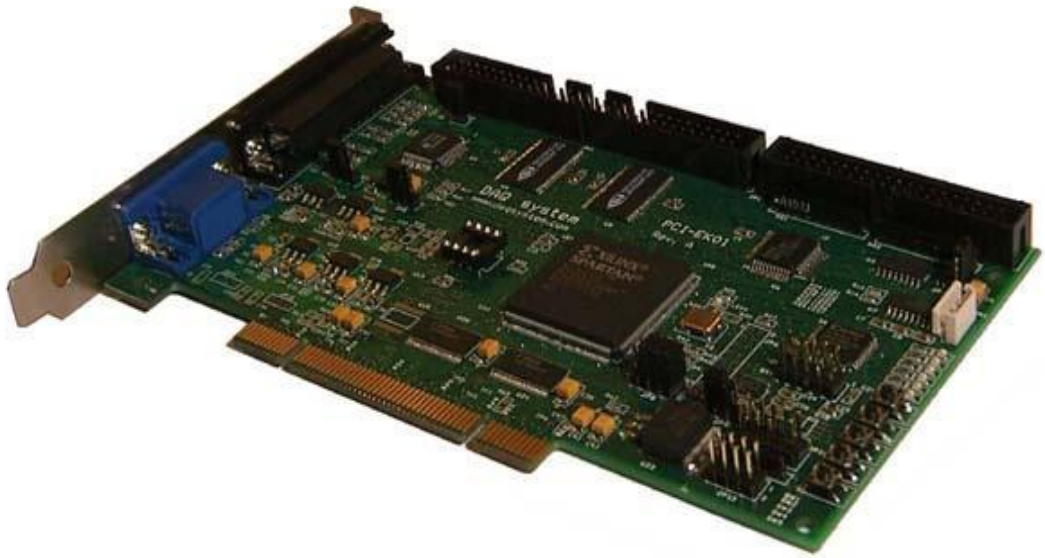
PCI-EK01

User Manual

Version 1.0



DAQ SYSTEM
Measurement & Automation

Contents

1. Introduction

Since the PC (Personal computer) was announced by IBM in 1981, the PC has become deeply embedded not only in industry but also in individual lives.

Perhaps, it is difficult today to imagine daily life without computers. The reason why PCs were able to develop so rapidly is because many devices were designed and utilized for each purpose by making standardized interfaces and architectures available to the public.

In early PCs, sufficient performance and purpose could be achieved just by using ISA-type devices. However, as the number of general users who want data speed, better scalability, and easy use increased, new interfaces (connections) were required, and standard interface methods such as USB and PCI were established. PCI (Peripheral Component Interconnect) is designed to be suitable for processing large amounts of information using a parallel interface, while USB (Universal Serial Bus) uses a serial connection for simpler connection and easier mobility. Both of the above methods use the Plug and Play method so that the user does not need to set up the device. (In the case of the old ISA board, the user directly set memory, I/O, interrupts, DMA, etc. using jumpers and switches. The problem with this method was that the user had to be fully aware of the occurrence of conflicts between boards and resolve it.).

In particular, in the case of USB, it is hot-pluggable (attaching the device while power is supplied), so it can be connected and used while the equipment (PC) is in operation, and in the case of PCI, it is a Plug & Play method that allows the board to meet the requirements of the system. Depending on the content, system resources are automatically allocated and used. Additionally, it has an incomparably faster data transmission speed than ISA, allowing large amounts of data to be transmitted at high speed.

In developing these PCI devices, the development period can be shortened by providing a board that developers can easily use and expand easily, and various functions such as Analog Input/Analog Output/Digital Input/High current drive are provided. The board was produced with a focus on enabling application in the field.

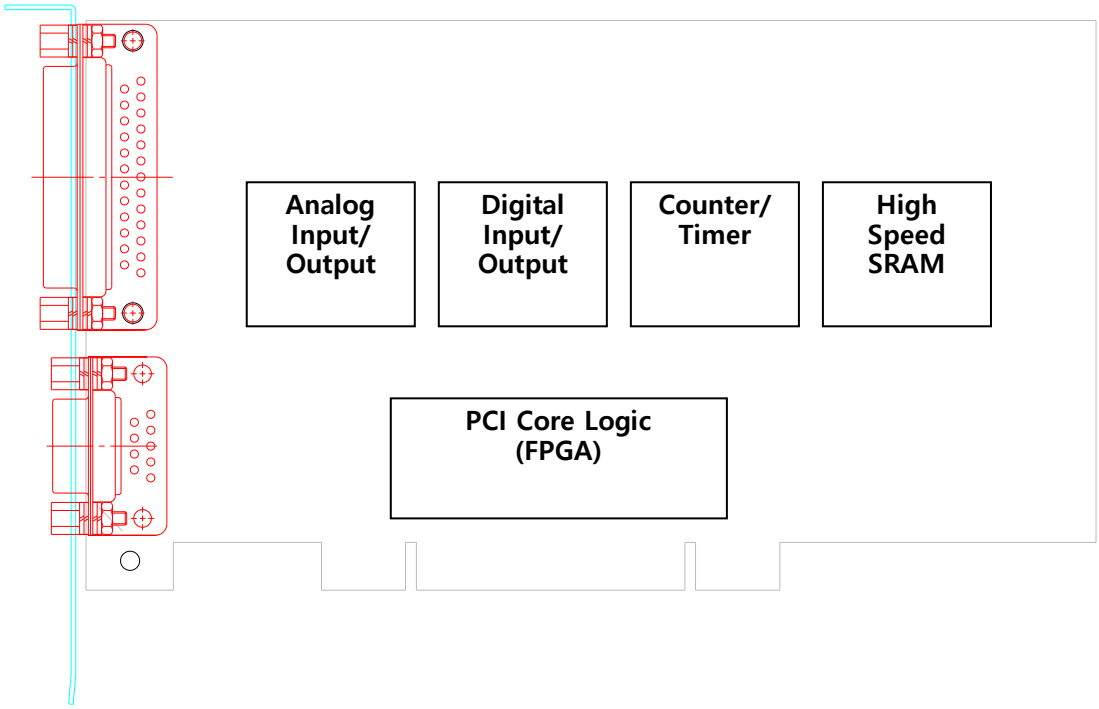
1-1 Product Features

Items	Description	Remark
Hardware		
PC Interface	PCI 32bit/33Mhz	
Operation Power	+5VDC/ Max 1A	
I/O Port	D-Sub9 / D-Sub25	
Feature	12bit 8 channel A/D input 12bit 4 channel D/A output 24 general purpose I/O 32bit Timer/Counter	8M bit(4M bit with Type A) High speed (12nSEC) SRAM
Analog Input	12bit resolution 8 Single ended or 4 differential On-board 1024 x 16 data FIFO On-board 512K(type B) x 16 data SRAM	0 to +3.3V, ±1.65V input range MAX 200Ksps(5uSEC) conversion time ±1 (LSB) INL/DNL ±1uA analog input leakage current 20pF analog input capacitance User can select ADC data storage, FIFO or SRAM
Analog Output	12bit resolution 4 channel output 0 to +3.3V output range MAX 1M (1uSEC) update rate On-board 1024 x 16 waveform generation dual-port RAM	Simultaneous update of outputs ±16 (LSB) INL ±1 (LSB) DNL ±3 (LSB) Offset error ±1 (LSB) Gain error Slew Rate 0.7V/usec
Digital In/Out	On-board 82C55 chip 24bit general purpose I/O Three 8bit group(Port A/B/C)	Port B has high current sink capability (Max. 500mA) 3.3V CMOS logic level Power on floating or logic low
Operating temperature range	0 ~ 70℃	
Storage temperature range	-20 ~ 80℃	
Humidity range	20 ~ 80%	Non-condensing
Board size	175mm X 95mm	PCB Board Size
Software		
OS	Windows 2000/XP/7/8/10 (32/64bit)	
API	Windows Client DLL API	
Support	Sample Program	VC++

2. PCI-EK01 Function

The PCI-EK01 board is a board for design and logic development for PCI master. In addition, it is a board that can be used for multi-functional applications such as analog-to-digital converter (ADC), digital-to-analog converter (DAC), wave generator, high-power driver, and fast-moving data.

As shown in the picture below, PCI-EK01 is equipped with functions so that it can be applied to various tests and various application fields.



[Figure 2-1. PCI-EK01 Functional Blocks]

2-1 Product Application

- ◆ PCI development and evaluation
- ◆ Data acquisition
- ◆ Laboratory instrumentation
- ◆ Process control systems
- ◆ Factory automation

3. Installation

Prior to installation, check whether the packaging contents are intact as shown in the picture below.

3-1 Product contents

[Figure 3-1. PCI-EK01 Product contents]

Product contents

1. PCI-EK01 Board
2. Windows2000/XP Driver (CDROM)
3. Application sample source (CDROM)
4. Circuit / Parts list, etc (CDROM)

3-2 Hardware Installation

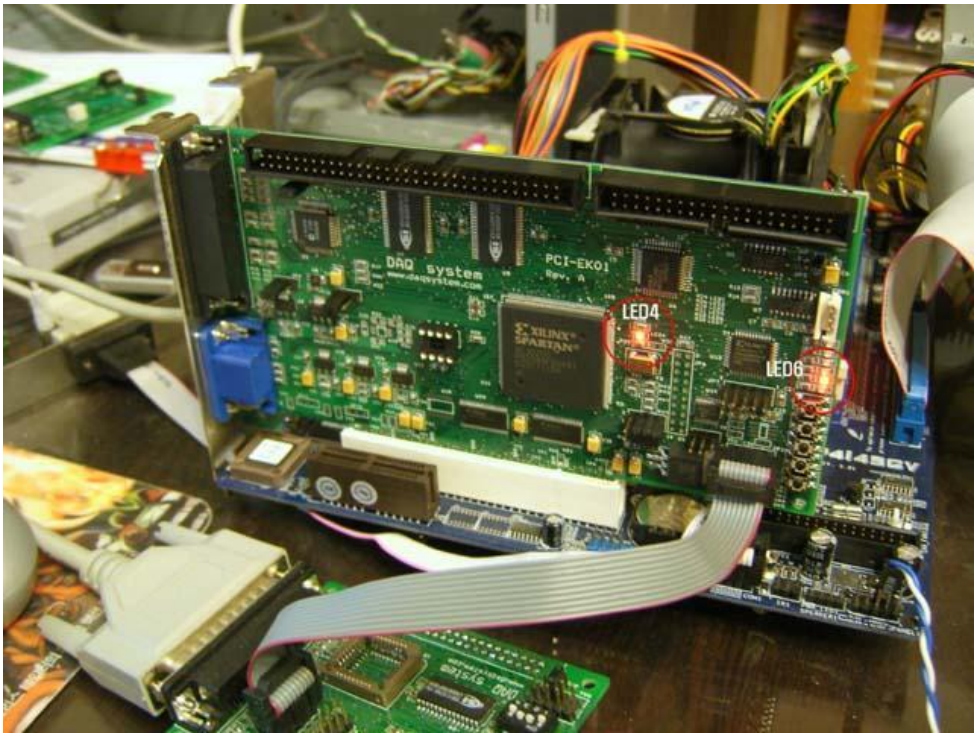
To install the board on a PC, follow the steps below. In the case of the PCI board, there is no special jumper to set for board installation as it is a Plug & Play device.

