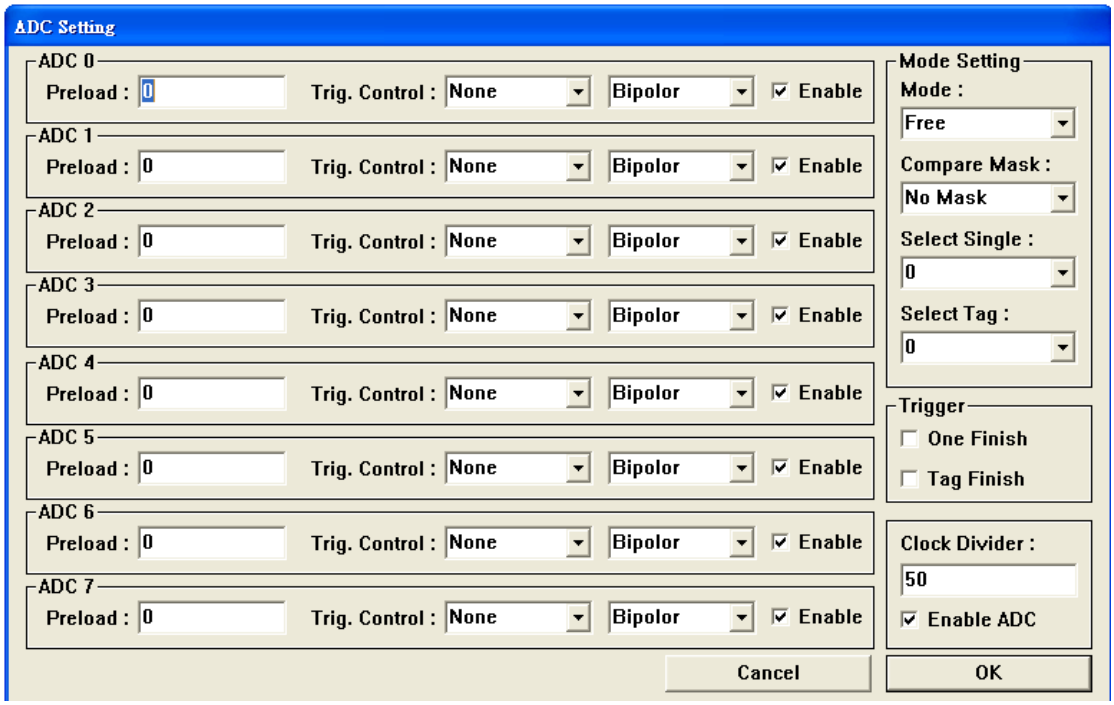

Figure 3-7.2

D. Click *Run*.

Observe the input voltage values in the "ADC" fields (as shown in Figure 3-7.3).

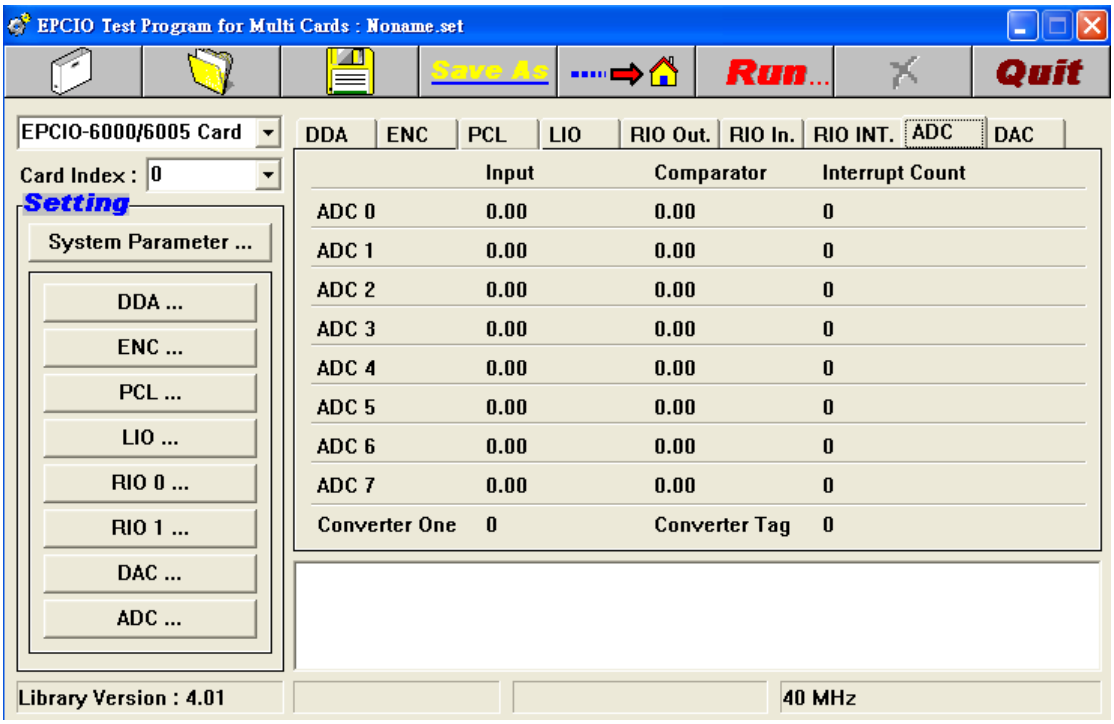

Figure 3-7.3

3.8 DAC Output Test

3.8.1 Hardware wiring (refer to Section 3.1 Basic Installation and the following description). Taking EPCIO-6000 for example, Figure 3-8.1 shows the ADC input connections of EPCIO-6000.

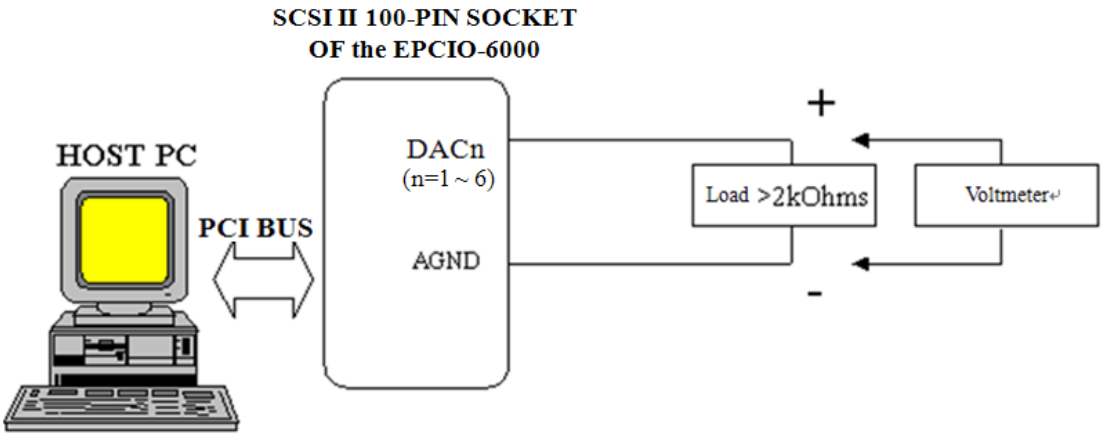

Figure 3-8.1

■ DACn is the nth voltage output and must be connected to a load greater than 2k ohms. The voltage output can be measured with a voltmeter connected across the load.

3.8.2 Testing

A. Start the computer and run the testing program.

B. Make sure the Emergency Stop input of the motion control card is disabled (refer to EPCIO-6000/6005 User Manuals).

C. Click *Run*. After the test is completed, measure the output voltage of each DAC channel.

D. Click Stop (*X*).

E. Adjust the offset adjustment knob of each axis of the motion control card until the output measurement of each axis is 0. The adjustment is made in the following manner:

F. Set DAC output value

➔In the DAC display area (as shown in Figure 3-8.2), adjust the value of each DAC channel to 0V.

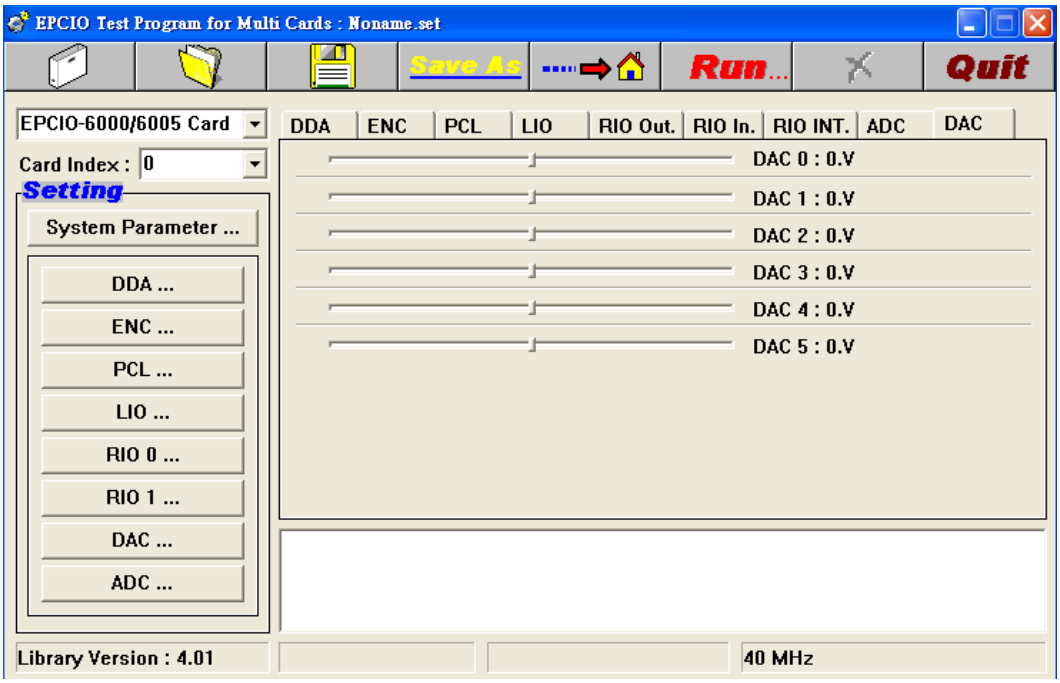

Figure 3-8.2

G. Click Run. Then adjust the DAC offset adjustment knob according to the voltage measurements until the measurements become 0.

H. Change the output value setting and output.

➔Set DAC output value. (In the DAC display area, adjust the output voltage of each DAC channel to 5V.)

Chapter 4 Parameter Settings and Function Descriptions

4.1 Main Function Options (refer to Figure 4-1.1)

(1) New (): open a new file.

(2) Open (): open an existing file.

(3) Save (): save the current parameter(s) into the opened configuration file.

(4) Save as (): save the current parameter(s) into a new file.

(5) Go Home (): return home.

(6) Run (): execute the EPCIO testing program.

(7) Stop (): stop the EPCIO testing program.

(8) Exit (): exit the EPCIO testing program.

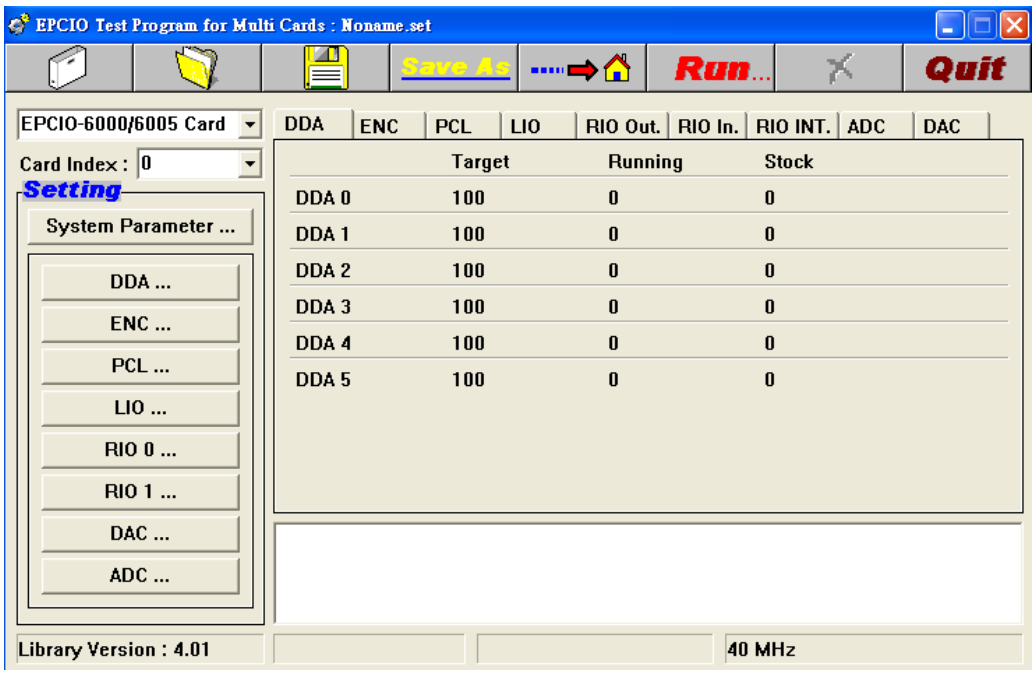

Figure 4-1.1

4.2 System Parameter Function Options

(1) Card index setting (as shown in Figure 4-1.2)

The motion control card index ranges from 0 to 3.

