ACLS-DLL1

Development Kit for
NuDAQ ISA-Bus Digital I/O Cards,
Windows 3.1/95/98/NT

User’s Guide

Manual Rev. 4.10: May 20, 1999

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, ACLS-DLL1 and ACL series product are registered trademarks 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.

Contents

How to Use This Guide

This manual is designed to help you use the ACLS-DLL1 software development kit for NuDAQ ISA-Bus digital I/O cards *ACL-7130*, *ACL-7120*, *ACL-7122*, *ACL-7124*, *ACL-7125*, *ACL-7225*, and *PET-48DIO*. The manual describes how to install DLLs(Synamic Link Libraries) and configure your NuDAQ ISA-Bus digital I/O devices. When you are familiar with the material in this manual, you can begin to use the *ACLS-DLL1 Function Reference Manual*(PDF file included in Manual directory). The ACLS-DLL1 Function Reference Manual contains detailed descriptions of the ACLS-DLL1 functions. You also can use the help file *ACL-DLL1.HLP*, located in your ACLS-DLL1 software, which contains all of the function reference material.

The *ACLS-DLL1 User Guide* is organized as follows:

- Chapter 1, “Introduction to ACLS-DLL1” gives an overview of the contents included in the software package and describes how to install the software and create your applications by using ACLS-DLL1.
- Chapter 2, “Software Overview” briefly describes the function calls ACLS-DLL1 provides.
- Chapter 3, “Sample Programs” describes some sample programs in the software diskette.

1

Introduction to ACLS-DLL1

1.1 What is ACLS-DLL1

ACLS-DLL1 is a package of software drivers for NuDAQ series ISA bus digital I/O cards ACL-7130, ACL-7120, ACL-7122, ACL-7124, ACL-7125, ACL-7225, and PET-48DIO. It includes high performance data acquisition drivers for developing custom applications under Windows 3.1, Windows 95, and Win-NT 4.0. These drivers are DLLs (Dynamic-Link Library) for using under Windows. They can work with any Windows programming language that allows calling a DLL, such as Microsoft Visual C/C++, Microsoft Visual Basic.

Built into these DLLs are sophisticated memory and data buffer management capabilities that free developers from having to deal with those complex issues.

Using these DLLs lets you take advantage of the power and features of Microsoft Windows for your data acquisition application. These include running multiple applications and using extended memory. Also, Visual Basic and ACL-DLL1's DLLs make it easy to create custom user interfaces and graphics.

In addition to the DLL drivers, some sample programs are also provided; you can refer to it and save a lot of programming time and get some other benefits as well.

1.2 Installing ACLS-DLL1

1.2.1 System Requirements

ACLS-DLL1 requires the following minimum configuration:

- An IBM PC/AT or 100% compatible with an 80386 or higher processor
- 8MB of available memory
- A hard disk with enough disk space to install ACLS-DLL1
- A CD ROM drive
- Microsoft Windows 3.1, Windows 95/98, or Win-NT 4.0

Note : ACLS-DLL1/95 can be used both in Windows95 and Windows 98 systems.

- Application development system: Microsoft C/C++ and Windows SDK, Microsoft Visual C/C++, or Microsoft Visual Basic
- ACL-7120, ACL-7130, ACL-7122, ACL-7124, ACL-7125, ACL-7225 or PET-48DIO Digital I/O card

1.2.2 Installation

The Setup program provided by ACLS-DLL1 performs all tasks necessary for installing the ACLS-DLL1 components.

To install ACLS-DLL1:

step 1 Place “ADLink All-In-One Compact Disc” into the appropriate CD driver.

step 2 Click the *Start* button on the Taskbar, and then choose *Run*.

step 3 Type *x*:\setup (*x* identifies the drive that contains the compact disc) in *Open* text box, and then click *OK*.

step4 Setup first displays the main screen. Select *Software Package*.

Step5 Setup then displays the ADLink's software product screen. Select *ACLS-DLL1*.

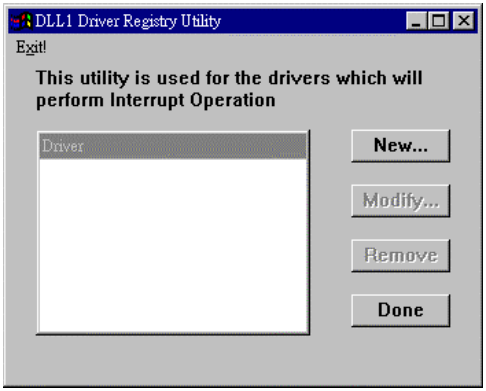
Setup first displays a welcome dialog box. Please click Next button to go on installation.

Setup then prompts a *user information* dialog box including *Name, Company* and *Serial Number* text fields. **The "Serial**

Setup then prompts a dialog box for you to specify the destination directory for ACLS-DLL1. The default path is C:\ADLink\ACL-DLL1. If you want to install ACLS-DLL1 in another directory, please click Browse button to change the destination directory. Then you can click Next button to begin installing ACLS-DLL1.

