PCI8602

User's Manual

Beijing ART Technology Development Co., Ltd.

Contents

Chapter 1 Overview

In the fields of Real-time Signal Processing, Digital Image Processing and others, high-speed and high-precision data acquisition modules are demanded. ART PCI8602 data acquisition module, which brings in advantages of similar products that produced in china and other countries, is convenient for use, high cost and stable performance.

ART PCI8602 is a data acquisition module based on PCI bus. It can be directly inserted into IBM-PC/AT or a computer which is compatible with PCI8602 to constitute the laboratory, product quality testing center and systems for different areas of data acquisition, waveform analysis and processing. It may also constitute the monitoring system for industrial production process.

Unpacking Checklist

Check the shipping carton for any damage. If the shipping carton and contents are damaged, notify the local dealer or sales for a replacement. Retain the shipping carton and packing material for inspection by the dealer.
Check for the following items in the package. If there are any missing items, contact your local dealer or sales.

- PCI8602 Data Acquisition Board
- ART Disk
  a) user's manual (pdf)
  b) drive
  c) catalog
- Warranty Card

FEATURES

Analog Input

- Input Range: ±10V, ±5V, ±2.5, 0~10V, 0~5V
- 16-bit resolution
- Sampling Rate: 250KHz

Note: The actual channel sampling rate = sampling rate / number of sampling channels
Frequency Formula: sampling frequency= main frequency/the number of frequency division, main frequency = 40MHz, 32-bit frequency division, the number of frequency division range: 160~40000000.

- Analog Input Mode: 32SE/16DI
- Data Read Mode: non-empty, half-full inquiry mode and DMA mode
- Memory Depth: 8K word FIFO memory
- Memory Signs: full, non-empty and half-full
- AD Mode: continuum sampling , grouping sampling
- Group Interval: software-configurable, minimum value is sampling period, maximum value is 419430us
- Loops of Group: software-configurable, minimum value is one time , maximum value is 255 times
- Board Clock Output Frequency: the real sampling frequency of the current AD
- Clock Source: external clock, internal clock (software-configurable)
- Trigger Mode: software trigger, hardware trigger (external trigger)
- Trigger Type: level trigger, edge trigger
- Trigger Direction: negative, positive, positive and negative trigger

- Trigger Source: ATR, DTR
- Analog Trigger Source (ATR)Input Range: -10V~+10V
- Trigger Level: -10V~+10V
- Trigger Source DTR Input Range: standard TTL level
- AD Conversion Time: ≤10us
- Programmable Gain: 1, 2, 4, 8 (AD8251 default) or 1, 2, 5, 10 (AD8250) or 1, 10, 100, 1000 (AD8253)
- Analog Input Impedance: 10MΩ
- Amplifier Set-up Time: 785ns(0.001%)(max)
- Non-linear error: ±3LSB(Maximum)
- System Measurement Accuracy: 0.01%
- Operating Temperature Range: 0℃~50℃
- Storage Temperature Range: -20℃~70℃

Analog Output

- Output Range: ±10V,±5V, 0~10V, 0~5V
- 12-bit resolution
- Set-up Time: 10μs (0.01%)
- Channel No.: 4-channel
- Non-linear error: ±1LSB(Maximum)
- Output Error (full-scale): ±1LSB
- Operating Temperature Range: 0℃~50℃
- Storage Temperature Range: -20℃~70℃

Digital Input

- Channel No.: 8-channel
- Electric Standard: TTL compatible
- High Voltage: ≧2V
- Low Voltage: ≦0.8V

Digital Output

- Channel No.: 8-channel
- Electrical Standard: TTL compatible
- High Voltage: ≧3.8V
- Low Voltage: ≦0.44V
- Power-on Reset

CNT Counter/timer

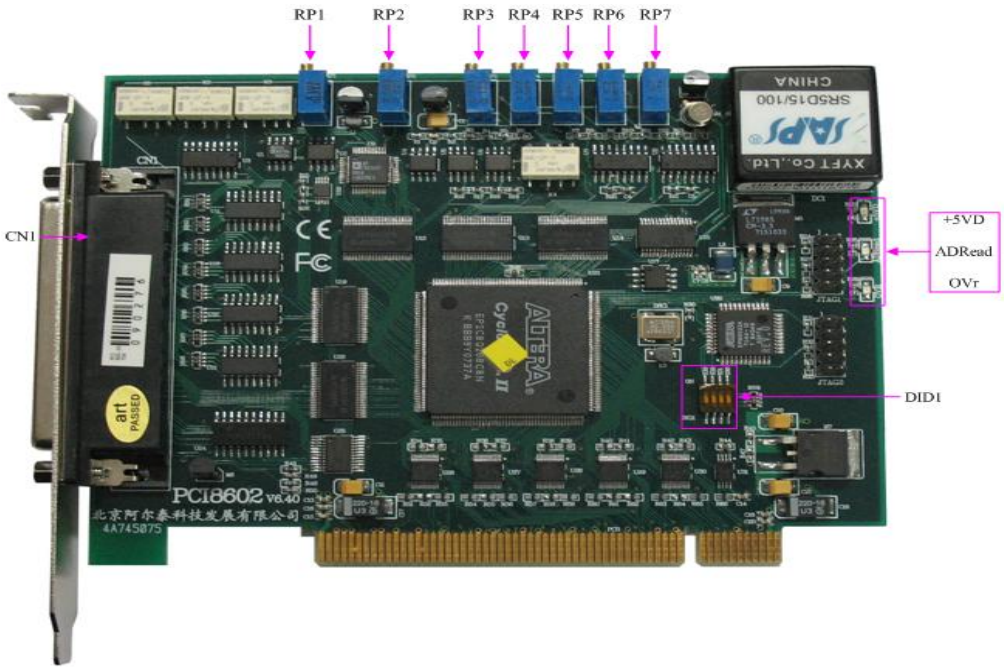
- Tiptop time base is20 MHz, 16-bit counter/timer
- Function Mode: counter and pulse generator
- Clock Source: local clock (620Hz~20MHz), external clock(maximum frequency 20MHz)
- Gate: rising edge, falling edge, high level, low level
- Counter Output: high level, low level

Other features

Board Clock Oscillation: 40MHz

Chapter 2 Components Layout Diagram and a Brief Description

2.1 The Main Component Layout Diagram



2.2 The Function Description for the Main Component

2.2.1 Signal Input and Output Connectors

CN1: analog signal input and output connectors

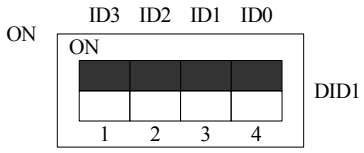
2.2.2 Potentiometer

RP1: AD analog signal input zero-point adjustment potentiometer
RP2: AD analog signal input full-scale adjustment potentiometer
RP3: DA analog signal output -2.5V norm voltage adjustment potentiometer
RP4: AO0 analog signal output full-scale adjustment potentiometer
RP5: AO1 analog signal output full-scale adjustment potentiometer
RP6: AO2 analog signal output full-scale adjustment potentiometer
RP7: AO3 analog signal output full-scale adjustment potentiometer

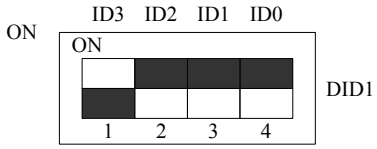
2.2.3 Physical ID of DIP Switch

DID1: Set physical ID number. When the PC is installed more than one PCI8602 , you can use the DIP switch to set a physical ID number for each board, which makes it very convenient for users to distinguish and visit each board in the

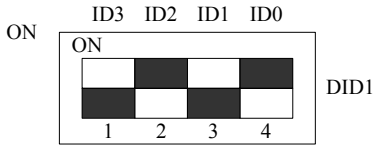
progress of the hardware configuration and software programming. The following four-place numbers are expressed by the binary system: When DIP switch points to "ON", that means "1", and when it points to the other side, that means "0." As they are shown in the following diagrams: place "ID3" is the high place."ID0" is the low place, and the black part in the diagram represents the location of the switch. (Test softwares of the company often use the logic ID management equipments and at this moment the physical ID DIP switch is invalid. If you want to use more than one kind of the equipments in one and the same system at the same time, please use the physical ID as much as possible.).



The above chart shows"1111", so it means that the physical ID is 15.



The above chart shows"0111", so it means that the physical ID is 7.



The above chart shows"0101", so it means that the physical ID is 5.

ID3	ID2	ID1	ID0	Physical ID（Hex）	Physical ID（Dec）
OFF（0）	OFF（0）	OFF（0）	OFF（0）	0	0
OFF（0）	OFF（0）	OFF（0）	ON（1）	1	1
OFF（0）	OFF（0）	ON（1）	OFF（0）	2	2
OFF（0）	OFF（0）	ON（1）	ON（1）	3	3
OFF（0）	ON（1）	OFF（0）	OFF（0）	4	4
OFF（0）	ON（1）	OFF（0）	ON（1）	5	5
OFF（0）	ON（1）	ON（1）	OFF（0）	6	6
OFF（0）	ON（1）	ON（1）	ON（1）	7	7
ON（1）	OFF（0）	OFF（0）	OFF（0）	8	8
ON（1）	OFF（0）	OFF（0）	ON（1）	9	9
ON（1）	OFF（0）	ON（1）	OFF（0）	A	10
ON（1）	OFF（0）	ON（1）	ON（1）	B	11
ON（1）	ON（1）	OFF（0）	OFF（0）	C	12
ON（1）	ON（1）	OFF（0）	ON（1）	D	13
ON（1）	ON（1）	ON（1）	OFF（0）	E	14
ON（1）	ON（1）	ON（1）	ON（1）	F	15

