

**NuDAQ®**

**ACL-8112 Series**

**Enhanced Multi-Functions**

**Data Acquisition Cards**

**User's Guide**

©Copyright 1996~2000 ADLINK Technology Inc.  
All Rights Reserved.

Manual Rev. 3.51: June 7, 2000

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

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

This document contains proprietary information protected by copyright. All rights are reserved. No part of this manual may be reproduced by any mechanical, electronic, or other means in any form without prior written permission of the manufacturer.

### **Trademarks**

NuDAQ, ACL-8112 are registered trademark of ADLINK Technology Inc..

Other product names mentioned herein are used for identification purposes only and may be trademarks and/or registered trademarks of their respective companies.

# Getting service from ADLINK

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

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

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

Detailed Company Information			
Company/Organization			
Contact Person			
E-mail Address			
Address			
Country			
TEL		FAX	
Web Site			
Questions			
Product Model			
Environment to Use	<input type="checkbox"/> OS : _____ <input type="checkbox"/> Computer Brand : _____ <input type="checkbox"/> MB : _____ <input type="checkbox"/> CPU : _____ <input type="checkbox"/> Chipset : _____ <input type="checkbox"/> BIOS : _____ <input type="checkbox"/> Video Card : _____ <input type="checkbox"/> Network Interface Card : _____ <input type="checkbox"/> Other : _____		
Challenge Description			
Suggestions for ADLink			



# Table of Contents

<b>Chapter 1 Introduction</b> .....	<b>1</b>
1.1 Features .....	3
1.2 Applications .....	4
1.3 Specifications.....	4
1.4 Software Support .....	8
1.4.1 <i>Programming Library</i> .....	8
1.4.2 <i>LabView Driver</i> .....	8
<b>Chapter 2 Installation</b> .....	<b>9</b>
2.1 What You Have .....	9
2.2 Unpacking .....	10
2.3 ACL-8112's Layout .....	11
2.4 Jumper and DIP Switch Description.....	13
2.5 Base Address Setting .....	13
2.6 Analog Input Channel Configuration .....	15
2.7 DMA Channel Setting .....	16
2.8 Internal/External Trigger Setting .....	17
2.9 Clock Source Setting.....	18
2.10 IRQ Level Setting.....	18
2.11 D/A Reference Voltage Setting.....	19
2.12 A/D Input Range Setting .....	20
<b>Chapter 3 Signal Connections</b> .....	<b>21</b>
3.1 Connectors Pin Assignment .....	22
3.2 Analog Input Signal Connection .....	25
3.3 Analog Output Signal Connection .....	28
3.4 Digital I/O Connection.....	28
3.5 Timer / Counter Connection .....	29
3.6 Daughter Board Connection .....	30
3.6.1 <i>Connect with ACLD-8125</i> .....	30
3.6.2 <i>Connect with ACLD-9137</i> .....	30
3.6.3 <i>Connect with ACLD-9182</i> .....	30
3.6.4 <i>Connect with ACLD-9185</i> .....	30
3.6.5 <i>Connect with ACLD-9188</i> .....	30
<b>Chapter 4 Registers Format</b> .....	<b>31</b>

4.1	I/O Port Address .....	32
4.2	A/D Data Registers .....	33
4.3	A/D Channel Multiplexer Register .....	33
4.4	A/D Range Control Register .....	35
4.5	A/D Operation Mode Control Register.....	37
4.6	Interrupt Status Register.....	38
4.7	Software Trigger Register .....	38
4.8	Digital I/O register .....	39
4.9	D/A Output Register .....	40
4.10	Internal Timer/Counter Register .....	41
<b>Chapter 5 Operation Theorem .....</b>		<b>42</b>
5.1	A/D Conversion .....	42
	5.1.1 A/D Conversion Procedure.....	42
	5.1.2 A/D Trigger Modes.....	43
	5.1.3 A/D Data Transfer Modes.....	44
5.2	D/A Conversion .....	45
5.3	Digital Input and Output .....	45
5.4	Timer/Counter Operation .....	46
<b>Chapter 6 Calibration &amp; Utilities.....</b>		<b>49</b>
6.1	What do you need .....	49
6.2	VR Assignment .....	50
6.3	A/D Adjustment .....	50
	6.3.1 Bipolar Calibration.....	50
	6.3.2 Unipolar Calibration .....	51
6.4	D/A Adjustment .....	52
<b>C Language Library.....</b>		<b>53</b>
7.1	_8112_Initial .....	54
7.2	_8112_Switch_Card_No.....	56
7.3	_8112_DI.....	57
7.4	_8112_DI_Channel .....	57
7.5	_8112_DO.....	59
7.6	_8112_DA .....	60
7.7	_8112_AD_Input_Mode.....	61
7.8	_8112_AD_Set_Channel .....	63
7.9	_8112_AD_Set_Range .....	64
7.10	_8112_AD_Set_Mode .....	67
7.11	_8112_AD_Soft_Trig.....	69
7.12	_8112_AD_Aquire .....	70

7.13	_8112_CLR_IRQ.....	71
7.14	_8112_AD_DMA_Start .....	72
7.15	_8112_AD_DMA_Status .....	74
7.16	_8112_AD_DMA_Stop .....	75
7.17	_8112_AD_INT_Start .....	76
7.18	_8112_AD_INT_Status .....	77
7.19	_8112_AD_INT_Stop .....	78
7.20	_8112_AD_Timer.....	79
7.21	_8112_TIMER_Start .....	80
7.22	_8112_TIMER_Read .....	81
7.23	_8112_TIMER_Stop.....	81
<b>Appendix A. Demo Programs .....</b>		<b>82</b>
<b>Product Warranty/Service.....</b>		<b>83</b>

# How to Use This Guide

This manual is designed to help you use the ACL-8112. The manual describes how to modify various settings on the ACL-8112 card to meet your requirements. It is divided into seven chapters:

- **Chapter 1**, "Introduction," gives an overview of the product features, applications, and specifications.
- **Chapter 2**, "Installation," describes how to install the ACL-8112. The layout of ACL-8112 is shown, the switch setting for base address, and jumper setting for analog input channel configuration, reference voltage setting, trigger source, interrupt level and DMA channel are specified.
- **Chapter 3**, "Signal Connection," describes the connectors' pin assignment and how to connect the outside signal and devices with the ACL-8112.
- **Chapter 4**, "Registers," describes the details of register format and structure of the ACL-8112, this information is very important for the programmers who want to control the hardware by low-level programming.
- **Chapter 5**, "Operation Theorem" describes how to operate the ACL-8112. The A/D, D/A, DIO and timer/counter functions are introduced. Also, some programming concepts are specified.
- **Chapter 6**, "Calibration & Utility" describes how to calibrate the ACL-8112 for accurate measurement.
- **Chapter 7**, "C Language Library," describes how to program the ACL-8112 by using the C language library in DOS environment.
- **Appendix A**, "Demo. Program," describes some demonstration programs.





# Introduction

The ACL-8112 is a high performance, high speed multi-function data acquisition card for the IBM PC or compatible computers.

The ACL-8112 series is designed to combine all the data acquisition functions, such as A/D, D/A, DIO, and timer/counter in a single board. The high-end specifications of the card makes it ideal for wide range of applications requiring high speed 12-bit data acquisition at low cost. The Figure 1.1 shows the block diagram of the ACL-8112.

The ACL-8112 Series consists of three products, the ACL-8112HG, ACL-8112DG and ACL-8112PG. The ACL-8112HG provides special high-gain programmable instrument amplifier for low level input applications, such as measure thermocouple signals. The ACL-8112DG provides high speed sample rate ( up to 100 KHz) at all gains ( x1, x2, x4, and x8). The ACL-8112PG provides 16 single-ended inputs at up to 100 KHz and 5 levels of gain (x1, x2, x4, x8, x16)

The ACL-8112 Series features 16 single-ended inputs or 8 differential inputs at up to 90 KHz, 2 channels multiplying 12-bit double-buffered analog output, 16 digital inputs and 16 digital outputs, and one timer/counter channel.

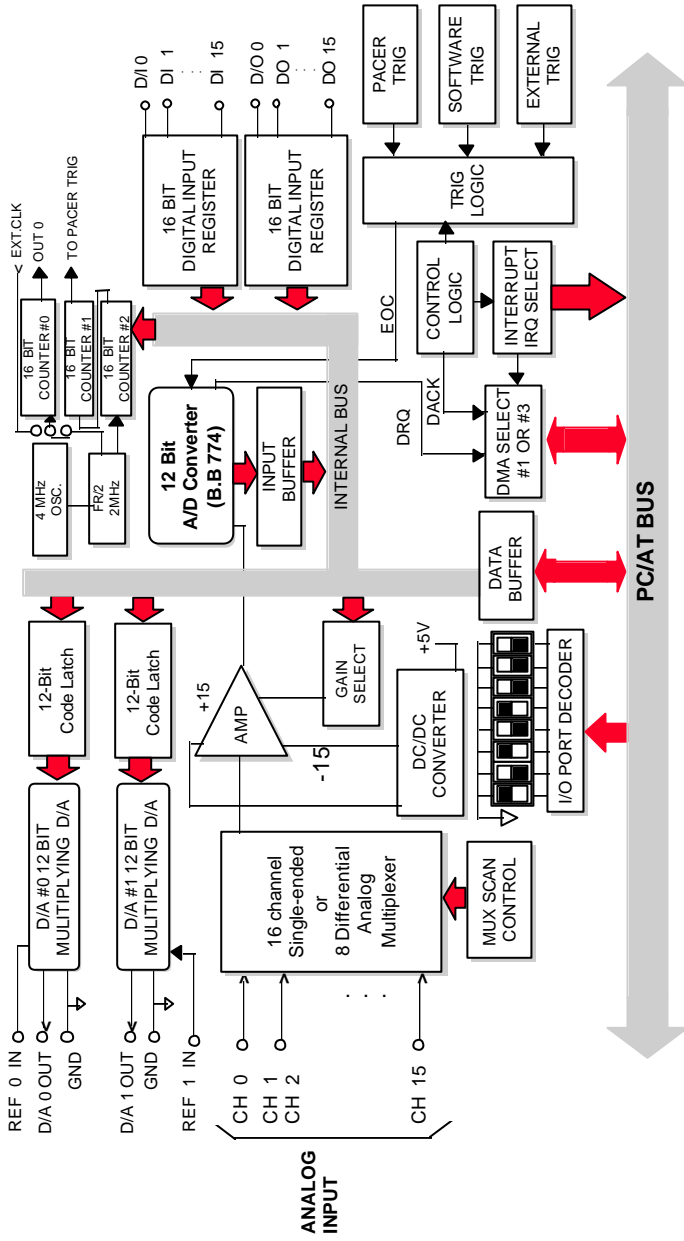


Figure 1.1 ACL - 8112 BLOCK DIAGRAM

---

## 1.1 Features

The ACL-8112 series Enhanced Multi-function Data Acquisition Card provides the following advanced features:

- AT-Bus
- 16 single-ended or 8 differential analog input channels for ACL-8112DG/HG, 16 single-ended for ACL-8112PG
- Bipolar or unipolar input signals for ACL-8112DG/HG, bipolar for ACL-8112PG
- Programmable gain
- High gain for ACL-8112HG:( x0.5, x1, x5, x10, x50, x100, x500, x1,000)
- Normal gain for ACL-8112DG (x0.5, x1, x2, x4, x8 )
- 5 Levels programmable gain for ACL-8112PG (x1, x2, x4, x8, x16), x0.5 gain can be set by jumper
- On-chip sample & hold
- Two 12-bit monolithic multiplying analog output channels
- 16 digital output channels
- 16 digital input channels
- 3 programmable 16-bit down counters
- Programmable sampling rate of up to 100KHz
- Three A/D trigger modes: software trigger, programmable pacer trigger, and external pulse trigger
- AT interrupt IRQ capability: 9 IRQ levels (IRQ3~IRQ15) are jumper selectable
- Integral DC-to-DC converter for stable analog power source
- 37-pin D-type connector
- Compact size: half-size PCB

---

## 1.2 Applications

- Industrial and laboratory ON/OFF control
- Energy management
- Annunciation
- 16 TTL/DTL compatible digital input channels
- Security controller
- Product test
- Period and pulse width measurement
- Event and frequency counting
- Waveform and pulse generation
- BCD interface driver

---

## 1.3 Specifications

### ◆ Analog Input (A/D)

- **Converter:** ADS774 or equivalent, successive approximation type
- **Resolution:** 12-bit
- **Number of channels:**  
ACL-8112DG/HG: 16 single-ended or 8 differential  
ACL-8112PG: 16 single-ended
- **Input Range:** (Programmable)
- **ACL-8112HG:**  
Bipolar :  $\pm 10V, \pm 5V, \pm 1V, \pm 500\text{ mV}, \pm 100\text{mV}, \pm 50\text{mV}, \pm 10\text{mV}, \pm 5\text{mV}$   
Unipolar: 0~10V, 0~1V, 0~0.1V, 0~0.01V
- **ACL-8112DG:**  
Bipolar :  $\pm 10V, \pm 5V, \pm 2.5V, \pm 1.25V, \pm 0.625$   
Unipolar: 0~10V, 0~5V, 0~2.5V, 0~1.25V
- **ACL-8112PG:**  
Bipolar :  $\pm 10V, \pm 5V, \pm 2.5V, \pm 1.25V, \pm 0.625V$   
Or  
Bipolar :  $\pm 5V, \pm 2.5V, \pm 1.25V, \pm 0.625V, \pm 0.3125V$

- **Conversion Time:** 8  $\mu$  sec
- **Overvoltage protection:** Continuous  $\pm$  35V maximum
- **Accuracy:**

(ACL-8112HG)

GAIN = 0.5, 1	0.01% of FSR $\pm$ 1 LSB
GAIN = 5, 10	0.02% of FSR $\pm$ 1 LSB
GAIN = 50, 100	0.04% of FSR $\pm$ 1 LSB
GAIN = 500, 1000	0.04% of FSR $\pm$ 1 LSB

(ACL-8112DG)

GAIN = 0.5, 1	0.01% of FSR $\pm$ 1 LSB
GAIN = 2, 4	0.02% of FSR $\pm$ 1 LSB
GAIN = 8	0.04% of FSR $\pm$ 1 LSB

(ACL-8112PG)

GAIN = 0.5, 1, 2, 4	0.015% of FSR $\pm$ 1 LSB
GAIN = 8, 16	0.02% of FSR $\pm$ 1 LSB

- **Input Impedance:** 10 M $\Omega$
- **AD conversion trigger modes:** Software, Pacer, and External trigger
- **Data Transfer:** Pooling, DMA, Interrupt
- **Sampling Rate:**
  - 100 KHz maximum for single channel
  - 100 KHz maximum for multiplexing on ACL-8112PG
  - 20 KHz maximum for multiplexing on ACL-8112DG/HG

#### Analog Output (D/A)

- **Converter:** DAC7541 or equivalent, monolithic multiplying
- **Number of channels:** 2 double-buffered analog outputs
- **Resolution:** 12-bit
- **Output Range:**
  - Internal reference: (unipolar) 0~5V or 0~10V
  - External reference: (unipolar) max. +10V or -10V
- **Settling Time:** 30  $\mu$  sec
- **Linearity:**  $\pm$ 1/2 bit LSB
- **Output driving capability:**  $\pm$ 5mA max.