(1) First, turn off the computer.

(2) Take off the computer case and insert the board into the empty PCI slot. At this time, if there are several empty slots, it is better to use the slot closest to the CPU.

(3) Secure the board bracket to the computer using screws and cover the case.

(4) Once case assembly is complete, turn on the power switch.



[Figuure 3-2. Screen with PCI-EK01 mounted on PC]

(Note) The picture above shows the board mounted without a PC case to help users understand. Typically, a PC main board is installed and used inside a case.

(5) When the power is turned on, two LEDs light up as shown in [Figure 3-2]. In the case of LED4, it lights up when the board configuration is completed normally, and in the case of LED6, it blinks at approximately 1 second intervals to check if there is an error in the internal clock.

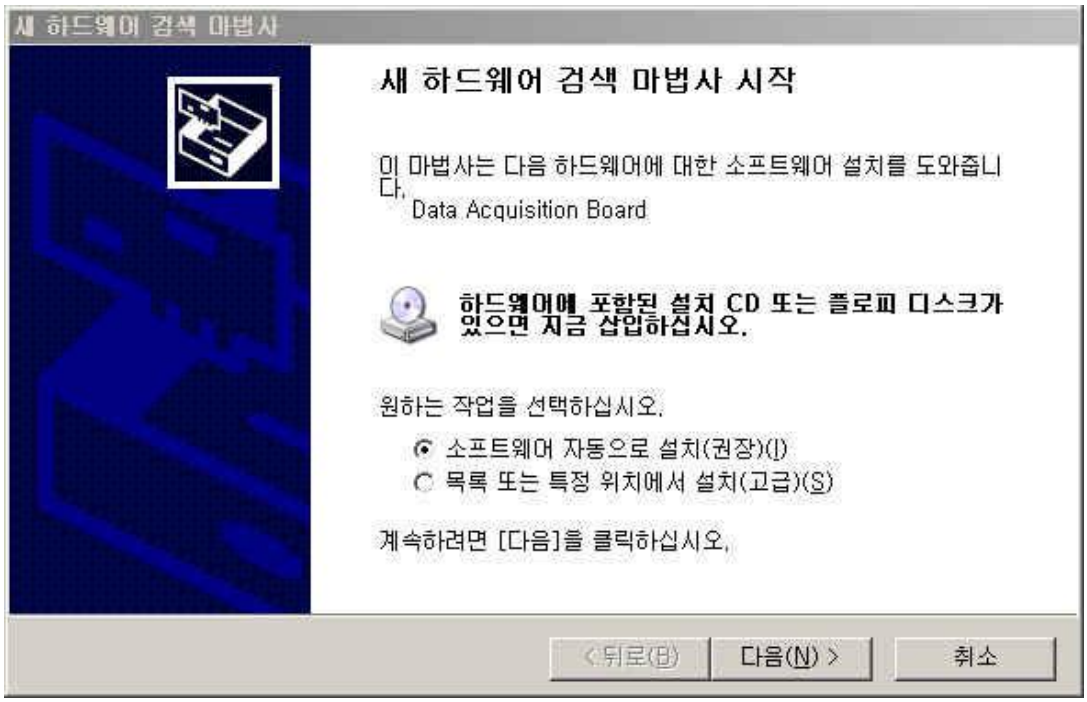
(6) LED1 to LED5 indicate the operating status of the PCI bus, so they blink according to the operating status of the PCI bus.

3-3 Driver Installation

Once board installation is complete, install drivers and sample applications to run the board on the PC. For installation, use the CD provided.

The installation order is as follows, and unless otherwise specified, the explanation is based on Windows XP.

(1) When the computer completes booting, it will search for newly installed devices. If a new device is discovered, the operating system (Windows XP) asks you to install the driver for the device.

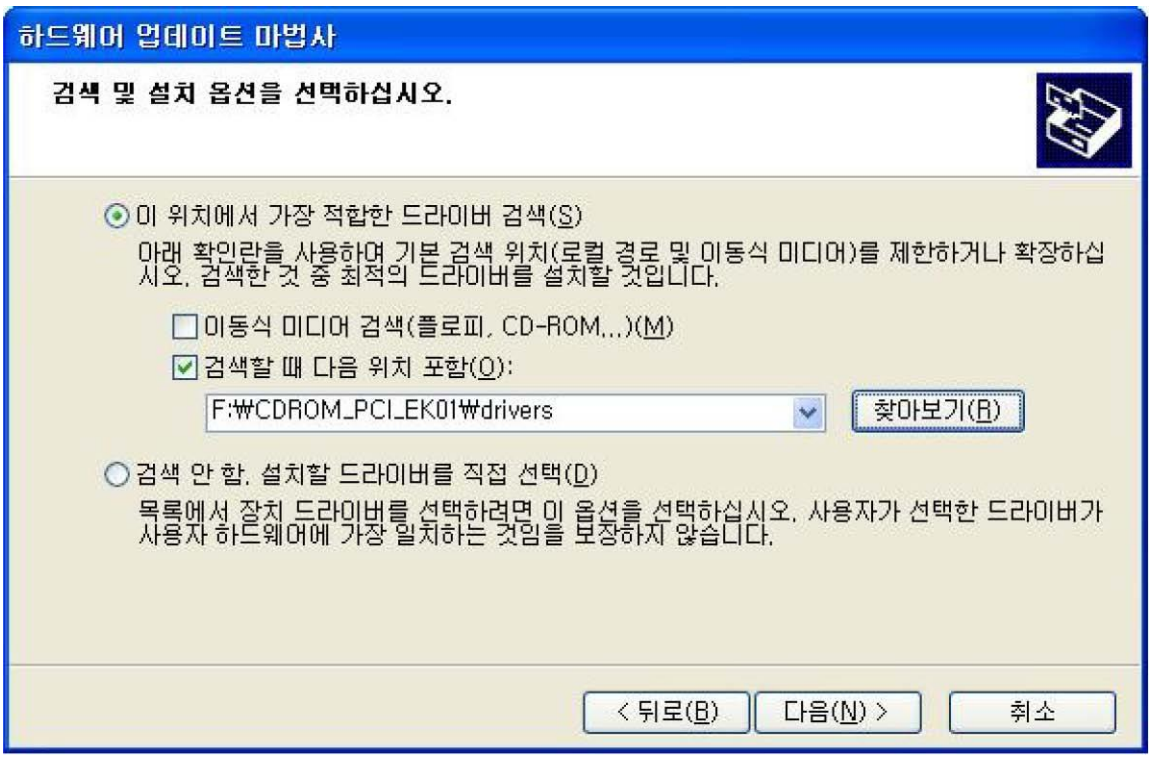


[Figure 3-3. PCI-EK01 device discovery screen]

(2) If you click the Next button to install the driver in the picture above, the driver search screen appears. On the [Figure 3-4] screen, specify the Driver folder of the CD that contains the driver. (Select Install from a list or specific location)

Ex) F:₩CDROM_PCI_EK01₩drivers

The driver folder contains **"pci_ek01.inf"** and **"pci_ek01.sys"** files required for driver installation.



[Figure 3-4. PCI-EK01 device driver search screen]

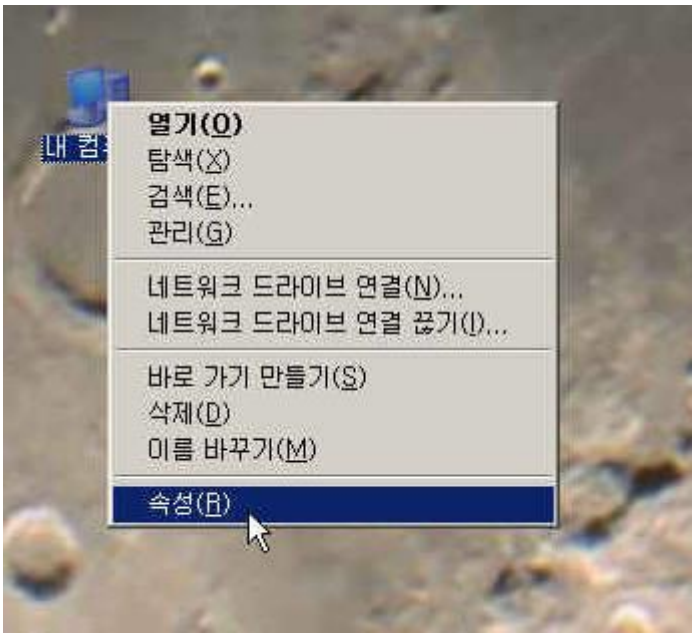
(3) When installation is completed normally, it looks like the picture below.

[Figure 3-5. PCI-EK01 Device driver installation completion screen]

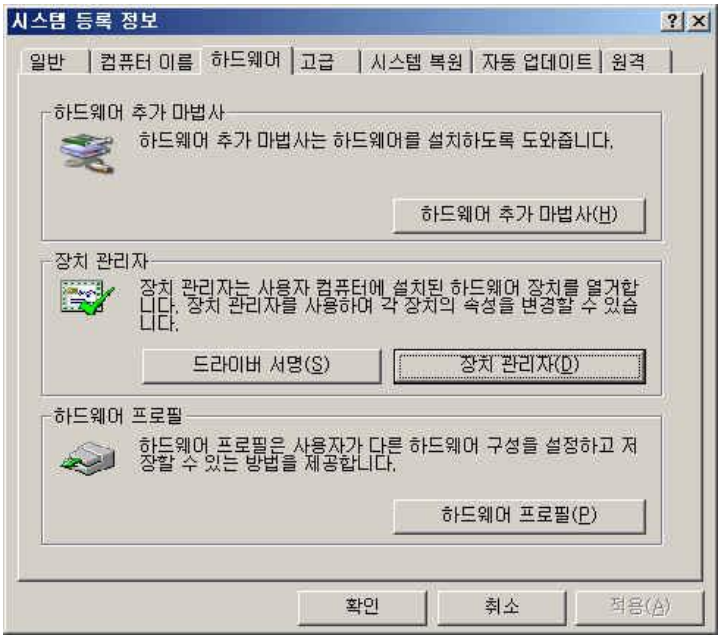
(4) Once installation is complete, you can use the PCI-EK01 board right away. Before using, check again whether the driver has been installed properly using the following method.