[Window NT only]

When the software component installation process is completed and the system that DLL1 is installed on is ***Windows NT***, Setup will launch the *Driver Registry Utility* for you to make the registry of the drivers that you want to perform ***interrupt operation***.
The *Driver Registry Utility* first shows the follwing window. If any DLL1 driver has been registered, it will be shown on the *Driver* list.

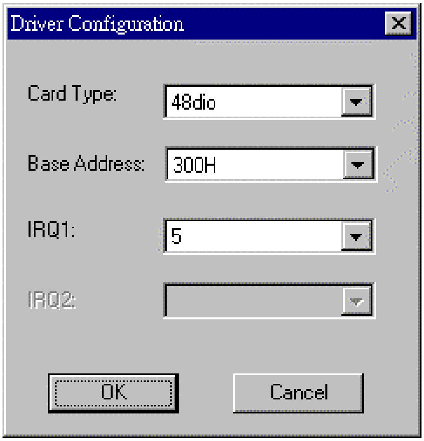

DLL1 Driver Registry Utility

Exit!

This utility is used for the drivers which will perform Interrupt Operation

If the card you wish to operate **DOES NOT** want to perform interrupt operation', please click "Done" or "Exit!" to exit this utility.

Click "New…" button and then a *Driver Configuration* window appears for you to set IRQ levels and base address. Except ACL-7130, the other cards that ACLS-DLL1 supported have only one IRQ line. The *Driver Configuration* window for the cards that support only one interrupt line is as the following figure:


Driver Configuration

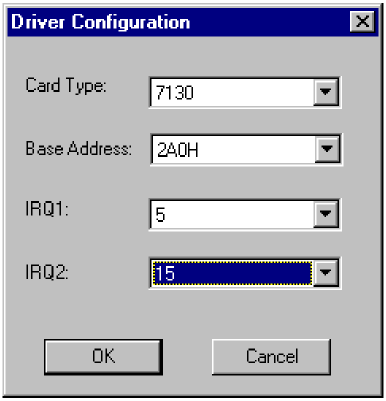
Card Type: 48dio

Base Address: 300H

IRQ1: 5

IRQ2:

ACL-7130 supports two interrupt lines. One is IRQ1 from external digital I/O signal and the other is IRQ2 from internal timer pacer. The “Driver Configuration” window for ACL-7130 is:



Driver Configuration

Card Type: 7130

Base Address: 2A0H

IRQ1: 5

IRQ2: 15

If you don’t need to use both of these two IRQ lines, set the unused IRQ level as “0”. Then you can save one IRQ level for your system.

After the setting for IRQs level and base address, click “OK” to register the driver.

When you finish the dirvers reigstry, select “Done” or “Exit!” to exit this utility. To make the registered drivers work, you have to re-start Windows NT system.

When you have completed the installation process, the DLL1 directory should contain the following files and sub-directories:

File/Subdirectory	Description
LIB <DIR>	ACLS-DLL1 import libraries and DLLs Note: *.*lib* for Microsoft C/C++ and Visual C/C++ programming, e.g. 7120.lib **_bc.lib* for Borland C++ programming, e.g. 7120_bc.lib
INCLUDE <DIR>	include files for application programming *DLL1.BAS* for Visual Basic Programming *DLL1.H* for Visual C/C++ and Borland C++ programming *DLL1.PAS* for Delphi programming
UTIL<DIR> (Windows NT only)	Registry utilities of DLL1's device drivers
Manual<DIR>	PDF manual files, include User Guide and Function Reference
Help<Windows 95 or NT only>	On-line help file
SAMPLES\SDK\48DIO <DIR>	PET-48DIO Visual C/C++ sample program
SAMPLES\SDK\7120 <DIR>	ACL-7120 Visual C/C++ sample program
SAMPLES\SDK\7122 <DIR>	ACL-7122 Visual C/C++ sample program
SAMPLES\SDK\7124 <DIR>	ACL-7124 Visual C/C++ sample program
SAMPLES\SDK\7125 <DIR>	ACL-7125 Visual C/C++ sample program
SAMPLES\SDK\7130 <DIR>	ACL-7130 Visual C/C++ sample program
SAMPLES\SDK\7225 <DIR>	ACL-7225 Visual C/C++ sample program
SAMPLES\SDK\48DIOINT <DIR>	PET-48DIO Interrupt DI/DO Visual C/C++ sample program
SAMPLES\SDK\48DIOIOP <DIR>	PET-48DIO Interrupt DI/DO Visual C/C++ sample program
SAMPLES\SDK\7120INT <DIR>	ACL-7120 Interrupt DI/DO Visual C/C++ sample program
SAMPLES\SDK\7122INT <DIR>	ACL-7122 Interrupt DI/DO Visual C/C++ sample program
SAMPLES\SDK\7122IOP <DIR>	ACL-7122 Interrupt DI/DO Visual C/C++ sample program
SAMPLES\SDK\7124INT <DIR>	ACL-7124 Interrupt DI/DO Visual C/C++ sample program
SAMPLES\SDK\7124IOP <DIR>	ACL-7124 Interrupt DI/DO Visual C/C++ sample program
SAMPLES\SDK\7130INT <DIR>	ACL-7130 Interrupt DI/DO Visual C/C++ sample program
SAMPLES\SDK\7130INT2 <DIR>	ACL-7130 Interrupt DI/DO Visual C/C++ sample program