#### ◆ Digital I/O (DIO)

- **Number of channels:** 16 TTL compatible inputs and outputs
- **Input Voltage:**  
Low: Min. 0V ; Max. 0.8V  
High: Min. +2.0V
- **Input Load:**  
Low: +0.5V @ -0.2mA max.  
High: +2.7V @+20mA max.
- **Output Voltage:**  
Low: Min. 0V ; Max. 0.4V  
High: Min. +2.4V
- **Driving Capacity:**  
Low: Max. +0.5V at 8.0mA ( Sink)  
High: Min. 2.7V at 0.4mA( Source)

#### ◆ Programmable Counter

- **Device:** 8254
- **A/D pacer:** 32-bit timer( two 16-bit counter cascaded together) with a 2MHz time base
- **Pacer Output:** 0.00046 Hz ~ 100 KHz
- **Counter:** One 16-bit counter with internal 2MHz time base or external clock source

#### ◆ General Specifications

- **I/O Base Address:** 16 consecutive address location
- **Interrupt IRQ:** IRQ3,5,6,7,9,10,11,12,15 (9 levels)
- **DMA Channel:** CH1 and CH3 (Jumper selectable)
- **Connector:** 37-pin D-type connector
- **Operating Temperature:** 0 °C ~ 55 °C
- **Storage Temperature:** -20 °C ~ 80°C
- **Humidity:** 5 ~ 95%, non-condensing
- **Power Consumption:**  
ACL-8112DG/HG: +5 V @ 430 mA typical

+12V @ 150 mA typical

ACL-8112PG: +5 V @ 450 mA typical

12 V @ 150 mA typical

- **Dimension:**

ACL-8112DG/HG: 162mm(L) x 115mm(W)

ACL-8112PG: 163 mm(L) x 123 mm(W)

---

## **1.4 Software Support**

### **1.4.1 Programming Library**

For the customers who are writing their own programs, we provide MS-DOS Borland C/C++ programming library.

ACLS-DLL2 is the Development Kit for NuDAQ ISA-Bus Cards with Analog I/O, windows 3.1/95(98)/NT. ACLS-DLL2 can be used for many programming environments, such as VC++, VB, Delphi. ACLS-DLL2 is included in the ADLINK CD. It need license.

### **1.4.2 LabView Driver**

The ACLS-LVIEW includes the ACL-8112's Vis, which is used to interface with NI's LabView software package. The ACLS-LVIEW supports Windows-95(98)/NT. ACLS-LVIEW is included in the ADLINK CD. It need license.



# 2

## Installation

This chapter describes how to install the ACL-8112 series products. Please follow the following steps to install.

- Check what you have (section 2.1)
- Unpacking (section 2.2)
- Check the PCB and jumper location(section 2.3)
- Install the hardware and setup and jumpers and switch (section 2.4~2.12)
- Cabling with external devices (section 2.13)

---

### 2.1 What You Have

In addition to this *User's Manual*, the package includes the following items:

- ACL-8112 Enhanced Multi-function Data Acquisition Card
- ADLINK CD

If any of these items is missing or damaged, contact the dealer from whom you purchased the product. Save the shipping materials and carton in case you want to ship or store the product in the future.

---

**Note:**The utilities and libraries in CD –ROM only support ACL-8112 series under DOS enviroment. If you need to develop application under Windows 3.1. , Windows 95 or Windows NT, please contact our dealer for purchasing software development kit ACLS-DLL2.

---

---

## 2.2 Unpacking

Your ACL-8112 card contains sensitive electronic components that can be easily damaged by static electricity.

The card should be done on a grounded anti-static mat. The operator should be wearing an anti-static wristband, grounded at the same point as the anti-static mat.

Inspect the card module carton for obvious damage. Shipping and handling may cause damage to your module. Be sure there are no shipping and handling damages on the module before processing.

After opening the card module carton, extract the system module and place it only on a grounded anti-static surface component side up.

Again inspect the module for damage. Press down on all the socketed IC's to make sure that they are properly seated. Do this only with the module place on a firm flat surface.

---

**Note:** DO NOT APPLY POWER TO THE CARD IF IT HAS BEEN DAMAGED.

---

***You are now ready to install your ACL-8112.***

### 2.3 ACL-8112's Layout

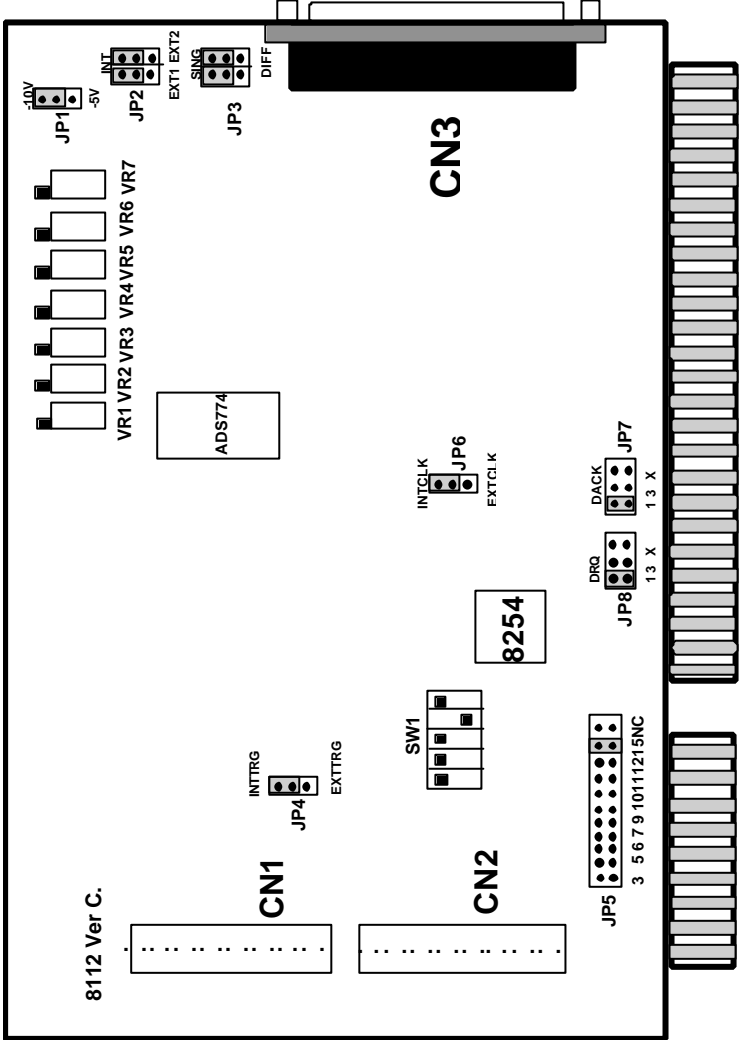


Figure 2.1-1 PCB Layout of the ACL-8112DG/HG Ver C.

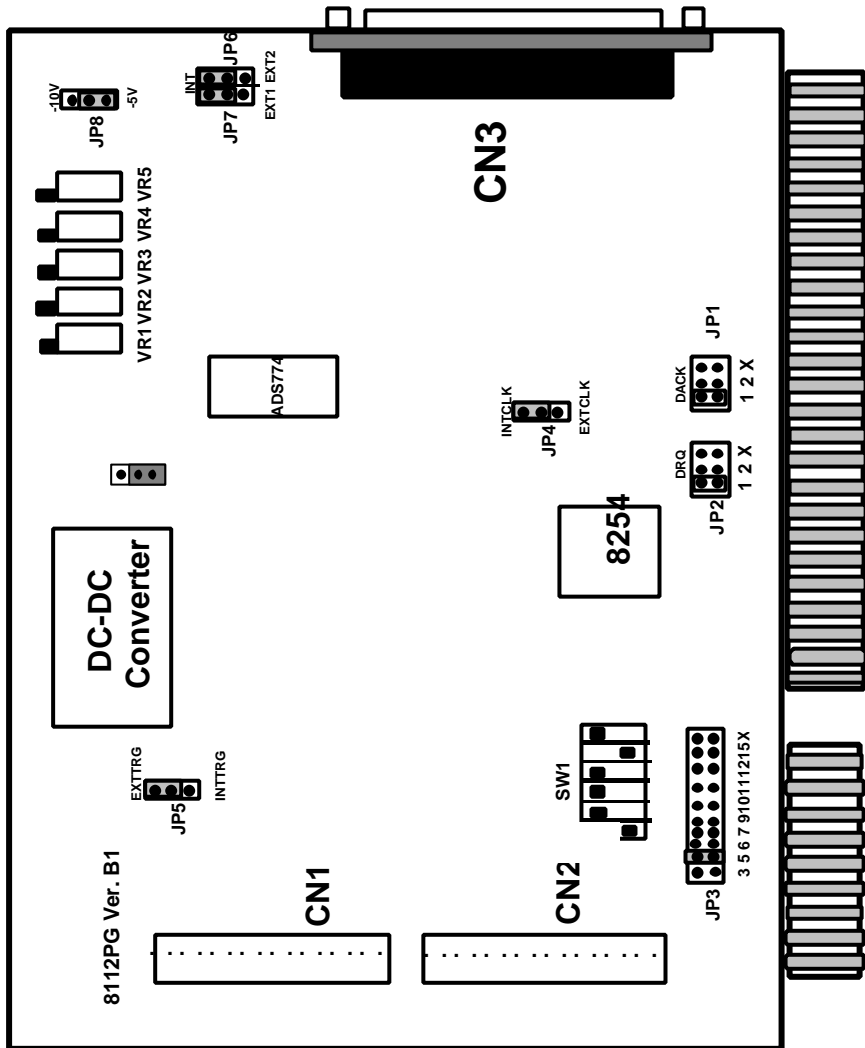


Figure 2.1-2 PCB Layout of the ACL-8112PG

---

## 2.4 Jumper and DIP Switch Description

You can change the ACL8112's channels and the base address by setting jumpers and DIP switches on the card. The card's jumpers and switches are preset at the factory. You can change the jumper settings for your own applications.

A jumper switch is closed (sometimes referred to as "shorted") with the plastic cap inserted over two pins of the jumper. A jumper is open with the plastic cap inserted over one or no pin(s) of the jumper.

---

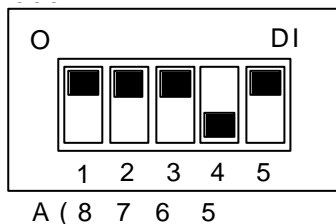
## 2.5 Base Address Setting

The ACL-8112 requires 16 consecutive address locations in the I/O address space. The base address of the ACL-8112 is restricted by the following conditions.

1. The base address must be within the range *Hex 200 to Hex 3FF*.
2. The base address should not conflict with any PC reserved I/O address. see Appendix A.
3. The base address must not conflict with any add-on card on your own PC. Please check your PC before installing the ACL-8112.

The ACL-8112's base address of registers is selected by an 6 positions DIP switch **SW1**. The default setting of base address is set to be **HEX 220**. All possible base address combinations are listed as Table 2.2. You may modify the base address if the address *HEX 220* has been occupied by another add-on card.

**SW1 : Base Address =**



**Figure 2.2 Default Base Address Setting**

I/O port Address(Hex)	A9	1 A8	2 A7	3 A6	4 A5	5 A4
200-20F	-- (1)	ON (0)	ON (0)	ON (0)	ON (0)	ON (0)
210-21F	-- (1)	ON (0)	ON (0)	ON (0)	ON (0)	OFF (1)
220-22F (default)	-- (1)	ON (0)	ON (0)	ON (0)	OFF (1)	ON (0)
230-23F	-- (1)	ON (0)	ON (0)	ON (0)	OFF (1)	OFF (1)
:						
300-30F	-- (1)	OFF (1)	ON (0)	ON (0)	ON (0)	ON (0)
:						
3F0-3FF	-- (1)	OFF (1)	OFF (1)	OFF (1)	OFF (1)	OFF (1)

A0, ..., A9 is corresponding to PC Bus address lines  
A9 is fixed as "1".