My Computer -> Properties -> Hardware -> Device Manager screen

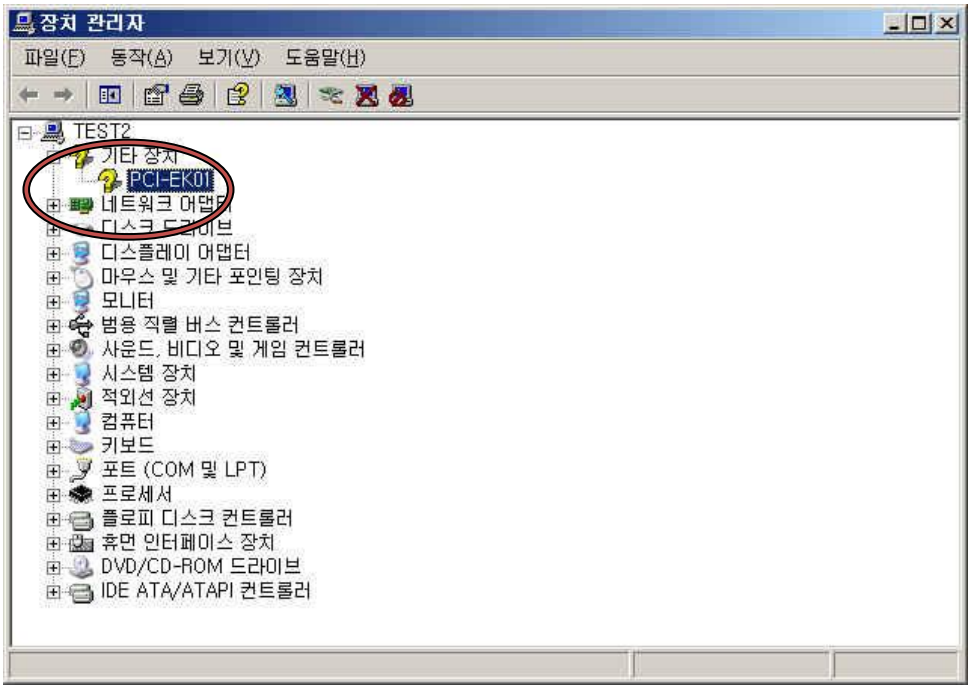
Other devices -> Check **whether PCI-EK01** is installed. If it appears as shown in the picture below, the installation has been completed successfully.



[Figure 3-6. "My Computer" properties execution screen]



[Figure 3-7. System Properties Window]



[Figure 3-8. Device manager execution screen]

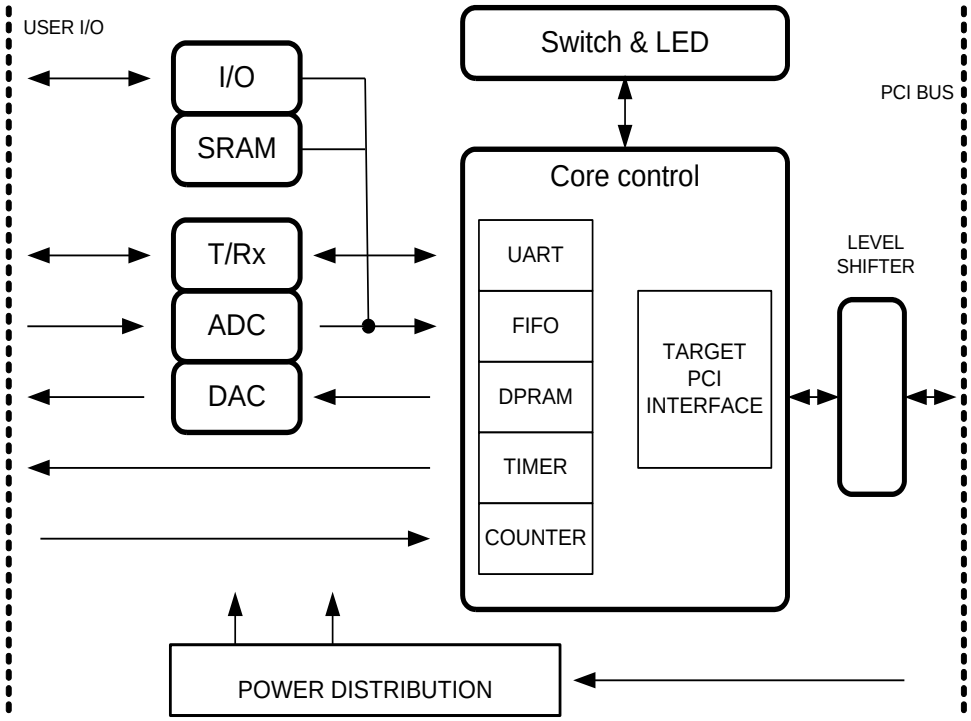
The picture above shows a screen where the PCI-EK01 board is normally installed on the PC. (Check the red circle)

(Note) After initial installation, it is recommended to reboot the PC for normal operation.

4. PCI-EK01 Board Description

Briefly explain the board functions. For detailed function specifications, please refer to the specifications section.

4-1 PCI-EK01 Concept



[Figure 4-1. PCI-EK01 Functional Block Diagram]

Looking at the picture above, PCI-EK01 includes several functions and is configured to enable various tests using the board. A description of each component function is as follows.

(1) **Core Control**

The PCI-EK01 composes a PCI Target, external ADC, DAC interface and FIFO/TIMER/Counter with using a FPGA (Spartan3). It is not used a UART function, it will be upgrade later.

(2) **Level Shifter**

A Level shifter exchanges a logic level of PCI to a logic level of FPGA. So, it is handily usable just place to a slot without jumper setup of PCI 3.3V system or PCI 5V bus system.

(3) **Power Distribution**

It gets from 5V power at PCI bus and it supply as making the necessary 1.2V, 2.5V, 3V, 3.3V power to use at board.

(4) **Switch & LED**

There is the LED which can confirm action state of a board. There are 5 switches 매 that a user test it to a manual.

(5) **I/O**

It can interface with the outside by 24bit I/O as it equip 82C55. It is a 3.3V logic level and 8bits of these 24bits can drive 500mA.

(6) **SRAM**

It is used to high speed data transfer (Max 33Mhz) and ADC's data buffer as it equip maximum 8Mbit(1M Byte) high speed RAM.

(7) **T/Rx**

It is a RS232C Transceiver. It can use it to connect at PC or device.

(8) **ADC**

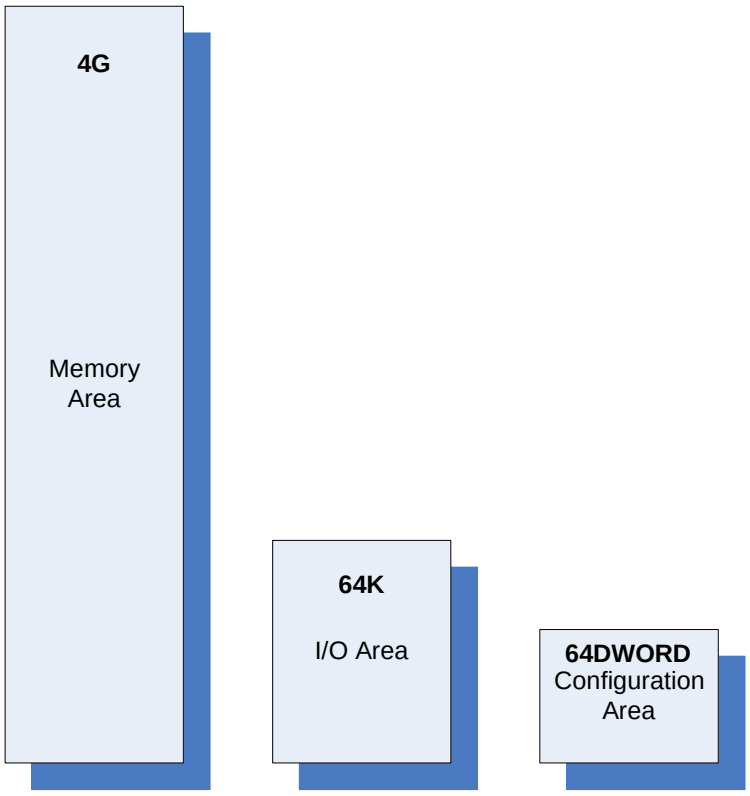
It's an Analog to Digital Converter. It can use the methods of 8-Ch Single Ended or 4-Ch Differential, can make sampling 200K(5uSEC) per second.

(9) **DAC**

It's 4-Ch Digital to Analog Converter, it can update a DAC's value with maximum 1M (1uSEC).

4-2 Address Map

The IBM PC we mainly use uses an x86 series CPU, so it can be broadly divided into memory and I/O areas. However, in the case of the PCI bus, a separate configuration area is created to support Plug & Play so that the device can have the necessary resources and device status control registers.



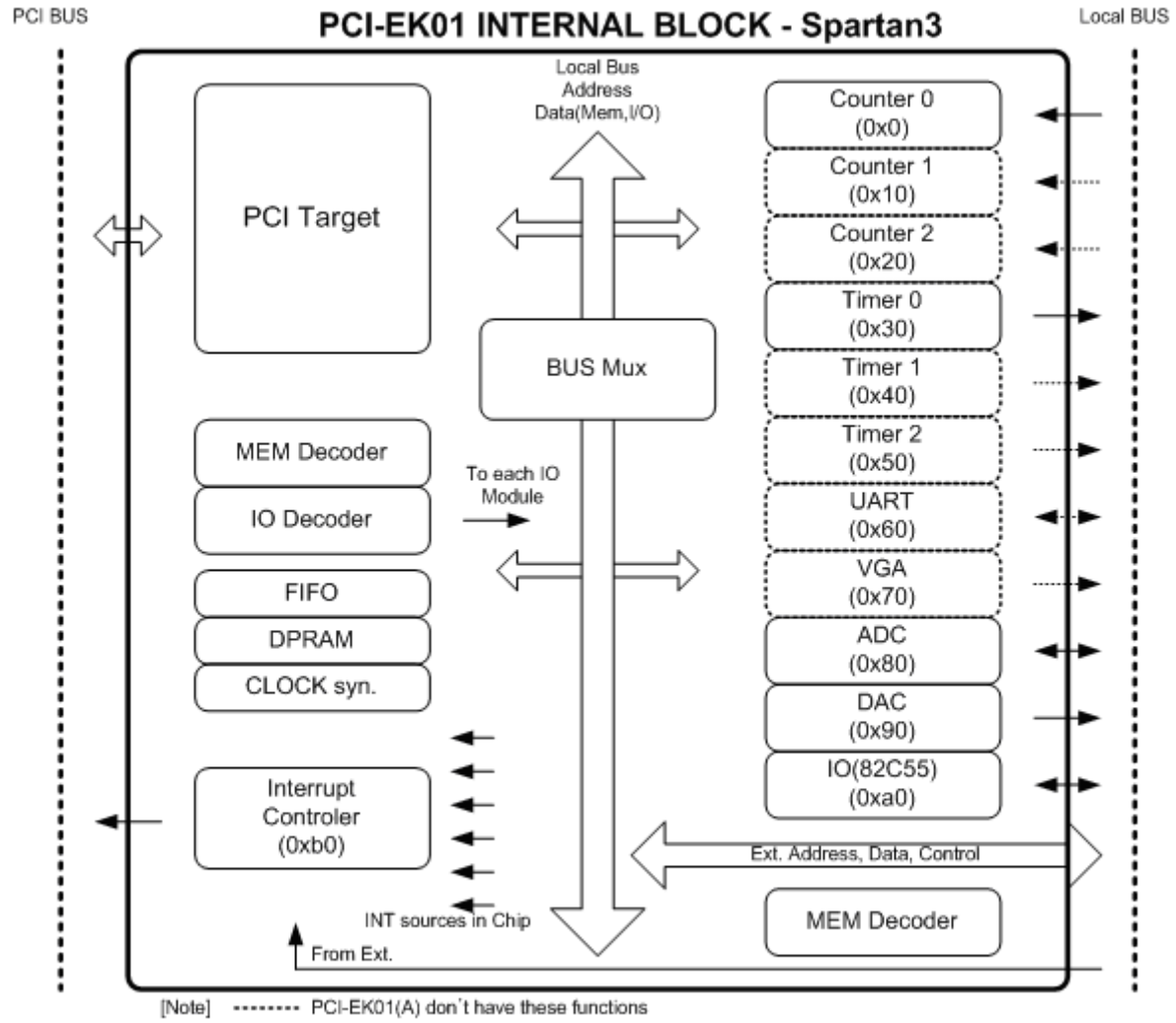
[Figure 4-2. Address area of IBM PC series]

In the case of PCI-EK01, the memory and I/O required for operation are allocated and used by the system, and the requirements are as follows.