SAMPLES\VB\48DIO <DIR>	PET-48DIO Visual Basic sample program
SAMPLES\VB\7120 <DIR>	ACL-7120 Visual Basic sample program
SAMPLES\VB\7122 <DIR>	ACL-7122 Visual Basic sample program
SAMPLES\VB\7124 <DIR>	ACL-7124 Visual Basic sample program
SAMPLES\VB\7125 <DIR>	ACL-7125 Visual Basic sample program
SAMPLES\VB\7130 <DIR>	ACL-7130 Visual Basic sample program
SAMPLES\VB\7225 <DIR>	ACL-7225 Visual Basic sample program

All ACLS-DLL1’s DLLs are also copied to Windows System directory (default is C:\WINDOWS\SYSTEM for Windows 3.11 or Win-95, C:\Winnt\System32 for Win-NT):

The driver files are also copies the appropriated directory:

- Win-95 Version: W95_DLL1.VXD is copied to Win-95 System directory.
- Win-NT Version: DIO.SYS, 48DIO.SYS, 7120.SYS, 7122.SYS, 7124.SYS and 7130.SYS are copied to Win-NT System Drivers directory (default is C:\Winnt\System32\Drivers).

1.2.3 ACLS-DLL1 Un-installation

ACLS-DLL1 software has the capability of automatic un-installation.

To un-install ACLS-DLL1, open the “Control Panel”, double-click “Add/Remove Programs”, and select ACLS-DLL1/95 or ACLS-DLL1/NT to un-install it.

1.3 ACLS-DLL1 Hardware Support

The ACLS-DLL1 supports the following hardware:

PET-48DIO	: Programmable 48-bit DIO Event and Timer Card
ACL-7120	: 32 Digital I/O & 6 Counter/Timer Card
ACL-7122	: 144-bit Digital I/O Card
ACL-7124	: 24-bit DIgital I/O Card
ACL-7125	: 8 Channels Relay & 8 Channels Isolated DIO Card
ACL-7130	: Isolated Digital Input & Output Card
ACL-7225	: 16 Channels Relay Actuator & Isolated D/I Card

The old version ACL card can also use this DLL library to program its functionality. The relationship between old version and new version are listed as the following table:

Old ACL Digital I/O cards	**New ACL Digital I/O cards**
ACL-722	ACL-7122
ACL-724	ACL-7124
ACL-725	ACL-7125
ACL-725B	ACL-7225

The register structure and format of old version ACL digital I/O cards are the same as new version cards. For example, users can use all of the functions in the ACL-7122‘s DLL library to program ACL-722 directly.

1.4 ACLS-DLL1 Language Support

ACLS-DLL1 are DLLs (Dynamic-Link Library) for use under Windows 3.1/95/98/NT. It can work with any Windows programming language that allows calls to a DLL, such as Microsoft Visual C/C++ (4.0 or above), Borland C++ (5.0 or above), or Microsoft Visual Basic (4.0 or above), etc.

1.5 ACLS-DLL1 System Architecture

The following diagram shows the interface between your applications and ACLS-DLL1 drivers:

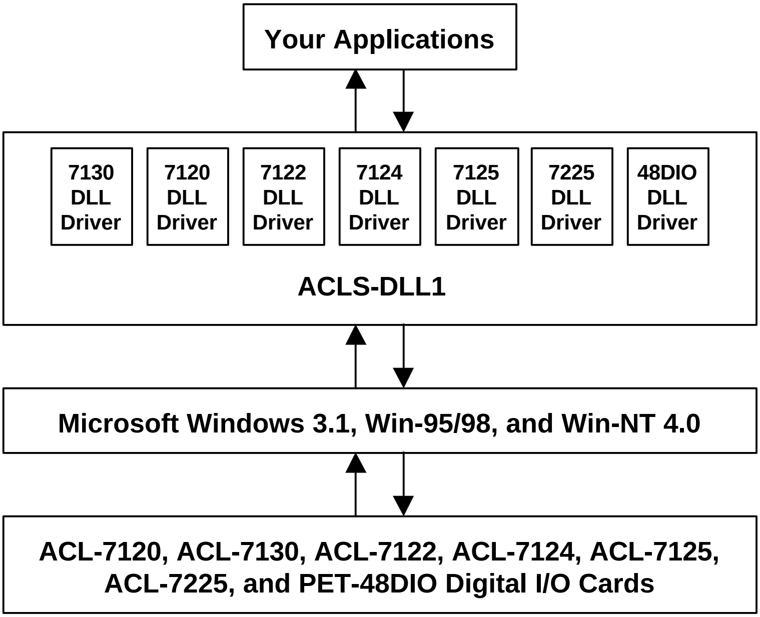
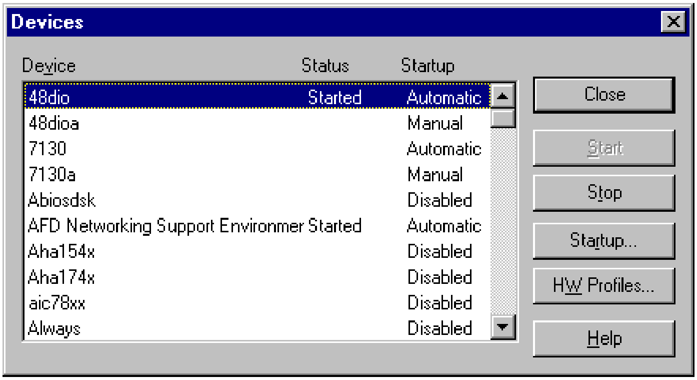