Table 2.2 Possible Base Address Combinations

How to define the base address for the ACL-8112 ?  
The DIP1 to DIP6 in the switch SW1 are one to one corresponding to the PC bus address line A8 to A4. A9 is always 1 and A0~A3 are always 0. If you want to change the base address, you can only change the values of A8 to A4 (the shadow area of below table). The following table is an example, which shows you how to define the base address as **Hex 220**

Base Address: **Hex 220**

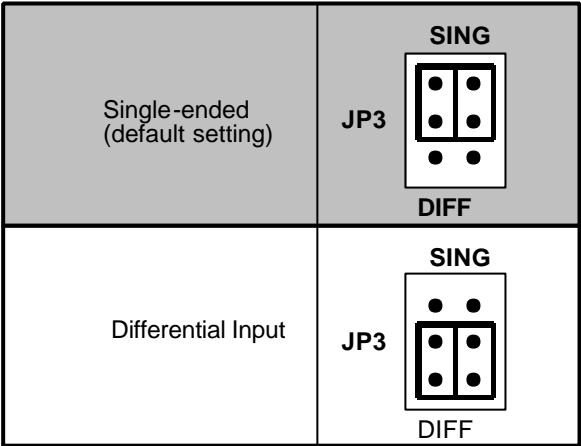
2		2				0			
1	0	0	0	1	0	0	0	0	0
A9	A8	A7	A6	A5	A4	A3	A2	A1	A0

---

## 2.6 Analog Input Channel Configuration

(This section is for ACL-8112DG and ACL-8112HG only.)

The ACL-8112 offer 16 single-ended or 8 differential analog input channels. The jumper JP3 controls the analog input channel configuration. The setting of JP3 is specified as following illustration.



**Figure 2.3 Analog Input Channels Configuration**

---

## 2.7 DMA Channel Setting

The A/D data transfer of ACL-8112 is designed with DMA transfer capability. The setting of DMA channel 1 or channel 3 is controlled by the jumpers JP7 and JP8 of ACL-8112DG/HG, or JP1 and JP2 of ACL-8112PG. The possible settings are shown below:

---

**Note:** On floppy disk only machine, we suggest you to set DMA level 3. If you have hard disk equipped computer, level 1 is preferable.

---

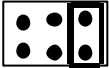
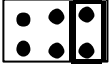


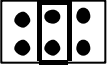
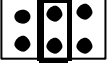
	DRQ JP8/JP2	DACK JP7/JP1
NO DMA	 1 3 X	 1 3 X
DMA 1 (Default)		
DMA 3		

Figure 2.4 DMA Channel Setting



---

## 2.8 Internal/External Trigger Setting

The A/D conversion trigger source of ACL-8112 comes from internal or external. The internal or external trigger source is setting by JP4 of ACL-8112DG/HG or JP5 of ACL-8112PG, as shown on Figure 2.5. Note that there are two internal on-board trigger sources, one is the software trigger and the other is the programmable pacer trigger, which is controlled by the mode control register(see section 4.5).

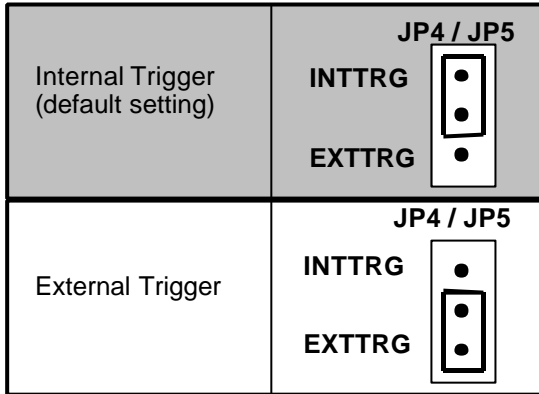


Figure 2.5 Trigger Source Setting

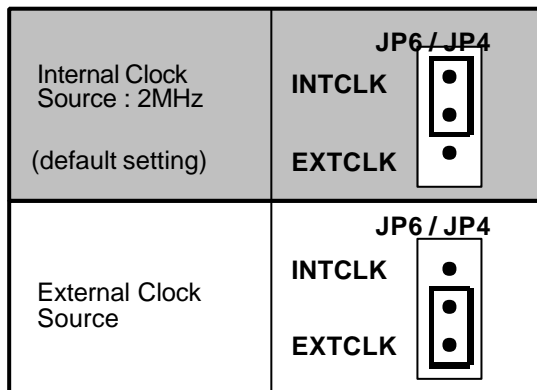


Figure 2.6 Timer's Clock Source Setting

---

## 2.9 Clock Source Setting

The 8254 programmable interval timer is used in the ACL-8112. It provides 3 independent 16-bit programmable down counters. The input of counter 2 is connected to a precision 2MHz oscillator for internal pacer. The input of counter 1 is cascaded from the output of counter 2. The channel 0 is free for user's applications. There are two selections for the clock source of channel 0: the internal 2MHz clock or the external clock signal from connector CN3 pin 37. The setting of clock is shown as Figure 2.6.

---

## 2.10 IRQ Level Setting

The ACL-8112 can connect to any one of the interrupt lines of the PC I/O channel. The interrupt line is selected by the jumper JP5 of ACL-8112DG/HG or JP3 of ACL-8112PG. If you wish to use the interrupt capability of ACL-8112, you must select an interrupt level and place the jumper in the appropriate position to enable the particular interrupt line.

The default interrupt level is IRQ5, which is selected by placing the jumper on the pins in row number 5. Figure 2.7 shows the default interrupt jumper setting IRQ5. You only remove the jumper from IRQ5 to other new pins, if you want to change to another IRQ level.

---

**Note:** Be aware that there is no other add-on card shares the same interrupt level at the same system.

---

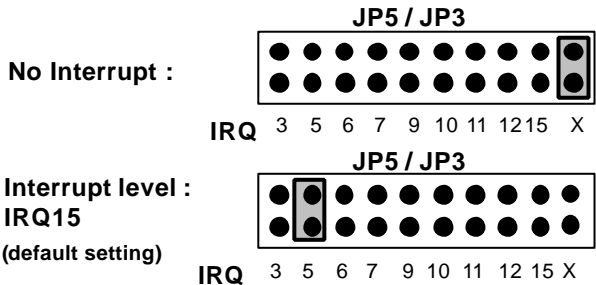


Figure 2.7 IRQ Level Setting

## 2.11 D/A Reference Voltage Setting

The D/A converter's reference voltage source can be internal or external generated. The external reference voltage comes from connector CN3 pin 31 (*ExtRef1*) and pin12 (*ExtRef2*), see section 3.1. The reference source of D/A channel 1 and channel 2 are selected by JP2 of ACL-8112DG/HG or JP6 and JP7 of ACL-8112PG respectively. Their possible settings are shown as below:









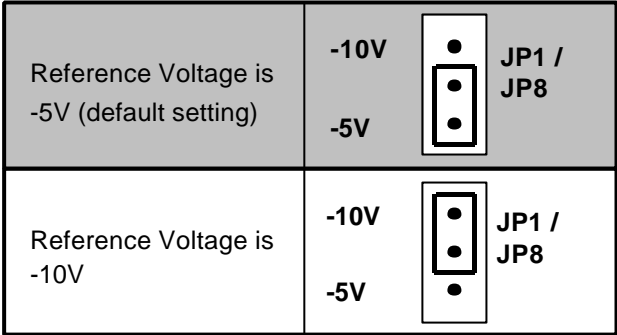
D/A CH1 is External D/A CH2 is External	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>JP2 or JP7</p> <p>INTREF</p> <p>ExtRef1</p>  </div> <div style="text-align: center;"> <p>JP6</p> <p>INTREF</p> <p>ExtRef2</p>  </div> </div>
D/A CH1 is External D/A CH2 is Internal	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>JP2 or JP7</p> <p>INTREF</p> <p>ExtRef1</p>  </div> <div style="text-align: center;"> <p>JP6</p> <p>INTREF</p> <p>ExtRef2</p>  </div> </div>
D/A CH1 is Internal D/A CH2 is External	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>JP2 or JP7</p> <p>INTREF</p> <p>ExtRef1</p>  </div> <div style="text-align: center;"> <p>JP6</p> <p>INTREF</p> <p>ExtRef2</p>  </div> </div>
D/A CH1 is Internal D/A CH2 is Internal	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>JP2 or JP7</p> <p>INTREF</p> <p>ExtRef1</p>  </div> <div style="text-align: center;"> <p>JP6</p> <p>INTREF</p> <p>ExtRef2</p>  </div> </div>

Figure 2.8 D/A Voltage Setting

The internal voltage is -5V or -10V which can be selected by JP1 of ACL-8112DG/HG or JP8 of ACL-8112PG. The possible configurations are specified as Figure 2.9. Note that the internal reference voltage is used only when the JP2 of ACL-8112DG/HG or JP6 and JP7 of ACL-8112PG is set to internal reference.



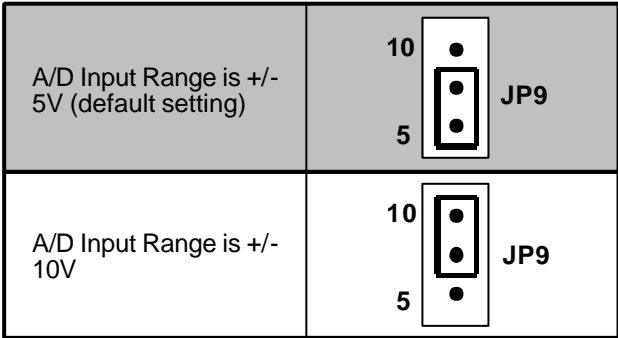
**Figure 2.9 Internal Reference Voltage Setting**

---

## 2.12 A/D Input Range Setting

(This section is for ACL-8112PG only)

The A/D input range of ACL-8112PG can be set to  $\pm 5V$  or  $\pm 10V$  by jumper JP9.



# 3

## Signal Connections

This chapter describes the connector of the ACL-8112, also the signal connection between the ACL-8112 and external devices, such as daughter boards or other devices.

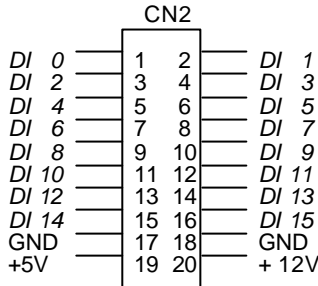
---

### 3.1 Connectors Pin Assignment

The ACL-8112 comes equipped with two 20-pin insulation displacement connectors - CN1 and CN2 and one 37-pin D-type connector - CN3. The CN1 and CN2 are located on board and CN3 located at the rear plate.

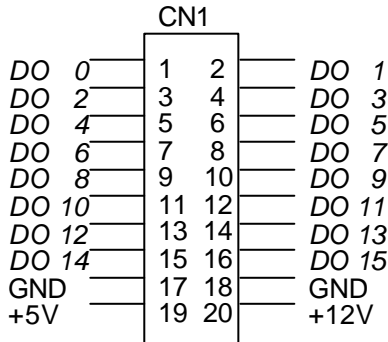
CN2 is used for digital signal input, CN1 for digital signal output, CN3 for analog input, analog output and timer/counter's signals. The pin assignment for each connectors are illustrated in the Figure 3.1 ~ Figure 3.3.

- **CN2:** Digital Signal Input (*DI 0 - 15*)



**Figure 3.1. Pin Assignment of CN2**

- **CN1:** Digital Signal Output (*DO 0 - 15*)



**Figure 3.2. Pin Assignment of CN1**

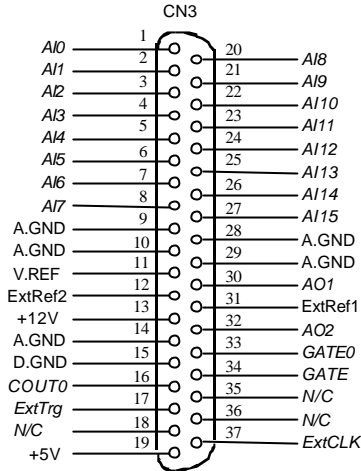
**Legend:**

*DO n*: Digital output signal channel *n*

*DI n*: Digital input signal channel *n*

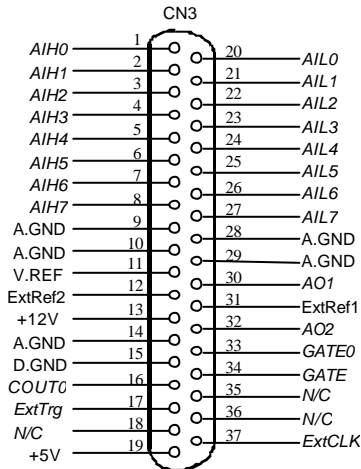
GND: Digital ground

- **CN3: Analog Input/Output & Counter/Timer**  
( for single-ended connection: ACL-8112DG/HG/PG)



**Figure 3.3a. Pin Assignment of CN3**

- **CN3: Analog Input/Output & Counter/Timer**  
( for differential connection:: ACL-8112DG/HG)



**Figure 3.3b. Pin Assignment of CN3**

**Legend:**

*AI<sub>n</sub>*: Analog Input Channel *n* ( single-ended)  
*AIH<sub>n</sub>*: Analog High Input Channel *n* ( differential)  
*AIL<sub>n</sub>*: Analog Low Input Channel *n* ( differential)  
*ExtRef<sub>n</sub>*: External Reference Voltage for D/A CH *n*  
*AO<sub>n</sub>*: Analog Output Channel *n*  
*ExtCLK*: External Clock Input  
*ExtTrig*: External Trigger Signal  
*CLK*: Clock input for 8254  
*GATE*: Gate input for 8254  
*COU<sub>T n</sub>*: Signal output of Counter *n*  
V.ERF: Voltage Reference  
A.GND: Analog Ground  
GND: Ground



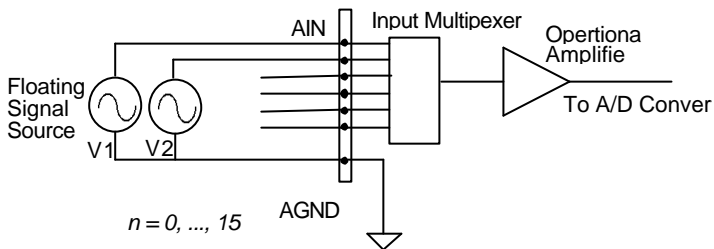
---

## 3.2 Analog Input Signal Connection

The ACL-8112 provides 16 single-ended or 8 differential analog input channels. The analog signal can be converted to digital value by the A/D converter. To avoid ground loops and get more accuracy measurement of A/D conversion, it is quite important to understand the signal source type and how to choose the analog input modes: signal-ended and differential. The ACL-8112 offers jumpers to select 16 single-ended or 8 different analog inputs.