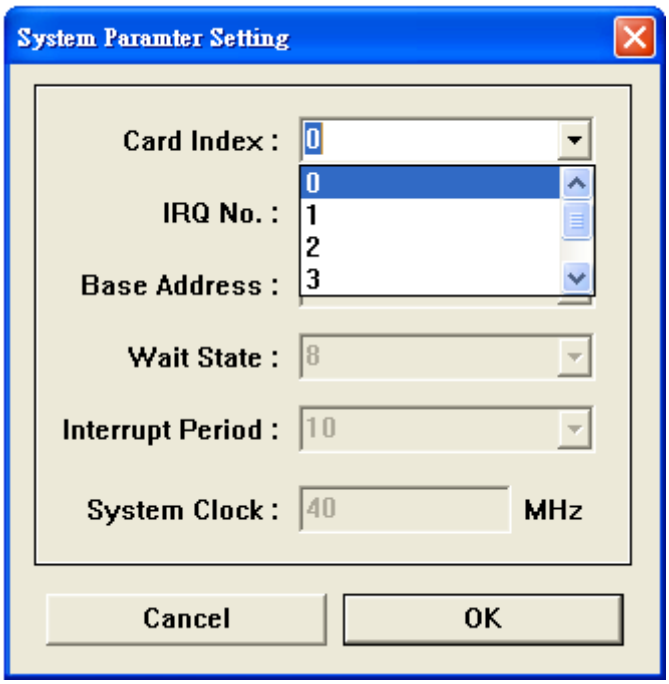

Figure 4-1.2

4.3 DDA Main Function Options

(1) Set DDA time (as shown in Figure 4-3.1)

This is to set the DDA time.

(2) Set minimum FIFO stock

This is to set the minimum number of commands stored in the DDA FIFO. If used in combination with the interrupt setting, an interrupt signal will be sent out when the number of commands stored in the DDA FIFO becomes smaller than the minimum stock.

(3) Set DDA start/stop (as shown in Figure 4-3.1)

This is to enable/disable pulse output from the DDA of each axis. To enable the output function of a certain DDA channel, the "Output" box of that channel (axis) must be checked. If the "Output" box of any axis has been checked, the Start Engine box must also be checked.

If the "Start Engine" box is not checked, DDA output from all the axes will be disabled, whether the "Output" box of any axis has been checked or not.

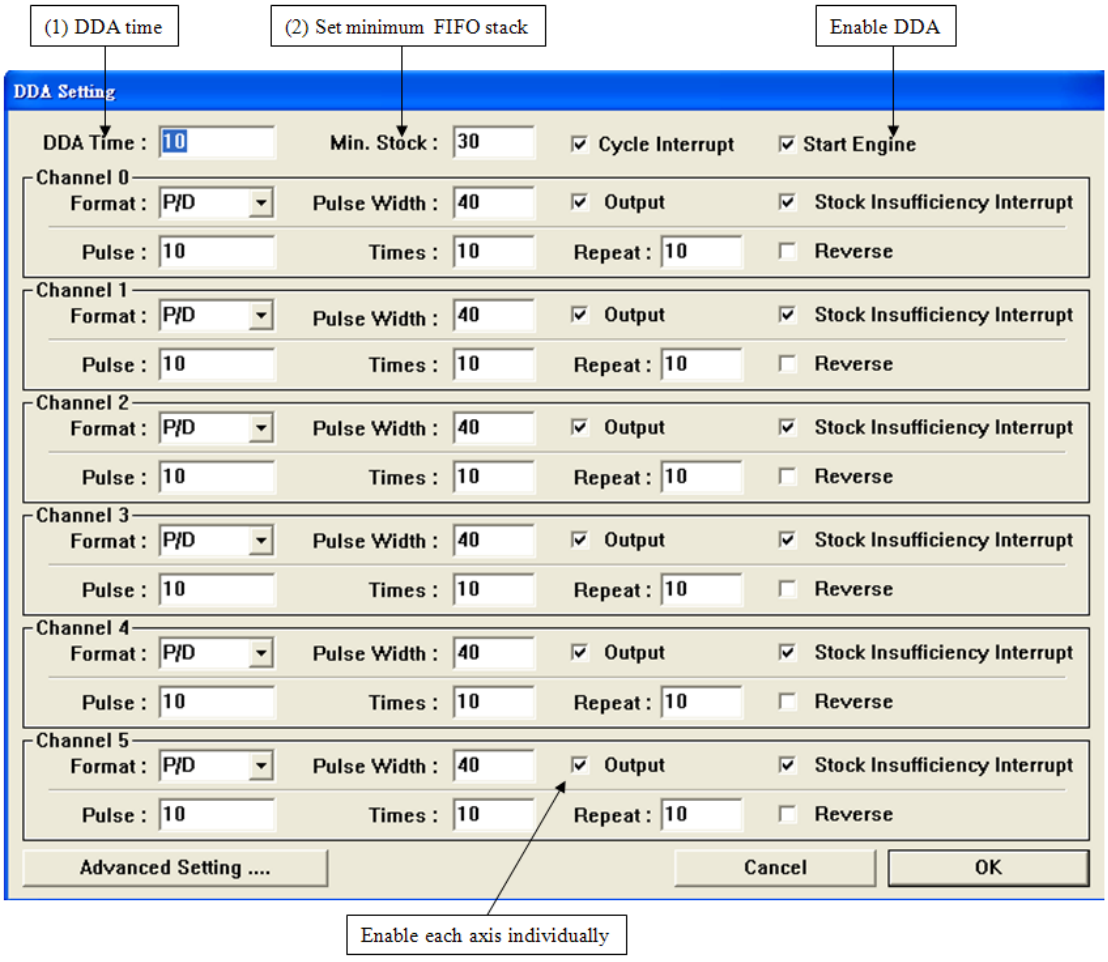

Figure 4-3.1

(4) Set DDA pulse format (as shown in Figure 4-3.2)

The DDA can be set with different pulse output formats, including Pulse/Direction, CW/CCW, A/B, and input inhibit. In addition, Advanced Setting can be selected in order to swap or invert the A, B phases of output pulses.

Inverse A: to invert the phase of signal A.

Inverse B: to invert the phase of signal B.

Swap AB: to swap signals A and B.

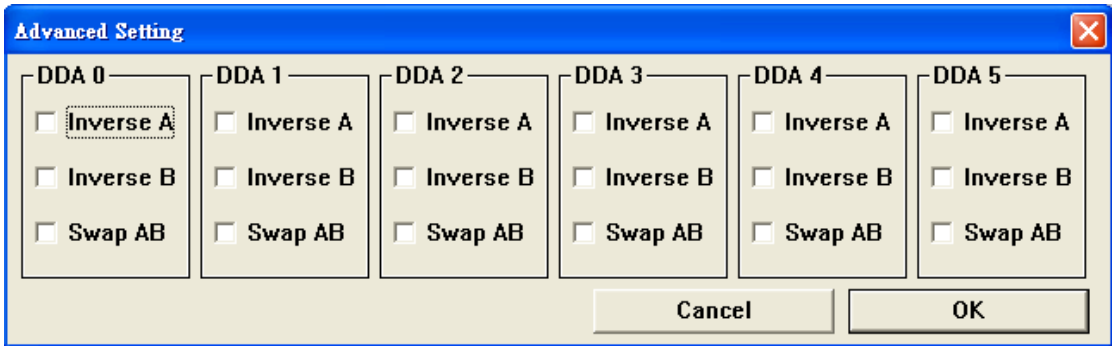

Figure 4-3.2

(5) Set DDA pulse width

When the DDA pulse output format is set to Pulse/Direction or CW/CCW, pulse width can be programmed via a software setting (as shown in Figure 4-3.3) as follows. When the DDA pulse width setting is N, DDA pulse width can be determined by equation (3)

$$DDA\ Pulse\ Width = N \times System\ Clock \qquad (3)$$

The setting should be made according to driver requirements. Figure 4-3.4 shows the output waveform.

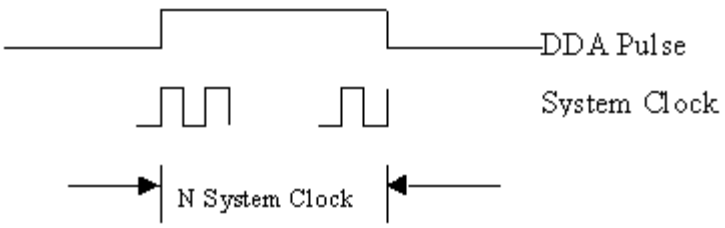

Figure 4-3.4

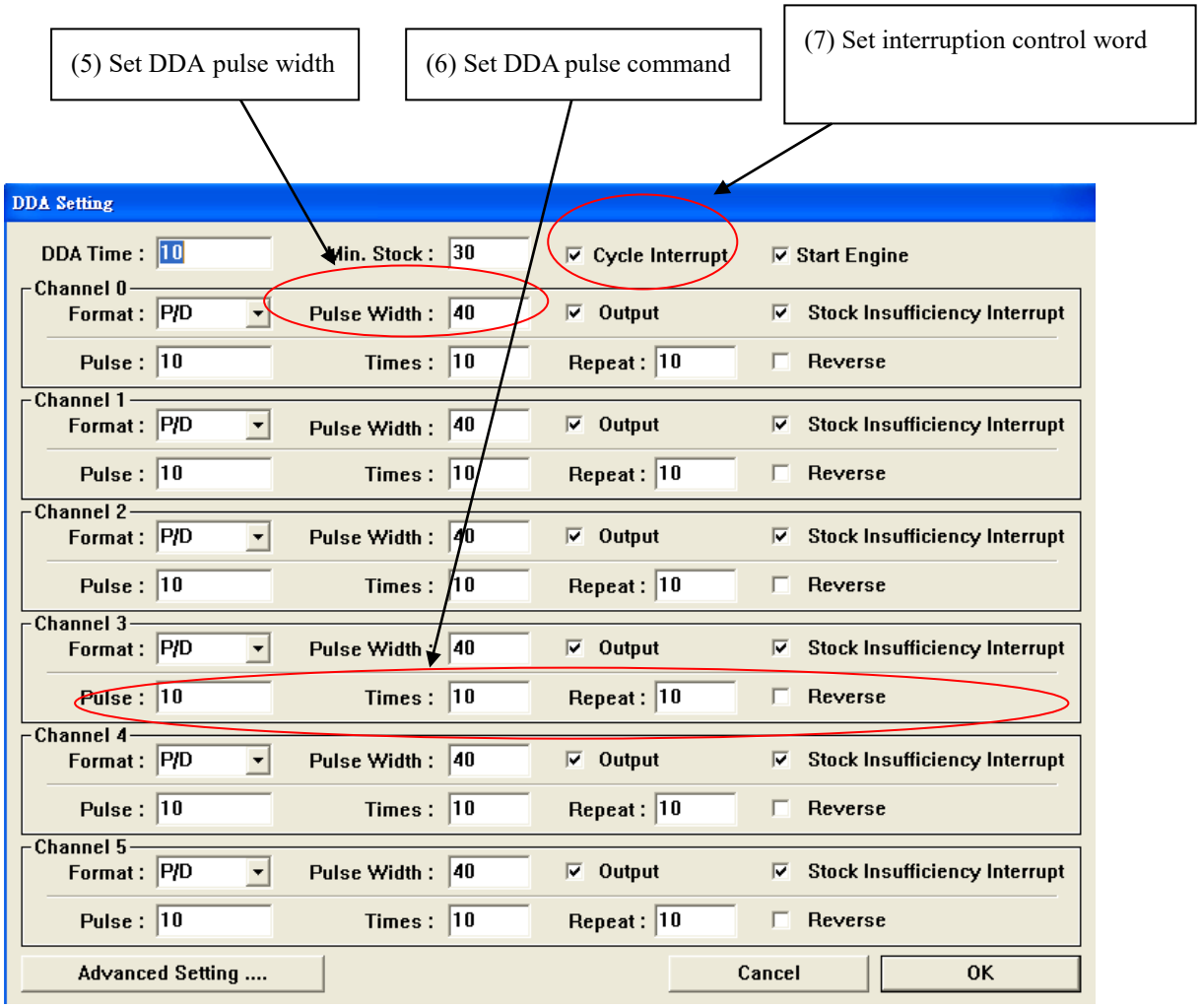

Figure 4-3.3

(6) Set DDA pulse command (as shown in Figure 4-3.3)

Pulse: the number of pulses to be sent out by each command.