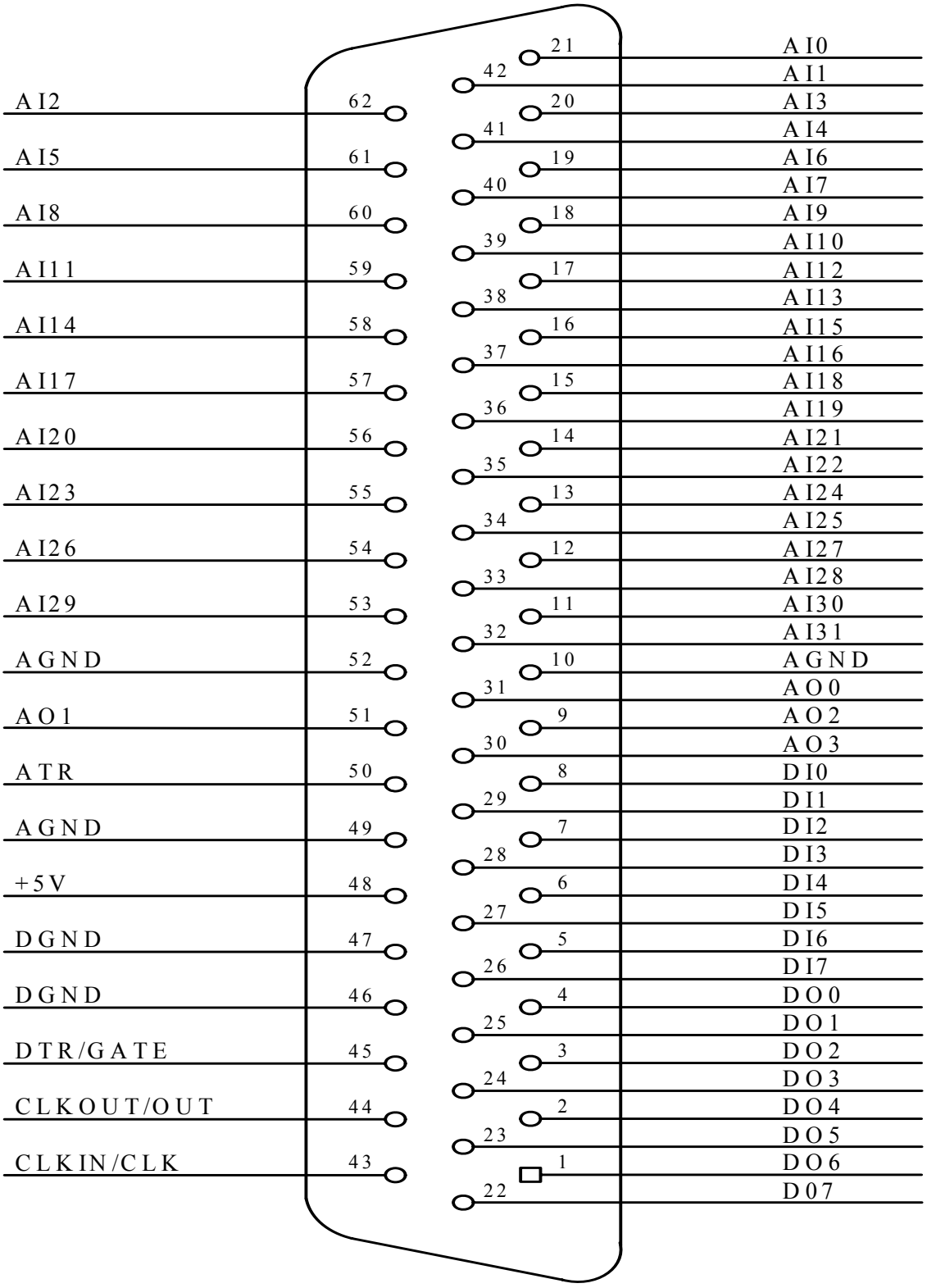
2.2.3 Indicator

+5VD: 5V power indicator, on for normal.
ADRead: read FIFO indicator, flashing for reading.
OVR: FIFO overflow indicator, on for overflow.

Chapter 3 Signal Connectors

3.1 The Definition of Signal Input and Output Connectors

62 core plug on the CN1 pin definition



Pin introduction: CLKIN/CLK, CLKOUT/OUT and DTR/GATE are complex pins. When use AD function, the pins above-mentioned are used as CLKIN, CLKOUT and DTR, if use counter/timer function, the pins are used as CLK, OUT and GATE. Each pin function is defined in the table.

Pin definition about AD:

Pin name	Type	Pin function definition
AI0～AI31	Input	AD analog input, reference ground is AGND.
AO0～AO3	Output	DA analog output.
DI0～DI7	Input	Digital input.
DO0～DO7	Output	Digital output.
AGND		Analog ground. This AGND pin should be connected to the system's AGND plane.
DGND		Digital ground. Ground reference for Digital circuitry. This DGND pin should be connected to the system's DGND plane.
CLKIN	Input	External clock input, please use DGND as reference ground.
CLKOUT	Output	Internal clock output, when allow clock output, it is internal clock output, otherwise it is CNT counter output. DGND for reference ground.
ATR	Input	Analog trigger signal input, choose AGND as reference ground.
DTR	Input	Digital trigger signal input, choose DGND as reference ground.
+5V	Output	Output 5V voltage.

Pin definition about counter/timer:

Pin name	Pin feature	Pin function definition
CLK	Input	Counter/timer clock source input, DGND for reference ground. When counter uses external clock, counter clock frequency maximum value is20MHz, default counter clock is internal clock LOCAL_CLK, frequency is 620Hz～20MHz
OUT	Output	Counter/timer output, it is counter output (OUT) when forbid clock output, otherwise it is AD clock output (CLKOUT). Default counter output (OUT), DGND for reference ground.
GATE	Input	Counter/timer gate input, choose DGND as reference ground.
DGND		Digital signals ground, when use counter/timer we best choose it as reference ground.

Chapter 4 Connection Ways for Each Signal

4.1 AD Single-ended Input Connection

Single-ended mode can achieve a signal input by one channel, and several signals use the common reference ground. This mode is widely applied in occasions of the small interference and relatively many channels.

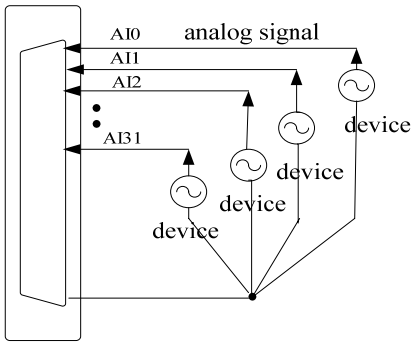


Figure 4.1 single-ended input connection

4.2 AD Double-ended Input Mode

Double-ended input mode, which was also called differential input mode, uses positive and negative channels to input a signal. This mode is mostly used when biggish interference happens and the channel numbers are few. Single-ended/double-ended mode can be set by the software, please refer to PCI8602 software manual.

According to the diagram below, PCI8602 board can be connected as analog voltage double-ended input mode, which can effectively suppress common-mode interference signal to improve the accuracy of acquisition. Positive side of the 16-channel analog input signal is connected to AI0~AI15, the negative side of the analog input signal is connected to AI16~AI31, equipments in industrial sites share the AGND with PCI8602 board.

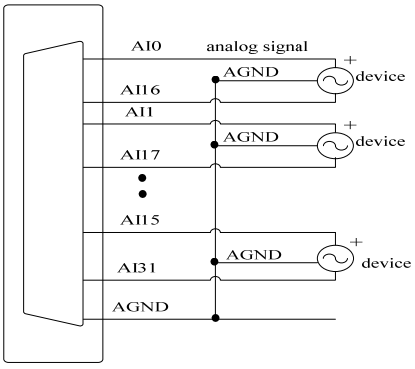


Figure 4.2 double-ended input connection

4.3 Other Connections

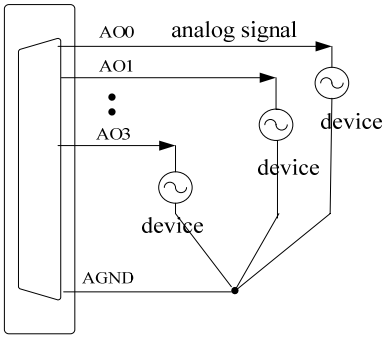


Figure 4.3 analog signal output connection

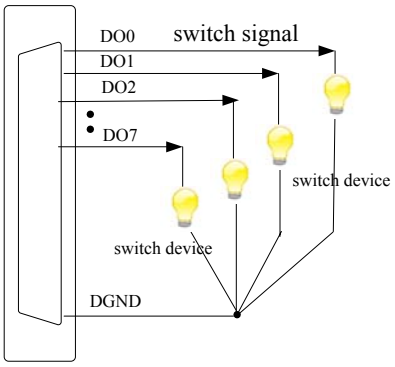


Figure 4.5 digital signal output connection

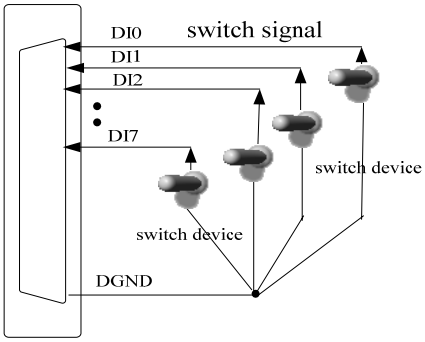


Figure 4.4 digital signal input connection

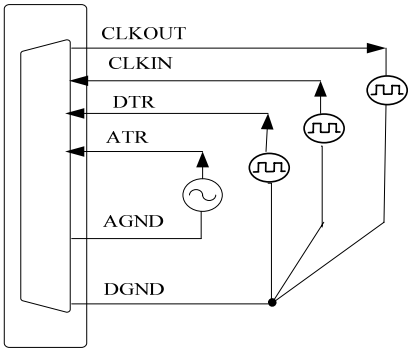
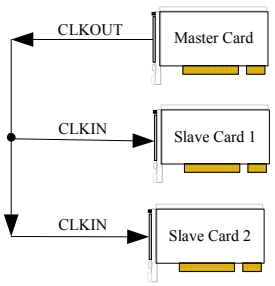


Figure 4.6 Clock Input/Output and Trigger Signal Connection

4.4 Methods of Realizing the Multi-card Synchronization

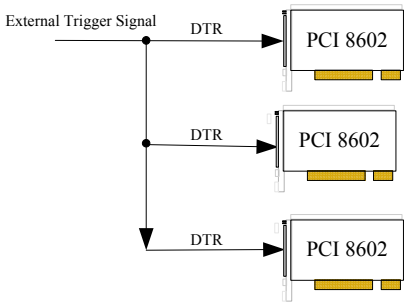
Three methods can realize the synchronization for the PCI8602, the first method is using the cascade master-slave card, the second one is using the common external trigger, and the last one is using the common external clock.

When using master-slave cascade card programs, the master card generally uses the internal clock source model, while the slave card uses the external clock source mode. After the master card and the slave card are initialized according to the corresponding clock source mode. At first, start all the slave cards, as the main card has not been activated and there is no output clock signal, so the slave card enters the wait state until the main card was activated. At this moment, the multi-card synchronization has been realized. When you need to sample more than channels of a card, you could consider using the multi-card cascaded model to expand the number of channels.



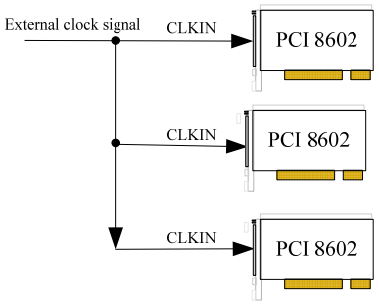
When using the common external trigger, please make sure all parameters of different PCI8602 are the same. At first, configure hardware parameters, and use analog or digital signal triggering (ATR or DTR), then connect the signal that will be sampled by PCI8602, input triggering

signal from ART pin or DTR pin, then click “Start Sampling” button, at this time, PCI8602 does not sample any signal but waits for external trigger signal. When each module is waiting for external trigger signal, use the common external trigger signal to startup modules, at last, we can realize synchronization data acquisition in this way. See the following figure:



Note: when using the DTR, select the internal clock mode

When using the common external clock trigger, please make sure all parameters of different PCI8602 are the same. At first, configure hardware parameters, and use external clock, then connect the signal that will be sampled by PCI8602, input trigger signal from ART pin or DTR pin, then click “Start Sampling” button, at this time, PCI8602 does not sample any signal, but wait for external clock signal. When each module is waiting for external clock signal, use the common external clock signal to startup modules, at last, we realize synchronization data acquisition in this way. See the following figure:



Chapter 5 The Instruction of the AD Trigger Function

5.1 AD Internal Trigger Mode

When AD is in the initialization, if the AD hardware parameter ADPara.TriggerMode = PCI8602_TRIGMODE_SOFT, we can achieve the internal trigger acquisition. In this function, when calling the StartDeviceProAD function, it will generate AD start pulse, AD immediately access to the conversion process and not wait for the conditions of any other external hardware. It also can be interpreted as the software trigger.

As for the specific process, please see the figure below, the cycle of the AD work pulse is decided by the sampling frequency.