### Single-ended Mode:

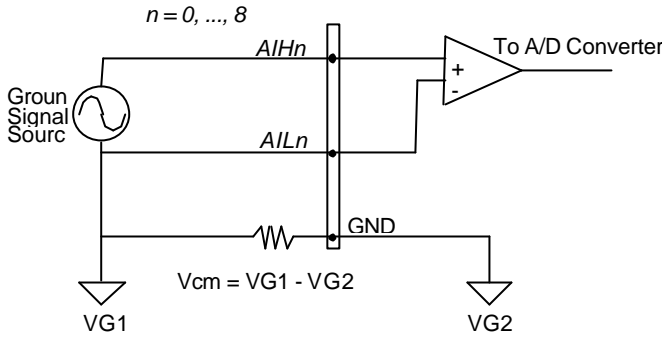
The single-ended mode has only one input relative to ground and it suitable for connecting with the *floating signal source*. The floating source means it does not have any connection to ground. Figure 3.4 shows the single-ended connection. Note that when more than two floating sources are connected, the sources must be with common ground.



**Figure 3.4 Floating source and single-ended**

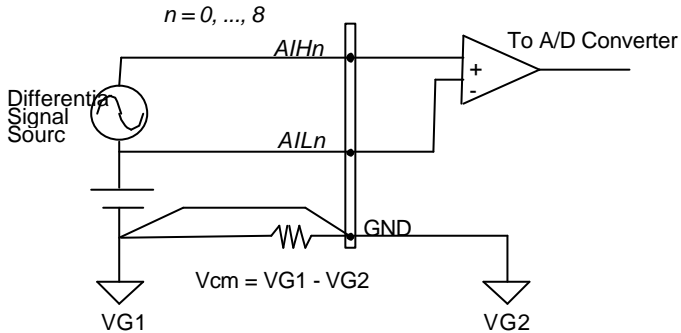
### Differential input mode

The differential input mode provides two inputs that respond to the difference signal between them. If the signal source has one side connected to local ground, the differential mode can be used for reducing ground loop. Figure 3.5 shows the connection of the differential input mode. However, even if the signal source is local grounded, the single-ended still can be used when the  $V_{cm}$  ( Common Mode Voltage) is very small and the effect of ground loop can be negated.



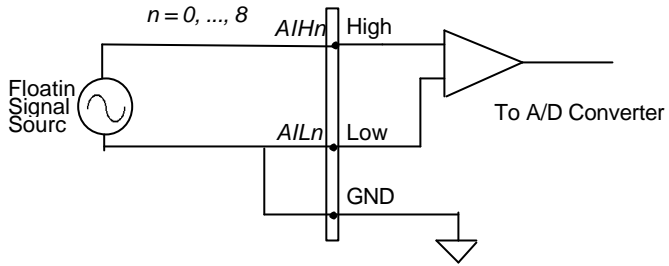
**Figure 3.5 Ground source and differential input**

A differential mode must be used when the signal source is differential. A differential source means the ends of the signal are not grounded. To avoid the danger of high voltage between the local ground of signal and the ground of the PC system, a shorted ground path must be connected. Figure 3.6 shows the connection of differential source.



**Figure 3.6 Differential source and differential input**

If your signal source is both floating and local ground, you should use the differential mode, and the floating signal source should be connected as the Figure 3.7 .



**Figure 3.7 Floating source and differential input**

---

### 3.3 Analog Output Signal Connection

The ACL-8112 has two unipolar analog output channels. To make the D/A output connections from the appropriate D/A output, please refer Figure 3.7.

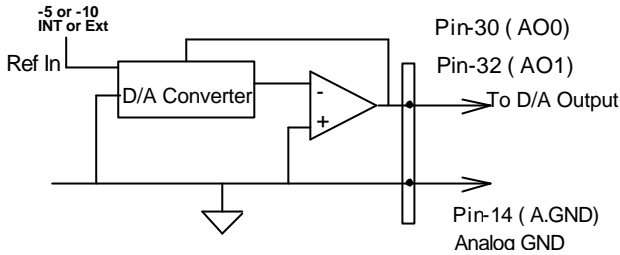


Figure 3.7 Connection of Analog Output Connection

---

### 3.4 Digital I/O Connection

The ACL-8112 provides 16 digital input and 16 digital output channels through the connector CN1 and CN2 on board. The digital I/O signal are fully TTL/DTL compatible. The detailed digital I/O signal specification can be referred in section 1.3.

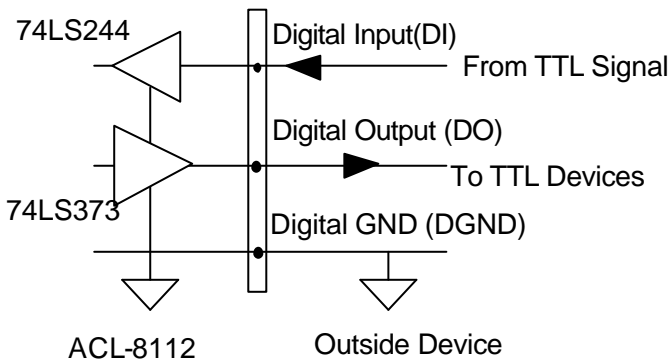
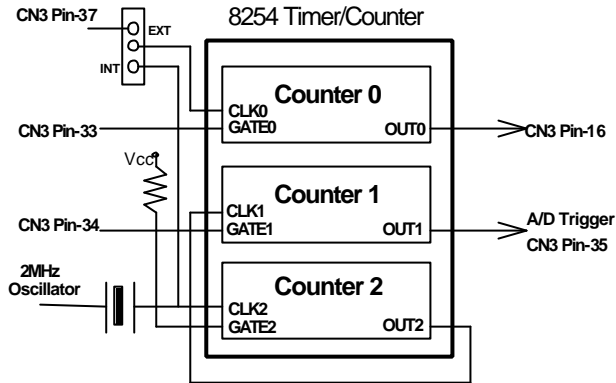


Figure 3.8 Digital I/O Connection

---

### 3.5 Timer / Counter Connection

The ACL-8112 has an interval timer/counter 8254 on board. It offers 3 independent 16-bit programmable down counters; counter 1 and counter 2 are cascaded together for A/D timer pacer trigger of A/D conversion, and counter 0 is free for your applications. Figure 3.9 shows the 8254 timer/counter connection.



**Figure 3.9 Block Diagram of 8254 Timer/Counter**

The clock source of counter 0 can be internal or external, while the gate can be controlled externally and the output is send to the connector CN3. As to counter 0 and counter 1, the clock source is internally fixed, while the gate can be controlled externally and the output is send to the connector CN3 too. All the timer/ counter signals are TTL compatible.

---

## **3.6 Daughter Board Connection**

The ACL-8112 can be connected with five different daughter boards, ACLD-8125, ACLD-9137, ACLD9182, ACLD9185, and ACLD9188. The functionality and connections are specified as follows.

### **3.6.1 Connect with ACLD-8125**

The ACLD-8125 has a 37-pin D-sub connector, which can connect with ACL-8112HG through 37-pin assemble cable. The most outstanding feature of this daughter board is a CJC ( cold junction compensation) circuit on board. You can directly connect the thermocouple on the ACL-8125 board.

### **3.6.2 Connect with ACLD-9137**

The ACLD-9137 is a direct connector for the card which is equipped with 37-pin D-sub connector. This board provides a simple way for connection. It is very suitable for the simple applications that do not need complex signal condition before the A/D conversion is performed.

### **3.6.3 Connect with ACLD-9182**

The ACLD-9182 is a 16 channel isolated digital input board. This board is connected with CN1 of ACL-8112 via 20-pin flat cable. The advantage of board is an 500Vdc isolation voltage is provided, and it can protect your PC system from damage when an abnormal input signal is occurred.

### **3.6.4 Connect with ACLD-9185**

The ACLD-9185 is a 16 channel SPDT relay output board. This board is connected with CN2 of ACL-8112 via 20-pin flat cable. by using this board, you can control outside device through the digital output signals.

### **3.6.5 Connect with ACLD-9188**

ACLD-9188 is a general purpose terminal board for all the card which comes equipped with 37-pin D-sub connector.

# 4

## Registers Format

The detailed descriptions of the register format and structure of the ACL-8112 are specified in this chapter. This information is quite useful for the programmer who wish to handle the card by low-level program. In addition, the low level programming syntax is introduced. This information can help the beginners to operate the ACL-8112 in the shortest learning time.

---

## 4.1 I/O Port Address

The ACL-8112 requires 16 consecutive addresses in the PC I/O address space. The Table 4.1 shows the I/O address of each register with respect to the base address. The function of each register also be shown.

I/O Address	Read	Write
Base + 0	Counter 0	Counter 0
Base + 1	Counter 1	Counter 1
Base + 2	Counter 2	Counter 2
Base + 3	Not Used	8254 Counter Control
Base + 4	A/D low byte	CH1 D/A low byte
Base + 5	A/D high byte	CH1 D/A high byte
Base + 6	DI low byte	CH2 D/A low byte
Base + 7	DI high byte	CH2 D/A high byte
Base + 8	Not Used	Clear Interrupt Request
Base + 9	Not Used	A/D Range Control
Base + 10	Not Used	Channel MUX
Base + 11	Not Used	Mode Control
Base + 12	Not Used	Software A/D trigger
Base + 13	Not Used	DO low byte
Base + 14	Not Used	DO high byte
Base + 15	Not Used	Not Used

**Table 4.1 I/O Address**



---

## 4.2 A/D Data Registers

The ACL-8112 series are 12-bit resolution for analog input channels, the digital data will store in the A/D data registers. The A/D data is put into two 8 bits registers. The low byte data (8 LSBs) are put in address BASE+4 and the high byte data (4 MSBs) are put in address BASE+5. A DRDY bit is used to indicate the status of A/D conversion. DRDY goes to low level means A/D conversion is completed.

Address : BASE + 4 and BASE + 5

Attribute: read only

Data Format:

Bit	7	6	5	4	3	2	1	0
BASE+4	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
BASE+5	0	0	0	DRDY	AD11	AD10	AD9	AD8

AD11 .... AD0: Analog to digital data. AD11 is the Most Significant Bit (MSB). AD0 is the Least Significant Bit(LSB).

DRDY: Data Ready Signal.

1: A/D data is not ready

0: A/D conversion is completed.

It will be set to 1, when reading the low byte.

---

## 4.3 A/D Channel Multiplexer Register

This register is used to control the A/D channels to be converted. It's a write only register. When the channel number is written to the register, the multiplexer switches to the new channel and await for conversion.

Address : BASE + 10

Attribute: write only

Data Format:

Bit	7	6	5	4	3	2	1	0
BASE+10	X	X	CS1	CS0	CL3	CL2	CL1	CL0

CLn: multiplexer channel number (n=0~3).

CL3 is MSB, and CL0 is LSB.

CS0, CS1: Single-ended and Differential Selection (ACL-8112DG/HG only)

CS0 and CS1 are used to determine which MPC508A chip is selected. The MPC508A is used to multiplex channel from channel, when CS0 is set as 1, the analog input channels from 0 to 7 is selectable, and CS1 is set, the ch 8 to ch 15 can be selectable. When both CS0 and CS1 are set as 1, it means the analog input is differential mode. The possible analog input channel selections are listed as the table below.

For ACL-8112PG, CS0 and CS1 always be zero, CL3–CL0 is used to select the 16 single-ended channels.