Times: the number of commands.

Repeat: the number of times of repetition.

Back: whether or not backward rotation is desired.

(7) Set interruption control word

If the “Stock Insufficiency Interrupt” box of DDA channel X is checked, the corresponding axis will send out an interrupt signal to the CPU when the commands in the FIFO are reduced, by being consumed, to less than the minimum stock setting.

If the “Cycle Interrupt” box is checked, the corresponding axis/axes will send out an interrupt signal to the CPU upon consumption of each command.

4.4 ENC Main Function Options

(1) Set encoder counter input control word (as shown in Figure 4-4.1)

Inverse A: to invert the phase of signal A.

Inverse B: to invert the phase of signal B.

Inverse C: to invert the phase of signal C.

Swap AB: to swap signals A and B.

Type: to set the input signals to the A/B type, the CW/CCW type, the Pulse/Direction type, or none.

Multiple: to set the decoding multiplier when Type is set to A/B.

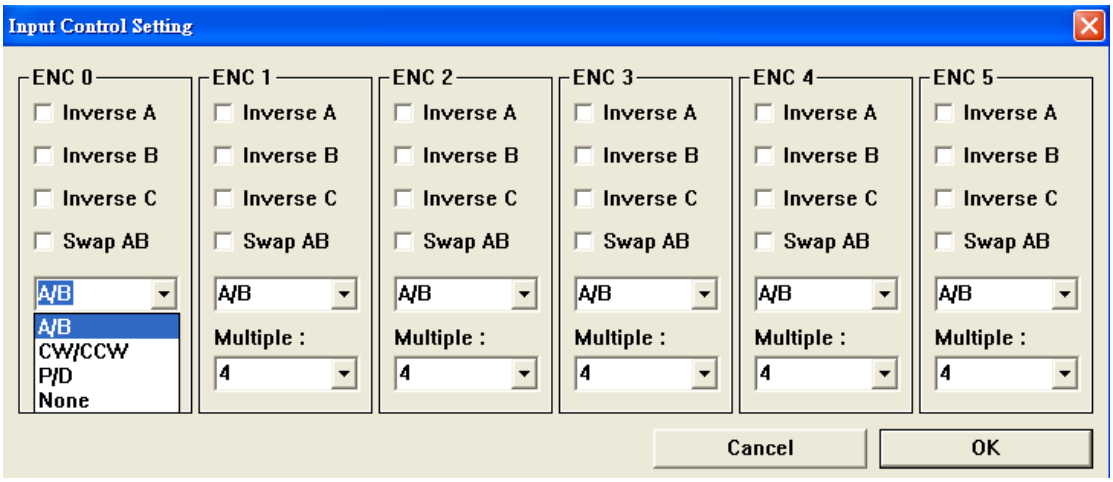

Figure 4-4.1

(2) Set index latch (as shown in Figure 4-4.2)

The index signal of any encoder channel can be used to trigger the encoder counter latch of another encoder channel. Index 0 ~ Index 5 represent the index signals of encoder counter channels 0 ~ 5 respectively. The trigger mode can be set to single or continuous trigger. (For the setting method, please refer to the following section (3).)

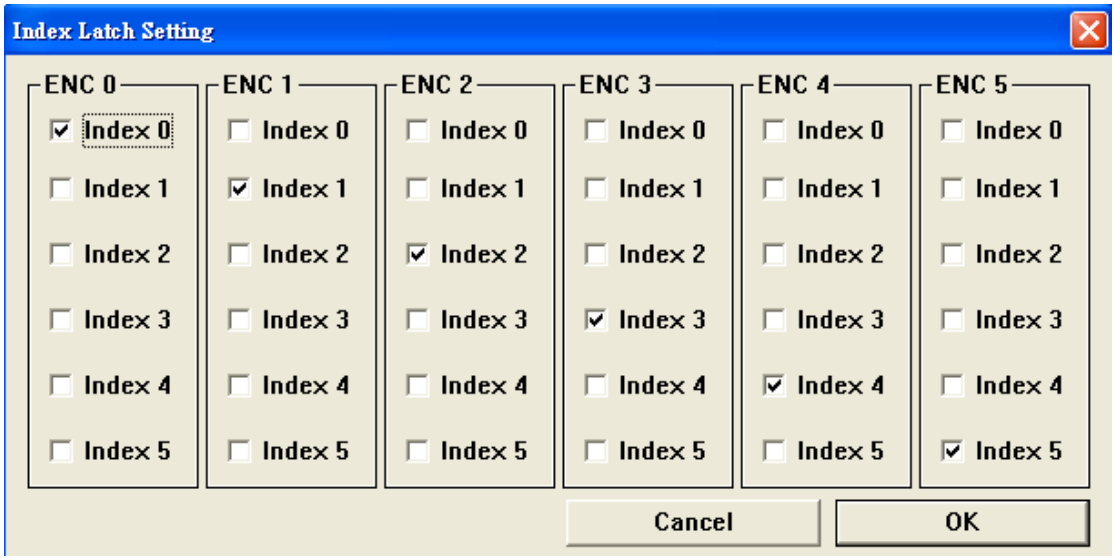

Figure 4-4.2

(3) Set general index latch (as shown in Figure 4-4.3)

Apart from being triggered by an index signal, the encoder counter latch can be triggered by a LIO input, a RIO input, or an ADC comparator, if so set. In the following figure,

LIO 0 ~ LIO 1: the 0th and the 1st LIO inputs can be set as the trigger source.

RIO 0 ~ RIO 1: the 0th and the 1st inputs of slave 0 of RIO set 0 can be set as the trigger source.

ADC 0 ~ ADC 1: comparator interrupts based on the 0th and the 1st ADC inputs can be set as the trigger source.

Trig. Repeat: If this box is checked, the trigger mode is set to the continuous trigger; otherwise, it is the single trigger.

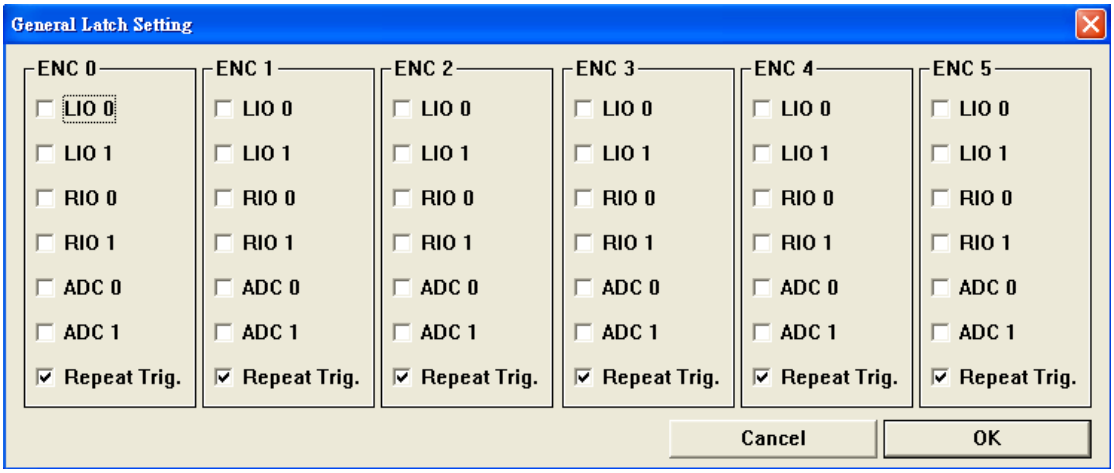

Figure 4-4.3

(4) Set encoder counter compared value (as shown in Figure 4-4.4)

To set the comparison value of the encoder counter, select Other Setting. If used in combination with the comparator setting and the interrupt index setting, an interrupt signal will be sent to the CPU when the value of the encounter counter reaches this value. Besides, the interrupt signal can be used to trigger automatic loading of DAC output. Please refer to Sections 4.9(2) and 4.9(3).

(5) Set encoder counter interruption (as shown in Figure 4-4.4)

Comparator: to set the comparator interrupt function. CPU interruption takes place when the encoder input value is equal to the comparison value. Besides, the interrupt signal can be used to trigger the DAC preload function. (Please refer to Sections 4.4(4), 4.9(2), and 4.9(3).)

Interrupt Index: to set the index interrupt function of ENC 0 ~ ENC 5 individually.

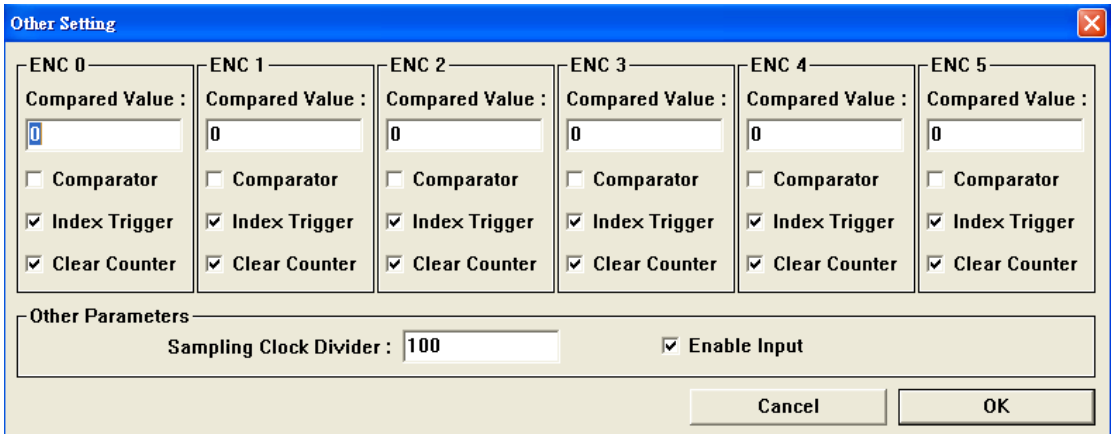

Figure 4-4.4

(6) Clear encoder counter (as shown in Figure 4-4.4)

This is to clear the counter value of the selected encoder channel(s). Once an encoder channel is selected, the counter value of that encoder channel will be cleared upon program execution. The Clear Encoder Counter function is disabled at boot-up. Please check the "Clear Counter" box when the program is run for the first time.

(7) Set encoder sampling clock divider (as shown in Figure 4-4.4)

The encoder is a digital filter. When the Sampling Clock Divider setting is N, the sampling rate of the encoder can be determined by Equation (4)

$$Period\ of\ Sampling\ Rate = (N+1) / System\ Clock \quad (4)$$

An input signal must be High in three consecutive samples to be considered as High by the system. Therefore, the valid pulse width can be determined by equation (5)

$$Valid\ Pulse\ Width = 3 \times Period\ of\ Sampling\ Rate \quad (5)$$

Assume System Clock is 40 MHz and Clock Divider is set to 10. Then the pulse width of the input signal must at least be greater than 1.68μs, or the input signal will be filtered out by the digital filter.

$$Valid\ Pulse\ Width = 3 \times (10+1) / 40MHz = 1.68\mu s$$

(8) Set encoder start/stop (as shown in Figure 4.4-4)

This is to enable/disable the encoder counting function. If it is desired to enable any encoder channel, the "Enable Input" box must be checked.

4.5 PCL Main Function Options

(1) Set closed loop gain (as shown in Figure 4.5-1)

The closed loop gain setting has two parameters: the proportional gain and the scaling gain.