Your Applications
7130 DLL Driver
7120 DLL Driver
7122 DLL Driver
7124 DLL Driver
7125 DLL Driver
7225 DLL Driver
48DIO DLL Driver
ACLS-DLL1
Microsoft Windows 3.1, Win-95/98, and Win-NT 4.0
ACL-7120, ACL-7130, ACL-7122, ACL-7124, ACL-7125,
ACL-7225, and PET-48DIO Digital I/O Cards

Figure 1.1

1.6 Device Driver Handling under Win-NT

When you completed the ACLS-DLL1 installation in Win-NT, please be careful of the following issue:

Make sure the required driver “*DIO*” for ACLS-DLL1 and the registered device drives (48dio, 7122, 7124, 7120 or 7130) have already been started. You can press the “Control Panel”, double-click “Devices”, and a Devices windows will be shown as below:



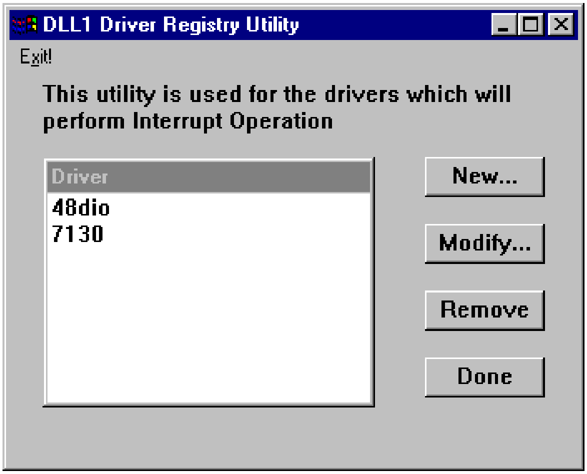
Device	Status	Startup
48dio	Started	Automatic
48dioa		Manual
7130		Automatic
7130a		Manual
Abiosdsk		Disabled
AFD Networking Support Environmer	Started	Automatic
Aha154x		Disabled
Aha174x		Disabled
aic78xx		Disabled
Always		Disabled

If the device status is none, you have to select the device (DIO, 48dio, 7122, 7124, 7120 or 7130) and then press the “ Start” button.

Note : If your device driver can not be started, please check if the resources (I/O Port Address, IRQ Level, or DMA Channel) conflict with other hardware device. You can change the resource setting by the procedures described in section 1.6.1.
48dio, 7122, 7124, 7120 or 7130 is only needed for interrupt operation. If you do not perform interrupt operation, this driver is not needed.

1.6.1 ACLS-DLL1/NT Device Driver Configuration

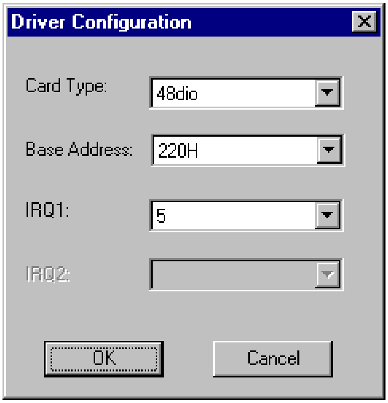
ACLS-DLL1 provides a utility, *Driver Registry Utility*. This utility is used for users to register new DLL1 drivers (the drivers you want to perform interrupt operation), remove installed drivers and view/modify the base address and IRQ level settings of installed drivers.


DLL1 Driver Registry Utility
Exit!
This utility is used for the drivers which will perform Interrupt Operation
Driver
48dio
7130
New...
Modify...
Remove
Done

The *Driver Registry Utility* is installed with ACLS-DLL1 and located in <InstallDir>\Util directory.

Using this utility to install a new driver, please refer to section 1.2.2.

Using *Driver Registry Utility* to view/change IRQ level or Base Address of ACLS-DLL1 device drivers, select the the driver from the Driver list and click “Modify…” button and then a “Driver Configuration” window is shown as below:


Driver Configuration

Card Type: 48dio

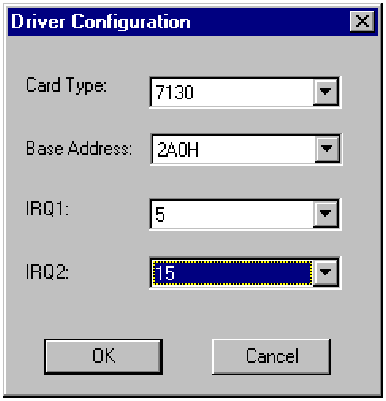
Base Address: 220H

IRQ1: 5

IRQ2:

Inside the Base Address and IRQ fields are the originally set values. Modify the values and then click "OK" button. The settings for the drivers will be changed as you modified.