Memory Max. 64MByte

I/O 256Byte

The allocated address area is divided and used within PCI-EK01 as shown in [Figure 4-3]. The control and status registers of all peripheral devices are located in the I/O area, and only high-speed SRAM is located in the memory area.



[Figure 4-3. PCI-EK01 Address Map]

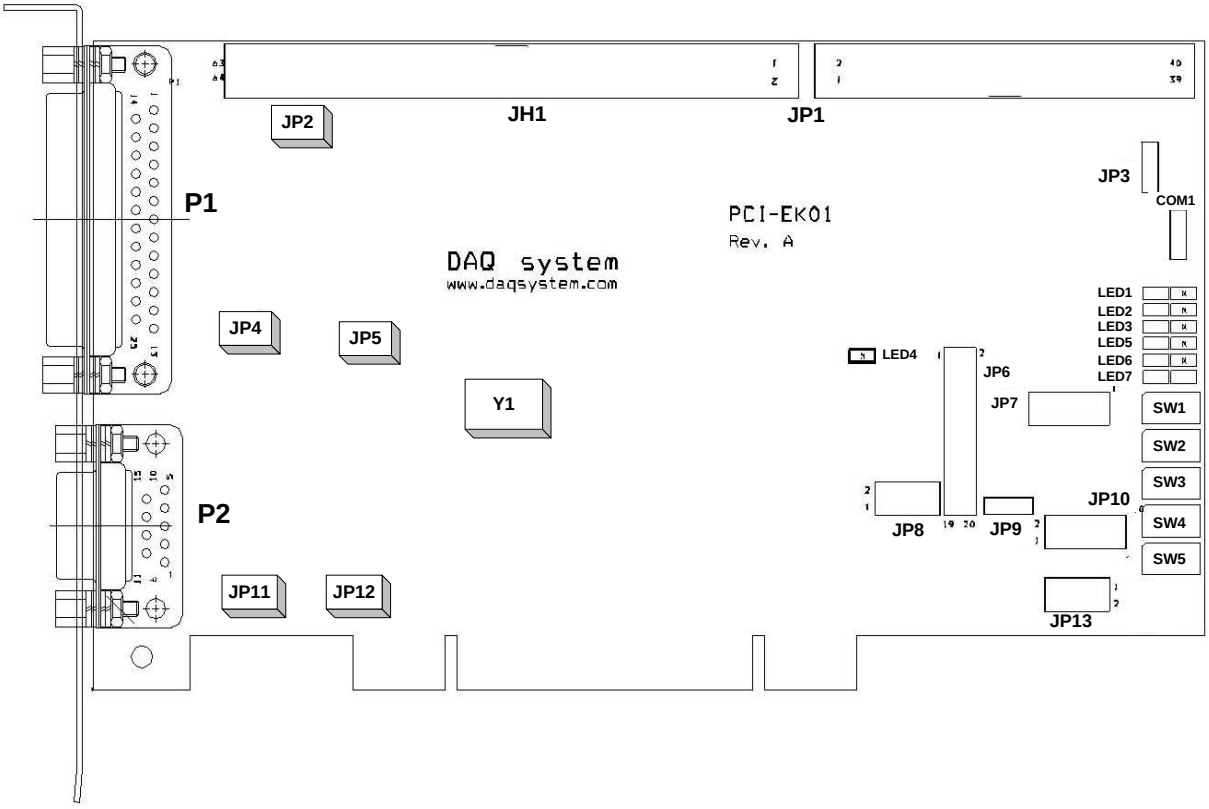
(Note) 1. The functions indicated by the solid line are not implemented in the PCI-EK01(A) model.

2. In the case of UART, the function is not currently implemented, and functions will be added in the future.

3. In the case of VGA, the function is not currently implemented, and functions will be added in the future.

4-3 Jumper Option and Connector Pin-out

PCI-EK01 is equipped with several jumpers and connectors. Each jumper and connector is used to expand or select functions, and it is recommended that the basic settings be used as set when the product is shipped. However, in order to perform special tests, the user must change the settings and connections, which will be explained here.



[Figure 4-4. PCI-EK01 External Connectors and Jumpers]

[Table 1. Connector and Jumper List]

No.	Name	Description	Remark
1	**P1**	Analog Input/Output, Counter/Timer	
2	**P2**	VGA video signal	Future Upgrade
3	**JH1**	Local Address & Data Bus	
4	**JP1**	Digital Input/Output	
5	**JP2**	ADC Reference Voltage	Close
6	**JP3**	Select Digital Power	3.3V or 5V
7	**JP4**	DAC Power Selection	AVCC or DVCC
8	**JP5**	DAC Reference Voltage Selection	ADC reference out or DAC Power
9	**JP6**	FPGA Parallel configuration	Unused
10	**JP7**	EPLD(XC9536XL) Program JTAG	

11	**JP8**	FPGA configuration Mode Selecion	
12	**JP9**	Serial Flash Mode Selection	Future Upgrade
13	**JP10**	FPGA & Serial Flash Program JTAG	
14	**JP11**	+/- 12V Power	Unused
15	**JP12**	PCI I/O Power	Unused
16	**JP13**	Connected to custom jumper Tact switch	
17	**LED1-3** **LED5**	For PCI Transition Monitor	
18	**LED4**	FPGA configuration Indicator	
19	**LED6**	Clock Indicator	
20	**LED7**	PCI Target select	
21	**SW1-3**	User defined	Unused
22	**SW4**	FPGA Re-configuration	
23	**SW5**	User defined	Unused
24	**COM1**	RS232C	Future Upgrade
25	**Y1**	The user-defined Clock Timer/Counter function allows users to Clock is available for use.	Future Upgrade

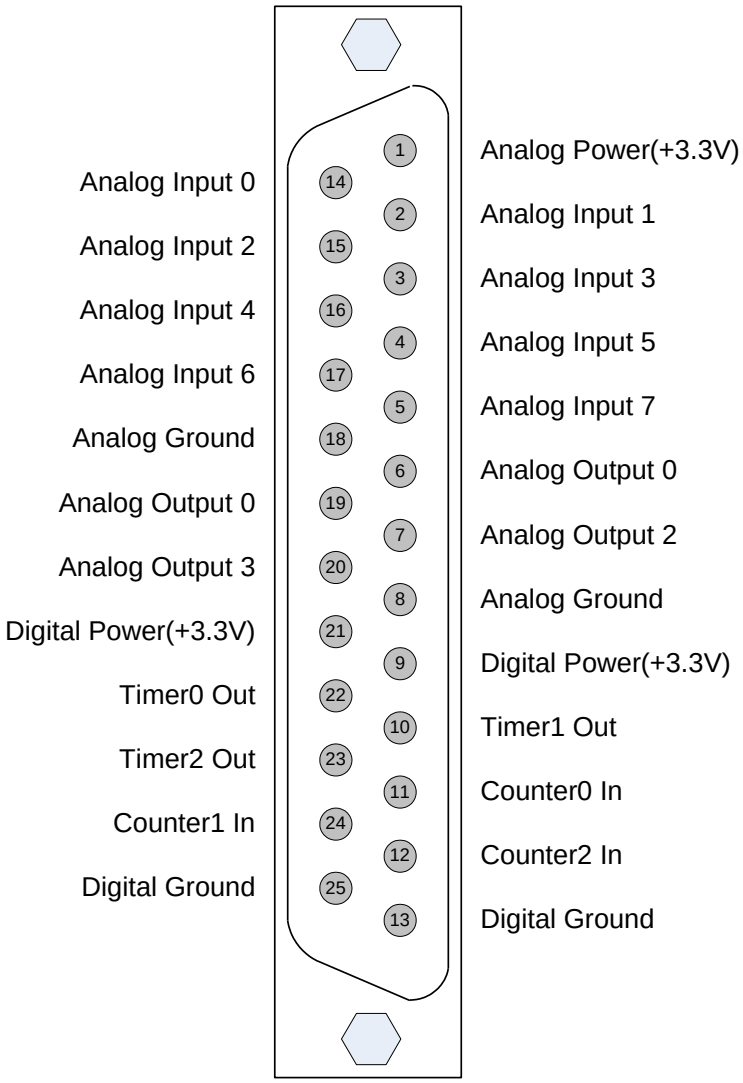
The basic explanation is as in [Table 1]. Detailed descriptions of each connector and jumper continue. Unless otherwise explained, please refer to the PCI-EK01 drawing.

Unless otherwise stated, the explanation is based on PCI-EK01(B).