Bit Channel	7 X	6 X	5 CS1	4 CS0	3 CL3	2 CL2	1 CL1	0 CL0
S.E. CH0	X	X	0	1	0	0	0	0
S.E. CH1	X	X	0	1	0	0	0	1
S.E. CH2	X	X	0	1	0	0	1	0
S.E. CH3	X	X	0	1	0	0	1	1
S.E. CH4	X	X	0	1	0	1	0	0
S.E. CH5	X	X	0	1	0	1	0	1
S.E. CH6	X	X	0	1	0	1	1	0
S.E. CH7	X	X	0	1	0	1	1	1
S.E. CH8	X	X	1	0	1	0	0	0
S.E. CH9	X	X	1	0	1	0	0	1
S.E. CH10	X	X	1	0	1	0	1	0
S.E. CH11	X	X	1	0	1	0	1	1
S.E. CH12	X	X	1	0	1	1	0	0
S.E. CH13	X	X	1	0	1	1	0	1
S.E. CH14	X	X	1	0	1	1	1	0
S.E. CH15	X	X	1	0	1	1	1	1
D.I. CH0	X	X	1	1	0	0	0	0
D.I. CH1	X	X	1	1	0	0	0	1
D.I. CH2	X	X	1	1	0	0	1	0
D.I. CH3	X	X	1	1	0	0	1	1
D.I. CH4	X	X	1	1	0	1	0	0
D.I. CH5	X	X	1	1	0	1	0	1
D.I. CH6	X	X	1	1	0	1	1	0
D.I. CH7	X	X	1	1	0	1	1	1

**S.E.:** Single-ended Analog Input

**D.I.:** Differential Analog Input

## 4.4 A/D Range Control Register

The A/D range register is used to adjust the analog input ranges for A/D channels. Two factor will effect the input range: Gain and Bipolar/Unipolar. For ACL-8112PG, This register contols the PGA( programmable gain) directly and there is no Unipolar setting. When different gain value is set, the analog input range will be changed. For ACL-8112DG/HG, both of these issues can be controlled by this register. The Table 4.2 shows the relationship between the register data and the A/D input range.

Address : BASE + 9

Attribute: write only

Data Format:

Bit	7	6	5	4	3	2	1	0
BASE+9	X	X	X	X	G3	G2	G1	G0

G0~G3: Gain / Range selection, G3 is not used for ACL-8112PG

( This table is only for ACL-8112HG: High Gain Card)

G3	G2	G1	G0	GAIN	Bipolar or Unipolar	Input Range
0	0	0	0	1	Bipolar	±5V
0	0	0	1	10	Bipolar	±0.5V
0	0	1	0	100	Bipolar	±0.05V
0	0	1	1	1,000	Bipolar	±0.005V
0	1	0	0	1	Unipolar	0V ~ 10V
0	1	0	1	10	Unipolar	0V ~ 1V
0	1	1	0	100	Unipolar	0V ~ 0.1V
0	1	1	1	1,000	Unipolar	0V ~ 0.01V
1	0	0	0	0.5	Bipolar	±10V
1	0	0	1	5	Bipolar	±1V
1	0	1	0	50	Bipolar	±0.1V
1	0	1	1	500	Bipolar	±0.01V
1	1	0	0	1	Unipolar	N/A
1	1	0	1	10	Unipolar	N/A
1	1	1	0	100	Unipolar	N/A
1	1	1	1	1,000	Unipolar	N/A

Table 4.2-1 Function of the Gain Control Bits

( This table is only for ACL-8112DG: Low Gain Card)

G3	G2	G1	G0	GAIN	Bipolar or Unipolar	Input Range
1	0	0	0	0.5	Bipolar	$\pm 10V$
0	0	0	0	1	Bipolar	$\pm 5V$
0	0	0	1	2	Bipolar	$\pm 2.5V$
0	0	1	0	4	Bipolar	$\pm 1.25V$
0	0	1	1	8	Bipolar	$\pm 0.625V$
0	1	0	0	1	Unipolar	0V ~ 10V
0	1	0	1	2	Unipolar	0V ~ 5V
0	1	1	0	4	Unipolar	0V ~ 2.5V
0	1	1	1	8	Unipolar	0V ~ 1.25V

**Table 4.2-2 Function of the Gain Control Bits**

In the ACL-8112PG, the maximum can be changed by hardware jumper setting. The jumper JP9 is used to change the maximum analog input range from  $\pm 5V$  or  $\pm 10V$ . If the JP9 is set as  $\pm 5V$ , the analog input range is listed as below.

G3	G2	G1	G0	GAIN	Analog Input Range
0	0	0	0	1	$\pm 5V$
0	0	0	1	2	$\pm 2.5V$
0	0	1	0	4	$\pm 1.25V$
0	0	1	1	8	$\pm 0.625V$
0	1	0	0	16	$\pm 0.3125V$

**Table 4.2-3 Analog Input Range ( max. is  $\pm 5V$ )**

If the JP9 is set as  $\pm 5V$ , the analog input range is listed as below

G3	G2	G1	G0	GAIN	Analog Input Range
0	0	0	0	1	$\pm 10V$
0	0	0	1	2	$\pm 5V$
0	0	1	0	4	$\pm 2.5V$
0	0	1	1	8	$\pm 1.25V$
0	1	0	0	16	$\pm 0.625V$

**Table 4.2-4 Analog Input Range (max. is  $\pm 10V$ )**

---

## 4.5 A/D Operation Mode Control Register

The A/D operation includes the analog signal conversion and the data transformation. This register controls the internal trigger mode and data transformation method. It is initialized as software trigger and program polling transfer when your PC is reset or power on. The details of the A/D operation is illustrated in Chapter 5. There are four operation modes shown as following .

Address : BASE + 11

Attribute: write only

Data Format:

Bit	7	6	5	4	3	2	1	0
BASE+11	X	X	X	X	X	S2	S1	S0

S2	S1	S0	Operation Mode Description
0	0	0	Internal trigger is disable
0	0	1	software trigger and program polling (default)
0	1	0	timer pacer trigger and DMA transfer
1	1	0	timer pacer trigger and interrupt transfer.

---

### Note:

1. When your system power on or reset, the A/D operation will be initialized as " software trigger and program polling" mode.
  2. No matter which mode is selected, the external trigger is available if the JP4 is set to be external trigger.
  3. As long as not the DMA mode is not used, the program polling is always possible. The synchronization of A/D conversion and data transfer should be concerned when use program polling.
  4. The interrupt will be occurred after end of conversion if the "timer pacer trigger and interrupt transfer" mode is selected. If you want to use pacer trigger and interrupt transfer mode, please enable the IRQ level.
-

---

## 4.6 Interrupt Status Register

The Interrupt Status Register is used to clear the interrupt status for next new interrupt can be generated. If the ACL-8112 is in interrupt data transfer mode, a hardware status flag will be set after each A/D conversion. You have to clear the status flag by just writing any data to this register, let the ACL-8112 can generate next interrupt if a new A/D conversion is happen.

Address : BASE + 8

Attribute: write only

Data Format:

Bit	7	6	5	4	3	2	1	0
BASE+8	X	X	X	X	X	X	X	X

---

## 4.7 Software Trigger Register

If you want to generate a trigger pulse to the ACL-8112 for A/D conversion, you just write any data to this register, and then the A/D converter will be triggered.

Address : BASE + 12

Attribute: write only

Data Format:

Bit	7	6	5	4	3	2	1	0
BASE+12	X	X	X	X	X	X	X	X

---

---

## 4.8 Digital I/O register

There are 16 digital input channels and 16 digital output channels are provided by the ACL-8112. The address Base + 6 and Base + 7 are used for digital input channels, and the address Base + 13 and Base + 14 are used for digital output channels.

Address : BASE + 6 & BASE + 7

Attribute: read only

Data Format:

Bit	7	6	5	4	3	2	1	0
Base + 6	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
Base + 7	DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8

Address : BASE + 13 & BASE + 14

Attribute: write only

Data Format:

Bit	7	6	5	4	3	2	1	0
Base + 13	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0
Base + 14	DO15	DO14	DO13	DO12	DO11	DO10	DO9	DO8

---

## 4.9 D/A Output Register

The D/A converter will convert the D/A output register data to the analog signal. The register data of the address Base + 4 and Base + 5 are used for D/A channel 1, Base +6 and Base +7 are used for D/A channel 2.

Address : BASE + 4 & BASE + 5

Attribute: write only

Data Format: (for D/A Channel 1)

Bit	7	6	5	4	3	2	1	0
Base + 4	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
Base + 5	X	X	X	X	DA11	DA10	DA9	DA8

Address : BASE + 6 & BASE + 7

Attribute: write only

Data Format: (for D/A Channel 2)

Bit	7	6	5	4	3	2	1	0
Base + 6	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
Base + 7	X	X	X	X	DA11	DA10	DA9	DA8

DA0 is the LSB and DA11 is the MSB of the 12 bits data.

X: don't care

---

**Note:** The D/A registers are "**double buffered**" so that the D/A analog output signals will not updated until the second (**high**) byte is written. This can insure a single step transition when the D/A conversion.

---



---

## 4.10 Internal Timer/Counter Register

Two counter of 8254 are used for periodically triggering the A/D conversion, the left one is left free for user applications. The 8254 occupies 4 I/O address locations in the ACL-8112 as shown below. Users can refer to NEC's or Intel's data sheet for a full description of the 8254 features, condensed information is specified in Appendix B.

Address : BASE + 0 ~ BASE + 3

**Attribute:** read / write

Data Format:

Base + 0	Counter 0 Register ( R/W)
Base + 1	Counter 1 Register ( R/W)
Base + 2	Counter 2 Register ( R/W)
Base + 3	8254 CONTROL BYTE

# 5

## Operation Theorem

The operation theorem of the functions on ACL-8112 card is described in this chapter. The functions include the A/D conversion, D/A conversion, digital I/O and counter / timer. The operation theorem can help you to understand how to manipulate or to program the ACL-8112.

---

### 5.1 A/D Conversion

Before programming the ACL-8112 to perform the A/D conversion, you should understand the following issues:

- A/D conversion procedure
- A/D trigger mode
- A/D data transfer mode
- Signal Connection

#### 5.1.1 A/D Conversion Procedure

The A/D conversion is starting by a trigger source, then the A/D converter will start to convert the signal to a digital value. The ACL-8112 provides three trigger modes, see section 5.1.2.

While A/D conversion, the **DRDY** bit in A/D data register is cleared to indicate the data is not ready. After conversion being completed, the **DRDY** bit will return to low(0) level. It means users can read the converted data from the A/D data registers. Please refer section 4.2 for the A/D data format.

The A/D data should be transferred into PC's memory for further using. The ACL-8112 provides three data transfer modes that allow users to optimize the DAS system. Refer to section 5.1.3 for data transfer modes.

### **5.1.2 A/D Trigger Modes**

In the ACL-8112, A/D conversion can be triggered by the *Internal* or *External* trigger source. The jumper JP5 is used to select the internal or external trigger, please refer to section 2.8 for details. Whenever the external source is set, the internal sources are disable.

The two internal sources are the software trigger and the timer pacer trigger which is controlled by the A/D operation mode control register (BASE+11). Total three trigger sources are possible in the ACL-8112. The different trigger conditions are specified as follows:

#### **Software trigger**

The trigger source is software controllable in this mode. That is, the A/D conversion is starting when any value is written into the software trigger register (BASE+12). This trigger mode is suitable for low speed A/D conversion. Under this mode, the timing of the A/D conversion is fully controlled under software. However, it is difficult to control the fixed A/D conversion rate except another timer interrupt service routine is used to generate a fixed rate trigger.

#### **Timer Pacer Trigger**

An on-board timer / counter chip 8254 is used to provide a trigger source for A/D conversion at a fixed rate. Two counters of the 8254 chip are cascaded together to generate trigger pulse with precise period. Please refer to section 5.4 for 8254 architecture. This mode is ideal for high speed A/D conversion. It can be combined with the DMA or the interrupt data transfer. It's recommend to use this mode if your applications need a fixed and precise A/D sampling rate.

#### **External Trigger**

Through the pin-17 of CN3 (*ExtTrig*), the A/D conversion also can be performed when the a rising edge of external signal is occurred. The conversion rate of this mode is more flexible than the previous two modes, because the users can handle the external signal by outside device. The external trigger can combine with the DMA transfer, interrupt data transfer, or even program polling data transfer. Generally, the interrupt data transfer is often used when external trigger mode is used.

### 5.1.3 A/D Data Transfer Modes

On the ACL-8112, three A/D data transfer modes can be used when the conversion is completed. The data transfer mode is controlled by the mode control register (BASE+11). The different transfer modes are specified as follows:

#### Software Data Transfer

Usually, this mode is used with software A/D trigger mode. After the A/D conversion is triggered by software, the software should poll the *DRDY* bit until it becomes to high level. Whenever the low byte of A/D data is read, the *DRDY* bit will be cleared to indicate the data is read out.

It is possible to read A/D converted data without polling. The A/D conversion time will not exceed  $8\mu\text{s}$  on ACL-8112 card. Hence, after software trigger, the software can wait for at least  $8\mu\text{s}$  then read the A/D register without polling.

#### Interrupt Transfer

The ACL-8112 provides hardware interrupt capability. Under this mode, an interrupt signal is generated when the A/D conversion is ended and the data is ready to be read. It is useful to combine the interrupt transfer with the timer pacer trigger mode. Under this mode, the data transfer is essentially asynchronous with the control software.

When the interrupt transfer is used, you have to set the interrupt IRQ level by hardware jumper. Please refer section 2.10 for IRQ jumper setting. After the A/D conversion is completed, a hardware interrupt will be inserted and its corresponding ISR (Interrupt Service Routine) will be invoked and executed. The converted data is transferred by the ISR program.

#### DMA Transfer

The DMA (Direct Memory Access) allows data to be transferred directly between the ACL-8112 and the PC memory at the fastest possible rate, without using any CPU time. The A/D data is automatically transferred to PC's memory after conversion completed.