ACL-7130 can support two interrupt interrupt lines. One is IRQ1 from external digital I/O signal and the other is IRQ2 from internal timer pacer. The "Driver Configuration" window for ACL-7130 is:


Driver Configuration

Card Type: 7130

Base Address: 2A0H

IRQ1: 5

IRQ2: 15

If you don't need to use both of these two IRQ lines, set the unused IRQ level as "0". Then you can save one IRQ level for your system.

To remove a registered driver, select the driver from the Driver list and click "Remove" button. The selected driver will be deleted from the registry table.

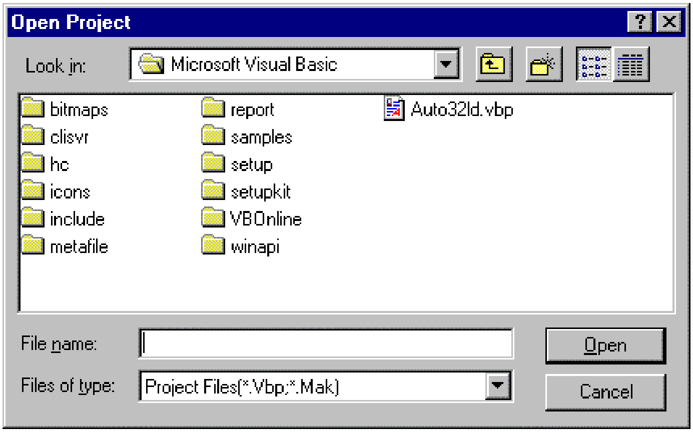
1.7 Using ACLS-DLL1

1.7.1 Creating an Application Using Visual Basic and ACLS-DLL1

To create a data acquisition application using ACLS-DLL1 and Visual Basic, follow these steps after entering Visual Basic:

step 1 Open the project in which you want to use ACLS-DLL1. This can be a new or existing project

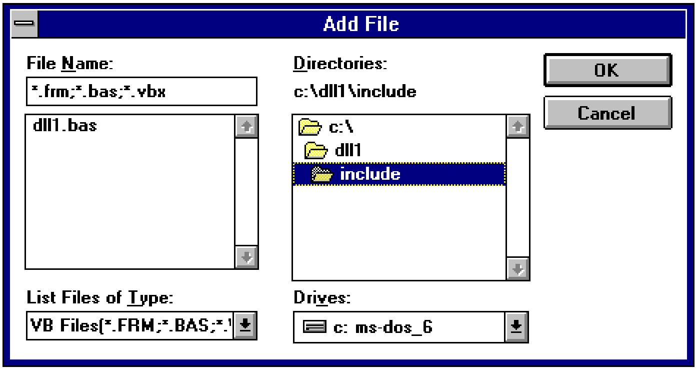
Open a new project by selecting the New Project command from the File menu. If it is an existing project, open it by selecting the Open Project command from the File menu. Then the Open Project dialog box appears.



Changed directory to the place the project file located. Double-click the project file name in the File Name list to load the project.

step 2 Add file DLL1.BAS into the project if this file is not included in the project. This file contains all the procedure declarations and constants that you can use to develop your data acquisition application.

From the File menu, select the Add File command. The Add File window appears, displaying a list of files in the current directory.



File Name: *.frm;*.bas;*.vbx

dll1.bas

Directories: c:\dll1\include

c:\
dll1
include

List Files of Type: VB Files(*.FRM;*.BAS;*.!

Drives: c: ms-dos_6

Select DLL1.BAS from the Files list by double-clicking on it. If you can't find this file in the list, make sure the list is displaying files from the correct directory. By default, DLL1.BAS is installed in C:\ACL-DLL1\INCLUDE.

step 3 Design the interface for the application.

To design the interface, you place the desired elements, such as command button, list box, text box, etc., on the Visual Basic form. These are standard controls from the Visual Basic Toolbox. To place a control on a form, you just move pointer to Toolbox, select the desired control and draw it on the form. Or you can double-click the control icon in the Toolbox to place it on the form.

step 4 Set properties for the controls.

To view the property list, click the desired control and then choose the Properties command from the Window menu or press F4, you can also click the Properties button on the toolbar.

step 5 Write the event code.

The event code defines the action you want to perform when an event occurs. To write the event code, you double click the desired control or form to view the code module then add code you want. You can call the procedures that declared in the file DLL1.BAS to perform data acquisition operations.

step 6 Run your application.

To run the application, choose Start from the Run menu, or click the Start icon ▶ on the toolbar (you can also press F5).

step 7 Distribute your application.

Once you have finished a project, you can save the application as an executable (.EXE) file by using the Make EXE File command on the File menu. And once you have saved your application as an executable file, you've ready to distribute it. When you distribute your application, remember also to include the ACLS-DLL1's DLL and driver files. These files should be copied to their appropriated directory as section 2.1.3 described.