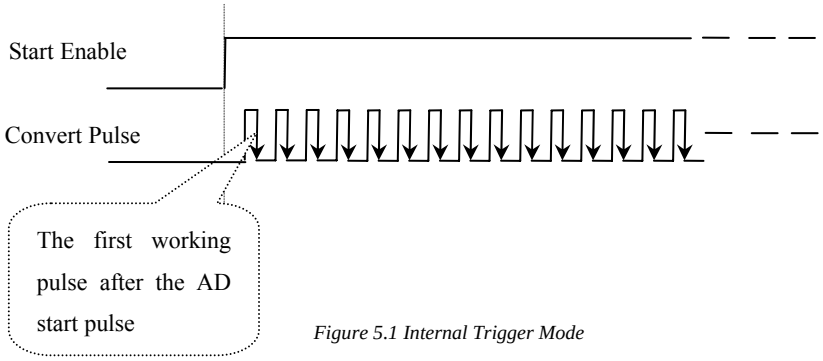


Figure 5.1 Internal Trigger Mode

5.2 AD External Trigger Mode

When AD is in the initialization, if the AD hardware parameter ADPara.TriggerMode = PCI8602_TRIGMODE_POST, we can achieve the external trigger acquisition. In this function, when calling the StartDeviceProAD function, AD will not immediately access to the conversion process but wait for the external trigger source signals accord with the condition, then start converting the data. It also can be interpreted as the hardware trigger. Trigger source includes the DTR (Digital Trigger Source) and ATR (Analog Trigger Source)

(一) ATR Trigger

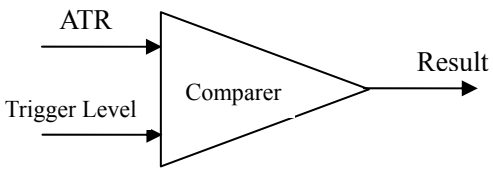


Figure 5.2 Analog compare

The trigger modes include the edge trigger and level trigger.

(1) Edge trigger function

Edge trigger is to capture the characteristics of the changes between the trigger source signal and the trigger level signal to trigger AD conversion. When TriggerType=PCI8602_TRIGTYPE_EDGE, it is the edge trigger type.
When ADPara.TriggerDir = PCI8602_TRIGDIR_NEGATIVE, choose the trigger mode as the falling edge trigger. That is, when the ATR trigger signal is on the falling edge, AD will immediately access to the conversion process, and its follow-up changes have no effect on AD acquisition.

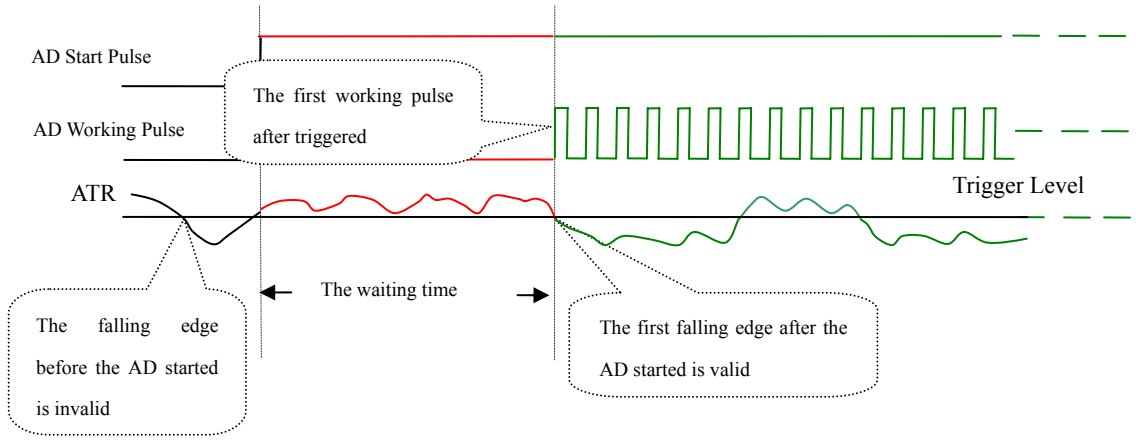


Figure 5.2.1 Falling edge Trigger

When ADPara.TriggerDir = PCI8602_TRIGDIR_POSITIVE, choose the trigger mode as rising edge trigger. That is, when the ATR trigger signal is on the rising edge, AD will immediately access to the conversion process, and its follow-up changes have no effect on AD acquisition.

When ADPara.TriggerDir = PCI8602_TRIGDIR_POSIT_NEGAT, choose the trigger mode as rising or falling edge trigger. That is, when the ATR trigger signal is on the rising or falling edge, AD will immediately access to the conversion process, and its follow-up changes have no effect on AD acquisition. This function can be used in the case that the acquisition will occur if the exoteric signal changes.

（2）Triggering level function

Level trigger is to capture the condition that trigger signal is higher or lower than the trigger level to trigger AD conversion. When ADPara.TriggerType = PCI8602_TRIGTYPE_PULSE, it is level trigger type.
When ADPara.TriggerDir = PCI8602_TRIGDIR_NEGATIVE, AD is in the conversion process if the ATR is lower than the trigger level. And AD conversion will automatically stop if the ATR is higher than the trigger level. AD's work status changes with changes of ATR.

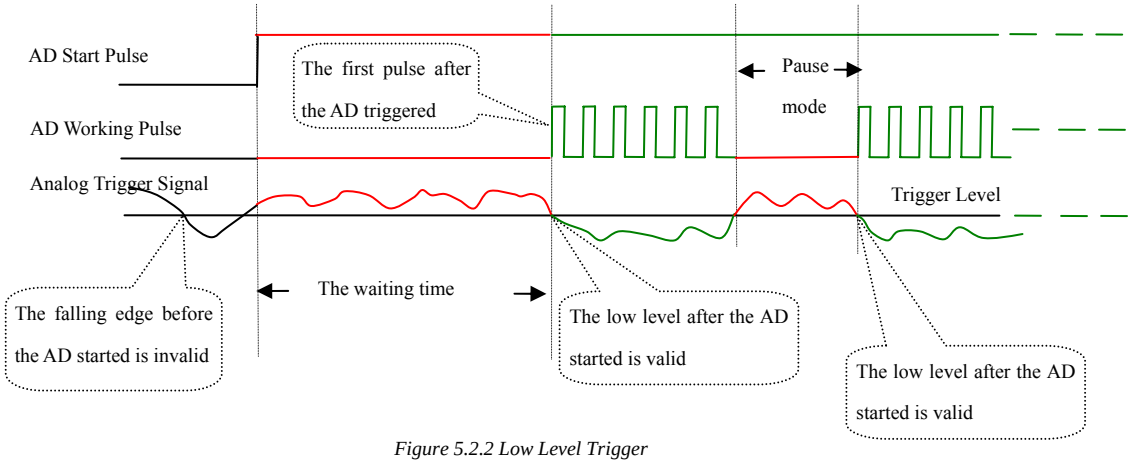


Figure 5.2.2 Low Level Trigger

When ADPara.TriggerDir = PCI8602_TRIGDIR_POSITIVE, AD is in the conversion process if the ATR is higher than the trigger level. And AD conversion will automatically stop if the ATR is lower than the trigger level. AD's work status changes with changes of ATR.

When ADPara.TriggerDir = PCI8602_TRIGDIR_POSIT_NEGAT, it means the trigger level is low. The effect is the same as the internal software trigger.

When the trigger signal is the analog signal, using the ATR trigger source. Trigger level needs to be set when using the ATR trigger source, two channels' output voltages of DA (AO0 and AO1) codetermines the trigger level; in this case, we need set 0-channel output voltage higher than 1-channel output voltage. There are two trigger types: edge trigger and level trigger

(二) DTR Trigger

When the trigger signal is the digital signal (standard TTL-level), using the DTR trigger source.

(1) Edge trigger function

Edge trigger is to capture the characteristics of the changes between the trigger source signal and the trigger level signal to trigger AD conversion.

When ADPara.TriggerDir = PCI8602_TRIGDIR_NEGATIVE, choose the trigger mode as the falling edge trigger. That is, when the DTR trigger signal is on the falling edge, AD will immediately access to the conversion process, and its follow-up changes have no effect on AD acquisition.

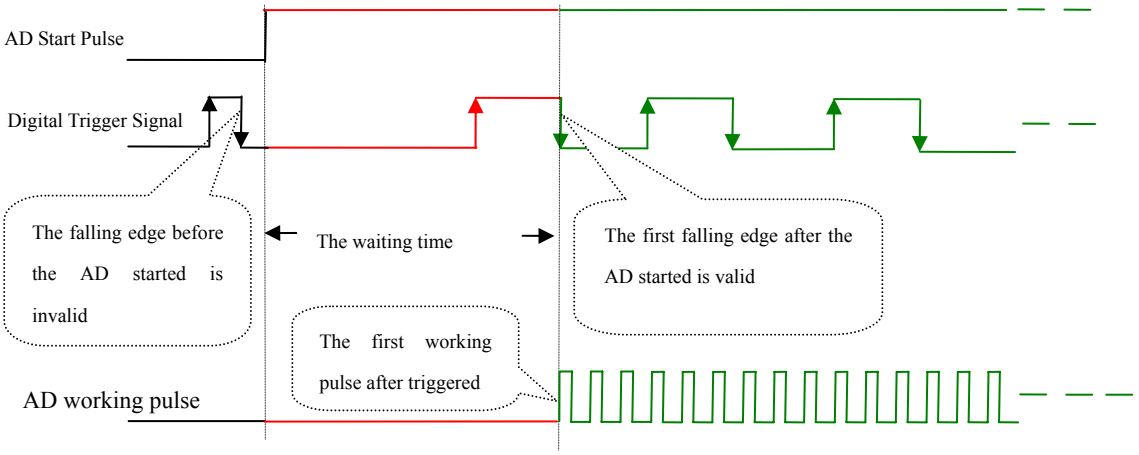


Figure 5.2.4 Falling edge Trigger

When ADPara.TriggerDir = PCI8602_TRIGDIR_POSITIVE, choose the trigger mode as rising edge trigger. That is, when the DTR trigger signal is on the rising edge, AD will immediately access to the conversion process, and its follow-up changes have no effect on AD acquisition.

When ADPara.TriggerDir = PCI8602_TRIGDIR_POSIT_NEGAT, choose the trigger mode as rising or falling edge trigger. That is, when the DTR trigger signal is on the rising or falling edge, AD will immediately access to the conversion process, and its follow-up changes have no effect on AD acquisition. This function can be used in the case that the acquisition will occur if the exoteric signal changes.

（2）Triggering level function

Level trigger is to capture the condition that trigger signal is higher or lower than the trigger level to trigger AD conversion.

When ADPara.TriggerDir = PCI8602_TRIGDIR_NEGATIVE, it means the trigger level is low. When DTR trigger signal is in low level, AD is in the conversion process, once the trigger signal is in the high level, AD conversion will automatically stop, when the trigger signal is in the low level again, AD will re-access to the conversion process, that is, only converting the data when the trigger signal is in the low level.

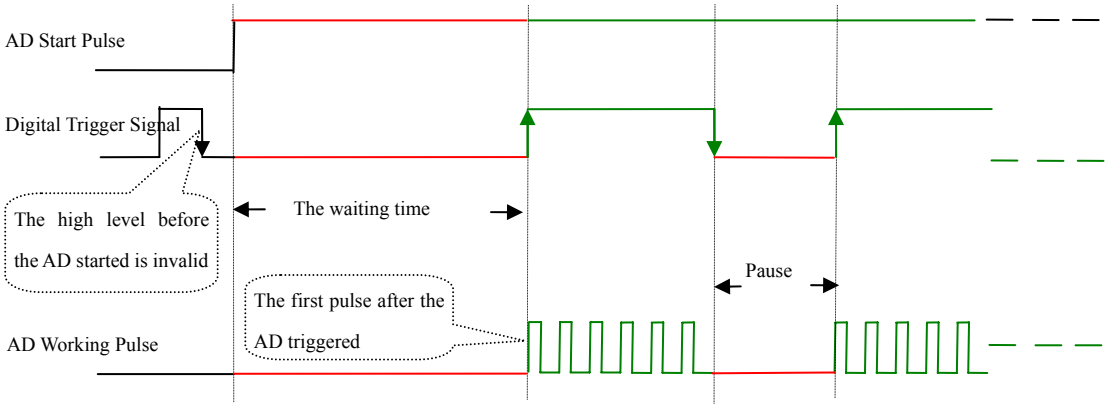


Figure 5.2.4 High Level Trigger

When ADPara.TriggerDir = PCI8602_TRIGDIR_POSITIVE, it means the trigger level is high. When DTR trigger signal is in high level, AD is in the conversion process, once the trigger signal is in the low level, AD conversion will automatically stop, when the trigger signal is in the high level again, AD will re-access to the conversion process, that is, only converting the data when the trigger signal is in the high level.