M Gain is the proportional gain. Note: This value should be set to a positive value. A negative value may result in positive feedback.

S Gain is the scaling gain. A positive value means multiplying by a multiple of 2; a negative value means dividing by a multiple of 2.

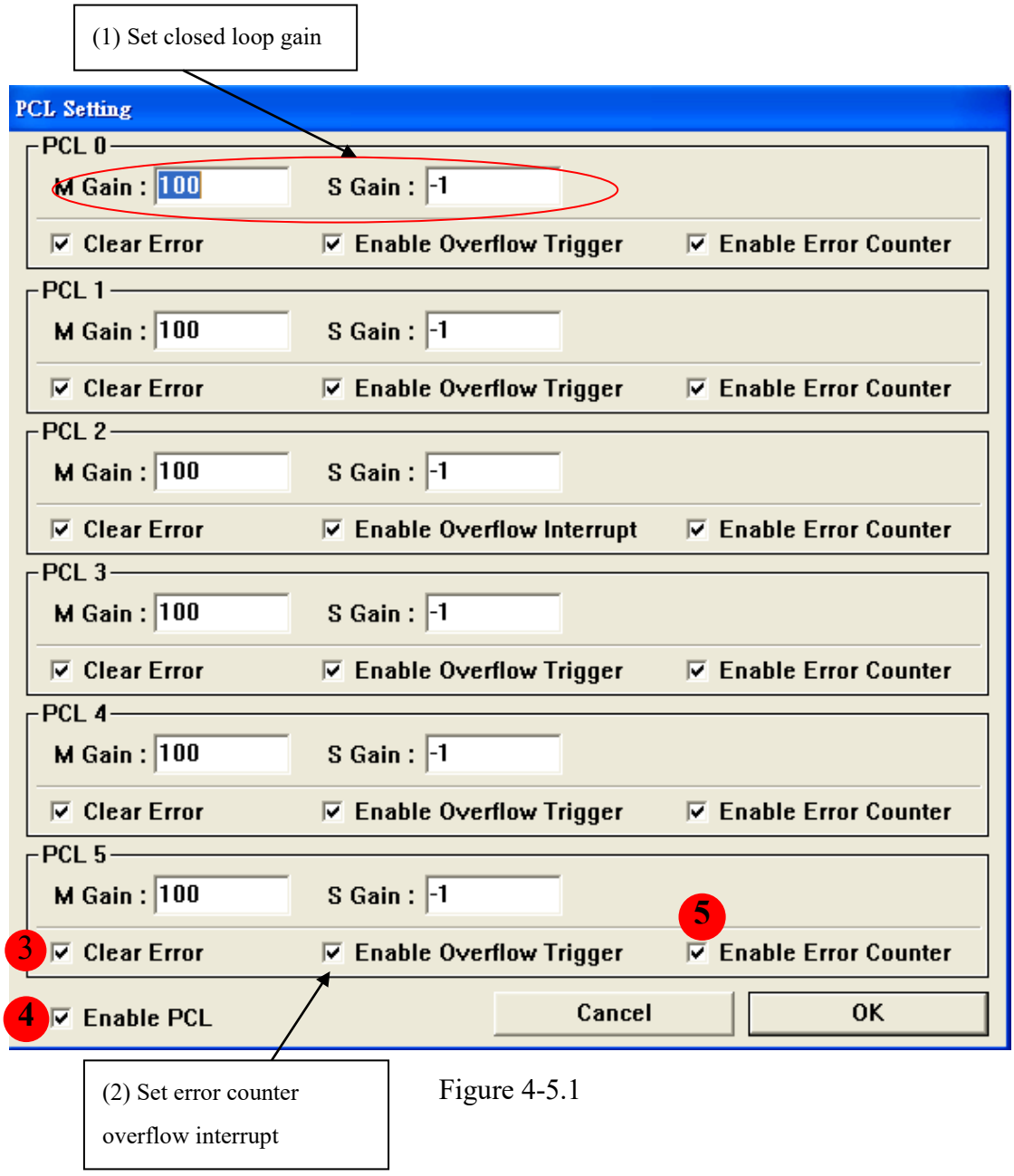

Figure 4-5.1

(2) Set overflow interrupt (as shown in Figure 4-5.1)

If the “Enable Interrupt” box is checked, an interrupt signal will be sent to the CPU when the error counter overflows.

(3) Set error counter clear (as shown in Figure 4-5.1)

This is to clear the error counter value(s) of the specified axis (or axes) and terminate the error counter overflow status. This setting will clear the error counter value(s) of the specified axis (or axes) upon program execution.

(4) Set closed loop enable (as shown in Figure 4-5.1)

This is to enable PCL control of each individual axis. When PCL control of a certain axis is desired, the “Enable PCL X” box of that axis must be checked. If the “Enable PCL X” box of any axis has been checked, the “Enable PCL” box must also be checked in order for PCL control to take effect. If the “Enable PCL” box is not checked, PCL control will be disabled, whether the “Enable PCL X” box of any axis has been checked or not.

To use the PCL hardware closed loop function, it is required to also set the DDA, ENC, and DAC functions. For the details of parameter settings, please refer to Sections 4.3, 4.4, and 4.6.

(5) Set enable error counter (as shown in Figure 4-5.1)

If the “Enable Error Counter” box is checked, an interrupt signal will be sent to the CPU when the error counter overflows.

4.6 LIO Function Options

(1) Set local digital output (as shown in Figure 4-6.1)

This is to set the output statuses of the local I/O connections. The local I/O connections can be grouped into groups of four and programmed for input or output via a software setting. All the local I/O connections are programmed for input at boot-up. LD 27 ~ LD 0 are the 27th to the 0th outputs respectively.

(2) Set local digital output enable (as shown in Figure 4-6.1)

This is to enable/disable output from the local I/O connections. To enable output, please check the "Output" box(es) and wire the hardware modules accordingly. When output is enabled, the output status(es) can be read back via software.

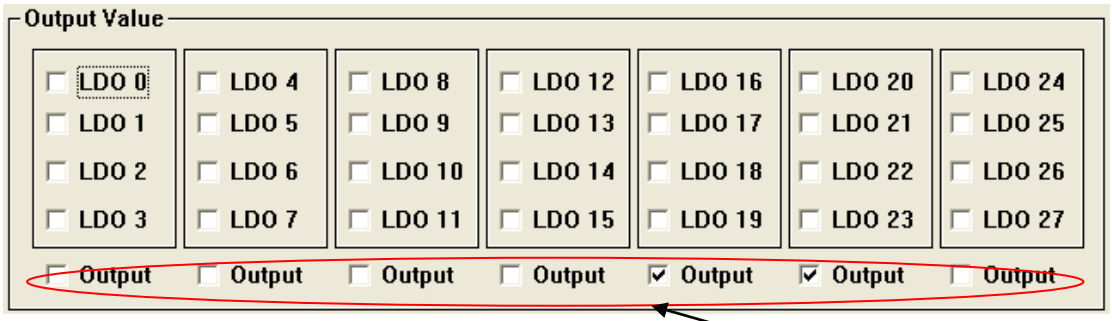

Figure 4-6.1

(3) Set watchdog control (as shown in Figure 4-6.2)

When the watchdog timer value is set to *W*, the watchdog timer's time can be determined by Equation (6)

$$Period\ of\ Watchdog\ Timer = W \times Period\ of\ Interval\ Timer \quad (6)$$

Once the watchdog timer is enabled, the user must clear the watchdog timer before watchdog timeout; otherwise, a Reset signal will be output upon watchdog timeout to restore all the hardware to its initial state. The Reset duration is pre-programmable and can be determined by equation (7)

$$Reset\ Duration = Timer / System\ Clock \quad (7)$$

This testing program can simulate watchdog timeout as follows. When Refresh Times is set to R, the testing program will clear the watchdog timer R times at a fixed interval equal to the Period of Interval Timer. When Refresh Times is set to 0, the program will refresh the watchdog timer continuously. To use the watchdog timer function, please check the "Enable" box.

Figure 4-6.2

(4) Set trigger control (as shown in Figure 4-6.3)

The local input connections can be programmed for trigger. In Figure 4-6.3, LDI 0 ~ LDI 7 are programmed as the 0th to the 7th inputs respectively while DFI 0 ~ DFI 6 are the 0th to the 6th double-function inputs respectively. The interrupt trigger mode can be set to rising edge trigger (Rise), falling edge trigger (Fall), or level change trigger (Both).

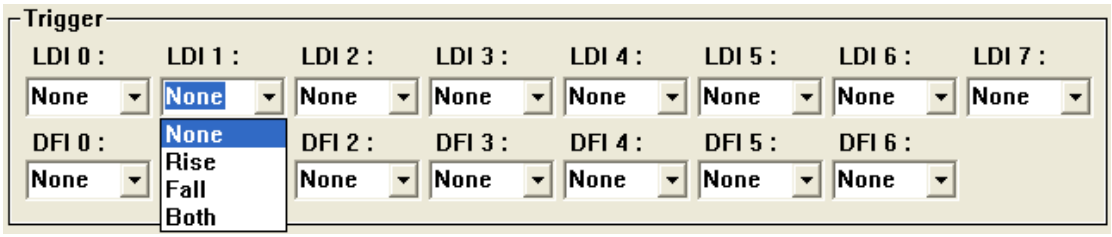

Figure 4-6.3

(5) Set timer value (as shown in Figure 4-6.4)

This is to set the timer's time, which can be determined by equation (8)

$$Period\ of\ Interval\ Timer = Timer / System\ Clock \qquad (8)$$

When setting the timer, it is required that the interrupt function is also enabled so that the hardware will trigger an interrupt at a fixed time, which is used as the time base (please refer to the interrupt setting and start/stop setting). In addition, whether or not the interrupt function is enabled, the timer's time setting (the time base) will be used to trigger the watchdog timer (please refer to the watchdog setting).

To set the Interval Timer interrupt function, please check the "Enable Timer" box.

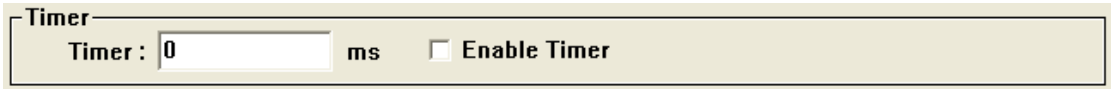

Figure 4-6.4

4.7 RIO 0 Main Function Options

(1) Set remote I/O output value (as shown in Figure 4-7.1)

The EPCIO ASIC can be connected with 2 parallel remote I/O sets, and each set can be serially connected with 3 slaves, or remote serial I/O modules (for EDIO-S00X), each remote serial I/O module having 64 inputs and 64 outputs. Therefore, in terms of expansion, a maximum of 384 input connections and 384 output connections can be provided.

Select the desired output connections, and they will be shown on EDIO-S00X.

