NuDAQ®

ACL-8111

16-bit High Resolution
Data Acquisition Card

User's Guide

Recycled Paper

Manual Rev. 2.60: April 28, 2003

Part No: 50-11010-102

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.

Trademarks

NuDAQ is a 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 the most important priority for ADLINK Tech Inc. If you need any help or service, please contact us.

ADLINK Technology Inc.			
Web Site	http://www.adlinktech.com		
Sales & Service	Service@adlinktech.com		
Technical Support	NuDAQ + USBDAQ + PXI	nudaq@adlinktech.com	
	Automation	automation@adlinktech.com	
	NuIPC	nuipc@adlinktech.com	
	NuPRO / EBC	nupro@adlinktech.com	
TEL	+886-2-82265877	FAX	+886-2-82265717
Address	9F, No. 166, Jian Yi Road, Chungho City, Taipei, 235 Taiwan.		

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

Detailed Company Information			
Company/Organization			
Contact Person			
E-mail Address			
Address			
Country			
TEL		FAX	
Web Site			
Questions			
Product Model			
Environment to Use	OS: Computer Brand: M/B: Chipset: Video Card: Network Interface Card: Other:	CPU: BIOS:	
Detail Description			
Suggestions to ADLINK			

Table of Contents

1

Introduction

The ACL-8111 is a multi-function, high performance, and general-purpose data acquisition card for the IBM Personal Computer and compatible computers. The registers map is fully compatible with PCL-711B.

The ACL-8111 is designed to combine all the data acquisition functions, such A/D, D/A, D/O, and D/I in a single board, The high-end specifications of the card makes it ideal for wide range of applications in industrial and laboratory automation.

1.1 Features

The ACL-8111 Enhanced Multi-function Data Acquisition Card provides the following advanced features:

- 8 single-ended 12-bit analog input channels
- On-chip sample & hold
- Variable input range: ± 5V, ±2.5V, ±1.25V, ±0.625V, and ±0.3125V
- One 12-bit monolithic multiplying analog output channel
- 16 digital output channels
- 16 digital input channels
- Three A/D trigger modes: Software trigger, Programmable pacer and external trigger
- Programmable IRQ level for A/D data transfer
- Rugged 37-pin D-sub connector
- Compact 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

1.3 Specifications

♦ **Analog Input (A/D)**

- **Converter:** ADS-774 or equivalent one, successive approximation type
- **Resolution:** 12-bit
- **Number of channels:** 8 single-ended
- **Programmable Gain:** 1, 2, 4, 8, 16 (programmable)
- **Input Range:** (programmable)

  Bipolar: ±5V, ± 2.5V, ±1.25V, ±0.625V, and ±0.3125V
- **Conversion Time:** 8 μ sec
- **Maximum Sampling Rate: ≤ 50**KHz
- **Over-voltage protection:** Continuous ± 35V maximum
- **Accuracy:** 0.015% of reading ± 1 Bit LSB
- **Input Impedance:** 10 MΩ
- **AD Clock Sources (trigger modes):** Software trigger, internal timer pacer, and external pulse trigger
- **Data Transfer Modes:** Polling and Interrupt

♦ **Analog Output (D/A)**

- **Converter:** DAC7548 or equivalent, monolithic multiplying
- **Number of channel:** 1 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 μ sec
- **Linearity:** ±1/2 bit LSB
- **Output driving capability:** ±5mA max.

- ♦ **Digital I/O (DIO)**
  - **Number of channel:** 16 DI & 16 DO (TTL compatible)
  - **Input Voltage:**
  - Low: Min. 0V; Max. 0.8V
  - High: Min. +2.0V
  - **Input Load:**
  - Low: +0.5V @ -0.2mA max.
  - High: +2.7V @+20 uA max.
  - **Output Voltage:**
  - Low: Min. 0V; Max. 0.4V
  - High: Min. +2.4V
  - **Output Driving Capacity:**
  - Low: Max. +0.5V at 8.0mA (Sink)
  - High: Min. 2.7V at 0.4mA(Source)
- ♦ **Programmable Counter**
  - **Device:** 82C54
  - **A/D pacer Source:** 32-bit timer (two 16-bit counter cascaded together) with a 2MHz time base
  - **Pacer Output:** 0.00046 Hz ~ 100K Hz
- ♦ **General Specifications**
  - **I/O Base Address**: 16 consecutive address location
  - **Interrupt IRQ**: IRQ 2,3,4,5,6,7 (programmable)
  - **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**:

- +5 V @ 440 mA typical
- +12V @ 60 mA typical
- **Dimension:** 158mm X 108mm

1.4 Software Supporting

The ACL-8111 is programmed using simple 8-bit I/O port commands. Users can use high level language, such as BASIC, C, or PASCAL, or low- level language, such as assembly to program the board. For the programming under Windows or LabView, please contact your dealer to purchase ACLS-DLL2 or ACLD-LVIEW.

1.4.1 ACLS-DLL2

For easily program the board under Windows environment, we also provide **ACLS-DLL2**, which include the DLL for Windows 95/98/NT. With ACLS-DLL2, you can use compilers such as VB, VC/C++, or Delphi. ACLS-DLL2 is included in ADLINK CD. It need license.