The DMA transfer mode is very complex to program. It is recommended to use the high level program library to operate this card. If you wish to program the software which can handle the DMA data transfer, please refer to more information about 8237 DMA controller.

---

## 5.2 D/A Conversion

The operation of D/A conversion is more simple than A/D operation. You only need to write digital values into the D/A data registers and the corresponding voltage will be output from the AO1 or AO2. Refer to section 4.9 for information about the D/A data registers. The mathematical relationship between the digital number  $DAn$  and the output voltage is formulated as following:

$$V_{out} = -V_{ref} \times \frac{DAn}{4096}$$

where the  $V_{ref}$  is the reference voltage, the  $V_{out}$  is the output voltage, and the  $DAn$  is the digital value in D/A data registers.

Before performing the D/A conversion, users should care about the D/A reference voltage which set by the JP1,JP2 and JP3. Please refer section 2.11 for jumper setting. The reference voltage will effect the output voltage. If the reference voltage is -5V, the D/A output scaling will be 0~-5V. If the reference voltage is -10V, the D/A output scaling will be 0~-10V.

Note that the D/A registers are "**double buffered**", so that the D/A analog output signals will not be updated until the high byte is written. When write 12 bits data to D/A registers of the ACL-8112, the low byte must be written before the high byte. This procedure can insure a single step transition when the D/A conversion.

---

## 5.3 Digital Input and Output

To program digital I/O operation is fairly straight forward. The digital input operation is just to read data from the corresponding registers, and the digital output operation is to write data to the corresponding registers. The digital I/O registers' format are shown in section 4.9. Note that the DIO data channel can only be read or written in form of 8 bits together. It is impossible to access individual bit channel.

---

## 5.4 Timer/Counter Operation

The ACL-8112 has an interval timer/counter 8254 on board. Refer to section 3.5 for the signal connection and the configuration of the counters.

### The 8254 Timer / Counter Chip

The Intel (NEC) 8254 contains three independent, programmable, multi-mode 16 bit counter/timers. The three independent 16 bit counters can be clocked at rates from DC to 5 MHz. Each counter can be individually programmed with 6 different operating modes by appropriately formatted control words. The most commonly uses for the 8254 in microprocessor based system are:

- programmable baud rate generator
- event counter
- binary rate multiplier
- real-time clock
- digital one-shot
- motor control

For more information about the 8254 , please refer to the NEC Microprocessors and peripherals or Intel Microsystems Components Handbook.

### Pacer Trigger Source

The counter 1 and counter 2 are cascaded together to generate the timer pacer trigger of A/D conversion. The frequency of the pacer trigger is software controllable. The maximum pacer signal rate is  $2\text{MHz}/4=500\text{K}$  which excess the maximum A/D conversion rate of the ACL-8112. The minimum signal rate is  $2\text{MHz}/65535/65535$ , which is a very slow frequency that user may never use it.

### General Purpose Timer/ Counter

The counter 0 is free for users' applications. The clock source, gate control signal and the output signal is send to the connector CN3. The general purpose timer / counter can be used as event counter, or used for measuring frequency, or others functions. See the 'Timer/Counter Applications' section for examples.

**I/O Address**

The 8254 in the ACL-8112 occupies 4 I/O address as shown below.

BASE + 0	LSB OR MSB OF COUNTER 0
BASE + 1	LSB OR MSB OF COUNTER 1
BASE + 2	LSB OR MSB OF COUNTER 2
BASE + 3	CONTROL BYTE

The programming of 8254 is control by the registers BASE+0 to BASE+3. The functionality of each register is specified this section. For more detailed information, please refer handbook of 8254 chip.

**Control Byte**

Before loading or reading any of these individual counters, the control byte (BASE+3) must be loaded first. The format of the control byte is:

Bit	7	6	5	4	3	2	1	0
	SC1	SC0	RL1	RL0	M2	M1	M0	BCD

◆ **SC1 & SC0 - Select Counter ( Bit7 & Bit 6)**

SC1	SC0	COUNTER
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	ILLEGAL

◆ **RL1 & RL0 - Select Read/Load operation ( Bit 5 & Bit 4)**

RL1	RL0	OPERATION
0	0	COUNTER LATCH FOR STABLE READ
0	1	READ/LOAD LSB ONLY
1	0	READ/LOAD MSB ONLY
1	1	READ/LOAD LSB FIRST, THEN MSB

◆ **M2, M1 & M0 - Select Operating Mode ( Bit 3, Bit 2, & Bit 1)**

M2	M1	M0	MODE
0	0	0	0
0	0	1	1
x	1	0	2
x	1	1	3
1	0	0	4
1	0	1	5

◆ **BCD - Select Binary/BCD Counting ( Bit 0)**

0	16-BITS BINARY COUNTER
1	BINARY CODED DECIMAL (BCD) COUNTER (4 DIGITAL)
<b>Note</b>	The count of the binary counter is from 0 up to 65,535 and the count of the BCD counter is from 0 up to 9,999

### Mode Definitions

In 8254, six operating modes can be selected. they are:

- **Mode 0:** Interrupt on Terminal Count
- **Mode 1:** Programmable One-Shot.
- **Mode 2:** Rate Generator
- **Mode 3:** Square Wave Rate Generator
- **Mode 4:** Software Triggered Strobe
- **Mode 5:** Hardware Triggered Strobe

All detailed description of these six modes are written in Intel Microsystems Components Handbook Volume II Peripherals.



# 6

## Calibration & Utilities

In data acquisition process, how to calibrate your measurement devices to maintain its accuracy is very important. Users can calibrate the analog input and analog output channels under the users' operating environment for optimizing the accuracy. This chapter will guide you to calibrate your ACL-8112 to an accuracy condition.

---

### 6.1 What do you need

Before calibrating your ACL-8112 card, you should prepare some equipment's for the calibration:

- Calibration program: Once the program is executed, it will guide you to do the calibration. This program is included in the delivered package.
- A 5 1/2 digit multimeter ( 6 1/2 is recommended)
- A voltage calibrator or a very stable and noise free DC voltage generator.

---

## 6.2 VR Assignment

There are 6 variable resistors (VR) on the ACL-8112DG/HG board to allow you making accurate adjustment on A/D and D/A channels. The function of each VR is specified as Table 6.1-1.

VR1	A/D bipolar offset adjustment
VR2	A/D full scale adjustment
VR3	D/A channel 1 full scale adjustment
VR4	D/A channel 2 full scale adjustment
VR5	A/D programmable amplifier offset adjustment
VR6	A/D unipolar offset adjustment

**Table 6.1-1 Function of VRs**

There are five variable resistors (VR) on the ACL-8112PG board to allow you making accurate adjustment on A/D and D/A channels. The function of each VR is specified as Table 6.1-2.

VR1	D/A channel 1 full scale adjustment
VR2	D/A channel 2 full scale adjustment
VR3	A/D full scale adjustment
VR4	A/D programmable amplifier offset adjustment
VR5	A/D offset adjustment

**Table 6.1-2 Function of VRs**

---

## 6.3 A/D Adjustment

### 6.3.1. Bipolar Calibration

1. Set the analog input range as: +/- 5V, i.e. the gain = 1 and input mode = Bipolar.
2. Short the A/D channel 0 ( pin 1 of CN3) to ground(GND), and connect the TP1(+) and TP2(-) with your DVM. Trim the variable resistor VR5 to obtain a value as close as possible to 0V.( For ACL-8112DG, users can skip this step)
3. Applied a +5V input signal to A/D channel 0, and trim the VR2 to obtain reading between 4094~4095.
4. Applied a +0V input signal to A/D channel 0, and trim the VR1 to obtain reading flickers between 2048~2049.

- 5 Repeat step 2 and step 3, adjust VR2 and VR1.

### **6.3.2 Unipolar Calibration**

1. Set the analog input range as: 0 ~ 10 V, i.e. the gain = 1 and input mode = Unipolar.
2. Applied a +10V input signal to A/D channel 0, and trim the VR6 to obtain reading flickers between 4094 ~ 4095.

---

## 6.4 D/A Adjustment

There are two steps to calibrate the analog output channels, D/A 1 and D/A 2. The first step is to adjust the reference voltage, and the second step is to adjust each channel of D/A.

### D/A Channel Calibration

D/A CH1 calibration:

1. Connect VDM (+) to CN3 pin-30 ( AO1) and VDM(-) to A.GND.
2. Write the digital value 0x0FFF into registers ( BASE+ 4 and BASE+ 5).
3. Trim the variable resister VR3 to obtain +5V reading in the DVM.

D/A CH2 calibration:

1. Connect VDM (+) to CN3 pin-32 ( AO2) and VDM(-) to A.GND.
2. Write the digital value 0x0FFF into registers ( Base + 6 and + 7).
3. Trim the variable resister VR4 to obtain +5V reading in the DVM.

A calibration utility is supported in the software diskette which is included in the product package. The detailed calibration procedures and description can be found in the utility. Users only need to run the software calibration utility and follow the procedures. You will get the accurate measure data.

# 7

## C Language Library

There are 23 function calls provided by the C Language Library, all the functions of ACL-8112 are covered by this library, its capabilities include A/D conversion, D/A conversion, Digital Input and Output, etc.

Using the C Language library saves a lot of programming time and has some other benefits as well. The library also supports data collection on interrupt or DMA from internal time clock for A/D conversion. Note that the DMA data transfer just processes on one fixed A/D channel.

A lot of demonstration programs are included in this disk. It will help you understand the library more quickly. The detailed description of each function in the library is specified in the following sections.

Please note that the function name is using the following conventions:

`_8112XX_function()`. For ACL-8112DG and ACL-8112HG, the function name is, for example, `_8112_function()`. For ACL-8112PG, the function name is, for example, `_8112pg_function()`.

---

## 7.1 `_8112_Initial`

### **Description**

An ACL-8112 card is initialized according to the card number and the corresponding base address. Every ACL-8112 Multi-Function Data Acquisition Card have to be initialized by this function before calling other functions.

### **Syntax**

```
int _8112_Initial(int card_number, int type, int
                 base_address )
int _8112pg_Initial(int card_number, int
                   base_address )
```

### **Argument:**

**card\_number:** the card number to be initialized, only two cards can be initialized, the card number must be `CARD_1` or `CARD_2`.

**Type** :there are 4 different types of ACL-8112 cards,

they are,

`A8112B_HG`: 8112 High Gain card Ver. B

`A8112B_DG`: 8112 Low Gain card Ver. B

`A8112C_HG`: 8112 High Gain card Ver. C

`A8112C_DG`: 8112 Low Gain card Ver. C

Note: the difference between Ver.B and Ver.C is Multi-Scan Register. The control code for different version has a little difference. For details, please refer hardware manual for each ACL-8112 card.

**base\_address:** the I/O port base address of the card, the default address on the hardware setting is Hex 220.

### **Return Code:**

```
ERR_NoError
ERR_InvalidBoardNumber
ERR_BaseAddressError
```

### **Example:**

```
#include "8112.h"

main()
{
    int ErrCode;
```

```
Errcode = _8112_Initial( CARD_1, A8112B_HG, 0x210 );
if ( ErrCode != ERR_NoError )
    exit(0);

Errcode = _8112_Initial( CARD_2, A8112B_DG, 0x220 );
if ( ErrCode != ERR_NoError )
    exit(0);
.
.
}
```

---

## 7.2 `_8112_Switch_Card_No`

### *Description*

This function is used on dual-cards system. After initialized two ACL-8112 cards, this function is used to select which card is used currently.

---

**Note:** In this library, only two ACL-8112 can be initialized. The reason is only two DMA channels are supported in the card.

---

### *Syntax*

```
int _8112_Switch_Card_No(int card_number)
int _8112pg_Switch_Card_No(int card_number)
```

### *Argument:*

**card\_number:** The card number to be initialized, only two cards can be initialized, the card number must be `CARD_1` or `CARD_2`.

### *Return Code:*

```
ERR_NoError
ERR_InvalidBoardNumber
```

### *Example:*

```
#include "8112.h"

main()
{
    _8112_Initial( CARD_1, A8112B_HG, 0x210 );
    _8112_Initial( CARD_2, A8112B_DG, 0x220 );
    /* Assume NoError when Initialize ACL-8112 */

    _8112_Switch_Card_No( CARD_1 );
    /*..... You can perform certain functions
    to Card_1 here*/

    _8112_Switch_Card_No( CARD_2 );
    /*..... You can perform certain functions
    to Card_2 here*/
}
```



---

## 7.3 \_8112\_DI

### *Description*

This function is used to read data from digital input port. There are 16-bit digital inputs on the ACL-8112. The bit 0 to bit 7 are defined as **low byte** and the bit 8 to bit 15 are defined as the **high byte**.

### *Syntax*

```
int _8112_DI( int port_number, unsigned char *data )
int _8112pg_DI( int port_number, unsigned char *data )
```

### *Argument:*

**port\_number:** To indicate which port is read,  
DI\_LO\_BYTE  
or DI\_HI\_BYTE.  
DI\_LO\_BYTE: bit 0 ~ bit 7,  
DI\_HI\_BYTE: bit8 ~ bit15

**data:** return value from digital port.

### *Return Code:*

```
ERR_NoError
ERR_BoardNoInit
ERR_PortError
```

### *Example:*

See Demo program . Demo Program 'DI\_DEMO.C'

---

## 7.4 \_8112\_DI\_Channel

### *Description*

This function is used to read data from digital input channels (bit). There are 16 digital input channels on the ACL-8112. When performs this function, the digital input port is read and the value of the corresponding channel is returned.