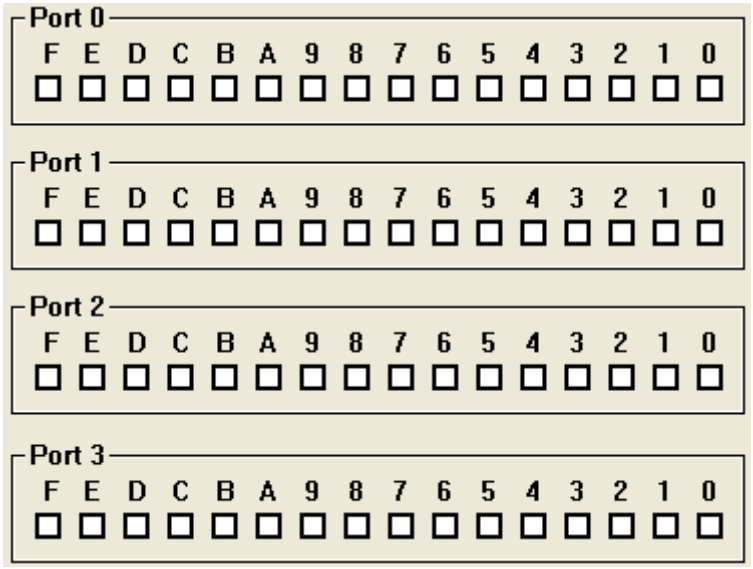

Figure 4-7.1

(2) Set remote I/O clock divider (as shown in Figure 4-7.2)

This is to set the transmission clock of the remote serial I/O module(s). In Figure 4-7.2, RIO 0 Clock Divider is the transmission clock divider of RIO set 0. When Clock Divider is set to N, the transmission clock can be determined by the following equation:

$$\textit{Transmission Clock: } SCLK = \textit{System Clock} / 2(N+1) \quad (0 \leqq N \leqq 255) \qquad (8)$$

When a remote I/O set is connected with M remote serial I/O modules, the time required for data update is approximately:

$$\textit{Data Update Time: } 100 \times M \times SCLK \quad (1 \leqq M \leqq 3) \qquad (9)$$

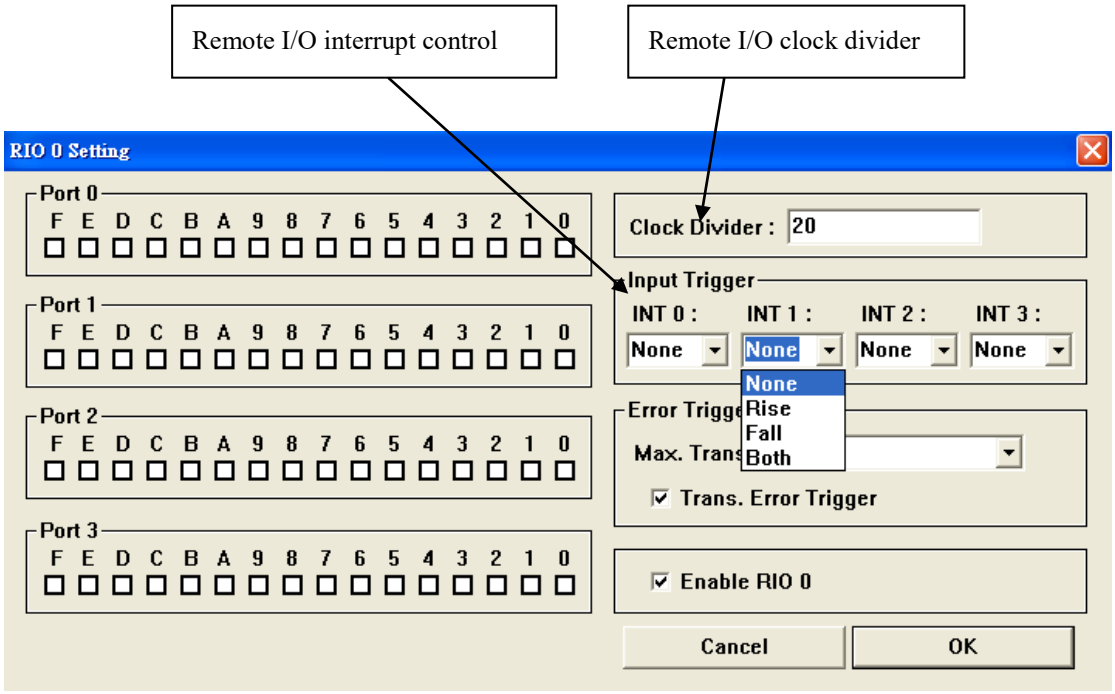

Figure 4-7.2

(3) Set remote I/O interrupt control (as shown in Figure 4-7.2)

Each remote serial I/O module has 4 inputs which can be programmed for interrupt. The interrupt trigger mode can be rising edge trigger (Rise), falling edge trigger (Fall), or level change trigger (Both). In Figure 4-7.2, INT 0 ~ INT 3 are the 0th to the 3rd inputs respectively.

(4) Set remote I/O maximum transmission error (as shown in Figure 4-7.3)
This is to set the maximum number of times of data retransmission that is allowed when a remote I/O data transmission error occurs. If the value is set to 0, the system will automatically retransmit for up to 16 times when an error occurs.

(5) Set remote I/O transmission error interrupt (as shown in Figure 4-7.3)
This can be so set that, if an error occurs during data transmission by the remote I/O, an interrupt will be triggered so that the hardware automatically stops data transmission. The number of times of retransmission can be set as needed. Please refer to Section 4.7(4) for the maximum transmission error setting.

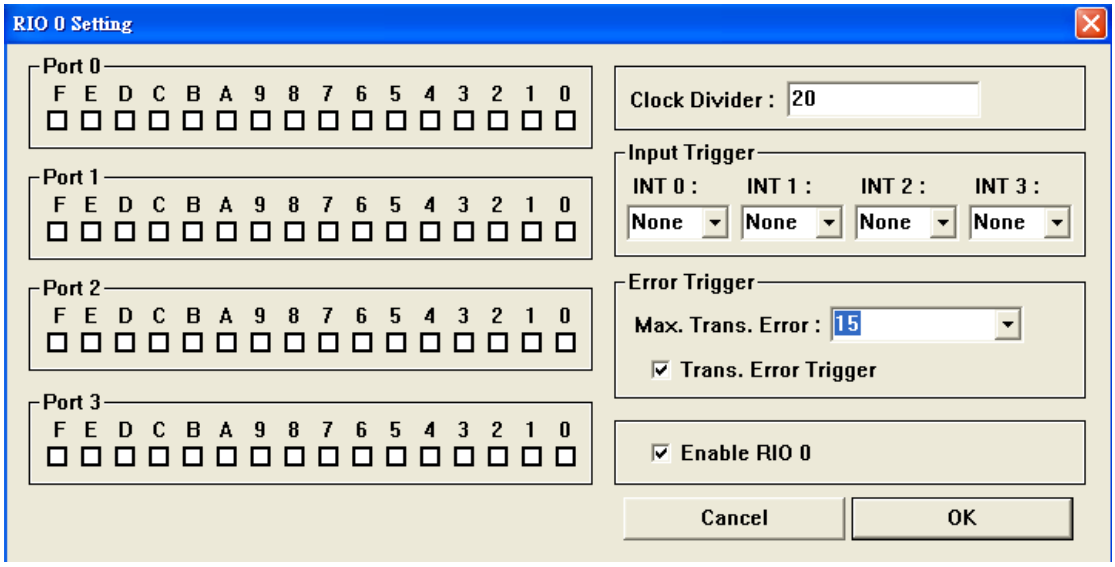

Figure 4-7.3

(6) Set remote I/O enable status (as shown in Figure 4-7.3)
This is to enable or disable transmission of remote I/O set 0. Check the "Enable RIO 0" box when it is desired to enable remote serial I/O set 0. When the system is connected with only one remote serial I/O module, be sure to enable the corresponding I/O set. An incorrect setting will lead to data transmission/reception error and bring system operation to a halt.
This testing program can simulate output in a sequential and cyclic manner. To run the simulation, please use the Free Run mode. The data update time is determined by the time programming of the timer. Please refer to Sections 4.5(4) and 4.5(5).

4.8 RIO 1 Main Function Options

(1) Set remote I/O output value (as shown in Figure 4-8.1)

The EPCIO ASIC can be connected with 2 parallel remote I/O sets, and each set can be serially connected with 3 slaves, or remote serial I/O modules (for EDIO-S00X), each remote serial I/O module having 64 inputs and 64 outputs. Therefore, in terms of expansion, a maximum of 384 input connections and 384 output connections can be provided.

Select the desired output connections, and they will be shown on EDIO-S00X.

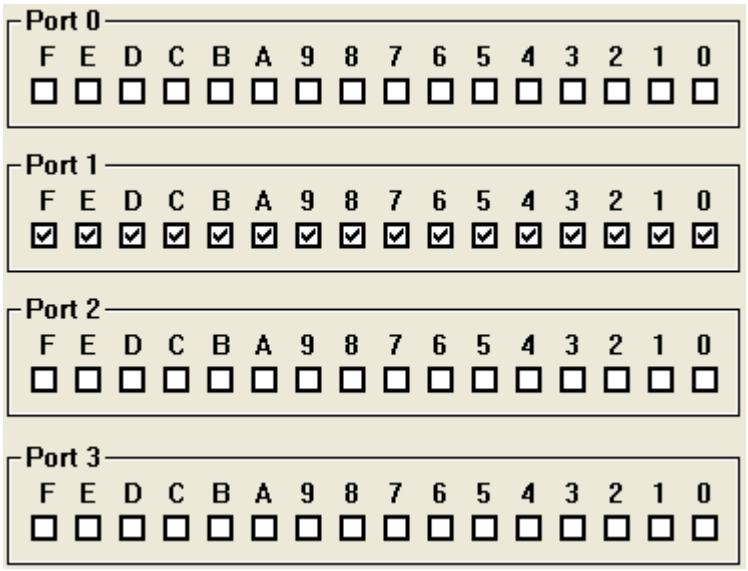

Figure 4-8.1

(2) Set remote I/O clock divider (as shown in Figure 4-8.2)

This is to set the transmission clock of the remote serial I/O module(s). In Figure 4-8.2, RIO 1 Clock Divider is the transmission clock divider of RIO set 1. When Clock Divider is set to N, the transmission clock can be determined by equation (10):

$$\textit{Transmission Clock: SCLK} = \textit{System Clock} / 2(N+1) \quad (0 \leqq N \leqq 255) \qquad (10)$$

When a remote I/O set is connected with M remote serial I/O modules, the approximate time required for data update can be determined by equation (11):

$$\textit{Data Update Time: } 100 \times M \times \textit{SCLK} \ (1 \leqq M \leqq 3) \qquad (11)$$

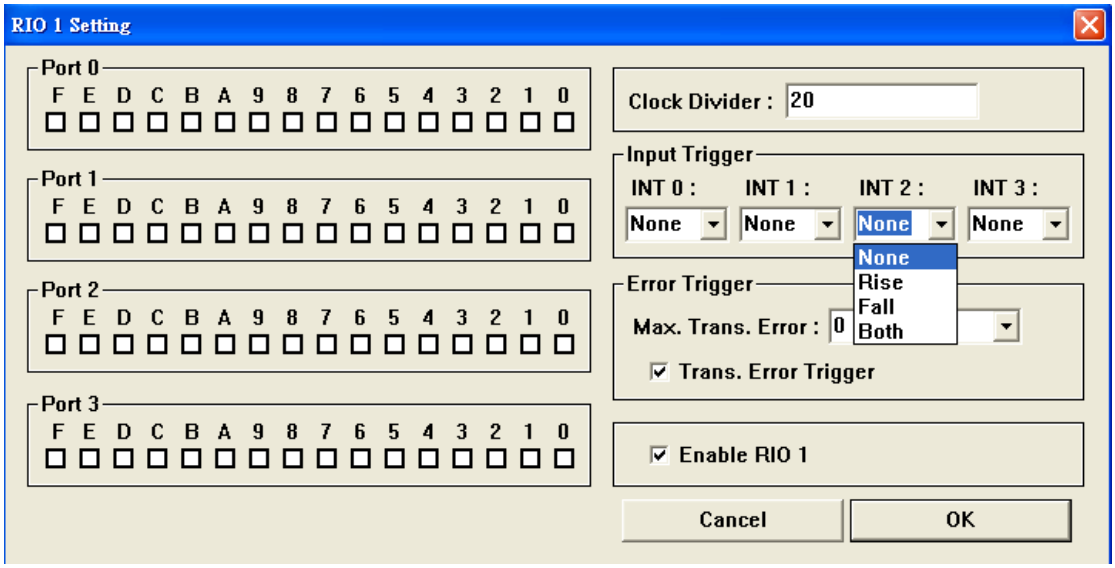

Figure 4-8.2

(3)Set Remote I/O trigger control (as shown in Figure 4-8.2)

Each remote serial I/O module has 4 inputs which can be programmed for interrupt. The interrupt trigger mode can be rising edge trigger (Rise), falling edge trigger (Fall), or level change trigger (Both). In Figure 4-8.2, INT 0 ~ INT 3 are the 0th to the 3rd inputs respectively.

(4) Set remote I/O maximum transmission error (as shown in Figure 4-8.3)

This is to set the maximum number of times of data retransmission that is allowed when a remote I/O data transmission error occurs. If the value is set to 0, the system will automatically retransmit for up to 16 times when an error occurs.

(5) Set remote I/O transmission error interrupt (as shown in Figure 4-8.3)

If an error occurs during data transmission by the remote I/O, an interrupt will be triggered so that the hardware automatically stops data transmission. The number of times of retransmission can be set as needed. Please refer to Section 4.8(4) for the maximum transmission error setting.