1.4.2 ACLS-LVIEW

For easily link the ACL-8111 with LabView of National Instrument, the ACLS-LVIEW includes the Vis of the ACL-8111 under for Windows 3.1/95/98 and NT. ACLS-LVIEW is included in ADLINK CD. It need license.

2

Installation

This chapter describes how to install the ACL-8111. The following list shows the installation procedures:

1. Unpacking and check what you have.
2. Check the PCB and the location of jumper and switch.
3. Setup the jumpers according to the system and applications setting.
4. Installation the external termination boards.
5. Wiring with signals according to the connector pin assignments.

2.1 Unpacking

The ACL-8111 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 handing 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-8111.

2.2 What You Have

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

- ACL-8111 multi-function data acquisition card
- ADLINK Software 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.

2.3 PCB Layout

In ACL-8111, there are three connectors (CN1~CN3), one switch (SW1), one jumper (JP1) and 5 variable resistors (VR1~VR5). Please refer Figure 2.3 to identify the location these components.

The card's jumpers and switches are preset at the factory. Please setup the switch and jumper according to the system setting and the application requirements.

A jumper 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.

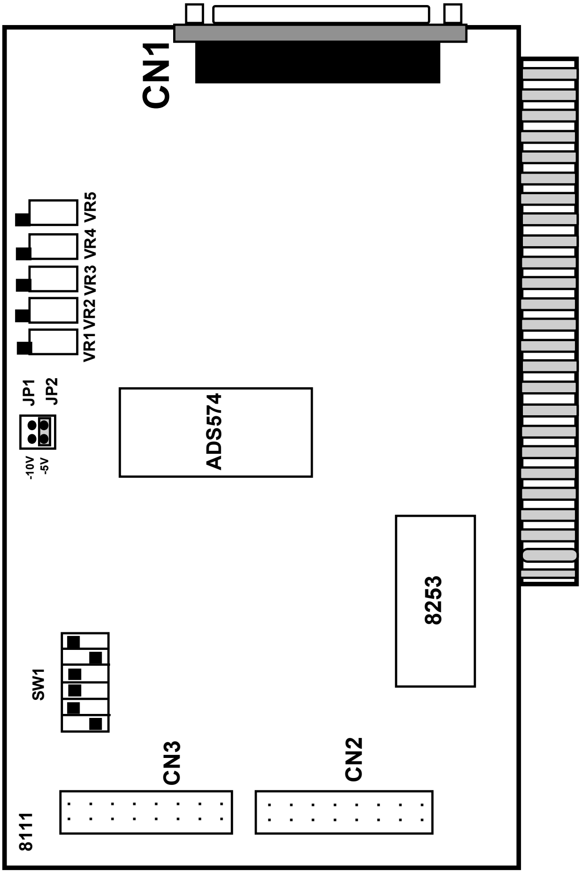


Figure 2.3 ACL-8111's PCB layout

2.4 Base Address Setting

The ACL-8111 requires 16 consecutive address locations in I/O address space. The base address of the ACL-8111 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-8111.

The ACL-8111's base address is selected by a 6 position 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.

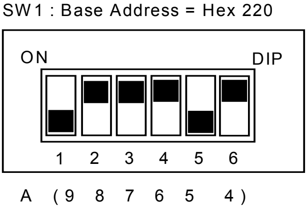


Figure 2.4 Default Base Address Setting

I/O port address(Hex)	**1 A9**	**2 A8**	**3 A7**	**4 A6**	**5 A5**	**6 A4**
000-00F	ON (0)	ON (0)	ON (0)	ON (0)	ON (0)	ON (0)
100-10F	ON (0)	OFF (1)	ON (0)	ON (0)	ON (0)	ON (0)
:						
200-20F	OFF (1)	ON (0)	ON (0)	ON (0)	ON (0)	ON (0)
210-21F	OFF (1)	ON (0)	ON (0)	ON (0)	ON (0)	OFF (0)
220-22F (default)	OFF (1)	ON (0)	ON (0)	ON (0)	OFF (1)	ON (0)
:						
300-30F	OFF (1)	OFF (1)	ON (0)	ON (0)	ON (0)	ON (0)
3F0-3FF	OFF (1)	OFF (1)	OFF (1)	OFF (1)	OFF (01	OFF (1)

ON: 0 OFF: 1.

A4 ... A9 are correspond to PC address lines.

How to Define a Base Address for the ACL-8111?

The DIP1 to DIP6 in the switch SW1are one to one corresponding to the PC bus address line A9 to A4. A0, A1, A2, and A3 are always 0. If you want to change the base address, you can only change the values of A9 to A4 (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**

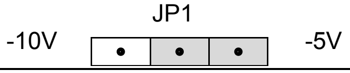
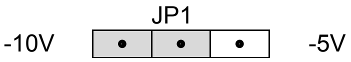
2			**2**					**0**			
1	**0**		**0**	**0**	**1**	**0**		**0**	**0**	**0**	**0**
A9	A8		A7	A6	A5	A4		A3	A2	A1	A0

2.5 D/A Reference Voltage Setting

The analog output range of ACL-8111 depends on the reference voltage, which is selected by JP1. There are two range of reference voltages can be assigned as either -5V or -10V, that will generate output range 0V ~ +5V or 0V ~ +10V, respectively.