When ADPara.TriggerDir = PCI8602_TRIGDIR_POSIT_NEGAT, it means the trigger level is low or high. The effect is the same as the internal software trigger.

Chapter 6 Methods of using AD Internal and External Clock Function

6.1 Internal Clock Function of AD

Internal Clock Function refers to the use of on-board clock oscillator and the clock signals which are produced by the user-specified frequency to trigger the AD conversion regularly. To use the clock function, the hardware parameters ADPara.ClockSource = PCI8602 _CLOCKSRC_IN should be installed in the software. The frequency of the clock in the software depends on the hardware parameters ADPara.Frequency. For example, if Frequency = 100000, that means AD work frequency is 100000Hz (that is, 100 KHz, 10 μs /point).

6.2 External Clock Function of AD

External Clock Function refers to the use of the outside clock signals to trigger the AD conversion regularly. The clock signals are provide by the CLKIN pin of the CN1 connector. The outside clock can be provided by PCI8602 clock output (CLKOUT of CN1), as well as other equipments, for example clock frequency generators. To use the external clock function, the hardware parameters ADPara.ClockSource = PCI8602_CLOCKSRC_OUT should be installed in the software. The clock frequency depends on the frequency of the external clock, and the clock frequency on-board (that is, the frequency depends on the hardware parameters ADPara.Frequency) only functions in the packet acquisition mode and its sampling frequency of the AD is fully controlled by the external clock frequency.

6.3 Methods of Using AD Continuum and Grouping Sampling Function

6.3.1 AD Continuum Sampling Function

The continuous acquisition function means the sampling periods for every two data points are completely equal in the sampling process of AD, that is, completely uniform speed acquisition, without any pause, so we call that continuous acquisition.

To use the continuous acquisition function, the hardware parameters ADPara.ADMode = PCI8602_ADMODE_SEQUENCE should be installed in the software. For example, in the internal clock mode, hardware parameters ADPara.Frequency = 100000 (100KHz) should be installed, and 10 microseconds after the AD converts the first data point, the second data point conversion starts, and then 10 microseconds later the third data point begins to convert, and so on.

6.3.2 AD Grouping Sampling Function

Grouping acquisition (pseudo-synchronous acquisition) function refers to the sampling clock frequency conversion among the channels of the group in the AD sampling process, and also a certain waiting time exists between every two groups, this period of time is known as the Group Interval. Loops of group refer to numbers of the cycle acquisition for each channel in the same group. In the internal clock mode and the fixed-frequency external clock mode, the time between the groups is known as group cycle. The conversion process of this acquisition mode as follows: a short time stop after the channels conversion in the group (that is, Group Interval), and then converting the next group, followed by repeated operations in order, so we call it grouping acquisition.

The purpose of the application of the grouping acquisition is that: at a relatively slow frequency, to ensure that all of the time difference between channels to become smaller in order to make the phase difference become smaller, thus to ensure the synchronization of the channels, so we also say it is the pseudo-synchronous acquisition function. In a group, the higher the sampling frequency is, the longer Group Interval is, and the better the relative synchronization signal is. The sampling frequency in a group depends on ADPara. Frequency, Loops of group depends on ADPara.LoopsOfGroup, the Group Interval depend on ADPara. Group Interval.

Based on the grouping function, it can be divided into the internal clock mode and the external clock mode. Under the internal clock mode, the group cycle is decided by the internal clock sampling period, the total number of sampling channels, Loops of group and Group Interval together. In each cycle of a group, AD only collects a set of data. Under the external clock mode, external clock cycle ≥ internal clock sampling cycle × the total number of sampling channels × Loops of group + AD chip conversion time, AD data acquisition is controlled and triggered by external clock. The external clock mode is divided into fixed frequency external clock mode and unfixed frequency external clock mode. Under the fixed frequency external clock mode, the group cycle is the sampling period of the external clock.

The formula for calculating the external signal frequency is as follows:

Under the internal clock mode:
Group Cycle = the internal clock sampling period × the total number of sample channels × Loops of group + AD chips conversion time + Group Interval
External signal cycle = (cycle signal points / Loops of group) × Group Cycle
External signal frequency = 1 / external signal cycle

Under the external clock mode: (a fixed-frequency external clock)
Group Cycle = external clock cycle
External signal cycle = (cycle signal points / Loops of group) × Group Cycle
External signal frequency = 1 / external signal cycle

Formula Notes:
The internal sampling clock cycle = 1 / (AD Para. Frequency)
The total number of sampling channels = AD Para. Last Channel – AD Para. First Channel + 1
Loops of group == ADPara.LoopsOfGroup
AD Chips conversion time = see "AD Analog Input Function" parameter

Group Interval = AD Para. Group Interval
Signal Cycle Points = with the display of the waveform signal in test procedures, we can use the mouse to measure the signal cycle points.

Under the internal clock mode, for example, sample two-channel 0, 1, and then 0 and 1 become a group. Sampling frequency (Frequency) = 100000Hz (cycle is 10μs), Loops of group is 1, Group Interval = 50μs, then the acquisition process is to collect a set of data first, including a data of channel 0 and a data of channel 1. We need 10μs to sample the two data, 20μs to convert the data from the two channels. After the conversion time of an AD chip, AD will automatically cut-off to enter into the waiting state until the 50μs group interval ends. We start the next group, begin to convert the data of channel 0 and 1, and then enter into the waiting state again, and the conversion is going on in this way, as the diagram following shows:

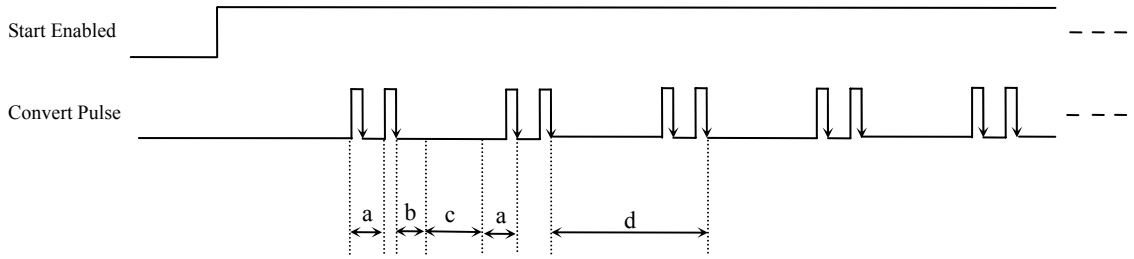


Figure 6.1 Grouping Sampling which grouping cycle No is 1 under the Internal Clock Mode

Note: a— internal clock sample cycle
b— AD chips conversion time
c—Group Interval
d— group cycle

Change the loops of group into 2, then the acquisition process is to collect the first set of data, including two data of channel 0 and two data of channel 1, the conversion order is 0,1,0,1. We need 10μs to sample each of the four data. After the conversion time of an AD chip, AD will automatically stop to enter into the waiting state until the 50μs Group Interval ends. We start the next group, begin to convert the data of channel 0 and 1, and then enter into the waiting state again, and the conversion is going on in this way, as the diagram following shows:

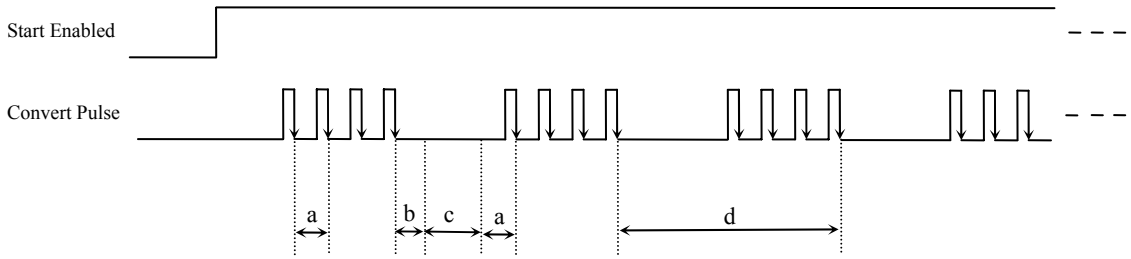


Figure 6.2 Grouping Sampling which grouping cycle No is 2 under the Internal Clock Mode

Notes: a— internal clock sample cycle
b— AD chips conversion time
c—Group Interval

d— group cycle

Under the external clock mode, the requirement is: the external clock cycle ≥ the internal clock sampling period × the total number of sampling channels× Loops of group + AD chip conversion time, otherwise, the external clock appearing in the group conversion time will be ignored.

Under the fixed-frequency external clock mode, for example, when sampling data of two-channel 0, 1, then channel 0 and channel 1 consist of a group. Sampling frequency (Frequency) = 100000Hz (the cycle is 10μs), Loops of group is 2, then the acquisition process is to collect the first set of data, including two data of channel 0 and two data of channel 1, the order of conversion 0,1,0,1, We need 10μs to sample the four data and 40μs to convert of the four data. After the conversion time of an AD chip, AD will automatically stop to enter into the waiting state until the next edge of the external clock triggers AD to do the next acquisition, and the conversion is going on in this way, as the diagram following shows:

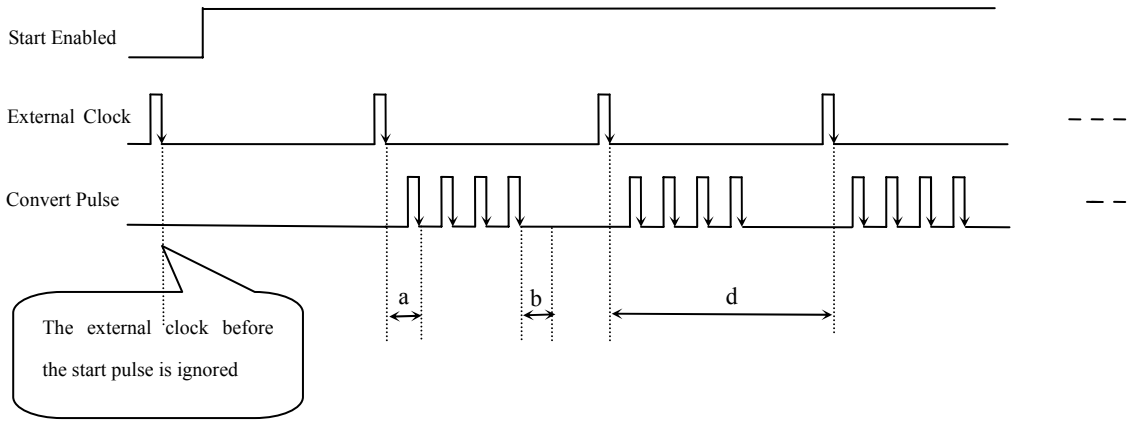


Figure 6.3 Grouping sampling under the fixed frequency external clock mode

Notes: a— internal clock sample cycle
b—AD chips conversion time
d—group cycle（external clock cycle）

Under an unfixed-frequency external clock mode, for example, the grouping sampling principle is the same as that of the fixed-frequency external clock mode. Under this mode, users can control any channel and any number of data. Users will connect the control signals with the clock input of the card (CLKIN), set the sampling channels and Loops of group. When there are external clock signals, it will sample the data which is set by users. Because the external clock frequency is not fixed, the size of external clock cycle is inconsistent but to meet: the external clock cycle ≥ the internal clock sampling period × the total number of sampling channels × Loops of group + AD chip conversion time, , otherwise, the external clock edge appearing in the group conversion time will be ignored.