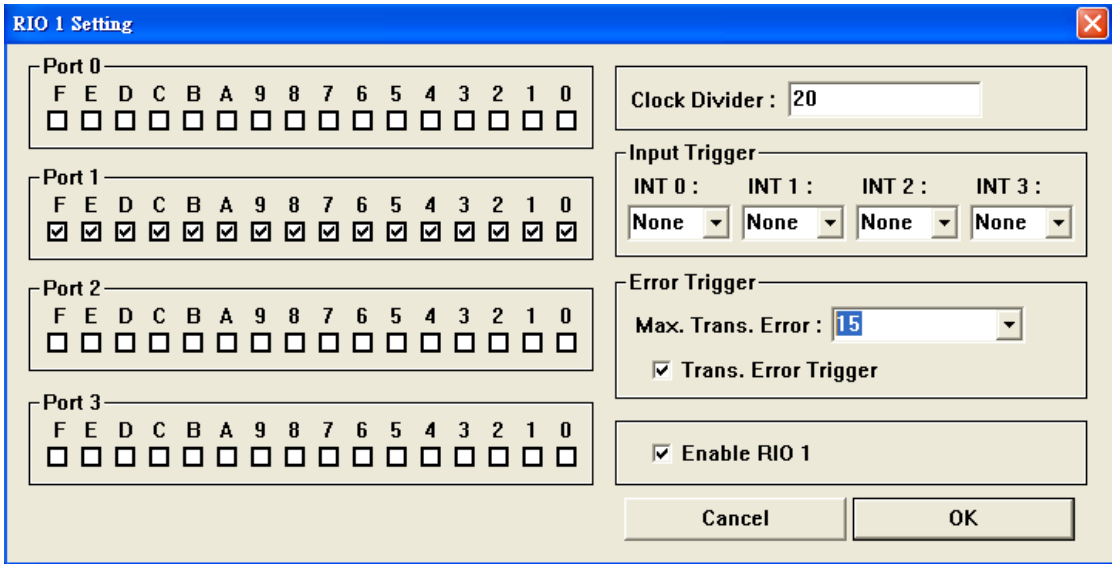

Figure 4-8.3

(6) Set remote I/O enable status (as shown in Figure 4-8.3)

This is to enable or disable transmission of remote I/O set 1. Check the "Enable RIO 1" box when it is desired to enable remote serial I/O set 1. When the system is connected with only one remote serial I/O module, be sure to enable the corresponding I/O set. An incorrect setting will lead to data transmission/reception error and bring system operation to a halt.

This testing program can simulate output in a sequential and cyclic manner.

4.9 DAC Main Function Options

(1) Set DAC output value

The output value of the DAC can be output as a sawtooth wave (as shown in Figure 4-9.1) and be set as shown in Figure 4-9.2. The user can set a fixed voltage output for each axis, and this can be set by adjusting the output voltage value of each axis with a mouse.

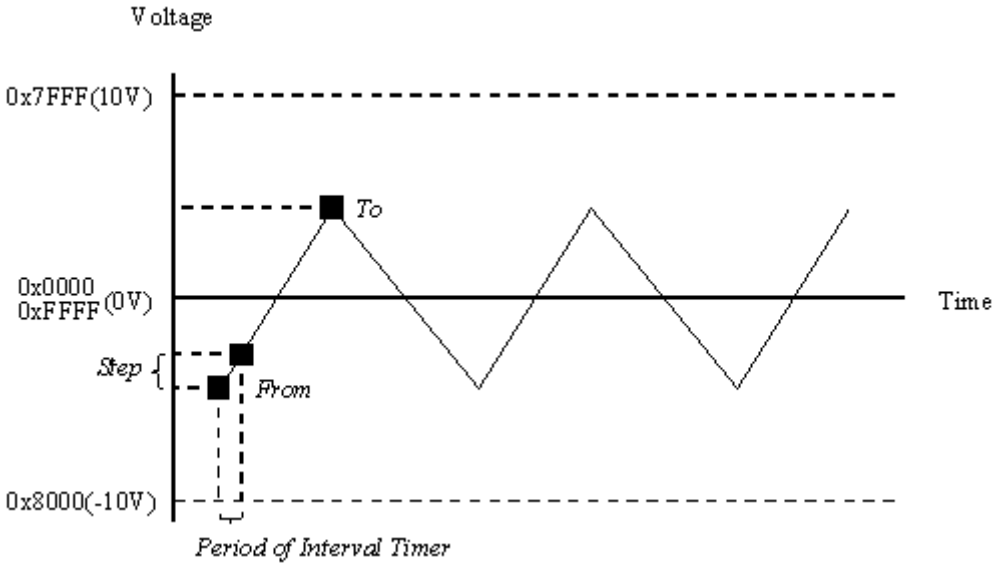

Figure 4-9.1

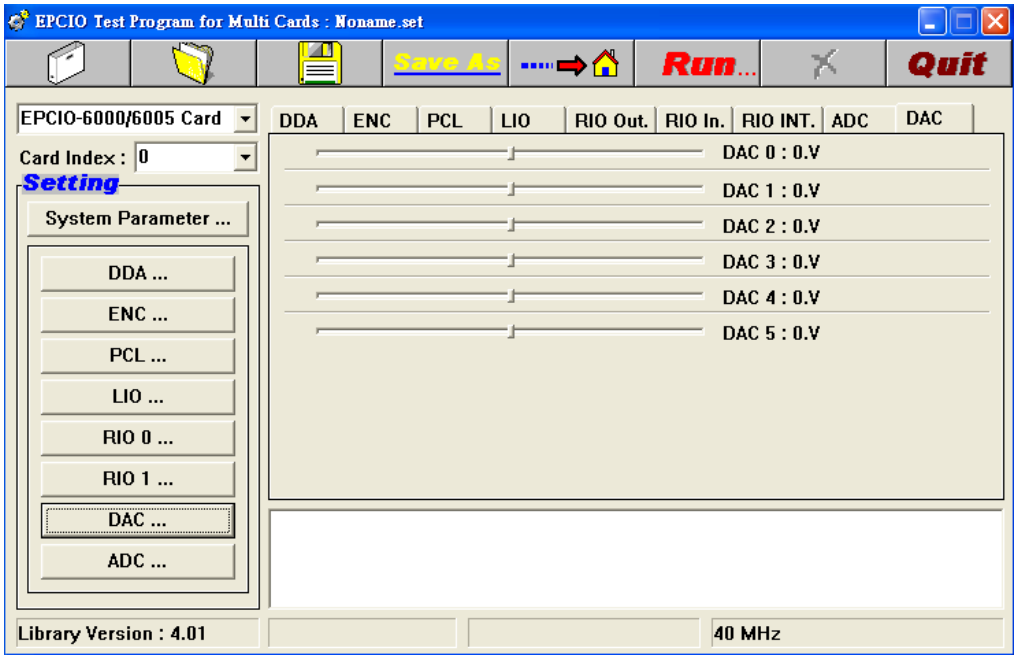

Figure 4-9.2

(2) Set DAC preload value (as shown in Figure 4-9.3)

This is to preload the voltage to be automatically output from the DAC. When a trigger condition associated with any of ENC, LIO, RIO, DAC, and ADC is met, and automatic output of the preloaded voltage from the DAC is enabled (refer to Sections (3), (4), (5), (6), and (7)), the hardware will automatically output the preloaded voltage.

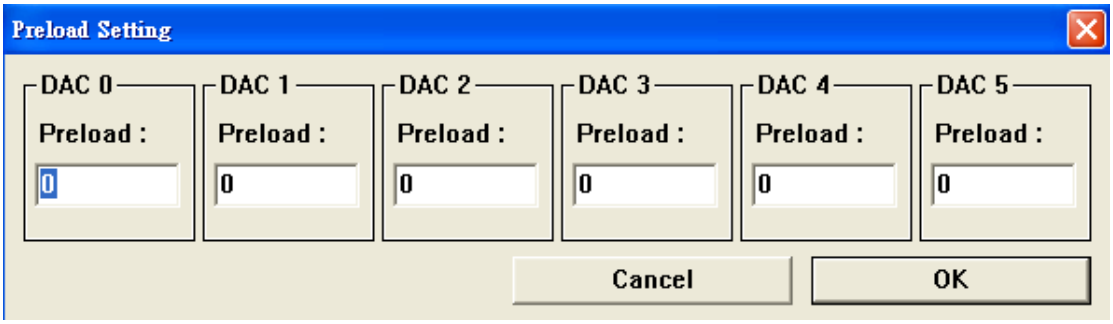

Figure 4-9.3

(3) Set encoder as DAC trigger source (as shown in Figure 4-9.4)

This is to set the output of an interrupt signal from the comparator of a certain encoder channel as the trigger source for the DAC's automatic voltage output.

ENC 0 ~ ENC 5: check boxes for the comparator interrupt function of encoder channels 0 to 5. To use this function, please check the "Comparator" box in Section 4.4(5).

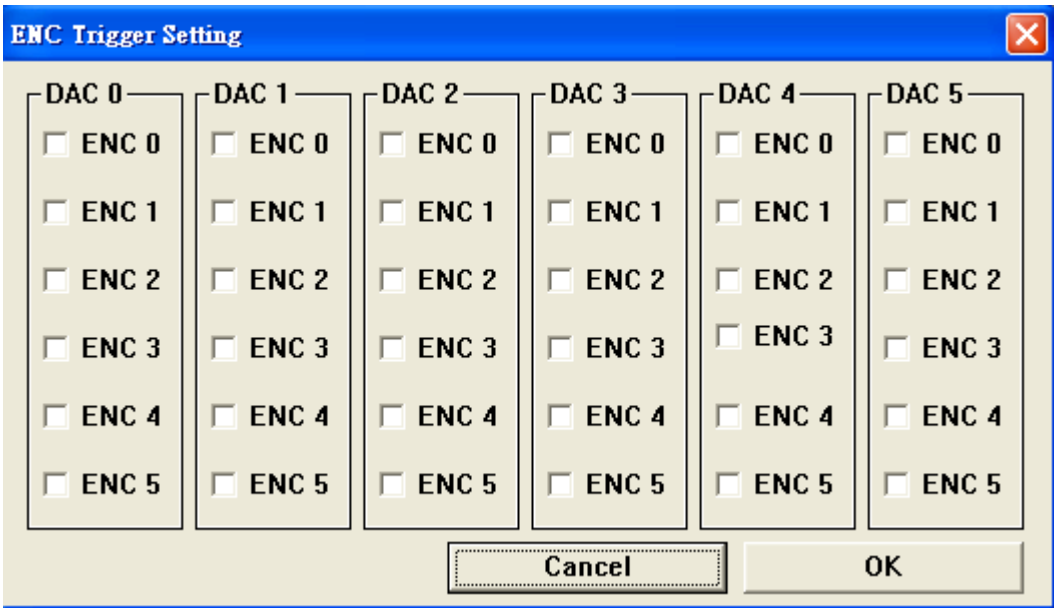

Figure 4-9.4

(4) Set ADC as DAC trigger source (as shown in Figure 4-9.5)

This is to set the output of an interrupt signal from the comparator of any of ADC 0 ~ ADC 7 as the trigger source for the DAC's automatic voltage output. ADC 0 ~ ADC 7: check boxes for the comparator interrupt function of ADC channels 0 to 7. To use this function, please enable the interrupt control function in Section 4.10(2).