The setting of D/A reference voltage is specified as following:

When you select -10V for the D/A reference voltage, you should be very careful the analog output. Because the -10 V voltage source is supplied from your PC system, and it may be up to 11.5V only. Therefore, it will cause you analog output is less than +10V.

Reference Voltage is -5V **(default)**	
Reference Voltage is -10V	

2.6 Connectors Pin Assignments

The ACL-8111 has three connectors: one 37-pin D-type connector (CN1) and two 20-pin insulation displacement connectors (CN2 and CN3).

CN1 is for analog input and analog output signals. CN2 is for digital input signal. CN3 is for digital output signal. The pin assignments for each connector are illustrated in the Figure 3.1 ~ Figure 3.3.

CN1: Analog input/output

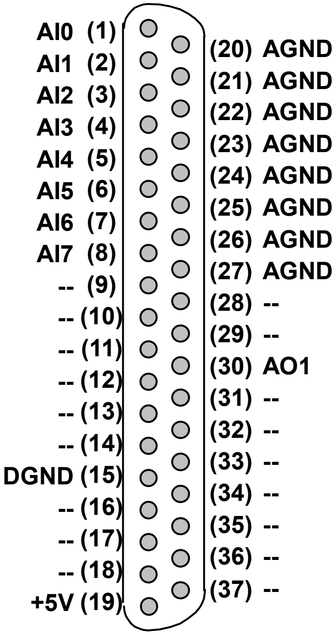
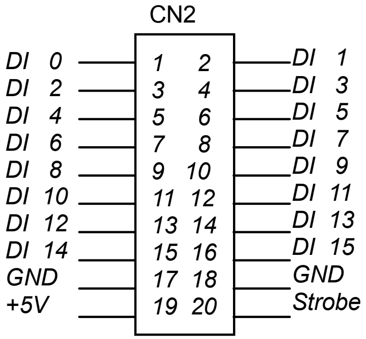


Figure 3.1. Pin Assignment of CN1

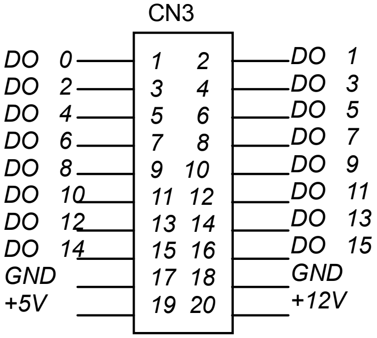
Legend:

AIn	: Analog Input Channel *n* (single-ended)
AO	: Analog Output Channel
A.GND	: Analog Ground
D.GND	: Ground
+5V	: +5V from system power supply

*CN 2: Digital Input Signals (*DI 0 - 15)



*CN 3: Digital Output Signals (*DO 0 - 15)



Legend:

DO n	: Digital output signal channel *n*
DI n	: Digital input signal channel *n*
GND	: Digital ground
+12V	: +12V power from system
Strobe	: Digital input strobe signal

2.7 Daughter Board Connection

The ACL-8111 can be connected with five different daughter boards ACLD-9137, 9182, 9185, and 9188. The functionality and connections are specified as following:

2.7.1 Connect with ACLD-9137

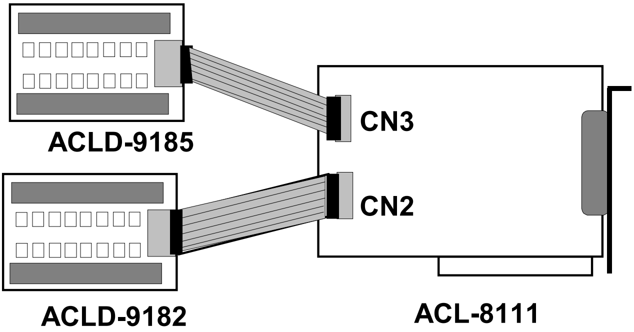
The ACLD-9137 is a direct connector for the card, which is equipped with 37-pin D-sub connector. This board provides a simple way for connection. It is very suitable for the simple applications that do not need complex signal condition before the A/D conversion is performed.

2.7.2 Connect with ACLD-9182

The ACLD-9182 is a 16 channel isolated digital input board. This board is connected with CN2 of ACL-8111 via 20-pin flat cable. The advantage of this board is a 500Vdc isolation voltage is provided, and it can protect your PC system from damage when an abnormal input signal is occurred.

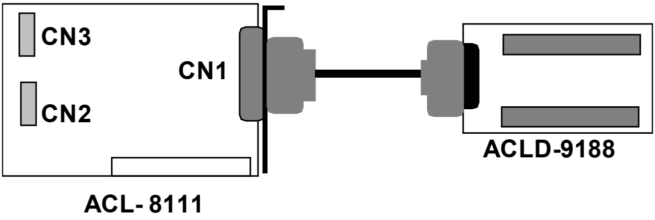
2.7.3 Connect with ACLD-9185

The ACLD-9185 is a 16 channel SPDT relay output board. This board is connected with CN3 of ACL-8111 via 20-pin flat cable. By using this board, you can control outside device through the digital output signals.