4-3-1 P1 Connector

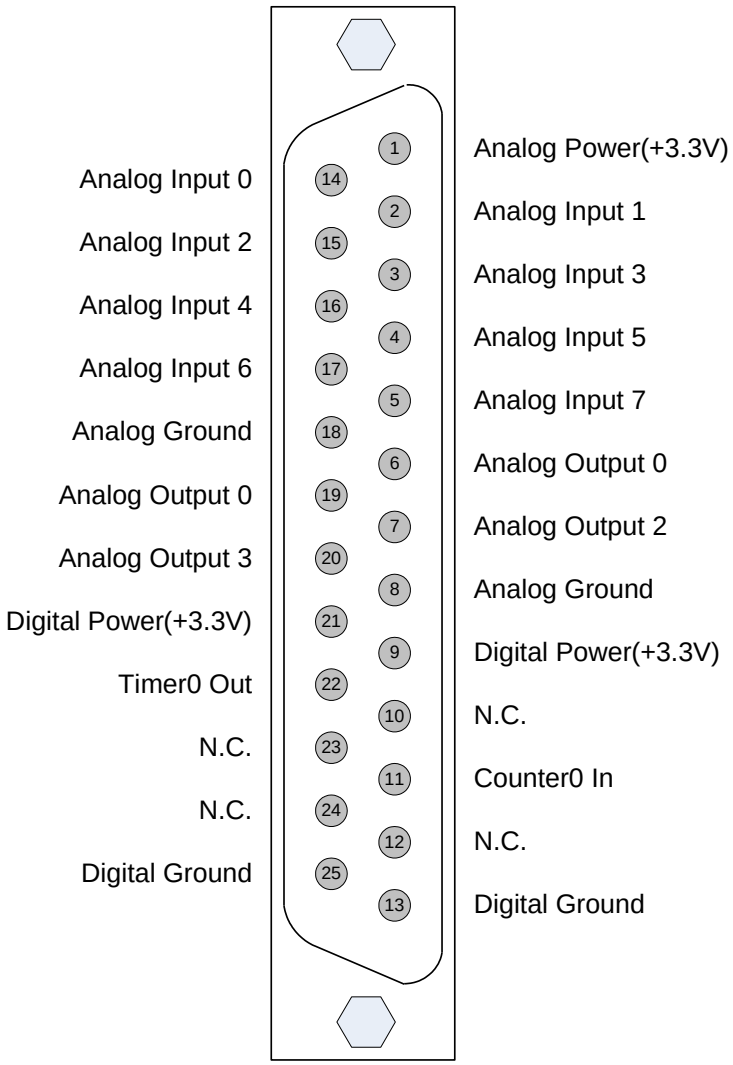
Analog input and output, Timer output, and Counter input are connected to the DSUB-25PIN PLUG connector. Please refer to the Specification section for the analog input/output range of ADC and DAC, and the input/output level of Timer and Counter is 3.3V CMOS logic level.

One thing to note is that Analog Ground must be used when interfacing with external analog signals.



[Figure 4-5. PCI-EK01 P1 Connector Pin-out]

The picture below shows the connection details of the P1 connector of the PCI-EK01 (A) model. Compared to the (B) model, Timer 1/2 and Counter 1/2 are missing.

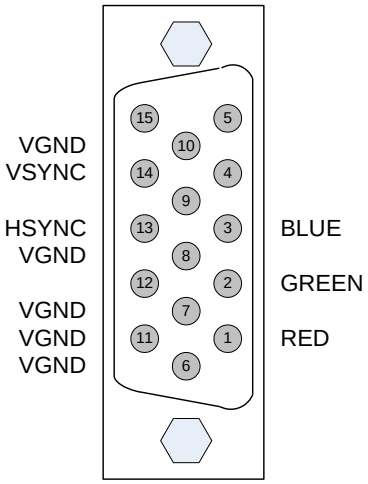


[Figure 4-6. PCI-EK01(A) P1 Connector Pin-out]

4-3-2 P2 Connector

VGA Video signal generation (not currently implemented, but for PCI-EK01(B) model, features will be added through future upgrades.

Note: For the PCI-EK01(A) model, there are no plans to add a VGA connector or other features in the future.

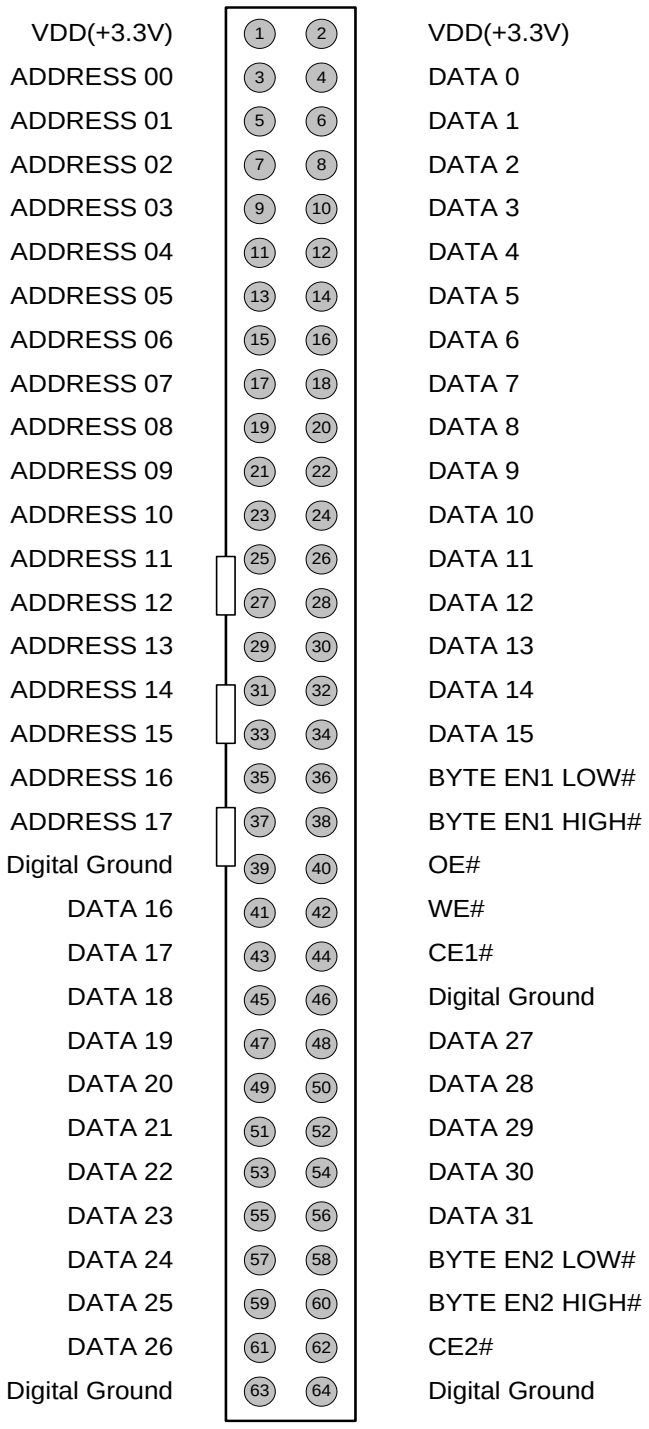


[Figure 4-7. PCI-EK01(B) P2 Connector Pin-out]

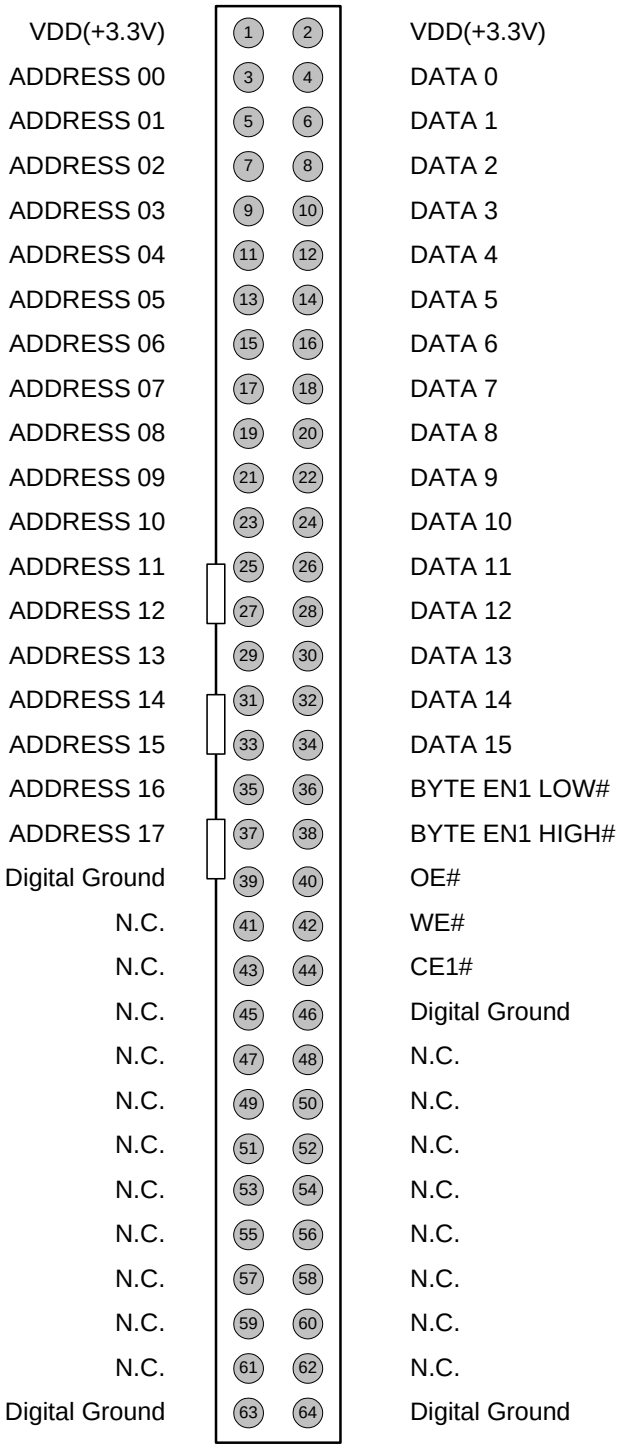
4-3-3 JH1 Connector

Local addresses and data signals on the board were prepared in connectors to facilitate external interfacing. Users will be able to add hardware functions using this connector signal.

(Note) In the case of the PCI-EK01(A) model, only 16Bit data is implemented.



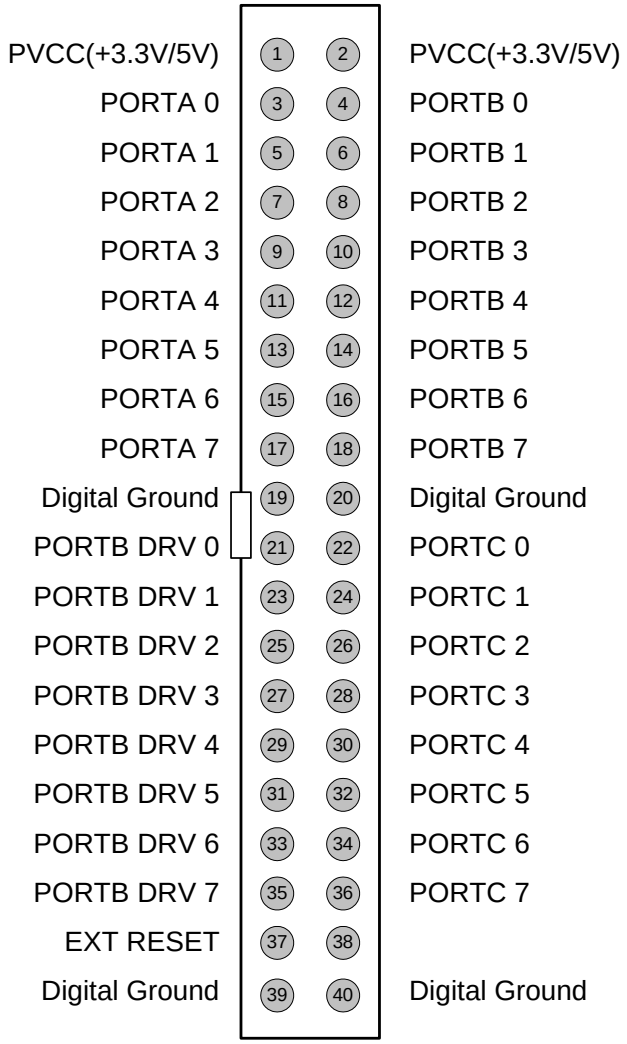
[Figure 4-8. PCI-EK01(B) JH1 Connector Pin-out]



[Figure 4-9. PCI-EK01(A) JH1 Connector Pin-out]

4-3-4 JP1 Connector

The I/O port of 82C55 is connected. In the case of 82C55, there are three 8-bit ports (A/B/C), which can be used as input or output depending on program settings. Looking at the connector picture, there are "PORTB DRV 0" to "PORTB DRV 7", which means connecting a driver that can flow a lot of current to port B and connecting its output to the connector.
(See drawing for details)



[Figure 4-10. PCI-EK01(B) JP1 Connector Pin-out]

4-3-5 JTAG and Other Connector

For JP7, it is equipped to program EPLD (Electrically Programmable Logic Device) XC9536XL, and for JP10, it is used to program Serial Flash and FPGA.