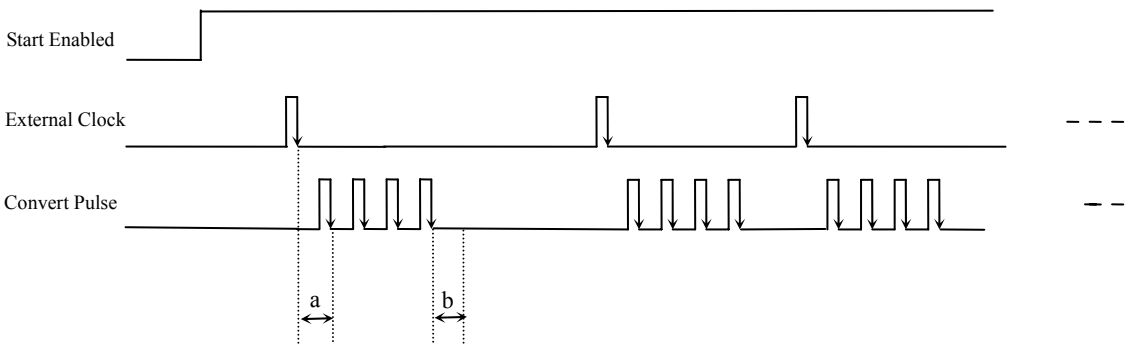


Figure 6.4 Grouping sampling under the not fixed frequency external clock mode

Note: a— internal clock sample cycle
b—AD chips conversion time

Chapter 7 Timer/Counter Function

7.1 Overview

The counter/timer of PCI8602 is designed to meet the counting and timing requirements, which has the functions that other common and off-the-shelf counter/timer devices could not reach. Counter/timer devices provide a wide range of solutions for measurement, including measuring a number of time-related variables, event count or the cumulative plus. The counter/timer of PCI8602 is a 16-bit counter, which is usually used as the component to perform key timing and synchronization functions in the complicated measurement system. According to work mode, it can be divided into timing pulse generator and counter mode, and the counter mode includes the simple counter and the buffer counter. Under the counter mode, we define the buffer count prohibition as the simple count and the buffer count permission as the buffer count. Each model will have a variety of options for the gate controlling, and they are adapted to the following aspects:

- ✧ Frequency measurement
- ✧ Edge or event count (the cumulative plus)
- ✧ Condition count
- ✧ Pulse width measurement
- ✧ Time tagging of events
- ✧ Frequency generation
- ✧ Pulse sequence generation and pulse width modulation (PWM)

The OUT output of the counter keeps the low level when power is on again. In terms of their needs, users can change "the level direction of counter output" (the software parameter OutputDir) to select that it is the low level or the high level when the counter stops count. The default setting is the low level when the counter stops count.

Please note the CLK, GATE and OUT that mentioned in this chapter which belong to the corresponding pin of the connector CN1, and for the specific definition, please refer to the chapter "Signal Input and Output Connectors".

The gate controlling edge of this counter is captured by an oscillation of 40M.

7.2 Count mode

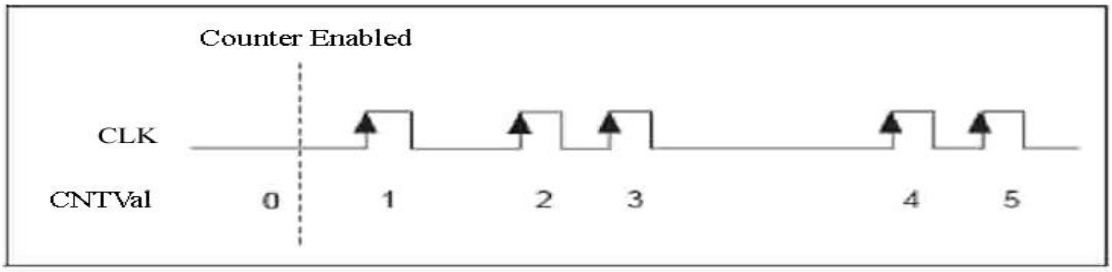
Under the counter function mode, the OUT output in the way of working has the same rule: each time after count, the counter adds “1”. When the count value reaches 65,535, the overflow happens, and the overflow symbol is "1". Users can select the "stop count" or the "keep count" after the overflow. When you choose the "stop count", the count value keeps 65535 after the overflow, and the OUT output of the counter maintains high level; when you choose the "keep count", the counter starts to count from the initial number adding "1" each time after the overflow, and the OUT output is the high level, but for the second overflow, the OUT output is the low level, and so is the following count mode. The default count mode is the "stop count".

The initial count value is controlled by CNTVal parameters, the buffer count number is stored by Width Val under the

buffer count, and the setting of this parameter is achieved by the function SetDeviceCNT.

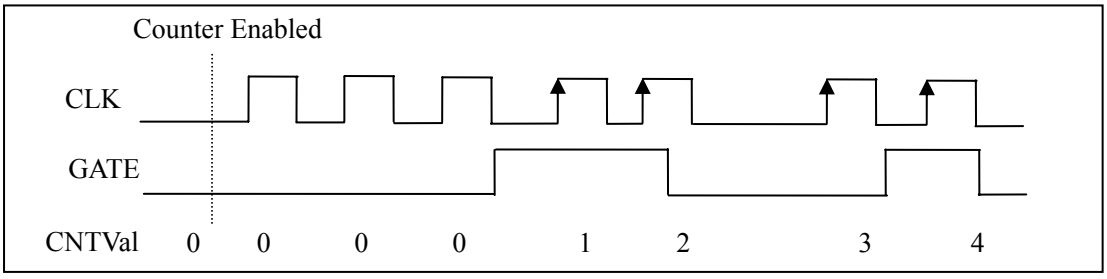
7.2.1 The functions of simple count and time measurement

Mode 0: do not use gate controlling signal



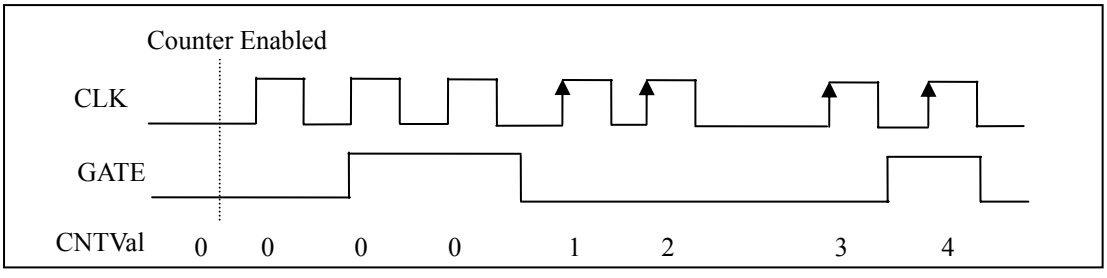
After the user writes the initial number and the count is allowed, the counter starts to increase from the initial number adding "1" each time if CLK generates a rising edge, and so is the following count mode.

Mode 1: The count is triggered on the rising edge of GATE and the following edges are invalid.



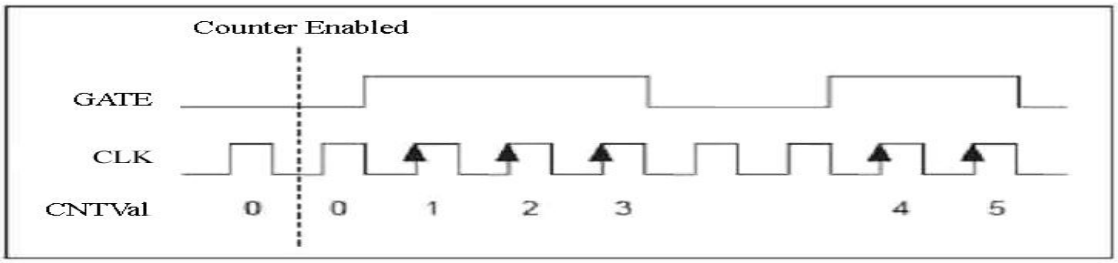
After the user writes the initial number and the count is allowed, the counter starts to count from the initial number adding "1" each time on the rising edge of the CLK only when GATE changes from the low level to the high level and also it means GATE produces a rising edge, and the change of the follow-up GATE is invalid, and so is the following count mode. Under this mode, it is equivalent to use the first rising edge of GATE as the counter's startup signal.

Mode 2: The count is triggered on the falling edge of GATE and the following edges are invalid.



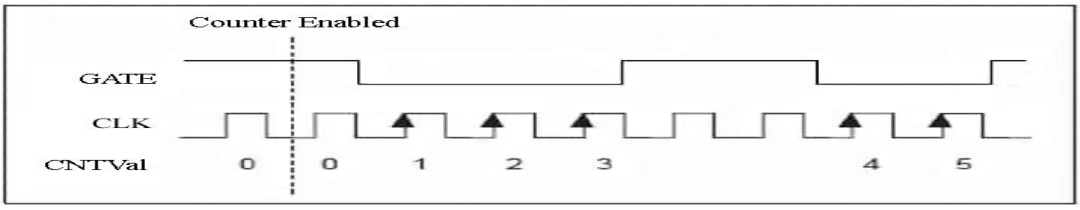
After the user writes the initial number and the count is allowed, the counter starts to count from the initial number adding "1" each time on the falling edge of the CLK only when GATE changes from the high level to the low level and also it means GATE produces a falling edge, and the change of the follow-up GATE is invalid, and so is the following count mode. Under this mode, it is equivalent to use the first falling edge of GATE as the counter's startup signal.

Mode 3: The high level is valid.



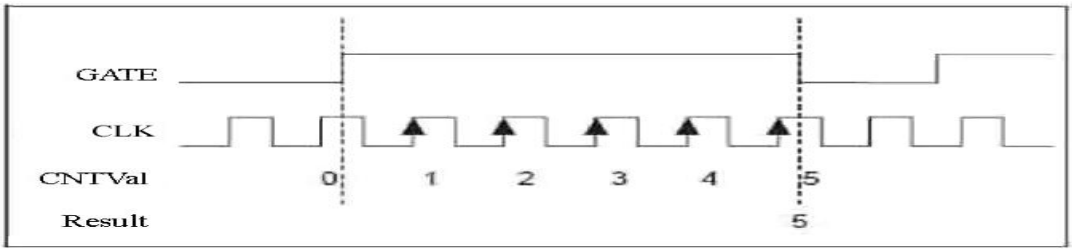
After the user writes the initial number and the count is allowed, the counter starts to count from the initial number adding "1" each time on the rising edge of the CLK when GATE is the high level, and when GATE is the low level, the counter stop count and the count value remain unchanged. If GATE turns to be the high level again, the counter starts to count from the former value adding "1" each time, and so is the following count mode. This function is suitable for the unilateral-condition count.

Mode 4: The low level is valid.



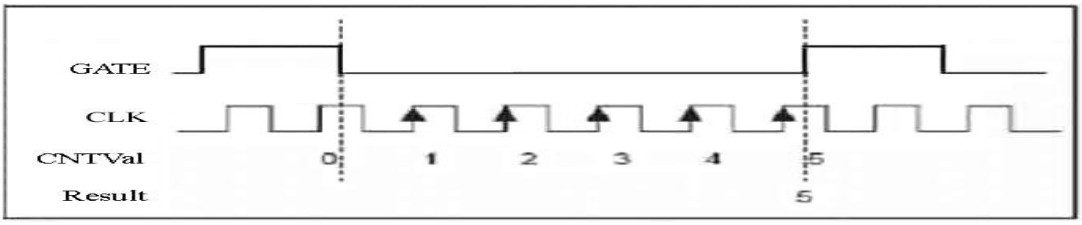
After the user writes the initial number and the count is allowed, the counter starts to count from the initial number adding "1" each time on the rising edge of the CLK when GATE is the low level, and when GATE is the high level, the counter stop count and the count value remain unchanged. If GATE turns to be the low level again, the counter starts to count from the former value adding "1" each time, and so is the following count mode. This function is suitable for the unilateral-condition count.