2.7.4 Connect with ACLD-9188

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



3

Registers Format

The chapter specifies the detailed descriptions of the register format. This information is useful for the programmers who wish to handle the card by low-level program.

3.1 Registers Address Map

The ACL-8111 requires 16 consecutive addresses in the I/O address space. The following table (Table 3.1) shows the location of each register relative to the base address and register descriptions.

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	**Counter Control**
Base + 4	**A/D low byte**	**D/A low byte**
Base + 5	**A/D high byte**	**D/A high byte**
Base + 6	**DI low byte**	Not Used
Base + 7	**DI high byte**	Not Used
Base + 8	Not Used	**Clear Interrupt Request**
Base + 9	Not Used	**Gain Control**
Base + 10	Not Used	**Multiplexer Scan Control**
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 3.1 I/O Port Address Map

3.2 A/D Data Registers

ACL-8111 provides 8 single-end A/D input channels. The 12 bit A/D data is put in two 8-bit registers. The low byte date (8 LSBs) is at address BASE+4 and the high byte data (4 MSBs) is at address BASE+5. The 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 after reading out the low byte data

3.3 A/D Channel Multiplexer Register

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

Address: BASE + 10

Attribute: write only

Data Format:

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

CLn: multiplexer channel number.

CL2 is MSB, and CL0 is LSB.

Where CL2, CL1, and CL0 are multiplexer channel selection:

Channel NO.	**CL2**	**CL1**	**CL0**	
0	0	0	0	Power Up Setting
1	0	0	1	
2	0	1	0	
3	0	1	1	
4	1	0	0	
5	1	0	1	
6	1	1	0	
7	1	1	1	

3.4 A/D Input Range Control Register

The gain control register is a write-only register using address Base + 9. It is used to adjust the gain of the analog input programmable amplifier. The data format is specified as follows:

Address: BASE + 9

Attribute: write only

Data Format:

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

Where G2, G1, and G0 are Gain selection:

G2	G1	G0	GAIN	Input Range (V)	
0	0	0	x1	± 5V	Power Up Setting
0	0	1	x2	± 2.5V	
0	1	0	x4	± 1.25V	
0	1	1	x8	± 0.625V	
1	0	0	x16	± 0.3125V	
1	0	1	Invalid	X	
1	1	0	Invalid	X	
1	1	1	Invalid	X	

The relationship between input voltage range, maximum input voltage and gain is specified as the equation:

Analog Input Range = Maximum Input Voltage / Gain

3.5 Mode and Interrupt Control Register

The Mode and Interrupt control register is a write-only register. This register control the AD operating modes and the IRQ level of the ACL-8111. The AD operation modes include the AD clock source and the AD data transfer method.

Address: BASE + 11

Attribute: write only

Data Format:

Bit	7	6	5	4	3	2	1	0
BASE+11	X	IR2	IR1	IR0	X	M2	M1	M0

Where M0, M1, and M2 are mode selection:

M2	M1	M0	Description
0	0	0	Software trigger with program polling
0	0	1	(No interrupt generated)
0	1	0	External trigger with program polling
0	1	1	External trigger with interrupt polling (With End-of-conversion “EOC” interrupt)
1	0	0	Internal time pacer trigger and polling (No interrupt generating)
1	0	1	Not Used
1	1	0	Internal timer pacer trigger with interrupt data transfer(With End-of-conversion “EOC” interrupt)
1	1	1	Not Used

Note: The external trigger signal comes from the DI 0 (pin-1 of CN2).

Where IR0, IR1, and IR2 are interrupt IRQ level settings:

IR2	IR1	IR0	IRQ Level
0	0	0	IRQ2
0	0	1	Not Used
0	1	0	IRQ2
0	1	1	IRQ3
1	0	0	IRQ4
1	0	1	IRQ5
1	1	0	IRQ6
1	1	1	IRQ7

3.6 Interrupt Status Register

The Interrupt Status Register is used to clear the interrupt status for next new interrupt can be generated. If the ACL-8111 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-8111 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

3.7 Software Trigger Register

If you want to generate a trigger pulse to the ACL-8111 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

3.8 Digital I/O register

There are 16 digital input channels and 16 digital output channels are provided by the ACL-8111. 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

3.9 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 conversion.

Address: BASE + 4 & BASE + 5

Attribute: write only

Data Format:

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

DA0 is the LSB and DA11 is the MSB of the 12 bit 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

Operation Theorem

The chapter describe the operation theorem of ACL-8111. The functions include the A/D conversion, D/A conversion, and digital I/O. The operation theorem can help you to understand how to manipulate and to program the ACL-8111.

4.1 A/D Conversion

Before programming the ACL-8111 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

4.1.1 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-8111 provides three trigger modes, see section 4.1.2.

The DRDY bit in the A/D data register (Base+5) will become to 0 once the A/D conversion is completed and data on the Base+4 and Base+5 are ready. A read data command on BASE+4 will automatically reset the DRDY bit to be 1 which means data is not ready on the Base+4 and Base+5. The A/D data should be transferred into PC's memory for further using. The ACL-8111 provides three data transfer modes that allow users to optimize the DAS system. Refer to section 4.1.3 for data transfer modes.