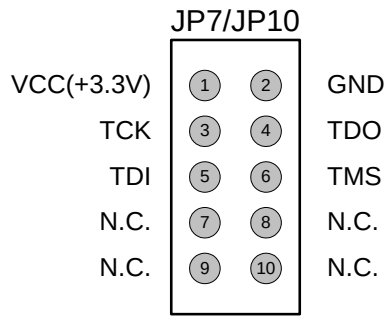
For JP8, set the program mode of the FPGA. The settings for each program mode are as follows.

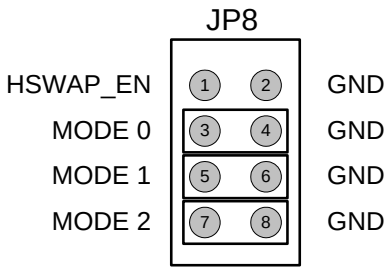
M2	M1	M0	Configuration Mode
0	0	0	Master Serial
1	1	1	Slave Serial
1	1	0	Master Parallel
0	1	1	Slave Parallel
1	0	1	JTAG

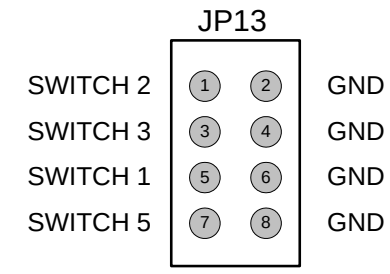
(Note) 1. When the jumper is connected, it becomes '0'.

2. HSWAP_EN has an internal pull-up, and when the jumper is connected, it becomes '0'. If it is '0', the I/O pin is full up during the configuration period.

In the case of JP13, it is connected in parallel with the switch connected to the EPLD, so it is used when the user wants to manually test a special function. (refer to circuit diagram)



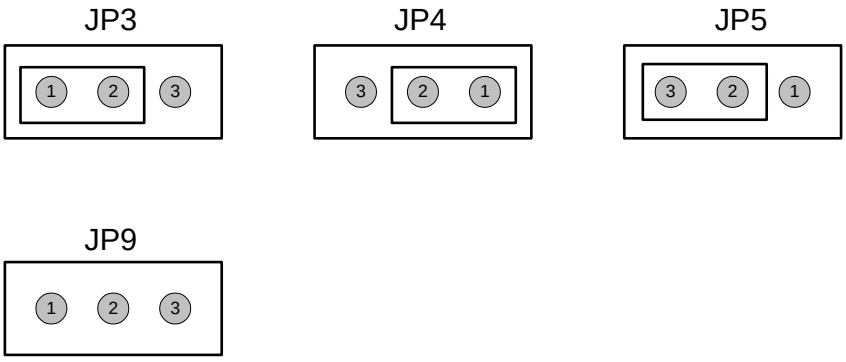




4-3-6 Default Jumper Setup

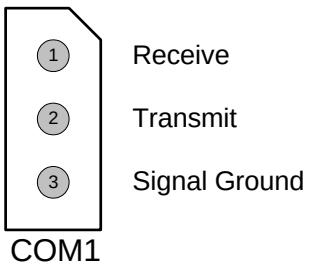
JP3 selects the power source of the external I/O connector (JP1). For the 1-2 connection, +3.3V is output, and for the 2-3 connection, +5V is output.

JP4 allows the power of the DAC chip to be selected as DVCC or AVCC. It is best to use the remaining jumpers as default settings by referring to the circuit diagram.



4-3-7 RS232C

It has been prepared as a reserve so that UART functions can be added in the future. The PCI-EK01 board is equipped with an RS232C Transceiver, which is connected to the COM1 connector and FPGA.



4-4 CDROM Folder Description

The CDROM provided with the board contains sample sources needed to install the board on a PC and write the programs the user needs.

First, the "drivers" folder contains drivers and installation files, and the "manual" folder contains user manuals/circuit diagrams/application notes, etc.

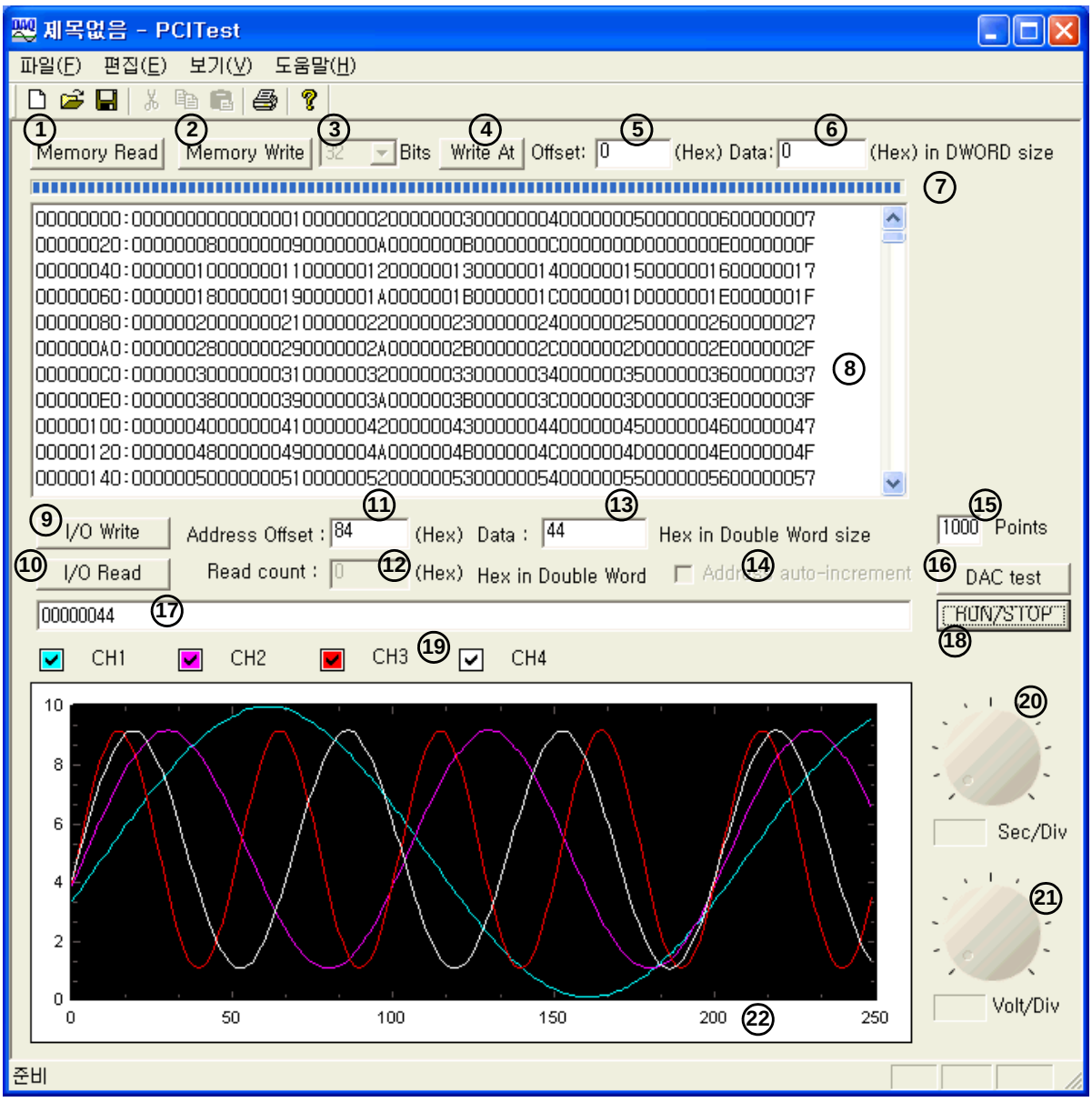
The "Sample" folder contains the source and executable files of the "PCITest" program used in this manual.

The "readme" folder contains the latest files and files to read before installation.

5. Test

Functional tests will be conducted in this chapter to determine whether there are any problems with the board and to learn how to use it. The test is run using the **"PCITest.exe"** program in the SAMPLE folder of the CDROM on a PC with the PCI-EK01 board installed.

The sample folder on the CDROM contains the executable file and source file of "PCITest." The executable file is used for testing, and the provided sample source file can be modified and used to implement the functions needed by the user.



[Figure 5-1. PCITest.exe Execution Screen]

The picture above is the execution screen of the "PCITest.exe" program. First, we will briefly explain how to use the program. A number is written for each control on the screen. The meaning and use of the controls for each number are as follows.

No.	Name	Description	Remark
1	**Memory read**	When you press the button, a total of 1 MByte from address 0 to 100000h of the memory area is read and displayed in area (8).	
2	**Memory Write**	When you press the button, records are sequentially recorded from 0 to 100000h in the memory area. Therefore, first perform Memory Write and then Memory Read to check whether accurate data is written and read.	
3	**Access Bits**	When reading or writing memory, set how many bytes will be processed.	Not currently applicable. Always processed as 32 bit.
4	**Write At**	The data value specified at a specific address in the memory area is recorded as a 32-bit word.	
5	**Memory Offset**	This is the offset value of the address when recording at a specific address in the memory area.	16bit Hexa-Decimal
6	**Memory Write Data**	Data value to be written to a specific address in the memory area. When you want to write to a specific address in the memory area, first set the Offset and Data values to the desired values and press the "Write At" button. To check whether it has been recorded accurately, press the "Memory read" button.	16bit Hexa-Decimal
7	**Progress Bar**	Displays the progress when reading memory.	
8	**Memory Data**	Data read from the memory area is displayed on the screen.	
9	**I/O Write**	When you press the button, data is recorded at a specific I/O address.	
10	**I/O Read**	When you press the button, data is read from a specific I/O address and displayed on the screen.	
11	**I/O Offset**	Set the Offset value of the address you want to read or write in the I/O area.	16bit Hexa-Decimal
12	**I/O Read Count**	Set the amount of I/O data to be read. (Note) Writing or reading I/O data is always 32-bit DWORD size.	Not currently applicable. One DWORD operation
13	**I/O Write Data**	Set the data to be recorded in the I/O area.	16bit Hexa-Decimal
14	**I/O Auto Increment**	Set whether to automatically increase the I/O address when reading or writing multiple I/O data.	Not currently applicable.