1.7.2 Creating an Application Using Microsoft C/C++, Windows SDK, and ACLS-DLL1

To create a data acquisition application using ACLS-DLL1, Microsoft Visual C/C++, follow these steps after entering Visual C/C++:

step 1 Open the project in which you want to use ACLS-DLL1. This can be a new or existing project

step 2 Include header files DLL1.H in the C/C++ source files that call ACLS-DLL1 functions. DLL1.H contains all the function declarations and constants that you can use to develop your data acquisition application. Incorporate the following statement in your code to include the header file.

```
#include  “DLL1.H”
```

step 3 Build your application

Setting the appropriate compile and link options, then build your application by selecting the Build command from Build menu (Visual C/C++ 4.0) or Project menu (Visual C/C++ 1.52). Remember to link appropriate ACLS-DLL1’s import libraries.

2

Software Overview

Each NuDAQ digital I/O card has its own DLL driver. How to use these DLL to build your own application has been described in section 1.7. The function calls in these DLLs use intuitive names that reflect the operations they perform. For example, W_7122_DI performs *Digital Input* for *ACL-7122* card.

The functionality of these function calls can be classified to the following capabilities,

1. Initialization : set the hardware base I/O address
2. Digital I/O : input or output digital signals
3. Interrupt operation: input or output digital signals through interrupt operation
4. Timer/Counter : Timer/Counter operation

In addition, some sample programs are also included in this disk. They help you to understand how to use the driver more quickly. The ACLS-DLL1 contains seven digital I/O cards' DLL drivers, they are 48DIO.DLL, 7120.DLL, 7122.DLL, 7124.DLL, 7125.DLL, 7130.DLL, and 7225.DLL. The detailed description of each function in the DLL driver is specified in the following sections.

2.1 Software Driver Naming Convention

The functions of ACL-DLL1 use full-names to represent the real meaning of the functions. The naming convention rules are:

In DOS Environment :

_{hardware_model}_{action_name}. e.g. **_7122_Initial** ().

In order to recognize the difference between DOS library and Windows library, a capital "**W**" is put on the head of each function name of the Windows DLL driver. e.g. **W_7122_Initial** ()

2.2 Initialization and General Configuration Functions

{hardware_model}_Initial	Initializes the hardware and the software status of a NuDAQ ISA-Based digital I/O card according to the card number, the corresponding base address and IRQ level.
{hardware_model}_Set_Card	Sets the status of the card you want to operate to be active in a multi-cards system.
{hardware_model}_Get_Card	Gets the card number which is operated currently in a multi-cards system.

2.3 Digital I/O Functions

{hardware_model}_DI	Reads the digital data from the specified input port.
{hardware_model}_DI_Channel	Reads the digital data from the specified input channel.
{hardware_model}_DO	Writes the digital data to the specified output port.
{hardware_model}_DO_Channel	Writes the digital data to the specified output line.
{hardware_model}_Read_Back	Read back the digital data from the specified output channel.

2.4 Interrupt Operation Functions

{hardware_model}_INT_Enable	Initializes and starts up the interrupt control.
{hardware_model}_INT_Disable	Stops interrupt signal generation.
{hardware_model}_INTOP_Start	Performs digital input or output N times with interrupt data transfer by using external/internal interrupt trigger..
{hardware_model}_INTOP_Staus	Checks the current status of the interrupt operation.
{hardware_model}_INTOP_Stop	Stops the interrupt data transfer.
{hardware_model}_INT_Op	Set up the operation type of each port when the interrupt is triggered.
{hardware_model}_INT_Reset	Reset all ports' interrupt transfer operation types to be NO_OP (no operation).

2.5 Timer/Counter Operation Functions

{hardware_model}_Count_Start
Starts up Counter #0 to operate in the specified mode.

{hardware_model}_Count_Status
Reads the current contents of Counter #0.

{hardware_model}_Count_Stop
Stops Counter #0 operation.

{hardware_model}_INT_Timer_Start
Start up the internal timers for generating constant interrupt trigger signal dedicatedly.

{hardware_model}_INT_Timer_Stop
Stop the internal timers .

{hardware_model}_Timer_Read
Reads the current contents of the selected counter.

{hardware_model}_Timer_Start
Starts up the specified counter to operate in the specified mode..

{hardware_model}_Timer_Stop
Stop the specified counter operation.