Mode 5: the trigger count on the rising edge and the stop count on the falling edge



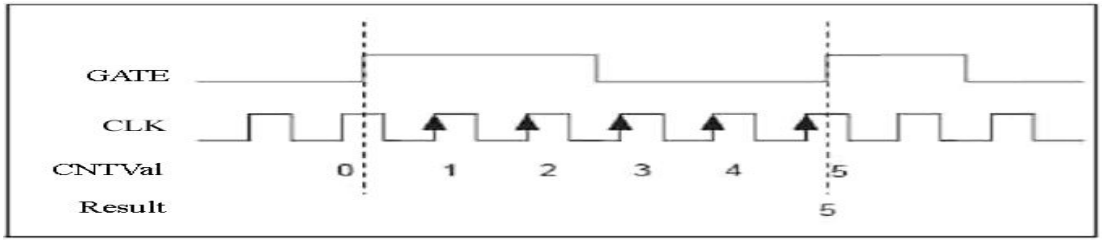
After the user writes the initial number and the count is allowed, the counter starts to count from the initial number adding "1" each time on the rising edge of the CLK when GATE generates the rising edge, and the count stops until the GATE generates the falling edge. The change of the follow-up GATE is invalid. This function is suitable for positive pulse width measurement.

Mode 6: the trigger count on the falling edge and the stop count on the rising edge



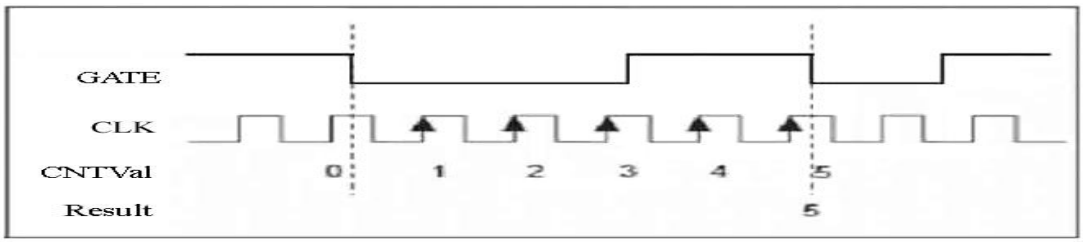
After the user writes the initial number and the count is allowed, the counter starts to count from the initial number adding "1" each time on the rising edge of the CLK when GATE generates the falling edge, and the count stops until the GATE generates the rising edge. The change of the follow-up GATE is invalid. This function is suitable for negative pulse width measurement. The measured signal should be connected to the GATE pin of the CN1, and the clock benchmark signal can be input from the CLK pin of the CN1 (You can also choose the LOCAL_CLK). When the counter does not overflow, we suggest that the high clock benchmark should be used as much as possible in order to improve the measurement precision.

Mode 7: the trigger count on the rising edge and the stop count on the next rising edge



After the user writes the initial number and the count is allowed, the counter starts to count from the initial number adding "1" each time on the rising edge of the CLK when GATE generates the rising edge, and the count stops until the GATE generates the next rising edge. The change of the follow-up GATE is invalid. This function is suitable for the whole cycle of the pulse width measurement.

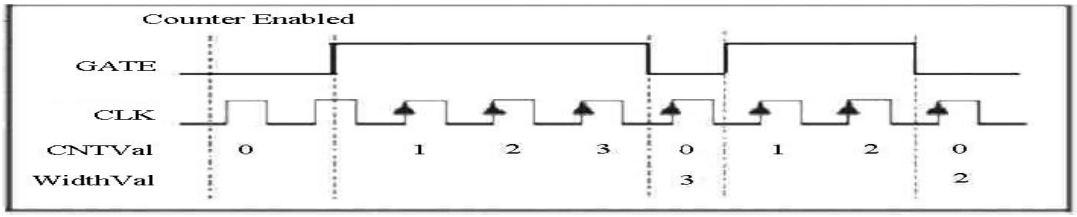
Mode 8: the trigger count on the falling edge and the stop count on the next falling edge



After the user writes the initial number and the count is allowed, the counter starts to count from the initial number adding "1" each time on the rising edge of the CLK when GATE generates the falling edge, and the count stops until the GATE generates the next falling edge. The change of the follow-up GATE is invalid. This function is suitable for the whole cycle of the pulse width measurement.

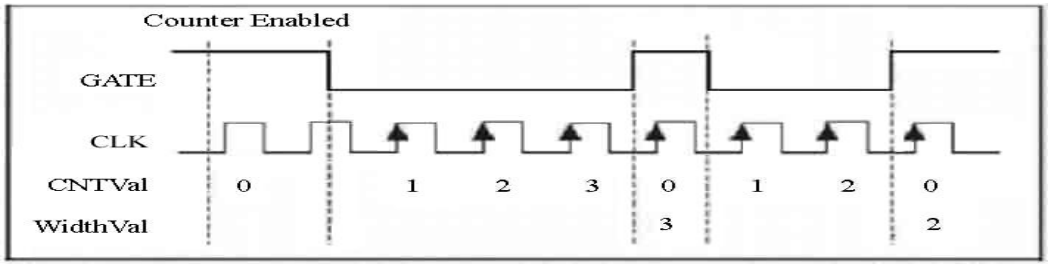
7.2.2 The functions of buffer count and time measurement

Mode 3: The high level is valid.



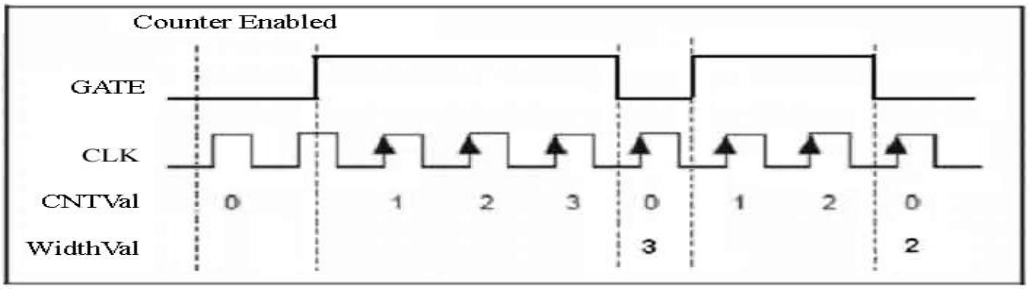
After the user writes the initial number and the count is allowed, the counter starts to count from the initial number adding "1" each time on the rising edge of the CLK when GATE is the high level, and when GATE turns to be the low level, the counter latches the count value to the buffer register for the user to read out, and at the same time to clear numerical terms. When GATE turns to be the high level again, the counter starts to count from 0 adding "1" each time, when GATE turns to be the low level, the counter latches the count value to the buffer register, and so is the following count mode.

Mode 4: The low level is valid.



After the user writes the initial number and the count is allowed, the counter starts to count from the initial number adding "1" each time on the rising edge of the CLK when GATE is the low level, and when GATE turns to be the high level, the counter latches the count value to the buffer register for the user to read out, and at the same time to clear numerical terms. When GATE turns to be the low level again, the counter starts to count from 0 adding "1" each time, when GATE turns to be the high level, the counter latches the count value to the buffer register, and so is the following count mode.

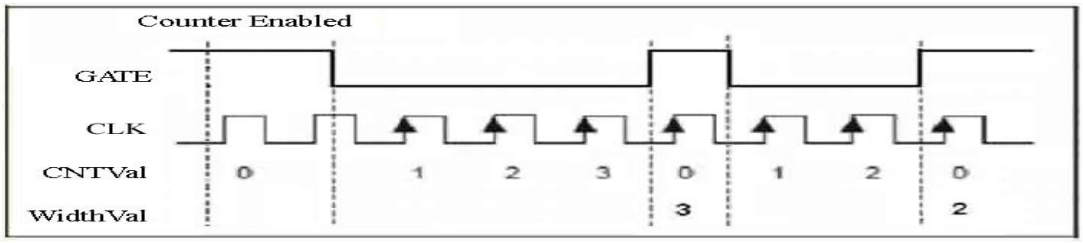
Mode 5: the trigger count on the rising edge and the stop count on the falling edge



After the user writes the initial number and the count is allowed, the counter starts to count from the initial number adding "1" each time on the rising edge of the CLK when GATE generates the rising edge, and the counter generates

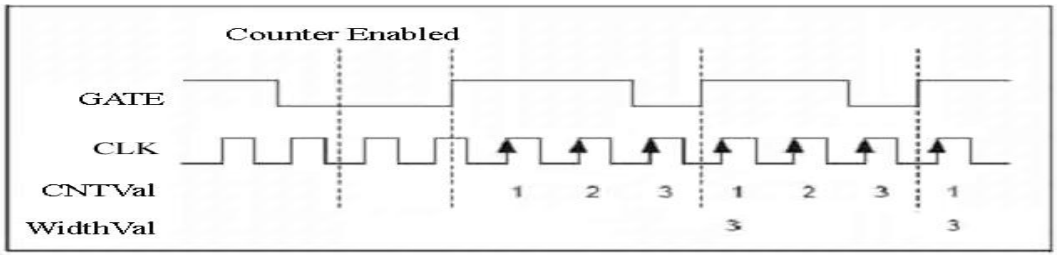
the falling edge, the counter latches the count value to the buffer register for the user to read out, and at the same time to clear numerical terms. When GATE generates the rising edge again, the counter starts to count from 0 adding "1" each time, when GATE turns to generate the falling edge, the counter latches the count value to the buffer register, and so is the following count mode.

Mode 6: the trigger count on the falling edge and the stop count on the rising edge



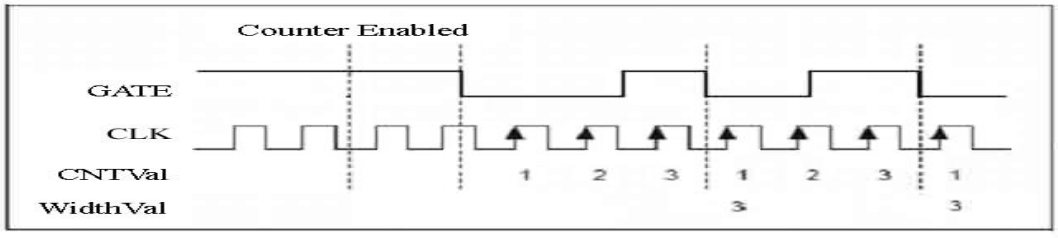
After the user writes the initial number and the count is allowed, the counter starts to count from the initial number adding "1" each time on the rising edge of the CLK when GATE generates the falling edge, and the counter generates the rising edge, the counter latches the count value to the buffer register for the user to read out, and at the same time to clear numerical terms. When GATE generates the falling edge again, the counter starts to count from 0 adding "1" each time, when GATE turns to generate the rising edge, the counter latches the count value to the buffer register, and so is the following count mode.

Mode 7: the trigger count on the rising edge and the stop count on the next rising edge



After the user writes the initial number and the count is allowed, the counter starts to count from 0 adding "1" each time on the rising edge of the CLK when GATE generates the rising edge, and when the counter generates the rising edge again, the counter latches the count value to the buffer register for the user to read out, and at the same time the counter starts to count from 0; When GATE generates the rising edge again, the counter latches the count value to the buffer register again, and at the same time the counter starts to count from 0 again, and so is the following count mode.