15	**DAC FIFO Write Points**	Set how many points will be recorded in the DAC FIFO for waveform generation. Minimum 1 point and maximum 1000 points	
16	**DAC FIFO Write**	Data is recorded in FIFO with the number of points set above. (Note) The recorded data is sine wave data.	
17	**I/O Read Data**	Displays data read from a specific address in the I/O area.	
18	**ADC Run/Stop**	When you press the button, the ADC value is read from the buffer periodically (0.5 seconds) and displayed on the screen or stopped.	
19	**ADC Channel Enable**	ADC values can display multiple channels on the screen. Set the channel you want to display. In the program, the value read from one channel is displayed in four channels.	
20	**Time Knob**	When displaying ADC values on the screen, you can set the time axis.	Not currently applicable.
21	**Voltage Knob**	When displaying ADC values on the screen, you can set the size (voltage) axis.	Not currently applicable.
22	**ADC Graph**	The read ADC data is displayed on the screen.	

(Note) 1. The PCITest program uses ActiveX to display the ADC Graph, so a separate OCX must be installed to run it. (Please refer to the CDROM for registration method)

2. When running the program in Windows 2000, "GDIPlus.dll" must be copied to the execution folder or Windows system folder. In Windows XP, there is no need to copy "GDIPlus.dll" because it is included in the operating system.
3. In order to compile the provided PCITest sample source, you must register the OCX described above and install the Windows platform SDK. The SDK can be obtained from the Microsoft website.
4. The program was compiled and tested in Visual C++ 6.0/Service Pack 5.
5. System recommendations

   To obtain more stable performance, it is recommended to use the following system.

   PC equipped with Pentium 3 or higher or equivalent CPU (no problem with operation even on Pentium 2)

   RAM 256M or more (no problem when tested with RAM 64M)

   Monitor 1024 x 768 or higher / Hard disk 10G or more

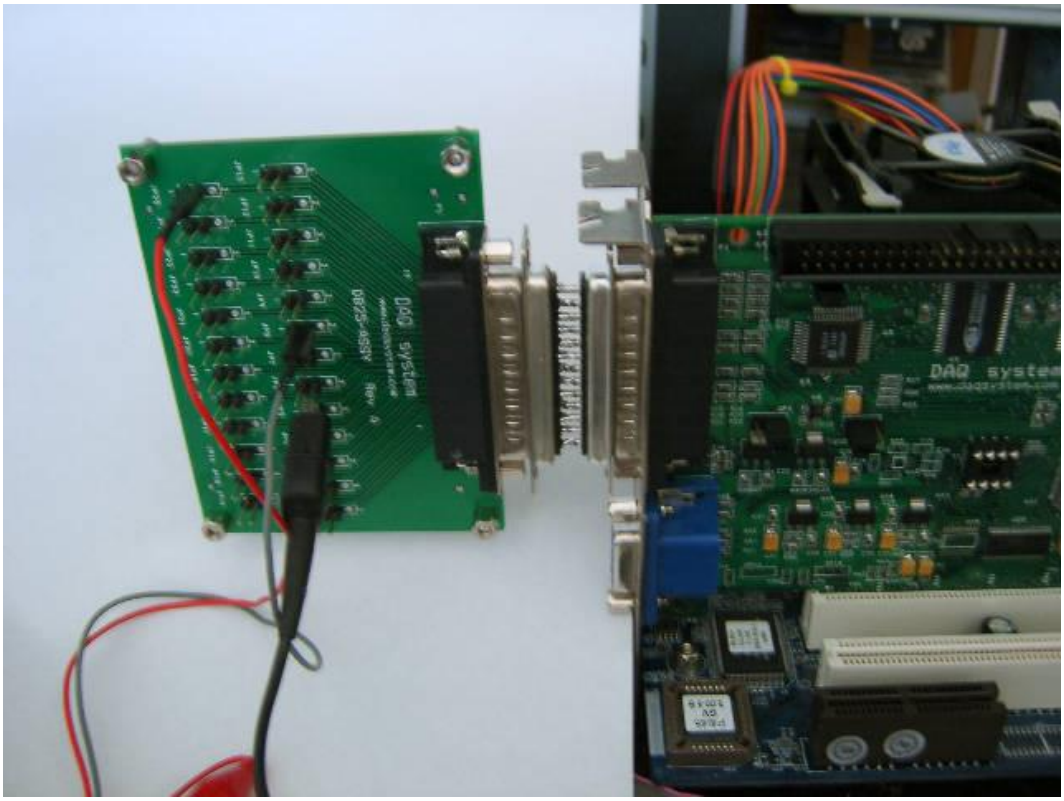
   Windows XP Professional or Windows 2000
6. For smooth testing, it is necessary to understand the meaning of each register by referring to AN203 (PCI-EK01-Register Level Application Guide).

5-1 Analog Output

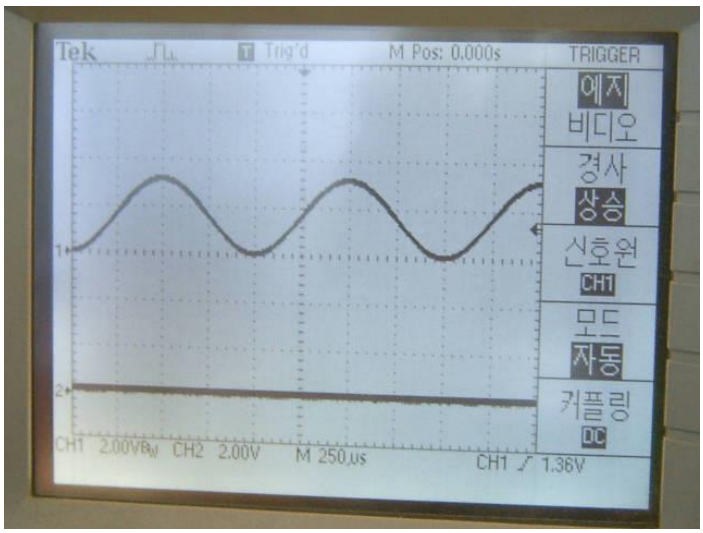
(1) Enter 1000 in **DAC FIFO Write Points** and press the **DAC test (DAC FIFO Write)** button. This records 1000 points of sinusoidal data to create a waveform in the DAC FIFO.
The above 1000 points are data from DAC channels 0 to 3.

(2) Enter "94" in I/O address offset and "1" in Data, then press **the I/O Write** button.

(3) Using an oscilloscope, check whether the sine wave output actually comes from the P1 (D-sub 25pin plug) connector as shown in [Figure 5-2] below.



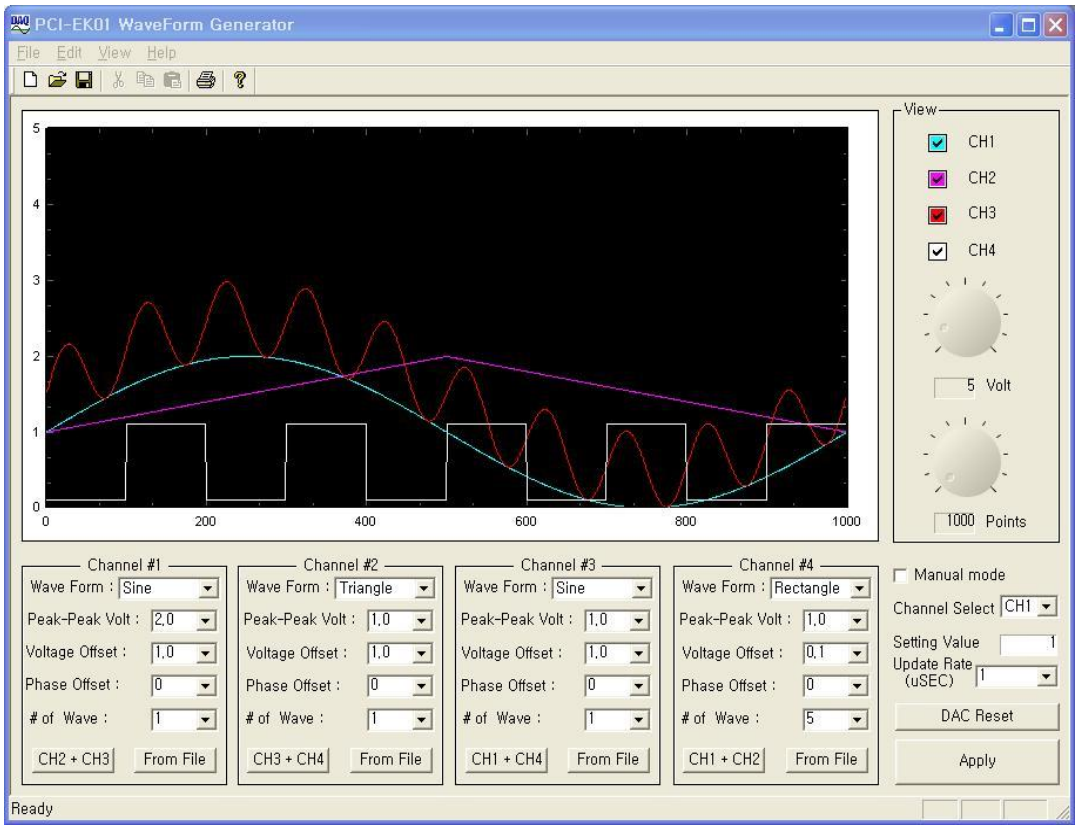
[Figure 5-2. Connector connection for testing]



[Figure 5-3. Check analog output with oscilloscope]

(4) Since the output above comes from all four analog out output channels, check all four channels using an oscilloscope.

(5) In order to easily generate wave forms, you can easily create and use various output waveforms by using the wave form generator as shown in [Figure 5-4]. Once you create the desired waveform, click the Apply button to apply the created waveform to the board.