4.1.2 A/D Clock Sources (Trigger Modes)

In the ACL-8111, two Internal or one external clock sources can trigger A/D conversion. 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). The A/D operation modes combine the AD clock sources and the data transfer mode together. Please also refer to the next section for detail data transfer modes. The available operation modes include:

- Software trigger and software polling transfer
- Internal timer pacer and interrupt transfer
- Internal timer pacer and software polling transfer
- External trigger and interrupt transfer
- External trigger and software polling transfer

Three AD clock sources (or trigger sources) are available in the ACL-8111.

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). Under this mode, the timing of the A/D conversion is fully controlled by software, it is suitable for low speed A/D conversion. However, it is difficult to control the fixed A/D conversion rate except a timer interrupt service routine is used and the software trigger is programmed inside the interrupt service routines.

Timer Pacer Trigger

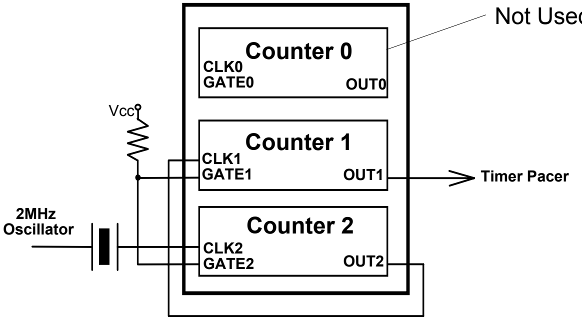
An on-board timer / counter chip 8253 provide a pacer trigger source at a fixed rate. Two counters of the 8253 chip are cascaded together to generate trigger pulse with precise frequency. It's recommend to use this mode if your applications need a fixed and precise A/D sampling rate. It can be combined with the EOC (end-of-conversion) interrupt data transfer.

On the ACL-8111, the 8253 chip for timer pacer trigger source is configured as below:

The pacer rate of above configuration is determined by the formula:

$$\boxed{\text{Pacer rate} = 2\text{ MHz} / (C1 * C2)}$$

8253 Timer/Counter



The maximum pacer signal rate is 2MHz/4=500K, which excess the maximum A/D conversion rate of the ACL-8111. The minimum signal rate is 2MHz/65535/65535, which is a very slow frequency that user may never use it.

For example, if you wish to get a pacer rate 2.5 kHz, you can set C1 = 40 and C2 = 10. That is

2.5KHz = 2Mhz / (40 x 20)

External Trigger

Through the pin-1 of CN2 (*DI0*), the A/D conversion also can be performed when 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 interrupt data transfer, or even program polling data transfer. Generally, the interrupt data transfer is often used when external trigger mode is used.

4.1.3 A/D Data Transfer Modes

On the ACL-8111, 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 excess 8μs on ACL-8111 card. Hence, after software trigger, the software can wait for at least 25μs then read the A/D register without polling.

EOC Interrupt Transfer

The ACL-8111 provides hardware interrupt capability. Under this mode, an interrupt signal is generated at end of A/D conversion (EOC) then 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.

4.2 D/A Conversion

The operation of D/A conversion is simpler 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 AO. The ACL-8111 has one uni-polar analog output channels. To make the D/A output connections from the appropriate D/A output, please refer Figure 4.2.

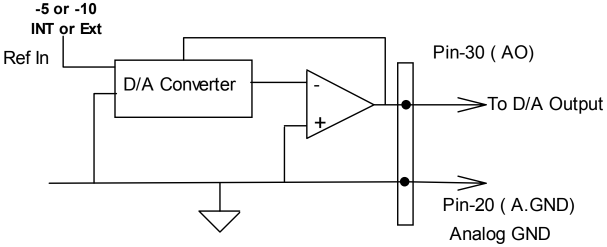


Figure 4.2 Connection of Analog Output Connection

The mathematical relationship between the digital number DAn and the output voltage is formulated as following:

$$Vout = -Vref \times \frac{DAn}{4096}$$

Where the *Vref* is the reference voltage, the *Vout* 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. Please refer section 2.6 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. The relationship of reference voltage and output range is summary in the following table.

Reference Voltage	DA Output Voltage Range
-5 V	0V ~ +5V
-10 V	0V ~ +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-8111, the low byte must be written before the high byte. This procedure can insure a single step transition when the D/A conversion.

Note: when writing digital data to D/A register, **the low byte data should be written first**, because the register will temporarily hold and not released to output. After the high byte is written, both high and low will add and passed to A/D converter.

4.3 Digital Input and Output

The ACL-8111 provides 16 digital input and 16 digital output channels through the connector CN2 and CN3 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.

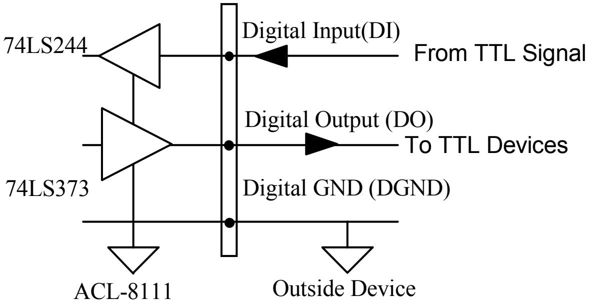


Figure 4.3 Digital I/O Connection

To program digital I/O operation is fairly straightforward. 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. 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

C/C++ Library for DOS

This chapter describes the DOS software library, which is free supplied. The DOS library software includes a utility program, C language library, and some demonstration programs, which can help you reduce the programming work.