Mode 8: the trigger count on the falling edge and the stop count on the next falling edge



After the user writes the initial number and the count is allowed, the counter starts to count from 0 adding "1" each time

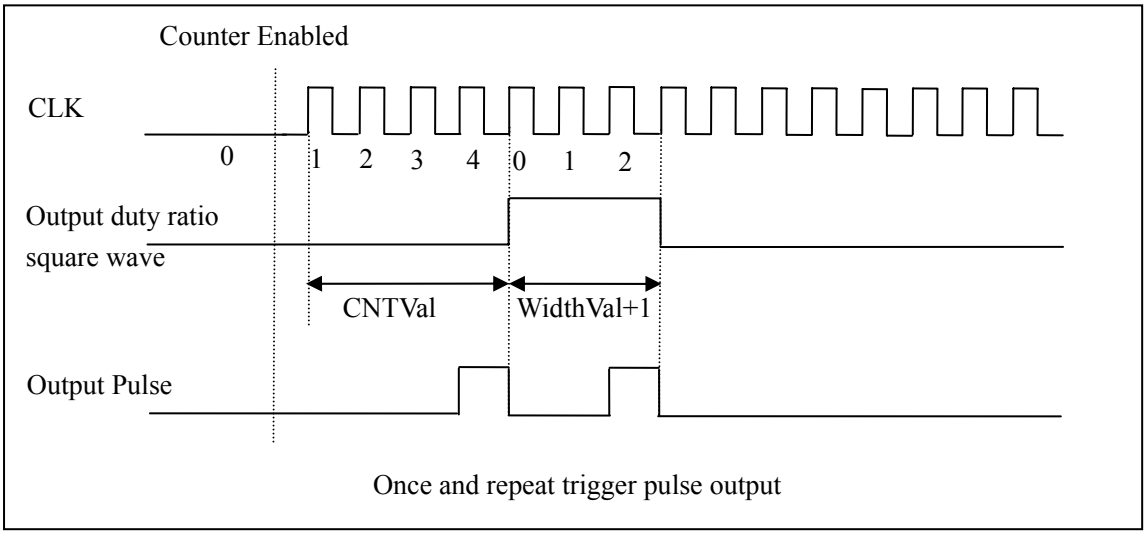
on the rising edge of the CLK when GATE generates the falling edge, and when the counter generates the falling edge again, the counter latches the count value to the buffer register for the user to read out, and at the same time the counter starts to count from 0; When GATE generates the falling edge again, the counter latches the count value to the buffer register again, and at the same time the counter starts to count from 0 again, and so is the following count mode.

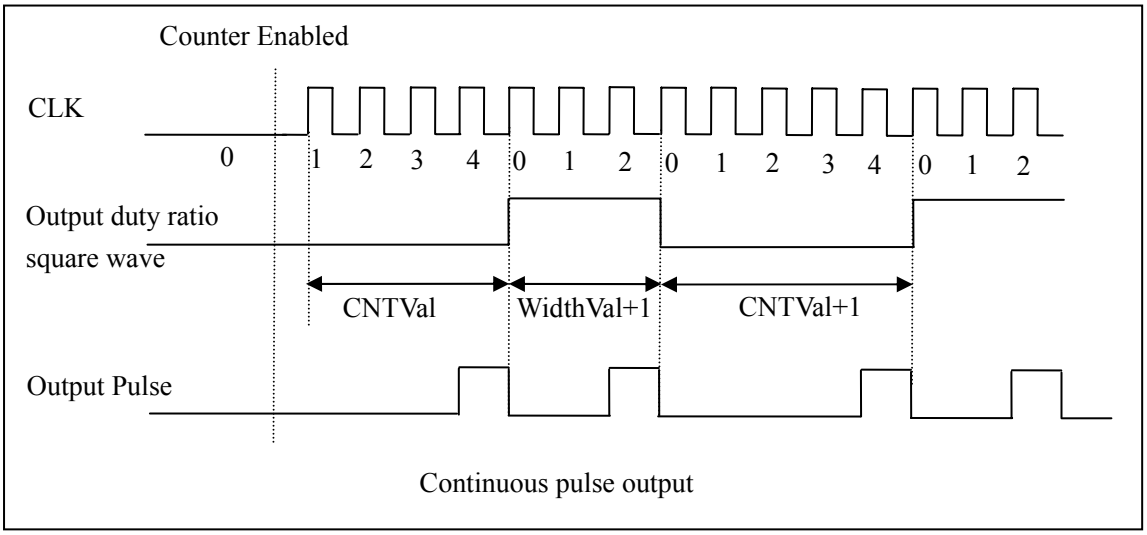
7.3 Pulse Generator Method

7.3.1The Output Types of the Pulse Generator

Under the pulse generator mode, the output types of OUT can be divided into the duty cycle square wave and the pulse. It needs to set two numerical numbers: initial count value (CNTVal) and pulse width (WidthVal). Initial count (CNTVal) can set the duration of low-level output; pulse width (WidthVal) can set the duration of high-level output. Each count time benchmark is decided by the CLK pulse cycle. The two parameters are set by the SetDeviceCNT in the software. The pulse generator outputs can be divided into one-shot trigger pulse output (mode 0 ~ 2), repeated trigger pulse output (mode 3 ~ 4) and continuous pulse train output (mode 5 ~ 8).

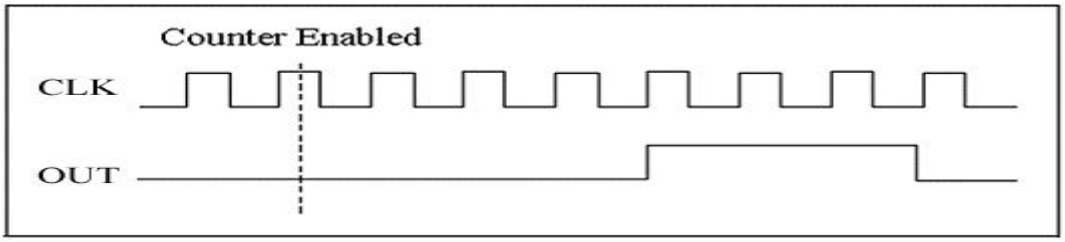
The designed initial count value (CNTVal) = 4, the designed pulse width (WidthVal) = 2





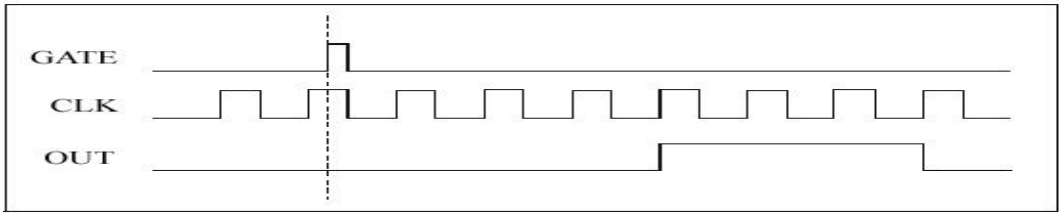
7.3.2 Functions of the Pulse Generator

Mode 0: Do not use the one-shot pulse generation of the GATE.



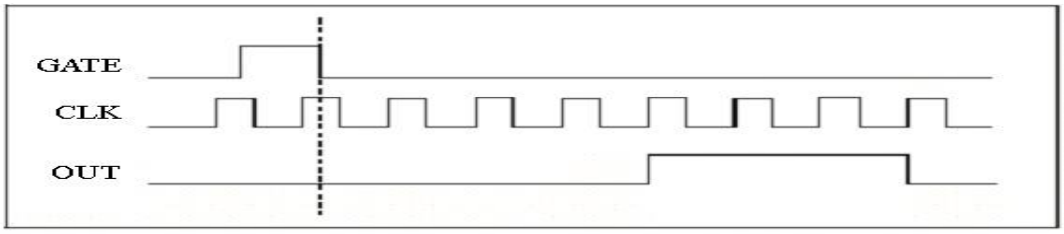
After the user writes the appointed initial count value and pulse width, and the count is allowed, the OUT output of the generator will turn into a delay progress of the low level. The delay time is decided by the initial count value parameter written by the users. After the delay finishes, it will output a high level. The high level delay time is decided by the pulse width written by the users. Later, the output keeps the low level. The mode is equivalent to start the occurrence of one pulse, which is started by users' software.

Mode 1: the one-shot pulse generation of the rising edge of GATE



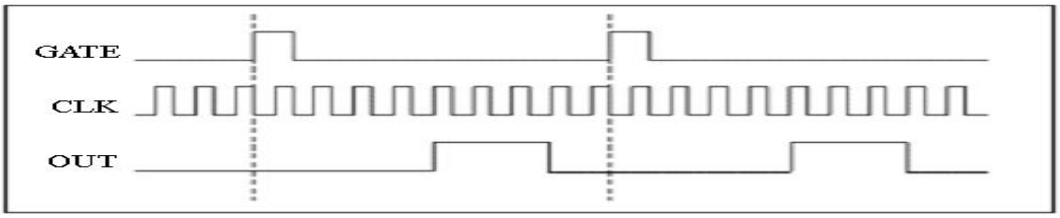
After the user writes the appointed initial count value and pulse width, and the count is allowed, when GATE has a rising edge, the OUT output of the generator will turn into a delay progress of the low level. The delay time is decided by the initial count value parameter written by the users. After the delay finishes, it will output a high level. The high level delay time is decided by the pulse width written by the users. Later, the output keeps the low level. In this process, other edges of GATE are invalid, so are the following edges. The mode is equivalent to start the occurrence of one pulse, which is started by the rising edge of an external hardware.

Mode 2：the one-shot trigger pulse generation of the falling edge of GATE



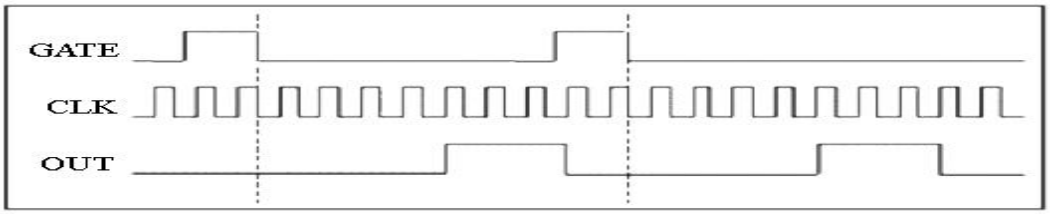
After the user writes the appointed initial count value and pulse width, and the count is allowed, when GATE has a falling edge, the OUT output of the generator will turn into a delay progress of the low level. The delay time is decided by the initial count value parameter written by the users. After the delay finishes, it will output a high level. The high level delay time is decided by the pulse width written by the users. Later, the output keeps the low level. In this process, other edges of GATE are invalid, so are the following edges. The mode is equivalent to start the occurrence of one pulse, which is started by the falling edge of an external hardware.

Mode 3：the repeated trigger pulse generation of the rising edge of GATE



This mode is similar to Mode 1, but the process is repeatedly controlled by GATE. After the user writes the appointed initial count value and pulse width, and the count is allowed, when GATE has a rising edge, the OUT output of the generator will turn into a delay progress of the low level. The delay time is decided by the initial count value parameter written by the users. After the delay finishes, it will output a high level. The high level delay time is decided by the pulse width written by the users. Later, the output keeps the low level. In this process, other edges of GATE are invalid, so are the following edges. The mode is equivalent to start the occurrence of one pulse, which is repeatedly started by the rising edge of an external hardware.