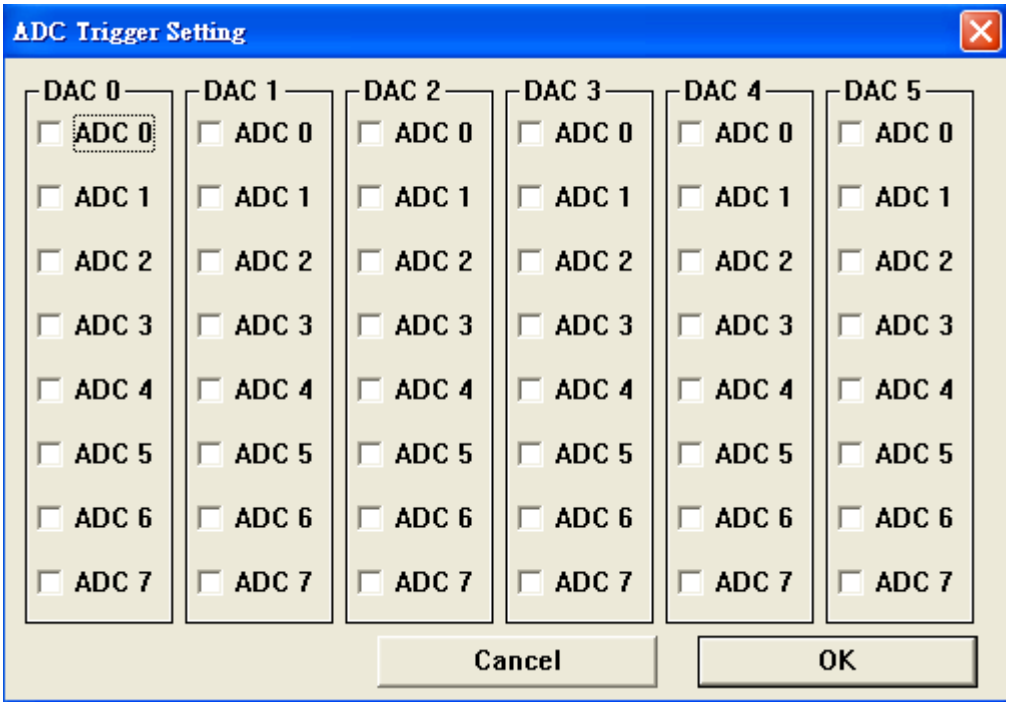

Figure 4-9.5

(5) Set LIO as DAC trigger source (as shown in Figure 4-9.6)

This to select the interrupt signal of any of the first 4 LDI inputs and first 4 DFI inputs as the trigger source for the DAC's automatic voltage output.

LI 0 ~ LI 3: check boxes for the interrupt function of LDI 0 ~ 3. To use this function, please specify the trigger mode as described in Section 4.6(4).

DI 0 ~ DI 3: check boxes for the interrupt function of DFI 0 ~ 3. To use this function, please specify the trigger mode as described in Section 4.6(4).

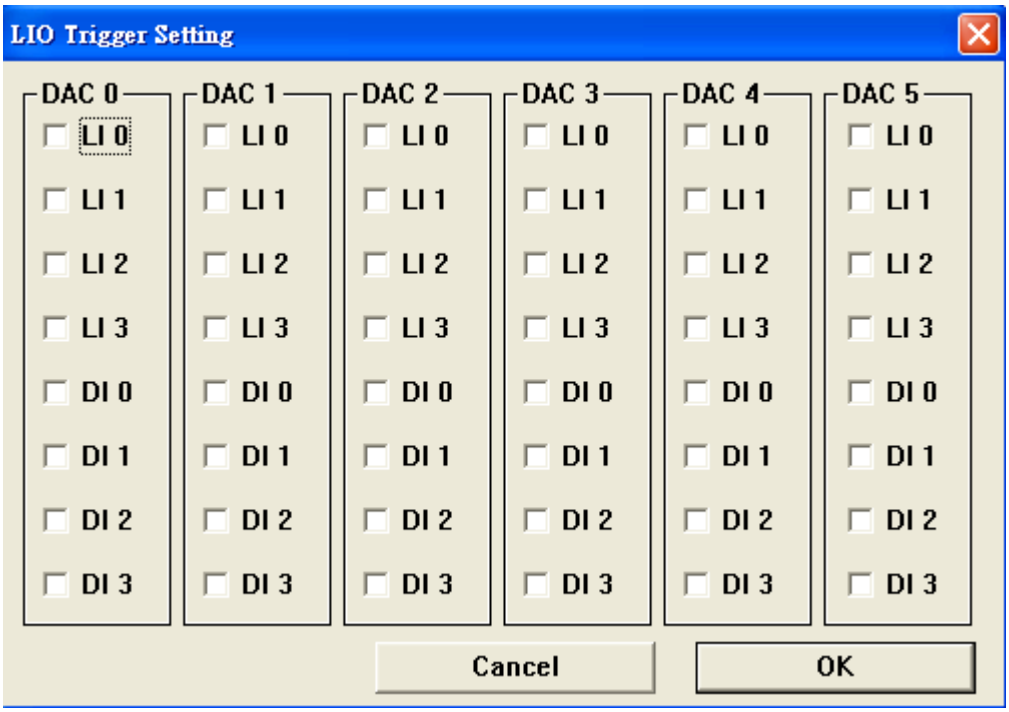

Figure 4-9.6

(6) Set RIO as DAC trigger source (as shown in Figure 4-9.7)

This is to select the interrupt signal of any of the first 4 inputs of slave 0 of RIO set 0 or the interrupt signal of any of the first 4 inputs of slave 0 of RIO set 1 as the trigger source for the DAC's automatic voltage output.

R 00 ~ R 03: check boxes for the interrupt function of the first four inputs of slave 0 of RIO set 0. To use this function, please enable the interrupt function in Section 4.7(3).

R 10 ~ R 13: check boxes for the interrupt function of the first four inputs of slave 0 of RIO set 1. To use this function, please enable the interrupt function in Section 4.8(3).

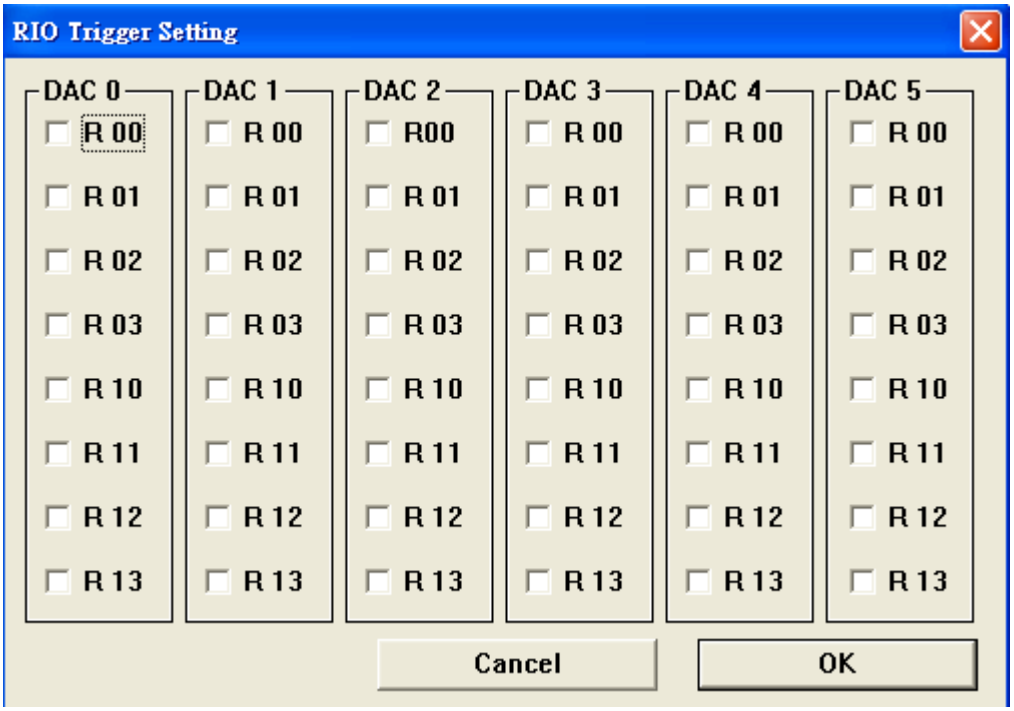

Figure 4-9.7

(7) Set DAC source selection (as shown in Figure 4-9.8)

Trigger: The “Trigger” box should be checked if it is desired to use the automatic DAC voltage output function stated in Sections (2), (3), (4), (5), and (6).

Source: When DAC is selected as the Source, output commands are directly programmed by software. When PCL is selected as the Source, output commands are automatically generated by the hardware closed loop (PCL). (Refer to Section 4.5.)

Please note that automatic DAC output is effective only when DAC is selected as the Source.

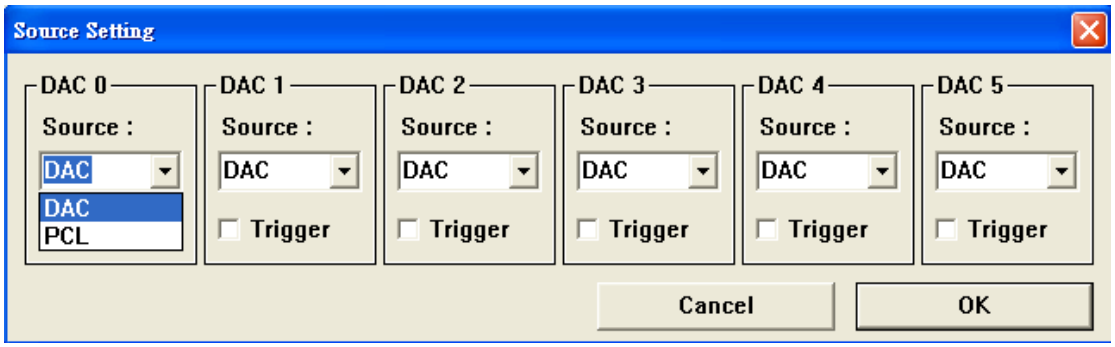

Figure 4-9.8

(8) Set DAC clock divider (as shown in Figure 4-9.9)

This is to set the DAC's transmission clock divider. When "Clock Divider" field is set to N, the DAC's transmission clock can be determined by equation (12):

$$Period\ of\ DAC\ Clock = 4(N+1) / System\ Clock \quad (12)$$

The DAC data update time can be determined by equation (13):

$$DAC\ Data\ Update\ Time = 20 \times Period\ of\ DAC\ Clock \quad (13)$$

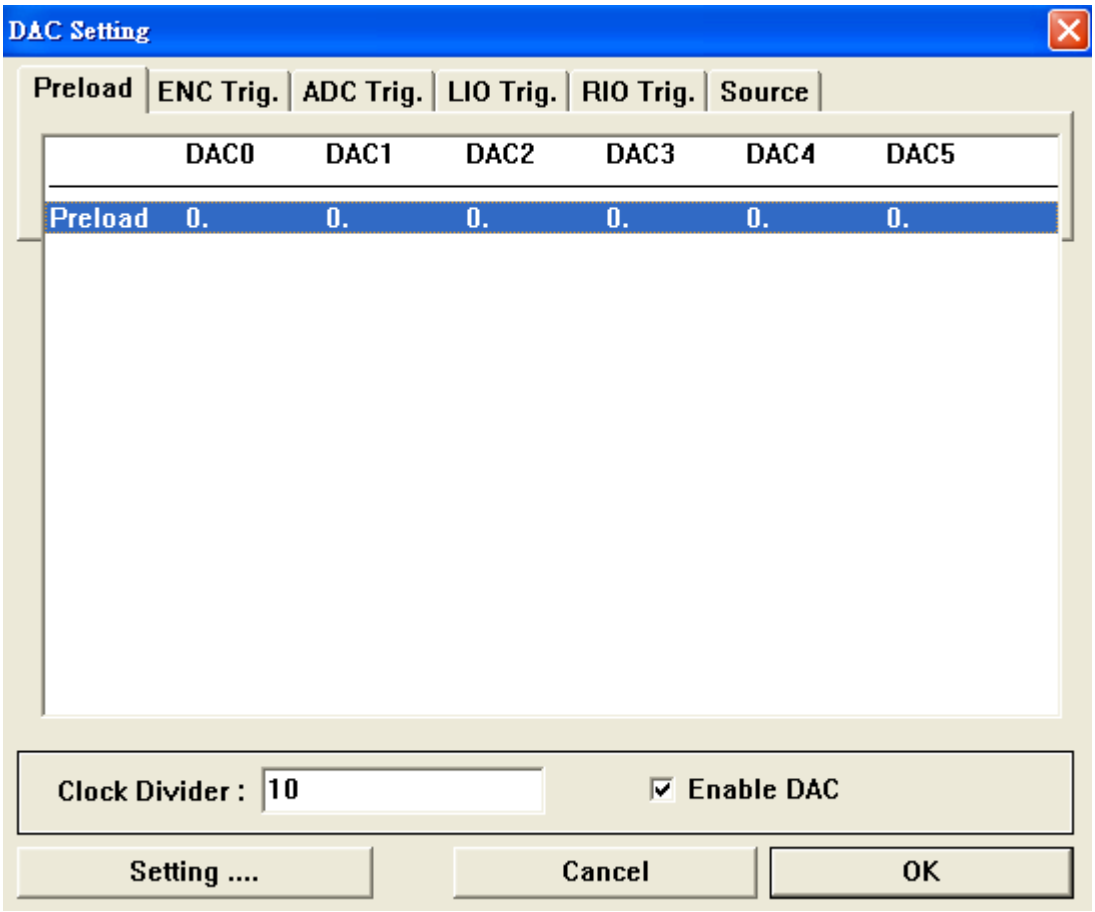

Figure 4-9.9

(9) Set DAC enable (as shown in Figure 4-9.9)

Check the "Enable DAC" box if it is desired to enable DAC output.