To program in Windows environment, please use ACLS-DLL2. The function reference manual of ACLS-DLL2 is included in the ADLINK CD. It needs license.

5.1 DOS Software Library Installation

5.1.1 Installation

To install the DOS library software and utilities, please follow the following installation procedures:

1. Put *ADLINK CD* into the appropriate CD-ROM drive.

2. Type the following commands to change to the card's directory (*X* indicates the CD-ROM drive):

   X:\>**CD \NuDAQISA\8111**

3. Execute the setup batch program to install the software:

   X:\NuDAQISA\8111>**SETUP**

5.1.2 Running Utility

After finishing the installation, you can execute the utility by typing as follows:

```
C>cd ADLINK\8111\DOS\UTIL

C>8111UTIL
```

5.1.3 Example Programs

Six example programs in C language are supported in the directory <SAMPLES>, you can refer these samples and modify them for your own application. They can help you understand the library more quickly.

5.2 _8111_Initial

@ Description

An ACL-8111 card is initialized according to the card number and the corresponding base address. Every ACL-8111 Multi-Function Data Acquisition Card have to be initialized by this function before calling other functions.

@ Syntax

```
int _8111_Initial(int card_number, int base_addresss )
```

@ 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.
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  "8111.h"

main()
{
            int ErrCode;

            Errcode = _8111_Initial( CARD_1, 0x210 );
            if ( ErrCode != ERR_NoError )
        exit(0);

           ErrCode = _8111_Initial( CARD_2,  0x220 );
            if ( ErrCode != ERR_NoError )
        exit(0);
         ...
}
```

5.3 _8111_Switch_Card_No

@ Description

This function is used on dual-cards system. After two ACL-8111 cards are initialized by the initial function, you can select one card which you want to operate.

@ Syntax

```
int _8111_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  "aclerr.h"
#include  "8111.h"

main()
{
         _8111_Initial( CARD_1,  0x210 );
         _8111_Initial( CARD_2,  0x220 );
         /* Assume NoError when Initialize ACL-8111*/

         _8111_Switch_Card_No( CARD_1 );
         /*..... You can perform certain functions
         to Card_1 here*/

         _8111_Switch_Card_No( CARD_2 );
         /*..... You can perform certain functions
         to Card_2 here*/

}
```

5.4 _8111_DI

@ Description

This function is used to read data from digital input port. There are 16-bit digital inputs on the ACL-8111. 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 _8111_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'
```

5.5 _8111_DI _Channel

@ *Description*

This function is used to read data from digital input channels (bit). There are 16 digital input channels on the ACL-8111. When performs this function, the digital input port is read and the value of the corresponding channel is returned.

Note: channel means each bit of digital input ports.

@ *Syntax*

```
int _8111_DI_Channel(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  "8111.h"

main()
{
             unsigned int data;
             int  ch;

             _8111_Initial( CARD_1, 0x220 );
         /* Assume NoError when Initialize ACL-8111
         */
    .
    .
    .

             for( ch=0; ch<16; ch++ )
             {
             _8111_DI_Channel( ch , &data );
                 printf( "The value of DI channel %d
         is %d.\n" , ch , data );
             }

}
```

5.6 _8111_DO

@ *Description*

This function is used to write data to digital output ports. There are 16 digital outputs on the ACL-8111, 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 _8111_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  “aclerr.h”
#include  “8111.h”

main()
{

         _8111_Initial( CARD_1,   0x220 );
         /* Assume NoError when Initialize ACL-8111
         card */
    .
    .
         _8111_DO( DO_LO_BYTE , 0x55 );
         printf( "The low  byte is now 0x55.\n" );

             _8111_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'

5.7 _8111_DA

@ *Description*

This function is used to write data to D/A converters. There is a Digital-to-Analog conversion channel on the ACL-8111. The resolution is 12-bit, i.e. the range is from 0 to 4095.

@ *Syntax*

```
int _8111_DA(unsigned int data )
```

@ *Argument:*

```
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  "aclerr.h"
#include  "8111.h"

main()
{

         _8111_Initial( CARD_1, 0x220 );
              /* Assume NoError when Initialize ACL-
         8111 */
     /* if the hardware setting for DA output range
         is 0~5V */

             _8111_DA(  0x800 );
             printf( "The output voltage is  2.5V
         \n" );

}
```

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

5.8 _8111_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 8 single-ended A/D channels in ACL-8111, so the channel number should be set between 0 to 7 only. The initial state is channel 0 which is a default setting by the ACL-8111 hardware configuration.