[Figure 5-4. Wave Form Generator execution screen]

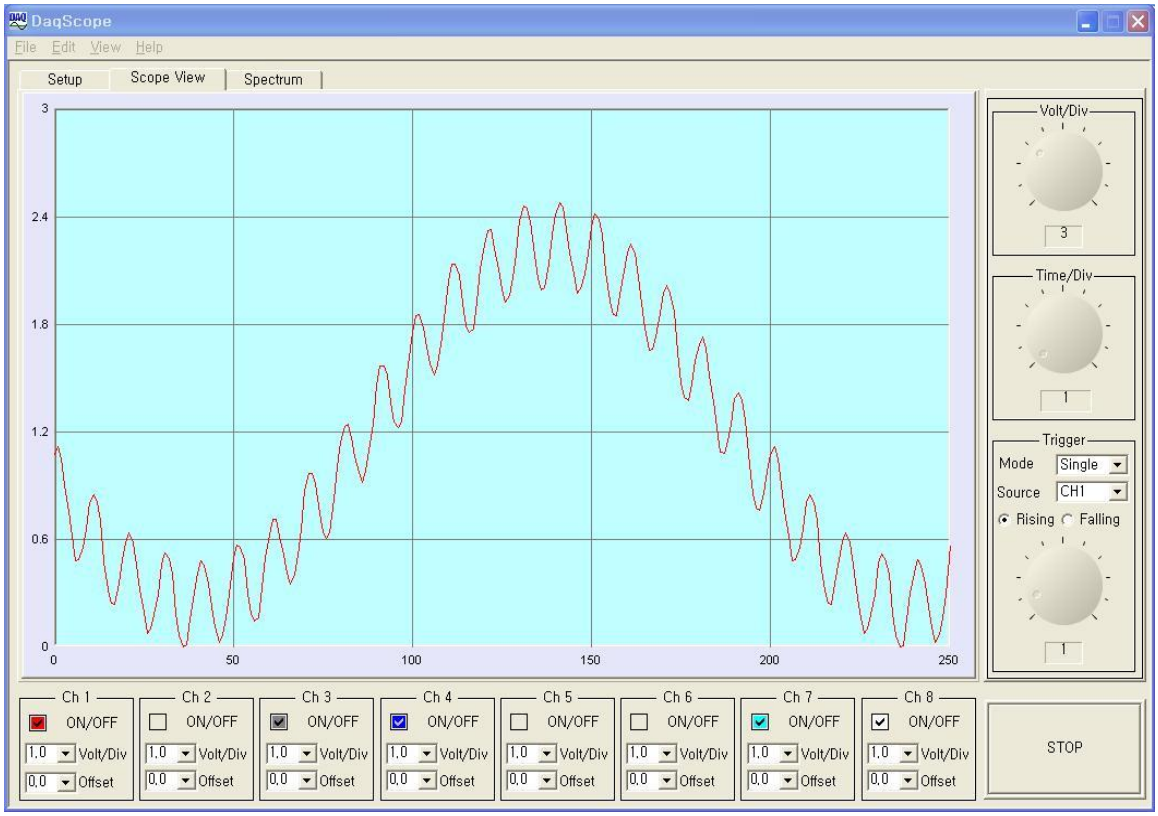
5-2 Analog Input

(1) Connect numbers 6 (Analog out 0) and 14 (Analog In 0) of the P1 (D-sub 25pin plug) connector using a jumper wire.

(2) From the test results above, enter "84" in I/O address offset and "01" in Data, then press the I/O Write button. This is to stop the automatic operation of the ADC and to test the high-speed RAM used as the ADC buffer.

(3) Press the **Memory Write** button and then press the **Memory Read** button again. As a result, the memory data is displayed on the screen. Check whether "0" at address 0 and "3FFFF" at address FFFFCh are recorded as double word values.

(4) Again, enter "84" in I/O address offset and "00" in Data, then press the **I/O Write** button.



[Figure 5-5. DaqScope execution screen]

(5) As shown in the picture above, when you run the DaqScope program and press the RUN/STOP button, the ADC value of the analog out channel will be displayed on the screen.

(6) Press the **ADC Channel Enable** button to display only the desired channels.

(7) If you press the RUN/STOP button again, the ADC value will stop being displayed on the screen.

5-3 Digital Input/Output

(1) Enter "84" in **I/O address offset** and "0" in **Data**, then press the **I/O Write** button.
This is to stop the automatic operation of the ADC and use the local address and data bus to control the 82C55.

(2) Enter "A0" in **I/O address offset** and "80" in **Data** and press the **I/O Write** button.
This sets all ports (A/B/C) of the 82C55 to Mode 0, output.

(3) Enter "A4" in **I/O address offset** and "FF" in **Data** and press the **I/O Write** button.
This records the "FF" value in the A port of the 82C55.

(4) Check whether the output is accurate by referring to the detailed description of the oscilloscope and JP1.

(5) Enter "A4" in **I/O address offset** and "0" in **Data**, then press the **I/O Write** button and check the output with an oscilloscope. This records the "FF" value in the A port of the 82C55.

(6) Perform steps 3 to 5 above for ports B/C.

5-4 Counter

(1) Enter "8" in **I/O address offset** and "1" in **Data** and press the **I/O Write** button.
This enables the operation of Counter 0.

(2) Enter "4" in **I/O address offset** and "3" in **Data** and press the **I/O Write** button.
This records 3 in the Target setting counter.

(3) Apply High Active pulse once to the counter 0 input (pin 11) of connector P1. After authorization, enter "0" in the **I/O address offset** and press the **I/O Read** button to check whether the current count is "1".

(4) Apply two more count pulses and the current count value is "3", and read the STATUS register of counter 0 to check whether a count over has occurred.

(5) In the case of PCI-EK01(B), there are three counters, so perform the same test as above on the remaining two counters. This records 3 in the Target setting counter.

5-5 Timer

(1) Enter "34" in **I/O address offset** and "4" in **Data** and press the **I/O Write** button.
This loads the setting value of Timer 0 into the register.

(2) Enter "38" in **I/O address offset** and "5" in **Data** and press the **I/O Write** button.
This reverses the output when Timer 0 times out in Auto-reload mode.
If you check the frequency at this time, it is 5MHz.
If we express this in the formula
Frequency = 25M / (TMR_SET + 1).
The TMR_SET value must be at least 1.

(3) The timer result is output from connector P2's pin 22 (Timer 0), so connect the oscilloscope to pin 22 to check.

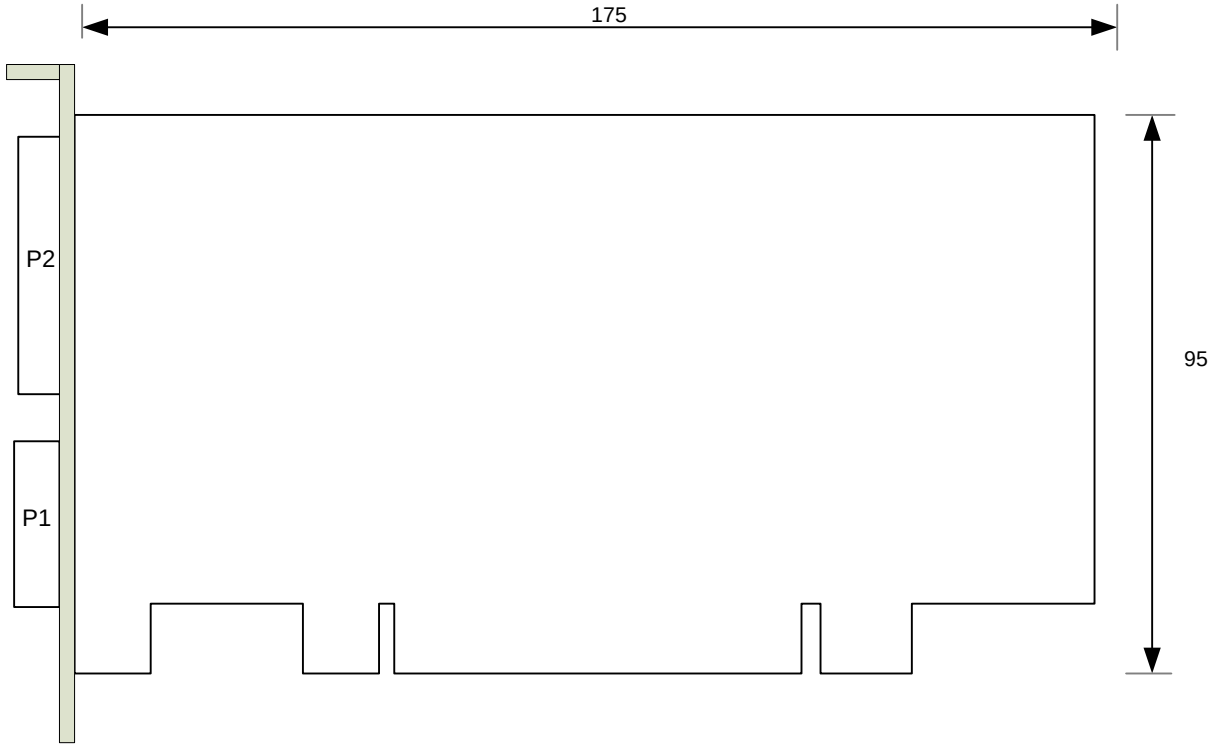
(4) Enter "38" in **I/O address offset** and "d" in **Data** and press the **I/O Write** button.
This outputs a 20nSEC pulse when Timer 0 times out in Auto-reload mode.
If you check the frequency at this time, it becomes 10MHz.
If we express this in the formula
Frequency = 50M / (TMR_SET + 1).
The TMR_SET value must be at least 1.

(5) In the case of PCI-EK01(B), there are three timers, so perform the above test on the remaining two timers.

Appendix

A-1 Board Size

The external dimensions of the board are as follows.



A-2 Repair Regulations

Thank you for purchasing DAQ SYSTEM's product. Please refer to the following regarding Customer Service stipulated by DAQ SYSTEM.

(1) Please read the user's manual and follow the instructions before using the DAQ SYSTEM product.

(2) When returning the product to be repaired, please send it to the head office with the symptoms of the malfunction as well.

(3) All DAQ SYSTEM products have a one-year warranty.
-. The warranty period is counted from the date the product is shipped from DAQ SYSTEM.
-. Peripherals and third-party products not manufactured by DAQ SYSTEM are covered by the manufacturer's warranty.
-. If repair is required, please contact the contact points below.

(4) Even during the free repair warranty period, paid repairs are made in the following cases.
① Failure or damage caused by not following the user's manual
② Failure or damage caused by customer negligence during product transportation after purchase
③ Natural phenomena such as fire, earthquake, flood, lightning, pollution, etc. or power supply exceeding the recommended range malfunction or damage
④ Failures caused by inappropriate storage environment (eg, high temperature, high humidity, volatile chemicals, etc.) damaged
⑤ Failure or damage due to unreasonable repair or modification
⑥ Products whose serial number has been changed or intentionally removed
⑦ In the event that DAQ SYSTEM determines that it is the customer's negligence for other reasons

(5) The customer must bear the shipping cost of returning the repaired product to DAQ SYSTEM.

(6) The manufacturer is not responsible for any problems caused by incorrect use regardless of our Warranty provisions.

References

1. PCI System Architecture

-- MindShare Inc.

2. PCI Local Bus Specification

-- PCI-SIG

3. AN201 How to build application using APIs

-- DAQ system

4. AN242 PCI-DIO64xx Series API Programming

-- DAQ system

MEMO

Contact Point

Web sit : https://www.daqsystem.com

Email : postmaster@daqsystem.com

DAQ SYSTEM
Measurement & Automation