3

Sample Programs

3.1 Sample Programs Included

There are several sample programs provided in this software diskette. They could help you to program your own applications by using ACLS-DLL1 easily. The brief descriptions of these programs are specified as follows:

SDK 48DIO /	D/I and D/O of PET-48DIO
VB 48DIO	Microsoft C/C++ Program
	Visual Basic Program
SDK 7120 /	D/I and D/O of ACL-7120
VB 7120	Microsoft C/C++ Program
	Visual Basic Program
SDK 7122 /	D/I and D/O of ACL-7122
VB 7122	Microsoft C/C++ Program
	Visual Basic Program
SDK 7124 /	D/I and D/O of ACL-7124
VB 7124	Microsoft C/C++ Program
	Visual Basic Program

SDK 7125 / VB 7125	D/I and D/O of ACL-7125 Microsoft C/C++ Program Visual Basic Program
SDK 7130 / VB 7130	D/I and D/O of ACL-7130 Microsoft C/C++ Program Visual Basic Program
SDK 7225 / VB 7225	D/I and D/O of ACL-7225 Microsoft C/C++ Program Visual Basic Program
SDK 48DIOINT (1)	D/I and D/O of PET-48DIO through Interrupt operation (Using internal timer pacer) Microsoft C/C++ Program
SDK 48DIOINT (2)	D/I and D/O of PET-48DIO through Interrupt operation (Using internal timer pacer) Microsoft C/C++ Program
SDK 7122INT (1)	D/I and D/O of ACL-7122 through Interrupt operation (Using a test output signal to emulate an interrupt) Microsoft C/C++ Program
SDK 7122INT (2)	D/I and D/O of ACL-7122 through Interrupt operation (Using a test output signal to emulate an interrupt) Microsoft C/C++ Program
SDK 7124INT (1)	D/I and D/O of ACL-7124 through Interrupt operation (Using a test output signal to emulate an interrupt) Microsoft C/C++ Program

SDK 7124INT (2)
D/I and D/O of ACL-7124 through Interrupt operation (Using a test output signal to emulate an interrupt)
Microsoft C/C++ Program

SDK 7120INT
D/I and D/O of ACL-7120 through Interrupt operation (Using inernal timer pacer)
Microsoft C/C++ Program

SDK 7130INT
D/I and D/O of ACL-7130 through Interrupt operation (using higher irq: internal timer pacer)
Microsoft C/C++ Program

SDK 7130INT2
D/I and D/O of ACL-7130 through Interrupt operation (using both irq lines: external digital I/O signal and internal timer pacer)
Microsoft C/C++ Program

3.2 Sample Programs Developed Environment

3.2.1 Visual Basic Sample Programs

There are seven Visual Basic sample programs provided in this software package. By default, they are located in directory C:\ACL-DLL1\SAMPLES\VB. The following files are included in each sample program (Using VB 7120 as an example):

- VB project file --- Util7120.VBP
- VB form files --- Util7120.FRM,Util7120.FRX
- Executable file --- UTIL7120.EXE

You must have Microsoft Visual Basic 4.0 Professional Edition or above to deal with these sample programs. Please refer to Visual Basic Manual or related reference books to get the information about how to use Visual Basic 4.0.

3.2.2 Microsoft C/C++ Sample Programs

We provide fifteen Microsoft C/C++ sample programs in this package. By default, they are located in directory C:\ACL-DLL1\SAMPLES\SDK. The following files are included in each sample program (Using SDK 7120 as an example):

- C source file --- UTIL7120.C
- Workspace file --- UTIL7120.MDP
- Resource script file --- UTIL7120.RC, RESOURCE.H
- Make file --- UTIL7120.MAK
- Executable file --- UTIL7120.EXE

You can use any editor or Microsoft Visual C++ 4.0 to view or modify these source files. However, to build the executable UTIL7120.EXE, you must have Microsoft Visual C++ 4.0 or above. Please refer to Visual C++ Manual or related reference books to get the information about how to use Visual C++ 4.0.

3.3 Execute Sample Programs

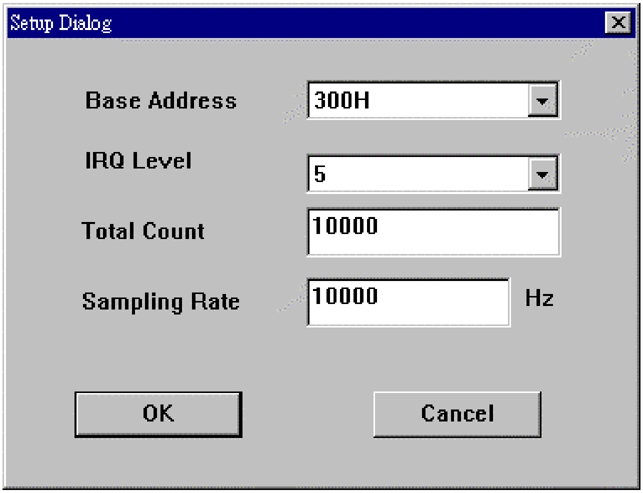
To run the sample programs, please follow these steps (Using SDK48DIO/SDK48DIOINT as example):

step 1 Open the sample program

You can use Microsoft Visual C++ 4.0 or Visual Basic 4.0 to open and execute the sample programs. Or you can run the executable files directly.

step 2 Configuration Setting

According to the configuration setting on your NuDAQ card, input the related Base Address and IRQ of the card in “setup” dialog box (Figure 3.1) or in the main screen (Figure 3.2).



Base Address 300H

IRQ Level 5

Total Count 10000

Sampling Rate 10000 Hz

Figure 3.1

Note : If your environment is Windows NT and your irq level is not the same as the default value, please refer to section 2.4 to learn how to change irq level.