\* channel means each bit of digital input ports.

### *Syntax*

```
int _8112_DI_Channel(int di_ch_no, unsigned int
*data )
int _8112pg_DI(int di_ch_no, unsigned int *data )
```

**Argument:**

**di\_ch\_no:** the DI channel number, the value has to be set from 0 to 15.  
**data:** return value, either 0 or 1.

**Return Code:**

ERR\_NoError  
ERR\_BoardNoInit  
ERR\_InvalidDIChannel

**Example:**

```
#include "8112.h"

main()
{
    unsigned int data;
    int ch;

    _8112_Initial( CARD_1, A8112B_HG, 0x220 );
    /* Assume NoError when Initialize ACL-8112 */
    .
    .
    .
    for( ch=0; ch<16; ch++ )
    {
        _8112_DI_Channel( ch , &data );
        printf( "The value of DI channel %d is
%d.\n" , ch , data );
    }
}
```

---

## 7.5 \_8112\_DO

### **Description**

This function is used to write data to digital output port. There are 16 digital outputs on the ACL-8112, they are divided by two ports, DO\_LO\_BYTE and DO\_HI\_BYTE. The channel 0 to channel 7 are defined in DO\_LO\_BYTE port and the channel 8 to channel 15 are defined as the DO\_HI\_BYTE port.

### **Syntax**

```
int _8112_DO(int port_number, unsigned char data )
int _8112pg_DO(int port_number, unsigned char data )
```

### **Argument:**

**port\_number:** DO\_LO\_BYTE or DO\_HI\_BYTE  
**data:** value will be written to digital output port

### **Return Code:**

```
ERR_NoError
ERR_BoardNoInit
ERR_PortError
```

### **Example:**

```
#include "8112.h"

main()
{
    _8112_Initial( CARD_1, A8112B_DG, 0x220 );
    /* Assume NoError when Initialize ACL-8112/DG ver.B
    card */
    .
    .
    _8112_DO( DO_LO_BYTE , 0x55 );
    printf( "The low byte is now 0x55.\n" );

    _8112_DO( DO_HI_BYTE , 0xAA );
    printf( "The high byte is now 0xAA.\n" );

}
```

A more detailed program is provided in this software  
'DO\_DEMO.C'

---

## 7.6 `_8112_DA`

### **Description**

This function is used to write data to D/A converter. There are two Digital-to-Analog conversion channels on the ACL-8112. The resolution of each channel is 12-bit, i.e. the digital data range is from 0 to 4095.

### **Syntax**

```
int _8112_DA(int da_ch_no, unsigned int data )
int _8112pg_DA(int da_ch_no, unsigned int data )
```

### **Argument:**

**da\_ch\_no:** D/A channel number, DA\_CH\_1 or DA\_CH\_2.  
**data:** D/A converted value, if the value is greater than 4095, the higher 4-bits are negligent.

### **Return Code:**

```
ERR_NoError
ERR_BoardNoInit
ERR_InvalidDAChannel
```

### **Example:**

```
#include "8112.h"

main()
{
    _8112_Initial( CARD_1, A8112B_DG, 0x220 );
    /* Assume NoError when Initialize ACL-8112 */
    /* if the hardware setting for DA output range
    is 0~5V */

    _8112_DA( DA_CH_1 , 0x800 );
    printf( "The output voltage of CH1 is 2.5V \n" );

    _8112_DA( DA_CH_2 , 0xFFFF );
    printf( "The output voltage of CH2 is 5V \n" );

}A more complete program is provided in this software.
'DA_DEMO.C'
```

---

## 7.7 `_8112_AD_Input_Mode`

### **Description**

This function is only useful for ACL-8112 ver B series.

The ACL-8112 can offer either 16 single-ended analog input channels or 8 differential analog input channels. If you use the ACL-8112 ver B card, you have to call this function before the A/D operation is processed.

### **Syntax**

```
int _8112_AD_Input_Mode( int ad_mode )
```

### **Argument:**

**ad\_ch\_mode:**

*SINGLE\_ENDED*: the analog inputs are single-ended mode.

*DIFFERENTIAL*: the analog inputs are differential.

### **Return Code:**

```
ERR_NoError  
ERR_BoardNoInit  
ERR_InvalidADChannel
```

### **Example:**

```
#include "8112.h"  
  
main()  
{  
  int j;  
  
  _8112_Initial( CARD_1, A8112B_DG, 0x210 );  
  /* Assume ERR_NoError when Initialize ACL-8112 */  
  
  _8112_Initial( CARD_2, A8112B_HG, 0x220 );  
  /* Assume ERR_NoError when Initialize ACL-8112 */  
  
  _8112_AD_Input_Mode( DIFFERENTIAL );  
  /* set analog input mode as "differential" mode */  
  /* if this function is not called, the default input  
  mode is single-ended mode */  
  
  for( j = 0; j < 7 ; j++)  
  {  
    _8112_AD_Set_Channel( j );  
  }  
}
```

```
    printf( "AD channel %d is now selected.\n", j );
}
_8112_Switch_Card_No(CARD_1);
_8112_AD_Input_Mode( SINGLE_ENDED) ;
for( j = 0; j < 7 ; j++)
{
    _8112_AD_Set_Channel( j );

    printf( "AD channel %d is now selected.\n", j );
}
/* the following A/D's operation is based on channel
3 */
}
```

---

## 7.8 `_8112_AD_Set_Channel`

### **Description**

This function is used to set AD channel by means of writing data to the multiplexer scan channel register. There are 16 single-ended A/D channels in ACL-8112, so the channel number should be set between 0 to 15 only. The initial state is channel 0 which is a default setting by the ACL-8112 hardware configuration.

### **Syntax**

```
int _8112_AD_Set_Channel( int ad_ch_no )
int _8112pg_AD_Set_Channel( int ad_ch_no )
```

### **Argument:**

**ad\_ch\_no:** channel number to perform AD onversion  
for single-ended mode: channel no. is from 0-15  
for differential mode: channel no. is from 0-7

### **Return Code:**

```
ERR_NoError
ERR_BoardNoInit
ERR_InvalidADChannel
```

### **Example:**

```
#include "8112.h"
main()
{
  _8112_Initial( CARD_1, A8112B_DG, 0x220 );
  /* Assume NoError when Initialize ACL-8112 */

  _8112_AD_Input_Mode( DIFFERENTIAL );
  /* set analog input mode as "differential" mode */

  _8112_AD_Set_Channel( 3 );
  printf( "AD channel 3 is now selected.\n" );

  ...

  /* the following A/D's operation is based on channel
  3 */
}
```

---

## 7.9 \_8112\_AD\_Set\_Range

### **Description**

This function is used to set the A/D analog input range by means of writing data to the A/D range control register. There are two factors will change the analog input range - *Gain* and *Input type*.

The Gain can be choice from 0.5, 1, 5, 10, 50, 100, 500, and 1,000 for ACL-8112HG card.

The input type is either *Bipolar* or *Unipolar*.

The initial value of gain is '1' and input type is bipolar, which are pre-set by the ACL-8112 hardware. The relationship between analog input voltage range, gain and input mode are specified by following tables:

\*\* this table is suitable for ACL-8112HG ( high gain) card.

AD_INPUT	GAIN	Input type (Bipolar or Unipolar)	Input Range
AD_B_5_V	1	Bipolar	±5V
AD_B_0_5_V	10	Bipolar	±0.5V
AD_B_0_05_V	100	Bipolar	±0.05V
AD_B_0_005_V	1,000	Bipolar	±0.005V
AD_U_10_V	1	Unipolar	0V ~ 10V
AD_U_1_V	10	Unipolar	0V ~ 1V
AD_U_0_1_V	100	Unipolar	0V ~ 0.1V
AD_U_0_01_V	1,000	Unipolar	0V ~ 0.01V
AD_B_10_V	0.5	Bipolar	±10V
AD_B_1_V	5	Bipolar	±1V
AD_B_0_1_V	50	Bipolar	±0.1V
AD_B_0_01_V	500	Bipolar	±0.01V



For ACL-8112DG card, the gain values only support 1, 2, 4, 8 four levels. The relationship between analog input voltage range, gain and input type are specified by table on next page.

\*\* this table is suitable for ACL-8112DG ( low gain) card.

AD_INPUT	GAIN	Input type Bipolar or Unipolar)	Input Range
AD_B_5_V	1	Bipolar	±5V
AD_B_2_5_V	2	Bipolar	±2.5V
AD_B_1_25_V	4	Bipolar	±1.25V
AD_B_0_625_V	8	Bipolar	±0.625V
AD_U_10_V	1	Unipolar	0V ~ 10V
AD_U_5_V	2	Unipolar	0V ~ 5V
AD_U_2_5_V	4	Unipolar	0V ~ 2.5V
AD_U_1_25_V	8	Unipolar	0V ~ 1.25V

For ACL-8112DG card, the gain values support 1,2,4,8,and 16.

The initial value of gain is '1'. The relationship between gain and input voltage ranges is specified by following tables:

when input voltage range is set to ±5 V (JP9),

Gain Code	Gain	Input Range (V)
AD_GAIN_1	X 1	±5 V
AD_GAIN_2	X 2	±2.5 V
AD_GAIN_4	X 4	±1.25 V
AD_GAIN_8	X 8	±0.625 V
AD_GAIN_16	X 16	±0.3125V

when input voltage range is set to ±10 V(JP9),

Gain Code	Gain	Input Range (V)
AD_GAIN_1	X 1	±10 V
AD_GAIN_2	X 2	±5 V
AD_GAIN_4	X 4	±2.5 V
AD_GAIN_8	X 8	±1.25 V
AD_GAIN_16	X 16	±0.625 V

### Syntax

```
int _8112_AD_Set_Range( int ad_range )
int _8112pg_AD_Set_Gain( int ad_range )
```

**Argument:**

int ad\_range: the programmable range of A/D conversion, please refer to above tables for the possible values .

**Return Code:**

ERR\_NoError  
ERR\_BoardNoInit  
ERR\_AD\_InvalidRange

**Example:**

```
#include "8112.h"
main()
{
    _8112_Initial( CARD_1, A8112B_HG, 0x220 );
    /* Assume NoError when Initialize ACL-8112 */

    _8112_AD_Input_Mode( DIFFERENTIAL) ;
    /* set analog input mode as "differential" mode */

    _8112_AD_Set_Range( AD_B_5_V );
    printf( "The A/D analog input range is +/- 5V \n" );

    for( j = 0; j < 7; j++)
    {
        _8112_AD_Set_Channel( j );
        printf( "AD channel j is now selected.\n" );

        /* all analog input operations are based on
           analog differential mode,
           input range is from -5V to +5V */
    }
    ...
}
```

---

## 7.10 \_8112\_AD\_Set\_Mode

### Description

This function is used to set the A/D trigger and data transfer mode by means of writing data to the mode control register. The hardware initial state of the ACL-8112 is set as AD\_MODE\_1 software( internal) trigger with program polling data.

A/D Mode	Description
AD_MODE_0	External Trigger, Software Polling
AD_MODE_1	Software Trigger, Software Polling
AD_MODE_2	Timer Trigger, DMA Transfer
AD_MODE_3	External Trigger, DMA Transfer
AD_MODE_4	External Trigger, Interrupt Transfer
AD_MODE_5	Software Trigger, Interrupt Transfer
AD_MODE_6	Timer Trigger, Interrupt Transfer
AD_MODE_7	Not Used

---

**Note:** All analog input modes selection should be go with hardware settings, which is described in the hardware users manual.

---

### Syntax

```
int _8112_AD_Set_Mode(int ad_mode )
int _8112pg_AD_Set_Mode(int ad_mode )
```

### Argument:

**ad\_mode:** AD trigger and data transfer mode  
( please refer above table.)

### Return Code:

```
ERR_NoError
ERR_BoardNoInit
ERR_InvalidMode
```

### Example:

```
#include "8112.h"

main()
{
  _8112_Initial( CARD_1, A8112B_HG, 0x220 );
  /* Assume NoError when Initialize ACL-8112 */

  _8112_AD_Input_Mode( DIFFERENTIAL) ;
```

```
/* set analog input mode as "differential" mode */

    _8112_AD_Set_Range( AD_B_5_V );
    printf( "The A/D analog input range is +/- 5V \n" );

    _8112_AD_Set_Mode( AD_MODE_6 );
    printf( "Now, disable internal trigger.\n" );

/* All A/D conversion will be trigger by internal
timer pacer, and the converted data should be
transferred in the interrupt service routine. ( ISR).
*/

}
```

---

## 7.11 `_8112_AD_Soft_Trig`

### **Description**

This function is used to trigger the A/D conversion by software. When the function is called, a trigger pulse will be generated and the converted data will be stored in the base address Base +4 and Base +5, and can be retrieved by function `_8112_AD_Acquire()`. Please refer to section 7.12.

### **Syntax**

```
int _8112_AD_Soft_Trig( void )
int _8112pg_AD_Soft_Trig( void )
```

### **Argument:**

None

### **Return Code:**

```
ERR_NoError
ERR_BoardNoInit
```

### **Example:**

```
#include "8112.h"

main()
{
    _8112_Initial( CARD_1, 8112DG, 0x220 );
    /* Assume NoError when Initialize ACL-8112 */

    _8112_AD_Soft_Trig();
    printf( "Now, AD is triggered.\n" );
    .
    .
    _8112_AD_Aquire( &data);
}
```

---

## 7.12 \_8112\_AD\_Aquire

### **Description**

This function is used to poll the AD conversion data. It will trigger the AD conversion, and read the 12-bit A/D data when the data is ready ('data ready' bit becomes low).