@ Syntax

```
int _8111_AD_Set_Channel( int ad_ch_no )
```

@ Argument:

```
ad_ch_no: channel number to perform AD conversion
```

@ Return Code:

```
ERR_NoError
ERR_BoardNoInit
ERR_InvalidADChannel
```

@ Example:

```
#include  "aclerr.h"
#include  "8111.h"

main()
{

         _8111_Initial( CARD_1,  0x220 );
         /* Assume NoError when Initialize ACL-8111
         */

         _8111_AD_Set_Channel( 3 );
         printf( "AD channel 3 is now selected.\n" );

...

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

}
```

5.9 _8111_AD_Set_Gain

@ *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. The initial value of gain is “1’ which is set by the ACL-8111 harware. The relationship between gain and input voltage ranges is specified by following tables:

The initial value of gain is '1‘ and input type is bipolar, which are pre-set by the ACL-8111 hardware. The relationship between analog input voltage range, gain and input mode are specified by following tables:

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

@ *Syntax*

```
int _8111_AD_Set_Gain( int ad_range )
```

@ *Argument:*

```
ad_range: the programmable range of A/D conversion,
          please refer to above tabbs for the
          possible values .
```

@ *Return Code:*

```
ERR_NoError
ERR_BoardNoInit
ERR_AD_InvalidRange
```

@ *Example:*

```
#include  “aclerr.h”
#include  “8111.h”
main()
{
           _8111_Initial( CARD_1,  0x220 );
           /* Assume NoError when Initialize ACL-8111
          */

           _8111_AD_Set_Gain( AD_GAIN_8 );
          printf( "The A/D analog input range is +/-
          0.625V now.\n" );
}
```

5.10 _8111_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-8111 is set as AD_MODE_1 software(internal) trigger with program polling data.

A/D Mode	Description
AD_MODE_0	Software Trigger, Software Polling Transfer
AD_MODE_1	Software Trigger, Interrupt Transfer
AD_MODE_2	External Trigger, Software Polling Transfer
AD_MODE_3	External Trigger, Interrupt Transfer
AD_MODE_4	Timer Trigger, Software Polling Transfer
AD_MODE_5	Timer Trigger, Interrupt Transfer

@ Syntax

```
int _8111_AD_Set_Mode(int irq_no, int ad_mode )
```

@ Argument:

```
irq_no:   interrupt IRQ level
ad_mode: AD trigger and data transfer mode
         ( please to refer above table.)
```

@ Return Code:

```
ERR_NoError
ERR_BoardNoInit
ERR_InvalidMode
```

@ Example:

```
#include  "aclerr.h"
#include  "8111.h"
main()
{

         _8111_Initial( CARD_1 , 0x220 );
         /* Assume NoError when Initialize ACL-8111
         */

         _8111_AD_Set_Mode( IRQ5, AD_MODE_0 );
        printf( "Now, disable internal trigger.\n" );
}
```

5.11 _8111_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 _8111_AD_Aquire(). Please refer to section 6.11.

@ *Syntax*

```
int _8111_AD_Soft_Trig( void )
```

@ *Argument:*

```
None
```

@ *Return Code:*

```
ERR_NoError
ERR_BoardNoInit
```

@ *Example:*

Please refer the example in next section.

5.12 _8111_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 until the data is ready ('data ready' bit becomes low).

@ Syntax

```
int _8111_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  "aclerr.h"
#include  "8111.h"
main()
{
         int  ad_data;
         int  ErrCode;

         _8111_Initial( CARD_1,  0x220 );
         /* Assume NoError when Initialize ACL-8111
         */

             /* Set to software trigger at first*/
             _8111_AD_Set_Mode( IRQ5, AD_MODE_1 );
             /* then trigger the AD */
         _8111_AD_Soft_Trig();
         /* wait for AD data ready then read it */
         ErrCode = _8111_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_DEMO1.C'

5.13 _8111_CLR_IRQ

@ Description

This function is used to clear interrupt request which requested by the ACL-8111. 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 _8111_CLR_IRQ( void )
```

@ Argument

```
None
```

@ Return Code:

```
ERR_NoError
ERR_BoardNoInit
```

5.14 _8111_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 _8111_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 8111_AD_INT_Status(). The function is perform on single A/D channel with fixed analog input range.

@ *Syntax*

```
int _8111_INT_Start( int ad_ch_no, int ad_range,
         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 6.9 for the possible values.
irq_ch_no:IRQ channel number used to transfer AD data,
          the possible value is defined in ' 8111.h '
count:    the numbers 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 timer
          channel #1
c2:       the 16-bit timer frequency divider of timer
          channel #2
```

@ *Return Code:*

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

@ *Example:*

```
See demo Program 'AD_Demo2.C'
```

5.15 _8111_AD_INT_Status

@ *Description*

Since the _8111_AD_INT_Start() function executes in background, you can issue the function _8111_AD_INT_Status to check the status of interrupt operation.

@ *Syntax*

```
int _8111_AD_INT_Status( int *status , int *count )
```

@ *Argument:*

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

@ *Return Code:*

```
ERR_NoError
ERR_BoardNoInit
ERR_AD_INTNotSet
```

@ *Example:*

```
See demo program 'AD_Demo2.C'
```

5.16 _8111_AD_INT_Stop

@ *Description*

This function is used to stop the interrupt data transfer function. After executing this function, the internal AD trigger is disable 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 _8111_AD_INT_Start() itself.

@ *Syntax*

```
int _8111_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'
```

5.17 _8111_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-8111 can not be exceeded 30 KHz,Thus the multiplication of the dividers must be larger than 70.