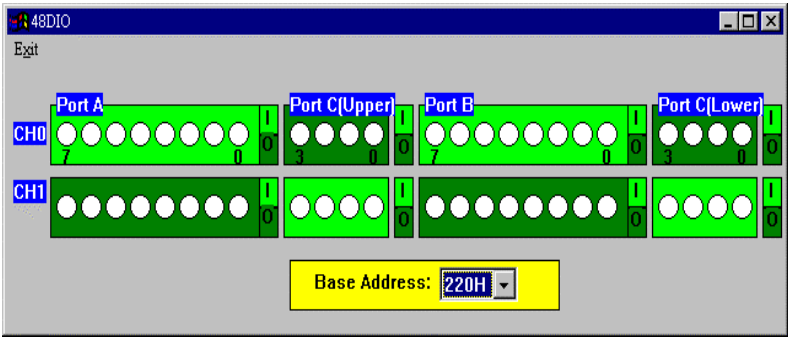


Figure 3.2

step 3 Option Setting

According to your requirements, select the input/output channels and ports (Figure 3.2), sampling rate or interrupt count (Figure 3.1). To set the I/O status of each port, click the “I” (Input) or “O” (Output) button at the right side of each port (Figure 4.2).

step 4 Push “interrupt trigger” button or click output channel lights to run the program.

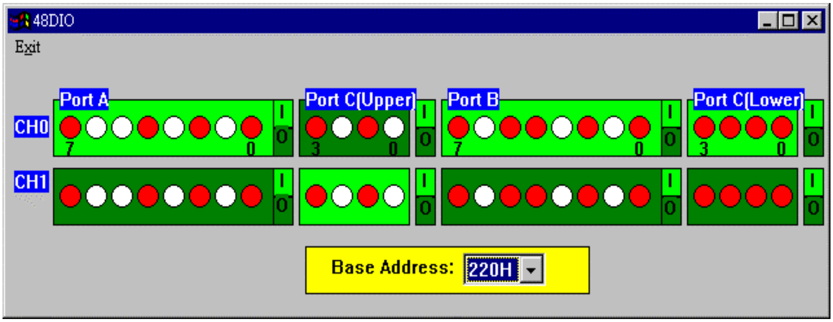
3.4 The Detailed Descriptions of these Sample Programs

There are two kinds of sample programs provided in this software package. The descriptions of these two types are the following (Using the screens of SDK48DIO and SDK 48DIOINT as the figure examples) :

3.4.1 D/I and D/O

This kind of samples are used to demonstrate how to use ACLS-DLL1 to Read/Write data from digital input/output channels with program polling. To set the output value, click the channel lights. The input data are shown by the channel lights. The red light means “on” and the white light means “off”.

The main screen of this kind of programs is shown below (Figure 3.3):

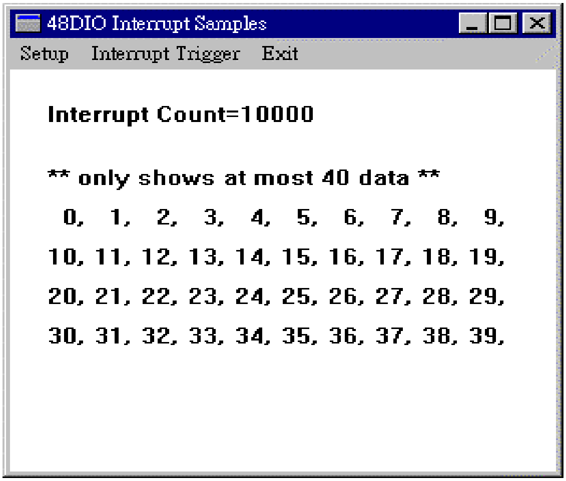


Base Address: 220H

Figure 3.3

3.4.2 DI and DO through Interrupt

This kind of programs are used to demonstrate how to use ACLS-DLL1 to operate data transfer through Interrupt operation. The screen of this kind of programs is shown below (Figure 3.4) :



```
Interrupt Count=10000

** only shows at most 40 data **
 0,  1,  2,  3,  4,  5,  6,  7,  8,  9,
10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
20, 21, 22, 23, 24, 25, 26, 27, 28, 29,
30, 31, 32, 33, 34, 35, 36, 37, 38, 39,
```

Figure 3.4

In this kind of programs you can select total count and Sampling Rate (for PET-48DIO, ACL-7120 and ACL-7130) as you wish. Part of the input data are shown in the main screen as data transfer operation is finished. Since there is no 8253 Counter/Timer on board for ACL-7122 and ACL-7124, the interrupt signals of these two cards are received from external peripherals or generated by software. The 7122INT and 7124INT sample programs provided in this software diskette demonstrate how to use software to generate interrupt signals. To generate an interrupt signal, push “interrupt trigger” button. If the total count of I/O operations is 10, you should press this button 10 times. However, there is a 8253 timer/counter on PET-48DIO, ACL-7120 and ACL-7130 boards. The 48DIOINT 7120INT and 7130INT sample programs provided in this software diskette demonstrate how to use internal timer to generate interrupt signals. After setting the total count of interrupt, you just need to press “interrupt trigger” button once to start the timer interrupt.