### **Syntax**

```
int _8112_AD_Aquire( int *ad_data )
int _8112pg_AD_Aquire( int *ad_data )
```

### **Argument:**

**ad\_data:** 12-bit A/D converted value, the value should  
within 0 to 4095.

### **Return Code:**

```
ERR_NoError
ERR_BoardNoInit
ERR_AD_AquireTimeOut
```

### **Example:**

```
#include "8112.h"
main()
{
    int ad_data;
    int ErrCode;

    _8112_Initial( CARD_1, 8112B-DG, 0x220 );
    /* Assume NoError when Initialize ACL-8112 */

    /* Set to software trigger at first*/
    _8112_AD_Set_Mode( AD_MODE_1 );
    /* then trigger the AD */
    _8112_AD_Soft_Trig();
    /* wait for AD data ready then read it */
    ErrCode = _8112_AD_Aquire( &ad_data );
    if( ErrCode == ERR_NoError )
        printf( "The AD value is %d.\n", ad_data );
    else
        printf( "AD conversion error happen\n" );
}
Also see deme program 'AD_DEMO.C'
```

---

## 7.13 `_8112_CLR_IRQ`

### **Description**

This function is used to clear interrupt request which requested by the ACL-8112. If you use interrupt to transfer A/D converted data, you should use this function to clear interrupt request status, otherwise no new coming interrupt will be generated.

### **Syntax**

```
int _8112_CLR_IRQ( void )  
int _8112pg_CLR_IRQ( void )
```

### **Argument:**

None

### **Return Code:**

```
ERR_NoError  
ERR_BoardNoInit
```

---

## 7.14 \_8112\_AD\_DMA\_Start

### Description

The function will perform A/D conversion N times with DMA data transfer by using the pacer trigger ( internal timer trigger). It takes place in the background which will not be stop until the Nth conversion has been completed or your program execute `_8112_AD_DMA_Stop()` function to stop the process. After executing this function, it is necessary to check the status of the operation by using the function `_8112_AD_DMA_Status()`. The function is performed on single A/D channel with fixed analog input range.

### Syntax

```
int _8112_DMA_Start( int ad_ch_no, int ad_range,
                    intdma_ch_no, int irq_ch_no
                    int count , int *ad_buffer
                    unsigned int c1, unsigned int c2)
int _8112pg_DMA_Start( int ad_ch_no, int ad_gain,
                      intdma_ch_no, int irq_ch_no
                      int count , int *ad_buffer
                      unsigned int c1, unsigned int c2)
```

### Argument:

**ad\_ch\_no:** A/D channel number

**ad\_range:** A/D analog input range, please refer to the Section 7.9 to find the possible values.

**dma\_ch\_no:** DMA channel number, DMA\_CH\_1 or DMA\_CH\_3

---

**Note:** Make sure your hardware configuration is set to right DMA channel.

---

**irq\_ch\_no:** IRQ channel number, used to stop DMA

---

**Note:** Make sure your hardware configuration is set to right IRQ interrupt level.

---

**count:** the number of A/D conversion

**ad\_buffer:** the start address of the memory buffer to

store the AD data, the buffer size must large than the numbers of AD conversion.

**c1:** the 16-bit timer frequency divider of channel #1

**c2:** the 16-bit timer frequency divider of channel #2



**Return Code:**

```
ERR_NoError  
ERR_BoardNoInit,  
ERR_InvalidADChannel,  
ERR_AD_InvalidRange,  
ERR_InvalidDMAChannel,  
ERR_InvalidIRQChannel,  
ERR_InvalidTimerValue
```

**Example:**

See Demo Program 'AD\_Demo4.C

---

## 7.15 `_8112_AD_DMA_Status`

### **Description**

Since the `_8112_AD_DMA_Start` function is executed in background, you can issue the function `_8112_AD_DMA_Status` to check its operation status.

### **Syntax**

```
int _8112_AD_DMA_Status( int *status , int *count )  
int _8112pg_AD_DMA_Status( int *status , int *count )
```

### **Argument:**

<b>status:</b>	status of the DMA data transfer 0: AD DMA is not completed 1: AD DMA is completed
<b>count:</b>	the number of A/D data which has been transferred.

### **Return Code:**

```
ERR_NoError  
ERR_BoardNoInit  
ERR_AD_DMANotSet
```

### **Example:**

See demo program 'AD\_Demo4.C'

---

## 7.16 `_8112_AD_DMA_Stop`

### **Description**

This function is used to stop the DMA data transfer. After executing this function, the internal A/D trigger is disabled and the A/D timer ( timer #1 and #2) is stopped. The function returns the number of the data which has been transferred, no matter if the A/D DMA data transfer is stopped by this function or by the DMA terminal count ISR.

### **Syntax**

```
int _8112_AD_DMA_Stop( int *count )  
int _8112pg_AD_DMA_Stop( int *count )
```

### **Argument:**

**count:**                    the number of A/D converted data which  
has                            been transferred.

### **Return Code:**

```
ERR_NoError  
ERR_BoardNoInit  
ERR_AD_DMANotSet
```

### **Example:**

See demo program '`AD_Demo4.C`'

---

## 7.17 \_8112\_AD\_INT\_Start

### Description

The function will perform A/D conversion N times with interrupt data transfer by using pacer trigger. It takes place in the background which will not be stopped until the Nth conversion has been completed or your program execute `_8112_AD_INT_Stop()` function to stop the process. After executing this function, it is necessary to check the status of the operation by using the function `8112_AD_INT_Status()`. The function is performed on single A/D channel with fixed analog input range.

### Syntax

```
int _8112_INT_Start( int ad_ch_no, int ad_range,
                    int irq_ch_no, int count, int
                    *ad_buffer,   unsigned int c1, unsigned int c2)
int _8112pg_INT_Start( int ad_ch_no, int ad_gain,
                       int irq_ch_no, int count, int
                       *ad_buffer,   unsigned int c1, unsigned int c2)
```

### Argument:

**ad\_ch\_no:** A/D channel number  
**ad\_range:** A/D analog input range, please refer to section 7.9 for the possible values.

**irq\_ch\_no:** IRQ channel number used to transfer AD data, the possible value is defined in '8112.h'

**count:** the number of A/D conversion  
**ad\_buffer:** the start address of the memory buffer to store the A/D data, the buffer size must be large than the number of A/D conversion.

**c1:** the 16-bit timer frequency divider of channel #1  
**c2:** the 16-bit timer frequency divider of channel #2

### Return Code:

```
ERR_NoError
ERR_BoardNoInit
ERR_InvalidADChannel
ERR_AD_InvalidRange
ERR_InvalidIRQChannel
ERR_InvalidTimerValue
```

**Example:**

See demo Program 'AD\_Demo2.C'

---

**7.18 \_8112\_AD\_INT\_Status****Description**

Since the `_8112_AD_INT_Start()` function is executed in background, you can issue the function `_8112_AD_INT_Status` to check the status of interrupt operation.

**Syntax**

```
int _8112_AD_INT_Status( int *status , int *count )  
int _8112pg_AD_INT_Status( int *status , int *count )
```

**Argument:**

<b>status:</b>	status of the INT data transfer 0: A/D INT is completed 1: A/D INT is not completed
<b>count:</b>	current conversion count number.

**Return Code:**

```
ERR_NoError  
ERR_BoardNoInit  
ERR_AD_INTNotSet
```

**Example:**

See demo program 'AD\_Demo2.C'

---

## 7.19 \_8112\_AD\_INT\_Stop

### **Description**

This function is used to stop the interrupt data transfer function. After executing this function, the internal AD trigger is disabled and the AD timer is stopped. The function returns the number of the data which has been transferred, no matter whether the AD interrupt data transfer is stopped by this function or by the \_8112\_AD\_INT\_Start() itself.

### **Syntax**

```
int _8112_AD_INT_Stop( int *count )  
int _8112pg_AD_INT_Stop( int *count )
```

### **Argument:**

count:                   the number of A/D data which has been transferred.

### **Return Code:**

```
ERR_NoError  
ERR_BoardNoInit  
ERR_AD_INTNotSet
```

### **Example:**

See Demo Program 'AD\_Demo2.C'

---

## 7.20 `_8112_AD_Timer`

### **Description**

This function is used to setup the Timer #1 and Timer #2.

Timer #1 & #2 are used as frequency divider for generating constant A/D sampling rate dedicatedly. It is possible to stop the pacer trigger by setting any one of the dividers as 0. The AD conversion rate is limited due to the conversion time of the AD converter, the highest sampling rate of the ACL-8112 can not be exceeded 100 KHz, Thus the multiplication of the dividers must be larger than 20.

### **Syntax**

```
int _8112_AD_Timer( unsigned int c1 , unsigned int
c2 )
int _8112pg_AD_Timer( unsigned int c1 , unsigned int
c2 )
```

### **Argument:**

```
c1:          frequency divider of timer #1
c2:          frequency divider of timer #2,
```

---

**Note:** the A/D sampling rate is equal to:

$$2\text{MHz} / (c1 * c2),$$

when  $c1 = 0$  or  $c2 = 0$ , the pacer trigger will be stopped.

---

### **Return Code:**

```
ERR_NoError
ERR_BoardNoInit
ERR_InvalidTimerValue
```

### **Example:**

```
main()
{
    int ErrCode;

    _8112_Initial( CARD_1, A8112B_HG, 0x220 );
    /* Assume ERR_NoError when Initialize ACL-8112 */

    _8112_AD_Timer( 10 , 10 );
    /* set AD sampling rate to 2MHz/(10*10) */

    ..
}
```

```
_8112_AD_Timer( 0 , 0 );  
/* stop the pacer trigger */  
}
```

---

## 7.21 \_8112\_TIMER\_Start

### **Description**

The Timer #0 on the ACL-8112 can be freely programmed by the users. This function is used to program the Timer #0. This timer can be used as frequency generator if internal clock is used. It also can be used as event counter if external clock is used. All the 8253 mode is available.

### **Syntax**

```
int _8112_TIMER_Start( int timer_mode, unsigned int  
c0 )  
int _8112pg_TIMER_Start( int timer_mode, unsigned int  
c0 )
```

### **Argument:**

**timer\_mode:** the 8253 timer mode, the possible values are:  
TIMER\_MODE0, TIMER\_MODE1,  
TIMER\_MODE2, TIMER\_MODE3,  
TIMER\_MODE4, TIMER\_MODE5.  
**c0:** the counter value of timer

### **Return Code:**

```
ERR_NoError  
ERR_BoardNoInit  
ERR_InvalidTimerMode  
ERR_InvalidTimerValue
```

### **Example:**

See demo program 'TMR\_DEMO.C'



---

## 7.22 `_8112_TIMER_Read`

### **Description**

This function is used to read the counter value of the Timer #0.

### **Syntax**

```
int _8112_TIMER_Read( unsigned int *counter_value )
int _8112pg_TIMER_Read( unsigned int *counter_value )
```

### **Argument:**

**counter\_value:** the counter value of the Timer #0

### **Return Code:**

```
ERR_NoError
ERR_BoardNoInit
```

### **Example:**

See demo program 'TMR\_DEMO.C'

---

## 7.23 `_8112_TIMER_Stop`

### **Description**

This function is used to stop the timer operation. The timer is set to the 'One-shot' mode with counter value '0'. That is, the clock output signal will be set to high after executing this function.

### **Syntax**

```
int _8112_TIMER_Stop( unsigned int *counter_value )
int _8112pg_TIMER_Stop( unsigned int *counter_value )
```

### **Argument:**

**\*counter\_value:** the current counter value of the Timer #0

### **Return Code:**

```
ERR_NoError
ERR_BoardNoInit
```

### **Example:**

See demo program 'TMR\_DEMO.C'

# Appendix A. Demo Programs

In this software diskette, there are 8 demonstration programs are provided. They could help you to program your application by using C Language Library easily. The description of these programs are specified as follows:

AD_DEMO1.C:	A/D conversion uses software trigger and program data transfer.
AD_DEMO2.C	A/D conversion uses interrupt and program data transfer.
AD_DEMO3.C:	A/D conversion uses DMA data transfer.
AD_DEMO4.C:	A/D conversion uses multiple channels and input range programmable, triggers by PC timer, and saves all data to a file 'ad_demo4.data'.
DA_DEMO.C:	D/A conversion
DI_DEMO.C:	Read data from digital input channels
DO_DEMO.C:	Write data to digital output channels
TMR_DEMO.C:	Handle 8253 Timer/Counter

## Product Warranty/Service

ADLINK warrants that equipment furnished will be free from defects in material and workmanship for a period of one year from the date of shipment. During the warranty period, we shall, at our option, either repair or replace any product that proves to be defective under normal operation.

This warranty shall not apply to equipment that has been previously repaired or altered outside our plant in any way as to, in the judgment of the manufacturer, affect its reliability. Nor will it apply if the equipment has been used in a manner exceeding its specifications or if the serial number has been removed.

ADLINK does not assume any liability for consequential damages as a result from our product uses, and in any event our liability shall not exceed the original selling price of the equipment. The remedies provided herein are the customer's sole and exclusive remedies. In no event shall ADLINK be liable for direct, indirect, special or consequential damages whether based on contract of any other legal theory.

The equipment must be returned postage-prepaid. Package it securely and insure it. You will be charged for parts and labor if the warranty period is expired or the product is proves to be misuse, abuse or unauthorized repair or modification.