@ Syntax

```
int _8111_AD_Timer( unsigned int c1 , unsigned int
         c2 )
```

@ Argument:

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

@ Return Code:

```
ERR_NoError
ERR_BoardNoInitERR_InvalidTimerValue
```

@ Example:

```
main()
{
          int ErrCode;

         _8111_Initial( CARD_1, 0x220 );
         /* Assume ERR_NoError when Initialize ACL-
         8111 */

         _8111_AD_Timer( 100 , 100 );
         /* set AD sampling rate to 2MHz/(10*10) */

..

         _8111_AD_Timer( 0 ,  0 );
         /* stop the pacer trigger */
}
```

6

Calibration

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-8111 to an accuracy condition.

Note: Your ACL-8111 board has been carefully calibrated in the factory before it is shipped to you.

6.1 What do you need

Before calibrating your ACL-8111 card, you should prepare some equipment for the calibration:

1. Calibration program: once the program is executed, it will guide you to do the calibration. This program is included in the delivered package.

2. One 5 1/2 digit multi-meter (6 1/2 digital multi-meter is recommended).

3. A voltage calibrator or very stable and noise free DC voltage generator.

6.2 VR Assignments

There are five variable resistors (VR) on the ACL-8111 board to allow you making accurate adjustment on A/D and D/A channels. The function of each VR is specified below:

VR No.	Function
VR1	D/A full scale adjustment
VR2	D/A zero offset adjustment
VR3	A/D offset adjustment
VR4	A/D full scale adjustment
VR5	A/D programmable amplifier offset adjustment

6.3 D/A Calibration

The D/A input reference voltage should be connected to for calibration. You may use the JP1 to select the -5V or -10V reference. The zero offset and full scale of D/A channel can be turned through VR1 and VR2. The full-scale gain of D/A is adjusted by VR1, and zero offsets are adjusted by VR2.

Note: A precision voltmeter to measure the D/A output is very important.

The procedures of calibration are as follows:

Full scale adjustment :

1. Set JP1 to select -5V as reference voltage.

2. Connect VDM (+) to CN1 pin 30 and VDM (-) to A.GND.

3. Write digital data '4095' (0x0FFF) into D/A's digital data Register (BASE+4 and BASE+ 5).

4. Adjust **VR1** until your voltmeter's voltage reading is equal to +5V (reference voltage with opposite sign).

Offset zero adjustment:

1. Write digital data ' 0' (0x0000) to D/A's digital data register (BASE+4 and BASE+ 5).

2. Adjust **VR2** until the your voltmeter's voltage reading is 0V.

6.4 A/D Calibration

Since the ACL-8111 provides versatile A/D input ranges, the calibration on one A/D range may cause a small offset on other ranges. It is suggested that you calibrate the A/D range which you need best accuracy.

The procedures of calibration are as follows:

1. Set the analog input range as: +/- 5V, i.e. the gain = 1.

2. Short the A/D channel 0 (pin 1 of CN1) to ground (GND), and connect the TP1 (+) and TP2 (-) with your DVM. Trim the variable resister VR5 to obtain 0 reading in the DVM.

3. Applied a +5V input signal to A/D channel 0, and trim the VR4 to obtain reading between 4094~4095.

4. Applied a +0V input signal to A/D channel 0, and trim the VR3 to obtain reading flickers between 2048~2049.

5. Repeat step 2 and step 3, adjust VR3 and VR4.

Appendix A. Demo. Programs

In this software diskette, there are 6 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.
DA_DEMO.C:	D/A conversion
DI_DEMO.C:	Read data from digital input channels
DO_DEMO.C:	Write data to digital output channels

Warranty Policy

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

1. Before using ADLINK'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: http://rma.adlinktech.com/policy/.
2. All ADLINK products come with a limited two-year warranty, one year for products bought in China.
   - The warranty period starts on the day the product is shipped from ADLINK's factory.
   - Peripherals and third-party products not manufactured by ADLINK will be covered by the original manufacturers' warranty.
   - For products containing storage devices (hard drives, flash cards, etc.), please back up your data before sending them for repair. ADLINK is not responsible for any loss of data.
   - Please ensure the use of properly licensed software with our systems. ADLINK does not condone the use of pirated software and will not service systems using such software. ADLINK will not be held legally responsible for products shipped with unlicensed software installed by the user.
   - For general repairs, please do not include peripheral accessories. If peripherals need to be included, be certain to specify which items you sent on the RMA Request & Confirmation Form. ADLINK is not responsible for items not listed on the RMA Request & Confirmation Form.
3. Our repair service is not covered by ADLINK'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 fire, earthquakes, floods, lightening, pollution, other acts of God, and/or incorrect usage of voltage transformers.
   - Damage caused by inappropriate storage environments such as with high temperatures, high humidity, or volatile chemicals.

- Damage caused by leakage of battery fluid during or after change of batteries by customer/user.
- Damage from improper repair by unauthorized ADLINK technicians.
- Products with altered and/or damaged serial numbers are not entitled to our service.
- This warranty is not transferable or extendible.
- Other categories not protected under our warranty.

4. Customers are responsible for all fees necessary to transport damaged products to ADLINK.

For further questions, please e-mail our FAE staff: service@adlinktech.com