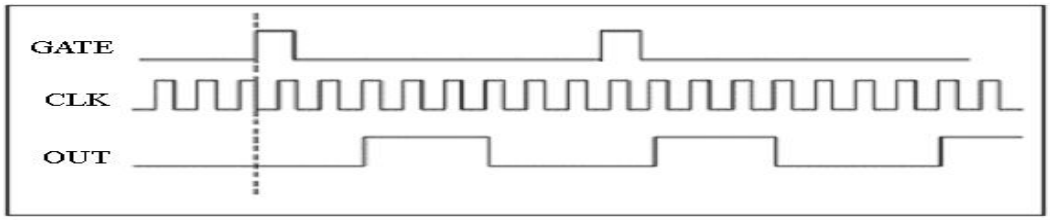
Mode 4：the repeated trigger pulse generation of the falling edge of GATE



This mode is similar to Mode 4, but the process is repeatedly controlled by GATE. After the user writes the appointed initial count value and pulse width, and the count is allowed, when GATE has a falling edge, the OUT output of the generator will turn into a delay progress of the low level. The delay time is decided by the initial count value parameter written by the users. After the delay finishes, it will output a high level. The high level delay time is decided by the

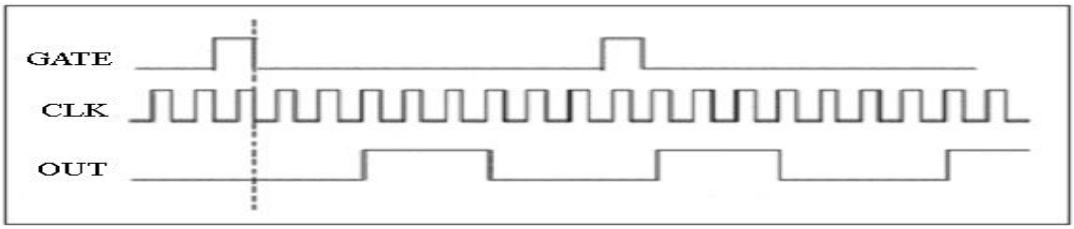
pulse width written by the users. Later, the output keeps the low level. In this process, other edges of GATE are invalid, so are the following edges. The mode is equivalent to start the occurrence of one pulse, which is repeatedly started by the falling edge of an external hardware.

Mode 5: the one-shot trigger continuous pulse train generator of the rising edge of GATE



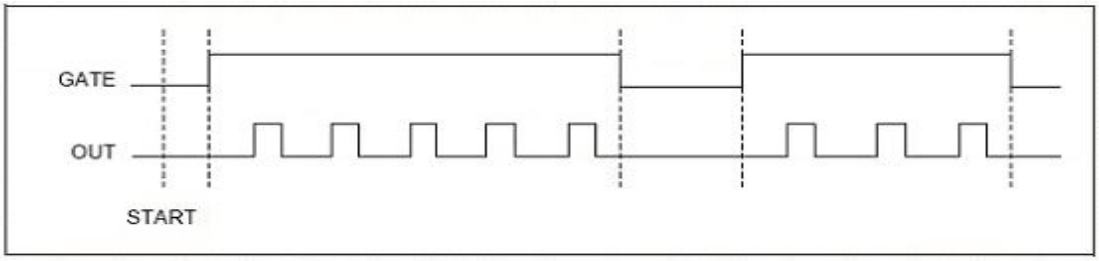
After the user writes the appointed initial count value and pulse width, and the count is allowed, when GATE has a rising edge, the OUT output of the generator will turn into a delay progress of the low level. The delay time is decided by the initial count value parameter written by the users. After the delay finishes, it will output a high level. The high level delay time is decided by the pulse width written by the users. Later, it turns into a delay progress of the low level, and then generates a continuous designated frequency and duty ratiopulse train. The mode is equivalent to start the following continuous pulse generation, which is started by the first rising edge of an external hardware.

Mode 6: the one-shot trigger continuous pulse train generator of the falling edge of GATE.



After the user writes the appointed initial count value and pulse width, and the count is allowed, when GATE has a falling edge, the OUT output of the generator will turn into a delay progress of the low level. The delay time is decided by the initial count value parameter written by the users. After the delay finishes, it will output a high level. The high level delay time is decided by the pulse width written by the users. Later, it turns into a delay progress of the low level, and then generates a continuous designated frequency and duty ratiopulse train. The mode is equivalent to start the following continuous pulse generation, which is started by the first falling edge of an external hardware.

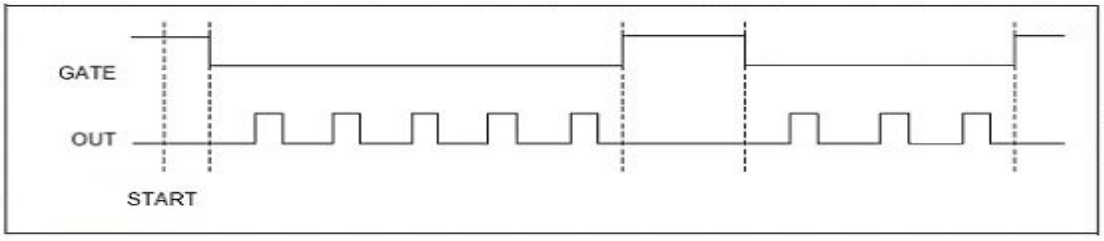
Mode 7: GATE high level permitted continuous pulse train generator



After the user writes the appointed initial count value and pulse width, and the count is allowed, when GATE is a high

level, the OUT output of the generator will turn into a delay progress of the low level. The delay time is decided by the initial count value parameter written by the users. After the delay finishes, it will output a high level. The high level delay time is decided by the pulse width written by the users. Later, it turns into a delay progress of the low level, and then generates a continuous designated frequency and duty ratiopulse train. When GATE is low level, then the OUT output immediately resets to the initial state. If GATE turns into high level again, then OUT continues to output. The mode is equivalent to start the continuous pulse output which is started by the rising edge of an external hardware and also unilateral high level keeps the pulse output state.

Mode 8: GATE low level permitted continuous pulse train generator



After the user writes the appointed initial count value and pulse width, and the count is allowed, when GATE is a low level, the OUT output of the generator will turn into a delay progress of the low level. The delay time is decided by the initial count value parameter written by the users. After the delay finishes, it will output a high level. The high level delay time is decided by the pulse width written by the users. Later, it turns into a delay progress of the low level, and then generates a continuous designated frequency and duty ratiopulse train. When GATE is high level, then the OUT output immediately resets to the initial state. If GATE turns into low level again, then OUT continues to output. The mode is equivalent to start the continuous pulse output which is started by the rising edge of an external hardware and also unilateral low level keeps the pulse output state.

Chapter 8 Notes, Calibration and Warranty Policy

8.1 Notes

In our products' packing, user can find a user manual, a PCI8602 module and a quality guarantee card. Users must keep quality guarantee card carefully, if the products have some problems and need repairing, please send products together with quality guarantee card to ART, we will provide good after-sale service and solve the problem as quickly as we can.

When using PCI8602, in order to prevent the IC (chip) from electrostatic harm, please do not touch IC (chip) in the front panel of PCI8602 module.

8.2 AD Analog Signal Input Calibration

Every device has to be calibrated before sending from the factory. It is necessary to calibrate the module again if users want to after using for a period of time or changing the input range. PCI8602 default input range: ±10V, in the manual, we introduce how to calibrate PCI8602 in ±10V, calibrations of other input ranges are similar.

Prepare a digital voltage instrument which the resolution is more than 5.5 bit, install the PCI8602 module, and then power on, warm-up for fifteen minutes.

1) Zero adjustment: select one channel of analog inputs, take the channel AI0for example, connect 0V to AI0, and then run ART Data Acquisition Measurement Suite in the WINDOWS. Choose channel 0, ±10V input range and start sampling, adjust potentiometer RP1 in order to make voltage value is 0.000V or about 0.000V. Zero adjustment of other channels is alike.
2) Full-scale adjustment: select one channel of analog inputs, take the channel AI0 for example, connect 9999.69mV to AI0, and then run ART Data Acquisition Measurement Suite in the WINDOWS. Choose channel 0, ±10V input range and start sampling, adjust potentiometer RP2 in order to make voltage value is 9999.69mV or about 9999.69mV. Full-scale adjustment of other channels is alike.
3) Repeat steps above until meet the requirement.

8.3 DA Analog Signal Output Calibration

In the manual, we introduce how to calibrate PCI8602 in ±10V range; calibrations of other input ranges are similar.

1) Connect the ground of the digital voltage meter to any analog AGND of the CN162 core D-type plug. Connect the input side of the voltage meter to the DA input channel which needs calibration. Run PCI8602 test procedure under Windows, select the DA output detection.
2) By adjusting the DA reference voltage to adjust potentiometer RP3 in order to make pin 6 of U29 output -2.5V.
3) To set DA output is 10V, adjust potentiometer RP4 in order to make AO0 output voltage value is 9995.11mV (adjust potentiometer RP5~RP7 to make AO1~AO3 output value is 9995.11mV).
4) Repeat steps above until meet the requirement.

8.4 DA use

In demonstration program, the continuous output interval of waveform output can not be carried out; the main objective is to test the strength of DA output.

8.5 Warranty Policy

Thank you for choosing ART. To understand your rights and enjoy all the after-sales services we offer, please read the following carefully.

1. Before using ART's products please read the user manual and follow the instructions exactly. When sending in damaged products for repair, please attach an RMA application form which can be downloaded from: www.art-control.com.
2. All ART products come with a limited two-year warranty:
   - The warranty period starts on the day the product is shipped from ART's factory
   - For products containing storage devices (hard drives, flash cards, etc.), please back up your data before sending them for repair. ART is not responsible for any loss of data.
   - Please ensure the use of properly licensed software with our systems. ART does not condone the use of pirated software and will not service systems using such software. ART will not be held legally responsible for products shipped with unlicensed software installed by the user.
3. Our repair service is not covered by ART's guarantee in the following situations:
   - Damage caused by not following instructions in the User's Manual.
   - Damage caused by carelessness on the user's part during product transportation.
   - Damage caused by unsuitable storage environments (i.e. high temperatures, high humidity, or volatile chemicals).
   - Damage from improper repair by unauthorized ART technicians.
   - Products with altered and/or damaged serial numbers are not entitled to our service.
4. Customers are responsible for shipping costs to transport damaged products to our company or sales office.
5. To ensure the speed and quality of product repair, please download an RMA application form from our company website.

Products Rapid Installation and Self-check

Rapid Installation

Product-driven procedure is the operating system adaptive installation mode. After inserting the disc, you can select the appropriate board type on the pop-up interface, click the button【driver installation】; or select CD-ROM drive in Resource Explorer, locate the product catalog and enter into the APP folder, and implement Setup.exe file. After the installation, pop-up CD-ROM, shut off your computer, insert the PCI card. If it is a USB product, it can be directly inserted into the device. When the system prompts that it finds a new hardware, you do not specify a drive path, the operating system can automatically look up it from the system directory, and then you can complete the installation.

Self-check

At this moment, there should be installation information of the installed device in the Device Manager (when the device does not work, you can check this item.). Open "Start -> Programs -> ART Demonstration Monitoring and Control System -> Corresponding Board -> Advanced Testing Presentation System", the program is a standard testing procedure. Based on the specification of Pin definition, connect the signal acquisition data and test whether AD is normal or not. Connect the input pins to the corresponding output pins and use the testing procedure to test whether the switch is normal or not.

Delete Wrong Installation

When you select the wrong drive, or viruses lead to driver error, you can carry out the following operations: In Resource Explorer, open CD-ROM drive, run Others-> SUPPORT-> PCI.bat procedures, and delete the hardware information that relevant to our boards, and then carry out the process of section I all over again, we can complete the new installation.