4.10 ADC Main Function Options

(1) Set ADC preload value

Each ADC channel has a comparator. The comparison value of each comparator needs to be preloaded first. When the comparison condition is met, an interrupt signal will be sent to the CPU.

■ ADC comparator operation:

A. The comparator first masks the last 1, 2, or 3 bits of the voltage read from the ADC to form the masked value (i.e., to form the masked value by regarding the last 1, 2, or 3 bits of the voltage read from the ADC as 0), and then compares the masked value with the preloaded comparison value in the comparator. The comparison result is made known to the CPU through an interrupt. The method of comparison is described below.

Note 1: The masked value will be updated when the voltage read from the ADC is refreshed.

Note 2: The number of masked bits can also be 0, meaning no bits are masked. In that case, the masked value is equal to the original voltage read from the ADC.

B. Method of comparison: Any of the following three comparison methods can be selected to trigger an interrupt:

(1) When the masked value changes from less than the preloaded value to greater than or equal to the preloaded value.

(2) When the masked value changes from greater than the preloaded value to less than or equal to the preloaded value.

(3) Both of the above two conditions can trigger an interrupt.

■ ADC comparator software operation

A. To preload comparison value

➔In the “Preload” field in the ADC Setting window (refer to Figure 4.10.1), set the preloaded comparison value according to the following:

(1) When the ADC input voltage mode is the unipolar mode, the comparison value setting range is from 0V to 20V.

(2) When the ADC input voltage mode is the bipolar mode, the comparison value setting range is from -10V to 10V.

B. To set the trigger mode of comparator interrupt

➔Whether an ADC channel is to trigger comparator interrupt and the method of comparison can be set in the “Trig. Control” list of each channel in the ADC Setting window (refer to Figure 4.10.1). The parameters in the “Trig. Control” list are described as follows:

(1) None: no comparator interrupt.

(2) L2GE (less to greater or equal): when the masked value changes from less than the preloaded value to greater than or equal to the preloaded value.

(3) GE2L (greater to equal or less): when the masked value changes from greater than the preloaded value to less than or equal to the preloaded value.

(4) L2GE&GE2L: both of the above two conditions can trigger an interrupt.

C. To set masked value

➔The masked value can be set in the “Compare Mask” list in the Mode Setting section of the ADC Setting window (Refer to Figure 4-10.1).

D. To enable ADC

➔To enable a certain ADC channel, check the “Enable” box and “Enable ADC” box in the ADC Setting window (Refer to Figure 4-10.1).

E. To execute

➔Click Run.

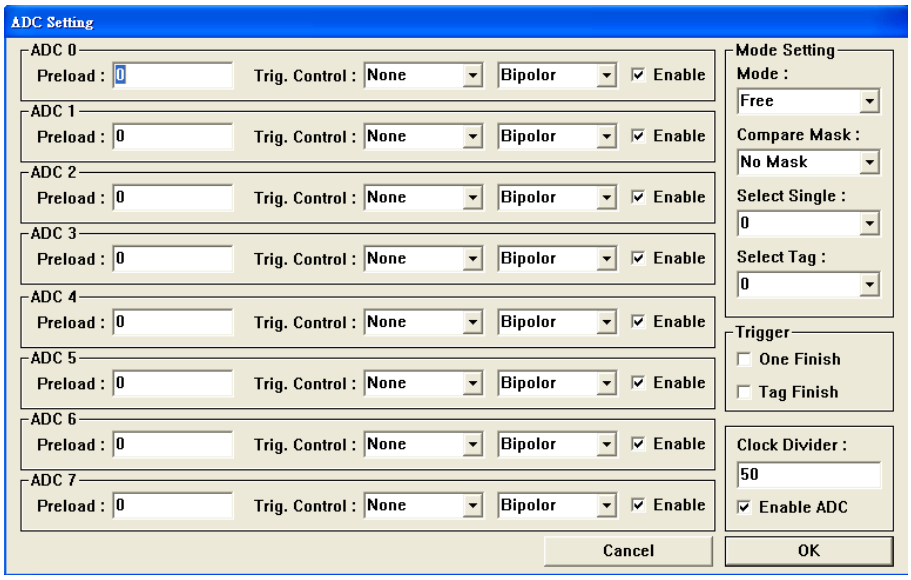

Figure 4-10.1

(2) Set ADC interrupt (as shown in Figure 4-10.1)

Apart from comparator interrupt, the ADC can be set to trigger other types of interrupt, i.e., to have one of the following trigger sources:

■ Conversion complete interrupt: 1

When one of the ADC inputs has completed conversion, an interrupt is generated to trigger a user-defined interrupt service routine.

➔Check the "One Finish" box in the Trigger section of the ADC Setting window.

■ Tag channel conversion completion interrupt: 1

Set one input channel as "tag" channel. When the channel has completed conversion, it generates an interrupt.

➔In the "Select Tag" list in the ADC Setting window (Refer to Figure 4-10.1), select one of channels 0 ~ 7 as the tag channel.

➔Check the "Tag Finish" box in the Trigger section of the ADC Setting window.

(3) Set mode selection

"Compare Mask" list and "Select Tag" list have been described in 4.10(1) and 4.10(2), respectively. "Select Single" list and "Mode" list will be described as follows:

ADC voltage input conversion can be divided into the following two types:

■ Single mode:

A: The only specified one of the eight ADC channels is enabled; the others are disabled. Conversion will not occur again once it is finished.

B: Data Update Time (including data transmission time) = *(1 /Serial_Clock) X 20*

Note: For Serial Clock, please refer to 4.10(5).

C: Single mode software setting

(1) Select Single

➔In the "Mode" list in the ADC Setting window, select Single rather than Free.

(2) Select a channel

➔In the "Select Single" list in the ADC Setting window, select one of channels 0 ~ 7.

(3) Enable ADC

➔In the ADC Setting window (refer to Figure 4-10.2), check the "Enable" box of the ADC channel selected in the previous step. In addition, the "Enable ADC" box must be checked.

(4) Adjust the data update time, if necessary.

➔Fill in the “Clock Divider” field in the ADC Setting window (Refer to Figure 4-10.2) to set Serial Clock to adjust the data update time.

(5) Execute

➔Click Run.

■ Free mode:

A: Only the specified ones of the eight ADC channels are enabled; the others are disabled. Conversion will only be switched among the enabled channels.

B: *Data Update Time* (including data transmission time) = *(1 / Serial_Clock) × 20 × (Number of Channels Enabled)*

C: Free mode software setting

(1) Select Free

➔In the “Mode” field in the ADC Setting window, select Free rather than Single.

(2) Enable ADC

➔In the ADC Setting window (refer to Figure 4-10.2), check the “Enable” boxes of the ADC channels selected in the previous step. In addition, the “Enable ADC” box must be checked.

(3) Adjust the data update time, if necessary.

➔Fill in the “Clock Divider” field in the ADC Setting window (refer to Figure 4-10.2) to set a proper value to adjust the data update time.

(4) Execute

➔Click Run.

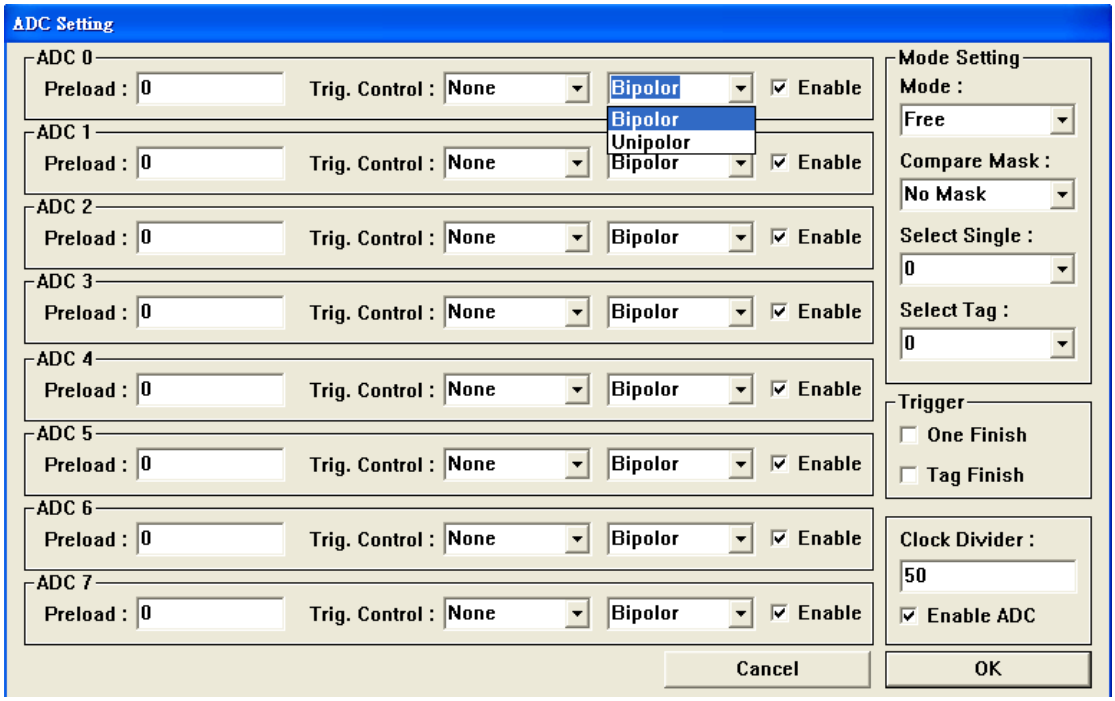
ADC Setting
ADC 0
Preload : 0
Trig. Control : None
Bipolor
Enable
Bipolor
Unipolor
ADC 1
Preload : 0
Trig. Control : None
Bipolor
Enable
ADC 2
Preload : 0
Trig. Control : None
Bipolor
Enable
ADC 3
Preload : 0
Trig. Control : None
Bipolor
Enable
ADC 4
Preload : 0
Trig. Control : None
Bipolor
Enable
ADC 5
Preload : 0
Trig. Control : None
Bipolor
Enable
ADC 6
Preload : 0
Trig. Control : None
Bipolor
Enable
ADC 7
Preload : 0
Trig. Control : None
Bipolor
Enable
Cancel
Mode Setting
Mode :
Free
Compare Mask :
No Mask
Select Single :
0
Select Tag :
0
Trigger
One Finish
Tag Finish
Clock Divider :
50
Enable ADC
OK

Figure 4-10.2

(4) Set bipolar/unipolar mode (as shown in Figure 4-10.2)

There are two ADC voltage input ranges:

Bipolar mode: -10V ~ 10V

Unipolar mode: 0V ~ 20V

Description: Please select a mode according to the input voltage range and assign the mode to both hardware (circuit board) and software. For hardware setting, refer to EPCIO-400X and EPCIO-600X User Manuals. Software setting is carried out as follows:

➔Select a proper mode in the bi/unipolar selection field in the ADC Setting windows.

(Note: This must be set if it is desired to use any ADC function. Besides, the same mode must be selected for all the eight channels.)

(5) Set ADC clock divider (as shown in Figure 4-10.2)

This is to set the ADC Serial Clock. This must be set if it is desired to use any ADC function. To make the setting, set a value N in the “Clock Divider” field in the ADC Setting window. The relationship between N and Serial Clock is as follows:

$$\textit{Serial Clock} = \textit{System Clock} / 4(N+1)\ ,\quad N=0 \sim 255\ (\textit{default: } N=0) \qquad (14)$$

By setting Serial Clock, the ADC’s data update time in the “Single” mode and in the “Free” mode will be changed (refer to Section 4.10(3)).

(6) Set start/stop ADC (as shown in Figure 4-10.2)

1. The “Enable ADC” box is the main switch of ADC. All the ADC functions will be disabled if the “Enable ADC” box is not checked.
2. If it is desired to enable analog-to-digital conversion in certain channels, the “Enable ADC” box must be checked in addition to the “Enable” boxes of the ADC channels to be enabled.
3. The “Enable ADC” box must be checked if it is desired to use any ADC function.